MODELLISTICA E CALCOLO SCIENTIFICO

MODELING AND SCIENTIFIC COMPUTING

MOX–Report No. 25/2010

High Order Methods for the Approximation of the Incompressible Navier-Stokes Equations in a Moving Domain

G. Pena, C. Prud'homme, Alfio Quarteroni

MOX, Dipartimento di Matematica "F. Brioschi"
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

mox@mate.polimi.it http://mox.polimi.it

High Order Methods for the Approximation of the Incompressible Navier-Stokes Equations in a Moving Domain

G. Pena[1], C. Prud'homme[2] and A. Quarteroni[3,4]

[1] DMUC, University of Coimbra
Largo D. Dinis, Apartado 3008, 3001-454 Coimbra, Portugal

[2] Laboratoire Jean Kuntzmann, Université Joseph Fourier Grenoble 1
BP 53 38041 Grenoble Cedex 9, France

[3] SB-MATHICSE-CMCS, École Polytechnique Fédérale de Lausanne
MA C2 573 (Bâtiment MA), Station 8, CH-1015 Lausanne, Switzerland

[4] MOX, Politechico di Milano,
Piazza Leonardo da Vinci, 32-20133 Milan, Italy

e-mails: *gpena@mat.uc.pt*, *christophe.prudhomme@ujf-grenoble.fr*, *alfio.quarteroni@epfl.ch*

July 5, 2010

Abstract

In this paper we address the numerical approximation of the incompressible Navier-Stokes equations in a moving domain by the spectral element method and high order time integrators. We present the Arbitrary Lagrangian Eulerian (ALE) formulation of the incompressible Navier-Stokes equations and propose a numerical method based on the following kernels: a Lagrange basis associated with Fekete points in the spectral element method context, BDF time integrators, an ALE map of high degree, and a robust algebraic linear solver. In particular, the high degree ALE map is appropriate to deal with a computational domain whose boundary is described with curved elements. Finally, we apply the proposed strategy to a test case.

1 Introduction

The accurate approximation of the incompressible Navier-Stokes equations for flows in moving domains is an important subject of research in applied mathematics. This type of problem appears in many important fluid dynamics applications, including fluid-structure interaction problems [8, 37, 11, 33] or free surface flows [22, 3]. The main difficulties of simulating this problem are:

(i) how to discretize the system of equations in a domain that evolves in time, see [23, 9];

(ii) the techniques to solve the associated algebraic system.

The first item is related with the problem formulation and its space/time discretization. Since the domain changes its shape in time, a common technique to keep track of its evolution is the Arbitrary Lagrangian Eulerian (ALE) frame [23, 9]. The latter introduces a vector function that represents the domain velocity of deformation. Its numerical approximation has been discussed in the context of the spectral element method, Ho and Rønquist [22] and Bouffanais [3], or the finite element method, Nobile [30]. Another option is the map sketched in Pena and Prud'homme in [33] that we present in this paper in full detail. A relevant aspect in devising numerical schemes in the ALE framework is the so called *Geometric Conservation Law* (GCL). A numerical scheme satisfies the GCL if it can represent a constant solution through time. Although it is neither a necessary nor a sufficient condition for convergence/stability of the schemes, in some cases, the fulfillment of the GCL implies stability independently of the domain's rate of deformation, see Nobile [30].

Regarding the space discretization, the starting point in discretizing in space the Navier-Stokes equations (in a fixed domain) in the primitive variable formulation is the choice of discrete spaces for velocity and pressure. It is a well known fact that the discrete velocity and pressure spaces cannot be chosen independently. Indeed, a discrete compatibility condition enforces that a certain gap must exist between these spaces. If such a condition is violated, then the linear system associated with the discretization fails to have a unique solution. This is the so called Brezzi-Babuska-Ladizenskaya *inf-sup condition*, see Quarteroni and Valli [41]. In the literature one can find a few possible choices of spaces that fulfill such condition. For some examples, see Bernardi and Maday [2], Schwab and Suri [45], Ainsworth and Coggins [1] and Stenberg and Suri [49]. For an extensive analysis, see Brezzi and Fortin [4]. In the context of the spectral element method, it is known that, for instance, choosing velocities as continuous polynomials of degree N and pressures as piecewise discontinuous polynomials of degree N or $N-1$ violates the inf-sup condition, see Bernardi and Maday [2]. At an algebraic level, this violation is reflected by the existence of non-constant pressures (defined all over the domain) whose discrete gradient is zero, leading to the non uniqueness of the solution of the Stokes/Navier-Stokes equations. One of the most popular and widely used discretizations that is free of spurious pressure modes was studied by Bernadi and Maday [2] and Rønquist [42]. It consists of approximating the velocities with polynomials of degree N and pressures with piecewise discontinuous polynomials of degree $N-2$. However, the corresponding error estimates are not optimal regarding the polynomial order of the approximation spaces. This is due to the fact that the inf-sup constant decreases as the polynomial order increases. A comparison of the approximation properties of these spaces can be found in Pena and Prud'homme [33].

In the context of the spectral element method, the work by Patera in [31] provided the bases for the modern *multidomain spectral method*. This version of the SEM pushes the method to deal with arbitrary geometries, combining its spectral properties with the flexibility of the finite element method. Although in the beginning it was only applied to geometries that were partitioned into quadrangular sub-domains, the monography by Sherwin and Karniadakis [25] provided a further extension of this method to geometries that could be partitioned into simplices, thus giving even more flexibility to the SEM. The definition of global basis functions can be done using Lagrange polynomials associated with suitable point sets. In the literature, several point sets have been proposed, for tensorized and simplicial domains. The Gaussian points (Gauss, Gauss-Radau and Gauss-Lobatto) are usually employed to construct Lagrange bases in tensorized geometries due to their well behaved Lebesgue constants. For simplicial domains, there is no equivalent of the Gaussian points. The Equidistributed points, see [5], are a first alternative, but these do not have low Lebesgue constants and, in the context of the Galerkin method, they lead to very ill conditioned linear systems, see Pena [32]. Other choices in the triangular case, more robust with respect to interpolation, are the Electrostatic [21], Fekete [50], Heinrichs [20] and more recently, Warpblend [51] points. A very interesting property shared by the Electrostatic, Fekete and Warpblend points is that, on the edges of the triangle where they are defined, they coincide with the Gauss-Lobatto points. This feature allows the use of hybrid meshes (composed of quadrangles and triangles) in a continuous Galerkin setting. For the triangular spectral element method, the Fekete points are usually a good choice since they provide good numerical stability properties to the linear systems involved, see Pena [32].

Apart from the space discretization problem, several solution strategies have been proposed to solve the unsteady Navier-Stokes equations. We highlight two of them: (i) *fully coupled methods* and (ii) *splitting methods*. Splitting methods decouple the calculation of the velocity and pressure field, by performing a splitting, either in the differential equations, see for instance [17, 18, 19], or at the algebraic level, see [39, 40, 44, 13, 12]. Such decoupling of the variables makes the calculation of the solution faster, however at the cost of introducing some error in the approximation, called *splitting error*. The differential type of splitting also introduces an artificial boundary condition (that needs to be derived) for the pressure operator. On the other hand, algebraic splitting methods do not have this requirement. Fully coupled algorithms do not introduce splitting error. Instead, they try to solve the fully coupled velocity-pressure system of equations, e.g. by the Uzawa approach or with a suitable preconditioner for the whole linear system, see [27, 28, 46]. See Canuto, Hussaini, Quarteroni and Zang [5, 6] for an extensive discussion.

In the following sections, we propose a numerical strategy to solve the incompressible unsteady Navier-Stokes equations set in a moving domain. In section 2, we present the equations written in the ALE frame of reference. Regarding the space discretization, the spectral element method is briefly introduced in section 3.1. The description of a high order ALE map, responsible, at each time step, for describing the computational domain where the Navier-Stokes equations are to be solved, is detailed in section 3.2. In the following sections 3.3 and 3.4, the fully discrete numerical method is presented. Then we introduce a combination of Backward Differentiation Formulas (BDF) and an extrapolation formula of the same order to fully discretize in time the system of equations. The same BDFq formula is used to approximate the mesh velocity associated with the ALE map. Once the differential system is fully discretized and a linear system is obtained (see section 3.5), we consider three approaches to solve it: a LU factorization, the GMRES method combined with an ILU factorization or a block type preconditioner, see section 3.6. Section 4 is dedicated to illustrate the numerical convergence properties of the several strategies proposed. In section 4.3, we use a LU factorization to solve the system or the extension of the Yosida-q schemes to the ALE context and compare the order of convergence achieved by using both strategies.

We remark that all the computations in this paper were done with the FEEL++ (Finite Element Embedded Library in C++), formely known as the LIFE library, see [35, 34, 36, 33].

2 Differential problem

Let us denote by Ω_{t_0} a reference configuration, for instance, the domain filled by the fluid at time $t = t_0$ in which we want to solve the Navier-Stokes equations. The position of a point in the current domain Ω_t, $t > t_0$, is denoted by $\mathbf{x}$ (in the Eulerian coordinate system) and by $\mathbf{Y}$ in the reference domain Ω_{t_0}. The system's evolution is studied in the interval $I = [t_0, T]$.

We introduce a family of mappings $\mathcal{A}_t$ that for each t, assoaciates to a point $\mathbf{Y} \in \Omega_{t_0}$ a point $\mathbf{x} \in \Omega_t$:

$$\mathcal{A}_t : \Omega_{t_0} \longrightarrow \Omega_t, \quad \mathbf{x}(\mathbf{Y}, t) = \mathcal{A}_t(\mathbf{Y}), \quad t \in I. \tag{1}$$

For every t, $\mathcal{A}_t$ is assumed to be an homeomorphism in $\overline{\Omega}_{t_0}$, i.e., $\mathcal{A}_t$ is a continuous bijection from the closure $\overline{\Omega}_{t_0}$ onto $\overline{\Omega}_t$, as well as its inverse, from $\overline{\Omega}_t$ onto $\overline{\Omega}_{t_0}$. We also assume that the application

$$t \mapsto \mathbf{x}(\mathbf{Y}, t), \quad \mathbf{Y} \in \Omega_{t_0}$$

is differentiable almost everywhere in I. The application $\mathcal{A}_t$ is called *ALE map*.

Let $f : \Omega_t \times I \longrightarrow \mathbb{R}$ be a function defined in the Eulerian frame, and $\hat{f} := f \circ \mathcal{A}_t$ the corresponding function defined in the ALE framework, defined as

$$\hat{f} : \Omega_{t_0} \times I \longrightarrow \mathbb{R}, \quad \hat{f}(\mathbf{Y}, t) = f(\mathcal{A}_t(\mathbf{Y}), t) \tag{2}$$

and conversely,

$$f(\mathbf{x}, t) = \hat{f}(\mathcal{A}_t^{-1}(\mathbf{x}), t).$$

Another ingredient is the ALE time derivative of f, defined as

$$\left.\frac{\partial f}{\partial t}\right|_{\mathbf{Y}} : \Omega_t \times I \longrightarrow \mathbb{R}, \qquad \left.\frac{\partial f}{\partial t}\right|_{\mathbf{Y}}(\mathbf{x}, t) = \frac{\partial \hat{f}}{\partial t}(\mathcal{A}_t^{-1}(\mathbf{x}), t).$$

We then define the *domain velocity of deformation* as

$$\mathbf{w}(\mathbf{x}, t) = \left.\frac{\partial \mathbf{x}}{\partial t}\right|_{\mathbf{Y}}. \tag{3}$$

In the ALE framework, the unsteady incompressible Navier-Stokes equations read as

$$\begin{aligned} \rho \left.\frac{\partial \mathbf{u}}{\partial t}\right|_{\mathbf{Y}} - \mathbf{div}_{\mathbf{x}}(2\nu \mathbf{D}_{\mathbf{x}}(\mathbf{u})) + \rho((\mathbf{u}-\mathbf{w})\cdot \boldsymbol{\nabla}_{\mathbf{x}})\mathbf{u} + \nabla_{\mathbf{x}} p &= \mathbf{f}, \quad \text{in } \Omega_t \times I & (4)\\ \mathrm{div}_{\mathbf{x}}(\mathbf{u}) &= 0, \quad \text{in } \Omega_t \times I & (5)\end{aligned}$$

where all differential operators are defined w.r.t. the Eulerian coordinate system, except the ALE time derivative. Without loss of generality, we suppose that the *strain tensor* is linear and defined as

$$\mathbf{D}_{\mathbf{x}}(\mathbf{u}) = \frac{1}{2}\left(\boldsymbol{\nabla}_{\mathbf{x}}\mathbf{u} + (\boldsymbol{\nabla}_{\mathbf{x}}\mathbf{u})^T\right).$$

The constant ρ is the *density* of the fluid. For simplicity of the exposition, we will consider homogeneous Dirichlet on Γ_t^D and Neumann boundary conditions on Γ_t^N. These subsets of the boundary satisfy $\partial\Omega_t = \Gamma_t^D \cup \Gamma_t^N$, $\Gamma_t^D \cap \Gamma_t^N = \emptyset$.

In order to derive the weak formulation for problem (4)-(5), we introduce function spaces for trial and test functions built with the ALE map $\mathcal{A}_t$ and spaces defined in the reference domain. Let $\mathbf{V}(\Omega_t)$ and $Q(\Omega_t)$ be defined as

$$\mathbf{V}(\Omega_t) = \left\{\mathbf{v} : \Omega_t \times I \longrightarrow \mathbb{R}^d, \quad \mathbf{v} = \hat{\mathbf{v}} \circ \mathcal{A}_t^{-1}, \quad \hat{\mathbf{v}} \in \mathbf{H}^1_{\Gamma^D}(\Omega_{t_0})\right\} \tag{6}$$

and

$$Q(\Omega_t) = \left\{q : \Omega_t \times I \longrightarrow \mathbb{R}, \quad q = \hat{q} \circ \mathcal{A}_t^{-1}, \quad \hat{q} \in L^2(\Omega_{t_0})\right\}. \tag{7}$$

where $\mathbf{H}^1_{\Gamma^D}(\Omega_{t_0})$ is the subset of $\mathbf{H}^1(\Omega_{t_0})$ whose functions are vector valued and have zero trace on $\Gamma^D = \mathcal{A}_t^{-1}(\Gamma_t^D)$.

