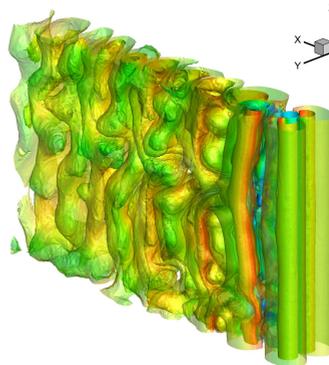
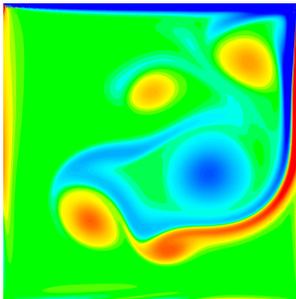


**Arbitrary high order discontinuous Galerkin methods
for the shallow water and incompressible Navier-Stokes
equations on unstructured staggered meshes**

Tavelli Maurizio



UNIVERSITÀ DEGLI STUDI DI TRENTO
Dipartimento di Matematica

2015

Doctoral thesis in **Applied Mathematics**, XXVII cycle.

Department of Mathematics,
University of Trento.

Academic year **2015/2016.**

Supervisor : **Prof. Dr.-Ing. Michael Dumbser**

University of Trento
Trento, Italy
2015

Alla mia famiglia Martina ed Emily

Preface

The material presented in this thesis has been developed during the three-year Doctoral School in Mathematics at the University of Trento under the supervision of prof. Michael Dumbser.

The research presented in this thesis was partially funded by the European Research Council (ERC) under the European Union's Seventh Framework Programme (FP7/2007-2013) within the research project *STiMulUs*, ERC Grant agreement no. 278267.

Some of the last simulations shown in this thesis have been made using SuperMUC supercomputer of the Leibniz Rechenzentrum in Munich (Germany), for which we gained access under the project “STiMulUs - Lagrangian Space-Time Methods for Multi-Fluid Problems on Unstructured Meshes”, proposals no. 2014112638.

Trento, December 2015

Tavelli Maurizio

Acknowledgments

I really want to thank my supervisor prof. Michael Dumbser for these wonderful three years of research. I have enjoyed them and I am honored to work in this great research group. I really thank Michael for giving me the possibility to touch by hands high level research in a very friendly environment that encourages also personal growth.

I want also to thank prof. Vincenzo Casulli for the discussions and the ideas given in these three years.

I thank my wife Martina and my little daughter Emily to be my Family and light each day of my life.

Contents

Abstract	x
1 Introduction	1
1.1 Incompressible Navier-Stokes equations	4
1.2 Shallow Water equations	5
1.3 Organization of the present PhD thesis	7
2 Two-dimensional shallow water equations	9
2.1 Overview	9
2.1.1 Governing equations	9
2.2 DG scheme for the 2D shallow water equations	10
2.2.1 Unstructured grid	10
2.2.2 Basis functions	10
2.2.3 Semi-Implicit DG scheme	12
2.2.4 Nonlinear convection	19
2.2.5 Integration	20
2.2.6 Boundary conditions	22
2.2.7 Curved Elements	26
2.3 Numerical tests	28
2.3.1 Convergence test	28
2.3.2 Well-balancedness (C-property)	31
2.3.3 Smooth surface wave propagation	33
2.3.4 Dambreak over a bottom step	37
2.3.5 Low Froude number flow around a circular cylinder	38
3 Two-dimensional incompressible Navier-Stokes equations	47
3.1 Overview	47
3.1.1 Governing equations	47
3.2 Space-time DG scheme for the 2D incompressible Navier-Stokes equations	48
3.2.1 Space-time extension of the unstructured staggered grid	48
3.2.2 Space-time basis functions	48

3.2.3	Semi-implicit space-time DG scheme	49
3.2.4	Nonlinear convection-diffusion	56
3.2.5	Pressure correction formulation and final algorithm	58
3.3	The particular case $p_\gamma = 0$ using a simple Crank-Nicolson time discretization	60
3.3.1	Remarks on the main system and further improvements	62
3.3.2	Remarks on the stability	67
3.4	Splitting of the space-time matrices into a spatial and temporal part	71
3.5	Stability analysis of the space-time DG method	71
3.6	Numerical test problems	77
3.6.1	Numerical test for the space only high order method	77
3.6.1.1	Convergence test	77
3.6.1.2	Blasius boundary layer	78
3.6.1.3	Lid-driven cavity flow	82
3.6.1.4	Backward-facing step.	83
3.6.1.5	Rotational flow past a circular half-cylinder	86
3.6.1.6	Flow over a circular cylinder	87
3.6.2	Numerical tests for the space-time DG method	93
3.6.2.1	Convergence test using a manufactured solution	93
3.6.2.2	The Womersley problem	93
3.6.2.3	Taylor-Green vortex	97
3.6.2.4	Double shear layer	98
3.6.2.5	Flow over a circular cylinder	99
4	Three-dimensional incompressible Navier-Stokes equations	105
4.1	Overview	105
4.1.1	Governing equations	105
4.2	DG scheme for the 3D incompressible Navier-Stokes equations	106
4.2.1	Unstructured staggered space-time grid	106
4.2.2	Space-Time basis functions	106
4.2.3	Semi-Implicit DG scheme	110
4.2.4	Nonlinear convection-diffusion	110
4.2.4.1	An alternative implementation for the viscous contribution	111
4.2.5	Extension to a pressure correction formulation	115
4.2.6	Remarks on the special case of piecewise constant polynomials in time ($p_\gamma = 0$)	116
4.2.6.1	Pressure system	116

4.2.6.2	Viscous system	117
4.2.7	Extension to curved elements	118
4.3	Numerical test problems	119
4.3.1	Three-dimensional lid driven cavity	119
4.3.2	Convergence test	122
4.3.3	Taylor Green Vortex	126
4.3.4	Womersley Profile	132
4.3.5	Blasius boundary layer	136
4.3.6	Backward-facing step.	138
4.3.7	Flow around a sphere	141
4.3.8	Flow past a circular cylinder	146
4.4	A discussion about the dual convective-viscous computation	150
5	Conclusions and Outlook	155
A	Appendixes	157
A.1	Further notes about $2D$ shallow water	157
A.1.1	Compute the nonlinear convection term	157
A.2	High Performance Computing	160
A.2.1	Some details about MPI parallel implementation	160
A.2.2	Periodic boundary conditions using MPI	163
A.2.3	Some $2D$ examples	164
A.2.4	A three dimensional test	167
A.3	Solving the Laplace equation for $3D$ isoparametric elements	171
A.4	Isoparametric Hexahedral characterization	174
A.5	A brief discussion about optimal order estimation	176
A.5.1	Minimize the computational cost	178
A.5.1.1	One dimensional case	178
A.5.1.2	Two dimensional case	180
A.6	Experimental estimation of the optimal polynomial degree	183
	Bibliography	187
	List of Tables	203
	List of Figures	205

Abstract

In this work we present a new class of well-balanced, arbitrary high order accurate semi-implicit discontinuous Galerkin methods for the solution of the shallow water and incompressible Navier-Stokes equations on staggered unstructured curved meshes. Isoparametric finite elements are used to take into account curved domain boundaries.

Regarding two-dimensional shallow water equations, the discrete free surface elevation is defined on a primal triangular grid, while the discrete total height and the discrete velocity field are defined on an edge-based staggered dual grid. Similarly, for the two-dimensional incompressible Navier-Stokes case, the discrete pressure is defined on the main triangular grid and the velocity field is defined on the edge-based staggered grid. While staggered meshes are state of the art in classical finite difference approximations of the incompressible Navier-Stokes equations, their use in the context of high order DG schemes is novel and still quite rare. High order (better than second order) in time can be achieved by using a space-time finite element framework, where the basis and test functions are piecewise polynomials in both space and time. Formal substitution of the discrete momentum equation on the dual grid into the discrete continuity equation on the primary grid yields a very sparse system for the scalar pressure involving only the direct neighbor elements, so that it becomes a block four-point system in $2D$ and a block five-point system for $3D$ tetrahedral meshes. The resulting linear system is conveniently solved with a matrix-free GMRES algorithm. Note that the same space-time DG scheme on a collocated grid would lead to ten non-zero blocks per element in $2D$ and seventeen non-zero blocks in $3D$, since substituting the discrete velocity into the discrete continuity equation on a collocated mesh would involve also neighbors of neighbors. From numerical experiments we find that our linear system is well-behaved and that the GMRES method converges quickly even *without* the use of *any* preconditioner, which is a unique feature in the context of high order implicit DG schemes. A very simple and efficient Picard iteration is then used in order to derive a space-time pressure correction algorithm that achieves also high order of accuracy in time, which is in general a non-trivial task in the context of high order discretizations for the incompressible Navier-Stokes equations. The special case of high order in space low order in time allows us to recover further regularity about the main linear system for the pressure, such as the symmetry and the positive semi-definiteness in the general case. This allows us to use a very fast linear solver such as the conjugate gradient (CG) method. The flexibility and accuracy of high order space-time DG methods on

curved unstructured meshes allows to discretize even complex physical domains with very coarse grids in both space and time.

We will further extend the previous method to three-dimensional incompressible Navier-Stokes system using a tetrahedral main grid and a corresponding face-based hexahedral dual grid. The resulting dual mesh consists in non-standard 5-vertex hexahedral elements that cannot be represented using tensor products of one dimensional basis functions. Indeed a modal polynomial basis will be used for the dual mesh.

This new family of numerical schemes is verified by solving a series of typical numerical test problems and by comparing the obtained numerical results with available exact analytical solutions or other numerical reference data. Furthermore, the comparison with available experimental results will be presented for incompressible Navier-Stokes equations.

1 Introduction

Fluid dynamics represents a vast sector of outcoming engineering and mathematical research with a wide importance to real world applications. There are several applications to free-surface flows, incompressible and compressible fluid dynamics. Other important applications concern the evolution of magnetic fluids and incorporate both fluid mechanics and Maxwell equations with a strong interaction between the magnetic and the motion part.

If we take a look at the shallow water or the incompressible Navier-Stokes equations we apparently have a system of governing partial differential equations that involves only a few unknown variables, such as the pressure and the velocity field and the known bottom bathymetry in the case of the shallow water equations. On the other hand, once we start to study the equations, we discover a series of non-trivial behaviours and the possibility to generate really complex solutions. This is due to the nonlinearity of the governing PDE systems.

Several of these behaviors involving both small and large scale structures, can be observed in experiments and, in general, we do not have an analytical solution, especially in the complete three-dimensional case.

In the great majority of cases, we have to solve the equations numerically in order to forecast the expected behaviour of the fluid. In this sense there are several procedures that we can use, such as the finite volume, the finite difference or the finite element method. At this point, I would briefly review the difference between finite differences (Taylor series expansions and the discrete solution being represented by point-values $u_i^n = u(x_i, t^n)$), finite volumes (integral form of the conservation law and the discrete solution being represented by cell averages $u_i^n = \frac{1}{\Delta x} \int_{x_{i-1/2}}^{x_{i+1/2}} u(x, t^n) dx$) and finite elements with their discrete solution represented in a function space spanned by piecewise polynomial basis functions $u_h(x, t^n) = \sum_i \phi_i(x) \hat{u}_i^n$) at the example of a simple scalar conservation law $u_t + f_x = 0$.

One of the oldest and one of the simplest possible discretizations of the equations of fluid mechanics consists in an explicit finite difference approximation. It leads to an explicit formula for the evolution of the unknown variables, but it imposes a rather severe time step restriction in order to get a stable scheme, the

famous CFL condition, first discovered by Courant, Friedrichs and Lewy in [59]. On the contrary, fully implicit schemes usually avoid this limitation, but they require the solution of a linear or even nonlinear algebraic system, which might become quite cumbersome. Indeed, there is a class of so called semi-implicit schemes that mix both explicit and implicit contributions. An important contribution to semi-implicit schemes applied to fluid mechanics and related topics was made by Harlow and Welch [95], Patankar and Spalding [126], and by V. Casulli [31, 32, 33, 34, 36, 37, 38, 39, 35, 40, 42, 41, 43, 44]. The introduction of an auxiliary parameter $\theta \in [0, 1]$ allows us to write a generic time discretization for $\xi' = f(\xi)$ such as

$$\frac{\xi^{n+1} - \xi^n}{\Delta t} = \theta f(\xi^n) + (1 - \theta)f(\xi^{n+1}),$$

so that for $\theta = 0$ we get a fully explicit Euler method and for $\theta = 1$ we have a fully implicit one, but we can choose also intermediate values. The special case of $\theta = \frac{1}{2}$ was studied by J.Crank and P.Nicolson in the original paper of 1947 [60] for heat-conduction type equations and is therefore called the Crank-Nicolson scheme. In general we have an higher order accuracy in this last case as well as unconditional stability for $\theta \in [\frac{1}{2}, 1]$, see [36].

Finite element schemes are typically used to achieve high order solution and can be divided into a class of continuous finite elements and another class of discontinuous Galerkin (DG) finite element methods. The continuous finite element case in general involves high order polynomials that are continuous on each boundary interface; on the contrary, the discontinuous case allows jumps on the element boundaries. This second case is particularly suitable when we have discontinuous solutions, such as shocks or discontinuous boundary conditions or bathymetry in the shallow water context. The price of this generality is the requirement of more information to be represented in each element (i.e. we need to duplicate the degrees of freedom on the element interface, so we have more degrees of freedom for a general numerical solution).

The DG finite element method has been originally introduced by Reed and Hill [132] for the solution of neutron transport equations and has been successively extended to general nonlinear hyperbolic conservation laws by Cockburn and Shu in a famous series of papers [56, 55, 54, 53, 57]. Jiang and Shu [108] were able to derive a cell entropy inequality for semi-discrete DG schemes and, as a consequence, they were also able to prove nonlinear stability of the DG method in L_2 norm. General unstructured meshes and curved isoparametric elements can be naturally included in the DG framework, together with hanging nodes and hp -adaptivity [106, 105, 104]. This easily explains the recent

and continuously growing success of the DG method. Indeed its intrinsically high scalability for parallel high performance computing (HPC) as well as high resolution make the DG methods particularly suitable for large scale simulations of complex phenomena and therefore real world applications. However, explicit DG schemes suffer from a very severe time step restriction, which is the most severe the higher the polynomial degree of the basis and test functions. High order implicit time discretizations for DG methods are possible [128, 127, 11, 123, 61, 10, 8, 18, 96, 97], but the resulting system matrices are denser and have a worse condition number compared to classical continuous finite elements. Therefore, recent attempts have been made to improve either the CFL condition of explicit DG schemes [155], or to use *semi-implicit* DG schemes [63, 64, 65, 91, 151], which combine the simplicity of explicit methods for nonlinear hyperbolic PDEs with the stability and efficiency of implicit time discretizations.

We can finally use a single grid or a set of different overlapping grids, consisting of a main grid and a staggered dual one where the main variables are not located at the same points. The use of staggered grids is a very common choice in the finite difference community and leads to a very natural discretization of the velocity divergence operator on the main grid. On the contrary, the use of a collocated grid needs some interpolations of the velocities at the boundary interface, see e.g. [120]. The use of staggered grids in the context of high order DG schemes is novel and still quite rare.

Very recently, a new class of spatially arbitrary high order accurate semi-implicit DG schemes has been introduced in [70] using *staggered* grids. The use of a staggered mesh significantly improves the sparsity pattern of the resulting linear algebraic system to be solved. While the method introduced in [70] leads to a simple block penta-diagonal system on a Cartesian grid in two space dimensions thanks to the use of mesh staggering, the semi-implicit collocated grid approach presented in [151] requires the solution of a block 13-diagonal system on $2D$ Cartesian meshes. Both approaches [70] and [151] have in common that at the end a linear equation system is solved only for one scalar quantity of the PDE (e.g. the water depth, or the fluid pressure), while in [63, 64, 65] a coupled system for all unknown state variables of the PDE must be solved. To the knowledge of the author, the first staggered DG schemes have been recently proposed in [117, 117], [51, 52, 49, 50] and [70]. While the methods [117, 118] use a *vertex-based* staggering of the mesh, the schemes [51, 52, 49, 50] and [70] apply an *edge-based* grid staggering.

In this PhD thesis we want to extend the idea of staggered semi-implicit methods to high order DG schemes for the shallow water and the incompressible

Navier-Stokes equations in order to achieve, from one side, high resolution and grid generality typical of the finite element community and, from the other side, well conditioned and very sparse linear systems known from the finite difference community.

1.1 Incompressible Navier-Stokes equations

The Navier-Stokes equations are the fundamental governing PDE system in the context of fluid dynamics. In general we can describe compressible and incompressible fluids and we have in both cases to satisfy a momentum conservation, as well as a mass conservation law. The mass continuity equation is in general given in the form

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{v}) = 0,$$

where ρ is the fluid density and $\vec{v} = (u, v, w)$ represents the velocity vector field. For the case of incompressible fluids where $\rho = \text{constant}$, the previous relation becomes consistent with a divergence-free condition for the fluid velocity

$$\nabla \cdot \vec{v} = 0.$$

In addition, the momentum equation, that reads for the incompressible case such as

$$\frac{\partial \vec{v}}{\partial t} + (\vec{v} \cdot \nabla) \vec{v} + \nabla p = \nu \Delta \vec{v},$$

which contains a nonlinear convective term, a parabolic contribution in the general case of $\nu > 0$ and a pressure contribution. Here $p = P/\rho$ is the normalized pressure and $\nu = \mu/\rho$ represents the kinematic viscosity coefficient. Applying the divergence operator to the momentum equation we get

$$\begin{aligned} \nabla \cdot \left[\frac{\partial \vec{v}}{\partial t} + (\vec{v} \cdot \nabla) \vec{v} \right] + \nabla \cdot \nabla p &= \nu \nabla \cdot \Delta \vec{v}, \\ \frac{\partial}{\partial t} (\nabla \cdot \vec{v}) + \nabla \cdot [(\vec{v} \cdot \nabla) \vec{v}] + \Delta p &= \nu \nabla \cdot \Delta \vec{v}. \end{aligned}$$

Using $\Delta \vec{v} = \nabla(\nabla \cdot \vec{v}) - \nabla \times \nabla \times \vec{v}$, the properties of the divergence operator and the divergence free condition $\nabla \cdot \vec{v} = 0$, we derive the continuous pressure Poisson equation

$$\Delta p = -\nabla \cdot (\vec{v} \cdot \nabla) \vec{v}. \tag{1.1}$$

The main difficulty in the numerical solution of the incompressible Navier-Stokes equations lies in this pressure Poisson equation and the associated linear equation system to be solved on the discrete level that is obtained by substitution of the velocity field given by the momentum equation into the continuity equation. This is closely related to the elliptic nature of these equations, where boundary conditions affect instantly the solution everywhere inside the domain. While finite difference schemes for the incompressible Navier-Stokes equations are well-established for several decades [95, 126, 125, 152], as well as continuous finite element methods [146, 22, 107, 84, 153, 101, 102], the development of high order discontinuous Galerkin (DG) finite element methods for the incompressible Navier-Stokes equations is still a very active topic of ongoing research. Several high order DG methods for the incompressible Navier-Stokes equations have been recently presented in literature, see for example [9, 140, 83, 122, 135, 136, 62, 111], or the work of Bassi et al. [7] based on the technique of artificial compressibility, originally introduced by Chorin in [47, 48]. In this thesis we present a new family of arbitrary high order semi-implicit space-time DG schemes for the solution of the incompressible Navier-Stokes equations on staggered unstructured curved meshes in two and three space dimensions, following the philosophy of semi-implicit staggered finite difference schemes, which have been successfully used in the past for the solution of the incompressible Navier-Stokes equations [95, 126, 125, 152] and the free surface shallow water and Navier-Stokes equations, see [103, 36, 37, 40, 154, 33].

1.2 Shallow Water equations

The two-dimensional shallow water equations can be derived from the mass conservation law and the conservation of momentum and can be obtained depth-averaging of the three-dimensional incompressible Navier-Stokes equations with moving free surface. As a consequence, it can be applied to problems where the vertical dynamics can be neglected compared with the one in the $x - y$ plane, such as tidal waves or the propagation of tsunami waves in the ocean, or the flooding occurring during a storm surge and in river basins. The incompressible Navier-Stokes equations in the presence of gravity are given by

$$\begin{aligned}\nabla \cdot \vec{v} &= 0 \\ \frac{\partial \vec{v}}{\partial t} + (\vec{v} \cdot \nabla) \vec{v} + \nabla p &= \nu \Delta \vec{v} + f\end{aligned}$$

where $f = (0, 0, -g)^\top$ and g is the gravity constant. Assuming that the free surface and the bottom profile can be expressed as a single valued function (i.e.

$z = \eta(x, y, t)$ and $z = -b(x, y)$), we derive the following equations:

$$\eta_t + u^s \eta_x + v^s \eta_y = w^s, \quad (1.2)$$

$$u^s b_x + v^s b_y + w^s = 0, \quad (1.3)$$

where $(u^s, v^s, w^s) = \vec{v}^s = \vec{v}(x, y, \eta, t)$ and $(u^b, v^b, w^b) = \vec{v}^b = \vec{v}(x, y, -b, t)$ represent respectively the velocity field computed at the free surface and at the bottom; $b = b(x, y)$ is the water depth measured from undisturbed water surface. The first condition (1.2) is also known as the kinetic condition of the free surface while the second one (1.3) states that no water passes through the bottom. Integration of the continuity equation from the bottom $-b$ to the free surface η and the use of conditions (1.2)-(1.3) lead to the following equation:

$$\eta_t + \left(\int_{-b}^{\eta} u dz \right)_x + \left(\int_{-b}^{\eta} v dz \right)_y = 0. \quad (1.4)$$

Assuming that the vertical dynamics can be neglected compared with the one in the x - y plane we can derive that $p_z = -g$ and hence the following expression for the hydrostatic pressure:

$$p(x, y, z, t) = p_a(x, y, t) + g(\eta(x, y, t) - z). \quad (1.5)$$

Here p_a represents the atmospheric pressure that is assumed constant, so that $p_x = g\eta_x$ and $p_y = g\eta_y$ can be substituted in the momentum equation. We finally derive the frictionless $2D$ vertically averaged model by setting $\nu = 0$; defining the averaged velocities $\bar{u}(x, y, t) = \frac{1}{H} \int_{-b}^{\eta} u dz$, $\bar{v}(x, y, t) = \frac{1}{H} \int_{-b}^{\eta} v dz$ where $H(x, y, t) = b(x, y) + \eta(x, y, t)$ is the total water depth; vertical integrating the momentum equation for u and v ; and using standard approximations of the vertical velocities with their vertically average. We obtain

$$\begin{aligned} (H\bar{u})_t + (H\bar{u}\bar{u})_x + (H\bar{u}\bar{v})_y &= -gH\eta_x, \\ (H\bar{v})_t + (H\bar{v}\bar{u})_x + (H\bar{v}\bar{v})_y &= -gH\eta_y, \end{aligned}$$

from the momentum equation and

$$\eta_t + (H\bar{u})_x + (H\bar{v})_y = 0 \quad (1.6)$$

from the free surface equation. In the two dimensional context we further identify $u(x, y) = \bar{u}(x, y)$ and $v(x, y) = \bar{v}(x, y)$. The previous system is then

written in terms of conserved quantities $U = Hu$, $V = Hv$ and in compact vectorial form as

$$\frac{\partial}{\partial t}\eta + \nabla \cdot \vec{V} = 0, \quad (1.7)$$

$$\frac{\partial}{\partial t}\vec{V} + \nabla \cdot \mathbf{F} + gH\nabla\eta = 0, \quad (1.8)$$

where

$$\mathbf{F} = \frac{1}{H} \begin{pmatrix} UU & UV \\ VU & VV \end{pmatrix},$$

and $\vec{V} = (U, V) = (Hu, Hv)$.

The unstructured semi-implicit staggered DG scheme presented here for the solution of the two-dimensional shallow water equations can be seen as a natural high order extension of the family of staggered semi-implicit finite volume and finite difference schemes introduced by Casulli et al. in [32, 37, 154, 40, 20] on Cartesian and orthogonal unstructured meshes using an edge-based grid staggering, where the dual mesh is chosen like the dual grid used in [15, 150] and [49]. These schemes have been recently extended to handle subgrid topography features and to treat wetting and drying in a rigorous nonlinear and mass conservative way, see [34, 39]. Extensions of these efficient semi-implicit methods to the simulation of subsurface flow in porous media and to blood flow in systems of compliant arteries have been presented in [43, 38, 81]. A rigorous convergence analysis of these nonlinear algorithms was provided in the work of Brugnano et al. [24, 25] and in [44]. A momentum conservative formulation of these semi-implicit schemes, similar to the one used here, has been introduced by Stelling et al. in [141, 114].

1.3 Organization of the present PhD thesis

The main achievements of the PhD research activity reported here are:

- the natural extension of the method introduced by M. Dumbser and V. Casulli in [70] to unstructured staggered meshes where the complete velocity field $\vec{V} = (U, V)$ is defined on a single edge-based staggered dual grid;
- the extension of the previous method to the two-dimensional incompressible Navier-Stokes equations obtaining a semi-implicit method, of high order of accuracy in space;

- the generalization of the previous method to arbitrary high order in space and time, using space-time polynomial test and basis functions and a simple Picard procedure, that leads to a novel space-time pressure correction algorithm;
- the extension of the arbitrary high order accurate space-time DG scheme on staggered grids to three-dimensional tetrahedral-hexahedral meshes.

In this last case two treatments can be implemented for the convective-viscous contribution: the natural extension of the two dimensional method or a mean velocity method following the idea used in [70]. Due to higher computational performance, the latter methodology is preferable, but some comparisons and discussions can be found in the work presented here.

The rest of the thesis is organized as follows. In Chapter 2 we derive and test the method for the two-dimensional shallow water equations. In Chapter 3 we extend the method to the two-dimensional incompressible Navier-Stokes equations introducing a novel family of arbitrary high order accurate staggered space-time DG schemes. Indeed, a discussion about the special case of low order in time, high order in space is given in Section 3.3. Chapter 4 contains results and discussions about the fully three-dimensional incompressible Navier-Stokes equations. In Chapter 5 there are some concluding remarks and a short outlook to future work. Finally, the appendix contains several technical details, as well as a discussion about the parallel implementation using MPI for staggered grids.

2 Two-dimensional shallow water equations

2.1 Overview

In this chapter we will derive a high order semi-implicit DG method for the two dimensional shallow water equations on staggered unstructured meshes. The presented method can be seen as the natural extension of the one proposed in [70] for 2D Cartesian grid to unstructured curved meshes. The chapter is organized as follows: in Section 2.2 a high order staggered semi-implicit discontinuous Galerkin scheme is derived for the 2D shallow water equations. in Section 2.2.7 we extend the method also to curved isoparametric elements and finally, in Section 2.3, the method is validated against exact and numerical reference solutions.

2.1.1 Governing equations

The frictionless two dimensional shallow water equations, written in terms of conserved quantities, are given by

$$\frac{\partial}{\partial t}\eta + \nabla \cdot \vec{V} = 0, \quad (2.1)$$

$$\frac{\partial}{\partial t}\vec{V} + \nabla \cdot \mathbf{F} + gH\nabla\eta = 0, \quad (2.2)$$

where $\eta = \eta(x, y, t)$ indicates the free surface elevation; $H = H(x, y, t)$ is the total water height, defined as $H(x, y, t) = \eta(x, y, t) + b(x, y)$ where b is a specified bottom profile; $\vec{V} = (U, V)$ is the depth-averaged momentum vector with $U = Hu$ and $V = Hv$; u and v are the velocity components respectively in the x and y directions; and $\vec{v} = (u, v)$ denotes the velocity vector; $g = 9.81$ is the gravity acceleration and $\mathbf{F} = \mathbf{F}(\vec{V}, H)$ is the flux tensor of the nonlinear convective terms, namely:

$$\mathbf{F} = \frac{1}{H} \begin{pmatrix} UU & UV \\ VU & VV \end{pmatrix}.$$

We further use the abbreviation $L(\vec{V}) = \frac{\partial}{\partial t}\vec{V} + \nabla \cdot \mathbf{F}$.

2.2 DG scheme for the 2D shallow water equations

2.2.1 Unstructured grid

Let $\Omega \subset \mathbb{R}^2$ denote the computational domain, which is covered with a set of non-overlapping triangles \mathbf{T}_i with $i = 1 \dots N_e$, defined on a set of nodes $\aleph(\Omega)$. By denoting with N_d the total number of edges, the j -th edge will be called Γ_j . We will indicate with $\mathcal{B}(\Omega)$ the set of indexes corresponding to boundary edges (i.e. $\mathcal{B}(\Omega) := \{j \in [1, N_d] \mid \Gamma_j \text{ is a boundary edge}\}$). Every triangle \mathbf{T}_i has exactly three edges indexed by $S_i = \{j \in [1, N_d] \mid \Gamma_j \text{ is an edge of } \mathbf{T}_i\}$. For every $j \in [1 \dots N_d] - \mathcal{B}(\Omega)$ there exist two triangles i_1 and i_2 that share Γ_j . It is possible to assign arbitrarily a left and a right triangle called $\ell(j)$ and $r(j)$, respectively. The standard positive direction is assumed to be from the left to the right. Let \vec{n}_j denote the standard normal vector defined on the edge j and oriented with respect to the standard direction. For every triangular element i and edge $j \in S_i$, the triangle \mathbf{T}_i could be the right or the left element with respect to j ; $\wp(i, j)$ will denote the *neighbor* triangular element of \mathbf{T}_i on the common edge Γ_j .

For every $j \in [1, N_d] - \mathcal{B}(\Omega)$ the quadrilateral element associated to j is called \mathbf{R}_j and it is defined by the two centers of gravity of $\ell(j)$ and $r(j)$ and the two terminal nodes of Γ_j , see also [15, 150]. The intersection element $\mathbf{R}_j \cap \mathbf{T}_i$ for every $i, j \in S_i$, is a non-empty element indicated with $\mathbf{T}_{i,j}$ (see Figure 2.1). The mesh of triangular elements $\{\mathbf{T}_i\}_{i \in [1, N_e]}$ will be called *main grid* or *primary grid* and the quadrilateral grid $\{\mathbf{R}_j\}_{j \in [1, N_d]}$ will be called *dual grid*. On the dual grid we define the same quantities as for the main grid, briefly: N_l is the total amount of edges of \mathbf{R}_j ; Γ_l indicates the l -th edge; $\forall j$, the set of edges l of j is indicated with S_j ; $\forall l$, $\ell_{jl}(l)$ and $r_{jl}(l)$ are the left and the right quadrilateral element, respectively; $\forall l$, \vec{n}_l is the standard normal vector defined on l and assumed positive with respect to the standard orientation on l (defined, as above, from the left to the right). Finally, each triangle \mathbf{T}_i is defined starting from an arbitrary node and oriented in counter-clockwise direction. Similarly, each quadrilateral element \mathbf{R}_j is defined starting from $\ell(j)$ and oriented in counter-clockwise direction.

2.2.2 Basis functions

We now want to construct a polynomial basis for polynomials of degree p on the standard reference elements needed for our staggered grid algorithm. In the one dimensional case this could be efficiently done using the Lagrange inter-

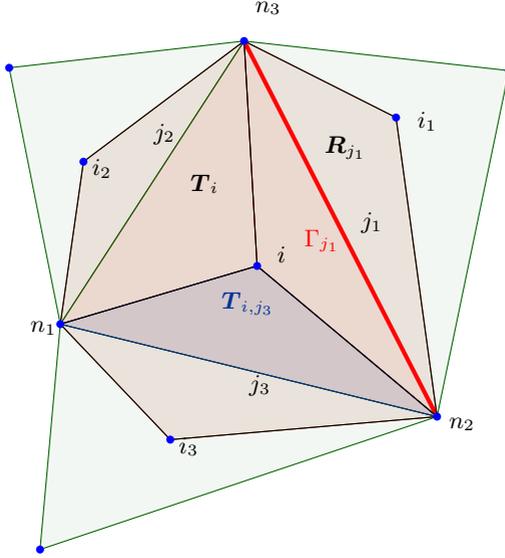


Figure 2.1: Example of a triangular mesh element with its three neighbors and the associated staggered edge-based dual control volumes, together with the notation used throughout the chapter.

polation polynomials passing through the Gauss-Legendre quadrature points [142], which was the nodal basis used in [70]. In this thesis we will define the polynomial basis on the primary triangular grid using the standard nodal approach of conforming continuous finite elements.

The standard (or reference) triangle is defined as $T_{std} = \{(\xi, \gamma) \in \mathbb{R}^{2,+} \mid \gamma \leq 1 - \xi\}$ while the standard (or reference) square is defined as $R_{std} = \{(\xi, \gamma) \in \mathbb{R}^{2,+} \mid \xi \leq 1, \gamma \leq 1\}$. On T_{std} the nodes associated with the basis functions are defined as

$$\vec{\xi}_k = (\xi_{k_1}, \gamma_{k_2}) = \left(\frac{k_1}{p}, \frac{k_2}{p} \right), \quad (2.3)$$

with the multi-index $k = (k_1, k_2)$ and the index ranges $0 \leq k_1 \leq p$ and $0 \leq$

$k_2 \leq p - k_1$. We can construct the standard nodal basis for T_{std} by imposing the classical Lagrange interpolation condition

$$\phi_k(\vec{\xi}_l) = \delta_{kl}, \quad (2.4)$$

between the k -th basis function $\phi_k = \phi_k(\vec{\xi}_l)$ and the l -th nodal point $\vec{\xi}_l$ in the reference triangle. In this way $N_\phi = \frac{(p+1)(p+2)}{2}$ basis functions $\{\phi_k\}_{k \in [1, N_\phi]}$ are obtained. In (2.4) δ_{kl} denotes the classical Kronecker symbol. Similarly, we obtain another set of nodal basis functions on the unit square R_{std} that will be indicated with $\{\psi_l\}_{l \in [1, N_\psi]}$ where $N_\psi = (p+1)^2$. On R_{std} the nodes are given by the tensor products of $p+1$ one-dimensional Newton-Cotes quadrature points, see [142], hence by the same formula (2.3), but with the index ranges $0 \leq k_1 \leq p$ and $0 \leq k_2 \leq p$. In [70] tensor-products of one-dimensional Gauss-Legendre quadrature points have been used on quadrilaterals. More details on nodal DG finite element methods on *collocated grids* can be found for example in [90, 99, 112, 113, 87, 100].

2.2.3 Semi-Implicit DG scheme

We define the spaces of piecewise polynomials used on the main and the dual grid as follows:

$$\begin{aligned} V_h^m &= \{\phi : \phi|_{\mathbf{T}_i} \in \mathbb{P}^p(\mathbf{T}_i), \forall i \in [1, N_e]\}, \\ V_h^d &= \{\psi : \psi|_{\mathbf{R}_j} \in \mathbb{Q}^p(\mathbf{R}_j), \forall j \in [1, N_d] - \mathcal{B}(\Omega)\}, \end{aligned} \quad (2.5)$$

where $\mathbb{P}^p(\mathbf{T}_i)$ is the space of polynomials of degree at most p on \mathbf{T}_i , while $\mathbb{Q}^p(\mathbf{R}_j)$ is the space of tensor products of one-dimensional polynomials of degree at most p on \mathbf{R}_j .

We define the discrete free surface elevation $\eta_h(x, y, t)$ on the main grid, namely $\eta_h \in V_h^m$ and so $\eta_h(x, y, t)|_{\mathbf{T}_i} = \eta_i(x, y, t)$. The discrete total height H_h and the momentum vector field \vec{V}_h are defined on the dual grid (i.e. $H_h, \vec{V}_h \in V_h^d$), so that $H_h(x, y, t)|_{\mathbf{R}_j} = H_j(x, y, t)$ and $\vec{V}_h(x, y, t)|_{\mathbf{R}_j} = \vec{V}_j(x, y, t)$.

The numerical solution of (2.1)-(2.2) is represented by piecewise polynomials as:

$$\eta_i(x, y, t) = \sum_{l=1}^{N_\phi} \phi_l(x, y) \hat{\eta}_{l,i}(t) =: \boldsymbol{\phi}(x, y) \hat{\boldsymbol{\eta}}_i(t), \quad (2.6)$$

$$H_j(x, y, t) = \sum_{l=1}^{N_\psi} \psi_l(x, y) \hat{H}_{l,j}(t) =: \boldsymbol{\psi}(x, y) \hat{\mathbf{H}}_j(t), \quad (2.7)$$

$$\vec{V}_j(x, y, t) = \sum_{l=1}^{N_\psi} \psi_l(x, y) \hat{V}_{l,j}(t) =: \boldsymbol{\psi}(x, y) \hat{\mathbf{V}}_j(t), \quad (2.8)$$

where the vector of basis functions $\phi(x, y)$ is generated from $\phi(\xi, \gamma)$ on T_{std} , while $\boldsymbol{\psi}(x, y)$ is generated from the set of basis functions $\boldsymbol{\psi}(\xi, \gamma)$ on R_{std} .

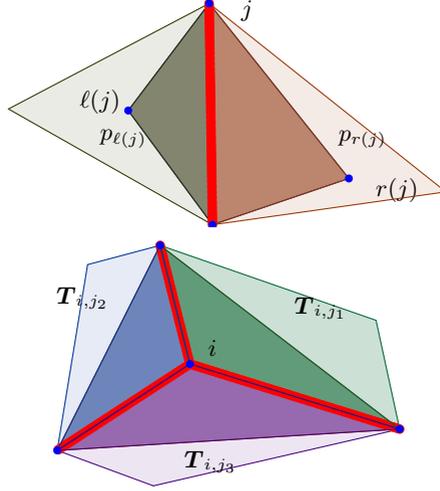


Figure 2.2: Jumps of η on the main grid (top) and of \vec{V} on the dual grid (bottom) highlighted by the bold red lines.

Multiplying Eq. (2.1) by ϕ and integrating over a control volume \mathbf{T}_i one gets, for every $k = 1 \dots N_\phi$

$$\int_{\mathbf{T}_i} \phi_k \frac{\partial \eta}{\partial t} d\mathbf{x} + \int_{\mathbf{T}_i} \phi_k \nabla \cdot \vec{V} d\mathbf{x} = 0. \quad (2.9)$$

Similarly, multiplication of the momentum equation (2.2) by $\boldsymbol{\psi}$ and integrating

over a control volume \mathbf{R}_j one obtains, componentwise,

$$\int_{\mathbf{R}_j} \psi_k \left(\frac{\partial \vec{V}}{\partial t} + \nabla \cdot \mathbf{F} \right) d\mathbf{x} + g \int_{\mathbf{R}_j} \psi_k H \nabla \eta d\mathbf{x} = 0, \quad (2.10)$$

for every $j = 1 \dots N_d$ and $k = 1 \dots N_\psi$. Using integration by parts Eq. (2.9) yields

$$\int_{\mathbf{T}_i} \phi_k \frac{\partial \eta}{\partial t} d\mathbf{x} + \oint_{\partial \mathbf{T}_i} \phi_k \vec{V} \cdot \vec{n}_i ds - \int_{\mathbf{T}_i} \nabla \phi_k \cdot \vec{V} d\mathbf{x} = 0, \quad (2.11)$$

where \vec{n}_i indicates the outward pointing unit normal vector. The discrete free surface elevation η_i is defined on the triangles and, in general, presents a discontinuity on Γ_j . It is useful to specify that every polynomial $\phi(x, y) = \phi^{(i)}(x, y)$ depends on the control volume and $\psi(x, y) = \psi^{(j)}(x, y)$. Also the discrete total height H_j and the velocity field \vec{V}_j jump on the edges of \mathbf{R}_j (see Figure 2.2). So equations (2.10) and (2.11) have to be split, unambiguously, as

$$\int_{\mathbf{T}_i} \phi_k^{(i)} \frac{\partial \eta_i}{\partial t} d\mathbf{x} + \sum_{j \in \mathcal{S}_i} \left(\int_{\Gamma_j} \phi_k^{(i)} \vec{V}_j \cdot \vec{n}_{ij} ds - \int_{\mathbf{T}_{i,j}} \nabla \phi_k^{(i)} \cdot \vec{V}_j d\mathbf{x} \right) = 0, \quad (2.12)$$

and

$$\begin{aligned} & \int_{\mathbf{R}_j} \psi_k^{(j)} \left(\frac{\partial \vec{V}_j}{\partial t} + \nabla \cdot \mathbf{F}_j \right) d\mathbf{x} \\ & + g \int_{\mathbf{T}_{\ell(j),j}} \psi_k^{(j)} H_j \nabla \eta_{\ell(j)} d\mathbf{x} + g \int_{\mathbf{T}_{r(j),j}} \psi_k^{(j)} H_j \nabla \eta_{r(j)} d\mathbf{x} \\ & + g \int_{\Gamma_j} \psi_k^{(j)} H_j (\eta_{r(j)} - \eta_{\ell(j)}) \vec{n}_j ds = 0, \end{aligned} \quad (2.13)$$

where $\vec{n}_{ij} = \vec{n}_i|_{\Gamma_j}$. Using the ansatz (2.6), (2.7) and (2.8) we can split the

above equations in terms of spatial and temporal variables, namely

$$\begin{aligned}
 & \int_{\mathbf{T}_i} \phi_k^{(i)} \phi_l^{(i)} d\mathbf{x} \frac{\partial}{\partial t} \hat{\eta}_{l,i} \\
 & + \sum_{j \in S_i} \left(\int_{\Gamma_j} \phi_k^{(i)} \psi_l^{(j)} \vec{n}_{ij} ds \vec{V}_{l,j} - \int_{\mathbf{T}^{i,j}} \nabla \phi_k^{(i)} \psi_l^{(j)} d\mathbf{x} \vec{V}_{l,j} \right) = 0, \quad (2.14)
 \end{aligned}$$

and, equally

$$\begin{aligned}
 & \int_{\mathbf{R}_j} \psi_k^{(j)} \psi_l^{(j)} d\mathbf{x} L_h \left(\vec{V}_{l,j} \right) + g \int_{\mathbf{T}^{\ell(j),j}} \psi_k^{(j)} \psi_m^{(j)} \nabla \phi_l^{(\ell(j))} d\mathbf{x} \hat{H}_{m,j} \hat{\eta}_{l,\ell(j)} \\
 & + g \int_{\mathbf{T}^{r(j),j}} \psi_k^{(j)} \psi_m^{(j)} \nabla \phi_l^{(r(j))} d\mathbf{x} \hat{H}_{m,j} \hat{\eta}_{l,r(j)} \\
 & + g \int_{\Gamma_j} \psi_k^{(j)} \psi_m^{(j)} \phi_l^{(r(j))} \vec{n}_j ds \hat{H}_{m,j} \hat{\eta}_{l,r(j)} \\
 & - g \int_{\Gamma_j} \psi_k^{(j)} \psi_m^{(j)} \phi_l^{(\ell(j))} \vec{n}_j ds \hat{H}_{m,j} \hat{\eta}_{l,\ell(j)} = 0, \quad (2.15)
 \end{aligned}$$

where we have used the standard Einstein summation convention for the repeated indices l and m . L_h is an appropriate discretization of the operator L and will be given later. For every i and j , Eqs. (2.14)-(2.15) are conveniently written in a compact matrix form as

$$\mathbf{M}_i^\phi \frac{\partial}{\partial t} \hat{\eta}_i + \sum_{j \in S_i} \mathcal{D}_{i,j} \hat{\mathbf{V}}_j = 0, \quad (2.16)$$

$$\mathbf{M}_j^\psi L_h \left(\hat{\mathbf{V}}_j \right) + g \left(\mathcal{R}_j \hat{\mathbf{H}}_j \hat{\eta}_{r(j)} - \mathcal{L}_j \hat{\mathbf{H}}_j \hat{\eta}_{\ell(j)} \right) = 0, \quad (2.17)$$

where:

$$\mathbf{M}_i^\phi = \int_{\mathbf{T}_i} \phi_k^{(i)} \phi_l^{(i)} d\mathbf{x}, \quad (2.18)$$

$$\mathbf{M}_j^\psi = \int_{\mathbf{R}_j} \psi_k^{(j)} \psi_l^{(j)} d\mathbf{x}, \quad (2.19)$$

$$\mathcal{D}_{i,j} = \int_{\Gamma_j} \phi_k^{(i)} \psi_l^{(j)} \vec{n}_{ij} ds - \int_{\mathbf{T}_{i,j}} \nabla \phi_k^{(i)} \psi_l^{(j)} d\mathbf{x}, \quad (2.20)$$

$$\mathcal{R}_j = \int_{\mathbf{T}_{r(j),j}} \psi_k^{(j)} \psi_m^{(j)} \nabla \phi_l^{(r(j))} d\mathbf{x} + \int_{\Gamma_j} \psi_k^{(j)} \psi_m^{(j)} \phi_l^{(r(j))} \vec{n}_j ds, \quad (2.21)$$

$$\mathcal{L}_j = \int_{\Gamma_j} \psi_k^{(j)} \psi_m^{(j)} \phi_l^{(\ell(j))} \vec{n}_j ds - \int_{\mathbf{T}_{\ell(j),j}} \psi_k^{(j)} \psi_m^{(j)} \nabla \phi_l^{(\ell(j))} d\mathbf{x}. \quad (2.22)$$

The notation for the vector, matrix and tensor multiplications used here becomes clear from (2.14)-(2.15) and follows the one introduced in [70].

Note that the tensors \mathcal{L} and \mathcal{R} represent the left and the right contribution to the velocity field, respectively. The similar form of these two tensors suggests to introduce a new operator that generalizes \mathcal{L} and \mathcal{R} and, at the same time, also facilitates the notation. Let $i \in [1, N_e]$ and $j \in S_i$, the new tensor $\mathcal{Q}_{i,j}$ is defined as

$$\mathcal{Q}_{i,j} = \int_{\mathbf{T}_{i,j}} \psi_k^{(j)} \psi_m^{(j)} \nabla \phi_l^{(i)} d\mathbf{x} - \int_{\Gamma_j} \psi_k^{(j)} \psi_m^{(j)} \phi_l^{(i)} \sigma_{i,j} \vec{n}_j ds, \quad (2.23)$$

where $\sigma_{i,j}$ is a sign function defined by

$$\sigma_{i,j} = \frac{r(j) - 2i + \ell(j)}{r(j) - \ell(j)}. \quad (2.24)$$

In this way $\mathcal{Q}_{\ell(j),j} = -\mathcal{L}_j$ and $\mathcal{Q}_{r(j),j} = \mathcal{R}_j$, and then Eq. (2.17) becomes in terms of \mathcal{Q}

$$\mathbf{M}_j^\psi L_h \left(\hat{\mathbf{V}}_j \right) + g \left(\mathcal{Q}_{r(j),j} \hat{\mathbf{H}}_j \hat{\boldsymbol{\eta}}_{r(j)} + \mathcal{Q}_{\ell(j),j} \hat{\mathbf{H}}_j \hat{\boldsymbol{\eta}}_{\ell(j)} \right) = 0. \quad (2.25)$$

Furthermore, Eq. (2.25) can also be equivalently rewritten for every i and $j \in S_i$, by using the contribution from element i and the neighbor element through the edge j as follows:

$$\mathbf{M}_j^\psi L_h \left(\hat{\mathbf{V}}_j \right) + g \left(\mathcal{Q}_{i,j} \hat{\mathbf{H}}_j \hat{\boldsymbol{\eta}}_i + \mathcal{Q}_{\varphi(i,j),j} \hat{\mathbf{H}}_j \hat{\boldsymbol{\eta}}_{\varphi(i,j)} \right) = 0. \quad (2.26)$$

We discretize the velocity field in Eq. (2.16) semi-implicitly by using the theta method. Also the free surface elevation in equation (2.17) is discretized semi-implicitly. Finally, in order to avoid the nonlinearity, we discretize the total height explicitly. The summary of the numerical discretization is shown in Eq. (2.27) below:

$$\begin{cases} M_i^\phi \frac{\partial}{\partial t} \hat{\eta}_i + \sum_{j \in S_i} \mathcal{D}_{i,j} \hat{\mathbf{V}}_j^{n+\theta} = 0, \\ M_j^\psi L_h(\hat{\mathbf{V}}_j) + g \left(\mathcal{Q}_{r(j),j} \hat{\mathbf{H}}_j^n \hat{\eta}_{r(j)}^{n+\theta} + \mathcal{Q}_{\ell(j),j} \hat{\mathbf{H}}_j^n \hat{\eta}_{\ell(j)}^{n+\theta} \right) = 0, \end{cases} \quad (2.27)$$

where $\hat{\eta}^{n+\theta} = \theta \hat{\eta}^{n+1} + (1-\theta) \hat{\eta}^n$; $\hat{\mathbf{V}}^{n+\theta} = \theta \hat{\mathbf{V}}^{n+1} + (1-\theta) \hat{\mathbf{V}}^n$; and θ is an implicitness factor to be taken in the range $\theta \in [\frac{1}{2}, 1]$, see e.g. [36]. Discretizing Eqs. (2.27) as described above and using the notation of Eq. (2.26), we get for every i and $j \in S_i$

$$M_i^\phi \frac{\hat{\eta}_i^{n+1} - \hat{\eta}_i^n}{\Delta t} + \theta \sum_{j \in S_i} \mathcal{D}_{i,j} \hat{\mathbf{V}}_j^{n+1} + (1-\theta) \sum_{j \in S_i} \mathcal{D}_{i,j} \hat{\mathbf{V}}_j^n = 0, \quad (2.28)$$

$$\begin{aligned} M_j^\psi \frac{\hat{\mathbf{V}}_j^{n+1} - \widehat{\mathbf{F}}_j^n}{\Delta t} + g\theta \left(\mathcal{Q}_{i,j} \hat{\mathbf{H}}_j^n \hat{\eta}_i^{n+1} + \mathcal{Q}_{\varphi(i,j),j} \hat{\mathbf{H}}_j^n \hat{\eta}_{\varphi(i,j)}^{n+1} \right) \\ + g(1-\theta) \left(\mathcal{Q}_{i,j} \hat{\mathbf{H}}_j^n \hat{\eta}_i^n + \mathcal{Q}_{\varphi(i,j),j} \hat{\mathbf{H}}_j^n \hat{\eta}_{\varphi(i,j)}^n \right) = 0, \end{aligned} \quad (2.29)$$

where $\widehat{\mathbf{F}}_j^n$ is an appropriate explicit discretization of the nonlinear convective terms. In this thesis we use a classical explicit RK DG method based on the Rusanov flux and a third order TVD Runge-Kutta scheme in time to compute the $\widehat{\mathbf{F}}_j^n$. The details will be presented later in Section 2.2.4. Formal substitution of the momentum equation (2.29) into the free surface equation (2.28), see also [32, 37, 70], yields

$$\begin{aligned} \left(M_i^\phi - g\theta^2 \Delta t^2 \sum_{j \in S_i} \mathcal{D}_{i,j} \left(M_j^\psi \right)^{-1} \mathcal{Q}_{i,j} \hat{\mathbf{H}}_j^n \right) \hat{\eta}_i^{n+1} \\ - g\theta^2 \Delta t^2 \sum_{j \in S_i} \mathcal{D}_{i,j} \left(M_j^\psi \right)^{-1} \mathcal{Q}_{\varphi(i,j),j} \hat{\mathbf{H}}_j^n \hat{\eta}_{\varphi(i,j)}^{n+1} = \mathbf{b}_i^n, \end{aligned} \quad (2.30)$$

where

$$\begin{aligned}
 \mathbf{b}_i^n &= \mathbf{M}_i^\phi \hat{\boldsymbol{\eta}}_i^n - \theta \Delta t \sum_{j \in S_i} \mathcal{D}_{i,j} \widehat{\mathbf{F}} \widehat{\mathbf{v}}_j^n - (1 - \theta) \Delta t \sum_{j \in S_i} \mathcal{D}_{i,j} \hat{\mathbf{V}}_j^n \\
 &\quad + g \theta (1 - \theta) \Delta t^2 \sum_{j \in S_i} \mathcal{D}_{i,j} \left(\mathbf{M}_j^\psi \right)^{-1} \left(\boldsymbol{\mathcal{Q}}_{i,j} \hat{\mathbf{H}}_j^n \hat{\boldsymbol{\eta}}_i^n + \boldsymbol{\mathcal{Q}}_{\varphi(i,j),j} \hat{\mathbf{H}}_j^n \hat{\boldsymbol{\eta}}_{\varphi(i,j)}^n \right)
 \end{aligned} \tag{2.31}$$

groups all the known terms at time t^n . One can recognize in Eq. (2.30) a block four-point system for the new free surface elevation $\hat{\boldsymbol{\eta}}_i^{n+1}$ that can be efficiently solved by using the GMRES algorithm [138]. Once the new free surface elevation has been computed, the new velocity field can be readily updated from Eq. (2.29) and the new free surface elevation as:

$$\begin{aligned}
 \hat{\mathbf{V}}_j^{n+1} &= \widehat{\mathbf{F}} \widehat{\mathbf{v}}_j^n - g \theta \Delta t \left(\mathbf{M}_j^\psi \right)^{-1} \left(\boldsymbol{\mathcal{Q}}_{i,j} \hat{\mathbf{H}}_j^n \hat{\boldsymbol{\eta}}_i^{n+1} + \boldsymbol{\mathcal{Q}}_{\varphi(i,j),j} \hat{\mathbf{H}}_j^n \hat{\boldsymbol{\eta}}_{\varphi(i,j)}^{n+1} \right) \\
 &\quad - g (1 - \theta) \Delta t \left(\mathbf{M}_j^\psi \right)^{-1} \left(\boldsymbol{\mathcal{Q}}_{i,j} \hat{\mathbf{H}}_j^n \hat{\boldsymbol{\eta}}_i^n + \boldsymbol{\mathcal{Q}}_{\varphi(i,j),j} \hat{\mathbf{H}}_j^n \hat{\boldsymbol{\eta}}_{\varphi(i,j)}^n \right).
 \end{aligned} \tag{2.32}$$

As implied by Eq. (2.30), the stencil of the present scheme only involves the i -th element and its direct neighbors. Thus, since $\#S_i = 3$, the system described by (2.30) is a block four-point one. The total height $\hat{\mathbf{H}}_j^n$ can be computed as the projection of the free surface elevation on the dual grid $\{\mathbf{R}_j\}_j$ and then adding the projection of the bottom profile:

$$\hat{\mathbf{H}}_j^n = \left(\mathbf{M}_j^\psi \right)^{-1} \left(\mathcal{P}_{\ell(j),j} \hat{\boldsymbol{\eta}}_{\ell(j)}^n + \mathcal{P}_{r(j),j} \hat{\boldsymbol{\eta}}_{r(j)}^n + \int_{\mathbf{R}_j} \psi_k b(x, y) dx \right), \tag{2.33}$$

where \mathcal{P} is an $N_\psi \times N_\phi$ matrix defined by

$$\mathcal{P}_{i,j} = \int_{\mathbf{T}_{i,j}} \psi_k^{(j)} \phi_l^{(i)} dx. \tag{2.34}$$

Note that the free surface is defined on the primary cells \mathbf{T}_i , hence for $p = 0$ the computation of $\hat{\mathbf{H}}_j^n$ on the dual edge-based grid, according to Eq. (2.33) corresponds to the use of a path-conservative method, where the non-conservative product is defined using a straight-line segment path, see [27, 124, 29, 119,

28, 121, 29, 30, 69, 73] for details on high order accurate path-conservative finite volume schemes on *collocated grids*. A centered path-conservative finite volume scheme on *staggered grids* can be found in [30]. An additional numerical flux can be added in Eq. (2.30) in order to introduce some numerical diffusion. In this case the momentum vector \vec{V}_j in Eq. (2.12) has to be replaced by $\vec{V}_j = \bar{V}_j - \frac{1}{2} \lambda_{\max,j} (\eta_{\phi(i,j)} - \eta_i)$ for every $i, j \in S_i$, where $\lambda_{\max,j} = \max(\sqrt{gH} + |\vec{v}|)$ is the maximum signal speed on Γ_j . This corresponds to the use of a Rusanov-type flux in the mass conservation equation.

An important remark is that all the matrices and tensors used above can be precomputed once and for all for every mesh and polynomial degree p . Note that the method is locally and globally mass-conservative for every p since, by taking a constant test function, Eq. (2.14) reduces to a flux form.

2.2.4 Nonlinear convection

In problems where the convective term can be neglected we can take $\widehat{\mathbf{F}}\vec{v}_j^n = \hat{\mathbf{V}}_j^n$ in Eq. (2.31). Otherwise, an explicit cell-centered RKDG method on the dual mesh is used in this chapter for the discretization of the nonlinear convective terms.

The semi-discrete DG scheme for the convective terms on the dual mesh reads

$$\frac{d}{dt} \hat{\mathbf{V}}_j = - \left(\mathbf{M}_j^{\psi} \right)^{-1} \left(\int_{\partial \mathbf{R}_j} \psi_k \mathbf{F}_h \cdot \vec{n} ds - \int_{\mathbf{R}_j} \nabla \psi_k \cdot \mathbf{F} dx \right), \quad (2.35)$$

and the Rusanov flux (see e.g. [137] and [147]) is given by

$$\mathbf{F}_h \cdot \vec{n} = \frac{1}{2} (\mathbf{F}^+ + \mathbf{F}^-) \cdot \vec{n} - \frac{1}{2} s_{\max} (\vec{V}^+ - \vec{V}^-). \quad (2.36)$$

Here, the maximum signal speed $s_{\max} = 2 \max(|\vec{v}^- \cdot \vec{n}|, |\vec{v}^+ \cdot \vec{n}|)$ is given by the maximum eigenvalue of the Jacobian matrix of the purely convective transport operator \mathbf{F} in normal direction, see [70]. Furthermore the \mathbf{F}^+ and \mathbf{F}^- as well as the \vec{V}^+ and \vec{V}^- denote the momentum fluxes and the momentum vectors extrapolated to the boundary of \mathbf{R}_j from within the element \mathbf{R}_j and from the neighbor element, respectively. A classical TVD Runge-Kutta method is used for time integration, see e.g. [57]. The use of a high order Runge-Kutta time discretization scheme is necessary, since the explicit first order Euler method used in combination with a better than first order DG scheme in space would

lead to a linearly unstable scheme. The above method requires that the time step size is restricted by a CFL-type restriction for DG schemes. This condition could be rather severe where both fine grids and high order polynomials are used. However, it is based on the local convective speed $|\vec{v}|$ rather than on the wave speed \sqrt{gH} . Furthermore, the time step of the global semi-implicit scheme is *not* affected by the local time step used for the time integration of the convective terms if a local time stepping / subcycling approach is employed, see [42, 145].

2.2.5 Integration

To close the problem it remains to compute the matrices and tensors \mathbf{M}^ϕ , \mathbf{M}^ψ , \mathcal{D} and \mathcal{Q} . For every triangulation $\{\mathbf{T}_i\}_i$ and polynomial degree p , we are interested in an explicit formulation for the matrices. In order to do this, it is convenient to introduce some classical transformation operators that map the physical control volumes into the reference control volumes, where the basis functions are defined, and vice versa. In this section, we first illustrate the procedure using a subparametric approach for the sake of simplicity and will later extend it to the generalized isoparametric case in Section 2.2.7.

Let \mathbf{T}_i be the i -th triangle, and let (X_k^i, Y_k^i) denote the k -th vertex of the triangle \mathbf{T}_i , then the subparametric transformation from the reference space (ξ, γ) to the physical space (x, y) is the linear transformation $T_i^{-1} : T_{std} \rightarrow \mathbf{T}_i$ defined by the action

$$(\xi, \gamma) \xrightarrow{T_i^{-1}} \begin{cases} x = \sum_{k=1}^3 \mathcal{N}_k^T(\xi, \gamma) X_k^i = X_1^i + \xi(X_2^i - X_1^i) + \gamma(X_3^i - X_1^i), \\ y = \sum_{k=1}^3 \mathcal{N}_k^T(\xi, \gamma) Y_k^i = Y_1^i + \xi(Y_2^i - Y_1^i) + \gamma(Y_3^i - Y_1^i), \end{cases} \quad (2.37)$$

where the functions $\mathcal{N}_1^T(\xi, \gamma) = 1 - \xi - \gamma$, $\mathcal{N}_2^T(\xi, \gamma) = \xi$ and $\mathcal{N}_3^T(\xi, \gamma) = \gamma$ are the linear shape functions associated with the vertices of the standard reference triangle. One can easily see that

$$\begin{aligned} J(T_i^{-1}) &= \begin{pmatrix} X_2^i - X_1^i & X_3^i - X_1^i \\ Y_2^i - Y_1^i & Y_3^i - Y_1^i \end{pmatrix}, \\ |J(T_i^{-1})| &= (X_2^i - X_1^i)(Y_3^i - Y_1^i) + (X_3^i - X_1^i)(Y_2^i - Y_1^i), \end{aligned} \quad (2.38)$$

where J and $|J|$ are the Jacobian matrix of the mapping and its determinant, respectively. In this simple case it is possible to construct by hand also the

inverse transformation $T_i : \mathbf{T}_i \rightarrow T_{std}$ as

$$T_i : (x, y) \rightarrow \left(\begin{array}{l} \xi = \frac{x(Y_3^i - Y_1^i) - y(X_3^i - X_1^i) + Y_1^i X_3^i - X_1^i Y_3^i}{|J(T_i^{-1})|} \\ \gamma = \frac{y(X_2^i - X_1^i) - x(Y_2^i - Y_1^i) + X_1^i Y_2^i - Y_1^i X_2^i}{|J(T_i^{-1})|} \end{array} \right). \quad (2.39)$$

In a similar way, we construct the subparametric transformation on a quadrilateral element \mathbf{R}_j . The transformation $T_j^{-1} : R_{std} \rightarrow \mathbf{R}_j$ is defined by

$$\begin{aligned} x &= \sum_{k=1}^4 \mathcal{N}_k^R(\xi, \gamma) X_k^j = X_1^j + (X_2^j - X_1^j)\xi + (X_4^j - X_1^j)\gamma \\ &\quad + (X_1^j - X_2^j + X_3^j - X_4^j)\xi\gamma, \\ y &= \sum_{k=1}^4 \mathcal{N}_k^R(\xi, \gamma) Y_k^j = Y_1^j + (Y_2^j - Y_1^j)\xi + (Y_4^j - Y_1^j)\gamma \\ &\quad + (Y_1^j - Y_2^j + Y_3^j - Y_4^j)\xi\gamma, \end{aligned} \quad (2.40)$$

with the bilinear shape functions $\mathcal{N}_1^R(\xi, \gamma) = (1-\xi)(1-\gamma)$, $\mathcal{N}_2^R(\xi, \gamma) = \xi(1-\gamma)$, $\mathcal{N}_3^R(\xi, \gamma) = \xi\gamma$ and $\mathcal{N}_4^R(\xi, \gamma) = (1-\xi)\gamma$ associated with the unit square. For the direct transformation from the physical to the reference space $T_j : \mathbf{R}_j \rightarrow R_{std}$ we use the Newton algorithm.

We now need to compute the matrices defined in (2.18), (2.19), (2.20) and (2.23). For this purpose we change variables in the integrals using the maps defined above. The mass matrices simply become for every i and j ,

$$\mathbf{M}_i^\phi(k, l) = \int_{T_{std}} \phi_k(\xi, \gamma) \phi_l(\xi, \gamma) |J(T_i^{-1})| d\xi d\gamma, \quad (2.41)$$

and

$$\mathbf{M}_j^\psi(k, l) = \int_{R_{std}} \psi_k(\xi, \gamma) \psi_l(\xi, \gamma) |J(T_j^{-1})| d\xi d\gamma. \quad (2.42)$$

Integrals (2.41) and (2.42) can be efficiently computed using an appropriate multidimensional Gaussian quadrature rule, see [142]. For the other matrices, where the two maps on triangular and quadrilateral elements interact in the same integral, the transformation has to be specified explicitly. In particular, we are interested in computing $\mathcal{D}_{i,j}$. For this purpose we split $\mathcal{D}_{i,j}$ into two

parts, the volume and the edge contribution, such as in Eq. (2.20). For the volume contribution, we have the following possibility to map the triangle $\mathbf{T}_{i,j}$:

$$\begin{array}{ccc}
 \mathbf{T}_{i,j} & \xrightarrow{T_j} & R_{std}|_{T_j}(\mathbf{T}_{i,j}) \\
 \downarrow T_i & \cdot \cdot & \downarrow \\
 T_{std}|_{T_i}(\mathbf{T}_{i,j}) & \longrightarrow & T_{std}
 \end{array} \tag{2.43}$$

The correct way to compute the integral is taking the direct transformation $\mathbf{T}_{i,j} \xrightarrow{T_{ij}} T_{std}$. The quadrature rule is taken on T_{std} and then the value of the functions are computed in the right space using the maps $T_j \cdot T_{ij}^{-1}$ for the functions generated by ψ and $T_i \cdot T_{ij}^{-1}$ for the functions generated by ϕ , respectively (see Figure 2.3). The line integral in the matrix $\mathcal{D}_{i,j}$ is computed using the direct transformation $\Gamma_j \xrightarrow{T_{in}} [0, 1]$. Since $\mathcal{Q}_{i,j}$ is defined on the dual grid, we have to pass through the map T_j and then use a quadrature rule on the triangular element $T_j(\mathbf{T}_{i,j})$, the correct way to compute the previous integral is taking the transformation $\mathbf{T}_{i,j} \xrightarrow{T_j} R_{std}|_{T_j}(\mathbf{T}_{i,j}) \xrightarrow{T_{tri}} T_{std}$. In particular the quadrature rule is taken on T_{std} and then the value of the functions can be evaluated in the right space (see Figure 2.4). Also in this case the line integral is computed directly using the transformation T_{in} .

2.2.6 Boundary conditions

Let us now consider the case in which we introduce some boundary conditions, such as a prescribed velocity at the boundary. In alternative, one can also prescribe the pressure (i.e. the free surface elevation) on the boundary, for which the treatment is similar. In this chapter, we do not consider the case of supercritical flow at the domain boundaries.

Observe how, for $i \in [1, N_e]$ and $j \in S_i \cap \mathcal{B}(\Omega)$, the boundary element $\mathbf{R}_j = \mathbf{T}_{i,j}$ is a *triangular element* and *not* a quadrilateral element. The basis functions to be used are the one generated on T_{std} . In this way the matrices \mathbf{M}_j^ψ , $\mathcal{D}_{i,j}$, $\mathcal{Q}_{i,j}$, $\mathcal{P}_{i,j}$ defined in (2.19), (2.20), (2.23) and (2.34), have to be modified for boundary elements.

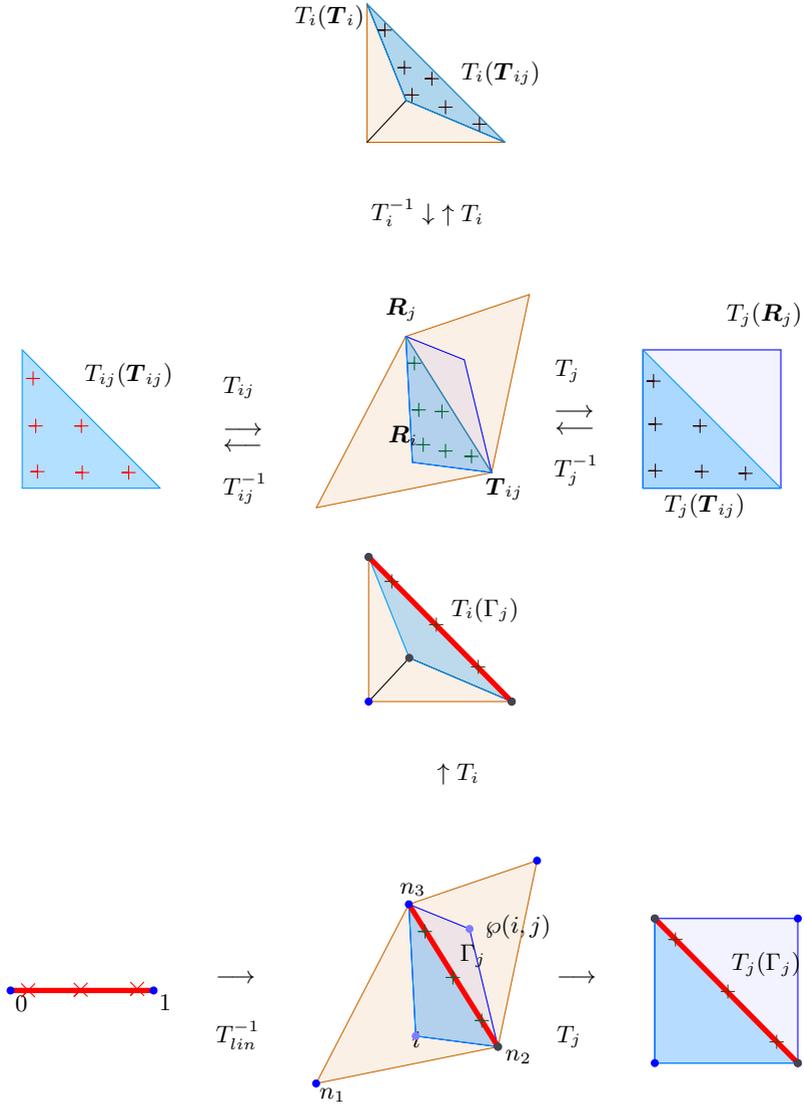


Figure 2.3: Volume (top) and line (bottom) integration paths for $\mathcal{D}_{i,j}$.

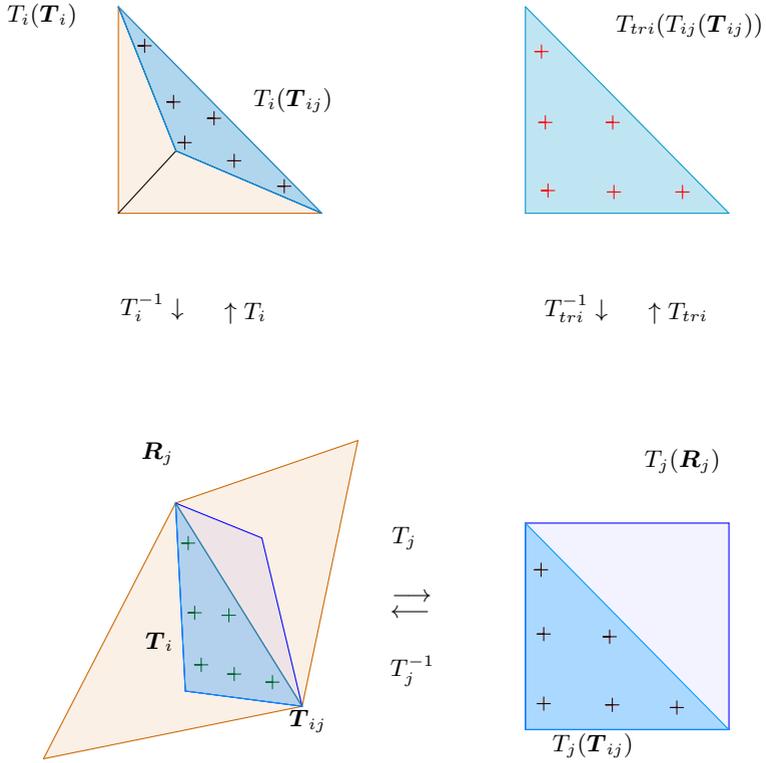


Figure 2.4: Integration for $\mathcal{Q}_{i,j}$, the quadrature points (red) are transformed using the subparametric maps T .

For every $j \in S_i \cap \mathcal{B}(\Omega)$

$$\begin{aligned} H_j &= \sum_{l=1}^{N_\phi} \phi_l \hat{H}_{l,j}, \\ \vec{V}_j &= \sum_{l=1}^{N_\phi} \phi_l \hat{V}_{l,j}, \end{aligned} \quad (2.44)$$

where the ϕ_l are the basis functions generated from the reference triangle T_{std} . The matrices can be recomputed for $j \in S_i \cap \mathcal{B}(\Omega)$ and will be called $\mathcal{D}_{i,j}^\partial$, $\mathcal{Q}_{i,j}^\partial$, $\mathcal{P}_{i,j}^\partial$.

We are interested in solving a problem with a given momentum flux through the boundary of Ω , i.e. $\vec{V}|_{\partial\Omega} = \vec{V}^b$. We rewrite the velocity contribution in Eq. (2.16), splitting the volume and the edge contributions as:

$$\begin{aligned} \mathcal{D}_{i,j}^\partial \hat{\vec{V}}_j &= \mathcal{D}_{i,j}^{\partial,L} \hat{\vec{V}}_j - \mathcal{D}_{i,j}^{\partial,V} \hat{\vec{V}}_j \\ &= \left\{ \int_{\Gamma_j} \phi_k^{\partial,(i)} \vec{V}^b ds \right\}_k - \mathcal{D}_{i,j}^{\partial,V} \hat{\vec{V}}_j \\ &:= (P\vec{v})_j^b - \mathcal{D}_{i,j}^{\partial,V} \hat{\vec{V}}_j, \end{aligned} \quad (2.45)$$

where $(P\vec{v})_j^b$ is a known quantity defined from the above equation; $\mathcal{D}_{i,j}^{\partial,L}$ and $\mathcal{D}_{i,j}^{\partial,V}$ are the volume and the edge contribution, respectively. In addition, if we do not introduce any jump contribution from the free surface elevation on Γ_j , $j \in \mathcal{B}(\Omega)$, then the edge contribution of $\mathcal{Q}_{i,j}^\partial$ can be neglected and so $\mathcal{Q}_{i,j}^\partial = \mathcal{Q}_{i,j}^{\partial,V}$. The resulting semi-implicit numerical scheme (2.32)-(2.30) is

$$\hat{\vec{V}}_j^{n+1} = \widehat{\mathbf{F}}\vec{v}_j^n - g\Delta t \left(M_j^\psi \right)^{-1} \mathcal{Q}_{i,j}^{\partial,V} \hat{H}_j^n \hat{\eta}_i^{n+\theta}, \quad j \in \mathcal{B}(\Omega), \quad (2.46)$$

and

$$\begin{aligned}
 & \left[M_i^\phi - g\theta^2 \Delta t^2 \sum_{j \in S_i - \mathcal{B}(\Omega)} \mathcal{D}_{i,j} \left(M_j^\psi \right)^{-1} \mathcal{Q}_{i,j} \hat{H}_j^n \right. \\
 & \quad \left. + g\theta^2 \Delta t^2 \sum_{j \in S_i \cap \mathcal{B}(\Omega)} \mathcal{D}_{i,j}^{\partial,V} \left(M_j^\psi \right)^{-1} \mathcal{Q}_{i,j}^{\partial,V} \hat{H}_j^n \right] \hat{\eta}_i^{n+1} \\
 & - g\theta^2 \Delta t^2 \sum_{j \in S_i - \mathcal{B}(\Omega)} \mathcal{D}_{i,j} \left(M_j^\psi \right)^{-1} \mathcal{Q}_{\varphi(i,j)} \hat{H}_j^n \hat{\eta}_{\varphi(i,j)}^{n+1} = \tilde{\mathbf{b}}_i^n, \quad (2.47)
 \end{aligned}$$

where now the vector of known terms is

$$\begin{aligned}
 \tilde{\mathbf{b}}_i^n &= M_i^\phi \hat{\eta}_i^n - \theta \Delta t \sum_{j \in S_i - \mathcal{B}(\Omega)} \mathcal{D}_{i,j} \widehat{\mathbf{F}} \tilde{\mathbf{v}}_j^n - (1 - \theta) \Delta t \sum_{j \in S_i - \mathcal{B}(\Omega)} \mathcal{D}_{i,j} \tilde{\mathbf{V}}_j^n \\
 & \quad + (1 - \theta) \Delta t \sum_{j \in S_i \cap \mathcal{B}(\Omega)} \mathcal{D}_{i,j}^{\partial,V} \tilde{\mathbf{V}}_j^n + \theta \Delta t \sum_{j \in S_i \cap \mathcal{B}(\Omega)} \mathcal{D}_{i,j}^{\partial,V} \widehat{\mathbf{F}} \tilde{\mathbf{v}}_j^n \\
 & - \Delta t \sum_{j \in S_i \cap \mathcal{B}(\Omega)} (P\tilde{\mathbf{v}}_j)^b + g\theta(1 - \theta) \Delta t^2 \sum_{j \in S_i \cap \mathcal{B}(\Omega)} \mathcal{D}_{i,j}^{\partial,V} \left(M_j^\psi \right)^{-1} \mathcal{Q}_{i,j}^{\partial,V} \hat{H}_j^n \hat{\eta}_i^n \\
 & + g\theta(1 - \theta) \Delta t^2 \sum_{j \in S_i - \mathcal{B}(\Omega)} \mathcal{D}_{i,j} \left(M_j^\psi \right)^{-1} \left(\mathcal{Q}_{i,j} \hat{H}_j^n \hat{\eta}_i^n + \mathcal{Q}_{\varphi(i,j),j} \hat{H}_j^n \hat{\eta}_{\varphi(i,j)}^n \right). \quad (2.48)
 \end{aligned}$$

The above numerical scheme allows to solve initial-boundary value problems when the mass flux is prescribed on $\partial\Omega$.