For $\mathbf{u}, \mathbf{v}, \boldsymbol{\beta} \in \mathbf{V}(\Omega_t)$ and $p, q \in Q(\Omega_t)$, we introduce the following notations

$$\begin{aligned} (\mathbf{u},\mathbf{v})_{\Omega_t} &= \int_{\Omega_t} \mathbf{u}\cdot\mathbf{v}\; dx\\ a\,(\mathbf{u},\mathbf{v})_{\Omega_t} &= 2\nu\int_{\Omega_t} \mathbf{D}_{\mathbf{x}}(\mathbf{u}) : \boldsymbol{\nabla}_{\mathbf{x}}\mathbf{v}\; dx\\ b\,(\mathbf{v},p)_{\Omega_t} &= \int_{\Omega_t} \mathrm{div}_{\mathbf{x}}(\mathbf{u})\; p\; dx\\ c\,(\mathbf{u},\mathbf{v};\boldsymbol{\beta})_{\Omega_t} &= \rho\int_{\Omega_t} [\boldsymbol{\beta}\cdot\boldsymbol{\nabla}_{\mathbf{x}}]\,\mathbf{u}\cdot\mathbf{v}\; dx.\end{aligned}$$

With these notations, the weak formulation of the Navier-Stokes equations in the ALE framework reads as follows

Problem 2.1. *For almost every $t \in I$, find $\mathbf{u}(t) \in \mathbf{V}(\Omega_t)$, with $\mathbf{u}(t_0) = \mathbf{u}_0$ in Ω_{t_0} and $p(t) \in Q(\Omega_t)$, such that*

$$\begin{aligned} \rho\left(\left.\frac{\partial \mathbf{u}}{\partial t}\right|_{\mathbf{Y}}, \mathbf{v}\right)_{\Omega_t} + c\,(\mathbf{u},\mathbf{v};\mathbf{u}-\mathbf{w})_{\Omega_t} + a\,(\mathbf{u},\mathbf{v})_{\Omega_t} + b\,(\mathbf{v},p)_{\Omega_t} &= (\mathbf{f},\mathbf{v})_{\Omega_t}, \quad \forall \mathbf{v} \in \mathbf{V}(\Omega_t)\\ b\,(\mathbf{u},q)_{\Omega_t} &= 0, \qquad\quad \forall q \in Q(\Omega_t)\end{aligned} \tag{8}$$

3 Numerical approximation

3.1 Construction of the spectral element space

We address now the discretization of the system of equations (8) and start by introducing some concepts and notations.

3.1.1 Notations and preliminaries

We denote by $\hat{\Omega}$ a reference element, which is either a $d-$*simplex* (interval, triangle, tetrahedron)

$$\mathcal{T}^d = \left\{(x_1,\ldots,x_d) \in \mathbb{R}^d \mid -1 < x_1,\ldots,x_d < 1, \quad x_1+\cdots+x_d < 0\right\}$$

or a $d-$*hypercube* (interval, quadrangle, hexahedron)

$$\mathcal{Q}^d = \left\{(x_1,\ldots,x_d) \in \mathbb{R}^d \mid -1 < x_1,\ldots,x_d < 1\right\},$$

where d is the topological dimension of Ω_{t_0}. In $\hat{\Omega}$ we build the polynomial spaces $\mathbb{P}_N\left(\mathcal{T}^d\right)$ and $\mathbb{Q}_N\left(\mathcal{Q}^d\right)$, corresponding respectively to the space of polynomials of total degree smaller or equal than N and the space of polynomials of degree smaller or equal than N, for $d = 1, 2, 3$.

Given an element, say Ω_e in a triangulation of Ω_{t_0}, we define the geometrical mapping $\boldsymbol{\varphi}_e : \hat{\Omega} \longrightarrow \Omega_e$, that is a polynomial in $\mathbb{P}_{N_{\text{geo}}}\left(\mathcal{T}^d\right)$, if $\hat{\Omega} = \mathcal{T}^d$ or $\mathbb{Q}_{N_{\text{geo}}}\left(\mathcal{Q}^d\right)$, if $\hat{\Omega} = \mathcal{Q}^d$. We assume N_{geo} as being the smallest positive integer such that $\boldsymbol{\varphi}_e$ is a homeomorfism.

We are now ready to define the polynomial spaces necessary for the spectral element method.

3.1.2 The spectral element space

Let $\mathcal{T}_{t_0,\delta}$ be a triangulation of the reference domain Ω_{t_0} into N_{el} elements that we denote by Ω_e, where $\delta = (h, N_{\text{geo}})$. Here, h denotes the maximum diameter of all the elements in the partition and N_{geo} is the polynomial degree of the geometrical mapping associated with each element in the partition.

Remark 3.1. *If δ is replaced only by h, then it is implicitly understood that the geometrical transformation $\boldsymbol{\varphi}_e$ is of degree 1, that is $N_{geo} = 1$.*

Let $\Omega_{t_0,\delta}$ be a domain, obtained by the union of all elements in the triangulation $\mathcal{T}_{t_0,\delta}$, that approximates Ω_{t_0}. Note that $\Omega_{t_0,\delta}$ and Ω_{t_0} may not coincide since the triangulation only approximates the domain.

The domain $\Omega_{t_0,\delta}$ and the elements of the triangulation $\mathcal{T}_{t_0,\delta}$ satisfy the following assumptions:

- $\overline{\Omega_{t_0,\delta}} = \bigcup_{e=1}^{N_{el}} \overline{\Omega_e}$;
- $\Omega_e \cap \Omega_i$ is empty whenever $e \neq i$;
- two neighbor subdomains can only share vertices, edges or faces;
- Ω_e is the image of a reference element by a geometrical mapping of the type described in section 3.1.1. We assume that N_{geo} is the same for every Ω_e.

Remark 3.2. *The domain $\Omega_{t_0,\delta}$ could be discretized in elements that are a mix of triangles and quadrangles, for example, in 2D. However, without loss of generality, we will consider that all triangulations are made of triangles in 2D, or tetrahedra in 3D, and consider $\hat{\Omega} = \mathcal{T}^d$, $d = 2, 3$, to generate the spectral element space.*

We define the *spectral element space* as

$$\mathcal{F}_N(\mathcal{T}_{t_0,\delta}) = \left\{v \in C^0(\overline{\Omega_{t_0,\delta}}) : \; v_{|\Omega_e} \in \mathbb{P}_N(\Omega_e), \; \forall \Omega_e \in \mathcal{T}_{t_0,\delta}\right\} \tag{9}$$

where $\mathbb{P}_N(\Omega_e)$ is the space

$$\mathbb{P}_N(\Omega_e) = \left\{p : \; p = \hat{p} \circ \boldsymbol{\varphi}_e^{-1}, \; \hat{p} \in \mathbb{P}_N(\hat{\Omega})\right\}. \tag{10}$$

In the following, for this space we build the Lagrange nodal basis associated with Fekete points, see [50]. This set of points, also called *high order nodes*, is associated with the space $\mathcal{F}_N(\mathcal{T}_{t_0,\delta})$. They are obtained by collecting, for each Ω_e, the image of the Fekete points in the reference element through the geometrical mapping. Details on the construction of function bases for this space can be found in Pena [32].

3.2 Construction of the discrete ALE map

We denote by $\Omega_{t,\delta}$ the discrete computational domain where the Navier-Stokes equations are to be solved, at time t and

$$\mathbf{g}_{t,\delta} : \partial\Omega_{t_0,\delta} \longrightarrow \partial\Omega_{t,\delta}$$

the map that transforms the boundary of $\Omega_{t_0,\delta}$ onto the boundary of $\Omega_{t,\delta}$ (which we assume known *a priori*). The discrete ALE map $\mathcal{A}_{t,\delta}$ then satisfies

$$\mathcal{A}_{t,\delta}{}_{|_{\Omega_{t_0,\delta}}} = \mathbf{g}_{t,\delta}, \qquad \mathcal{A}_{t,\delta}(\Omega_{t_0,\delta}) = \Omega_{t,\delta}. \tag{11}$$

To build $\mathcal{A}_{t,\delta}$, and given the fact that in practice we only have a description of the boundary of $\Omega_{t,\delta}$, we start by dividing the boundary $\partial\Omega_{t,\delta}$ into two parts: $\partial\Omega^{\mathcal{D}}_{t,\delta}$ a part of the boundary that remains fixed in time and $\partial\Omega^{\sigma}_{t,\delta}$ the part the moves with t. Clearly, $\partial\Omega_{t,\delta} = \partial\Omega^{\mathcal{D}}_{t,\delta} \cup \partial\Omega^{\sigma}_{t,\delta}$. Let us assume that we have a description of $\partial\Omega^{\sigma}_{t,\delta}$ in terms of polynomials of degree N.

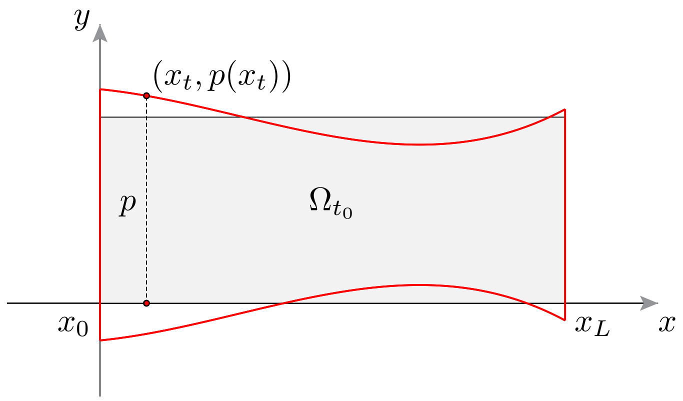


Figure 1: Description of the reference (shaded and rectangular region) domain $\Omega_{t_0,\delta}$ and computational domain $\Omega_{t,\delta}$ (curved region). The top and bottom boundaries of the computational domain are described in terms of polynomials.

We then make the following assumptions: (i) the upper and lower parts of the boundary $\partial\Omega^{\sigma}_{t,\delta}$ are described by polynomials p of degree N defined in $[x_0, x_L]$; (ii) $\Omega_{t_0,\delta}$ can be covered exactly by a triangulation composed of elements with straight edges. This assumption is not as restrictive as it seems because mesh generators can typically create triangulations for $N_{\text{geo}} = 1, 2$.

Remark 3.3. *Some popular open source mesh generators, such as GMSH, provide high order mesh generation, see [14]. This means that the mesh generator can produce triangulations such that $N_{geo} > 2$.*

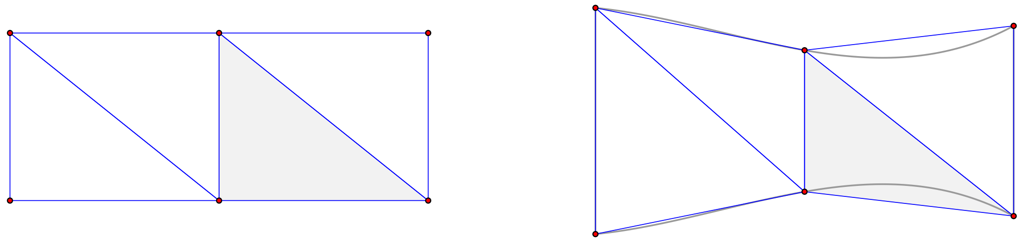
3.2.1 Standard harmonic extension

Then, given the description of the boundary in terms of polynomials, we perform a standard harmonic extension of the function $\mathbf{g}_{t,\delta}$ by solving

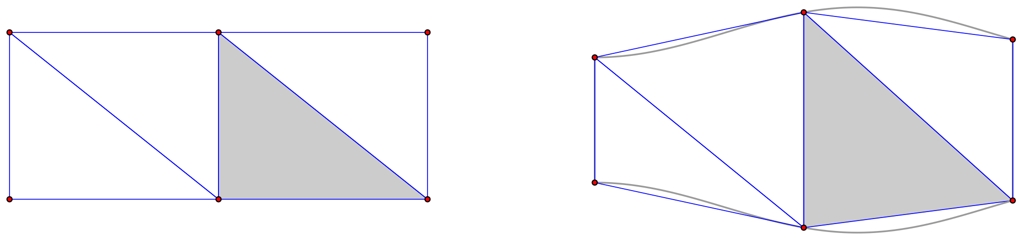
Problem 3.1. *Find $\mathcal{A}_{t,\delta_1} \in (\mathcal{F}_1(\mathcal{T}_{t_0,\delta}))^d$ such that*

$$\begin{cases} \displaystyle\int_{\Omega_{t_0,\delta}} \nabla\mathcal{A}_{t,\delta_1} : \nabla\mathbf{z}\; dx = 0, & \forall\mathbf{z} \in (\mathcal{F}_1(\mathcal{T}_{t_0,\delta}))^d \\ \mathcal{A}_{t,\delta_1} = \mathbf{g}_{t,\delta}, & \textit{on } \partial\Omega_{t_0,\delta} \end{cases} \tag{12}$$

We obtain an ALE map $\mathcal{A}_{t,\delta_1}$, where $\delta_1 = (h, 1)$, that once applied to the triangulation $\mathcal{T}_{t_0,\delta}$ generates a mesh $\mathcal{T}_{t,\delta_1}$ for $\Omega_{t,\delta}$, ie, $\mathcal{T}_{t,\delta_1} = \mathcal{A}_{t,\delta_1}(\mathcal{T}_{t_0,\delta})$. This process is depicted in Figure 2. To make the exposition of the following steps easier, we fix an element in $\mathcal{T}_{t_0,\delta}$ and its image through the transformation $\mathcal{A}_{t,\delta_1}$, say K_{t_0} and K_t, respectively. In Figure 2 we can identify these elements with the shaded elements on the left and right columns, respectively. In the same figure, we illustrate two different possible configurations using this technique, which we shall use throughout the presentation of this method.



(a) Configuration 1.



(b) Configuration 2.

Figure 2: Transformation from the triangulation in the reference domain $\Omega_{t_0,\delta}$ (left) to the triangulation in the computational domain, $\mathcal{T}_{t,\delta_1}$ through the map $\mathcal{A}_{t,\delta_1}$. On the right, the straight edges correspond to the triangulation $\mathcal{T}_{t,\delta_1}$ and the curved boundaries correspond to $\Omega_{t,\delta}$.