Finally, since the presented method is a discontinuous Galerkin scheme, hence boundary conditions are only satisfied in a weak manner, it may happen that if we specify the velocity vector field on the boundary this could not affect the internal part of the boundary elements as desired. In order to overcome this problem we can adjust the momentum vector on the boundary as $\tilde{\tilde{\mathbf{V}}} = \tilde{\mathbf{V}}^b + (\tilde{\mathbf{V}}^b - \tilde{\mathbf{V}}^-)$, where $\tilde{\mathbf{V}}^-$ is the boundary extrapolated momentum from within the computational domain. In this way we add a penalty term that forces also the interior of the element towards the prescribed boundary value.

2.2.7 Curved Elements

The scheme presented in Section 2.2.3 extends directly to a more general context by the use of a complete isoparametric approach. For this purpose, the

shape functions \mathcal{N}_k^T and \mathcal{N}_k^R in equations (2.37) and (2.40) are simply replaced by the usual nodal basis functions ϕ_k and ψ_k on the reference triangle and on the reference square, respectively. The advantage of this approach in the definition of the maps given in Section 2.2.5 is that the elements can be curved and then the computational domain can be adapted to the physical one (see Fig. 2.5). In the isoparametric approach, every element \mathbf{T}_i is characterized by

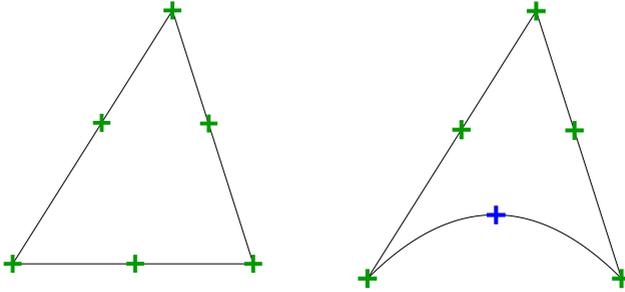


Figure 2.5: Subparametric approach (left) and Isoparametric approach (right) for $p = 2$

N_ϕ nodes and similarly each \mathbf{R}_j is characterized by N_ψ points instead of 3 and 4, respectively. Formally, the only change is the definition of the maps, which then become

$$x = \sum_k^{N_\phi} \phi_k X_k^i, \quad y = \sum_k^{N_\phi} \phi_k Y_k^i, \quad (2.49)$$

for triangles and

$$x = \sum_k^{N_\psi} \psi_k X_k^j, \quad y = \sum_k^{N_\psi} \psi_k Y_k^j, \quad (2.50)$$

for quadrilateral elements.

We then also need the Newton method for both direct maps T_i and T_j . Regarding the adaptation of the elements, once the points on the edge of the elements

have been moved according to the physical domain, we have to move in a consistent way also the internal points. We recompute the position of the internal points by solving the Laplace equation for the variations using a classical finite element method for elliptic boundary value problems in two dimensions.

In addition, the normal vectors are no longer constant along the edges, but they depend on the position. For curved elements, we can first find the tangential vectors and then compute the normal vectors that have to be included into the matrices due to the dependence on the position.

Remark that the isoparametric approach affects only the construction of the matrices in the preprocessor stage and hence it does *not* affect the computational time for the simulation at run time.

2.3 Numerical tests

In this section, the accuracy and efficiency of the proposed algorithm is assessed by comparing the numerical results against exact analytical solutions or other reliable reference solutions.

2.3.1 Convergence test

In this test case we consider a smooth steady state problem to measure the order of accuracy of the scheme (2.32) and (2.30). We take a flat bottom ($b \equiv 0$) and a square domain $\Omega = [-8.0, 8.0] \times [-8.0, 8.0]$. As initial condition we set

$$\eta(x, y, 0) = 1 - \frac{1}{2g} e^{-(r^2-1)}, \quad (2.51)$$

and

$$u(x, y, 0) = -u_\alpha \sin(\alpha), \quad v(x, y, 0) = u_\alpha \cos(\alpha), \quad (2.52)$$

where $u_\alpha(r, 0) = r e^{-\frac{1}{2}(r^2-1)}$; $\tan(\alpha) = \frac{y}{x}$; and $r = \sqrt{x^2 + y^2}$. The analytical solution is an exact steady state solution, which is given by the exact balance between the centrifugal and the pressure force, as we can see from the momentum equation in the radial direction (see [20, 70]):

$$\frac{\partial \eta}{\partial r} = \frac{u_\alpha^2}{gr}.$$

The problem is solved for successively refined meshes and increasing polynomial degrees $p = 0 \dots 6$. The L_2 error between the analytical and the numerical

solution is computed as

$$\epsilon(\eta) = \sqrt{\int_{\Omega} (\eta_h - \eta_e)^2 d\mathbf{x}},$$

for the free surface elevation and

$$\epsilon(\vec{v}) = \sqrt{\int_{\Omega} (\vec{v}_h - \vec{v}_e)^2 d\mathbf{x}},$$

for the velocity vector field. Here η_h is the numerical solution and η_e is the exact analytical solution for the free surface given above; \vec{v}_h and \vec{v}_e are the numerical and the exact solution for the velocity vector, respectively.

The time step is computed according to the CFL-type time step restriction of RKDG schemes based on the maximum convective speed $|\vec{v}_{\max}|$ and the smallest incircle diameter h_{\min} in the computational domain as

$$\Delta t = \frac{\text{CFL}}{(2p+1)} \cdot \frac{h_{\min}}{2|\vec{v}_{\max}|}.$$

For the two-dimensional test problems in this chapter we always use $\text{CFL} = 0.45 < 0.5$, according to the stability limits of unsplit Godunov-type schemes in multiple dimensions, as discussed in detail in [149]. The final simulation time is set to $t_{\text{end}} = 0.1$ and for this steady state problem we use $\theta = 1$. The results are summarized in Tab. 2.2, where also the CPU times for the preprocessing part of the scheme and for the simulation are listed separately. The computations are carried out on a single CPU core of an Intel Core 2 i7-3770 CPU with 3.40Ghz clock speed and 22 GB of RAM. We can observe from Tab. 2.2 that the order of accuracy of the method for high order polynomials is of the order of $p + \frac{1}{2}$ for the free surface elevation and p for the velocity. The new high order semi-implicit staggered DG method presented in this chapter is furthermore compared with high order accurate cell-centered path-conservative $P_N P_M$ schemes on unstructured triangular meshes [68, 69, 75] for the smooth vortex problem discussed in this section. The results of the comparison made for third and sixth order schemes can be found in Table 2.3. Note that the error norms of the semi-implicit DG schemes are comparable or even better than the ones obtained with $P_N P_M$ schemes, but the CPU times reported for the semi-implicit staggered DG scheme are higher. This is due to two reasons: first, the semi-implicit DG code has not been optimized concerning

2 Two-dimensional shallow water equations

Table 2.1: Numerical convergence results for $p = 0, 1, 2, 3$. "*Pre.T*" and "*Sim.T*" indicate the wallclock time expressed in seconds that was needed for the preprocessing stage and for the simulation, respectively.

p	N_e	$\epsilon(\eta)$	$\epsilon(\vec{v})$	$\sigma(eta)$	$\sigma(\vec{v})$	Pre.T	Sim.T
0	2048	1.2015E-02	0.1826	-	-	0.0	0.2
0	8192	6.8991E-03	9.2810E-02	0.8	1.0	0.3	1.5
0	32768	3.7964E-03	4.6914E-02	0.9	1.0	1.5	11.8
0	131072	2.0167E-03	2.3654E-02	0.9	1.0	5.9	87.5
0	524288	1.0451E-03	1.1902E-02	0.9	1.0	22.8	719.7
1	1250	3.1775E-03	3.6781E-02	-	-	0.2	0.7
1	5000	8.4026E-04	1.2463E-02	1.9	1.6	1.2	5.4
1	20000	2.2287E-04	4.5747E-03	1.9	1.4	4.5	38.45
1	45000	1.0489E-04	2.5070E-03	1.9	1.5	11.1	148.3
1	80000	6.1295E-05	1.6066E-03	1.9	1.5	17.7	299.8
2	1250	4.9209E-04	5.7017E-03	-	-	2.9	2.6
2	5000	6.3462E-05	1.2992E-03	3.0	2.1	11.4	20.7
2	20000	9.2566E-06	2.7516E-04	2.8	2.2	45.3	155.7
2	45000	2.9821E-06	1.0646E-04	2.8	2.3	107.9	586.1
2	80000	1.3187E-06	5.3427E-05	2.8	2.4	182.6	1290.4
3	1250	6.2368E-05	8.8073E-04	-	-	15.0	8.8
3	5000	5.2317E-06	1.0762E-04	3.6	3.0	60.3	70.2
3	20000	4.3497E-07	1.261E-05	3.6	3.1	243.6	546.2
3	45000	1.0028E-07	3.5606E-06	3.6	3.1	556.3	1841.4
3	80000	3.5759E-08	1.4248E-06	3.6	3.2	924.2	3887.0

speed, and second, the tolerance set for the GMRES algorithm used to solve the linear algebraic system was set to machine precision (10^{-16}), hence requiring a significant number of iterations.

Table 2.2: Numerical convergence results for $p = 4, 5, 6$. "Pre.T" and "Sim.T" indicate the wallclock time expressed in seconds that was needed for the preprocessing stage and for the simulation, respectively.

p	N_e	$\epsilon(\eta)$	$\epsilon(\vec{v})$	$\sigma(\eta)$	$\sigma(\vec{v})$	Pre.T	Sim.T
4	7200	1.3749E-07	3.4113E-06	-	-	379.7	385.4
4	9800	6.7092E-08	1.8283E-06	4.7	4.0	504.3	608.7
4	12800	3.5231E-08	1.0667E-06	4.8	4.0	658.5	895.3
4	16200	2.0008E-08	6.6388E-07	4.8	4.0	841.6	1296.7
4	20000	1.2110E-08	4.3445E-07	4.8	4.0	1032.2	1770.2
5	200	1.6498E-04	1.3008E-03	-	-	39.7	5.8
5	800	3.2247E-06	4.4656E-05	5.7	4.9	165.1	45.5
5	1800	3.4105E-07	6.2691E-06	5.5	4.8	373.2	156.9
5	3200	7.2570E-08	1.5150E-06	5.4	4.9	661.5	374.6
5	5000	2.1314E-08	5.0511E-07	5.5	4.9	1056.5	728.3
6	200	3.7045E-05	3.1150E-04	-	-	132.9	15.9
6	800	4.6704E-07	6.9979E-06	6.3	5.5	549.1	124.2
6	1800	3.5657E-08	6.3733E-07	6.3	5.9	1283.3	433.2
6	3200	5.4729E-09	1.1672E-07	6.5	5.9	2320.7	1003.1
6	5000	1.2306E-09	3.1242E-08	6.7	5.9	3597.6	2149.8

2.3.2 Well-balancedness (C-property)

The spatially high order accurate semi-implicit DG scheme on staggered unstructured triangular meshes presented here is by construction *well-balanced* in the sense of the *C-property* of [116, 94, 92, 93, 16, 86, 23, 5], i.e. the scheme preserves steady state solutions of the form $\eta = \text{const}$ and $\vec{v} = 0$ *exactly*. This becomes already obvious directly from the variational formulation (2.12)-(2.13). Next, the well-balanced property is verified numerically by using a classical test problem proposed by LeVeque in [116], consisting in a small perturbation of a free surface at rest over a smoothly varying bottom topography. The one-dimensional version of this test problem has already been run successfully with high order semi-implicit staggered DG schemes in [70] and is repeated here in two space dimensions on unstructured staggered triangular grids. The two-

Table 2.3: Comparison between third and sixth order staggered unstructured semi-implicit DG schemes and explicit cell-centered unstructured path-conservative $P_N P_M$ schemes for the smooth vortex problem. The third order simulations use a mesh with $N_e = 20000$ and the sixth order simulations were done on the grid with $N_e = 3200$.

Numerical method	L2 error (η)	CPU time [s]
Explicit cell-centered $P_0 P_2$	2.4416E-04	8.13
Explicit cell-centered $P_1 P_2$	4.1328E-05	26.53
Explicit cell-centered $P_2 P_2$	8.6345E-06	45.71
Semi-implicit staggered DG ($p = 2$)	9.2566E-06	155.7
Explicit cell-centered $P_0 P_5$	3.8115E-04	2.62
Explicit cell-centered $P_3 P_5$	1.1034E-06	16.2
Explicit cell-centered $P_5 P_5$	9.6951E-08	30.08
Semi-implicit staggered DG ($p = 5$)	7.2570E-08	374.6

dimensional computational domain used is $\Omega = [-2, 1] \times [-0.5, 0.5]$ and the initial condition for the velocity field is $u = v = 0$, while the initial free surface level η is defined as

$$\eta(x, y, 0) = \begin{cases} 1 + \epsilon, & \text{if } -0.95 \leq x \leq -0.85, \\ 1, & \text{else.} \end{cases}$$

The initial discontinuity is smoothed such as in [70] using $\delta = 0.02$. The bottom topography is given by an elliptical Gaussian bump of the form

$$b(x, y) = 0.8e^{(-5(x+0.1)^2 - 50y^2)}.$$

All simulations are carried out on a very coarse triangular mesh composed of 992 triangles with characteristic mesh size $h = 0.1$. A fixed timestep of $\Delta t = 0.001$ is used together with $\theta = 0.6$; the polynomial degree of the basis and test functions is $p = 4$. Recall that $\theta \geq 0.5$ is the factor of implicitness used in the time discretization scheme, where $\theta = 0.5$ corresponds to the classical Crank-Nicholson method. Four impermeable wall boundary conditions are used, i.e. a zero mass flux is imposed on all boundaries. The primary triangular mesh

Table 2.4: Numerical verification of the exact C -property (well-balanced property) of the unstructured staggered semi-implicit DG scheme ($p = 4$) for different machine precisions at time $t = 0.1$. The L_2 and L_∞ errors refer to the free surface elevation η .

Case	L_∞ error	L_2 error
Single precision	$3.8146 \cdot 10^{-6}$	$1.4187 \cdot 10^{-6}$
Double precision	$3.1086 \cdot 10^{-14}$	$8.3476 \cdot 10^{-15}$
Quadruple precision	$2.4074 \cdot 10^{-32}$	$7.1047 \cdot 10^{-33}$

and the dual quadrilateral grid used for this test problem are shown in Fig. 2.6.

First, the *exact* well-balanced property is verified using $\epsilon = 0$. The results are reported for various machine precisions in Table 2.4, confirming that the scheme is able to maintain the water at rest solution up to machine precision. Next, according to [116], a small perturbation is added by using $\epsilon = 0.01$. The results obtained for the free surface elevation with the high order unstructured staggered DG method of this chapter are depicted in Fig. 2.7 and agree qualitatively with the results obtained in [26]. As expected, the results show no spurious oscillations in the vicinity of the bottom bump, confirming that the present scheme is able to simulate accurately small perturbations of a steady state in the presence of a variable bottom topography.

2.3.3 Smooth surface wave propagation

Here the propagation of an initial smooth surface in the circular domain $\Omega = \{x^2 + y^2 \leq 1\}$ is considered, see [70]. The initial condition is given by

$$\eta(x, y, 0) = 1 + e^{-\frac{1}{2}\left(\frac{r}{\sigma}\right)^2}, \quad u(x, y, 0) = v(x, y, 0) = b(x, y) = 0,$$

where $\sigma = 0.1$ is the halfwidth of the initial pulse. Polynomials of degree $p = 5$ are considered in the circular computational domain Ω using $N_e = 1237$ triangular elements on the primary mesh. For the present test we choose $\theta = 0.5$; $\Delta t = 4 \times 10^{-4}$; and $t_{end} = 0.15$. Due to the radial symmetry of the problem, the reference solution can be obtained by solving the one-dimensional shallow water equations in radial direction with geometric reaction source terms, using

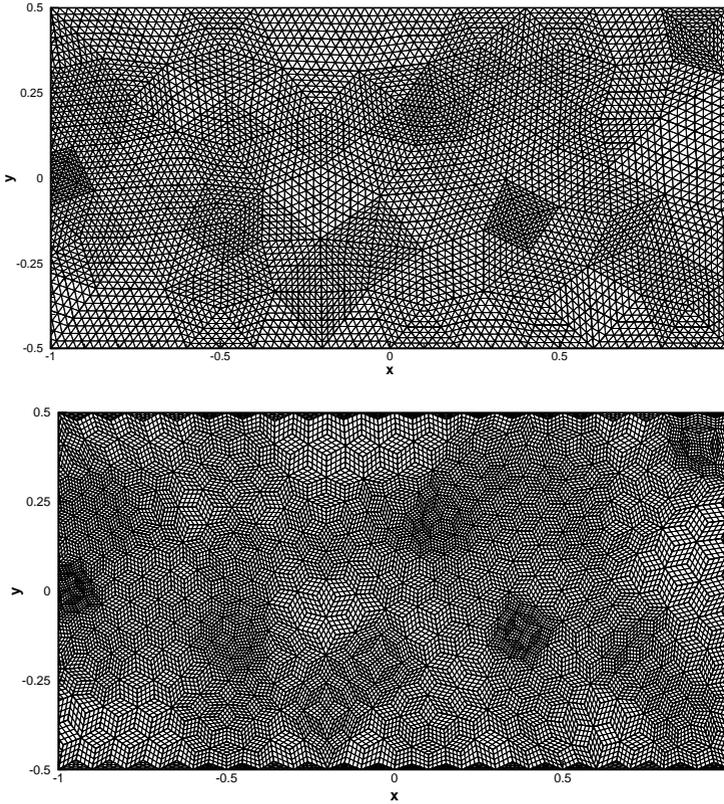


Figure 2.6: The thick solid lines indicate the primary triangular grid (top) and the dual quadrilateral grid (bottom) used for the small surface perturbation test of Leveque. The thin solid lines denote the sub-grid associated with the nodal degrees of freedom of the high order isoparametric DG finite element scheme.

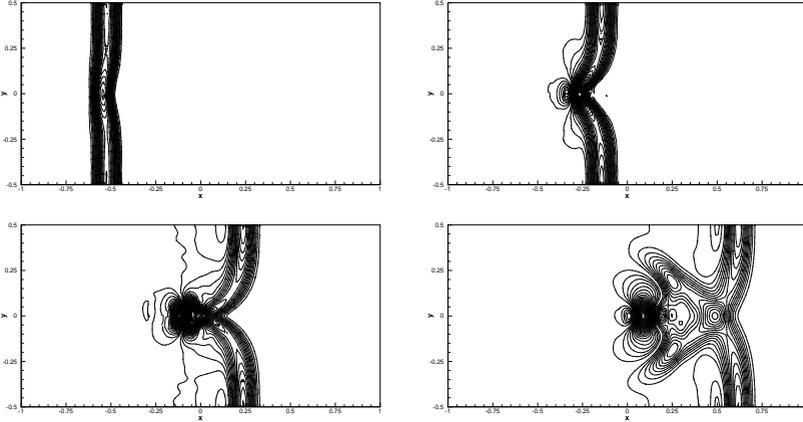


Figure 2.7: Numerical solution for the free surface η at times $t = 0.12$, $t = 0.24$, $t = 0.36$ and $t = 0.48$ for the 2D surface perturbation test problem ($\epsilon = 10^{-2}$) of LeVeque [116]. An unstructured staggered semi-implicit DG scheme with piecewise polynomials of degree $p = 4$ has been used, together with a mesh size of $h = 0.1$ and a time step of $\Delta t = 0.001$. 80 equidistant contour lines in the interval $\eta \in [0.99, 1.01]$ are shown.

a classical shock capturing TVD finite volume scheme with 5000 points (see e.g. [148, 70]). The scheme used for obtaining the 1D radial reference solution was based on the new Osher-type Riemann solver based on a numerical quadrature of the path integral presented in [76, 77].

A three dimensional plot of the free surface elevation at several times is shown in Fig. 2.9. The comparison between numerical results and the reference solution is presented in Fig. 2.8. A very good agreement between the numerical and reference solution is observed until $t = 0.1$. Spurious oscillations are observed at $t = 0.15$ when a discontinuity is generated by the nonlinear hyperbolic governing PDE system, since no limiters have been used in the present DG scheme.

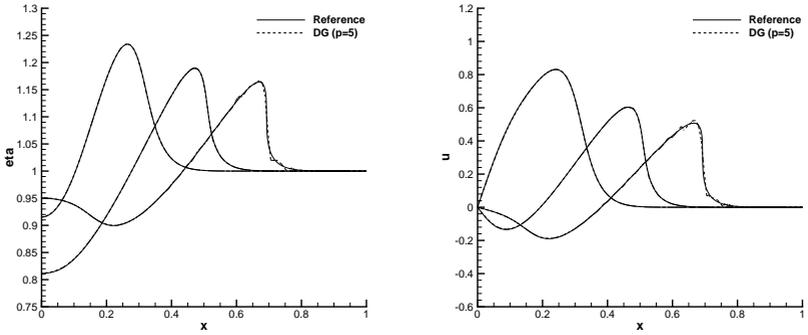


Figure 2.8: Comparison between numerical and reference solution at times $t = 0.05$, $t = 0.1$ and $t = 0.15$ for the free surface elevation (left) and the velocity u (right) in the x direction.

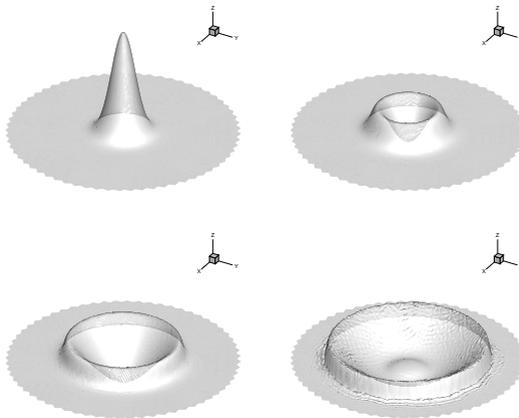


Figure 2.9: Three-dimensional plot of the free surface elevation at times $t = 0$, $t = 0.05$, $t = 0.1$ and $t = 0.15$

2.3.4 Dambreak over a bottom step

In the present test problem, taken from [70], the following initial condition is considered over $\Omega = \{(x, y) \in \mathbb{R}^2 \mid \sqrt{x^2 + y^2} \leq 2\}$:

$$\eta(x, y, 0) = \begin{cases} \eta^\ell & r \leq r_0 \\ \eta^r & r > r_0 \end{cases}$$

$$b(x, y) = \begin{cases} b^\ell & r \leq r_0 \\ b^r & r > r_0 \end{cases}$$

$$u(x, y, 0) = v(x, y, 0) = 0$$

The initial discontinuity is smoothed as follows:

$$\begin{aligned} \eta(r, 0) &= \frac{1}{2}(\eta^\ell + \eta^r) + \frac{1}{2}(\eta^r - \eta^\ell)\operatorname{erf}\left(\frac{r - r_0}{\delta}\right) \\ b(r) &= \frac{1}{2}(b^\ell + b^r) + \frac{1}{2}(b^r - b^\ell)\operatorname{erf}\left(\frac{r - r_0}{\delta}\right) \end{aligned} \quad (2.53)$$

For the present test we fix $\eta^\ell = 1$; $\eta^r = 0.5$; $b^\ell = -0.2$; $b^r = 0$; $r_0 = 1$; and $\delta = 0.01$. As in the previous test case, the reference solution is obtained by using a one-dimensional radial second order TVD scheme with 5000 points ([148, 77, 76]). The simulation is carried out with $p = 3$; $\theta = 1$; $\Delta t = 6 \times 10^{-4}$; and $t_{end} = 0.20$; The computational domain Ω is discretized using a total number of $N_e = 8192$ elements. The numerical results are compared against the reference solution and are presented in Fig. 2.10. A good agreement is shown, both for the free surface η and the velocity component u in x direction. We furthermore compare the solution obtained with our new high order semi-implicit staggered DG method with the numerical results obtained using a fourth order accurate path-conservative ADER-WENO finite volume scheme on unstructured triangular meshes, described in detail in [69, 74, 75]. The finite volume results are also depicted in Fig. 2.10. To make the results comparable, approximately the same number of degrees of freedom has been used in both cases, i.e. the finite volume mesh contains 81916 triangular elements, while the semi-implicit DG scheme with $p = 3$ on 8192 triangles leads to a total number of 81920 degrees of freedom on the primal mesh. The results of the quantitative comparison are summarized in Table 2.5.

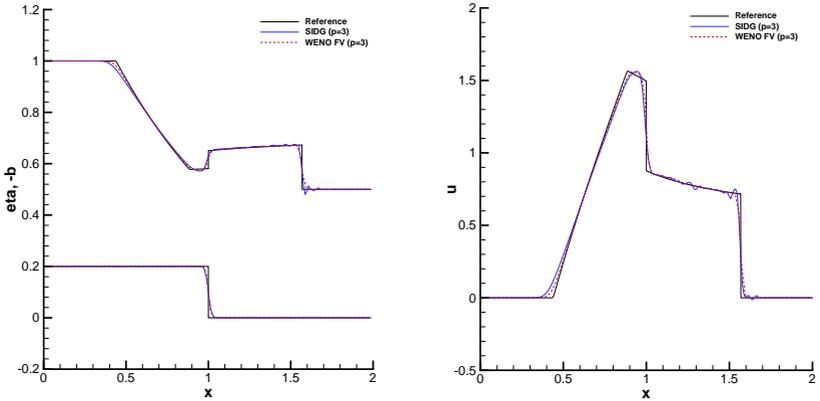


Figure 2.10: Comparison between numerical and reference solution at time $t_{end} = 0.20$: free surface elevation η and $-b$ on the left; velocity u in the x direction on the right.

2.3.5 Low Froude number flow around a circular cylinder

In this last test problem we consider a steady low Froude number flow around a circular cylinder. This test problem has been solved for the first time by Bassi and Rebay with high order isoparametric DG schemes, solving the Euler equations of compressible gas dynamics at low Mach numbers, see [13]. Bassi and Rebay have shown that a physically correct steady state solution can be only obtained when *curved* isoparametric elements are used. In the case of subparametric straight-line elements an unphysical, unsteady and highly oscillatory solution is obtained. Therefore, in this section we will also present results for both subparametric straight line elements and for isoparametric curved elements. The computational domain is

$$\Omega = [-3.0, 3.0] \times [-3.0, 3.0] - \{\sqrt{x^2 + y^2} \leq 0.5\}$$

and is discretized using a very coarse mesh of only $N_e = 308$ triangles.

A sketch of the main and the dual grid is plotted in Fig. 2.11, while a detail of the circular cylinder for the curved isoparametric and non-curved subparamet-

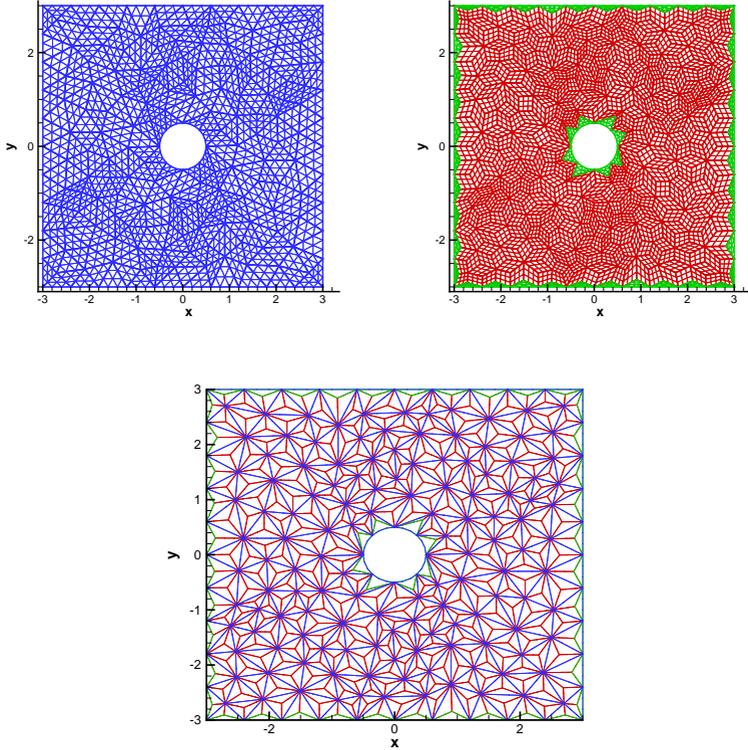


Figure 2.11: Main triangular grid with nodal subgrid (top left); dual grid with nodal subgrid (top right) and both grids (bottom) for the circular cylinder testcase; the blue triangles define the primary mesh, the red quadrilaterals define the dual mesh and the green triangular elements are the boundary elements for the dual grid.

Table 2.5: Comparison between fourth order staggered unstructured semi-implicit DG scheme and unstructured path-conservative ADER-WENO finite volume method for the circular dambreak problem over a bottom step.

	WENO finite volume	Semi-implicit DG
Mesh type	cell-centered	staggered (primal & dual)
Element type	triangles	mixed tri and quad
Formal order	4 in space	4 in space
No. of DOF	81916	81920
Time step size	$1.15 \cdot 10^{-3}$	$1.6 \cdot 10^{-3}$
Memory usage (RAM)	1.97 GB	1.60 GB
CPU time	1496.8 s	1502.4 s

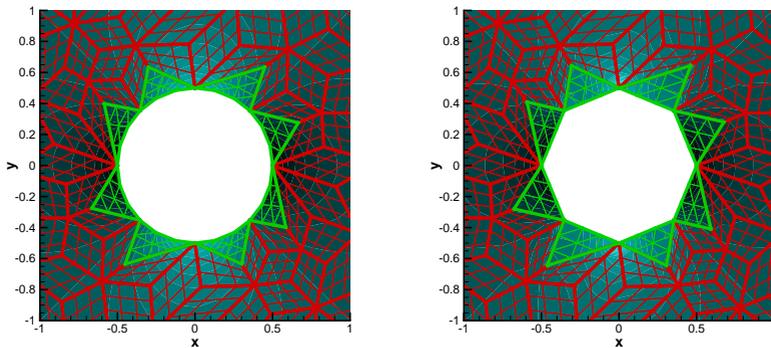


Figure 2.12: Differences between the grid obtained with the curved isoparametric approach (left) and the non-curved approach (right).

ric case is depicted in Fig. 2.12. The analytical solution for this test problem is known both for the velocity field and for the free surface elevation. In order to avoid the generation of strong initial transient waves we impose as initial condition the exact velocity field, but a flat free surface:

$$\eta(x, y, 0) = 1, \quad b(x, y) = 0, \quad (2.54)$$

and

$$\begin{aligned} v_r(r, \alpha, 0) &= \bar{u} \left(1 - \frac{R_c^2}{r^2} \right) \cos(\alpha), \\ v_\alpha(r, \alpha, 0) &= -\bar{u} \left(1 + \frac{R_c^2}{r^2} \right) \sin(\alpha), \end{aligned} \quad (2.55)$$

where v_r and v_α describe the velocity field in polar coordinates; R_c is the radius of the circular cylinder and \bar{u} is the initial velocity of this steady flow problem. Bernoulli's equation allows us to compute the exact free surface elevation from the velocity field:

$$\eta = \eta_\infty + \frac{1}{2g} \bar{u} \left(2 \frac{R_c^2}{r^2} \cos(2\alpha) - \frac{R_c^4}{r^4} \right), \quad (2.56)$$

where η_∞ is the asymptotic value of the free surface elevation far from the circular cylinder. For the present test we take $\eta_\infty = 1$; $\bar{u} = 0.01$ in order to obtain a very small Froude number $Fr = |\vec{v}|/\sqrt{gH} = 3.2 \cdot 10^{-3}$; furthermore $R_c = 0.5$; $\theta = 1.0$; $t_{end} = 4.75$; and $p = 3$. The time step size is taken as the one imposed by the CFL restriction for the nonlinear convective term.

As shown by Bassi and Rebay [13], this test does not work if we use high order straight line DG elements near the cylinder. The numerical results obtained by using high order curved isoparametric elements are shown in Fig. 2.15 and are compared against the exact solution of the problem. A very good agreement can be observed for this very coarse grid. If the grid is further refined, the numerical solution becomes indistinguishable from the exact one.

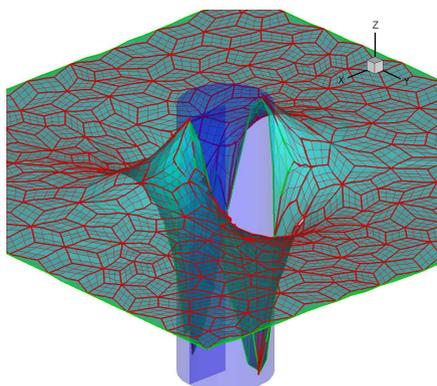


Figure 2.13: Profile of η at time $t = t_{end}$ using the isoparametric staggered DG approach.

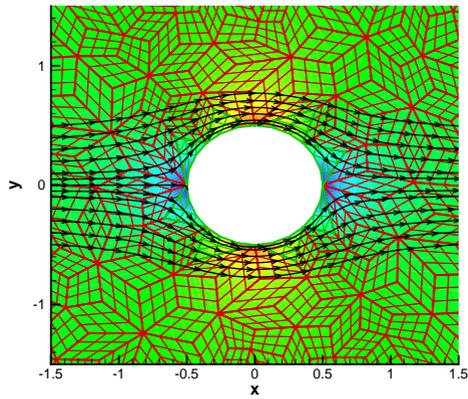


Figure 2.14: Streamlines and pressure contours using the isoparametric staggered semi-implicit DG scheme for the circular cylinder problem on an unstructured triangular mesh.

2 Two-dimensional shallow water equations

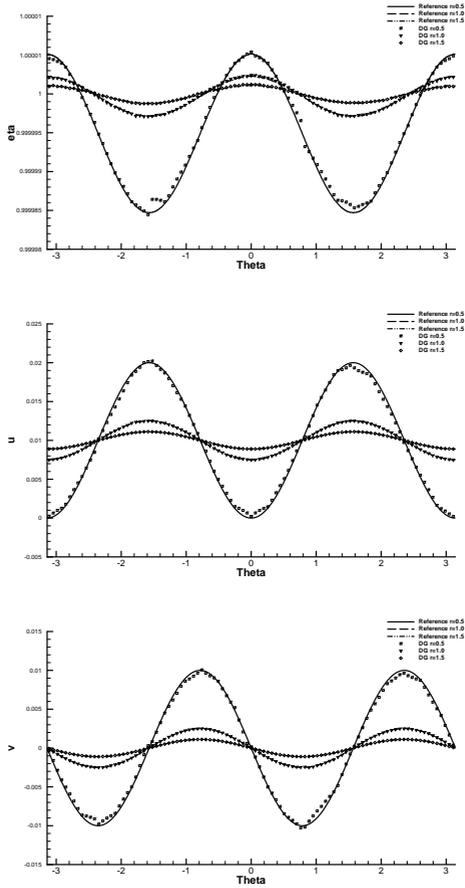


Figure 2.15: Comparison of the numerical solution against the exact solution for $\theta = [-\pi, \pi]$ and several radii. From the top to the bottom we have plotted the profile of η , u and v

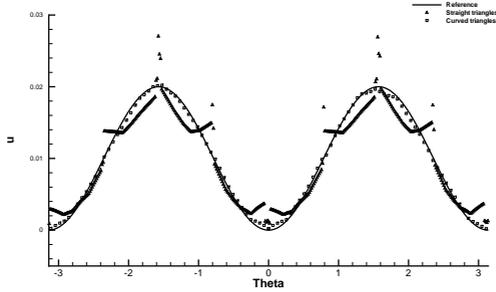


Figure 2.16: Comparison between the velocity u for the curved approach and the non-curved approach at $r = 0.5$, in this last case the convection term is taken such as $F\vec{v} = \vec{v}$.

In Fig 2.16 we plot a comparison between the velocity u at $r = 0.5$ obtained for curved isoparametric elements and for simple straight line elements. As shown, the velocity obtained with straight line elements presents unphysical peaks near the corners, due to the well-known corner singularity of the potential flow. In contrast, the high order curved isoparametric elements reproduce the physically correct smooth solution in the corners, without spurious oscillations. It is important to underline that also for the present high order *staggered* DG method the use of *curved* elements is crucial, which confirms the findings of Bassi and Rebay [13] for high order DG methods on *collocated* grids.

3 Two-dimensional incompressible Navier-Stokes equations

3.1 Overview

In this chapter we want to extend the staggered DG scheme presented in the previous Chapter 2 to the two dimensional incompressible Navier-Stokes equations. The apparently similar formal structure of the equations allows us to derive the method in a very similar way. In particular, for an inviscid flow we can recover the incompressible Navier-Stokes system from 2.1 – 2.2 by setting $H = g \equiv 1$, interpreting the free surface elevation as the pressure and neglect the term $\frac{\partial \eta}{\partial t}$ in the continuity equation. In this chapter we further introduce the viscosity as well as a source term in the momentum equation. As we will see these simple modifications in the equations lead to several important differences in the properties of the resulting algorithm. In this chapter we further derive a general method in order to achieve arbitrary high order in both space and time. The direct extension of the method presented in the previous chapter leads to the special case of high order only in space and can be found in Section 3.3, while a set of numerical tests are presented in Section 3.6 for the space and space-time version of the algorithm.