3.2.2 Introducing high order nodes

The next step is to project $\mathcal{A}_{t,\delta_1}$ onto the space $\left(\mathcal{F}_{N_{\text{geo}}}(\mathcal{T}_{t_0,\delta})\right)^d$. Let $\mathcal{B} = \{\boldsymbol{\phi}_i\}_i$ be a nodal basis for this space (in our simulations, $\mathcal{B}$ is the Lagrange basis associated with Fekete points). With these notations, the projection of $\mathcal{A}_{t,\delta_1}$ onto $\left(\mathcal{F}_{N_{\text{geo}}}(\mathcal{T}_{t_0,\delta})\right)^d$, denoted $\mathcal{A}^*_{t,\delta}$, is

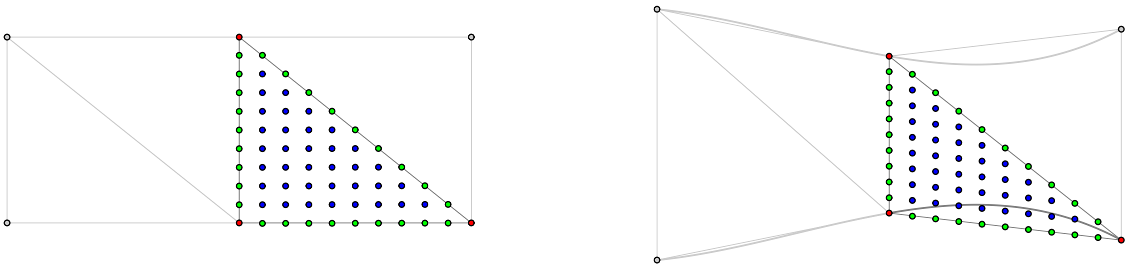
$$\mathcal{A}^*_{t,\delta} = \sum_i \alpha_i \boldsymbol{\phi}_i$$

where the coefficients α_i are determined by interpolating $\mathcal{A}_{t,\delta_1}$ over the nodal points associated with $\left(\mathcal{F}_{N_{\text{geo}}}(\mathcal{T}_{t_0,\delta})\right)^d$. Since the basis is nodal, these coefficients are no more than the evaluation of $\mathcal{A}_{t,\delta_1}$ at these nodes.

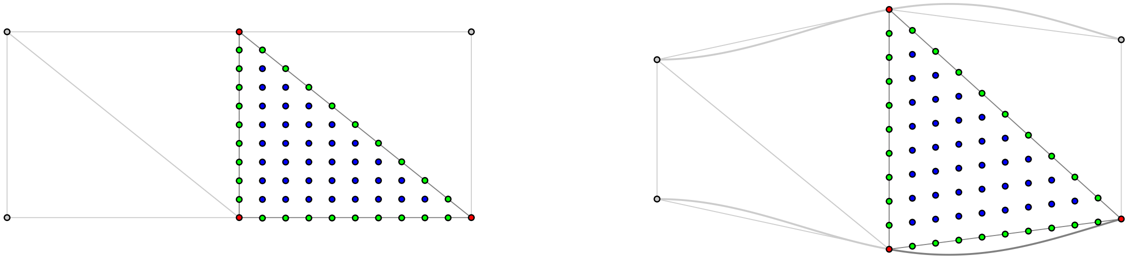
We now change the value of the degrees of freedom, associated with edges that are in contact with the curved wall. If $\mathbf{x}_0$ is a point belonging to the mentioned edge that corresponds to a degree of freedom and (x_t, y_t), see Figure 1, are the coordinates of its image through $\mathcal{A}^*_{t,\delta}$, then we take

$$\mathcal{A}^*_{t,\delta}(\mathbf{x}_0) = (x_t, p(x_t)).$$

This shifting in the coordinates solves the problem of making the edges of the elements conform with the curved boundary. However, this might create a map that has a singular Jacobian since part of the interior of the element in the reference domain is mapped outside the corresponding



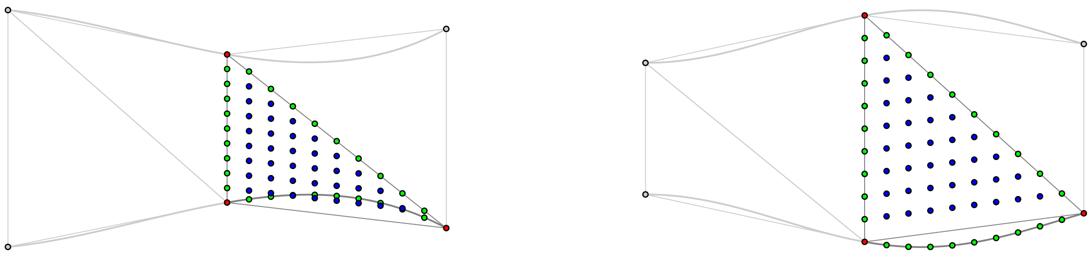
(a) Configuration 1.



(b) Configuration 2.

Figure 3: Effect of $\mathcal{A}^*_{t,\delta}$ on a equidistributed point set defined in an element of the reference mesh.

element in the computational mesh, see Figure 4(a). It may be also that there are no points mapped outside the element and the transformation is valid, see Figure 4(b), however the approximation may be poor.



(a) Configuration 1. (b) Configuration 2.

Figure 4: Effect of $\mathcal{A}^*_{t,\delta}$ if only the degrees of freedom in the edges are shifted.

3.2.3 Shifting the nodes on edges and faces

The final step to obtain a valid ALE map is to shift also the nodes on the faces of the elements of the reference domain and obtain a situation as in Figure 5. The coordinates of the new nodes are obtained using a transformation of Gordon-Hall type, see Gordon-Hall [15, 16], Pena [32] and Canuto Hussaini Quarteroni and Zang [5].

Let us assume that the edges, denoted Γ_i, of the curved elements in $\Omega_{t,\delta}$, denoted K_t^{curved}, are parameterized by functions $\pi_i : [-1,1] \longrightarrow \Gamma_i$. We assume that the parameterizations verify $\pi_0(-1) = \pi_2(1)$, $\pi_1(-1) = \pi_0(1)$ and $\pi_2(-1) = \pi_1(1)$, see Pena [32] for more details.

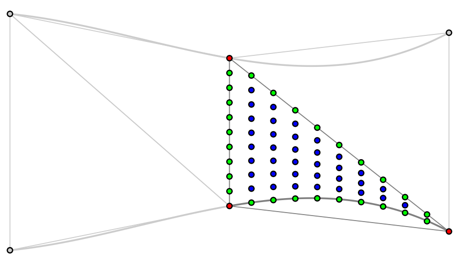
A transformation that extends smoothly the boundary mappings to the interior of K_t^{curved} can be found, for instance, in Canuto, Hussaini, Quarteroni and Zang [5] and Pena [32]. Here, we use the one from the latter manuscript. The idea is to define a transformation from the reference element, $\hat{\Omega}$, onto K_t^{curved}. The transformation has the form

$$\begin{aligned}
\varphi_{K_t^{curved}}(\xi,\eta) \quad = \quad & \frac{1-\eta}{2}\pi_2(\xi) - \frac{1+\xi}{2}\pi_2(-\eta) + \frac{1-\xi}{2}\pi_1(-\eta) - \frac{1+\eta}{2}\pi_1(\xi) \\
& + \left(1 + \frac{\xi+\eta}{2}\right)\pi_0(-\xi) - \frac{1+\xi}{2}\pi_0(-1-\xi-\eta) \\
& + \frac{1+\xi}{2}\pi_2(1) + \frac{\xi+\eta}{2}\pi_1(1),
\end{aligned}$$

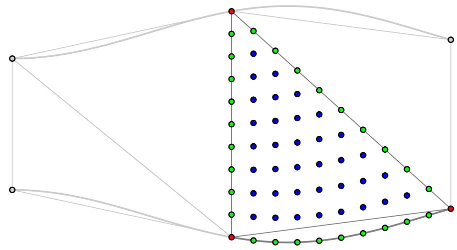
for all $(\xi,\eta) \in \hat{\Omega}$.

Let us denote the geometrical transformation from the element onto K_t by φ_{K_t} and M_i the set of high order nodes belonging to the topological subentity of dimension i. This means that M_0 are the vertices of K_t, M_1 are the nodes on the edges of K_t and M_2 are the high order nodes that need to be shifted. First, we apply $\varphi_{K_t}^{-1}$ to $M = M_0 \cup M_1 \cup M_2$. We obtain a set of points that lie exactely in $\hat{\Omega}$. Moreover, $\varphi_{K_t}^{-1}(M_0)$ are the vertices of $\hat{\Omega}$, $\varphi_{K_t}^{-1}(M_1)$ lie on the edges of $\hat{\Omega}$ and $\varphi_{K_t}^{-1}(M_2)$ lie on the face of $\hat{\Omega}$. We know *a priori*, which edges from K_t are curved and build parametrizations of them. Therefore, we can construct $\varphi_{K_t^{curved}}$ and apply it to $\varphi_{K_t}^{-1}(M_2)$. Like this, we obtain a new set of points, which are the shifted high order nodes in the element K_t^{curved}. Let us denote this new set of points by M_2^{curved}. The final stage is to replace the value of the degrees of freedom lying in the face of K_t in the map $\mathcal{A}_{t,\delta}^*$ with the values of M_2^{curved}. Let $\mathcal{A}_{t,\delta}$ denote the updated map.

Remark 3.4. *Although we did not make any considerations about the orientation of the vertices, edges of faces of the elements, all the previous transformations respect that orientation and therefore replacing the values in M_2 by the ones in M_2^{curved} is sufficient to build the correct ALE map. See Pena [32] for more details regarding the orientation of the elements of a triangulation.*



(a) Configuration 1.



(b) Configuration 2.

Figure 5: Transformation from the reference mesh to the computational one (update on face's degrees of freedom).

Remark 3.5. *In Pena [32] the deduction of a similar transformation for triangulations composed with quadrangles is also done. For the three dimensional case, we refer the reader to Sherwin and Karniadakis [25] or Solin, Segeth and Dolezel [47].*

Remark 3.6. *We highlight that the construction just presented does not depend on the extension operator that was used to generate the first mesh in the computational domain. Other procedures can be applied, see Bouffanais [3].*

A numerical study of the approximation properties of the ALE map just presented is done in section 4.1.

Advantages and disadvantages. A consequence of the definition of the map just presented is that it is affine for elements that do not share an edge with the curved boundary. For these elements T,

$$\mathcal{A}_{t,\delta_{|T}} = \mathcal{A}_{t,\delta_1}{}_{|T}.$$

This also means that the geometric mapping associated with these elements is affine. This situation is illustrated in Figure 6. The advantage of this property is that when integrating linear/bilinear

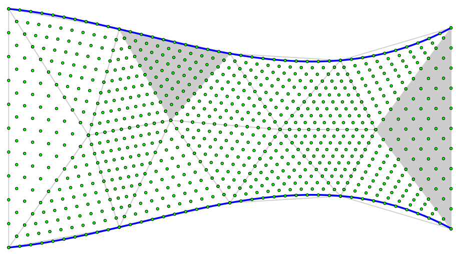


Figure 6: The effect of the mapping $\mathcal{A}_{t,\delta}$ to an equidistributed point set in the reference domain (Configuration 1). The shadowed elements are, from left to right, triangles where the geometrical transformation is of high degree or linear, respectively.

forms in these elements, a constant Jacobian is associated with the geometrical transformation and therefore a minimal order quadrature can be used (in the sense that the geometrical transformation does not need to be taken into account). Only the elements that intersect the curved boundary, the quadrature order has to account for the non constant Jacobian.

A final remark concerns possible strategies in the case the boundary's deformation is "large", as it can happen that an interior straight edge of the mesh intersects the curved boundary. This

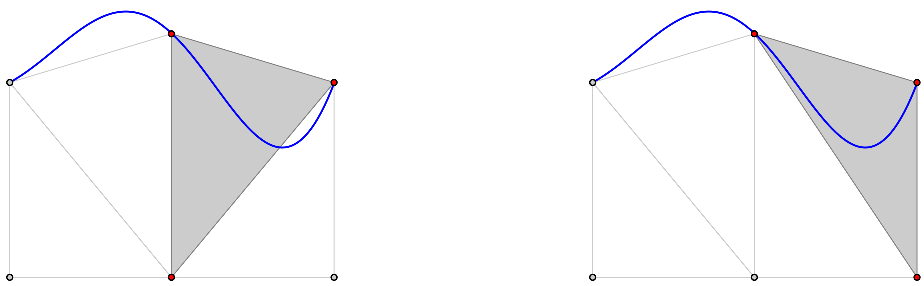


Figure 7: Invalid element created due to the high distortion in the boundary (left). Edge swap technique to correct possible invalid elements (right).

originates an invalid element and several techniques are proposed in literature to deal with this issue, see Sherwin and Karniadakis [26]. However, since we might now want to change the structure of our reference mesh, other possibilities consist of using a control function in the Laplace operator of the harmonic extension to follow the boundary movement, see for instance Kanchi and Masud [24], or employing a reference mesh that is refined near the curved boundary. A final, more costly possibility, is to re-mesh the whole domain. In our approach, since the displacements we consider are small, re-meshing is not necessary.

3.3 Variational formulation of the semi-discrete problem

We address now the discretization in space of the system of equations (8). Let

$$\mathbf{V}_\delta(\Omega_{t,\delta}) = \left\{ \mathbf{v} : \Omega_{t,\delta} \times I \longrightarrow \mathbb{R}^d, \quad \mathbf{v} = \hat{\mathbf{v}} \circ \mathcal{A}_{t,\delta}^{-1}, \quad \hat{\mathbf{v}} \in \mathbf{H}^1_{\Gamma^D}(\Omega_{t_0}) \cap (\mathcal{F}_N(\mathcal{T}_{t_0,\delta}))^d \right\} \tag{13}$$

and

$$Q_\delta(\Omega_{t,\delta}) = \left\{ q : \Omega_{t,\delta} \times I \longrightarrow \mathbb{R}, \quad q = \hat{q} \circ \mathcal{A}_{t,\delta}^{-1}, \quad \hat{q} \in \mathcal{F}_M(\mathcal{T}_{t_0,\delta}) \right\} \tag{14}$$

where $N \geqslant 2$ and $M = N-1$ or $M = N-2$. $\mathbf{V}_\delta(\Omega_{t,\delta})$ and $Q_\delta(\Omega_{t,\delta})$ are the finite dimensional function spaces in which velocity and pressure will be discretized for any time $t > 0$, respectively.