3.1.1 Governing equations

The two dimensional incompressible Navier-Stokes equations are given by

$$\frac{\partial \mathbf{v}}{\partial t} + \nabla \cdot \mathbf{F}_c + \nabla p = \nu \Delta \mathbf{v} + \mathbf{S}, \quad (3.1)$$

$$\nabla \cdot \mathbf{v} = 0, \quad (3.2)$$

where $p = P/\rho$ indicates the normalized fluid pressure; P is the physical pressure and ρ is the constant fluid density; $\nu = \mu/\rho$ is the kinematic viscosity coefficient; $\mathbf{v} = (u, v)$ is the velocity vector; u and v are the velocity components in the x and y direction, respectively; $\mathbf{S} = \mathbf{S}(\mathbf{v}, x, y, t)$ is a (nonlinear) algebraic source term; $\mathbf{F}_c = \mathbf{v} \otimes \mathbf{v}$ is the flux tensor of the nonlinear convective

terms, namely:

$$\mathbf{F}_c = \begin{pmatrix} uu & uv \\ vu & vv \end{pmatrix}.$$

Following the same idea of [88, 67], the viscosity term is first written as $\nu \Delta \mathbf{v} = \nabla \cdot (\nu \nabla \mathbf{v})$ and then grouped with the nonlinear convective term. In this way the momentum Eq. (3.1) can be rewritten as

$$\frac{\partial \mathbf{v}}{\partial t} + \nabla \cdot \mathbf{F} + \nabla p = \mathbf{S}, \quad (3.3)$$

where $\mathbf{F} = \mathbf{F}(\mathbf{v}, \nabla \mathbf{v}) = \mathbf{F}_c(\mathbf{v}) - \nu \nabla \mathbf{v}$ is a nonlinear tensor that depends on the velocity and its gradient, see e.g. [88, 67].

3.2 Space-time DG scheme for the 2D incompressible Navier-Stokes equations

3.2.1 Space-time extension of the unstructured staggered grid

The spatial staggered unstructured mesh is taken such as the one described in Section 2.2.1. In the time direction we cover the time interval $[0, T]$ with a sequence of times $0 = t^0 < t^1 < t^2 \dots < t^N < t^{N+1} = T$. We denote the time step by $\Delta t^{n+1} = t^{n+1} - t^n$ and the corresponding time interval by $T^{n+1} = [t^n, t^{n+1}]$ for $n = 0 \dots N$. In order to ease notation, sometimes we will use the abbreviation $\Delta t = \Delta t^{n+1}$. In this way the generic space-time element defined in the time interval $[t^n, t^{n+1}]$ is given by $\mathbf{T}_i^{st} = \mathbf{T}_i \times T^{n+1}$ for the main grid, and $\mathbf{R}_j^{st} = \mathbf{R}_j \times T^{n+1}$ for the dual grid. Figure 3.1 shows a graphical representation of the primary and dual space-time control volumes.

3.2.2 Space-time basis functions

We can easily extend the idea used in Section 2.2.2 and construct the time basis functions on a reference interval $[0, 1]$ for polynomials of degree p_γ . In this case the resulting $N_\gamma = p_\gamma + 1$ basis functions $\{\gamma_k\}_{k \in [1, N_\gamma]}$ are defined as the Lagrange interpolation polynomials passing through the Gauss-Legendre quadrature points for the unit interval. For every time interval $[t^n, t^{n+1}]$, the map between the reference interval and the physical one is simply given by $t = t^n + \tau \Delta t^{n+1}$ for every $\tau \in [0, 1]$. Using the tensor product we can finally construct the basis functions on the space-time elements \mathbf{T}_i^{st} and \mathbf{R}_j^{st} such as $\tilde{\phi}(\xi, \gamma, \tau) = \phi(\xi, \gamma) \cdot \gamma(\tau)$ and $\tilde{\psi}(\xi, \gamma, \tau) = \psi(\xi, \gamma) \cdot \gamma(\tau)$. The total number of basis functions becomes $N_\phi^{st} = N_\phi \cdot N_\gamma$ and $N_\psi^{st} = N_\psi \cdot N_\gamma$.

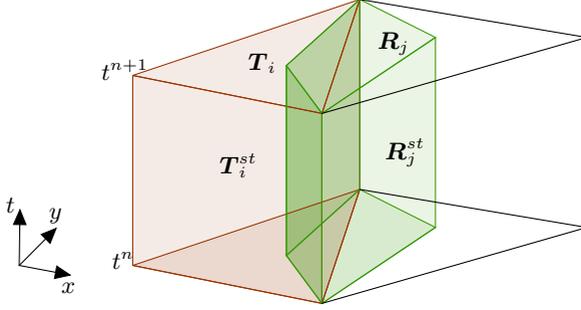


Figure 3.1: Example of space-time elements T_i^{st} (red) and R_j^{st} (green) with $j \in S_i$.

3.2.3 Semi-implicit space-time DG scheme

We extend the spaces of piecewise polynomials used on the main and the dual grid following definition (2.5), where now the space-time polynomials are taken as the one defined in the previous chapter and extended using tensor product of one-dimensional polynomials of degree at most p_γ on T^{n+1} . In this way we could have different polynomial degrees in space and time and hence we could analyze the special case of $p > 0$ and $p_\gamma = 0$.

The discrete pressure p_h is defined on the main grid, namely $p_h(x, y, t)|_{T_i^{st}} = p_i(x, y, t)$, while the discrete velocity vector field \mathbf{v}_h is defined on the dual grid, namely $\mathbf{v}_h(x, y, t)|_{R_j^{st}} = \mathbf{v}_j(x, y, t)$.

The numerical solution of (3.1)-(3.3) is represented inside the space-time control volumes of the primal and the dual grid during the current time interval T^{n+1}

by piecewise space-time polynomials as follows:

$$p_i(x, y, t) = \sum_{l=1}^{N_\phi^{st}} \tilde{\phi}_l^{(i)}(x, y, t) \hat{p}_{l,i}^{n+1} =: \tilde{\phi}^{(i)}(x, y, t) \hat{\mathbf{p}}_i^{n+1}, \quad (3.4)$$

$$\mathbf{v}_j(x, y, t) = \sum_{l=1}^{N_\psi^{st}} \tilde{\psi}_l^{(j)}(x, y, t) \hat{\mathbf{v}}_{l,j}^{n+1} =: \tilde{\psi}^{(j)}(x, y, t) \hat{\mathbf{v}}_j^{n+1}, \quad (3.5)$$

where the vectors of basis functions $\tilde{\phi}(x, y, t)$ and $\tilde{\psi}(x, y, t)$ are generated via the mappings from $\tilde{\phi}(\xi, \gamma, \tau)$ on $T_{std} \times [0, 1]$ and $\tilde{\psi}(\xi, \gamma, \tau)$ on $R_{std} \times [0, 1]$, respectively.

A weak formulation of the continuity equation (3.1) is obtained by multiplying it by $\tilde{\phi}$ and integrating over a control volume \mathbf{T}_i^{st} , for every $k = 1 \dots N_\phi^{st}$. The resulting weak formulation for the discrete velocity \mathbf{v}_h reads

$$\int_{\mathbf{T}_i^{st}} \tilde{\phi}_k^{(i)} \nabla \cdot \mathbf{v}_h \, d\mathbf{x}dt = 0. \quad (3.6)$$

Similarly, multiplication of the momentum equation (3.3) by $\tilde{\psi}$ and integrating over a control volume \mathbf{R}_j^{st} yields

$$\int_{\mathbf{R}_j^{st}} \tilde{\psi}_k^{(j)} \left(\frac{\partial \mathbf{v}_h}{\partial t} + \nabla \cdot \mathbf{F}_h \right) d\mathbf{x}dt + \int_{\mathbf{R}_j^{st}} \tilde{\psi}_k^{(j)} \nabla p_h \, d\mathbf{x}dt = \int_{\mathbf{R}_j^{st}} \tilde{\psi}_k^{(j)} \mathbf{S}_h \, d\mathbf{x}dt, \quad (3.7)$$

for every $j = 1 \dots N_d$ and $k = 1 \dots N_\psi^{st}$. Using integration by parts Eq. (3.6) becomes

$$\oint_{\partial \mathbf{T}_i^{st}} \tilde{\phi}_k^{(i)} \mathbf{v}_h \cdot \tilde{\mathbf{n}}_i \, dsdt - \int_{\mathbf{T}_i^{st}} \nabla \tilde{\phi}_k^{(i)} \cdot \mathbf{v}_h \, d\mathbf{x}dt = 0, \quad (3.8)$$

where $\tilde{\mathbf{n}}_i$ indicates the outward pointing unit normal vector. Since p_h is allowed to jump at the element boundaries of the primary triangular grid and \mathbf{v}_h may jump at the boundaries of the dual quadrilateral elements, the integrals appearing in equations (3.7) and (3.8) have to be split into elementwise contributions. Note, however, that thanks to the use of a *staggered grid* we do *not* need a Riemann solver here, since all the quantities are readily defined where needed for the flux computation. In particular, the velocity is continuous across

the boundaries of the triangles on the main grid and the pressure is continuous across the boundaries of the dual quadrilateral grid.

If we want to integrate the second term in (3.7) directly, without forward and backward integration by parts, the gradient of the discrete pressure p_h appearing in the weak form of the momentum equation (3.7) can be easily interpreted in the sense of distributions, similar to a non-conservative product, since the discrete pressure is allowed to jump at Γ_j^{st} inside a dual velocity control volume.¹ At this point we refer the reader to the extensive literature on high order accurate path-conservative finite volume and discontinuous Galerkin finite element schemes, where weak formulations of PDE with non-conservative products have already been derived and discussed in great detail, see e.g. [27, 124, 119, 28, 134, 69, 73, 143].

We therefore obtain the following weak form of the continuity and the momentum equation:

$$\sum_{j \in S_i} \left(\int_{\Gamma_j^{st}} \tilde{\phi}_k^{(i)} \mathbf{v}_j \cdot \tilde{\mathbf{n}}_{ij} \, ds dt - \int_{\mathbf{T}_{i,j}^{st}} \nabla \tilde{\phi}_k^{(i)} \cdot \mathbf{v}_j \, d\mathbf{x} dt \right) = 0, \quad (3.9)$$

and

$$\begin{aligned} & \int_{\mathbf{R}_j^{st}} \tilde{\psi}_k^{(j)} \left(\frac{\partial \mathbf{v}_j}{\partial t} + \nabla \cdot \mathbf{F}_j \right) d\mathbf{x} dt \\ & + \int_{\mathbf{T}_{\ell(j),j}^{st}} \tilde{\psi}_k^{(j)} \nabla p_{\ell(j)} d\mathbf{x} dt + \int_{\mathbf{T}_{r(j),j}^{st}} \tilde{\psi}_k^{(j)} \nabla p_{r(j)} d\mathbf{x} dt \\ & + \int_{\Gamma_j^{st}} \tilde{\psi}_k^{(j)} (p_{r(j)} - p_{\ell(j)}) \tilde{\mathbf{n}}_j \, ds dt = \int_{\mathbf{R}_j^{st}} \tilde{\psi}_k^{(j)} \mathbf{S} d\mathbf{x} dt, \end{aligned} \quad (3.10)$$

where $\tilde{\mathbf{n}}_{ij} = \tilde{\mathbf{n}}_i|_{\Gamma_j^{st}}$; $\mathbf{T}_{i,j}^{st} = \mathbf{T}_{i,j} \times T^{n+1}$; and $\Gamma_j^{st} = \Gamma_j \times T^{n+1}$. We stress again that the same weak form of the momentum equation (3.10) could have been also derived by forward and backward integration by parts of (3.7), following

¹Note that the term ∇p_h is not a true non-conservative product, since it can be written in a conservative divergence form as $\nabla p_h = \nabla \cdot (p_h \mathbf{I})$, with the identity matrix \mathbf{I} . Nevertheless, if p_h contains discontinuities, the term ∇p_h can be properly interpreted as a special case within the more general framework of PDE with non-conservative products.

the ideas of Bassi and Rebay [12]. Using definitions (3.4) and (3.5), we conveniently rewrite the above equations as

$$\sum_{j \in \mathcal{S}_i} \left(\int_{\Gamma_j^{st}} \tilde{\phi}_k^{(i)} \tilde{\psi}_l^{(j)} \tilde{n}_{ij} ds dt \hat{\mathbf{v}}_{l,j}^{n+1} - \int_{\mathbf{T}_{i,j}^{st}} \nabla \tilde{\phi}_k^{(i)} \tilde{\psi}_l^{(j)} d\mathbf{x} dt \hat{\mathbf{v}}_{l,j}^{n+1} \right) = 0, \quad (3.11)$$

and

$$\begin{aligned} & \int_{\mathbf{R}_j^{st}} \tilde{\psi}_k^{(j)} \frac{\partial \mathbf{v}_j}{\partial t} d\mathbf{x} dt + \int_{\mathbf{R}_j^{st}} \tilde{\psi}_k^{(j)} \nabla \cdot \mathbf{F} d\mathbf{x} dt \\ & + \int_{\mathbf{T}_{\ell(j),j}^{st}} \tilde{\psi}_k^{(j)} \nabla \tilde{\phi}_l^{(\ell(j))} d\mathbf{x} dt \hat{p}_{l,\ell(j)}^{n+1} + \int_{\mathbf{T}_{r(j),j}^{st}} \tilde{\psi}_k^{(j)} \nabla \tilde{\phi}_l^{(r(j))} d\mathbf{x} dt \hat{p}_{l,r(j)}^{n+1} \\ & + \int_{\Gamma_j^{st}} \tilde{\psi}_k^{(j)} \tilde{\phi}_l^{(r(j))} \tilde{n}_j ds dt \hat{p}_{l,r(j)}^{n+1} - \int_{\Gamma_j^{st}} \tilde{\psi}_k^{(j)} \tilde{\phi}_l^{(\ell(j))} \tilde{n}_j ds dt \hat{p}_{l,\ell(j)}^{n+1} = \int_{\mathbf{R}_j^{st}} \tilde{\psi}_k^{(j)} \mathbf{S} d\mathbf{x} dt, \end{aligned} \quad (3.12)$$

where we have used the standard summation convention for the repeated index l . Integrating the first integral in (3.12) by parts in time and assuming the mesh to be *fixed* in time (i.e. the elements do not move), we obtain

$$\begin{aligned} \int_{\mathbf{R}_j^{st}} \tilde{\psi}_k^{(j)} \frac{\partial \mathbf{v}_j}{\partial t} d\mathbf{x} dt &= \left[\int_{\mathbf{R}_j} \tilde{\psi}_k^{(j)} \mathbf{v}_j d\mathbf{x} \right]_{t=t^{n+1}} - \left[\int_{\mathbf{R}_j} \tilde{\psi}_k^{(j)} \mathbf{v}_j d\mathbf{x} \right]_{t=t^n} \\ &\quad - \int_{\mathbf{R}_j^{st}} \frac{\partial \tilde{\psi}_k^{(j)}}{\partial t} \mathbf{v}_j d\mathbf{x} dt \end{aligned} \quad (3.13)$$

In Eq. (3.13) we can recognize the fluxes between the current space-time element $\mathbf{R}_j \times T^{n+1}$, the future space-time slab and the past space-time elements, as well as an internal space-time volume contribution that connects the layers inside the space-time element \mathbf{R}_j^{st} in an asymmetric way. Note how the asymmetry affects only the space-time volume contribution in (3.13). This is due to the nature of the time derivative operator, which has a natural positive direction given by the causality principle in time. Also note that the surface

3.2 Space-time DG scheme for the 2D incompressible Navier-Stokes equations

integral over the element at the past time includes the initial condition of the velocity in a weak sense. It can also be interpreted as using an *upwind flux in time direction*. By substituting Eq. (3.13) into (3.12) we obtain the following weak formulation of the momentum equation in space-time:

$$\begin{aligned}
 & \left(\left[\int_{\mathbf{R}_j} \tilde{\psi}_k^{(j)} \tilde{\psi}_l^{(j)} d\mathbf{x} \right]_{t=t^{n+1}} - \int_{\mathbf{R}_j^{st}} \frac{\partial \tilde{\psi}_k^{(j)}}{\partial t} \tilde{\psi}_l^{(j)} d\mathbf{x} dt \right) \hat{\mathbf{v}}_{l,j}^{n+1} \\
 & + \int_{\mathbf{T}_{\ell(j),j}^{st}} \tilde{\psi}_k^{(j)} \nabla \tilde{\phi}_l^{(\ell(j))} d\mathbf{x} \hat{p}_{l,\ell(j)}^{n+1} + \int_{\mathbf{T}_{r(j),j}^{st}} \tilde{\psi}_k^{(j)} \nabla \tilde{\phi}_l^{(r(j))} d\mathbf{x} \hat{p}_{l,r(j)}^{n+1} \\
 & + \int_{\Gamma_j^{st}} \tilde{\psi}_k^{(j)} \tilde{\phi}_l^{(r(j))} \vec{n}_j ds \hat{p}_{l,r(j)}^{n+1} - \int_{\Gamma_j^{st}} \tilde{\psi}_k^{(j)} \tilde{\phi}_l^{(\ell(j))} \vec{n}_j ds \hat{p}_{l,\ell(j)}^{n+1} \\
 & = \left[\int_{\mathbf{R}_j} \tilde{\psi}_k^{(j)} \tilde{\psi}_l^{(j)} d\mathbf{x} \right]_{t=t^n} \hat{\mathbf{v}}_{l,j}^n - \int_{\mathbf{R}_j^{st}} \tilde{\psi}_k^{(j)} \nabla \cdot \mathbf{F} d\mathbf{x} + \int_{\mathbf{R}_j^{st}} \tilde{\psi}_k^{(j)} \mathbf{S} d\mathbf{x} dt,
 \end{aligned} \tag{3.14}$$

For every i and j , Eqs. (3.11)-(3.14) are written in a compact matrix form as

$$\sum_{j \in \mathcal{S}_i} \mathcal{D}_{i,j} \hat{\mathbf{v}}_j^{n+1} = 0, \tag{3.15}$$

and

$$(\mathbf{M}_j^+ - \mathbf{M}_j^\circ) \hat{\mathbf{v}}_j^{n+1} - \mathbf{M}_j^- \hat{\mathbf{v}}_j^n + \Upsilon_j(\mathbf{v}, \nabla \mathbf{v}) + \mathcal{R}_j \hat{\mathbf{p}}_{r(j)}^{n+1} - \mathcal{L}_j \hat{\mathbf{p}}_{\ell(j)}^{n+1} = \mathbf{S}_j, \tag{3.16}$$

respectively, where:

$$\mathbf{M}_j^+ = \int_{\mathbf{R}_j} \tilde{\psi}_k^{(j)}(x, y, t^{n+1}) \tilde{\psi}_l^{(j)}(x, y, t^{n+1}) d\mathbf{x}, \quad (3.17)$$

$$\mathbf{M}_j^- = \int_{\mathbf{R}_j} \tilde{\psi}_k^{(j)}(x, y, t^{n+1}) \tilde{\psi}_l^{(j)}(x, y, t^n) d\mathbf{x}, \quad (3.18)$$

$$\mathbf{M}_j^\circ = \int_{\mathbf{R}_j^{st}} \frac{\partial \tilde{\psi}_k^{(j)}}{\partial t} \tilde{\psi}_l^{(j)} d\mathbf{x} dt, \quad (3.19)$$

$$\Upsilon_j = \int_{\mathbf{R}_j^{st}} \tilde{\psi}_k^{(j)} \nabla \cdot \mathbf{F} d\mathbf{x} dt \quad (3.20)$$

$$\mathcal{D}_{i,j} = \int_{\Gamma_j^{st}} \tilde{\phi}_k^{(i)} \tilde{\psi}_l^{(j)} \tilde{n}_{ij} ds dt - \int_{\mathbf{T}_{i,j}^{st}} \nabla \tilde{\phi}_k^{(i)} \tilde{\psi}_l^{(j)} d\mathbf{x} dt, \quad (3.21)$$

$$\mathcal{R}_j = \int_{\Gamma_j^{st}} \tilde{\psi}_k^{(j)} \tilde{\phi}_l^{(r(j))} \tilde{n}_j ds dt + \int_{\mathbf{T}_{r(j),j}^{st}} \tilde{\psi}_k^{(j)} \nabla \tilde{\phi}_l^{(r(j))} d\mathbf{x} dt, \quad (3.22)$$

$$\mathcal{L}_j = \int_{\Gamma_j^{st}} \tilde{\psi}_k^{(j)} \tilde{\phi}_l^{(\ell(j))} \tilde{n}_j ds dt - \int_{\mathbf{T}_{\ell(j),j}^{st}} \tilde{\psi}_k^{(j)} \nabla \tilde{\phi}_l^{(\ell(j))} d\mathbf{x} dt, \quad (3.23)$$

$$\mathcal{S}_j = \int_{\mathbf{R}_j^{st}} \tilde{\psi}_k^{(j)} \mathbf{S} d\mathbf{x} dt. \quad (3.24)$$

Remark how \mathbf{M}_j° introduces, for polynomial degrees $p_\gamma > 0$, an asymmetric contribution in time. The action of matrices \mathcal{L} and \mathcal{R} can be generalized by introducing a new matrix $\mathcal{Q}_{i,j}$, defined as

$$\mathcal{Q}_{i,j} = \int_{\mathbf{T}_{i,j}^{st}} \tilde{\psi}_k^{(j)} \nabla \tilde{\phi}_l^{(i)} d\mathbf{x} dt - \int_{\Gamma_j^{st}} \tilde{\psi}_k^{(j)} \tilde{\phi}_l^{(i)} \sigma_{i,j} \tilde{n}_j ds dt, \quad (3.25)$$

where $\sigma_{i,j}$ is a sign function defined by

$$\sigma_{i,j} = \frac{r(j) - 2i + \ell(j)}{r(j) - \ell(j)}. \quad (3.26)$$

3.2 Space-time DG scheme for the 2D incompressible Navier-Stokes equations

In this way $\mathcal{Q}_{\ell(j),j} = -\mathcal{L}_j$ and $\mathcal{Q}_{r(j),j} = \mathcal{R}_j$, and then Eq. (3.16) becomes in terms of \mathcal{Q}

$$(M_j^+ - M_j^\circ) \hat{\mathbf{v}}_j^{n+1} - M_j^- \hat{\mathbf{v}}_j^n + \Upsilon_j(\mathbf{v}, \nabla \mathbf{v}) + \mathcal{Q}_{r(j),j} \hat{\mathbf{p}}_{r(j)}^{n+1} + \mathcal{Q}_{\ell(j),j} \hat{\mathbf{p}}_{\ell(j)}^{n+1} = \mathcal{S}_j, \quad (3.27)$$

or, equivalently,

$$(M_j^+ - M_j^\circ) \hat{\mathbf{v}}_j^{n+1} - M_j^- \hat{\mathbf{v}}_j^n + \Upsilon_j(\mathbf{v}, \nabla \mathbf{v}) + \mathcal{Q}_{i,j} \hat{\mathbf{p}}_i^{n+1} + \mathcal{Q}_{\varphi(i,j),j} \hat{\mathbf{p}}_{\varphi(i,j)}^{n+1} = \mathcal{S}_j. \quad (3.28)$$

In order to further ease notation, we will use the abbreviation $M_j = M_j^+ - M_j^\circ$ henceforth and will write Eqs. (3.15)-(3.16) as follows:

$$\sum_{j \in S_i} \mathcal{D}_{i,j} \hat{\mathbf{v}}_j^{n+1} = 0, \quad (3.29)$$

$$M_j \hat{\mathbf{v}}_j^{n+1} - M_j \widehat{\mathbf{F}} \hat{\mathbf{v}}_j + \mathcal{Q}_{r(j),j} \hat{\mathbf{p}}_{r(j)}^{n+1} + \mathcal{Q}_{\ell(j),j} \hat{\mathbf{p}}_{\ell(j)}^{n+1} = 0, \quad (3.30)$$

where $\widehat{\mathbf{F}} \hat{\mathbf{v}}_j$ is an appropriate discretization of the nonlinear convective, viscous and source terms. The details for the computation of $\widehat{\mathbf{F}} \hat{\mathbf{v}}_j$ will be presented later. Formal substitution of the discrete momentum equation (3.30) into the discrete continuity equation (3.29), see also [37, 70, 143, 144], yields

$$\sum_{j \in S_i} \mathcal{D}_{i,j} M_j^{-1} \mathcal{Q}_{i,j} \hat{\mathbf{p}}_i^{n+1} + \sum_{j \in S_i} \mathcal{D}_{i,j} M_j^{-1} \mathcal{Q}_{\varphi(i,j),j} \hat{\mathbf{p}}_{\varphi(i,j)}^{n+1} = \sum_{j \in S_i} \mathcal{D}_{i,j} \widehat{\mathbf{F}} \hat{\mathbf{v}}_j, \quad (3.31)$$

We have now to choose a time discretization for the nonlinear convective-viscous term. The simplest choice would be to take $\widehat{\mathbf{F}} \hat{\mathbf{v}}_j$ explicitly, so in this case $\sum_{j \in S_i} \mathcal{D}_{i,j} \widehat{\mathbf{F}} \hat{\mathbf{v}}_j^n$ becomes a known term at time t^n and hence Eq. (3.31) would represent a four-point block system for the new pressure $\hat{\mathbf{p}}_i^{n+1}$, as proposed in [144]. Unfortunately, in problems where the convective-viscous effects cannot be neglected, this will produce only a low order accurate method in time. The problem in this case is that the convective-viscous contribution in the time interval T^{n+1} is based on the old information T^n and does not see the effects of the new pressure in the time interval T^{n+1} . Furthermore, if we take $\widehat{\mathbf{F}} \hat{\mathbf{v}}_j$ implicitly, then system (3.31) becomes nonlinear and it would be very cumbersome to solve it. In order to overcome this problem we introduce a simple *Picard iteration* to get the information of the new pressure into the viscous and convective terms, but without introducing a nonlinearity in the final system to be solved. This approach is inspired by the local space-time

Galerkin predictor method proposed for the high order time discretization of $P_N P_M$ schemes in [68, 67]. Hence, for $k = 1, N_{pic}$, we rewrite system (3.31) as

$$\begin{aligned} \sum_{j \in S_i} \mathcal{D}_{i,j} M_j^{-1} \mathcal{Q}_{i,j} \hat{\mathbf{p}}_i^{n+1,k+1} + \sum_{j \in S_i} \mathcal{D}_{i,j} M_j^{-1} \mathcal{Q}_{\varphi(i,j),j} \hat{\mathbf{p}}_{\varphi(i,j)}^{n+1,k+1} \\ = \sum_{j \in S_i} \mathcal{D}_{i,j} \widehat{\mathbf{F}} \bar{\mathbf{v}}_j^{n+1,k+\frac{1}{2}}, \end{aligned} \quad (3.32)$$

or, by introducing the boundary elements (see e.g. [144]),

$$\begin{aligned} \left[\sum_{j \in S_i \cap \mathcal{B}(\Omega)} \mathcal{D}_{i,j}^{\partial} M_j^{-1} \mathcal{Q}_{i,j}^{\partial} - \sum_{j \in S_i - \mathcal{B}(\Omega)} \mathcal{D}_{i,j} M_j^{-1} \mathcal{Q}_{i,j} \right] \hat{\mathbf{p}}_i^{n+1,k+1} \\ - \sum_{j \in S_i - \mathcal{B}(\Omega)} \mathcal{D}_{i,j} M_j^{-1} \mathcal{Q}_{\varphi(i,j),j} \hat{\mathbf{p}}_{\varphi(i,j)}^{n+1,k+1} = \\ - \sum_{j \in S_i - \mathcal{B}(\Omega)} \mathcal{D}_{i,j} \widehat{\mathbf{F}} \bar{\mathbf{v}}_j^{n+1,k+\frac{1}{2}} + \sum_{j \in S_i \cap \mathcal{B}(\Omega)} \mathcal{D}_{i,j}^{\partial} \widehat{\mathbf{F}} \bar{\mathbf{v}}_j^{n+1,k+\frac{1}{2}}, \end{aligned} \quad (3.33)$$

where $\mathcal{D}_{i,j}^{\partial}$ and $\mathcal{Q}_{i,j}^{\partial}$ are the natural extension of \mathcal{D} and \mathcal{Q} on triangular dual boundary elements, see e.g. [144]. Now the right hand side of Eq. (3.31) can be computed by using the velocity field at the old Picard iteration k and including the viscous effects using a fractional step type procedure. In this way, Eq. (3.31) represents a block four-point system for the new pressure $\hat{\mathbf{p}}_i^{n+1,k+1}$. Once the new pressure field is known, the velocity vector field at the new Picard iteration $\hat{\mathbf{v}}_i^{n+1,k+1}$ can be readily updated from the momentum equation (3.30).

3.2.4 Nonlinear convection-diffusion

To close the problem it remains to specify how to construct the nonlinear convection-diffusion operator $\widehat{\mathbf{F}} \bar{\mathbf{v}}_j^{n+1,k+\frac{1}{2}}$. Following the ideas of [144], a space-time DG scheme for the convection-diffusion terms on the dual mesh is given by

$$\begin{aligned} \int_{\mathbf{R}_j^{st}} \tilde{\psi}_k \frac{\partial}{\partial t} \mathbf{v}_h \, d\mathbf{x}dt + \int_{\partial \mathbf{R}_j^{st}} \tilde{\psi}_k \mathbf{G}_h \cdot \bar{\mathbf{n}} \, dsdt \\ - \int_{\mathbf{R}_j^{st}} \nabla \tilde{\psi}_k \cdot \mathbf{F}(\mathbf{v}_h, \nabla \mathbf{v}_h) \, d\mathbf{x}dt = \int_{\mathbf{R}_j^{st}} \tilde{\psi}_k^{(j)} \mathbf{S} \, d\mathbf{x}dt, \end{aligned} \quad (3.34)$$

3.2 Space-time DG scheme for the 2D incompressible Navier-Stokes equations

and the numerical flux for both the convective and the viscous contribution, is given such as in [137, 88, 67], and reads

$$\mathbf{G}_h \cdot \vec{n} = \frac{1}{2} (\mathbf{F}(\mathbf{v}_h^+, \nabla \mathbf{v}_h^+) + \mathbf{F}(\mathbf{v}_h^-, \nabla \mathbf{v}_h^-)) \cdot \vec{n} - \frac{1}{2} s_{\max} (\mathbf{v}_h^+ - \mathbf{v}_h^-), \quad (3.35)$$

with

$$s_{\max} = 2 \max(|\mathbf{v}_h^- \cdot \vec{n}|, |\mathbf{v}_h^+ \cdot \vec{n}|) + \frac{2\nu}{h^+ + h^-} \frac{2p+1}{\sqrt{\frac{\pi}{2}}}, \quad (3.36)$$

which contains the maximum eigenvalue of the Jacobian matrix of the purely convective transport operator \mathbf{F}_c in normal direction, see [70], and the stabilization term for the viscous flux, see [67, 88]. Furthermore, the \mathbf{v}_h^\pm and $\nabla \mathbf{v}_h^\pm$ denote the velocity vectors and their gradients, extrapolated to the boundary of \mathbf{R}_j from within the element \mathbf{R}_j and from the neighbor element, respectively. h^+ and h^- are the maximum radii of the inscribed circle in \mathbf{R}_j and the neighbor element, respectively. We discretize the velocity \mathbf{v}_h explicitly but its gradient has to be taken implicitly, in order to avoid additional restrictions on the maximum time step given by the viscous terms. In viscosity dominated problems, this allows us to use both high viscosity and large time steps. After integration of the first term of (3.34) by parts in time the resulting fully discrete formulation of (3.34) becomes

$$\hat{\mathbf{v}}_j^{n+1, k+\frac{1}{2}} = \mathbf{M}_j^{-1} \mathbf{M}_j^- \hat{\mathbf{v}}_j^n - \mathbf{M}_j^{-1} \Upsilon_j(\mathbf{v}_h^{n+1, k}, \nabla \mathbf{v}_h^{n+1, k+\frac{1}{2}}) + \mathbf{M}_j^{-1} \mathcal{S}_j, \quad (3.37)$$

where

$$\begin{aligned} \Upsilon_j(\mathbf{v}_h, \nabla \mathbf{v}_h) &= \int_{\mathbf{R}_j^{st}} \tilde{\psi}_k^{(j)} \nabla \cdot \mathbf{F}(\mathbf{v}_h, \nabla \mathbf{v}_h) dx \\ &= \int_{\partial \mathbf{R}_j^{st}} \tilde{\psi}_k \mathbf{G}_h \cdot \vec{n} ds - \int_{\mathbf{R}_j^{st}} \nabla \tilde{\psi}_k \cdot \mathbf{F}(\mathbf{v}_h, \nabla \mathbf{v}_h) dx. \end{aligned} \quad (3.38)$$

Due to the explicit treatment of the nonlinear convective terms, the above method requires that the time step size is restricted by a CFL-type restriction for DG schemes, namely:

$$\Delta t = \frac{\text{CFL}}{2p+1} \cdot \frac{h_{\min}}{2|\mathbf{v}_{\max}|}, \quad (3.39)$$

where h_{min} is the smallest incircle diameter; $CFL < 0.5$; and \mathbf{v}_{max} is the maximum convective speed. Furthermore, the time step of the global semi-implicit scheme is *not* affected by the local time step used for the time integration of the convective terms if a local time stepping / subcycling approach is employed, see [42, 145]. Implicit discretization of the viscous contribution $\nabla \mathbf{v}$ in (3.34) involves two five-point block systems (one for each velocity component) that can be efficiently solved using a matrix-free GMRES algorithm [138]. The solution of this system is not necessary in problems where the viscous term is small enough to be integrated explicitly in time. In that case, i.e. for explicit discretizations of the viscous terms, one has to include the additional explicit time step restriction for parabolic PDE in eq. (3.39).

Once $\mathbf{v}_j^{n+1, k+\frac{1}{2}}$ has been computed, we set $\widehat{\mathbf{F}\bar{\mathbf{v}}}_j^{n+1, k+\frac{1}{2}} := \hat{\mathbf{v}}_j^{n+1, k+\frac{1}{2}}$. As initial guess $\hat{\mathbf{v}}_j^{n+1, 0}$ we can take the old velocity \mathbf{v}_h^n , or the extrapolation of \mathbf{v}_h^n into the interval T^{n+1} .

3.2.5 Pressure correction formulation and final algorithm

The preliminary algorithm described above, as formulated by Eqs. (3.37), (3.33) and (3.30) still contains an important drawback: indeed, Eq. (3.37) does not depend on the pressure of the previous Picard iteration and hence the algorithm does not see the effect of the pressure in the time interval T^{n+1} . In order to overcome the problem we introduce the contribution of the pressure from the previous Picard iteration directly into Eq. (3.37). Then, we update the velocity with the new pressure $\hat{\mathbf{p}}_i^{n+1, k+1}$. With this modification, Eqs. (3.37), (3.33), (3.30) and hence the final algorithm become:

$$\begin{aligned} \hat{\mathbf{v}}_j^{n+1, k+\frac{1}{2}} &= \mathbf{M}_j^{-1} \mathbf{M}_j^- \hat{\bar{\mathbf{v}}}_j^n - \mathbf{M}_j^{-1} \Upsilon_j(\mathbf{v}_h^{n+1, k}, \nabla \mathbf{v}_h^{n+1, k+\frac{1}{2}}) \\ &\quad - \mathcal{Q}_{r(j), j} \hat{\mathbf{p}}_{r(j)}^{n+1, k} - \mathcal{Q}_{\ell(j), j} \hat{\mathbf{p}}_{\ell(j)}^{n+1, k} + \mathbf{M}_j^{-1} \mathcal{S}_j, \end{aligned} \quad (3.40)$$

$$\begin{aligned} &\sum_{j \in S_i} \mathcal{D}_{i, j} \mathbf{M}_j^{-1} \mathcal{Q}_{i, j} \left(\hat{\mathbf{p}}_i^{n+1, k+1} - \hat{\mathbf{p}}_i^{n+1, k} \right) \\ &+ \sum_{j \in S_i} \mathcal{D}_{i, j} \mathbf{M}_j^{-1} \mathcal{Q}_{\varphi(i, j), j} \left(\hat{\mathbf{p}}_{\varphi(i, j)}^{n+1, k+1} - \hat{\mathbf{p}}_{\varphi(i, j)}^{n+1, k} \right) \\ &= \sum_{j \in S_i} \mathcal{D}_{i, j} \widehat{\mathbf{F}\bar{\mathbf{v}}}_j^{n+1, k+\frac{1}{2}}, \end{aligned} \quad (3.41)$$

$$\begin{aligned} \hat{\mathbf{v}}_j^{n+1,k+1} &= \widehat{\mathbf{F}}\hat{\mathbf{v}}_j^{n+1,k+\frac{1}{2}} - \mathbf{M}_j^{-1} \left[\mathcal{Q}_{r(j),j} \left(\hat{\mathbf{p}}_{r(j)}^{n+1,k+1} - \hat{\mathbf{p}}_{r(j)}^{n+1,k} \right) \right. \\ &\quad \left. - \mathcal{Q}_{\ell(j),j} \left(\hat{\mathbf{p}}_{\ell(j)}^{n+1,k+1} - \hat{\mathbf{p}}_{\ell(j)}^{n+1,k} \right) \right]. \end{aligned} \quad (3.42)$$

Note that Eqs. (3.41) and (3.42) are written in terms of the *pressure correction* $\Delta \hat{\mathbf{p}}_i^{n+1,k+1} = \left(\hat{\mathbf{p}}_i^{n+1,k+1} - \hat{\mathbf{p}}_i^{n+1,k} \right)$. Since both space and time are involved in this algorithm and since space and time discretization are intrinsically coupled, we call this method a coupled space-time pressure correction algorithm. The resulting linear system for the pressure correction is very sparse thanks to the use of the *staggered grid*, including only four non-zero blocks per element. Note that the same algorithm on a *collocated grid* would lead to a pressure system with **10** non-zero blocks per element, since substituting the discrete velocity into the continuity equation on a collocated grid involves also neighbors of neighbors. The significantly improved sparsity pattern of the linear system is indeed a key advantage of the algorithm presented in this paper. In all subsequent numerical examples, we were able to solve the system (3.41) using a simple matrix-free version of the GMRES algorithm [138] without using any preconditioner, which is a unique property in the context of high order implicit DG schemes.