We introduce the *semi-discrete domain velocity*, $\mathbf{w}_\delta$, defined as

$$\mathbf{w}_\delta(\mathbf{x}, t) = \frac{\partial \mathcal{A}_{t,\delta}}{\partial t} \circ \mathcal{A}_{t,\delta}^{-1}, \ \forall \mathbf{x} \in \Omega_{t,\delta}, \ t > 0. \tag{15}$$

Since the construction of the discrete ALE map lies upon the discretization of a differential problem for a given mesh, we also call this quantity *mesh velocity.*

The semi-discrete variational problem reads as

Problem 3.2. *For almost every $t \in I$, find $\mathbf{u}_\delta(t) \in (\mathcal{F}_N(\mathcal{T}_{t,\delta}))^d$, with $\mathbf{u}_\delta(t_0) = \mathbf{u}_{0,\delta}$ in $\Omega_{t_0,\delta}$ and $p_\delta(t) \in Q(\Omega_{t,\delta})$, such that*

$$\begin{aligned} \rho \left(\left.\frac{\partial \mathbf{u}_\delta}{\partial t}\right|_{\mathbf{Y}}, \mathbf{v} \right)_{\Omega_{t,\delta}} + c\left(\mathbf{u}_\delta, \mathbf{v}; \mathbf{u}_\delta - \mathbf{w}_\delta\right)_{\Omega_{t,\delta}} + & \\ a\left(\mathbf{u}_\delta, \mathbf{v}\right)_{\Omega_{t,\delta}} + b\left(\mathbf{v}, p\right)_{\Omega_{t,\delta}} &= (\mathbf{f}, \mathbf{v})_{\Omega_{t,\delta}}, \quad \forall \mathbf{v} \in \mathbf{V}_\delta(\Omega_{t,\delta}) \\ b\left(\mathbf{u}, q\right)_{\Omega_{t,\delta}} &= 0, \quad \forall q \in Q_\delta(\Omega_{t,\delta}) \end{aligned} \tag{16}$$

Remark 3.7. *To enhance the stability of the spatial discretization, we add to the first equation of system* (16) *the quantity $s\left(\mathbf{u}_\delta, \mathbf{v}; \mathbf{u}_\delta\right)_{\Omega_{t,\delta}}$ defined by*

$$s\left(\mathbf{u}, \mathbf{v}; \boldsymbol{\beta}\right)_{\Omega_t} = \frac{\rho}{2} \int_{\Omega_t} \mathrm{div}_{\mathbf{x}}(\boldsymbol{\beta}) \mathbf{u} \cdot \mathbf{v} \ dx.$$

This term is consistent with the Navier-Stokes equations, since at the fully continuous level, $\mathrm{div}_{\mathbf{x}}(\mathbf{u}) = 0$. *For a more detailed explanation of the importance of adding such term to the formulation in the context of stability, see Nobile [30].*

Interior penalty stabilization Often the fluid flows are dominated by the convection hence a suitable stabilization has to operated on the variational formulation. In our approach, we consider the *interior penalty* (IP) stabilization technique. Let us first introduce some notations. Let $\mathcal{F}_I$ be the set of internal faces of a triangulation $\mathcal{T}_\delta$, where $\delta = (h, N_{\mathrm{geo}})$. Given a face $F \in \mathcal{F}_I$, let T_1 and T_2 be the elements of $\mathcal{T}_\delta$ that share F, that is, $F = T_1 \cap T_2$. Let $v \in H^1(\Omega_\delta)$ and $\mathbf{v} \in \mathbf{H}^1(\Omega_\delta)$. We denote by v_1, v_2, respectively, $\mathbf{v}_1, \mathbf{v}_2$ the restrictions of v and $\mathbf{v}$ to the elements T_1 and T_2. Let $\mathbf{n}_1$ and $\mathbf{n}_2$ be the exterior normals of T_1 and T_2. Then, the *jumps* of v and $\mathbf{v}$ across F are defined as

$$\llbracket v \rrbracket_F = v_1 \mathbf{n}_1 + v_2 \mathbf{n}_2 \tag{17}$$

$$\llbracket \mathbf{v} \rrbracket_F = \mathbf{v}_1 \cdot \mathbf{n}_1 + \mathbf{v}_2 \cdot \mathbf{n}_2. \tag{18}$$

In the case of the tensor function $\boldsymbol{\nabla} \mathbf{v}$, we define the jump as

$$\llbracket \boldsymbol{\nabla} \mathbf{v} \rrbracket_F = \boldsymbol{\nabla} \mathbf{v}_1 \mathbf{n}_1 + \boldsymbol{\nabla} \mathbf{v}_2 \mathbf{n}_2.$$

The stabilization term to be added to the variational formulation reads

$$j\left(\mathbf{u}, \mathbf{v}; \boldsymbol{\beta}\right)_{\Omega_\delta} = \gamma \sum_{F \in \mathcal{F}_I} \int_F |\boldsymbol{\beta} \cdot \mathbf{n}| \frac{h_F^2}{N^{3.5}} \llbracket \boldsymbol{\nabla} \mathbf{u} \rrbracket_F \cdot \llbracket \boldsymbol{\nabla} \mathbf{v} \rrbracket_F \ ds \tag{19}$$

where h_F denotes the length of the face F and N the degree of the velocity approximation and γ is the *stabilization parameter.* In the system of equations (16), the term to add (to the first equation) to account for the IP stabilization is $j\left(\mathbf{u}_\delta, \mathbf{v}; \mathbf{u}_\delta - \mathbf{w}_\delta\right)_{\Omega_{t,\delta}}$.

3.4 Time integration

We start by approximating the time derivative by a *backward differentiation formula* of order q (BDFq) and linearize the nonlinear convective term by an extrapolation formula of order q. Given $\Delta t \in (0, T)$, we set $t_0 = 0$, $t_n = t_0 + n\Delta t$ (for any $n \geqslant 1$) and $N_T = \left[\frac{T}{\Delta t}\right]$ (ie, the integer part of $\frac{T}{\Delta t}$); then

Problem 3.3. *For each $n \geqslant q-1$, we look for the solution $(\mathbf{u}_\delta^{n+1}, p_\delta^{n+1}) \in \left(\mathcal{F}_N(\mathcal{T}_{t_{n+1},\delta})\right)^d \times Q_\delta(\Omega_{t_{n+1},\delta})$, with $\mathbf{u}_\delta^0 = \mathbf{u}_{0,\delta}$ in $\Omega_{t_0,\delta}$, such that*

$$
\begin{aligned}
\rho\frac{\beta_{-1}}{\Delta t}\left(\mathbf{u}_\delta^{n+1},\mathbf{v}\right)_{\Omega_{t_{n+1},\delta}} + c\left(\mathbf{u}_\delta^{n+1},\mathbf{v};\mathbf{u}_\delta^* - \mathbf{w}_\delta^{n+1}\right)_{\Omega_{t_{n+1},\delta}} + & \\
s\left(\mathbf{u}_\delta^{n+1},\mathbf{v};\mathbf{u}_\delta^*\right)_{\Omega_{t_{n+1},\delta}} + a\left(\mathbf{u}_\delta^{n+1},\mathbf{v}\right)_{\Omega_{t_{n+1},\delta}} + b\left(\mathbf{v},p_\delta^{n+1}\right)_{\Omega_{t_{n+1},\delta}} &= \left(\tilde{\mathbf{f}}_\delta^{n+1},\mathbf{v}\right)_{\Omega_{t_{n+1},\delta}}, \quad \forall \mathbf{v} \in \mathbf{V}_\delta(\Omega_{t_{n+1},\delta}) \\
b\left(\mathbf{u}_\delta^{n+1},q\right)_{\Omega_{t_{n+1},\delta}} &= 0, \quad \forall q \in Q_\delta(\Omega_{t_{n+1},\delta})
\end{aligned}
\tag{20}
$$

where

$$
\tilde{\mathbf{f}}_\delta^{n+1} = \mathbf{f}^{n+1} + \rho\sum_{j=0}^{q-1}\frac{\beta_j}{\Delta t}\mathbf{u}_\delta^{n-j}
$$

Notice that the functions $\mathbf{u}_\delta^{n-j}$ should be defined in $\Omega_{t_{n-j},\delta}$, which might not coincide with the integration domain $\Omega_{t_{n+1},\delta}$. However, these quantities can be ported from their domain of definition to the current one by applying ALE maps. More precisely, if we denote by $\mathbf{u}_\delta^{n-j,*}$ the approximation of $\mathbf{u}(t_{n-j})$ defined in $\Omega_{t_{n-j},\delta}$, then

$$
\mathbf{u}_\delta^{n-j} = \mathbf{u}_\delta^{n-j,*} \circ \mathcal{A}_{t_{n+1},\delta} \circ \mathcal{A}_{t_{n-j},\delta}^{-1}.
$$

Similar considerations are valid every time a quantity defined in a domain of the type $\Omega_{t_k,\delta}$ needs to be ported to the current computational domain $\Omega_{t_{n+1},\delta}$.

In equation (20), there are two quantities that we have not yet defined, or at least said how to calculate: $\mathbf{u}_\delta^*$ and $\mathbf{w}_\delta^{n+1}$. Regarding the former, this is a linearization of the convective term of the Navier-Stokes equations. We define $\mathbf{u}_\delta^*$ as (see Quarteroni, Sacco and Saleri [38])

$$
\mathbf{u}_\delta^* = \begin{cases}
\mathbf{u}_\delta^n, & q=1 \\
2\mathbf{u}_\delta^n - \mathbf{u}_\delta^{n-1}, & q=2 \\
3\mathbf{u}_\delta^n - 3\mathbf{u}_\delta^{n-1} + \mathbf{u}_\delta^{n-2}, & q=3 \\
4\mathbf{u}_\delta^n - 6\mathbf{u}_\delta^{n-1} + 4\mathbf{u}_\delta^{n-2} - \mathbf{u}_\delta^{n-3}, & q=4.
\end{cases}
\tag{21}
$$

Regarding $\mathbf{w}_\delta^{n+1}$, the discrete time derivative of the discrete ALE map, we also adopt the BDFq schemes to approximate it. For instance, for $q=2$, we have

$$
\mathbf{w}_\delta^{n+1} = \frac{1}{\Delta t}\left(\frac{3}{2}\mathcal{A}_{t_{n+1},\delta} - 2\mathcal{A}_{t_n,\delta} + \frac{1}{2}\mathcal{A}_{t_{n-1},\delta}\right) \circ \mathcal{A}_{t_{n+1},\delta}^{-1}.
\tag{22}
$$

Numerical schemes of the type (20) have been analyzed in literature in the context of a linear advection diffusion problem. It has been shown in Nobile [30] that when applying the Backward Euler time integration method (equivalent to our method with $q=1$) to the advection diffusion problem in the non-conservative form, the scheme is only conditionally stable. The stability condition (derived in [30]) is

$$
\Delta t < \left(\|\mathrm{div}(\mathbf{w}_\delta^n)\|_{L^\infty(\Omega_{t_n,\delta})} + \sup_{t\in(t_n,t_{n+1})}\left\|J_{\mathcal{A}_{t_n,t_{n+1}}}\mathrm{div}(\mathbf{w}_\delta)\right\|_{L^\infty(\Omega_{t,\delta})}\right)^{-1}
\tag{23}
$$

for all $n = 1,\ldots,N_T$. We remark that only geometrical quantities are involved in (23). If the mesh velocity is calculated such that it is divergence free, then the scheme is unconditionally stable. This is a sufficient condition to satisfy the Geometric Conservation Law (GCL), see remark 3.8.

Also in Nobile [30], for the case $q = 2$, again in the context of a linear advection diffusion equation, it is shown that the method is conditionally stable and the time step restriction depends only on geometrical quantities, just like (23).

Remark 3.8 (Geometric Conservation Law)**.** *We say that an equation/numerical scheme satisfies the* Geometric Conservation Law *(GCL) if it is able to reproduce a constant solution (in the absence of source terms and proper boundary conditions).*

Let us suppose that $\mathbf{u}_\delta^i \equiv \tilde{\mathbf{u}}$ and $p_\delta^i \equiv 0$ are constant, for all $i = 0, \ldots, n$. Notice that if a constant velocity is solution of the Navier-Stokes system, then the pressure is zero all over the domain, in the presence of homogeneous Neumann boundary conditions.

Then, from the system of equations (20), in order that $(\tilde{\mathbf{u}}, 0)$ be a solution of (20), we need that

$$\int_{\Omega_{t_{n+1}}} \frac{\beta_{-1}}{\Delta t} \mathbf{u}_\delta^{n+1} \cdot \mathbf{v} \, dx = \int_{\Omega_{t_{n+1}}} \sum_{j=0}^{q-1} \frac{\beta_j}{\Delta t} \mathbf{u}_\delta^{n-j} \cdot \mathbf{v} \, dx, \ \forall \mathbf{v} \in \mathbf{V}_\delta(\Omega_{t_{n+1},\delta})$$

which is true if

$$\beta_{-1} = \sum_{j=0}^{q-1} \beta_j. \tag{24}$$

The previous condition is necessary and sufficient if $\tilde{\mathbf{u}} \neq \mathbf{0}$. On the other hand, equality (24) is a consequence of the consistency of the BDFq schemes. Therefore, our formulation of the Navier-Stokes equations in the ALE frame satisfies the GCL, for all BDFq schemes considered.

3.5 Fully discrete system

Let us consider basis functions for the spaces $\mathbf{V}_\delta(\Omega_{t_{n+1},\delta})$ and $Q_\delta(\Omega_{t_{n+1},\delta})$, say

$$\mathbf{V}_\delta(\Omega_{t_{n+1},\delta}) = \operatorname{span}\{\boldsymbol{\phi}_i\}_{i=1}^{N_u}, \quad Q_\delta(\Omega_{t_{n+1},\delta}) = \operatorname{span}\{\psi_i\}_{i=1}^{N_p}.$$

In practice, the construction of these spaces is done by considering their reference counterparts, $\mathbf{V}_\delta(\Omega_{t_0,\delta})$ and $Q_\delta(\Omega_{t_0,\delta})$, and applying the ALE map to the reference triangulation, $\mathcal{T}_{t_0,\delta}$.