As initial guess for the pressure one can take $\mathbf{p}_h^{n+1,0} = 0$, but one could also choose the extrapolation of \mathbf{p}_h^n into T^{n+1} . One time step of the final algorithm can be summarized as follows:

1. Initialize $\mathbf{v}_h^{n+1,0}$ and $\mathbf{p}_h^{n+1,0}$ using the known information from T^n ;
2. Picard iteration over $k = 0 \dots N_{pic}$:
 - a) compute $\mathbf{v}_h^{n+1,k+\frac{1}{2}}$ using (3.40), i.e. convective terms are discretized explicitly and viscous terms implicitly;
then set $\widehat{\mathbf{F}}\hat{\mathbf{v}}_j^{n+1,k+\frac{1}{2}} := \hat{\mathbf{v}}_j^{n+1,k+\frac{1}{2}}$,
 - b) compute $\hat{\mathbf{p}}^{n+1,k+1}$ by solving the discrete pressure Poisson equation (3.41),
 - c) update $\hat{\mathbf{v}}_j^{n+1,k+1}$ explicitly from (3.42);
3. set $\hat{\mathbf{v}}_j^{n+1} = \hat{\mathbf{v}}_j^{n+1,k+1}$ and $\hat{\mathbf{p}}^{n+1} = \hat{\mathbf{p}}^{n+1,k+1}$.

3.3 The particular case $p_\gamma = 0$ using a simple Crank-Nicolson time discretization

In the particular case of $p_\gamma = 0$ the method described above reduces to a high order in space and low order in time scheme, but we can recover several good properties about the main system for the discrete pressure. Furthermore, in this case we can recover some precision in time by introducing a semi-implicit discretization using the theta method on the pressure inside the momentum equations. First of all we remark how, for $p_\gamma = 0$, $\frac{\partial \hat{\phi}}{\partial t} = \frac{\partial \hat{\psi}}{\partial t} = 0$, since $\tilde{\phi}(x, y, t) = \phi(x, y)$ and $\tilde{\psi}(x, y, t) = \psi(x, y)$ are constant in time. In addition, we need only a Picard iteration; since the pressure p^k can be neglected for this first Picard iteration, then the entire Picard procedure and pressure correction algorithm can be skipped in this particular case. For simplicity we take also $S \equiv 0$ in this section. The resulting weak formulation of (3.2)-(3.3) reads

$$\begin{cases} \sum_{j \in \mathcal{S}_i} \mathcal{D}_{i,j} \hat{\mathbf{v}}_j^{n+1} = 0, \\ \mathbf{M}_j \left(\hat{\mathbf{v}}_j^{n+1} - \widehat{\mathbf{F}} \hat{\mathbf{v}}_j^n \right) + \Delta t \mathcal{Q}_{r(j),j} \hat{\mathbf{p}}_{r(j)}^{n+\theta} + \Delta t \mathcal{Q}_{\ell(j),j} \hat{\mathbf{p}}_{\ell(j)}^{n+\theta} = 0, \end{cases} \quad (3.43)$$

where $\hat{\mathbf{p}}^{n+\theta} = \theta \hat{\mathbf{p}}^{n+1} + (1 - \theta) \hat{\mathbf{p}}^n$; and θ is an implicitness factor to be taken in the range $\theta \in [\frac{1}{2}, 1]$, see e.g. [36], and

$$\mathbf{M}_j = \int_{\mathbf{R}_j} \psi_k^{(j)} \psi_l^{(j)} d\mathbf{x}, \quad (3.44)$$

$$\mathcal{D}_{i,j} = \int_{\Gamma_j} \phi_k^{(i)} \psi_l^{(j)} \vec{n}_{ij} ds - \int_{\mathbf{T}_{i,j}} \nabla \phi_k^{(i)} \psi_l^{(j)} d\mathbf{x}, \quad (3.45)$$

$$\mathcal{Q}_{i,j} = \int_{\mathbf{T}_{i,j}} \psi_k^{(j)} \nabla \phi_l^{(i)} d\mathbf{x} - \int_{\Gamma_j} \psi_k^{(j)} \phi_l^{(i)} \sigma_{i,j} \vec{n}_j ds, \quad (3.46)$$

are computed from space only coordinates. It is important to emphasize that we cannot take the velocity field $\hat{\mathbf{v}}$ semi-implicitly in the discrete continuity equation following the same philosophy of Chapter 2 since the divergence free condition is a property that we require at any time step. As example, if we start from a velocity field that is not divergence free, then a semi-implicit discretization of the velocity in the continuity equation would produce a violation

3.3 The particular case $p_\gamma = 0$ using a simple Crank-Nicolson time discretization

also at the successive time. On the contrary, the divergence free condition becomes satisfied from the second time step on. The resulting final system for the pressure can be obtained by formal substitution of the velocity vector field in the momentum equation into the continuity one and reads:

$$-\theta\Delta t \sum_{j \in S_i} \mathcal{D}_{i,j} M_j^{-1} \mathcal{Q}_{i,j} \hat{\mathbf{p}}_i^{n+1} - \theta\Delta t \sum_{j \in S_i} \mathcal{D}_{i,j} M_j^{-1} \mathcal{Q}_{\varphi(i,j),j} \hat{\mathbf{p}}_{\varphi(i,j)}^{n+1} = \mathbf{b}_i^n, \quad (3.47)$$

where

$$\mathbf{b}_i^n = - \sum_{j \in S_i} \mathcal{D}_{i,j} \widehat{\mathbf{F}} \vec{\mathbf{v}}_j^n + (1-\theta)\Delta t \sum_{j \in S_i} \mathcal{D}_{i,j} (M_j)^{-1} (\mathcal{Q}_{i,j} \hat{\mathbf{p}}_i^n + \mathcal{Q}_{\varphi(i,j),j} \hat{\mathbf{p}}_{\varphi(i,j),j}^n), \quad (3.48)$$

groups all the known terms at time t^n .

It remains to complete the system by introducing the boundary conditions. Using the same reasoning of Section 2.2.6, equations (3.47)-(3.48) are computed with the triangular boundary elements and reads

$$\theta\Delta t \left[- \sum_{j \in S_i \cap \mathcal{B}(\Omega)} \mathcal{D}_{i,j}^\partial M_j^{-1} \mathcal{Q}_{i,j}^\partial - \sum_{j \in S_i - \mathcal{B}(\Omega)} \mathcal{D}_{i,j} M_j^{-1} \mathcal{Q}_{i,j} \right] \hat{\mathbf{p}}_i^{n+1} - \theta\Delta t \sum_{j \in S_i - \mathcal{B}(\Omega)} \mathcal{D}_{i,j} M_j^{-1} \mathcal{Q}_{\varphi(i,j),j} \hat{\mathbf{p}}_{\varphi(i,j)}^{n+1} = \tilde{\mathbf{b}}_i^n, \quad (3.49)$$

where now the vector of known terms is

$$\begin{aligned} \tilde{\mathbf{b}}_i^n &= - \sum_{j \in S_i - \mathcal{B}(\Omega)} \mathcal{D}_{i,j} \widehat{\mathbf{F}} \vec{\mathbf{v}}_j^n + \sum_{j \in S_i \cap \mathcal{B}(\Omega)} \mathcal{D}_{i,j}^\partial \widehat{\mathbf{F}} \vec{\mathbf{v}}_j^n \\ &+ (1-\theta)\Delta t \sum_{j \in S_i \cap \mathcal{B}(\Omega)} \mathcal{D}_{i,j}^\partial M_j^{-1} \mathcal{Q}_{i,j}^\partial \hat{\mathbf{p}}_i^n \\ &+ (1-\theta)\Delta t \sum_{j \in S_i - \mathcal{B}(\Omega)} \mathcal{D}_{i,j} M_j^{-1} (\mathcal{Q}_{i,j} \hat{\mathbf{p}}_i^n + \mathcal{Q}_{\varphi(i,j),j} \hat{\mathbf{p}}_{\varphi(i,j),j}^n). \end{aligned} \quad (3.50)$$

As we will show later, the system is symmetric and positive definite for appropriate boundary conditions, hence it can be efficiently solved by using a matrix-free implementation of the conjugate gradient algorithm [98]. Once the new pressure has been computed, the new velocity field can be readily updated

for every $j \notin \mathcal{B}(\Omega)$:

$$\begin{aligned} \hat{\mathbf{v}}_j^{n+1} &= \widehat{\mathbf{F}}\hat{\mathbf{v}}_j^n - \theta\Delta t \mathbf{M}_j^{-1} \left(\mathcal{Q}_{i,j}\hat{\mathbf{p}}_i^{n+1} + \mathcal{Q}_{\varphi(i,j),j}\hat{\mathbf{p}}_{\varphi(i,j)}^{n+1} \right) \\ &\quad - (1-\theta)\Delta t \mathbf{M}_j^{-1} \left(\mathcal{Q}_{i,j}\hat{\mathbf{p}}_i^n + \mathcal{Q}_{\varphi(i,j),j}\hat{\mathbf{p}}_{\varphi(i,j)}^n \right). \end{aligned} \quad (3.51)$$

The above equations (3.49),(3.50) and (3.51) can be modified for $j \in \mathcal{B}(\Omega)$ according to the type of boundary conditions (velocity or pressure boundary condition).

3.3.1 Remarks on the main system and further improvements

In this section we will show how the main system for the computation of the pressure such as described in Section 3.3 results symmetric and, in general, positive semi-definite. These results allow us to use very fast iterative methods to solve the linear system, such as the conjugate gradient method [98], with a significant gain in terms of computational time. In order to do this observe how, from the definitions (3.45) and (3.46), we can further generalize the action of \mathcal{D} in terms of \mathcal{Q} such as $\mathcal{D} = -\mathcal{Q}^\top$ since

$$\begin{aligned} -\mathcal{Q}_{i,j}^\top &= -\left(\int_{\Omega_{i,j}} \psi_k^{(j)} \nabla \phi_l^{(i)} d\mathbf{x} - \int_{\Gamma_j} \psi_k^{(j)} \phi_l^{(i)} \sigma_{i,j} \bar{\mathbf{n}}_j ds \right)^\top \\ &= -\int_{\Omega_{i,j}} \psi_l^{(j)} \nabla \phi_k^{(i)} d\mathbf{x} + \int_{\Gamma_j} \psi_l^{(j)} \phi_k^{(i)} \sigma_{i,j} \bar{\mathbf{n}}_j ds = \mathcal{D}_{i,j} \end{aligned} \quad (3.52)$$

and if $i = \ell(j)$, $\bar{\mathbf{n}}_{i,j}$ coincides with $\bar{\mathbf{n}}_j$, else, it is $-\bar{\mathbf{n}}_j$, $\forall i, j \in S_i$. Consequently, the main system (3.47) can be written as

$$\mathcal{A} : \theta\Delta t \sum_{j \in S_i} \mathcal{Q}_{i,j}^\top (\mathbf{M}_j)^{-1} \mathcal{Q}_{i,j} \hat{\mathbf{p}}_i^{n+1} + \theta\Delta t \sum_{j \in S_i} \mathcal{Q}_{i,j}^\top (\mathbf{M}_j)^{-1} \mathcal{Q}_{\varphi(i,j),j} \hat{\mathbf{p}}_{\varphi(i,j)}^{n+1} = \mathbf{b}_i^n,$$

that we will call in the following \mathcal{A} . If we do not introduce any boundary conditions, we have the following

Lemma 1 *Without any boundary conditions the system \mathcal{A} is singular.*

Proof 1 *Let $p_h \in V_h^m$, in order to show that \mathcal{A} is singular we investigate the kernel of the linear operator \mathcal{A} . Since $\det \mathcal{A} \neq 0 \Leftrightarrow \text{Ker} \mathcal{A} = \{0\}$, we would like*

3.3 The particular case $p_\gamma = 0$ using a simple Crank-Nicolson time discretization

to show that the kernel does not contain only the zero. A weak formulation of ∇p_h over Ω_j is given by $\mathcal{Q}_{\ell(j),j}\hat{p}_{\ell(j)} + \mathcal{Q}_{r(j),j}\hat{p}_{r(j)}$, then we have the identity

$$\mathcal{Q}_{\ell(j),j}\hat{p}_{\ell(j)} + \mathcal{Q}_{r(j),j}\hat{p}_{r(j)} \equiv 0 \Leftrightarrow \nabla p|_{\Omega_j} = 0 \quad (3.53)$$

We are looking for a $p_h \neq 0$ such that $\mathcal{A}p_h = 0$. For a fixed $i \in [1, N_e]$,

$$\begin{aligned} -\theta\Delta t \sum_{j \in S_i} \mathcal{D}_{i,j}(\mathbf{M}_j)^{-1} \mathcal{Q}_{i,j}\hat{p}_i^{n+1} - \theta\Delta t \sum_{j \in S_i} \mathcal{D}_{i,j}(\mathbf{M}_j)^{-1} \mathcal{Q}_{\varphi(i,j),j}\hat{p}_{\varphi(i,j)}^{n+1} &= 0 \\ -\theta\Delta t \sum_{j \in S_i} \mathcal{D}_{i,j}(\mathbf{M}_j)^{-1} \left[\mathcal{Q}_{i,j}\hat{p}_i^{n+1} + \mathcal{Q}_{\varphi(i,j),j}\hat{p}_{\varphi(i,j)}^{n+1} \right] &= 0 \\ -\theta\Delta t \sum_{j \in S_i} \mathcal{D}_{i,j}(\mathbf{M}_j)^{-1} \left[\mathcal{Q}_{\ell(j),j}\hat{p}_{\ell(j)}^{n+1} + \mathcal{Q}_{r(j),j}\hat{p}_{r(j)}^{n+1} \right] &= 0 \end{aligned} \quad (3.54)$$

Hence, if $p = \text{constant}$, the left side of (3.54) vanishes and then $\{p_i \equiv c \ \forall i, c \in \mathbb{R}\} \subset \ker \mathcal{A}$.

This represents a natural result since the incompressible Navier-Stokes equations depend only on the gradient of the pressure and not directly on the pressure. Once we have an exact solution for the pressure p_e , then every solution of the kind $p_e + c$ with $c \in \mathbb{R}$ is also a solution. If we introduce the boundary conditions and we specify the pressure in at least one point (i.e. in at least one degree of freedom), this is equivalent to choose the constant c and the system becomes non-singular. The following results state that the developed system has several important properties such as the symmetry and, in general, positive semi-definiteness:

Lemma 2 (Symmetry) *The system matrix of \mathcal{A} is symmetric.*

Proof 2 *In the following we denote with (i, k) the k -th degree of freedom of the i -th element. For the symmetry of \mathcal{A} we have to verify that (i, k) acts on (\tilde{i}, \tilde{k}) as (\tilde{i}, \tilde{k}) acts on (i, k) . If $i = \tilde{i}$, the action is described by $\sum_{j \in S_i} \mathcal{Q}_{i,j}^\top (\mathbf{M}_j)^{-1} \mathcal{Q}_{i,j}$ that is trivially symmetric since $\mathbf{M}_j = \mathbf{M}_j^\top$ is symmetric. If $\tilde{i} \notin \varphi(i, S_i)$ the two actions are zero so it is also trivially verified. Remains the case $\tilde{i} \in \varphi(i, S_i)$. In this case, the actions of the right element on the left one and vice versa are, respectively, $\mathcal{Q}_{\ell(j),j}^\top \mathbf{M}_j^{-1} \mathcal{Q}_{r(j),j}$ and $\mathcal{Q}_{r(j),j}^\top \mathbf{M}_j^{-1} \mathcal{Q}_{\ell(j),j}$. A simple computation leads to*

$$\mathbf{M}_j^{-1} \mathcal{Q}_{r(j),j}(k, l) = \sum_{\xi=1}^{N_\psi} \mathbf{M}_j^{-1}(k, \xi) \mathcal{Q}_{r(j),j}(\xi, l) \quad \forall k = 1 \dots N_\psi, l = 1 \dots N_\phi$$

and then $\forall k = 1 \dots N_\phi$, $l = 1 \dots N_\phi$,

$$\begin{aligned}
 \mathbf{Q}_{\ell(j),j}^\top M_j^{-1} \mathbf{Q}_{r(j),j}(k,l) &= \sum_{\gamma=1}^{N_\psi} \mathbf{Q}_{\ell(j),j}^\top(k,\gamma) \left(M_j^{-1} \mathbf{Q}_{r(j),j} \right) (\gamma,l) \\
 &= \sum_{\gamma=1}^{N_\psi} \mathbf{Q}_{\ell(j),j}^\top(k,\gamma) \sum_{\xi=1}^{N_\psi} M_j^{-1}(\gamma,\xi) \mathbf{Q}_{r(j),j}(\xi,l) \\
 &= \sum_{\gamma,\xi=1}^{N_\psi} \mathbf{Q}_{\ell(j),j}^\top(k,\gamma) M_j^{-1}(\gamma,\xi) \mathbf{Q}_{r(j),j}(\xi,l) \\
 &= \sum_{\gamma,\xi=1}^{N_\psi} \mathbf{Q}_{\ell(j),j}(\gamma,k) M_j^{-1}(\gamma,\xi) \mathbf{Q}_{r(j),j}^\top(l,\xi) \\
 &= \sum_{\gamma,\xi=1}^{N_\psi} \mathbf{Q}_{r(j),j}^\top(l,\xi) M_j^{-1}(\xi,\gamma) \mathbf{Q}_{\ell(j),j}(\gamma,k) \\
 &= \mathbf{Q}_{r(j),j}^\top M_j^{-1} \mathbf{Q}_{\ell(j),j}(k,l) \tag{3.55}
 \end{aligned}$$

Lemma 3 *The matrix \mathcal{A} is positive semi-definite, i.e. $x^\top \mathcal{A} x \geq 0 \ \forall x \in \mathbb{R}^{N_e \cdot N_\phi}$.*

Proof 3 *We do the computation directly. $x^\top \mathcal{A} x = \sum_i (x^\top \mathcal{A} x)_i$ and*

$$\begin{aligned}
 (x^\top \mathcal{A} x)_i &= x_i \sum_{j \in S_i} \mathbf{Q}_{i,j}^\top M_j^{-1} \mathbf{Q}_{i,j} x_i + x_i \sum_{j \in S_i} \mathbf{Q}_{i,j}^\top M_j^{-1} \mathbf{Q}_{\varphi(i,j),j} x_{\varphi(i,j)} \\
 &= \sum_{j \in S_i} \left(M_j^{-\frac{1}{2}} \mathbf{Q}_{i,j} x_i \right)^\top \left(M_j^{-\frac{1}{2}} \mathbf{Q}_{i,j} x_i \right) \\
 &\quad + \sum_{j \in S_i} \left(M_j^{-\frac{1}{2}} \mathbf{Q}_{i,j} x_i \right)^\top \left(M_j^{-\frac{1}{2}} \mathbf{Q}_{\varphi(i,j),j} x_{\varphi(i,j)} \right)
 \end{aligned}$$

where we used that M_j is symmetric and positive definite, hence M_j^{-1} is symmetric and positive definite and then exists the so called square operator, namely $\exists M_j^{-\frac{1}{2}}$ such that $M_j^{-1} = \left(M_j^{-\frac{1}{2}} \right)^\top \left(M_j^{-\frac{1}{2}} \right)$. By defining $T_{i,j} := M_j^{-\frac{1}{2}} \mathbf{Q}_{i,j}$

3.3 The particular case $p_\gamma = 0$ using a simple Crank-Nicolson time discretization

we obtain

$$(x^\top Ax)_i = \sum_{j \in S_i} (T_{i,j} x_i)^\top (T_{i,j} x_i) + \sum_{j \in S_i} (T_{i,j} x_i)^\top (T_{\wp(i,j),j} x_{\wp(i,j)}) \quad (3.56)$$

and consequently

$$x^\top Ax = \sum_{i=1}^{N_e} \sum_{j \in S_i} (T_{i,j} x_i)^\top (T_{i,j} x_i) + \sum_{i=1}^{N_e} \sum_{j \in S_i} (T_{i,j} x_i)^\top (T_{\wp(i,j),j} x_{\wp(i,j)}) \quad (3.57)$$

Remark that the double summation $\sum_{i=1}^{N_e} \sum_{j \in S_i}$ sum every element i and edge j . From the edge point of view, every edge gives two contributions, one given when $i = \ell(j)$ and one when $i = r(j)$. The double summation can be consequently inverted as follows:

$$\begin{aligned} \sum_{i=1}^{N_e} \sum_{j \in S_i} (T_{i,j} x_i)^\top (T_{i,j} x_i) &= \sum_{j=1}^{N_d} (T_{\ell(j),j} x_{\ell(j)})^\top (T_{\ell(j),j} x_{\ell(j)}) \\ &\quad + \sum_{j=1}^{N_d} (T_{r(j),j} x_{r(j)})^\top (T_{r(j),j} x_{r(j)}) \\ \sum_{i=1}^{N_e} \sum_{j \in S_i} (T_{i,j} x_i)^\top (T_{\wp(i,j),j} x_{\wp(i,j)}) &= \sum_{j=1}^{N_d} (T_{\ell(j),j} x_{\ell(j)})^\top (T_{r(j),j} x_{r(j)}) \\ &\quad + \sum_{j=1}^{N_d} (T_{r(j),j} x_{r(j)})^\top (T_{\ell(j),j} x_{\ell(j)}) \end{aligned} \quad (3.58)$$

and then, by recomposing everything

$$\begin{aligned}
 x^\top Ax &= \sum_{j=1}^{N_d} \left[(T_{\ell(j),j} x_{\ell(j)})^\top (T_{\ell(j),j} x_{\ell(j)}) + (T_{r(j),j} x_{r(j)})^\top (T_{r(j),j} x_{r(j)}) \right. \\
 &\quad \left. (T_{\ell(j),j} x_{\ell(j)})^\top (T_{r(j),j} x_{r(j)}) + (T_{r(j),j} x_{r(j)})^\top (T_{\ell(j),j} x_{\ell(j)}) \right] \\
 &= \sum_{j=1}^{N_d} (T_{\ell(j),j} x_{\ell(j)} + T_{r(j),j} x_{r(j)})^\top (T_{\ell(j),j} x_{\ell(j)} + T_{r(j),j} x_{r(j)}) \\
 &= \sum_{j=1}^{N_d} \left[\left(\begin{array}{cc} T_{\ell(j),j} & 0 \\ 0 & T_{r(j),j} \end{array} \right) \cdot \begin{pmatrix} x_{\ell(j)} \\ x_{r(j)} \end{pmatrix} \right]^\top \\
 &\quad \cdot \left[\left(\begin{array}{cc} T_{\ell(j),j} & 0 \\ 0 & T_{r(j),j} \end{array} \right) \cdot \begin{pmatrix} x_{\ell(j)} \\ x_{r(j)} \end{pmatrix} \right] \\
 &= \sum_{j=1}^{N_d} (x_{\ell(j)}, x_{r(j)}) \begin{pmatrix} T_{\ell(j),j} & 0 \\ 0 & T_{r(j),j} \end{pmatrix}^\top \\
 &\quad \cdot \begin{pmatrix} T_{\ell(j),j} & 0 \\ 0 & T_{r(j),j} \end{pmatrix} \begin{pmatrix} x_{\ell(j)} \\ x_{r(j)} \end{pmatrix} \\
 &= \sum_{j=1}^{N_d} \tilde{x}_j^\top \mathcal{T}^\top \mathcal{T} \tilde{x}_j \tag{3.59}
 \end{aligned}$$

And, since $\tilde{\mathcal{T}} := \mathcal{T}^\top \mathcal{T}$ is a positive semi-definite matrix by construction, $\tilde{x}_j^\top \tilde{\mathcal{T}} \tilde{x}_j \geq 0$ and then $x^\top Ax = \sum_j \tilde{x}_j^\top \tilde{\mathcal{T}} \tilde{x}_j \geq 0$.

We introduce now the boundary elements and, in particular,

$$\mathcal{D}_{i,j}^\partial = \int_{\Gamma_j} \phi_k^{(i)} \psi_l^{\partial(j)} \vec{n}_{ij} ds - \int_{\mathbf{T}_{i,j}} \nabla \phi_k^{(i)} \psi_l^{\partial(j)} d\mathbf{x}$$

and

$$\mathcal{Q}_{i,j}^\partial = \int_{\mathbf{T}_{i,j}} \psi_k^{\partial(j)} \nabla \phi_l^{(i)} d\mathbf{x} - \int_{\Gamma_j} \psi_k^{\partial(j)} \phi_l^{(i)} \sigma_{i,j} \vec{n}_j ds.$$

Then it is still true that $\mathcal{D}_{i,j}^\partial = -(\mathcal{Q}_{i,j}^\partial)^\top$ and the complete system $\tilde{\mathcal{A}}$ can be written as $\tilde{\mathcal{A}} = \mathcal{A} + \mathcal{B}$ where

$$\begin{aligned} \mathcal{B} : & \quad \theta \Delta t \sum_{j \in S_i \cap \mathcal{B}(\Omega)} \left(\mathcal{Q}_{i,j}^\partial \right)^\top M_j^{-1} \mathcal{Q}_{i,j}^\partial \hat{\mathbf{p}}_i^{n+1} \\ \tilde{\mathcal{A}} : & \quad \theta \Delta t \left[\sum_{j \in S_i - \mathcal{B}(\Omega)} \mathcal{Q}_{i,j}^\top M_j^{-1} \mathcal{Q}_{i,j} \right] \hat{\mathbf{p}}_i^{n+1} \\ & \quad + \theta \Delta t \sum_{j \in S_i - \mathcal{B}(\Omega)} \mathcal{Q}_{i,j}^\top M_j^{-1} \mathcal{Q}_{\varphi(i,j),j} \hat{\mathbf{p}}_{\varphi(i,j)}^{n+1} \end{aligned}$$

It is easy to check that \mathcal{B} is symmetric and at least positive semi-definite. We have to introduce now some types of boundary conditions in order to show that, if the pressure is specified on the boundary, the complete system $\tilde{\mathcal{A}}$ is positive definite.

Let us rewrite $x^\top \mathcal{B}x$ by including the external contribution and in the form of the Eq. (3.59), namely

$$x^\top \mathcal{B}x = \sum_{j=1}^{N_d} \left(T_{\ell(j),j}^\partial x_{\ell(j)} + [T^\partial x]_{ext,j} \right)^\top \left(T_{\ell(j),j}^\partial x_{\ell(j)} + [T^\partial x]_{ext,j} \right)$$

where $T_{i,j}^\partial = M_j^{-\frac{1}{2}} \mathcal{Q}_{i,j}^\partial$ and $[T^\partial x]_{ext,j}$ is a known external contribution that depends on the boundary conditions. In particular, if the pressure is specified at the boundary, then $T_{ext,j}^\partial = T_{\ell(j),j}^\partial$ and $[T^\partial x]_{ext,j}$ is a known quantity that in general is part of the known right hand side vector. Since the external pressure is specified, then $T_{\ell(j),j}^\partial x_{\ell(j)} + [T^\partial x]_{ext,j} = 0 \Leftrightarrow x_{\ell(j)} \equiv x_{ext,j}$. We take now $x^\top \mathcal{B}x = 0$ that implicitly fixes $x_{ext} = 0$. In this way $x_{\ell(j)} = 0 \forall j \in \mathcal{B}(\Omega)$. Using the same reasoning on the matrix \mathcal{A} we can conclude that $x \equiv 0$, and hence $\tilde{\mathcal{A}}$ is positive definite in this case. A possible way to specify the velocity at the boundary is to neglect the jump contribution for the pressure at the boundary or equivalent, taken $x_{ext,j} = x_{\ell(j)} \forall j \in \mathcal{B}(\Omega)$. It is easy to check that if we have only this type of boundary conditions then $x^\top \tilde{\mathcal{A}}x = 0$ for every x constant, and then the matrix $\tilde{\mathcal{A}}$ is only positive semi-definite.

3.3.2 Remarks on the stability

In this section we discuss the stability of the proposed method. In particular we will show that if the discretization of the nonlinear convective-viscous term is stable, then the discrete solution of the velocity is L_2 stable. Let $\mathbf{v}_h^\dagger =$

$(u_h, v_h) \in (V_h^d)^2$ and $p_h \in V_h^m$ be the numerical solution for velocity and pressure. Then the L_2 stability property for the velocity reads

$$\frac{d}{dt} \int_{\Omega} \frac{1}{2} (u_h^2 + v_h^2) \, d\mathbf{x} \leq 0. \quad (3.60)$$

By construction, the numerical solution satisfies the weak formulation of the continuity and momentum equation, namely:

$$\begin{aligned} \int_{\mathbf{R}_j} \psi_k^{(j)} \frac{\partial}{\partial t} \mathbf{v}_h \, d\mathbf{x} + \int_{\partial \mathbf{R}_j} \psi_k^{(j)} \mathbf{G}_h \cdot \vec{n} \, ds - \int_{\mathbf{R}_j} \nabla \psi_k^{(j)} \cdot \mathbf{F}(\mathbf{v}_h, \nabla \mathbf{v}_h) \, d\mathbf{x} \\ + \int_{\mathbf{T}_{\ell(j),j}} \psi_k^{(j)} \nabla p_{h,\ell(j)} \, d\mathbf{x} + \int_{\mathbf{T}_{r(j),j}} \psi_k^{(j)} \nabla p_{h,r(j)} \, d\mathbf{x} \\ + \int_{\Gamma_j} \psi_k^{(j)} (p_{h,r(j)} - p_{h,\ell(j)}) \vec{n}_{std} \, ds = 0 \end{aligned} \quad (3.61)$$

and

$$\sum_{j \in S_i} \left[\int_{\Gamma_j} \phi_k^{(i)} \mathbf{v}_h \cdot \vec{n}_{e,j} \, ds - \int_{\mathbf{T}_{i,j}} \nabla \phi_k^{(i)} \cdot \mathbf{v}_h \, d\mathbf{x} \right] = 0 \quad (3.62)$$

for every $i = 1 \dots N_e$ and $j = 1 \dots N_d$. Since $\{\phi_k^{(i)}\}_k$ forms a basis for the polynomial space $V_h^m|_{\mathbf{T}_i}$, we can take p_h as a test function in (3.62). Denoting with $p_{h,i}$ the discrete solution of the pressure inside element number i we directly obtain from (3.62)

$$\sum_{j \in S_i} \left[\int_{\Gamma_j} p_{h,i} \mathbf{v}_h \cdot \vec{n}_{e,j} \, ds - \int_{\mathbf{T}_{i,j}} \nabla p_{h,i} \cdot \mathbf{v}_h \, d\mathbf{x} \right] = 0 \quad (3.63)$$

3.3 The particular case $p_\gamma = 0$ using a simple Crank-Nicolson time discretization

We can now sum over $i = 1 \dots N_e$ in Eq. (3.63) and find

$$\begin{aligned} \sum_{i=1}^{N_e} \sum_{j \in S_i} \left[\int_{\Gamma_j} p_{h,i} \mathbf{v}_h \cdot \vec{n}_{e,j} ds - \int_{\mathbf{T}_{i,j}} \nabla p_{h,i} \cdot \mathbf{v}_h dx \right] &= 0 \\ \sum_{j=1}^{N_d} \left[\int_{\mathbf{T}_{\ell(j),j}} \nabla p_{h,\ell(j)} \cdot \mathbf{v}_h dx + \int_{\mathbf{T}_{r(j),j}} \nabla p_{h,r(j)} \cdot \mathbf{v}_h dx \right. \\ &\quad \left. + \int_{\Gamma_j} (p_{h,r(j)} - p_{h,\ell(j)}) \mathbf{v}_h \cdot \vec{n}_{std} ds \right] = 0 \end{aligned} \quad (3.64)$$

Using the same reasoning we can take as test function $\psi_k = u_h \in V_h^d$ in the first component of Eq. (3.61) and $\psi_k = v_h \in V_h^d$ in the second one. By summing the resulting two expressions we thus obtain

$$\begin{aligned} \int_{\mathbf{R}_j} \mathbf{v}_h \cdot \frac{\partial}{\partial t} \mathbf{v}_h dx + \int_{\partial \mathbf{R}_j} \mathbf{v}_h \cdot (\mathbf{G}_h \cdot \vec{n}) ds - \int_{\mathbf{R}_j} \nabla \mathbf{v}_h : \mathbf{F}(\mathbf{v}_h, \nabla \mathbf{v}_h) dx \\ + \int_{\mathbf{T}_{\ell(j),j}} \mathbf{v}_h \cdot \nabla p_{h,\ell(j)} dx + \int_{\mathbf{T}_{r(j),j}} \mathbf{v}_h \cdot \nabla p_{h,r(j)} dx \\ + \int_{\Gamma_j} (p_{h,r(j)} - p_{h,\ell(j)}) \mathbf{v}_h \cdot \vec{n}_{std} ds = 0, \end{aligned} \quad (3.65)$$

where $c = \mathbf{A} : \mathbf{B}$ denotes the tensor operator $c = A_{ij} B_{ij}$, written at the aid of the Einstein summation convention. Using equation (3.65) we get:

$$\frac{d}{dt} \int_{\Omega} \frac{1}{2} (u_h^2 + v_h^2) dx = \int_{\Omega} \left(\mathbf{v}_h \cdot \frac{\partial}{\partial t} \mathbf{v}_h \right) dx = \sum_{j=1}^{N_d} \int_{\mathbf{R}_j} \left(\mathbf{v}_h \cdot \frac{\partial}{\partial t} \mathbf{v}_h \right) dx$$

$$\begin{aligned}
 &= \sum_{j=1}^{N_d} \left[- \int_{\partial \mathbf{R}_j} \mathbf{v}_h \cdot (\mathbf{G}_h \cdot \vec{n}) \, ds + \int_{\mathbf{R}_j} \nabla \mathbf{v}_h : \mathbf{F}(\mathbf{v}_h, \nabla \mathbf{v}_h) \, d\mathbf{x} \right] \\
 &\quad - \sum_{j=1}^{N_d} \left[\int_{\mathbf{T}_{\ell(j),j}} \mathbf{v}_h \cdot \nabla p_{h,\ell(j)} \, d\mathbf{x} + \int_{\mathbf{T}_{r(j),j}} \mathbf{v}_h \cdot \nabla p_{h,r(j)} \, d\mathbf{x} \right. \\
 &\quad \left. + \int_{\Gamma_j} (p_{h,r(j)} - p_{h,\ell(j)}) \mathbf{v}_h \cdot \vec{n}_{std} \, ds \right]
 \end{aligned} \tag{3.66}$$

where the first sum on the right hand side contains the contribution of the nonlinear convective and viscous terms and the second sum, which contains the contribution of the pressure, vanishes thanks to (3.64). As a final result, we therefore obtain:

$$\begin{aligned}
 \frac{d}{dt} \int_{\Omega} \frac{1}{2} (u_h^2 + v_h^2) \, d\mathbf{x} &= - \sum_{j=1}^{N_d} \left[\int_{\partial \mathbf{R}_j} \mathbf{v}_h \cdot (\mathbf{G}_h \cdot \vec{n}) \, ds \right. \\
 &\quad \left. - \int_{\mathbf{R}_j} \nabla \mathbf{v}_h : \mathbf{F}(\mathbf{v}_h, \nabla \mathbf{v}_h) \, d\mathbf{x} \right].
 \end{aligned} \tag{3.67}$$

As a consequence of Eq. (3.67), the stability of our staggered DG method is determined by the stability of the discretization used for the nonlinear convective and the viscous term. We note that on the right hand side of Eq. (3.67) we have a standard DG discretization of these terms on the dual mesh, for which the standard DG stability results apply. In the literature there are stability results that prove that if the numerical flux \mathbf{G}_h is an entropy flux, then the right hand side is actually less or equal zero for the convective part. More details on the stability of high order DG schemes for convective problems can be found in [108] for the scalar case and in [6] for first order systems. Furthermore, if we have no convective-viscous contributions, Eq. (3.67) states that the total kinetic energy of the system is exactly conserved.

3.4 Splitting of the space-time matrices into a spatial and temporal part

Due to the tensor product construction of the space-time basis functions, we can split the main integrals (3.17)-(3.21) and (3.25) into a spatial and a temporal part. Briefly, the space-time matrices are generated from the spatial matrices of [144], componentwise, as:

$$\mathbf{M}_j^+(k, l) = \gamma_{\ell_2(k)}(t^{n+1})\gamma_{\ell_2(k)}(t^{n+1})\mathbf{M}_j^s(\ell_1(k), \ell_1(l)), \quad (3.68)$$

$$\mathbf{M}_j^-(k, l) = \gamma_{\ell_2(k)}(t^{n+1})\gamma_{\ell_2(k)}(t^n)\mathbf{M}_j^s(\ell_1(k), \ell_1(l)), \quad (3.69)$$

$$\mathbf{M}_j^o(k, l) = \mathbf{M}_j^s(\ell_1(k), \ell_1(l))\mathcal{D}^t(\ell_2(k), \ell_2(l)), \quad (3.70)$$

$$\mathcal{D}_{i,j}(k, l) = \Delta t^{n+1}\mathcal{D}_{i,j}^s(\ell_1(k), \ell_1(l))\mathbf{M}^t(\ell_2(k), \ell_2(l)), \quad (3.71)$$

$$\mathcal{Q}_{i,j}(k, l) = \Delta t^{n+1}\mathcal{Q}_{i,j}^s(\ell_1(k), \ell_1(l))\mathbf{M}^t(\ell_2(k), \ell_2(l)), \quad (3.72)$$

where the apex s means that the matrix is the spacial one constructed in [144]; \mathcal{D}^t and \mathbf{M}^t are two time matrices defined as

$$\mathcal{D}^t(\tilde{k}, \tilde{l}) = \int_0^1 \frac{d\gamma_{\tilde{k}}(\xi)}{d\xi} \gamma_{\tilde{l}}(\xi) d\xi, \quad (3.73)$$

$$\mathbf{M}^t(\tilde{k}, \tilde{l}) = \int_0^1 \gamma_{\tilde{k}}(\xi) \gamma_{\tilde{l}}(\xi) d\xi, \quad (3.74)$$

and ℓ_1, ℓ_2 are two appropriate sorting functions defined according to the number of space and time basis functions.

Remark how the action of the matrix \mathcal{D}^t defined in (3.73) is symmetric only if $p_\gamma = 0$.