We introduce the following matrices and vectors (we omit the superscript $n+1$ to indicate the dependence of the basis functions and the matrices on n to simplify the notation):

$$\begin{array}{rcll}
G_\delta(i,j) & = & -b\,(\boldsymbol{\phi}_i, \psi_j)_{\Omega_{t_{n+1},\delta}}, & 1 \leqslant i \leqslant N_u, 1 \leqslant j \leqslant N_p \\
D_\delta(i,j) & = & b\,(\boldsymbol{\phi}_j, \psi_i)_{\Omega_{t_{n+1},\delta}}, & 1 \leqslant j \leqslant N_u, 1 \leqslant i \leqslant N_p \\
H_\delta(i,j) & = & a\,(\boldsymbol{\phi}_i, \boldsymbol{\phi}_j)_{\Omega_{t_{n+1},\delta}}, & 1 \leqslant i,j \leqslant N_u \\
C_\delta(i,j) & = & c\left(\boldsymbol{\phi}_i, \boldsymbol{\phi}_j; \mathbf{u}_\delta^* - \mathbf{w}_\delta^{n+1}\right)_{\Omega_{t_{n+1},\delta}} + s\,(\boldsymbol{\phi}_i, \boldsymbol{\phi}_j; \mathbf{u}_\delta^*)_{\Omega_{t_{n+1},\delta}}, & 1 \leqslant i,j \leqslant N_u \\
M_\delta(i,j) & = & (\boldsymbol{\phi}_i, \boldsymbol{\phi}_j)_{\Omega_{t_{n+1},\delta}} & 1 \leqslant i,j \leqslant N_u \\
\mathbf{F}_\delta(j) & = & \left(\tilde{\mathbf{f}}_\delta^{n+1}, \boldsymbol{\phi}_j\right)_{\Omega_{t_{n+1},\delta}}, & 1 \leqslant j \leqslant N_u
\end{array}$$

and

$$F_\delta = \rho \frac{\beta_{-1}}{\Delta t} M_\delta + \nu H_\delta + C_\delta.$$

Then Problem 3.3 is equivalent to solve, for each $n \geqslant 1$ a system of the form

$$\underbrace{\begin{bmatrix} F_\delta & G_\delta \\ D_\delta & 0 \end{bmatrix}}_{A_N} \begin{bmatrix} \mathbf{U}_\delta^{n+1} \\ \mathbf{P}_\delta^{n+1} \end{bmatrix} = \begin{bmatrix} \mathbf{F}_\delta \\ 0 \end{bmatrix} \tag{25}$$

where $\mathbf{U}_\delta^{n+1}$ and $\mathbf{P}_\delta^{n+1}$ denote the vector representations of $\mathbf{u}_\delta^{n+1}$ and p_δ^{n+1} in the bases of $\mathbf{V}_\delta(\Omega_{t_{n+1},\delta})$ and $Q_\delta(\Omega_{t_{n+1},\delta})$, respectively and $N = N_u + N_p$.

Remark 3.9. *When the IP stabilization term is considered in the variational formulation, its contribution is added to matrix C_δ. In this case, the components of C_δ are defined as*

$$C_\delta(i,j) = c\left(\boldsymbol{\phi}_i, \boldsymbol{\phi}_j; \mathbf{u}_\delta^* - \mathbf{w}_\delta^{n+1}\right)_{\Omega_{t_{n+1},\delta}} + s\left(\boldsymbol{\phi}_i, \boldsymbol{\phi}_j; \mathbf{u}_\delta^*\right)_{\Omega_{t_{n+1},\delta}} + j\left(\boldsymbol{\phi}_i, \boldsymbol{\phi}_j; \mathbf{u}_\delta^* - \mathbf{w}_\delta^{n+1}\right)_{\Omega_{t_{n+1},\delta}},$$

for $1 \leqslant i,j \leqslant N_u$.

3.6 Linear algebra solution strategy

In the following, we analyse a family of block preconditioners for A_N and compare it to two other strategies: a direct solver using a LU factorization (see section 4.3.1) and a preconditioner performing an incomplete LU factorization. All three strategies are used in combination with the GMRES iterative method. Numerical results with the comparison of the three preconditioning strategies is presented in section 4.2.

We also consider as a solver for system (25), the Yosida-q schemes proposed in [39, 44, 13, 12]. Numerical results using these schemes are presented in section 4.3.2.

From now on, we subscript the matricies by N and not by δ.

3.6.1 LU and ILU preconditioner

The LU and ILU factorizations of the matrix A_N are calculated with the help of the Ifpack library provided by Trilinos, see [43]. In particular, the LU factorization is calculated using the KLU algorithm, see [7, 48]. When solving similar systems, as is the case in section 4.3, we reuse the LU factorization as preconditioner until the number of iterations needed to solve the linear system is equal to 10. Once this value is attained, the LU factorization is recalculated. A better strategy to determine when to recalculate the preconditioner is described in [52].

3.6.2 A block type preconditioner

We start by noticing that A_N can be factorized as follows

$$A_N = \underbrace{\begin{bmatrix} I_N & 0 \\ D_N F_N^{-1} & I_N \end{bmatrix}}_{L} \underbrace{\begin{bmatrix} F_N & 0 \\ 0 & S_N \end{bmatrix}}_{D} \underbrace{\begin{bmatrix} I_N & F_N^{-1} G_N \\ 0 & I_N \end{bmatrix}}_{U}. \tag{26}$$

where we denote by $S_N = -D_N F_N^{-1} G_N$ the *pressure Schur complement.*

If we use the matrix $P_L = LD$ as preconditioner for A_N and a Krylov subspace method, then the matrix $P_L^{-1} A_N$ has two distinct eigenvalues, thus convergence is achieved in at most two iterations, see Murphy, Golub and Wathen [29]. However, this preconditioner is prohibitive in practice due to the presence of the pressure Schur complement. The idea here is to build an effective preconditioner by replacing matrices F_N and S_N by cheap approximations, say $\tilde{F}_N$ and $\tilde{S}_N$. These approximate versions of the original operators should be chosen such that they constitute good preconditioners for F_N and S_N, respectively.

In the work of Elman and Sylvester [10], the matrix $P_R = DU$ was used as a right preconditioner together with the GMRES method to solve the steady Stokes and Navier-Stokes equations. This preconditioner has the property that the number of iterations stays bounded independently of the mesh size h or the polynomial degree of the approximation N. The preconditioner and these results were extended to the unsteady Navier-Stokes case for $N = 2$ in Silvester, Elman, Kay and Wathen [46].

We propose a left preconditioner, P, based on P_L and the ideas presented in Elman and Sylvester [10] and Silvester, Elman, Kay and Wathen [46]. The experiments in [46] are extended to spectral discretizations. We show in section 4.2 that we obtain the same properties. Let

$$P = \begin{bmatrix} \tilde{F}_N & 0 \\ D_N & \tilde{S}_N \end{bmatrix} \tag{27}$$

where $\tilde{F}_N$ and $\tilde{S}_N$ are suitable approximations of F_N and S_N.

The inverse of P is given by

$$P^{-1} = \begin{bmatrix} \tilde{F}_N^{-1} & 0 \\ R_N & \tilde{S}_N^{-1} \end{bmatrix} \tag{28}$$

where $R_N = -\tilde{S}_N^{-1} D_N \tilde{F}_N^{-1}$. If a Krylov subspace method is used to solve problem (25), then, at each iteration, we need to solve a system with matrix P. This means that for a given vector $(\mathbf{r}, \mathbf{s})$, we need to calculate $(\mathbf{v}, \mathbf{q})$ such that

$$\begin{bmatrix} \tilde{F}_N & 0 \\ D_N & \tilde{S}_N \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ \mathbf{q} \end{bmatrix} = \begin{bmatrix} \mathbf{r} \\ \mathbf{s} \end{bmatrix}. \tag{29}$$

In order to solve (29), we follow Algorithm 1.

Algorithm 1 Steps to solve a system with matrix $\mathcal{P}$.

given $\mathbf{r}$ and $\mathbf{s}$,
solve $\tilde{F}_N \mathbf{v} = \mathbf{r}$
solve $\tilde{S}_N \mathbf{q} = \mathbf{s} - D_N \mathbf{v}$
return $(\mathbf{v}, \mathbf{q})$

Choice for the operator $\tilde{F}_N$ In this section we will consider $\tilde{F}_N = F_N$ and solve any systems with this matrix using a LU factorization. If $\alpha = \frac{\beta_{-1}}{\Delta t}$ is "large", a cheap alternative is to take $\tilde{F}_N$ as the diagonal of αM_N. However, this choice is not considered in this work since we want to assess first the behavior of the preconditioner P by using the matrix F_N. Other more robust choices are additive Schwarz or multigrid methods. The latter were used in the works by Silvester, Elman, Kay and Wathen [46] and Kay, Loghin and Wathen [27].

Choice for the operator $\tilde{S}_N$ We follow the idea of Kay, Loghin and Wathen [27] and take as approximation of the pressure Schur complement the operator $\tilde{S}_N = A_p {F_p}^{-1} M_p$, where A_p, F_p and M_p are the discretizations of the pressure operators $-\Delta$, $\alpha I - \nu\Delta + \boldsymbol{\beta}\cdot\nabla$ and I, respectively. The quantity $\boldsymbol{\beta}$ is the velocity obtained after linearization of the non linear convective term of the momentum equation. If the velocity field is convection dominated, then the discretization of the convection-diffusion-reaction pressure operator should also be stabilized.

The preconditioner that we obtain with the choices of $\tilde{F}_N$ and $\tilde{S}_N$ is called *block triangular pressure convection diffusion* (BTPCD) preconditioner

$$\tilde{P} = \begin{bmatrix} F_N & 0 \\ D_N & A_p F_p^{-1} M_p \end{bmatrix}. \tag{30}$$

For a complete study of the properties of the preconditioner BTPCD, we refer the reader to Pena [32].

Regarding the overall computational cost of using $\tilde{S}_N$ as preconditioner, at each iteration, we have to solve a system associated with the mass matrix M_p and the discrete laplacian A_p (for which efficient solvers can be chosen, for instance, *preconditioned conjugate gradient method*) and apply operator F_p.

Inner loop solvers As mentioned before, for each iteration of a Krylov subspace method we have to solve a system like (29). This operation translates in solving three systems, with matrices $\tilde{F}_N$, M_p and A_p. We use LU factorizations to solve the three of them.

An alternative to a direct solve of the discrete pressure operators is to use the preconditioned conjugate gradient iterative method. This is a valid choice due to the fact that these matrices are symmetric and positive definite. Suitable preconditioners for this method can be obtained through incomplete Cholesky factorizations or multigrid method. We remark that the preconditioners for M_p and A_p only need to be calculated once and then reused at each iteration.

3.6.3 The Yosida-q schemes

An efficient solution technique of the Navier-Stokes equations are splitting methods, of either differential or algebraic type, see [18, 19, 17, 40, 39, 44, 5] for a few references. The former methods split the differential operators of the equations, while the latter splits the linear system like (25) using an inexact block LU factorization. In this section we present a class of algebraic factorization methods, named as *Yosida-q* schemes.

The starting point of the Yosida schemes is the factorization (26). In the first version of the Yosida scheme, introduced by Quarteroni, Saleri and Veneziani [40], the central ideas is to approximate the matrix F_N^{-1}, in the pressure Schur complement $S_N = -\frac{\Delta t}{\beta_{-1}} D_N F_N^{-1} G_N$, by a second order in time approximation

$$F_N^{-1} \approx \frac{\Delta t}{\beta_{-1}} M_N^{-1}.$$

This leads to the approximate matrix $\tilde{A}_N$ given by

$$\tilde{A}_N = \begin{bmatrix} F_N & 0 \\ D_N & -\frac{\Delta t}{\beta_{-1}} D_N M_N^{-1} G_N \end{bmatrix} \begin{bmatrix} I_N & F_N^{-1} G_N \\ 0 & I_N \end{bmatrix}.$$

It was shown by Quarteroni, Saleri and Veneziani [39] that this scheme applied to the unsteady Stokes equations, together with a BDF2 time discretization leads to second order in time convergence for the velocity, order $3/2$ for the pressure and unconditional stability.

Remark 3.10. *Replacing the pressure Schur complement by* $S_N^{app} = -\frac{\Delta t}{\beta_{-1}} D_N M_N^{-1} G_N$*, called* approximate pressure Schur complement*, allows to reduce the computational cost to solve system* (25)*, while introducing a splitting error of the same order as the time discretization used for the Navier-Stokes equations. Since* S_N^{app} *is s.p.d. we can use the preconditioned conjugate gradient method to efficiently invert it. Moreover, in the case matrix* M_N *is lumped, its inversion is very cheap.*

Later versions of this first Yosida scheme, now called *Yosida-2* scheme, have been proposed and improved the order of convergence in time for velocity and pressure. By introducing a matrix J_N in the inexact block LU factorization

$$\tilde{A}_N = \begin{bmatrix} F_N & 0 \\ D_N & -\frac{\Delta t}{\beta_{-1}} D_N M_N^{-1} G_N \end{bmatrix} \begin{bmatrix} I_N & F_N^{-1} G_N \\ 0 & J_N \end{bmatrix}$$

and choosing it carefully, one can obtain schemes of order q for the velocity and $q - 1/2$ for the pressure, named *Yosida-q*. This choice is based on the minimization of the splitting error originated from approximating A_N with $\tilde{A}_N$, see Saleri and Veneziani [44], Gervasio, Saleri and Veneziani [13] and Gervasio [12].

All three Yosida schemes differ only in the expression of matrix J_N. While for Yosida-*2* it is equal to the identity matrix, the higher order versions take more involved expressions. If we define

$$B_N = -D_N \frac{\Delta t}{\beta_{-1}} M_N^{-1} F_N \frac{\Delta t}{\beta_{-1}} M_N^{-1} G_N$$

then for Yosida-*3*

$$J_N = B_N^{-1} S_N^{app}. \tag{31}$$

The fourth order version of the Yosida schemes, Yosida-*4*, is obtained by replacing B_N in (31) by

$$\hat{B}_N = B_N (S_N^{app})^{-1} B_N + B_N + D_N \left(\frac{\Delta t}{\beta_{-1}} M_N^{-1} F_N \right)^2 \frac{\Delta t}{\beta_{-1}} M_N^{-1} G_N.$$

Though appearing complex to calculate, the three Yosida schemes can be summarized in Algorithm 2. A complete analysis on the convergence properties of all Yosida schemes, for a time-dependent Stokes problem, is provided in Gervasio [12].

Algorithm 2 A step of the Yosida algorithm.

given $\mathbf{f}$ and $\mathbf{g}$,
solve $F_N\tilde{\mathbf{u}} = \mathbf{f}$
solve $S_N^{app}\tilde{\mathbf{p}} = \mathbf{g} + D_N\tilde{\mathbf{u}}$
if $q > 2$ **then**
  solve $\mathbf{z} = B_N\tilde{\mathbf{p}}$
  solve $S_N^{app}\mathbf{p} = \mathbf{z}$
  if $q = 4$ **then**
    compute $\mathbf{p}_B = B_N\mathbf{p} + \mathbf{z} + D_N\left(\frac{\Delta t}{\beta_{-1}}M_N^{-1}F_N\right)^2 \frac{\Delta t}{\beta_{-1}}M_N^{-1}G_N\tilde{\mathbf{p}}$
    solve $S_N^{app}\mathbf{p} = \mathbf{p}_B$
  end if
else
  $\mathbf{p} = \tilde{\mathbf{p}}$
end if
$F_N(\mathbf{u} - \tilde{\mathbf{u}}) = -G_N\mathbf{p}$
return $(\mathbf{u}, \mathbf{p})$

4 Numerical experiments

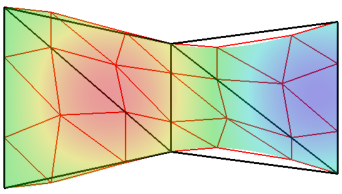
In this section we present some numerical tests to the ALE map, the preconditioning strategy and the whole solver for the Navier-Stokes equations.

4.1 ALE map approximation properties

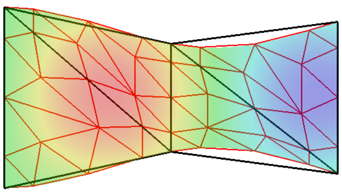
We present now some numerical results to assess the accuracy of the ALE map describing the boundary of a 2D domain. Let us consider the reference domain $\Omega_{t_0} = (0, 5) \times (-1, 1)$.

We define Ω_{t_1} as the domain we obtain by moving the upper and lower sides of the rectangle Ω_{t_0} using the following displacement functions:

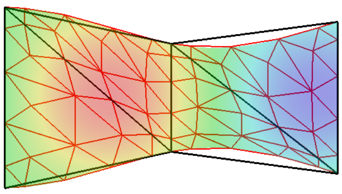
- upper boundary: $\boldsymbol{\eta}(x, 1) = [x, 1 + 0.3\cos(x)]^T$
- lower boundary: $\boldsymbol{\eta}(x, -1) = [x, -1.1 - 0.3\cos(x)]^T$



(a) $\mathcal{A}_{h,3}$ map



(b) $\mathcal{A}_{h,4}$ map



(c) $\mathcal{A}_{h,5}$ map

Figure 8: The thick lines define the $\mathbb{P}_1$ coarse mesh used in the construction of the ALE maps. Inside each of the elements of this mesh, we observe the $\mathbb{P}_1$ triangulation constructed on top of the high order nodes. In this figure, $h = 2$.

In Figures 8(a)-8(c) we show the application of the ALE maps $\mathcal{A}_{h,N_{\text{geo}}} : \Omega_{t_0} \longrightarrow \Omega_{t_1,h,N_{\text{geo}}}$ constructed using polynomials of degree two to five to a mesh of the reference domain.

We also wanted to determine the accuracy at which the ALE maps describe the boundary of the domain Ω_{t_1}. For this, we measured the error

$$\left\|(\mathcal{A}_{h,N_{\text{geo}}}(\cdot, 1) - \boldsymbol{\eta}(\cdot, 1)) \cdot \mathbf{e}_2\right\|_{L^2(0,5)}$$

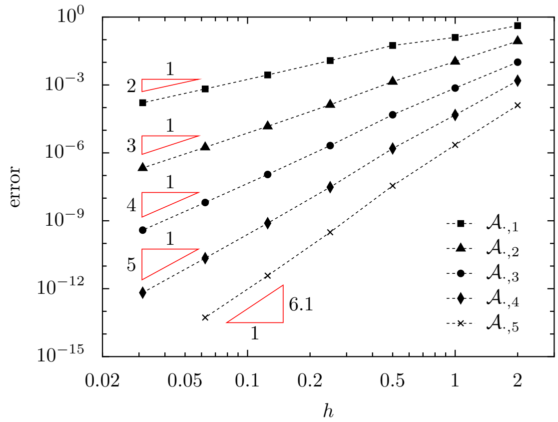


Figure 9: Convergence plot for the high order ALE maps

in the upper boundary of the reference domain and

$$\left\|(\mathcal{A}_{h,N_{\mathrm{geo}}}(\cdot,-1)-\boldsymbol{\eta}(\cdot,-1))\cdot\mathbf{e}_2\right\|_{L^2(0,5)}$$

in the lower part of Ω_{t_0}. We plot the sum of both quantities in Figure 9. The error decreases with the expected rates, ie, $\mathcal{O}(h^{N_{\mathrm{geo}}+1})$.

4.2 Comparison of preconditioners for linear algebra strategy

In this section, we compare the preconditioner (27) with two others strategies: a LU factorization (which translates in practice in solving the system (25) with this type of factorization) and an incomplete LU factorization, with fill-in 3 (denoted, from now on, as ILU(3)). We compare these three solution strategies in terms of the time to calculate the preconditioner and the time to solve the linear system, both regarding the mesh size h and the polynomial degree N. We highlight that our results are obtained using only one processor. The use of more processors and parallel implementations of the LU/ILU(3) factorization are not discussed in this work.

In the case of the LU and ILU(3), the preconditioner is calculated directly from the matrix of system (25). However, for the BTPCD preconditioner, at each iteration of the fixed point method, the cost of constructing this preconditioner is dependent on the calculation of the LU factorization of the F_N block and the assembly of the pressure convective term plus the construction of matrix F_p.

Let us consider the following problem

$$\begin{aligned} -\nu\boldsymbol{\Delta}\mathbf{u}+(\mathbf{u}\cdot\boldsymbol{\nabla})\mathbf{u}+\nabla p &= \mathbf{0}, \quad \text{in } \Omega \\ \operatorname{div}(\mathbf{u}) &= 0, \quad \text{in } \Omega \end{aligned} \tag{32}$$

with $\nu = 0.1$, applied to the backward facing step problem, where Ω is depicted in Figure 10. Regarding boundary conditions, we impose homogeneous Dirichlet conditions for the velocity everywhere, except in the inflow and outflow boundaries. At the inflow we impose a parabolic profile

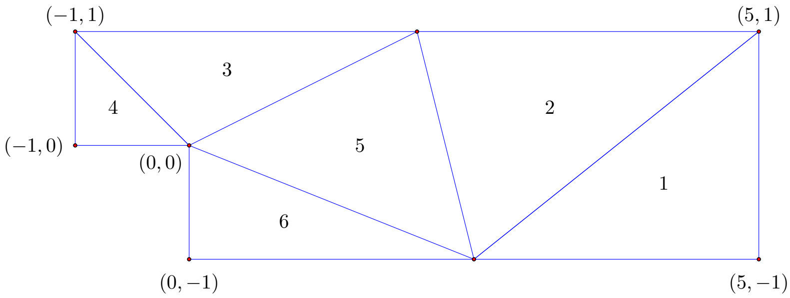
$$\mathbf{u} = [y(1-y), 0]^T$$

and at the outflow, homogeneous Neumann boundary conditions.

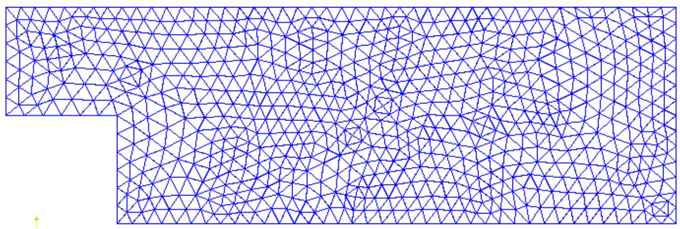
In order to solve the convection non linearity after space discretization, we will use fixed point (Picard) iterations, meaning, for $k > 0$, we solve the system

$$-\nu\boldsymbol{\Delta}\mathbf{u}^k+(\mathbf{u}^{k-1}\cdot\boldsymbol{\nabla})\mathbf{u}^k+\nabla p^k = \mathbf{0}, \quad \text{in } \Omega \tag{33}$$

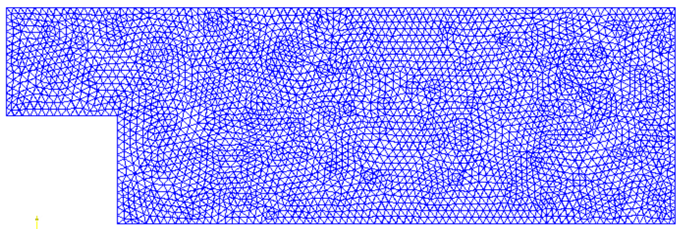
$$\operatorname{div}(\mathbf{u}^k) = 0, \quad \text{in } \Omega \tag{34}$$



(a) $h = 2$



(b) $h = 0.125$



(c) $h = 0.0625$

Figure 10: Three spectral element triangulations of the computational domain used for problem (32).

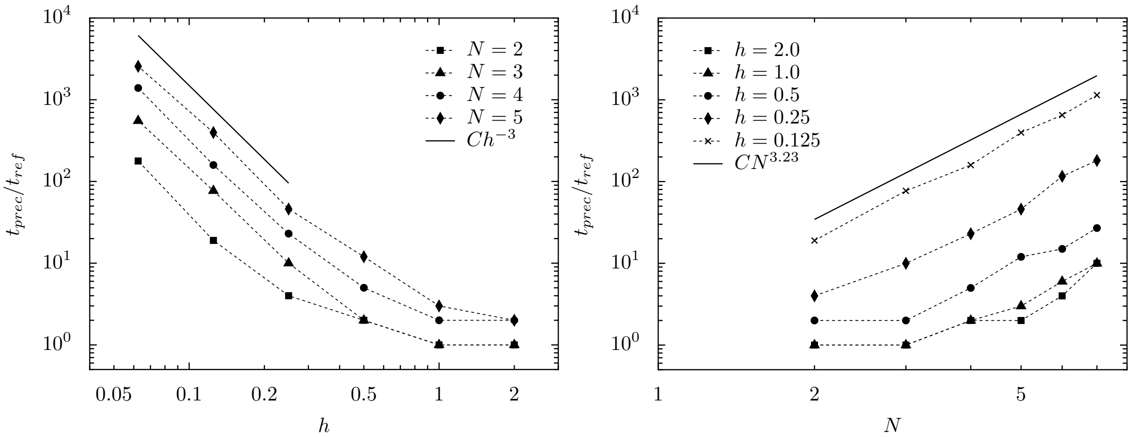
for $\mathbf{u}^k$ and p^k. At the space discretization level, we consider the $\mathbb{P}_N - \mathbb{P}_{N-1}$ method up to degree $N = 7$. The bases of the spaces for the discrete velocity and pressure are built with standard Lagrange polynomials associated with Fekete points.

The stopping criteria for the fixed points scheme is

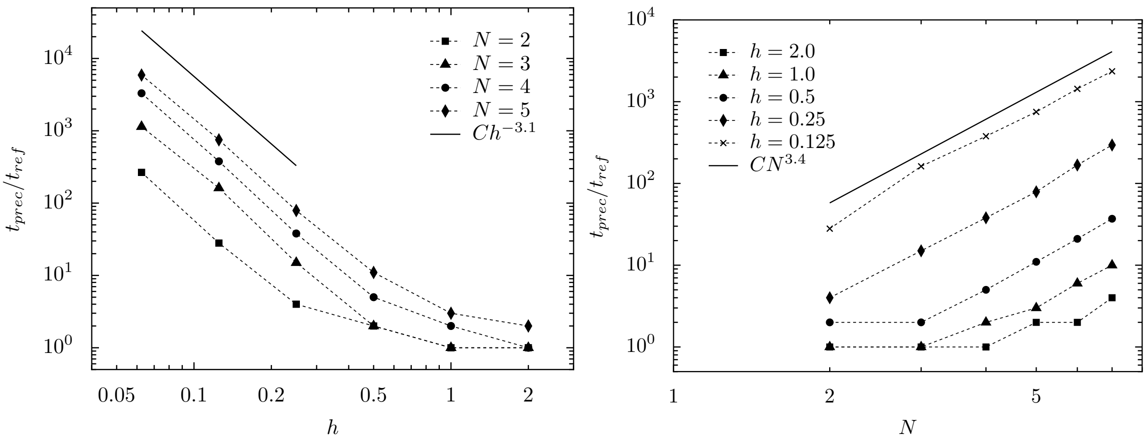
$$\left(\left\|\mathbf{u}_N^{k-1} - \mathbf{u}_N^k\right\|^2_{L^2(\Omega)} + \left\|p_N^{k-1} - p_N^k\right\|^2_{L^2(\Omega)}\right)^{1/2} < 10^{-6} \tag{35}$$

where $\mathbf{u}_N^k$ and p_N^k denote the spectral element approximations of $\mathbf{u}^k$ and p^k.

Tables 1 show the maximum number of iterations used by the GMRES method, N_{it}, to solve the steady Navier-Stokes problem (33)-(34), as well as the time to calculate the preconditioner, t_{prec}, and the maximum time to solve the linear system, t_{solve}. Different polynomial degree and different spectral element gridsize are used. These results were obtained using a Dual Core AMD Opteron(tm) Processor 270, 2GHz cpu and 3Gb of RAM memory.



(a) BTPCD preconditioner.



(b) LU factorization.

Figure 11: Plot of the relative time to build the preconditioner, varying the mesh size (left) and the polynomial degree (right).