3.5 Stability analysis of the space-time DG method

In this section we present a proof regarding the stability in the space-time DG framework. In the particular case of $p_\gamma = 0$, the stability follows directly from the stability proof given in [144] and reported in Section 3.3.2. In the general case of $p_\gamma > 0$ we can derive the following result:

Theorem 1 *The proposed staggered space-time DG method (3.8) and (3.14) is L_2 stable if the discretization of the nonlinear convective and viscous terms is L_2 stable.*

Proof 4 *By construction, the numerical solution satisfies the weak formulation of the continuity and momentum equation, namely:*

$$\sum_{j \in S_i} \left(\int_{\Gamma_j^{st}} \tilde{\phi}_k^{(i)} \mathbf{v}_h \cdot \tilde{\mathbf{n}}_{ij} \, ds \, dt - \int_{\mathbf{T}_{i,j}^{st}} \nabla \tilde{\phi}_k^{(i)} \cdot \mathbf{v}_h \, d\mathbf{x} \, dt \right) = 0, \quad (3.75)$$

and

$$\begin{aligned} & \left[\int_{\mathbf{R}_j} \tilde{\psi}_k^{(j)} \mathbf{v}_h \, d\mathbf{x} \right]_{t=t^{n+1}} - \left[\int_{\mathbf{R}_j} \tilde{\psi}_k^{(j)} \mathbf{v}_h \, d\mathbf{x} \right]_{t=t^n} - \int_{\mathbf{R}_j^{st}} \frac{\partial \tilde{\psi}_k^{(j)}}{\partial t} \mathbf{v}_h \, d\mathbf{x} \, dt \\ & + \int_{\mathbf{T}_{\ell(j),j}^{st}} \tilde{\psi}_k^{(j)} \nabla p_{\ell(j)} \, d\mathbf{x} \, dt + \int_{\mathbf{T}_{r(j),j}^{st}} \tilde{\psi}_k^{(j)} \nabla p_{r(j)} \, d\mathbf{x} \, dt \\ & + \int_{\Gamma_j^{st}} \tilde{\psi}_k^{(j)} (p_{r(j)} - p_{\ell(j)}) \tilde{\mathbf{n}}_j \, ds \, dt + \int_{\partial \mathbf{R}_j^{st}} \tilde{\psi}_k^{(j)} \mathbf{G}_h \cdot \mathbf{n} \, ds \, dt \\ & - \int_{\mathbf{R}_j^{st}} \nabla \tilde{\psi}_k^{(j)} \cdot \mathbf{F}_h \, d\mathbf{x} \, dt = \int_{\mathbf{R}_j^{st}} \tilde{\psi}_k^{(j)} \mathbf{S}_h \, d\mathbf{x} \, dt. \quad (3.76) \end{aligned}$$

In equation (3.75) we proceed as in [144] and take p_h as test function. Passing to the sum over each element \mathbf{T}_j^{st} and exchanging the two summations we obtain

$$\begin{aligned}
 & \sum_{j \in S_i} \left(\int_{\Gamma_j^{st}} p_{h,i} \mathbf{v}_h \cdot \vec{n}_{ij} \, ds \, dt - \int_{\mathbf{T}_{i,j}^{st}} \nabla p_{h,i} \cdot \mathbf{v}_h \, d\mathbf{x} \, dt \right) = 0, \\
 & \Downarrow \\
 & \sum_{i=1}^{N_e} \sum_{j \in S_i} \left(\int_{\Gamma_j^{st}} p_{h,i} \mathbf{v}_h \cdot \vec{n}_{ij} \, ds \, dt - \int_{\mathbf{T}_{i,j}^{st}} \nabla p_{h,i} \cdot \mathbf{v}_h \, d\mathbf{x} \, dt \right) = 0, \\
 & \Updownarrow \\
 & \sum_{j=1}^{N_d} \left[\int_{\mathbf{T}_{\ell(j),j}^{st}} \nabla p_{h,\ell(j)} \cdot \mathbf{v}_h \, d\mathbf{x} \, dt + \int_{\mathbf{T}_{r(j),j}^{st}} \nabla p_{h,r(j)} \cdot \mathbf{v}_h \, d\mathbf{x} \, dt \right. \\
 & \quad \left. + \int_{\Gamma_j^{st}} (p_{h,r(j)} - p_{h,\ell(j)}) \mathbf{v}_h \cdot \vec{n}_{std} \, ds \, dt \right] = 0.
 \end{aligned} \tag{3.77}$$

We now take \mathbf{v}_h as test function in (3.76) and sum up both components of the momentum equation. Furthermore, we sum up over all dual elements. Then, one obtains, thanks to (3.77), that all the pressure contributions add up to zero, as in [144]. Note that this property is also satisfied for the pressure at each Picard iteration, if the initial guess is taken according to (3.77), i.e. $p_h^0 = 0$. We can further suppose that $\mathbf{S} = 0$ and $\nu = 0$ so that $\mathbf{F}_h(\mathbf{v}_h, \nabla \mathbf{v}_h) = \mathbf{F}_h(\mathbf{v}_h)$ since the source term could increase naturally the total kinetic energy of the system and we want a stability result that is independent from the source term or the physical viscosity. From (3.76) and (3.77) and taking into account the

upwinding in time direction, we get

$$\begin{aligned}
 \sum_{j=1}^{N_d} \left(\left[\int_{\mathbf{R}_j} \mathbf{v}_h^- \cdot \mathbf{v}_h^- d\mathbf{x} \right]_{t=t^{n+1}} - \left[\int_{\mathbf{R}_j} \mathbf{v}_h^+ \cdot \mathbf{v}_h^- d\mathbf{x} \right]_{t=t^n} - \int_{\mathbf{R}_j^{st}} \frac{\partial \mathbf{v}_h}{\partial t} \cdot \mathbf{v}_h d\mathbf{x} dt \right. \\
 \left. + \int_{\partial \mathbf{R}_j^{st}} \mathbf{v}_h \cdot (\mathbf{G}_h \cdot \mathbf{n}) ds dt - \int_{\mathbf{R}_j^{st}} \nabla \mathbf{v}_h : \mathbf{F}_h d\mathbf{x} dt \right) = 0,
 \end{aligned} \tag{3.78}$$

where $\mathbf{v}_h^-|_{t=\bar{t}}$ and $\mathbf{v}_h^+|_{t=\bar{t}}$ are the left and the right value of \mathbf{v}_h at the interface according to the time normal vector $\bar{\mathbf{n}}_t$, respectively. We can now write the third integral in (3.78) as

$$\begin{aligned}
 \int_{\mathbf{R}_j^{st}} \frac{\partial \mathbf{v}_h}{\partial t} \cdot \mathbf{v}_h d\mathbf{x} dt &= \frac{1}{2} \int_{\mathbf{R}_j^{st}} \frac{\partial}{\partial t} \mathbf{v}_h^2 d\mathbf{x} dt \\
 &= \frac{1}{2} \left[\int_{\mathbf{R}_j} \mathbf{v}_h^- \cdot \mathbf{v}_h^- d\mathbf{x} \right]_{t=t^{n+1}} - \frac{1}{2} \left[\int_{\mathbf{R}_j} \mathbf{v}_h^+ \cdot \mathbf{v}_h^+ d\mathbf{x} \right]_{t=t^n}.
 \end{aligned} \tag{3.79}$$

In this way, equation (3.78) becomes

$$\begin{aligned}
 \sum_{j=1}^{N_d} \left(\left[\int_{\mathbf{R}_j} \mathbf{v}_h^- \mathbf{v}_h^- d\mathbf{x} \right]_{t=t^{n+1}} - \left[\int_{\mathbf{R}_j} \mathbf{v}_h^+ \cdot \mathbf{v}_h^- d\mathbf{x} \right]_{t=t^n} \right. \\
 - \frac{1}{2} \left[\int_{\mathbf{R}_j} \mathbf{v}_h^- \cdot \mathbf{v}_h^- d\mathbf{x} \right]_{t=t^{n+1}} + \frac{1}{2} \left[\int_{\mathbf{R}_j} \mathbf{v}_h^+ \cdot \mathbf{v}_h^+ d\mathbf{x} \right]_{t=t^n} \\
 \left. + \int_{\partial \mathbf{R}_j^{st}} \mathbf{v}_h \cdot (\mathbf{G}_h \cdot \mathbf{n}) ds dt - \int_{\mathbf{R}_j^{st}} \nabla \mathbf{v}_h : \mathbf{F}_h d\mathbf{x} dt \right) = 0.
 \end{aligned} \tag{3.80}$$

Following [72] we rearrange Eq. (3.80) by adding and subtracting the quantity $\frac{1}{2} \int \mathbf{v}_h^- \mathbf{v}_h^- d\mathbf{x} \Big|_{t=t^n}$ and thus obtain

$$\begin{aligned}
 & \sum_{j=1}^{N_d} \left(\frac{1}{2} \left[\int_{\mathbf{R}_j} \mathbf{v}_h^- \mathbf{v}_h^- d\mathbf{x} \right]_{t=t^{n+1}} - \frac{1}{2} \left[\int_{\mathbf{R}_j} \mathbf{v}_h^- \mathbf{v}_h^- d\mathbf{x} \right]_{t=t^n} \right. \\
 & \quad \left. + \frac{1}{2} \left[\int_{\mathbf{R}_j} \mathbf{v}_h^- \mathbf{v}_h^- d\mathbf{x} \right]_{t=t^n} + \frac{1}{2} \left[\int_{\mathbf{R}_j} \mathbf{v}_h^+ \mathbf{v}_h^+ d\mathbf{x} \right]_{t=t^n} \right. \\
 & \left. - \left[\int_{\mathbf{R}_j} \mathbf{v}_h^+ \mathbf{v}_h^- d\mathbf{x} \right]_{t=t^n} + \int_{\partial \mathbf{R}_j^{st}} \mathbf{v}_h \cdot (\mathbf{G}_h \cdot \mathbf{n}) ds dt - \int_{\mathbf{R}_j^{st}} \nabla \mathbf{v}_h : \mathbf{F}_h d\mathbf{x} dt \right) = 0.
 \end{aligned} \tag{3.81}$$

We can recognize that

$$\begin{aligned}
 & \frac{1}{2} \left[\int_{\mathbf{R}_j} \mathbf{v}_h^- \mathbf{v}_h^- d\mathbf{x} \right]_{t=t^n} + \frac{1}{2} \left[\int_{\mathbf{R}_j} \mathbf{v}_h^+ \mathbf{v}_h^+ d\mathbf{x} \right]_{t=t^n} \\
 & - \left[\int_{\mathbf{R}_j} \mathbf{v}_h^+ \mathbf{v}_h^- d\mathbf{x} \right]_{t=t^n} = \frac{1}{2} \left[\int_{\mathbf{R}_j} (\mathbf{v}_h^+ - \mathbf{v}_h^-)^2 d\mathbf{x} \right]_{t=t^n} \geq 0
 \end{aligned} \tag{3.82}$$

and hence

$$\begin{aligned}
 & \sum_{j=1}^{N_d} \left(\frac{1}{2} \left[\int_{\mathbf{R}_j} \mathbf{v}_h^- \mathbf{v}_h^- d\mathbf{x} \right]_{t=t^{n+1}} - \frac{1}{2} \left[\int_{\mathbf{R}_j} \mathbf{v}_h^- \mathbf{v}_h^- d\mathbf{x} \right]_{t=t^n} \right. \\
 & \left. + \int_{\partial \mathbf{R}_j^{st}} \mathbf{v}_h \cdot (\mathbf{G}_h \cdot \mathbf{n}) ds dt - \int_{\mathbf{R}_j^{st}} \nabla \mathbf{v}_h : \mathbf{F}_h d\mathbf{x} dt \right) \leq 0.
 \end{aligned} \tag{3.83}$$

One can observe how in (3.83) the difference of the kinetic energy is computed at the end of two time slices. As for the method proposed in [144], the stability

of the scheme is therefore determined by the stability of the discretization used for the nonlinear convective term, which is assured according to [108, 6] when appropriate monotone entropy fluxes are used. One thus obtains the result:

$$\frac{1}{2} \left[\int_{\Omega} \mathbf{v}_h^- \mathbf{v}_h^- d\mathbf{x} \right]_{t=t^{n+1}} - \frac{1}{2} \left[\int_{\Omega} \mathbf{v}_h^- \mathbf{v}_h^- d\mathbf{x} \right]_{t=t^n} \leq 0. \quad (3.84)$$

Finally, note that even for $\mathbf{G}_h = \mathbf{F}_h \equiv 0$ the high order space-time DG scheme in general dissipates the total kinetic energy, since the solution is allowed to jump at the two sides of a time slice and hence, in general, $\mathbf{v}_h^-|_{t^n} \neq \mathbf{v}_h^+|_{t^n}$.

3.6 Numerical test problems

We present here some classical test problems for the two-dimensional incompressible Navier-Stokes equations. In general we can consider some steady or almost steady test cases that can be solved using a low order in time method, eventually extended using the Crank-Nicolson procedure. In the other cases where we have an high unsteady solution or the chosen time step Δt is too large to resolve well the time evolution, we need to use an high order method also in time. The section is organized as follows: in the first part we present some test cases for the special case $p_\gamma = 0$, such as presented in Section 3.3, then we will test the complete space-time method described in Sections 3.2 and 3.5.

3.6.1 Numerical test for the space only high order method

3.6.1.1 Convergence test

We consider a smooth steady state problem in order to measure the spacial order of accuracy of the proposed method. For this purpose, the Navier-Stokes equations are first rewritten in cylindrical coordinates (r and φ), with $r^2 = x^2 + y^2$, $\tan \varphi = x/y$, the radial velocity component u_r and the angular velocity component u_φ . In order to derive an analytical solution we suppose a steady vortex-type flow with angular symmetry, i.e. $\partial/\partial t = 0$, $\partial/\partial \varphi = 0$ and $u_r = 0$. With these assumptions, the continuity equation is automatically satisfied and the system of incompressible Navier-Stokes equations reduces to

$$\begin{cases} \frac{\partial p}{\partial r} = \frac{u_\varphi^2}{r}, \\ r \frac{\partial^2 u_\varphi}{\partial r^2} + \frac{\partial u_\varphi}{\partial r} - \frac{u_\varphi}{r} = 0. \end{cases} \quad (3.85)$$

One can now recognize in the second equation of (3.85) a classical second order Cauchy Euler equation and so obtain two solutions for u_φ , namely:

$$u_\varphi = c_1 r, \quad (3.86)$$

$$u_\varphi = \frac{c_1}{r}, \quad (3.87)$$

for every $c_1 \in \mathbb{R}$. The corresponding pressures read

$$p = \frac{c_1^2 r^2}{2} + c_2, \quad (3.88)$$

$$p = -2\frac{c_1^2}{r^2} + c_2. \quad (3.89)$$

respectively. In this section we set the boundary conditions in order to obtain the non-trivial solution (3.87)-(3.89). Due to the singularity of u_φ for $r = 0$, let $\Omega = C(5) - C(1)$ where $C(r) = \{(x, y) \in \mathbb{R}^2 \mid \sqrt{x^2 + y^2} \leq r\}$. As initial condition we impose Eqs. (3.87)-(3.89) with $c_1 = u_\varphi(1) = 2$ and $c_2 = 0$. The exact velocity is imposed at the internal boundary while the exact pressure is specified at the external circle. The proposed algorithm is validated for several polynomial degrees p using successively refined grids. The chosen parameters for the numerical simulations are $t_{end} = 0.75$; $\theta = 1$; $\nu = 10^{-5}$; the time step Δt is taken according to the CFL time restriction for the explicit discretization of the nonlinear convective term (3.39). The L_2 error between the analytical and the numerical solution is computed as

$$\epsilon(p) = \sqrt{\int_{\Omega} (p_h - p_e)^2 dx}, \quad \epsilon(\vec{v}) = \sqrt{\int_{\Omega} (\mathbf{v}_h - \mathbf{v}_e)^2 dx}, \quad (3.90)$$

for the pressure and for the velocity vector field, respectively, where the subscript h indicates the numerical solution and e denotes the exact solution. Tables 3.1 and 3.2 show the L_2 convergence rates for successive refinements of the grid, where $\mathcal{O}(p)$ and $\mathcal{O}(\mathbf{v})$ represent the order of accuracy achieved for the pressure and the velocity field, respectively. The optimal convergence is reached up to $p = 2$ while for $p = 3$ the observable order of accuracy for the velocity vector field is closer to $p + \frac{1}{2}$ rather than $p + 1$.

3.6.1.2 Blasius boundary layer

Another classical test problem concerns the Blasius boundary layer. For the particular case of laminar stationary flow over a flat plate, a solution of Prandtl's boundary layer equations was found by Blasius in [19] and is determined by the solution of a third-order non-linear ODE, namely:

$$\begin{cases} f''' + ff'' = 0, \\ f(0) = 0, \quad f'(0) = 0, \quad \lim_{\xi \rightarrow \infty} f'(\xi) = 1, \end{cases} \quad (3.91)$$

where $\xi = y\sqrt{\frac{u_\infty}{2\nu x}}$ is the Blasius coordinate; $f' = \frac{u}{u_\infty}$; and u_∞ is the farfield velocity. The reference solution is computed here using a tenth-order DG ODE solver, see e.g. [67], together with a classical shooting method. In order to obtain the Blasius velocity profile in our simulations we consider a steady flow

Table 3.1: Numerical convergence results for $p = 0$ and $p = 1$.

N_e	$p = 0$			
	$\epsilon(p)$	$\epsilon(\mathbf{v})$	$\mathcal{O}(p)$	$\mathcal{O}(\mathbf{v})$
124	7.902E-01	1.095E-00	-	-
496	5.026E-01	7.086E-01	0.7	0.6
1984	2.982E-01	4.502E-01	0.8	0.7
7936	1.659E-01	2.797E-01	0.8	0.7
31744	8.797E-02	1.714E-01	0.9	0.7
N_e	$p = 1$			
	$\epsilon(p)$	$\epsilon(\mathbf{v})$	$\mathcal{O}(p)$	$\mathcal{O}(\mathbf{v})$
124	3.944E-01	4.311E-01	-	-
496	8.830E-02	1.221E-01	2.2	1.8
1984	2.325E-02	3.299E-02	1.9	1.9
7936	6.207E-03	8.725E-03	1.9	1.9
31744	1.615E-03	2.318E-03	1.9	1.9

over a a wedge-shaped object. As a result of the viscosity, a boundary layer appears along the obstacle. For the present test, we consider $\Omega = [0, 1] \times [-0.25, 0.25]$ and a wedge shape object with upper edge corresponding to the segment $x = [0, 1]$. An initially uniform flow $u(x, y, 0) = u_\infty = 1$, $v(x, y, 0) = 0$ and $p(x, y, 0) = 1$ is imposed as initial condition, while an inflow boundary is imposed on the left and outflow boundary conditions are imposed on the other edges of the external box. Finally, no-slip wall boundary conditions are considered over the wedge shape object. We cover Ω with a total amount of $N_e = 278$ triangles and use $\theta = 1$ and $p = 3$. The resulting Blasius velocity profile is shown in Figure 3.2, while the profile with respect to the Blasius coordinate ξ is shown in Figure 3.3 in order to verify whether the obtained solution is self-similar with respect to ξ . A comparison between the numerical results presented here and the reference solution is depicted in Figure 3.4 for $x = 0.4$ and $x = 0.6$. A good agreement between the reference solution and the numerical results obtained with the staggered semi-implicit DG scheme is obtained, despite the use of a very coarse grid. Note that the solution in terms

Table 3.2: Numerical convergence results for $p = 2$ and $p = 3$.

N_e	$p = 2$			
	$\epsilon(p)$	$\epsilon(\mathbf{v})$	$\mathcal{O}(p)$	$\mathcal{O}(\mathbf{v})$
124	9.366E-02	1.990E-01	-	-
496	1.054E-02	3.069E-02	3.2	2.7
1984	1.193E-03	3.686E-03	3.1	3.1
7936	1.438E-04	4.425E-04	3.1	3.1
N_e	$p = 3$			
	$\epsilon(p)$	$\epsilon(\mathbf{v})$	$\mathcal{O}(p)$	$\mathcal{O}(\mathbf{v})$
124	4.346E-02	9.317E-02	-	-
496	2.966E-03	8.027E-03	3.9	3.5
1984	1.783E-04	7.153E-04	4.1	3.5
7936	1.313E-05	5.997E-05	3.8	3.6

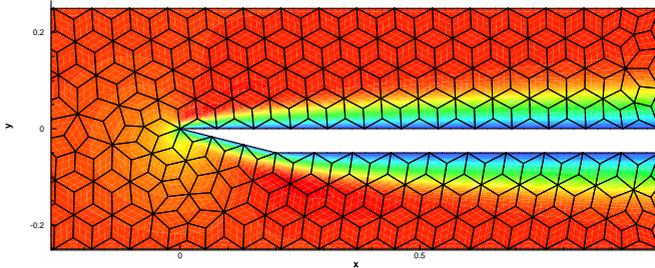


Figure 3.2: Computational domain used for the simulation of the Blasius boundary layer. The colors represent the horizontal velocity u .

of the Blasius coordinate ξ is independent from x . The numerical solution is also verified to maintain the self-similar Blasius profile in the (x, ξ) plane, see fig. 3.3.

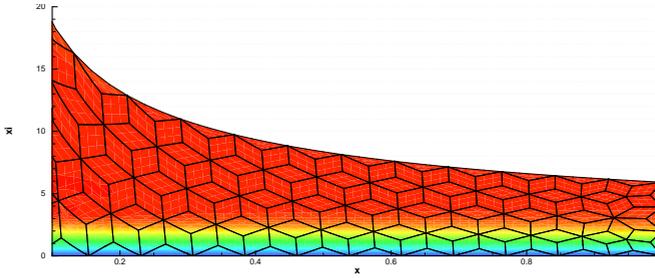


Figure 3.3: Velocity profile with respect to the Blasius coordinate ξ .

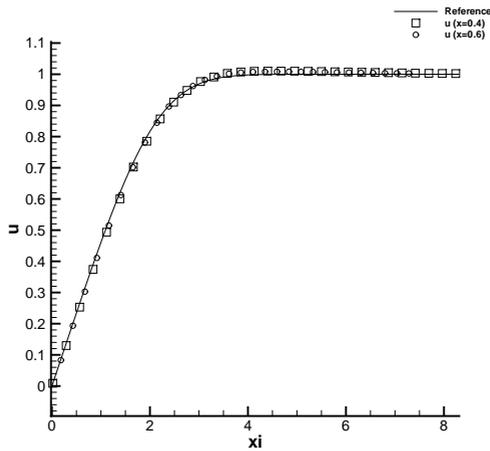


Figure 3.4: Numerical and reference solution for the Blasius boundary layer at $x = 0.4$ and $x = 0.6$.

3.6.1.3 Lid-driven cavity flow

We consider here another classical benchmark problem for the incompressible Navier-Stokes equations, namely the lid-driven cavity problem. This test problem is solved numerically with the new staggered DG scheme on very coarse grids using a polynomial degree of $p = 3$. Let $\Omega = [-0.5, 0.5] \times [-0.5, 0.5]$, set velocity boundary conditions $u = 1$ and $v = 0$ on the top boundary (i.e. $y = 0.5$) and impose no-slip wall boundary conditions on the other edges. As initial condition we take $u(x, y, 0) = v(x, y, 0) = 0$. We use a grid with $N_e = 73$ triangles for $Re = 100, 400, 1000$ and $N_e = 359$ triangles for $Re = 3200$. A sketch of the main and dual grid is shown in fig. 3.5.

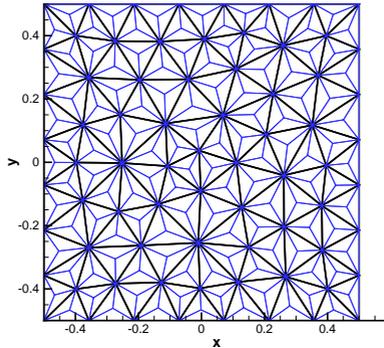


Figure 3.5: Main and dual grid used for the lid-driven cavity problem for $Re = 100, 400, 1000$.

For the present test $\theta = 1$; Δt is taken according to condition (3.39); and $t_{end} = 150$. According to [110, 89], primary and corner vortices appear from $Re = 100$ to $Re = 3200$, a comparison of the velocities against the data presented in [89], as well as the streamline plots are shown in Figure 3.6 and 3.7. A very good agreement is obtained in all cases, even if a very coarse grid has been used.

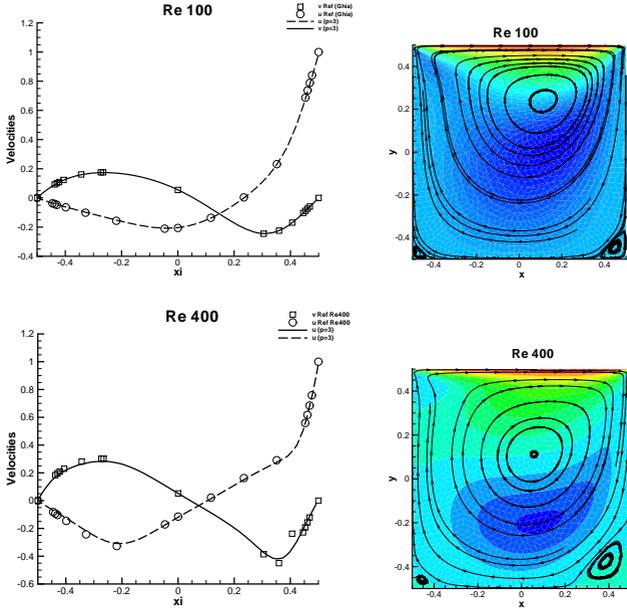


Figure 3.6: Velocity profiles (left) and streamlines (right) at $Re = 100$ and $Re = 400$ for the lid-driven cavity problem.

3.6.1.4 Backward-facing step.

In this section, the numerical solution for the fluid flow over a backward-facing step is considered. For this test problem, both experimental and numerical results are available at several Reynolds numbers (see e.g. [3, 79]). The computational domain Ω and the main notation are reported in Figure 3.8. The fluid flow is driven by a pressure gradient imposed at the left and the right ends of the computational domain. On all the other boundaries, no-slip wall boundary conditions are imposed. According to [3], we take $Re = \frac{DU}{\nu}$ where $D = 2h_{in}$; U is the mean inlet velocity; ν is the kinematic viscosity. The computational domain is covered with a total number of $N_e = 260$ triangles

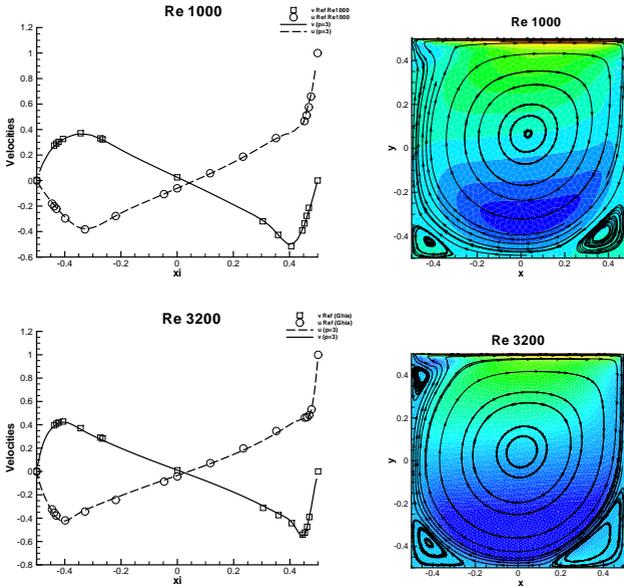


Figure 3.7: Velocity profiles (left) and streamlines (right) at $Re = 1000$ and $Re = 3200$ for the lid-driven cavity problem.

with characteristic size $h = 0.2$ for $x \leq 5$ and $h = 0.48$ for $x > 5$ (see Figure 3.8). Finally we use $p = 3$; $\theta = 1$ and Δt is the one given by the CFL condition for the nonlinear convective term; $t_{end} = 80s$. Figures 3.10 and 3.11 show the vortices generated at different Reynolds numbers, while in Figure 3.9 the main recirculation point $X1$ is compared with experimental data given by Armaly in [3], and the explicit second-order upwind finite difference scheme introduced in [17]. A good agreement with the experimental data is shown up to $Re = 316$ but, according to [3], the experiment becomes three dimensional for $Re > 400$, so the comparison can be done only up to $Re = 400$. Indeed, one can see in fig. 3.11 how the secondary vortex occurs for $Re = 426$, while in the experiments it appears at higher Reynolds numbers (see e.g. [3]).

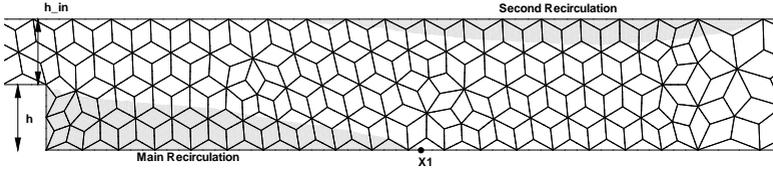


Figure 3.8: Grid and main notation used for the backward-facing step problem.

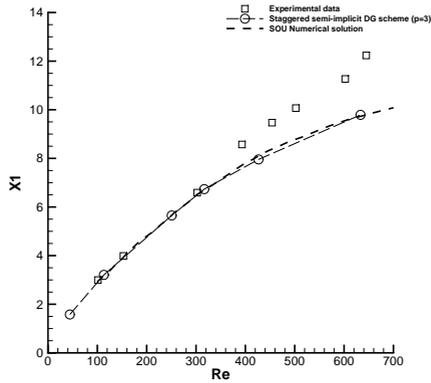


Figure 3.9: Comparison of the experimental data of Armaly et al. [3] with the numerical results obtained with the present semi-implicit staggered DG scheme and the numerical solution obtained in [79] for the reattachment point $X1$ in the backward-facing step problem.

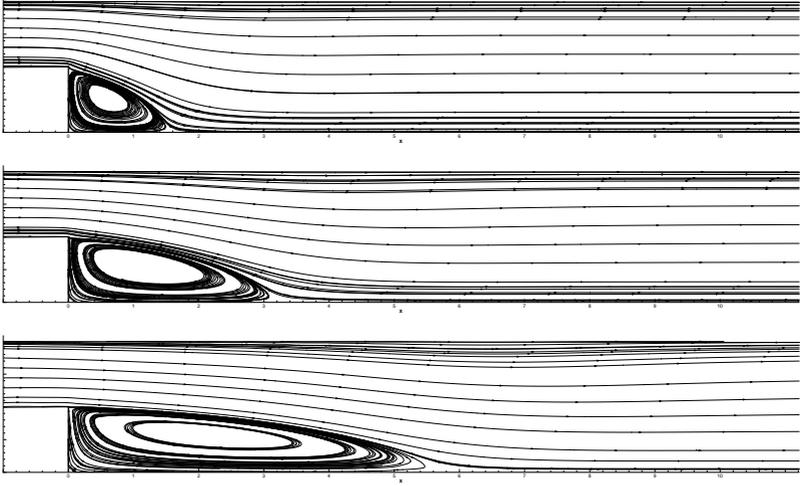


Figure 3.10: Streamlines at Reynolds numbers $Re = 44, 113$ and 250 from top to bottom.

3.6.1.5 Rotational flow past a circular half-cylinder

Here we consider a rotational flow past a circular half-cylinder. A comparison between numerical and exact analytical solution is possible for incompressible and inviscid fluid, i.e. here we set $\nu = 0$. We use the computational setup of Feistauer and Kucera [82], hence $\Omega = [-5, 5] \times [0, 5] - \{\sqrt{x^2 + y^2} \leq 0.5\}$; as boundary conditions we impose the velocity at the left boundary; homogeneous Neumann boundary conditions on the top and right boundaries and inviscid wall at the bottom and the surface of the half-cylinder. The farfield velocity field is given by $u = y$ and $v = 0$. The exact analytical solution to this problem was found by Fraenkel in [85]. For the present test we choose $p = 3$; Δt is set according to (3.39) and we cover Ω with $N_e = 800$ triangles, using only 6 triangles to describe the half-cylinder. Curved isoparametric elements are considered in order to represent the geometry of the half-cylinder properly. As initial conditions we impose $p(x, y, 0) = 1$; $u(x, y, 0) = y$ and $v(x, y, 0) = 0$.

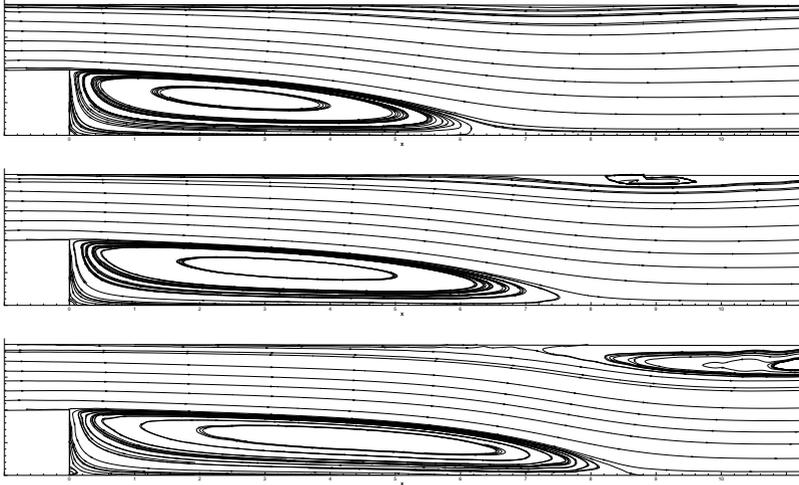


Figure 3.11: Streamlines at Reynolds numbers $Re = 316, 426$ and 633 from top to bottom.

Two vortices appear near the half-cylinder (see fig. 3.12 left), while a comparison between analytical and numerical velocity magnitude on the cylinder surface (i.e. $r = 0.5$) is shown on the right of fig. 3.12. A good agreement between analytical and numerical results is obtained also with a very coarse grid. An important remark is that for this test problem the use of isoparametric elements is crucial, as previously shown for inviscid flow past a circular cylinder by Bassi and Rebay in [12].

3.6.1.6 Flow over a circular cylinder

In this section we consider the flow over a circular cylinder. Also in this case, the use of the isoparametric approach is mandatory to represent the geometry of the cylinder wall, see [12, 143]. In particular, two cases are considered: first, an inviscid flow around the cylinder is assumed in order to obtain a steady potential flow; finally, the complete viscous case is considered in order to get

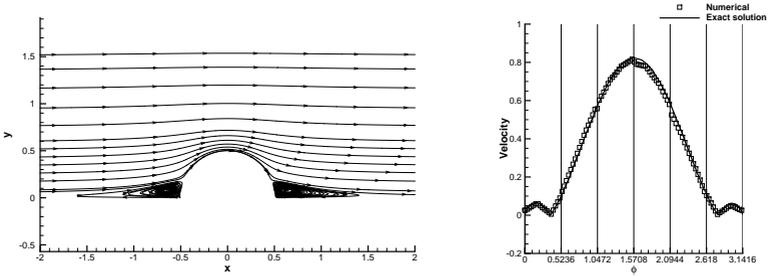


Figure 3.12: Rotational inviscid flow past a circular half-cylinder. Left: Streamlines. Right: Analytical and numerical results for $r = 0.5$. The vertical lines show the dimension of the six curved elements that cover the half-cylinder.

the unsteady von Karman vortex street. For the first case a sufficiently large domain $\Omega = [-8, 8] \times [-8, 8] - \{\sqrt{x^2 + y^2} \leq 1\}$ is employed. The exact solution for this case is known and reads:

$$\begin{aligned}
 u_r(r, \varphi) &= \bar{u} \left(1 - \frac{R_c^2}{r^2}\right) \cos(\varphi), & u_\varphi(r, \varphi) &= -\bar{u} \left(1 + \frac{R_c^2}{r^2}\right) \sin(\varphi), \\
 p &= \frac{1}{2} \bar{u}^2 \left(\frac{2R_c^2}{r^2} \cos(2\varphi) - \frac{R_c^4}{r^4} \right),
 \end{aligned} \tag{3.92}$$

where \bar{u} is the inflow velocity; R_c is the cylinder radius; u_r and u_φ are the radial and angular components of the velocity, respectively. An initial condition $\mathbf{v}(x, y, 0) = (\bar{u}, 0)$ is used, while the exact velocity distribution is taken as the external boundary condition. An inviscid wall boundary condition is imposed on the cylinder. For the present test $\bar{u} = 0.01$; $R_c = 1$; $\nu = 0$; $p = 3$; $\theta = 0.6$; Δt is the one taken according to the CFL restriction (3.39); $t_{end} = 10$. The domain Ω is covered with a total number of $N_e = 1464$ triangles and an isoparametric approach is considered to represent the cylinder wall properly. Figure 3.13 shows the streamlines and the pressure contours obtained at $t = 10$ as well as the comparison between exact and numerical solution at several radii. A very good agreement between exact and numerical solution is observed. We consider

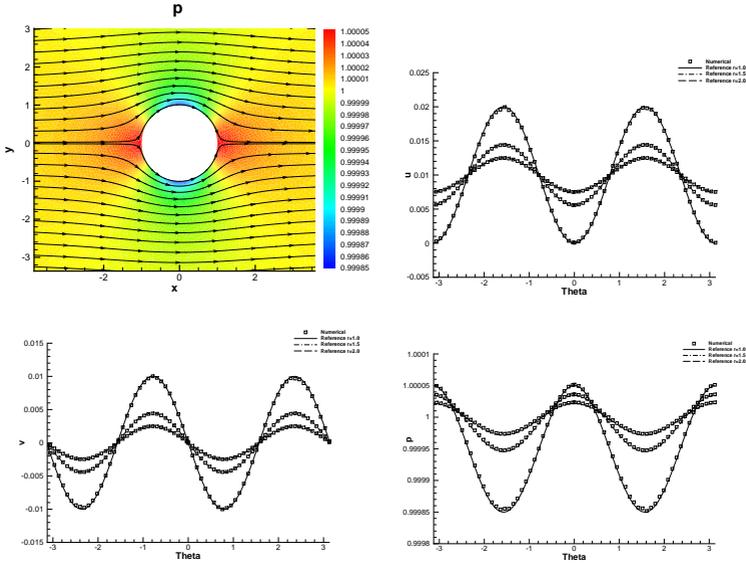


Figure 3.13: Steady flow of an inviscid incompressible fluid around a circular cylinder. On the top left: streamlines and pressure contours at $t_{end} = 10$; Numerical and exact solution at $r = 1.0$, $r = 1.5$ and $r = 2.0$ for the velocity components u , v and pressure p from top right to the bottom right, respectively.

now the fully viscous case in order to show the formation of the von Karman vortex street. Two domains are considered here: $\Omega_1 = [-20, 80] \times [-20, 20]$ covered with a $N_e = 1702$ triangles; and $\Omega_2 = [-5, 30] \times [-10, 10]$ covered with a $N_e = 1706$ triangles. As initial condition we set $\mathbf{v}(x, y, 0) = (\bar{u}, 0)$; $\theta = 0.6$; and $\bar{u} = 0.5$. Different viscosity coefficients are used in order to obtain different Reynolds numbers. For the present test we use Δt according to (3.39); and $p = 3$. The velocity $(\bar{u}, 0)$ is prescribed at the left boundary while homogeneous Neumann boundary conditions are imposed on the other external edges of the domain. Finally viscous wall boundary condition is imposed on

the cylinder surface. Figure 3.14 shows the obtained relationship between the

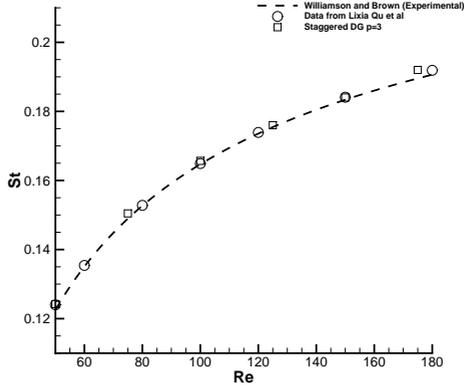


Figure 3.14: Strouhal-Reynolds number relationship for the present method, the method of Qu et al. [131] and experimental data of Williamson and Brown [157].

Strouhal number, computed as $St = \frac{2rf}{u_\infty}$, the numerical results given by Qu et al (see [131]) and the experimental law given in [157]. The simulations are performed on the domain Ω_1 . The numerical results fit well the experimental data and the numerical reference solution up to $Re = 150$. Better results can be obtained by further enlarging the computational domain. The velocity field and the vorticity show different structures when low and high Reynolds numbers are considered. The vorticity contours are shown in Figure 3.15 for $Re = 50$ and $Re = 125$ at time $t = 500$. In the case of $Re = 125$ the von Karman vortex street is fully developed while, for $Re = 50$, the two initial vortices remain present behind the cylinder for a longer time. This is due to the low value of the Reynolds number, taken close to the limit of $Re = 40$ for the generation of the vortex street.

The time evolution of the generation of the von Karman vortex street is presented at several times for $Re = 200$ on Ω_2 in Figure 3.16.

Finally, in Figure 3.17 we report a comparison between the computational time

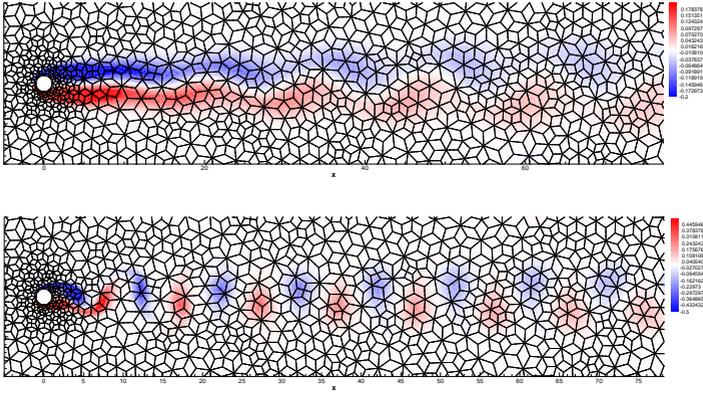


Figure 3.15: Dual mesh and vorticity contours of the von Karman vortex street generated at time $t = 500$ for $Re = 50$ (top) and $Re = 125$ (bottom).

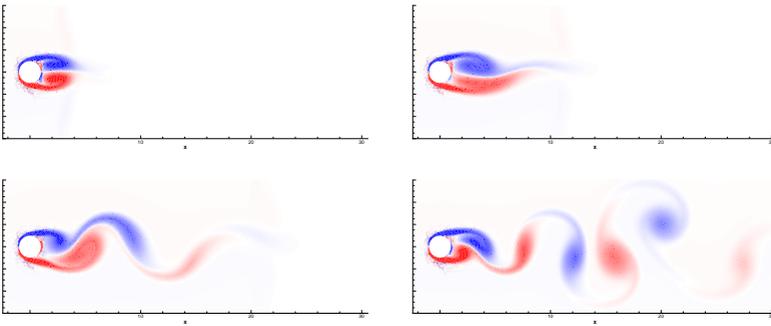


Figure 3.16: Temporal evolution of the vorticity profile for $t = 15$, $t = 30$, $t = 50$, $t = 75$ from top left to bottom right at $Re = 200$.

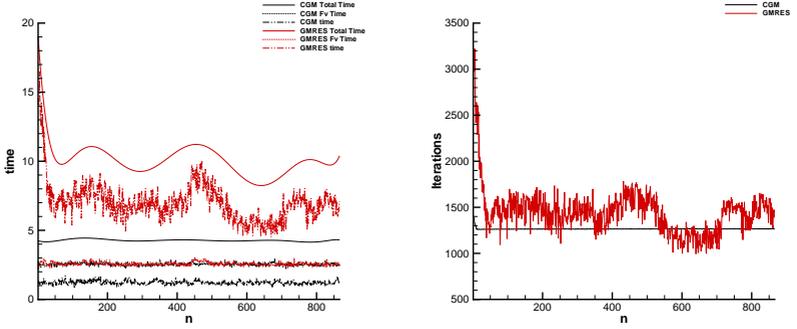


Figure 3.17: Left: comparison between the CPU time for the GMRES method and the CG method. Right: total number of iterations for the GMRES against the number of iterations for the CG method.

needed per time step for the main parts of the algorithm presented in this paper up to the time $t = 10s$ using $Re = 100$ on Ω_1 if we employ a GMRES method or the cheaper CG method for the solution of the linear system. Note that since our particular semi-implicit DG discretization of the incompressible Navier-Stokes equations on staggered grids leads to a symmetric and positive-definite linear system, we can employ the CG method. This is not always the case for DG schemes applied to the incompressible Navier-Stokes equations since some formulations may also lead to non-symmetric linear systems.

The time required to compute the convective-viscous term represents, in the second case, the main computational effort. Using the GMRES algorithm the computational time needed to solve the linear system increases a lot compared to the CG method and becomes the main cost of the algorithm. In particular, the mean time to solve the system using the GMRES algorithm is, for this test, $6.2s$ while using the CG method is only about $1.0s$. For all tests, the tolerance for solving the linear system was set to $tol = 10^{-12}$. We underline that for a fair comparison of the two methods, no preconditioners have been used and that faster convergence can be obtained by using a proper preconditioner for each iterative solver.

3.6.2 Numerical tests for the space-time DG method

3.6.2.1 Convergence test using a manufactured solution

In order to study the accuracy of the proposed space-time DG method, we need an exact unsteady solution of (3.1)-(3.3). For that purpose, we propose a so-called *manufactured solution* in this section, which also makes use of a linear source term of the type $\mathbf{S}(x, y, t)$. The exact analytical solution for the velocity and the pressure is constructed so that

$$\mathbf{v}_e = \mathbf{v}_0 \sin [k(x - y) - \omega t], \quad p_e = p_0 \sin [k(x - y) - \omega t], \quad (3.93)$$

with the amplitudes $\mathbf{v}_0 = (u_0, v_0)$ and p_0 . Using the manufactured solution (\mathbf{v}_e, p_e) we can compute all terms in (3.1) exactly and hence obtain a source term $\mathbf{S}(x, y, t)$ that balances the momentum equation. Remark that the velocity field must be divergence-free ($\nabla \cdot \mathbf{v} = 0$), hence $u_0 = v_0$. In the present test case, we take $u_0 = v_0 = 1$; $p_0 = 1$; $\omega = 2\pi$; $k = 10/2\pi$; $t_{end} = 0.5$; Δt according to condition (3.39); and $\nu = 0.01$. The temporal accuracy is chosen equal to the spatial one, the total number of Picard iterations is taken as $N_{pic} = p + 1$ and $p^{n+1,0} \equiv 0$ for the present test. The computational domain is $\Omega = [-0.5, 0.5]^2$; the exact velocity field and pressure are taken as initial conditions and the exact pressure is also specified on $\partial\Omega$ as boundary condition. The L_2 error between the analytical and the numerical solution is computed as

$$\epsilon(p) = \sqrt{\int_{\Omega} (p_h - p_e)^2 d\mathbf{x}} \quad , \quad \epsilon(\mathbf{v}) = \sqrt{\int_{\Omega} (\mathbf{v}_h - \mathbf{v}_e)^2 d\mathbf{x}}$$

where the subscript h refers to the numerical solution obtained at the final time $t = t_{end}$. The resulting rate of convergence is shown in Table (3.3). We observe that the optimal order of convergence is obtained up to $p = 4$ for the present unsteady test.

3.6.2.2 The Womersley problem

Here we consider an unsteady, viscosity-dominated test problem for which the incompressible Navier-Stokes equations have a nontrivial exact solution, namely the fluid flow inside a rigid planar pipe that is driven by a sinusoidal pressure gradient of the type

$$\frac{p_{out}(t) - p_{in}(t)}{L} = \Re \left(\frac{\tilde{P}}{\rho} e^{i\omega t} \right). \quad (3.94)$$

Table 3.3: Numerical convergence results for the manufactured solution test problem with polynomial degrees $p = 1$ to $p = 4$ in space and time.

N_e	$\epsilon(p)$	$\epsilon(\mathbf{v})$	$\sigma(p)$	$\sigma(\mathbf{v})$
$p = p_\gamma = 1$				
40	1.217E-01	9.572E-02	-	-
160	2.678E-02	2.362E-02	2.2	2.0
640	6.050E-03	5.527E-03	2.1	2.1
2560	1.758E-03	1.497E-03	1.8	1.9
$p = p_\gamma = 2$				
40	8.740E-03	1.052E-02	-	-
160	8.833E-04	1.065E-03	3.3	3.3
640	1.050E-04	9.103E-05	3.1	3.5
2560	1.347E-05	7.820E-06	3.0	3.5
$p = p_\gamma = 3$				
40	7.703E-04	1.425E-03	-	-
160	3.864E-05	4.999E-05	4.3	4.8
640	2.425E-06	1.974E-06	4.0	4.7
2560	1.789E-07	1.288E-07	3.8	3.9
$p = p_\gamma = 4$				
40	5.315E-05	7.135E-05	-	-
160	1.143E-06	1.418E-06	5.5	5.7
640	3.102E-08	2.945E-08	5.2	5.6

In this test L denotes the tube length; \tilde{P} is the amplitude of the pressure oscillation; ρ is the density of the fluid; ω is the frequency of the oscillation; p_{in} and p_{out} indicate the inlet and the outlet pressure, respectively; \Re is the real part operator. By imposing Eq. (3.94) at the tube ends, the exact analytical solution for the three dimensional, axially symmetric case was found by Womersley in [158]. It can be derived also for the two dimensional planar case. The resulting axial velocity is uniform in the x -direction and is given by

$$u(x, y, t) = \Re \left[i \frac{\tilde{P}}{\rho} \omega \left(1 - \frac{\cos[\lambda(y_c - 1)]}{\cos(\lambda)} \right) \right], \quad (3.95)$$

where $\lambda = \sqrt{-i\alpha^2}$; $\alpha = R\sqrt{\frac{\omega}{\nu}}$; $y_c = \frac{y - y_b}{R}$; and y_b is the y value of the bottom. For the present test $\Omega = [-0.5, 0.5] \times [-0.2, 0.2]$; and $\frac{\tilde{P}}{\rho} = 1$. We take a set of successively refined grids in order to show the convergence behaviour to the exact solution with respect to the order p in space and p_γ in time. According to [158] the nonlinear convection effect is neglected for the present test. Thus, the stability of our scheme is not restricted by the CFL condition on the fluid velocity. Since we use very large time steps and a high viscosity coefficient in this test, the implicit treatment of the viscous terms is necessary to allow large time steps. In particular we choose $\nu = 5 \cdot 10^{-2}$ and $t_{end} = 1.5$. On the coarsest grid we use $\Delta t = t_{end}/6$, then the time step is reduced proportional to the spatial grid size. No-slip boundary conditions are imposed on the top and the bottom boundary, while the pressure (3.94) is imposed at the inlet and the outlet boundary on the left and on the right, respectively. The number of Picard iterations is given by $N_p = p + 1$ for all simulations.

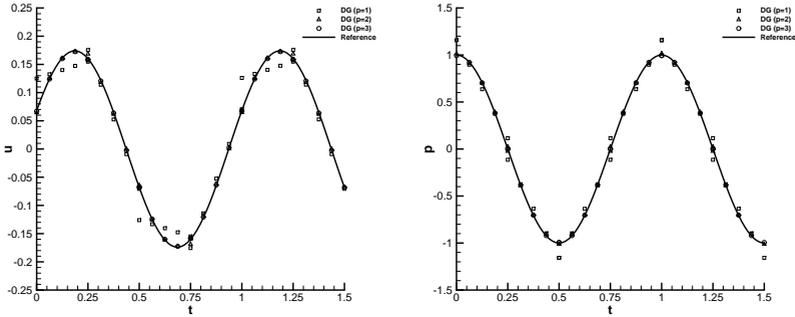


Figure 3.18: Time series for the axial velocity u and the pressure p computed at $(x, y) = (-0.5, 0)$ for the coarsest grid $N_e = 46$ and $N_t = 6$.

The resulting convergence results, using the L_2 -norm as in the previous example, are shown in Table 3.4. Observe how a non-optimal order of convergence p is achieved for the velocity for odd order schemes (even polynomial degree p), while the optimal convergence rate $p+1$ is achieved for the pressure for all polynomial degrees. Note that when using the semi-implicit staggered DG method

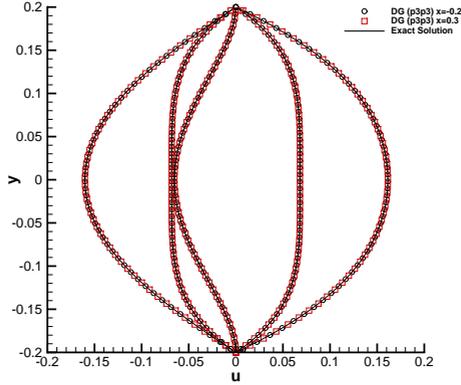


Figure 3.19: Radial velocity profiles for $x = -0.2$ and $x = 0.3$ at times, from left to right, $t = [0.75, 0.5, 0.875, 1.0, 0.125]$. Comparison between exact and numerical solution.

introduced in [143], only a second order of convergence could be achieved for this unsteady test problem, while full high order convergence in space and time is obtained with the new scheme presented in this paper. In Figure 3.18 we show the time series of the axial velocity and the pressure in a given point for the coarsest grid configuration $(N_e, N_t) = (46, 6)$. While piecewise linear space-time polynomials are not able to reproduce the sinusoidal signal well with only six time steps, the piecewise quadratic and higher order approximations in space and time yield an almost perfect match with the exact solution even on this extremely coarse space-time grid.

In Figure 3.19 we compare the resulting numerical velocity profiles $u(y)$ against the exact solution at several times for the case $(p, p_\gamma) = (3, 3)$ and $N_e = 736$. Two different locations, $x = -0.2$ and $x = 0.3$, are plotted in order to show that the profile is constant in the x -direction. One observes that there is no visible difference between numerical and exact solution.

Table 3.4: Numerical convergence results for the planar Womersley problem.

p	p_γ	N_e	N_t	$\epsilon(p)$	$\epsilon(\mathbf{v})$	$\sigma(p)$	$\sigma(\mathbf{v})$
1	1	46	6	5.7880182E-02	1.8848423E-03	-	-
1	1	184	12	1.7635947E-02	5.5901107E-04	1.7	1.8
1	1	736	24	4.6206559E-03	1.4587701E-04	1.9	1.9
1	1	2944	48	1.1683966E-03	3.7404869E-05	2.0	2.0
2	2	46	6	7.0716231E-03	2.6412698E-04	-	-
2	2	184	12	4.8160864E-04	3.8846170E-05	3.9	2.8
2	2	736	24	3.0677533E-05	7.2036760E-06	4.0	2.4
2	2	2944	48	1.9295385E-06	1.6070616E-06	4.0	2.2
3	3	46	6	9.8372146E-04	1.2793693E-05	-	-
3	3	184	12	7.7144497E-05	7.8462176E-07	3.7	4.0
3	3	736	24	5.0814347E-06	4.8795894E-08	3.9	4.0
3	3	2944	48	3.2173776E-07	3.0326872E-09	4.0	4.0
4	4	46	6	7.3692980E-05	5.1193160E-07	-	-
4	4	184	12	1.2539784E-06	2.1649081E-08	5.9	4.6
4	4	736	24	2.1930727E-08	1.1576584E-09	5.8	4.2
4	4	2944	48	1.0258845E-09	7.0131498E-11	4.4	4.0

3.6.2.3 Taylor-Green vortex

Another widely used testcase for the verification of numerical methods for the incompressible Navier-Stokes equations is the Taylor-Green vortex problem. The analytical unsteady solution is given by

$$u(x, y, t) = \sin(x) \cos(y) e^{-2\nu t}, \quad (3.96)$$

$$v(x, y, t) = -\cos(x) \sin(y) e^{-2\nu t}, \quad (3.97)$$

$$p(x, y, t) = \frac{1}{4}(\cos(2x) + \cos(2y)) e^{-4\nu t}. \quad (3.98)$$

The computational domain is $\Omega = [0, 2\pi]^2$ and is extended using periodic boundary conditions on all the boundaries.

As implied by Eqs. (3.96)-(3.98), the resulting velocity field initially appears as depicted in Figure 3.20 and then starts to lose energy according to the friction

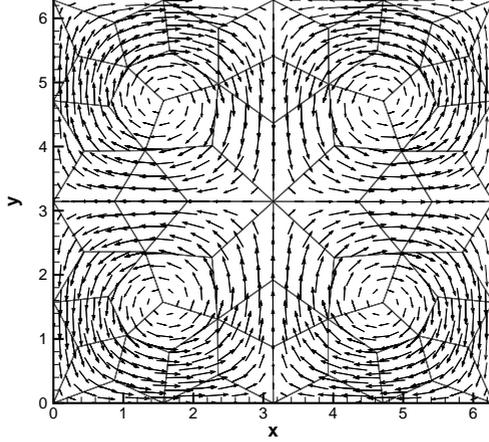


Figure 3.20: Velocity field of the Taylor-Green vortex on the coarse grid $N_e = 40$ with $p = 4$. The edge-based dual grid is shown.

effects. For the present test we consider several grid refinements; $t_{end} = 0.1$; $\nu = 0.1$; and Δt is chosen according to the CFL time restriction for the non-linear convective terms. The numerical convergence results are shown in Table 3.5. We find that the optimal convergence rate is achieved for this important nontrivial test problem with periodic boundary conditions.

3.6.2.4 Double shear layer

The numerical scheme is applied here to a test case studied in [14], which contains a high initial velocity gradient. We take $\Omega = [-1, 1]^2$ and, as initial

Table 3.5: Numerical convergence results for the velocity vector field of the Taylor-Green vortex.

N_e	$p = p_\gamma = 1$		$p = p_\gamma = 2$		$p = p_\gamma = 3$		$p = p_\gamma = 4$	
	$\epsilon(\mathbf{v})$	σ	$\epsilon(\mathbf{v})$	σ	$\epsilon(\mathbf{v})$	σ	$\epsilon(\mathbf{v})$	σ
40	3.088E-01	–	5.588E-02	–	5.895E-03	–	1.669E-03	–
160	8.868E-02	1.8	5.765E-03	3.3	4.730E-04	3.6	3.109E-05	5.7
640	2.267E-02	2.0	7.052E-04	3.0	2.387E-05	4.3	6.233E-07	5.6
2560	5.476E-03	2.0	8.452E-05	3.1	1.312E-06	4.2	1.297E-08	5.6

condition, we consider a perturbed double shear layer profile:

$$u_0 = \begin{cases} \tanh[\tilde{\rho}(y_n - 0.25)], & \text{if } y_n \leq 0.5, \\ \tanh[\tilde{\rho}(0.75 - y_n)], & \text{if } y_n > 0.5, \end{cases} \quad (3.99)$$

$$v_0 = \delta \sin(2\pi x_n), \quad (3.100)$$

$$p_0 = 1, \quad (3.101)$$

where $y_n = \frac{y+1}{2}$ and $x_n = \frac{x+1}{2}$ are the normalized vertical and horizontal coordinates, respectively; $\tilde{\rho}$ is a parameter that determines the slope of the shear layer; and δ is the amplitude of the initial perturbation. For the present test we set $\delta = 0.05$; $\tilde{\rho} = 30$; $\nu = 2 \cdot 10^{-4}$; $p = 4$ and $p_\gamma = 3$. The time step is chosen according to the CFL condition for the nonlinear convective terms and four Picard iterations have been used in this simulation. The domain Ω is covered with a total number of only $N_e = 640$ triangles and periodic boundary conditions are imposed everywhere. The resulting vorticity pattern is reported at several times in Figure 3.21. The two thin shear layers evolve into several vortices, as observed in [14], and overall the small flow structures seem to be relatively well resolved also at the final time $t = 1.8$, even if a very coarse grid has been used in space and time.

3.6.2.5 Flow over a circular cylinder

In this section we repeat the test done in Section 3.6.1.6 using the space-time high order version of the algorithm. As previously discussed, the use of an isoparametric finite element approach is mandatory to represent the curved

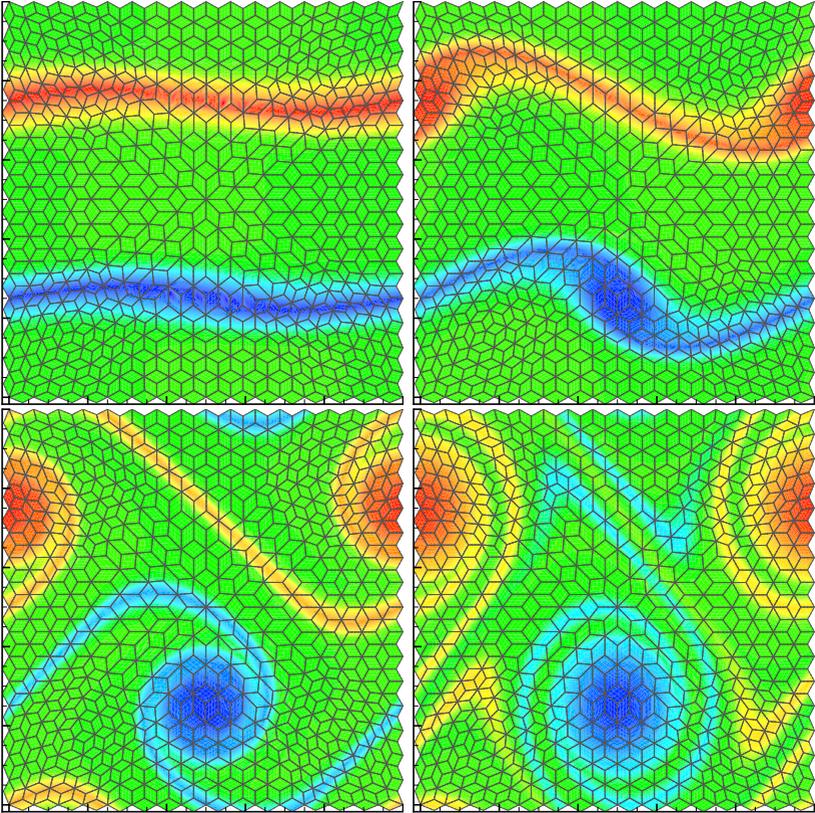


Figure 3.21: Vorticity pattern for the double shear layer test at times, from top left to bottom right, $t = 0.4; t = 0.8; t = 1.2; t = 1.8$.

geometry of the cylinder wall, see [12, 143]. We consider here directly the viscous case in order to show the formation of the von Karman vortex street. We take a sufficiently large domain $\Omega = [-20, 80] \times [-20, 20] - \{\sqrt{x^2 + y^2} \leq 1\}$ and we cover it with only $N_e = 1702$ triangles. Note that the chosen grid is extremely coarse compared to the dimension of the domain Ω . The

characteristic average size of the mesh is $h = 1.295$ and the smallest element size is about $h_{\min} = 0.347$. As initial condition we set $\mathbf{v}(x, y, 0) = (\bar{u}, 0)$, where \bar{u} is the inlet velocity, taking $\bar{u} = 0.5$ in our case. For the present test we use Δt according to (3.39); $p = 3$; $p_\gamma = 2$. The velocity $(\bar{u}, 0)$ is prescribed at the left boundary while transmissive boundary conditions are imposed on the other external edges of the domain. Finally a no-slip wall boundary condition is imposed on the cylinder surface. A plot of the streamlines is reported in Figure 3.22 at several output times. The resulting profiles for the vorticity and the horizontal velocity u are plotted in Figure 3.23, as well as the dual grid elements for $Re = 100$. As shown in Figure 3.22, two vortices are initially generated at the circular cylinder and then, several vortices leave the cylinder and generate the Von Karman street as we can see in Figure 3.23. The resulting Strouhal number for the present test is $St = \frac{f d}{\bar{u}} = 0.1647$ that is in good agreement with $St = 0.1649$ obtained by Qu et al. in [131].

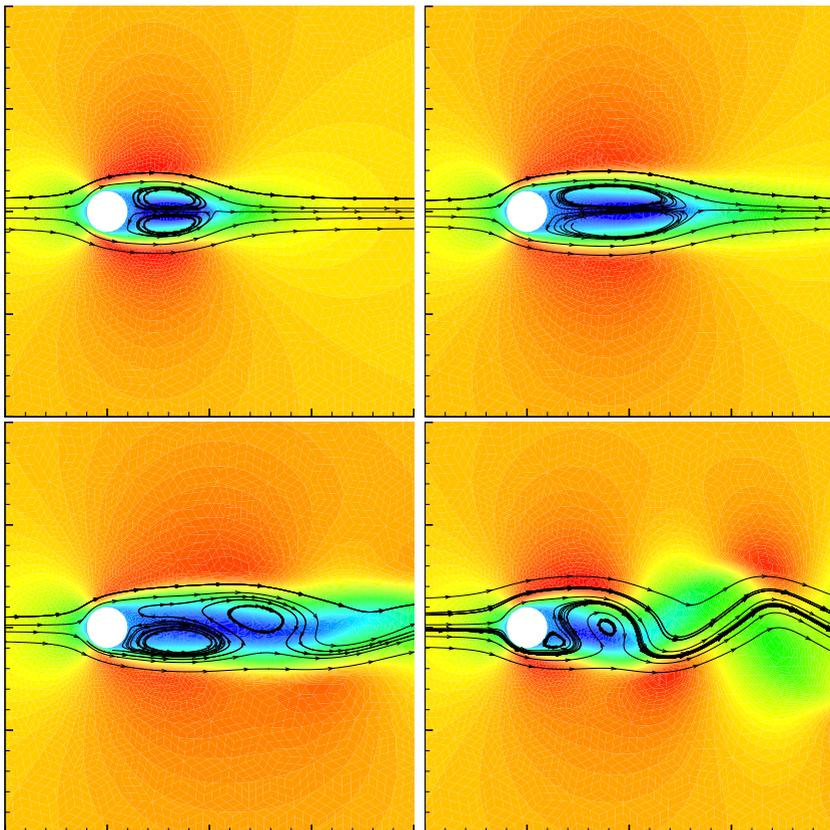


Figure 3.22: Streamlines along the circular cylinder at times, from top left to bottom right, $t = 25, 50, 100$ and $t = 200$.

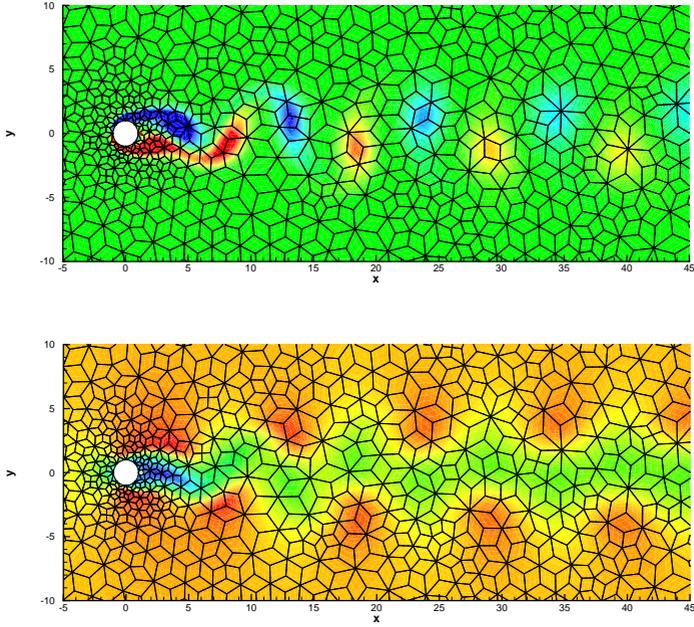


Figure 3.23: Laminar viscous flow past a circular cylinder. Profile for the vorticity and horizontal velocity u at time $t = 300$.

4 Three-dimensional incompressible Navier-Stokes equations

4.1 Overview

In this chapter we are going to extend the method presented in the previous Chapter 3 to three space dimensions. The method presented here can formally be derived in the same way such as described in Section 3.2.3. On the contrary, we have to change some details about basis functions or higher order geometry in order to solve the equations, such as will be discussed in Section 4.2.2 and in the appendix section A.4, respectively. In addition we present also an alternative computation of the nonlinear convective-viscous term that follows the same idea of the structured case [70]. A description about this new procedure for the convective-viscous contribution is presented in Section 4.2.4.1 while a numerical comparison between the natural extension and the one presented in this chapter is reported in Section 4.4. Several classical numerical tests for three-dimensional incompressible flows are reported in Section 4.3.

Furthermore, a parallel MPI implementation can be performed for this version of the code. The method developed using the new approach for the convective-viscous contribution, such as will be described in Section 4.2.4.1, would lead to a classical MPI implementation. On the contrary, if we use the natural extension of the convective-viscous term, then the MPI parallelization contains several issues given by the use of staggered grids, that lead to several special communications needed at the dual level between different MPI regions. The problematics in this case are similar to the ones obtained in the two dimensional case and are discussed in the appendix section A.2.

4.1.1 Governing equations

The three-dimensional incompressible Navier-Stokes equations are given by

$$\frac{\partial \vec{v}}{\partial t} + \nabla \cdot \mathbf{F} + \nabla p = S, \quad (4.1)$$

$$\nabla \cdot \vec{v} = 0, \quad (4.2)$$

that is formally the same such as in the section 3.1.1 but it contains an additional velocity component. Indeed in this case $\vec{v} = (u, v, w)$ is the velocity vector; u , v and w are the velocity components in the x , y and z direction, respectively; $\mathbf{F} = \mathbf{F}(\vec{v}, \nabla \vec{v}) = \mathbf{F}_c(\vec{v}) - \nu \nabla \vec{v}$ is a nonlinear tensor that depends on the velocity and its gradient, where $\mathbf{F}_c = \vec{v} \otimes \vec{v}$ is the flux tensor of the nonlinear convective terms, which in three-dimensions reads

$$\mathbf{F}_c = \begin{pmatrix} uu & uv & uw \\ vu & vv & vw \\ wu & vw & ww \end{pmatrix}.$$

4.2 DG scheme for the 3D incompressible Navier-Stokes equations

4.2.1 Unstructured staggered space-time grid

We directly extend the mesh and hence the notation used in chapter 3 to three dimensional main tetrahedral and dual hexahedral meshes.

Figures 4.1 and 4.2 summarize the used notation, the main tetrahedral and the dual hexahedral meshes.

In the time direction we do exactly the same procedure such as described in Section 3.2.2 and hence derive the final main space-time tetrahedral mesh $\{\mathbf{T}_i^{st}\}_{i \in [1, N_e]}$ and the dual hexahedral one $\{\mathbf{R}_j^{st}\}_{j \in [1, N_d]}$.

4.2.2 Space-Time basis functions

We construct first the space basis functions and then we extend them using the tensor product to space-time basis functions. We develop the basis functions on a generic tetrahedron through the basis functions on the standard tetrahedron defined such as $T_{ref} = \{(\xi, \gamma, \delta) \in \mathbb{R}^{3,+} \mid \xi + \gamma + \delta \leq 1\}$. We write a generic basis function such as

$$\phi_k(\xi, \gamma, \delta) = \sum_{r_3=0}^p \sum_{r_2=0}^{p-r_3} \sum_{r_1=0}^{p-r_3-r_2} \alpha_{r_1, r_2, r_3} \xi^{r_1} \gamma^{r_2} \delta^{r_3} \quad (4.3)$$

for some coefficients α_{r_1, r_2, r_3} . We set now $N_\phi = \frac{(p+1)(p+2)(p+3)}{6}$ nodal points $\{(\xi_k, \gamma_k, \delta_k)\}_k$ using the standard procedure for classical continuous conforming finite elements and we impose the condition $\phi_k(\xi_k, \gamma_k, \delta_k) = 1$, $\phi_k(\xi_l, \gamma_l, \delta_l) = 0$

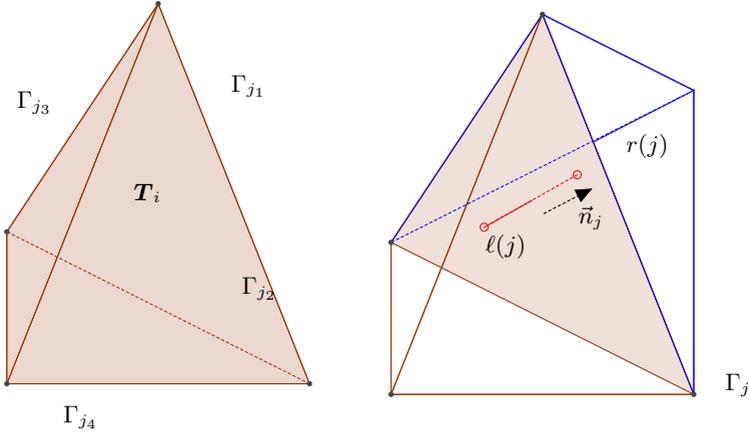


Figure 4.1: On the left: main tetrahedral mesh with $S_i = \{j_1, j_2, j_3, j_4\}$. On the right: standard orientation used through this chapter.

for all $l \neq k \in [1, N_\phi]$. This leads to a linear system for the coefficients α that we can solve analytically for every polynomial degree p on the reference tetrahedron. In this way we obtain N_ϕ basis functions on T_{ref} , $\{\phi_k\}_{k \in [1, N_\phi]}$. The connection between reference and physical space is performed by the map $T(\cdot, \mathbf{T}_i) = T_i : \mathbf{T}_i \rightarrow T_{ref}$ for every $i = 1 \dots N_e$ and its inverse, called $T^{-1}(\cdot, \mathbf{T}_i) = T_i^{-1} : \mathbf{T}_i \leftarrow T_{ref}$. The maps from the physical coordinates to the reference one can be constructed following a classical sub-parametric or a complete iso-parametric approach and in general we will write, for all $i = 1 \dots N_e$, $\phi_k^{(i)}(x, y, z) = \phi_k(T_i(x, y, z))$.

Unfortunately it is not so easy to construct the basis functions on the dual mesh since the additional degrees of freedom do not allow us to construct a bijective map between a reference square-based pyramid and the physical element. This problem can be skipped by introducing directly the basis functions in the physical space. The generic basis function will consequently depend on the element $j \in [1, N_d]$.

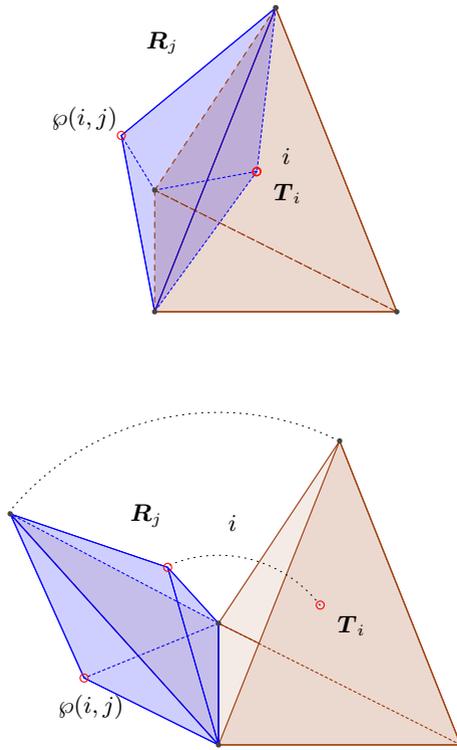


Figure 4.2: An example of dual hexahedral element based on the side Γ_j .