We observe from Tables 1 that for the size of problems we tested ($\lesssim$ 100000 degrees of freedom), the LU factorization applied to the linear system proves to be the fastest solution strategy. However, for bigger problems, this factorization takes too much memory and time to calculate and

	N																	
	2			**3**			**4**			**5**			**6**			**7**		
h	N_{it}	t_{prec}	t_{solve}	N_{it}	t_{prec}	t_{solve}	N_{it}	t_{prec}	t_{solve}	N_{it}	t_{prec}	t_{solve}	N_{it}	t_{prec}	t_{solve}	N_{it}	t_{prec}	t_{solve}
2.0	16	0.01	0.04	42	0.01	0.12	54	0.02	0.13	67	0.02	0.23	62	0.04	0.3	61	0.1	0.5
1.0	26	0.01	0.05	55	0.01	0.17	58	0.02	0.27	56	0.03	0.43	57	0.06	0.69	59	0.1	1.07
0.5	45	0.02	0.15	58	0.02	0.47	57	0.05	0.9	57	0.12	1.52	60	0.15	2.53	58	0.27	3.66
0.25	45	0.04	0.76	41	0.1	1.72	35	0.23	3.01	35	0.46	5.37	35	1.16	9.08	34	1.81	12.39
0.125	35	0.19	2.56	33	0.77	6.54	33	1.59	12.83	32	3.97	23.41	31	6.5	33.75	30	11.41	49.24
0.0625	30	1.78	8.95	30	5.53	24.58	30	13.93	47.8	30	25.64	84.45						

	N																	
	2			**3**			**4**			**5**			**6**			**7**		
h	N_{it}	t_{prec}	t_{solve}	N_{it}	t_{prec}	t_{solve}	N_{it}	t_{prec}	t_{solve}	N_{it}	t_{prec}	t_{solve}	N_{it}	t_{prec}	t_{solve}	N_{it}	t_{prec}	t_{solve}
2.0	1	0.01	0.01	1	0.02	0.01	1	0.05	0.01	1	0.14	0.01	1	0.36	0.01	1	0.88	0.02
1.0	2	0.02	0.01	3	0.09	0.01	2	0.42	0.02	3	1.59	0.02	3	4.73	0.03	2	11.82	0.04
0.5	5	0.1	0.01	5	1.09	0.02	5	6.43	0.05	5	24.79	0.12	5	75.07	0.24	5	189.6	0.43
0.25	9	1.39	0.05	10	15.85	0.22	10	90.64	0.68	10	353.99	1.68	10	1033.4	2.72	10	¿ 10^5	5.13
0.125	18	7.97	0.51	21	92.32	2.18	21	519.48	6.78	22	1887.45	13.42						
0.0625	51	30.28	4.42	62	398.59	27.56	62	2144.56	81.86									

	N											
	2		**3**		**4**		**5**		**6**		**7**	
h	t_{prec}	t_{solve}	t_{prec}	t_{solve}	t_{prec}	t_{solve}	t_{prec}	t_{solve}	t_{prec}	t_{solve}	t_{prec}	t_{solve}
2.0	0.01	0.01	0.01	0.01	0.01	0.01	0.02	0.01	0.02	0.01	0.04	0.01
1.0	0.01	0.01	0.01	0.01	0.02	0.01	0.03	0.01	0.06	0.02	0.1	0.02
0.5	0.02	0.01	0.02	0.01	0.05	0.01	0.11	0.02	0.21	0.03	0.37	0.04
0.25	0.04	0.01	0.15	0.02	0.38	0.04	0.79	0.08	1.68	0.13	2.95	0.21
0.125	0.28	0.03	1.62	0.08	3.78	0.26	7.5	0.44	14.35	0.65	23.52	0.92
0.0625	2.66	0.13	11.4	0.37	33.03	0.98	58.98	1.48				

Table 1: Timings (in seconds) to calculate preconditioners, solve the linear system and the number of GMRES iterations. BTPCD preconditioner (top), ILU(3) factorization (middle) and LU factorization (bottom).

stops being an acceptable option. The same conclusion can be taken from the results regarding the ILU(3) factorization as preconditioner. In this case, the cost of the calculation of the preconditioner is where the most amount of time is spent. We observe however that if we consider $t_{ref} = 0.01$ as a reference time scale, the time spent on building the preconditioner and solving the linear system is similar using the LU factorization and the BTCPD preconditioner (combined with the iterative GMRES method). In fact, see Figure 11, to build the block type preconditioner it takes $\mathcal{O}(h^{-3})$ or $\mathcal{O}(N^{3.23})$ in relative time, as for the LU factorization these values are slightly different: $\mathcal{O}(h^{-3.1})$ or $\mathcal{O}(N^{3.4})$, respectively. The amount of time to solve the linear system is proportional to $\mathcal{O}(h^{-2})$ or $\mathcal{O}(N^{2.37})$ in the BTPCD case and $\mathcal{O}(h^{-2.1})$ or $\mathcal{O}(N^{2.2})$ in the LU factorization case. We remark, although we do not show the graphics, that the ILU factorization takes only $\mathcal{O}(h^{-2.3})$ of relative time to calculate the preconditioner, but needs $\mathcal{O}(h^{3.45})$ to solve the linear problem. Regarding the variation in terms of polynomial order, it is $\mathcal{O}(N^{6})$ and $\mathcal{O}(N^{3.79})$ to calculate the preconditioner and solve the linear system, respectively.

Remark 4.1. *We stress that although the growth rates to build or solve the linear system using the BTPCD preconditioner or the LU factorization are similar, the magnitude of the time scale is quite different. This means the constants present in the growth rates associated with the LU factorization are much smaller than the ones for the BTPCD preconditioner. Moreover, the difference in magnitude can be seen in Table 1.*

Regarding the number of iterations used by each algorithm, the ILU(3) preconditioner together with GMRES, uses a number of iterations that stays bounded when we increase the polynomial degree. The same does not happen when the mesh size is decreased. In this case, the number of iterations increases.

4.3 Incompressible Navier-Stokes equations in a moving domain

Consider $\Omega_{t_0} = (0,5) \times (-1,1)$ and Ω_t obtained from the reference domain by applying the following displacement law

$$\mathbf{d}(\mathbf{x},t) = 0.02\left((\mathbf{x}-2.5)^2+5\right)\mathbf{x}(5-\mathbf{x})\left(f(t)\chi(t\in[1,3])+\chi(t>3)\right), \tag{36}$$

to the lower edge of the rectangle, with $t \in I = [0,5]$, $\chi(\cdot)$ the characteristic function and

$$\begin{aligned} f(t) \quad = \quad & -0.15625t^7 + 2.1875t^6 - 12.46875t^5 + 37.1875t^4 - 62.34375t^3 \\ & +59.0625t^2 - 29.53125t + 6.0625 \end{aligned}$$

Remark 4.2. *The function f satisfies $f(1) = 0$, $f(3) = 1$ and $f^{(k)}(1) = f^{(k)}(3) = 0, k = 1,2,3$. It is the only polynomial of degree 7 that satisfies these conditions. It was constructed such that the variation of the mesh velocity in time is smooth enough.*

Let $\hat{\mathbf{u}} = \left(1-y^2,0\right)^T$ and $\hat{p} = -2\nu(x-5)$ be the solution of the steady Navier-Stokes equations in the reference domain Ω_{t_0}.

We consider equations (4)-(5) defined in Ω_t with $\mathbf{f} \equiv \mathbf{0}$. Regarding boundary conditions, we define

$$\Gamma_t^N = \{5\} \times (-1,1) \qquad \text{and} \qquad \Gamma_t^D = \partial\Omega_t \setminus \Gamma_t^N.$$

We set as boundary conditions

$$\mathbf{u} = \hat{\mathbf{u}}, \quad \text{on } \Gamma_t^D \tag{37}$$

and

$$\left(-p\mathbf{I} + \mathbf{D}_{\mathbf{x}}(\mathbf{u})\right)\mathbf{n} = \left(-\hat{p}\mathbf{I} + \mathbf{D}_{\mathbf{x}}(\hat{\mathbf{u}})\right)\mathbf{n}, \quad \text{on } \Gamma_t^N$$

where $\mathbf{I}$ is the $d \times d$ identity matrix.

We remark that since the boundary of Ω_t deforms inside Ω_{t_0}, equation (37) makes sense in the part of the domain that changes in time. Also, the pair $(\hat{\mathbf{u}}, \hat{p})$ is the solution of (4)-(5) with the boundary conditions that we presented.

This benchmark test allows us to test the Navier-Stokes ALE framework with spectral elements in space, high order time integration, high order geometrical elements and also the IP stabilization.

The discretization of this problem is done with the scheme proposed in Problem 3.3. We try two strategies to solve the linear system (25): the first, using a direct method and the second, using algebraic splitting, more precisely, the Yosida-q schemes.

Let us first define the following error quantities that we are going to measure in order to assess the accuracy of the solver. We denote E_u the error in the veolocity and E_p the error in the pressure and we dine them as

$$E_u = \left(\Delta t \sum_{n=0}^{N_T} \|\mathbf{u}(t_n) - \mathbf{u}_\delta^n\|^2_{\mathbf{H}^1(\Omega_{t_n,\delta})} \right)^{1/2}$$

and

$$E_p = \left(\Delta t \sum_{n=0}^{N_T} \|p(t_n) - p_\delta^n\|^2_{L^2(\Omega_{t_n,\delta})} \right)^{1/2}.$$

4.3.1 Using a direct method

In Figure 12 we plot the error quantities E_u and E_p for two choices of approximation spaces for velocity and pressure and N_{geo}, $N = 2$, $M = 1$ and $N_{\text{geo}} = 1$, see Figure 12(a), and $N = 4$, $M = 2$ and $N_{\text{geo}} = 2$, see Figure 12(b). We also consider different integration time strategies. We considered for this test $h = 0.5$, $\nu = 10^{-3}$, $\rho = 1$. We highlight that the flow is convection dominated (without the stabilization term, the method would not converge) and we have stabilized the equations by the interior penalty term. We took $\gamma = 0.1$ in (19). These results were obtained by solving directly the linear system (25) with a LU factorization. We highlight that the preconditioner proposed in the previous chapter could have been used, but the main goal here was to study the numerical properties of the methods in terms of accuracy. Moreover, the size of the problems solved in these tests falls in the range where the LU factorization performs better than the block preconditioner proposed in section 3.6.2.

From Figure 12 we confirm the expected convergence order for the proposed methods in time. Using a BDFq time integrator, a linear extrapolation of the convective term of the same order and an approximation of the mesh velocity also with a BDFq formula, the error, in Δt is of the order of Δt^q, $q = 1, 2, 3, 4$. The convergence order of each scheme is seen in Figure 12 through the slope of each curve.

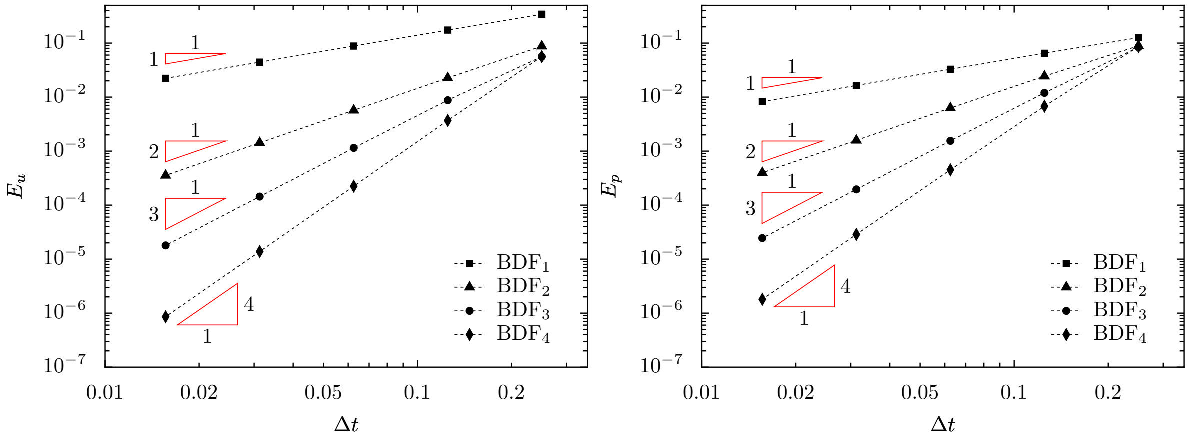
Due to stability constraints of the time integration technique used, when we increase the polynomial degree, the stability regions of the *BDF4* and *BDF3* start not to be big enough to handle the stiffness of the problem and we only get the schemes to give acceptable results when we decrease Δt. In Figure 12(b), we do not even plot the results for BDF*4* because the method was not stable for the range of Δt we considered. On the other hand, the BDF*1* and BDF*2* schemes remain stable.

We remark that in Figures 12(a) and 12(b), the numerical schemes used describe the solution of the problem exactly in space, though not the geometry. We also tested the above numerical schemes using a fourth order geometry. In this case, the orders of convergence in Δt are the same as the ones reported for the cases in Figures 12(a) and 12(b), although the stability limitations on Δt are more severe for BDF*3* and BDF*4*. Again, in this case, the BDF*1* and BDF*2* schemes remain stable.

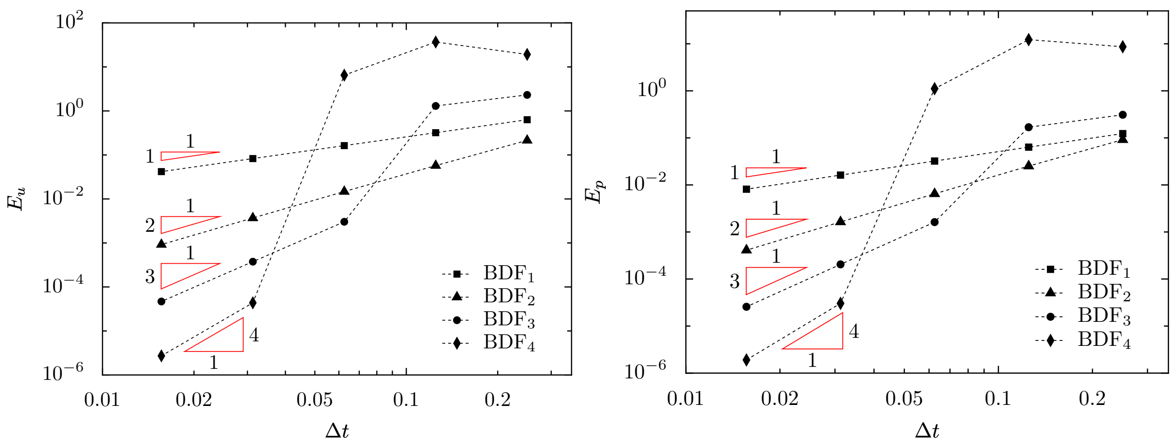
4.3.2 Using the Yosida-q schemes

In the benchmark using Yosida-q schemes, we considered $h = 0.5$, $\nu = 0.05$, $\rho = 1$ and the stabilization parameter $\gamma = 0$. The error quantities E_u and E_p are plotted in Figure 13. We notice that the convergence orders for the BDFq-Yosida-q agree with the rates predicted for the time-dependent Stokes equations in Gervasio [12].

Similar tests were conducted using $\nu = 10^{-3}$. The convergence orders for the pressure were slightly better in this case, for the range of Δt considered. This was due to the fact that the



(a) $N = 2, M = 1, N_{\text{geo}} = 1$.



(b) $N = 4, M = 2, N_{\text{geo}} = 2$.

Figure 12: Plot of the errors E_u and E_p for different choices of velocity-pressure spaces, geometrical elements and BDFq schemes.

splitting error introduced by the Yosida-q schemes was much smaller than the error introduced by the corresponding BDF method. Regarding the stability of the methods, we did not observe considerable differences between using BDFq and Yosida-q schemes, for $q = 2, 3, 4$.

5 Conclusions

In this paper, we propose a numerical strategy to solve the unsteady incompressible Navier-Stokes equations defined in a domain that changes in time.

A full discretization scheme is presented, using the triangular spectral element method combined with Lagrange basis functions constructed on Fekete points and BDFq schemes to discretize the time derivative and the ALE mesh velocity. The non-linear convective term of the Navier-Stokes equations is linearized with a formula of the same order as the BDFq scheme.

We propose a discrete ALE map that is able to describe curved boundaries, as long as the domain deformation is small. Its approximation properties in the moving boundary of the domain are of order $\mathcal{O}(h^{N_{\text{geo}}+1})$ in the $L^2(\Omega)$-norm. With respect to the linear algebra part of the solver, we presented a comparison between a block type preconditioner and two other solution strategies for the linear system. The LU factorization, a direct method, was the one that took less time to solve the system.

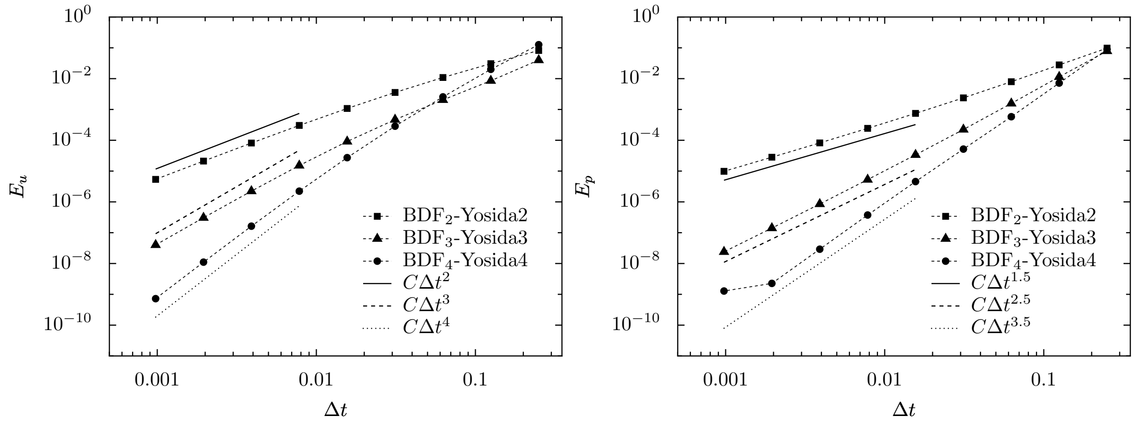


Figure 13: Plot of the errors E_u and E_p for $N = 2, M = 1, N_{\text{geo}} = 1$ and different BDFq and Yosida-q schemes.

Regarding the full method, we concluded that if the mesh velocity is approximated with the same BDF scheme as the velocity and the extrapolation formula, we showed that the method converges with order $\mathcal{O}(\Delta t^q)$ when using a direct solver for the linear system (that appears at each time step) and it converges with order $\mathcal{O}(\Delta t^q)$ and $\mathcal{O}(\Delta t^{q-\frac{1}{2}})$ in the velocity and pressure, respectively, when using the Yosida-q schemes.

Acknowledgements

The first author was partially supported by Fundação para a Ciência e Tecnologia through grant SFRH/BD/22243/2005 of POCI2010/FEDER. The second author was supported by the ISLE/CHPID grant from the region Rhônes-Alpes and the PEPS/META-LIFE grant from the CNRS. We acknowledge the support provided by the European Research Council Advanced Grant "Mathcard, Mathematical Modelling and Simulation of the Cardiovascular System" Project ERC-2008-AdG 227058.

References

[1] M. Ainsworth and P. Coggins. A uniformly stable family of mixed hp-finite elements with continuous pressures for incompressible flow. *IMA Journal of Numerical Analysis*, 22(2):307–327, April 2002.

[2] C. Bernardi and Y. Maday. Uniform inf-sup conditions for the spectral discretization of the Stokes problem. *Math. Models Meth. Appl. Sci.*, 9(3):395–414, 1999.

[3] R. Bouffanais. *Simulation of shear-driven flows: transition with a free surface and confined turbulence.* PhD thesis, n°3837, EPF Lausanne, 2007.

[4] F. Brezzi and M. Fortin. *Mixed and hybrid finite element methods.* Springer-Verlag New York, Inc., New York, NY, USA, 1991.

[5] C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang. *Spectral Methods: Fundamentals in Single Domains.* Springer-Verlag, New York and Berlin, 2006.

[6] C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang. *Spectral Methods: Evolution to Complex Geometries and Applications to Fluid Dynamics.* Springer-Verlag, New York and Berlin, 2007.

[7] T. A. Davis. *Direct Methods for Sparse Linear Systems (Fundamentals of Algorithms 2)*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2006.

[8] S. Deparis, M. Fernández, and L. Formaggia. Acceleration of a fixed point algorithm for fluid-structure interaction using transpiration conditions. *Mathematical Modelling and Numerical Analysis*, 37(4):601–16, 2003.

[9] J. Donea, S. Giuliani, and J. P. Halleux. An arbitrary Lagrangian-Eulerian finite element method for transient dynamic fluid-structure interactions. *Computer Methods in Applied Mechanics and Engineering*, 33(1-3):689–723, 1982.

[10] H. Elman and D. Silvester. Fast Nonsymmetric Iterations and Preconditioning for Navier–Stokes Equations. *SIAM J. Sci. Comput.*, 17(1):33–46, 1996.

[11] L. Formaggia, A. Quarteroni, and A. Veneziani, editors. *Cardiovascular Mathematics: Modeling and simulation of the circulatory system*, volume 1 of *MS & A*. Springer, 2009.

[12] P. Gervasio. Convergence Analysis of High Order Algebraic Fractional Step Schemes for Time-Dependent Stokes Equations. *SIAM J. Numer. Anal.*, 46(4):1682–1703, 2008.

[13] P. Gervasio, F. Saleri, and A. Veneziani. Algebraic fractional-step schemes with spectral methods for the incompressible Navier–Stokes equations. *J. Comput. Phys.*, 214(1):347–365, 2006.

[14] C. Geuzaine and J.-F. Remacle. Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79(11):1309–1331, 2009.

[15] W. J. Gordon and C. A. Hall. Construction of curvilinear coordinate systems and their applications to mesh generation. *Int. J. Numer. Meth. Eng.*, 7:461–477, 1973.

[16] W. J. Gordon and C. A. Hall. Transfinite element methods: blending-function interpolation over arbitrary curved element domains. *Numer. Math.*, 21:109–129, 1973.

[17] J. L. Guermond, P. Minev, and J. Shen. Error analysis of pressure-correction schemes for the time-dependent stokes equations with open boundary conditions. *SIAM J. Numer. Anal.*, 43(1):239–258, 2005.

[18] J. L. Guermond and J. Shen. A new class of truly consistent splitting schemes for incompressible flows. *J. Comput. Phys.*, 192(1):262–276, 2003.

[19] J. L. Guermond and J. Shen. Velocity-correction projection methods for incompressible flows. *SIAM J. Numer. Anal.*, 41(1):112–134, 2003.

[20] W. Heinrichs. Improved Lebesgue constants on the triangle. *J. Comput. Phys.*, 207(2):625–638, 2005.

[21] J. S. Hesthaven and C. H. Teng. Stable spectral methods on tetrahedral elements. *SIAM J. Sci. Comput.*, 21(6):2352–2380 (electronic), 2000.

[22] L. W. Ho and E. M. Rønquist. Spectral element solution of steady incompressible viscous free-surface flows. *Finite Elem. Anal. Des.*, 16(3-4):207–227, 1994.

[23] T. J. R. Hughes, W. K. Liu, and T. K. Zimmermann. Lagrangian-Eulerian finite element formulation for incompressible viscous flows. *Computer Methods in Applied Mechanics and Engineering*, 29(3):329–349, 1981.

[24] H. Kanchi and A. Masud. A 3D adaptive mesh moving scheme. *International Journal for Numerical Methods in Fluids*, 54(6-8):923–944, 2007.

[25] G. E. Karniadakis and S. J. Sherwin. *Spectral/hp element methods for computational fluid dynamics*. Oxford Universtity Press, Oxford, 2nd ed. edition, 2004.

[26] G. E. Karniadakis and S. J. Sherwin. *Spectral/hp element methods for computational fluid dynamics*, chapter Multi-dimensional formulation, pages 222–234. Oxford Universtity Press, Oxford, 2nd ed. edition, 2004.

[27] D. Kay, D. Loghin, and A. Wathen. A Preconditioner for the Steady–State Navier–Stokes Equations. *SIAM J. Sci. Comput.*, 24(1):237–256, 2002.

[28] D. Kay and E. Lungu. A Block Preconditioner for High–Order Mixed Finite Element Approximations to the Navier–Stokes Equations. *SIAM J. Sci. Comput.*, 27(6):1867–1880, 2006.

[29] M. F. Murphy, G. H. Golub, and A. J. Wathen. A Note on Preconditioning for Indefinite Linear Systems. *SIAM J. Sci. Comput.*, 21(6):1969–1972, 1999.

[30] F. Nobile. *Numerical approximation of fluid-structure interaction problems with application to haemodynamics*. PhD thesis, n°2458, EPF Lausanne, 2001.

[31] A. T. Patera. A spectral element methods for fluid dynamics: laminar flow in a channel expansion. *J. Comput. Phys.*, 54:468–488, 1984.

[32] G. Pena. *Spectral element approximation of the incompressible Navier-Stokes equations evolving in a moving domain and applications*. PhD thesis, École Polytechnique Fédérale de Lausanne, November 2009. n°4529.

[33] G. Pena and C. Prud'homme. Construction of a High Order Fluid-Structure Interaction Solver. *JCAM*, 2009. http://dx.doi.org/10.1016/j.cam.2009.08.093.

[34] C. Prud'homme. A domain specific embedded language in C++ for automatic differentiation, projection, integration and variational formulations. *Scientific Programming*, 14(2):81–110, 2006. http://iospress.metapress.com/link.asp?id=8xwd8r59hg1hmlcl.

[35] C. Prud'homme. Life: Overview of a unified C++ implementation of the finite and spectral element methods in 1D, 2D and 3D. In *Workshop On State-Of-The-Art In Scientific And Parallel Computing*, Lecture Notes in Computer Science, page 10. Springer-Verlag, 2006. Accepted.

[36] C. Prud'homme. Life: A modern and unified c++ implementation of finite-element and spectral-elements methods in 1d, 2d and 3d: overview and applications. In *ICIAM*, 2007. accepted.

[37] A. Quarteroni and L. Formaggia. Mathematical Modelling and Numerical Simulation of the Cardiovascular System. In *Modelling of Living Systems*, Handbook of Numerical Analysis Series. 2003.

[38] A. Quarteroni, R. Sacco, and F. Saleri. *Numerical Mathematics (Texts in Applied Mathematics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[39] A. Quarteroni, F. Saleri, and A. Veneziani. Analysis of the Yosida Method for the Incompressible Navier-Stokes Equations. *Journal de Mathématiques Pures et Appliquées*, 78(5):473–503, 1999.

[40] A. Quarteroni, F. Saleri, and A. Veneziani. Factorization methods for the numerical approximation of Navier-Stokes equations. *Comput. Methods Appl. Mech. Eng.*, 188(1-3):505–526, 2000.

[41] A. Quarteroni and A. Valli. *Numerical Approximation of Partial Differential Equations*. Springer-Verlag, Berlin, Heidelberg, New York, 1994.

[42] E. M. Rønquist. *Optimal spectral element methods for the unsteady three-dimensional incompressible Navier-Stokes equations.* PhD thesis, Massachusetts Institute of Technology, 1988.

[43] M. Sala and M. Heroux. Robust algebraic preconditioners with IFPACK 3.0. Technical Report SAND-0662, Sandia National Laboratories, 2005.

[44] F. Saleri and A. Veneziani. Pressure correction algebraic splitting methods for the incompressible Navier–Stokes equations. *SIAM J. Numer. Anal.*, 43(1):174–194, 2005.

[45] C. Schwab and M. Suri. Mixed hp finite element methods for Stokes and non-Newtonian flow. *Comput. Methods Appl. Mech. Engrg.*, 175:217–241, 1999.

[46] D. Silvester, H. Elman, D. Kay, and A. Wathen. Efficient Preconditioning of the Linearized Navier–Stokes Equations for Incompressible Flow. *J. Comput. Appl. Math.*, 128(1-2):261–279, 2001.

[47] P. Solin, K. Segeth, and I. Dolezel. *Higher-Order Finite Element Methods.* Chapman and Hall/CRC, Boca Raton, London, New York, 2004.

[48] K. Stanley. Klu: a "clark kent" sparse lu factorization algorithm for circuit matrices. In *2004 SIAM Conference on Parallel Processing for Scientific Computing (PP04)*, 2004.

[49] R. Stenberg and M. Suri. Mixed hp finite element methods for problems in elasticity and Stokes flow. *Numer. Math.*, 72(3):367–389, 1996.

[50] M. A. Taylor, B. A. Wingate, and R. E. Vincent. An algorithm for computing Fekete points in the triangle. *SIAM J. Numer. Anal.*, 38(5):1707–1720 (electronic), 2000.

[51] T. Warburton. An explicit construction for interpolation nodes on the simplex. *Journal of Engineering Mathematics*, 56(3):247–262, November 2006.

[52] C. Winkelmann. *Interior penalty finite element approximation of Navier-Stokes equations and application to free surface flows.* PhD thesis, Lausanne, 2007.

MOX Technical Reports, last issues

Dipartimento di Matematica "F. Brioschi",
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

25/2010 G. Pena, C. Prud'homme, Alfio Quarteroni:
High Order Methods for the Approximation of the Incompressible Navier-Stokes Equations in a Moving Domain

24/2010 Lorenzo Tamellini, Luca Formaggia, Edie Miglio, Anna Scotti:
An Uzawa iterative scheme for the simulation of floating boats

23/2010 Joakim Baeck, Fabio Nobile, Lorenzo Tamellini, Raul Tempone:
Stochastic Spectral Galerkin and collocation methods for PDEs with random coefficients: a numerical comparison

22/2010 Carlo D'Angelo, Paolo Zunino:
Numerical approximation with Nitsche's coupling of transient Stokes'/Darcy's flow problems applied to hemodynamics

21/2010 Niccolo' Grieco, Francesca Ieva, Anna Maria paganoni:
Provider Profiling Using Mixed Effects Models on a Case Study concerning STEMI Patients

20/2010 Fabio Nobile, Alfio Quarteroni, Ricardo Ruiz Baier:
Numerical solution of an active strain formulation for the electro-mechanical activity in the heart

19/2010 Loredana Gaudio, Alfio Quarteroni:
hN-adaptive spectral element discretization of optimal control problems for environmental applications

18/2010 Paola F. Antonietti, Nur Aiman Fadel, Marco Verani:
Modelling and numerical simulation of the polymeric extrusion process in textile products

17/2010 Alessandra Guglielmi, Francesca Ieva, Anna Maria Paganoni, Fabrizio Ruggeri:
A Bayesian random-effects model for survival probabilities after acute myocardial infarction

16/2010 Luca Formaggia, Anna Scotti:
Positivity and conservation properties of some integration schemes for mass action kinetics