The natural extension to the three-dimensional case of the nodal approach

would read

$$\psi_k^{(j)}(x, y, z) = \sum_{r_3=0}^p \sum_{r_2=0}^{p-r_3} \sum_{r_1=0}^{p-r_3} \alpha_{r_1, r_2, r_3}^{(j)} \frac{(x - x_0^{(j)})^{r_1} (y - y_0^{(j)})^{r_2} (z - z_0^{(j)})^{r_3}}{h_j^{r_1+r_2+r_3}}, \quad (4.4)$$

where $(x, y, z)_0^{(j)}$ are some appropriate initial points and h_j is the characteristic length of \mathbf{R}_j . Starting from $N_\psi = \frac{(p+1)(p+2)(2p+3)}{6}$ points defined such as

$$(\xi, \gamma, \delta)_{k(r_1, r_2, r_3)} = \left(\frac{r_1}{p}, \frac{r_2}{p}, \frac{r_3}{p} \right)$$

where $k(r_1, r_2, r_3) = r_1 + (p - r_3 + 1)r_2 + (p - r_3 + 1)^2 r_3$ is a reordering function, we derive the position of the physical degrees of freedom (DoF) using the maps T on the standard tetrahedron as follows:

$$(x, y, z)_k = \begin{cases} T^{-1}(\xi_k, \gamma_k, \delta_k, \mathbf{T}_{\ell(j), j}) & \text{if } \xi_k + \gamma_k + \delta_k \leq 1 \\ T^{-1}(1 - \gamma_k - \delta_k, 1 - \xi_k - \delta_k, \mathbf{T}_{r(j), j}) & \text{otherwise} \end{cases} \quad (4.5)$$

Imposing

$$\psi_k^{(j)}(x_l, y_l, z_l) = \delta_{kl} \quad \forall k, l \in [1, N_\psi] \quad (4.6)$$

we obtain a linear system for every j that we can solve and so get N_ψ basis functions $\{\psi_k^{(j)}\}_{k \in [1, N_\psi]}$. The main advantages to use a nodal approach is that in the computation of the nonlinear convective term we can use $\hat{F}(\vec{v}) = \hat{v} \otimes \hat{v}$. Unfortunately, the dual face-based hexahedral elements are non-standard, since they have six faces but only five nodes, so they are naturally represented by a generalization of a square based pyramid. The definition of Lagrange basis functions on this kind of elements is problematic and usually we have to pass to rational functions of polynomials instead of using polynomial functions (see e.g. [45]); for instance, the system defined in (4.6) becomes singular for some choices of $\{\vec{x}_l\}_l$ and hence we are not able to construct a polynomial basis function such as expressed in (4.4) for a generic element \mathbf{R}_j . An alternative basis function that we can develop is based on the optimal number of polynomial functions N_ϕ used to construct the polynomial basis on T_{std} and project them in the physical hexahedral element, that contains a sufficient number of basis function to represent the polynomial space of degree p . With this choice we get in general only a modal base at the dual hexahedral level, i.e. if the convective term is computed on the dual mesh according to the natural extension of the one

proposed in [144], then it has to be computed according to a modal approach, see Section 4.2.4.

Finally, the time basis functions are constructed on a reference interval $[0, 1]$ for polynomial of degree p_γ . In this case the resulting $N_\gamma = p_\gamma + 1$ basis functions $\{\gamma_k\}_{k \in [1, N_\gamma]}$ are defined as the Lagrange interpolation polynomials passing through the Gauss-Legendre quadrature points for the unit interval. For every time interval $[t^n, t^{n+1}]$, the map between the reference interval and the physical one is simply given by $t = t^n + \tau \Delta t^{n+1}$ for every $\tau \in [0, 1]$. Using the tensor product we can finally construct the basis functions on the space-time elements \mathbf{T}_i^{st} and \mathbf{R}_j^{st} such as $\hat{\phi}(\xi, \gamma, \delta, \tau) = \phi(\xi, \gamma, \delta) \cdot \gamma(\tau)$ and $\hat{\psi}^{(j)}(x, y, z, t) = \psi^{(j)}(x, y, z) \cdot \gamma(\tau(t))$. The total number of basis functions becomes $N_\phi^{st} = N_\phi \cdot N_\gamma$ and $N_\psi^{st} = N_\psi \cdot N_\gamma$.

4.2.3 Semi-Implicit DG scheme

The derivation of the three-dimensional equivalent of the method described in Chapter 3 can be done, thanks to the used structure and the vectorial formalism, exactly such as the one previously discussed in Section 3.2.3.

As a final result, the weak formulation of the system (4.1)-(4.2) reads

$$\sum_{j \in S_i} \mathcal{D}_{i,j} \hat{\mathbf{v}}_j^{n+1} = 0, \quad (4.7)$$

$$\mathbf{M}_j \hat{\mathbf{v}}_j^{n+1} - \mathbf{M}_j \widehat{\mathbf{F}} \hat{\mathbf{v}}_j + \mathcal{Q}_{r(j),j} \hat{\mathbf{p}}_{r(j)}^{n+1} + \mathcal{Q}_{\ell(j),j} \hat{\mathbf{p}}_{\ell(j)}^{n+1} = 0. \quad (4.8)$$

Formal substitution of the velocity vector field of the discrete momentum equation (4.8) into the continuity equation (4.7), as well as a Picard procedure in order to avoid nonlinearity, leads to a linear system for the scalar pressure as the only unknown. It reads

$$\begin{aligned} & \sum_{j \in S_i} \mathcal{D}_{i,j} \mathbf{M}_j^{-1} \mathcal{Q}_{i,j} \hat{\mathbf{p}}_i^{n+1, k+1} \\ & + \sum_{j \in S_i} \mathcal{D}_{i,j} \mathbf{M}_j^{-1} \mathcal{Q}_{\varphi(i,j),j} \hat{\mathbf{p}}_{\varphi(i,j)}^{n+1, k+1} = \sum_{j \in S_i} \mathcal{D}_{i,j} \widehat{\mathbf{F}} \hat{\mathbf{v}}_j^{n+1, k+\frac{1}{2}}. \end{aligned} \quad (4.9)$$

4.2.4 Nonlinear convection-diffusion

It remains to specify how to construct the nonlinear convective-diffusion operator $\widehat{\mathbf{F}} \hat{\mathbf{v}}_j^{n+\frac{1}{2}}$. At this point one can try to extend the procedure used in Section

3.2.4 to three space dimensions. In this case there are some issues that have to be taken into account. In particular, since we are using a modal base on the staggered dual hexahedral mesh, then we cannot use the simple nodal approximation for the nonlinear convective term $\hat{F}_c(\vec{v}) = F_c(\vec{v})$ that consists in a trivial point-wise evaluation of the nonlinear operator F_c . Furthermore if we also try to use a nodal polynomial base in the cases where this is possible (i.e. the system for the computation of the coefficients is not singular for each element), then this is still not a good approximation. Indeed observe how $\mathbf{F}_c(\vec{v}_h)$ lives in $V_h(2p)$. We can then locally project $\mathbf{F}_c(\vec{v}_h)$ to a higher order polynomial space and then use the L_2 projection on the subspace $V_h(p)$ in order to reconstruct $\hat{\mathbf{F}}_c$. In this way the coefficients of $\hat{\mathbf{F}}_c$ become optimal in the sense that they represent, by definition, the best $V_h(p)$ approximation of $\mathbf{F}_c(\vec{v}_h)$. In particular, after some computations, we get

$$\hat{\mathbf{F}}_c = \overline{\mathbf{M}}_j^{-1} \mathbf{T}_j \hat{v}_j \otimes \hat{v}_j, \quad (4.10)$$

where

$$\overline{\mathbf{M}}_j = \int_{\mathbf{R}_j^{st}} \tilde{\psi}_k^{(j)} \tilde{\psi}_l^{(j)} d\mathbf{x} dt, \quad \mathbf{T}_j = \int_{\mathbf{R}_j^{st}} \tilde{\psi}_k^{(j)} \tilde{\psi}_l^{(j)} \tilde{\psi}_m^{(j)} d\mathbf{x} dt. \quad (4.11)$$

From a computational point of view, this will require a lot of effort as well as a lot of memory usage in order to store the tensor \mathbf{T}_j for all $j \in [1, N_d]$. On the other hand it represents the natural way to discretize the nonlinear convective-viscous contribution since the velocity vector field lives on the dual mesh. Furthermore, in problems where we have discontinuities on the velocity at the boundary, we obtain by construction a discontinuity in the dual grid. This aspect will be clarified in Section 4.4. Following the same idea of [70] and inspired by the good properties achieved by the use of staggered grid, we propose a new procedure for the computation of the nonlinear convective-viscous term.

4.2.4.1 An alternative implementation for the viscous contribution

In this section we want to introduce an alternative treatment for the nonlinear convective and viscous contribution based on the high order projection of the velocity field on the main grid. This procedure was necessary in the formulation [70] since the velocity was splitted in a couple of staggered grids and here can be used as an alternative treatment of the nonlinear convective-viscous contribution.

An implicit discretization of the viscous terms based on the dual grid leads to a linear system for each velocity component that is a seven-point non symmetric block system but, however, is well conditioned since it can be written as a ν perturbation of the identity matrix, see e.g. Section 3.2.4. Here, we will develop a system for the viscous term that is a five-point one and, more important, is symmetric and at least positive definite for $\nu > 0$ and $p_\gamma = 0$ but is still better conditioned also in the case $p_\gamma > 0$.

In addition, the resulting computation of the nonlinear convective term on the main grid does not require to use the higher order projection as specified in Eq. (4.10). Hence, a faster computation can be performed using this strategy. Given a discrete velocity field \mathbf{v}_h on the dual grid in the time interval $[t^n, t^{n+1}]$, we can project the velocity field from the dual mesh to the main grid (denoted by $\bar{\mathbf{v}}$) via standard L_2 projection,

$$\bar{\mathbf{v}}_i^{n+1} = \mathbf{M}_i^{-1} \sum_{j \in S_i} \mathbf{M}_{i,j} \hat{\mathbf{v}}_j^{n+1}, \quad \forall i \in [1, N_e], \quad (4.12)$$

where $\bar{\mathbf{v}}_i^{n+1}$ denote the degrees of freedom of the velocity on the main grid and

$$\mathbf{M}_i = \int_{\mathbf{T}_i^{nst}} \tilde{\phi}_k^{(i)} \tilde{\phi}_l^{(i)} dxdt, \quad \mathbf{M}_{i,j} = \int_{\mathbf{T}_{i,j}^{nst}} \tilde{\phi}_k^{(i)} \tilde{\psi}_l^{(j)} dxdt. \quad (4.13)$$

The projection back onto the dual grid is given by

$$\hat{\mathbf{v}}_j^{n+1} = \bar{\mathbf{M}}_j^{-1} \left(\mathbf{M}_{\ell(j),j}^\top \bar{\mathbf{v}}_{\ell(j)}^{n+1} + \mathbf{M}_{r(j),j}^\top \bar{\mathbf{v}}_{r(j)}^{n+1} \right). \quad (4.14)$$

We can rewrite the nonlinear convective and viscous part of the momentum equation by introducing the viscous stress tensor $\boldsymbol{\sigma} = -\nu \nabla \mathbf{v}$ as auxiliary variable. The convective and viscous subsystem of the momentum equation then reads

$$\begin{aligned} \frac{\partial \mathbf{v}}{\partial t} + \nabla \cdot \mathbf{F}_c + \nabla \cdot \boldsymbol{\sigma} &= 0, \\ \boldsymbol{\sigma} &= -\nu \nabla \mathbf{v}. \end{aligned} \quad (4.15)$$

With the averaged velocity $\bar{\mathbf{v}}_i^{n+1} = \tilde{\phi}_l^{(i)} \bar{\mathbf{v}}_{l,i}^{n+1}$ defined on the main grid and the viscous stress tensor $\boldsymbol{\sigma}_j^{n+1} = \tilde{\psi}_l^{(j)} \boldsymbol{\sigma}_{l,j}^{n+1}$ defined on the dual grid, we obtain the

following weak formulation of (4.15):

$$\begin{aligned}
 & \int_{\mathbf{T}_i} \tilde{\phi}_k^{(i)}(\mathbf{x}, t^{n+1}) \bar{\mathbf{v}}_i^{n+1} d\mathbf{x} - \int_{\mathbf{T}_i} \tilde{\phi}_k^{(i)}(\mathbf{x}, t^n) \bar{\mathbf{v}}_i^n d\mathbf{x} - \int_{\mathbf{T}_i^{st}} \frac{\partial \tilde{\phi}_k^{(i)}}{\partial t} \bar{\mathbf{v}}_i^{n+1} d\mathbf{x} dt \\
 & + \int_{\partial \mathbf{T}_i^{st}} \tilde{\phi}_k^{(i)} \mathbf{F}_c^{\text{RS}}(\bar{\mathbf{v}}^-, \bar{\mathbf{v}}^+) \cdot \bar{\mathbf{n}}_i dS dt - \int_{\mathbf{T}_i^{st}} \nabla \tilde{\phi}_k^{(i)} \cdot \mathbf{F}_c(\bar{\mathbf{v}}_i^{n+1}) d\mathbf{x} dt \\
 & + \sum_{j \in S_i} \left(\int_{\Gamma_j^{st}} \tilde{\phi}_k^{(i)} \boldsymbol{\sigma}_j^{n+1} \cdot \bar{\mathbf{n}}_{ij} dS dt - \int_{\mathbf{T}_{i,j}^{st}} \nabla \tilde{\phi}_k^{(i)} \cdot \boldsymbol{\sigma}_j^{n+1} d\mathbf{x} dt \right) = 0, \\
 & \int_{\mathbf{R}_j^{st}} \tilde{\psi}_k^{(j)}(\mathbf{x}, t^{n+1}) \boldsymbol{\sigma}_j^{n+1} d\mathbf{x} = -\nu \left(\int_{\mathbf{T}_{\ell(j),j}^{st}} \tilde{\psi}_k^{(j)} \nabla \bar{\mathbf{v}}_{\ell(j)}^{n+1} d\mathbf{x} dt \right. \\
 & \left. + \int_{\mathbf{T}_{r(j),j}^{st}} \tilde{\psi}_k^{(j)} \nabla \bar{\mathbf{v}}_{r(j)}^{n+1} d\mathbf{x} dt + \int_{\Gamma_j^{st}} \tilde{\psi}_k^{(j)} (\bar{\mathbf{v}}_{r(j)}^{n+1} - \bar{\mathbf{v}}_{\ell(j)}^{n+1}) \otimes \bar{\mathbf{n}}_j dS dt \right). \quad (4.16)
 \end{aligned}$$

In a more compact matrix notation, (4.16) can be written as:

$$\begin{aligned}
 & (\bar{\mathbf{M}}_i^+ - \bar{\mathbf{M}}_i^o) \bar{\mathbf{v}}_i^{n+1} - \bar{\mathbf{M}}_i^- \bar{\mathbf{v}}_i^n + \sum_{j \in S_i} \mathcal{D}_{i,j} \boldsymbol{\sigma}_j^{n+1} + \bar{\Upsilon}_i^c = 0, \\
 & \bar{\mathbf{M}}_j \boldsymbol{\sigma}_j^{n+1} = -\nu \left(\mathcal{Q}_{\ell(j),j} \bar{\mathbf{v}}_{\ell(j)}^{n+1} + \mathcal{Q}_{r(j),j} \bar{\mathbf{v}}_{r(j)}^{n+1} \right), \quad (4.17)
 \end{aligned}$$

where

$$\bar{\mathbf{M}}_i^+ = \int_{\mathbf{T}_i} \tilde{\phi}_k^{(i)}(\mathbf{x}, t(1)) \tilde{\phi}_l^{(i)}(\mathbf{x}, t(1)) d\mathbf{x}, \quad (4.18)$$

$$\bar{\mathbf{M}}_i^- = \int_{\mathbf{T}_i} \tilde{\phi}_k^{(i)}(\mathbf{x}, t(0)) \tilde{\phi}_l^{(i)}(\mathbf{x}, t(1)) d\mathbf{x}, \quad (4.19)$$

$$\bar{\mathbf{M}}_i^o = \int_{\mathbf{T}_i^{st}} \frac{\partial \tilde{\phi}_k^{(i)}}{\partial t} \tilde{\phi}_l^{(i)} d\mathbf{x} dt. \quad (4.20)$$

$$(4.21)$$

In (4.17) we have defined the operator $\bar{\Upsilon}_i^c(\bar{\mathbf{v}})$, which is a standard DG discretization of the nonlinear convective terms on the tetrahedral elements of the main grid,

$$\bar{\Upsilon}_i^c(\bar{\mathbf{v}}) = \int_{\partial \mathbf{T}_i^{sst}} \tilde{\phi}_k^{(i)} \mathbf{F}_c^{\text{RS}}(\bar{\mathbf{v}}^-, \bar{\mathbf{v}}^+) \cdot \bar{\mathbf{n}}_i dSdt - \int_{\mathbf{T}_i^{sst}} \nabla \tilde{\phi}_k^{(i)} \cdot \mathbf{F}_c(\bar{\mathbf{v}}) dxdt, \quad (4.22)$$

with the the boundary extrapolated values \mathbf{v}^- and \mathbf{v}^+ from within the cell and from the neighbors, respectively. Here, the approximate Riemann solver \mathbf{F}_c^{RS} used at the element boundaries is given by the simple Rusanov flux [137]

$$\mathbf{F}_c^{\text{RS}}(\bar{\mathbf{v}}^-, \bar{\mathbf{v}}^+) \cdot \bar{\mathbf{n}}_i = \frac{1}{2} (\mathbf{F}_c(\bar{\mathbf{v}}^+) + \mathbf{F}_c(\bar{\mathbf{v}}^-)) \cdot \bar{\mathbf{n}}_i - \frac{1}{2} s_{\max} (\bar{\mathbf{v}}^+ - \bar{\mathbf{v}}^-), \quad (4.23)$$

where $s_{\max} = 2 \max(|\bar{\mathbf{v}}^+|, |\bar{\mathbf{v}}^-|)$ is the maximum eigenvalue of the convective operator \mathbf{F}_c . The final system for the variable $\bar{\mathbf{v}}$ can be found by formal substitution of $\boldsymbol{\sigma}$ given in the second equation of (4.17) into the first one:

$$\begin{aligned} & \left(\bar{\mathbf{M}}_i - \nu \sum_{j \in S_i} \mathcal{D}_{i,j} \bar{\mathbf{M}}_j^{-1} \mathcal{Q}_{i,j} \right) \bar{\mathbf{v}}_i^{n+1} \\ & - \nu \sum_{j \in S_i} \mathcal{D}_{i,j} \bar{\mathbf{M}}_j^{-1} \mathcal{Q}_{\varphi(i,j),j} \bar{\mathbf{v}}_{\varphi(i,j)}^{n+1} \\ & = \bar{\mathbf{M}}_i^- \bar{\mathbf{v}}_i^n - \bar{\Upsilon}_i^c(\bar{\mathbf{v}}^{n+1}), \end{aligned} \quad (4.24)$$

where we use the abbreviation $\bar{\mathbf{M}}_i = \bar{\mathbf{M}}_i^+ - \bar{\mathbf{M}}_i^o$. What we obtain is a discretization of the nonlinear convective and viscous terms on the main grid, where the stress tensor $\boldsymbol{\sigma}$ has been computed on the face-based dual mesh. In order to avoid the solution of a nonlinear system due to the nonlinear operator $\bar{\Upsilon}_i^c(\bar{\mathbf{v}}^{n+1})$, we introduce a fractional step scheme combined with an outer Picard iteration. Using the notation introduced in Section 3.2.3, we get

$$\begin{aligned} & \left(\bar{\mathbf{M}}_i - \nu \sum_{j \in S_i} \mathcal{D}_{i,j} \bar{\mathbf{M}}_j^{-1} \mathcal{Q}_{i,j} \right) \bar{\mathbf{v}}_i^{n+1, k + \frac{1}{2}} \\ & - \nu \sum_{j \in S_i} \mathcal{D}_{i,j} \bar{\mathbf{M}}_j^{-1} \mathcal{Q}_{\varphi(i,j),j} \bar{\mathbf{v}}_{\varphi(i,j)}^{n+1, k + \frac{1}{2}} \\ & = \bar{\mathbf{M}}_i^- \bar{\mathbf{v}}_i^n - \bar{\Upsilon}_i^c(\bar{\mathbf{v}}^{n+1, k}). \end{aligned} \quad (4.25)$$

4.2.5 Extension to a pressure correction formulation

As already discussed in Section 3.2.5, the computation of the nonlinear convective and viscous terms presented in Eq. (4.25) does not depend explicitly on the pressure of the previous Picard iteration, and hence it does not see the effect of the pressure in the time interval T^{n+1} , which is, however, needed to get a high order accurate scheme also in time. In order to overcome the problem we introduce directly in Eq. (4.25) the contribution of the pressure at the previous Picard iteration. Then, we update the velocity with the new pressure $\hat{\mathbf{p}}_i^{n+1,k+1}$. The equations (4.25), (4.9), (4.8) become:

$$\begin{aligned} & \left[\overline{\mathbf{M}}_i - \nu \sum_{j \in S_i} \mathcal{D}_{i,j} \overline{\mathbf{M}}_j^{-1} \mathcal{Q}_{i,j} \right] \overline{\mathbf{v}}_i^{n+1,k+\frac{1}{2}} \\ & \quad - \nu \sum_{j \in S_i} \mathcal{D}_{i,j} \overline{\mathbf{M}}_j^{-1} \mathcal{Q}_{\varphi(i,j),j} \overline{\mathbf{v}}_{\varphi(i,j)}^{n+1,k+\frac{1}{2}} \\ & = \overline{\mathbf{M}}_i \overline{\mathbf{v}}_i^n - \Upsilon_j^c \left(\overline{\mathbf{v}}^{n+1,k} \right) - \overline{\mathbf{M}}_i \Lambda \left(\hat{\mathbf{p}}^{n+1,k} \right), \end{aligned} \quad (4.26)$$

$$\Lambda \left(\hat{\mathbf{p}}^{n+1,k} \right) = \mathbf{M}_i^{-1} \sum_{j \in S_i} \mathbf{M}_{i,j} \left(\mathbf{M}_j^{-1} \left(\mathcal{Q}_{r(j),j} \hat{\mathbf{p}}_{r(j)}^{n+1,k} + \mathcal{Q}_{\ell(j),j} \hat{\mathbf{p}}_{\ell(j)}^{n+1,k} \right) \right), \quad (4.27)$$

$$\begin{aligned} & \sum_{j \in S_i} \mathcal{D}_{i,j} \mathbf{M}_j^{-1} \mathcal{Q}_{i,j} \left(\hat{\mathbf{p}}_i^{n+1,k+1} - \hat{\mathbf{p}}_i^{n+1,k+1} \right) \\ & + \sum_{j \in S_i} \mathcal{D}_{i,j} \mathbf{M}_j^{-1} \mathcal{Q}_{\varphi(i,j),j} \left(\hat{\mathbf{p}}_{\varphi(i,j)}^{n+1,k+1} - \hat{\mathbf{p}}_{\varphi(i,j)}^{n+1,k+1} \right) = \sum_{j \in S_i} \mathcal{D}_{i,j} \widehat{\mathbf{F}} \overline{\mathbf{v}}_j^{n+1,k+\frac{1}{2}}, \end{aligned} \quad (4.28)$$

$$\begin{aligned} \widehat{\mathbf{v}}_j^{n+1,k+1} & = \widehat{\mathbf{F}} \overline{\mathbf{v}}_j^{n+1,k+\frac{1}{2}} \\ & \quad - \mathbf{M}_j^{-1} \left[\mathcal{Q}_{r(j),j} \left(\hat{\mathbf{p}}_{r(j)}^{n+1,k+1} - \hat{\mathbf{p}}_{r(j)}^{n+1,k} \right) \right. \\ & \quad \left. + \mathcal{Q}_{\ell(j),j} \left(\hat{\mathbf{p}}_{\ell(j)}^{n+1,k+1} - \hat{\mathbf{p}}_{\ell(j)}^{n+1,k} \right) \right], \end{aligned} \quad (4.29)$$

where $\mathbf{\Lambda}(\hat{\mathbf{p}}^{n+1,k})$ represents the same additional contribution subtracted in (4.28) that lives on the dual mesh passed through the mean maps from the dual to the main grid.

As initial guess for the pressure we take $\hat{\mathbf{p}}^{n+1,0} = 0$ so that we obtain the classical first order method such as in [144] for the first Picard iteration. Eventually, we can take the Lagrange extrapolation of $\hat{\mathbf{p}}^n$ on T^{n+1} in order to reduce the number of Picard iterations.

Note how, if we use the natural extension of the algorithm presented in Section 3.2.4 to compute the nonlinear convective-viscous term, then we recover the stability property as a corollary of Theorem 1 presented in Section 3.5. Otherwise a stability proof has to take into account the high L_2 -projections (4.12) and (4.14) that in general do not act as the identity, for instance

$$\xi_i \neq \mathbf{M}_i^{-1} \sum_{j \in S_i} \mathbf{M}_{i,j} \mathbf{M}_j^{-1} \left[\mathbf{M}_{\ell(j),j}^\top \xi_{\ell(j)} + \mathbf{M}_{r(j),j}^\top \xi_{r(j)} \right]$$

for ξ defined on the main grid and vice versa. We can still recover a stability result in the case of a projection that acts as the identity since we are sure that the additional pressure term introduced at Picard iteration k does not change using the forward and backward mean. Indeed it does not introduce any additional kinetic energy.

4.2.6 Remarks on the special case of piecewise constant polynomials in time ($p_\gamma = 0$)

The method presented in the previous sections can be seen, for $p_\gamma = 0$, as the extension of the one presented in Section 3.3 to three space dimensions. This particular case is, in general, only first order accurate in time but high order accurate in space. In this case, we can recover several good properties for the main system for the pressure and for the linear systems that need to be solved for the implicit discretization of the viscous terms.

4.2.6.1 Pressure system

For $p_\gamma = 0$ we have $\mathbf{M}_j^\circ = 0$ then $\mathbf{M}_j = \mathbf{M}_j^+ = \mathbf{M}_j^-$ is symmetric for all $j \in 1 \dots N_d$. Consequently, the system (4.7)-(4.8) formally becomes the same method as in Section 3.3. The following results can therefore be readily obtained as corollaries of the theorems given in Section 3.3 regarding the system matrix \mathcal{A} of the main system for the pressure (4.9):

Corollary 1 (Symmetry) *Let $p_\gamma = 0$, the system matrix \mathcal{A} of the main system for the pressure is symmetric.*

Corollary 2 (Positive semi-definiteness) *Let $p_\gamma = 0$, the system matrix \mathcal{A} of the main system for the pressure is in general positive semi-definite.*

This means that in this particular case we can use faster iterative linear solvers, like the conjugate gradient (CG) method [98] to solve the main system for the pressure (4.9). This advantage makes the case $p_\gamma = 0$ particularly suitable for steady or almost steady problems. In order to recover some precision in time we can extend the algorithm by introducing a semi-implicit discretization, as suggested in Section 3.3. In this case, system (4.7)-(4.8) has to be discretized as

$$\sum_{j \in S_i} \mathcal{D}_{i,j} \hat{\mathbf{v}}_j^{n+1} = 0, \quad (4.30)$$

$$\mathbf{M}_j \hat{\mathbf{v}}_j^{n+1} - \mathbf{M}_j \widehat{\mathbf{F}} \hat{\mathbf{v}}_j^n + \Delta t \mathcal{Q}_{r(j),j} \hat{\mathbf{p}}_{r(j)}^{n+\theta} + \Delta t \mathcal{Q}_{\ell(j),j} \hat{\mathbf{p}}_{\ell(j)}^{n+\theta} = 0, \quad (4.31)$$

where $\hat{\mathbf{p}}^{n+\theta} = \theta \hat{\mathbf{p}}^{n+1} + (1 - \theta) \hat{\mathbf{p}}^n$ and θ is an implicitness factor to be taken in the range $\theta \in [\frac{1}{2}, 1]$, see e.g. [36]. For $\theta = \frac{1}{2}$, the Crank-Nicolson method is recovered. In this way we gain some extra precision in time without affecting the computational effort and using the same advantages given by Corollary 1 and 2 that can be easily extended for this case.

4.2.6.2 Viscous system

In the special case of piecewise constant polynomials in time ($p_\gamma = 0$), we get $\overline{\mathbf{M}}_i = \mathbf{M}_i$ and $\overline{\mathbf{M}}_j = \mathbf{M}_j$, so that the following results about the viscous system (4.25) can be derived:

Corollary 3 (Symmetry) *If $p_\gamma = 0$ then the system (4.25) is symmetric.*

Proof 5 *We can write the system matrix of system (4.25) as $(M + \nu \mathcal{A})$, where M is a block diagonal matrix with $\{\mathbf{M}_i\}_{i=1 \dots N_e}$ on the diagonal and \mathcal{A} is the matrix of the pressure system (4.9). Thanks to the results obtained in Corollary 1, \mathcal{A} is symmetric and also M is symmetric, since $\mathbf{M}_i = \mathbf{M}_i^\top$, see (4.13).*

Corollary 4 (Positive definiteness) *If $p_\gamma = 0$ then the system (4.25) is positive definite.*

Proof 6 As used in Corollary 3, we can write the system such as $M + \nu\mathcal{A}$ and we know, thanks to Corollary 2, that \mathcal{A} is in general positive semi-definite. A simple computation leads to

$$x(M + \nu\mathcal{A})x^\top = xMx^\top + \nu x\mathcal{A}x^\top > 0 \quad (4.32)$$

since $\nu x\mathcal{A}x^\top \geq 0$ and $xMx^\top > 0$ we have that the complete system is also positive definite.

In the general case of $p_\gamma > 0$ it is not true that we recover the pressure system, since $\overline{M} \neq M$. In this case, we can observe how the non symmetric contribution affects only \overline{M}_i . This allows us to write the previous system as $T + \nu H$ where T is a block diagonal non symmetric matrix and H is symmetric and positive semi-definite.

4.2.7 Extension to curved elements

The method described in the previous sections can readily be generalized by introducing also curved elements inside the computational domain. This generalization will affect only the pre-processing step. The extension is quite similar to the one introduced in Section 2.2.7 for the two dimensional case, but there are some differences due to the three dimensionality of the problem.

First of all, in the two dimensional case one could eventually consider curved only the main elements that touch the boundary, as well as all the dual elements such that $j \in \mathcal{B}(\Omega)$. In the 3D case this usually leads to a nonconforming mesh, so we have to generalize and move accordingly also the internal elements (see for example Fig.4.3). Each tetrahedral main and dual element is then characterized by N_ϕ nodes $\{(X, Y)_k\}_{k=1, N_\phi}$ while for internal hexahedral elements we have to split it into a left and a right tetrahedral, i.e. $\mathbf{R}_j^{iso} = \mathbf{T}_{\ell(j),j}^{iso} \cup \mathbf{T}_{r(j),j}^{iso}$ and the points that lie on Γ_j^{iso} are physically joined. In this way we have a fully characterization of the left and the right tetrahedral, needed to compute properly the integral contributions in the algorithm.

In order to move the entire grid we mount a tetrahelization that involves all the DoFs inside the domain and we solve the Laplace equation for the displacement using a classical $P1$ continuous finite element method and imposing the projection on the physical domain as boundary conditions. This procedure give us an elastic movement of each DoF inside the domain.

As shown in [144], the possibility to curve the grid is crucial when we try to represent complex domains with a very coarse grid. In any case, we emphasize

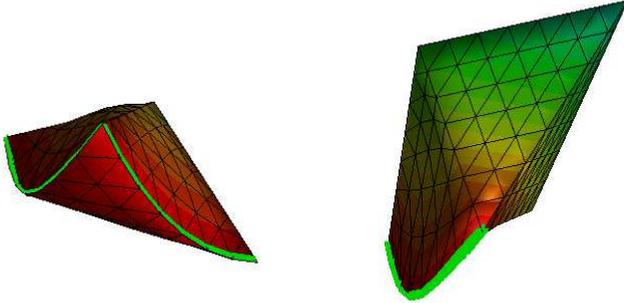


Figure 4.3: Isoparametric dual element example for $p = 3$. On the left a dual element with a $2D$ face on the curved boundary; On the right, an internal dual element, but with a $1D$ edge on the curved boundary.

that this generalization does not affect the computational cost during the runtime since it interests only the construction of the main matrices that can be done in a preprocessing step. For a more detailed discussion see appendix section A.3.

4.3 Numerical test problems

4.3.1 Three-dimensional lid driven cavity

We present in this section some results regarding the three-dimensional lid-driven cavity. For the two dimensional case there are a lot of well known results and reference solutions (see e.g. Ghia et al. and Erturk et al. [89, 80]) as well as for the three-dimensional case (see e.g. [2, 115, 1]). We take here a classical cubic cavity $\Omega = [-0.5, 0.5]^3$ and we cover it with a very coarse tetrahedral mesh with characteristic size $h = 0.2$. We set as initial conditions $p = 1$; $u = v = w = 0$. As boundary conditions we impose velocity $(u, v, w) = (1, 0, 0)$ at $y = 0.5$ while no-slip boundary conditions are used on the other boundaries. Since we are interested in steady state solutions, we take for the current test $p = 4$; $p_\gamma = 0$; $\theta = 1.0$ and several kinematic viscosity in order to obtain several

Reynolds numbers.

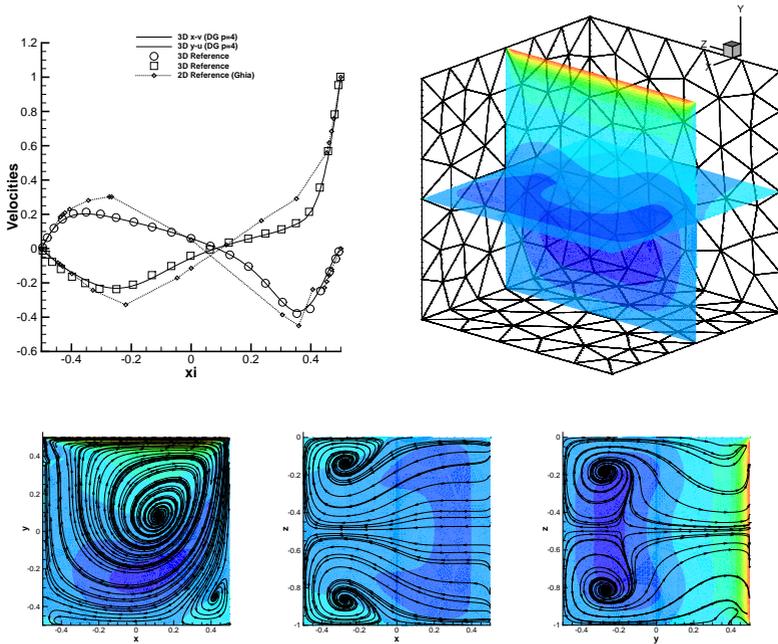


Figure 4.4: From top left to bottom right: Comparison between our numerical results, the one obtained by Albensoeder et al in [2], and the two dimensional data from Ghia et al [89] for $Re = 400$; three-dimensional plot of the two secondary slices and grid space; streamlines and magnitude of u on slices $x - y$, $x - z$ and $y - z$.

In Figure 4.4 are shown the results at time $t_{end} = 30s$ for $Re = 400$. In Figure 4.5 the same plots are given for $t_{end} = 40s$ and $Re = 1000$. In the top left plot are reported our numerical results against the one obtained in [2] for the fully three-dimensional case and the data given by Ghia et al for the two dimensional cavity at the same Reynolds number. A good agreement is achieved also if a very coarse grid has been used. The data show that the presence of the third

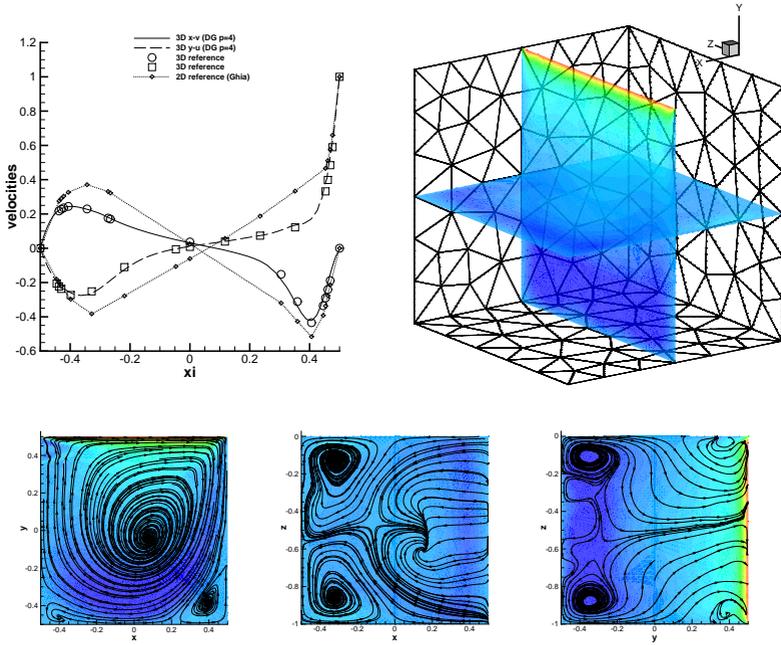


Figure 4.5: From top left to bottom right: Comparison between our numerical results, the one obtained by Albensoeder et al in [2], and the two dimensional data from Ghia et al [89] for $Re = 1000$; three-dimensional plot of the two secondary slices and grid space; streamlines and magnitude of u on slices $x - y$, $x - z$ and $y - z$.

axis significantly modifies the velocity profiles. Furthermore, several Taylor-Görtler like vortices appear in the secondary planes in a very similar way such as observed in other numerical and experimental investigations of this problem (see e.g. [115, 1]).

4.3.2 Convergence test

In this test we will investigate the Arnold-Beltrami-Childress flow that was originally introduced by Arnold in [4] and Childress in [46] as an interesting class of Beltrami flows and successively studied in a series of papers, see e.g. [66, 130, 129, 78]. In particular we consider:

$$\begin{aligned}
 u(x, y, z, t) &= [\sin(z) + \cos(y)] e^{-\nu t}, \\
 v(x, y, z, t) &= [\sin(x) + \cos(z)] e^{-\nu t}, \\
 w(x, y, z, t) &= [\sin(y) + \cos(x)] e^{-\nu t}, \\
 p(x, y, z, t) &= -[\cos(x) \sin(y) + \sin(x) \cos(z) + \sin(z) \cos(y)] e^{-2\nu t} + c
 \end{aligned}
 \tag{4.33}$$

where $c \in \mathbb{R}$. One can check that this is an exact solution for the complete three dimensional incompressible Navier-Stokes equations in a periodic domain, so this smooth configuration is suitable for several convergence tests. In particular if $\nu = 0$ we can check the accuracy of the spatial part of the algorithm, i.e. $p_\gamma = 0$, since the solution is a steady one. We take as computational domain $\Omega = [-\pi, \pi]^3$ and we extend it using periodic boundary conditions everywhere. Furthermore we take increasing values of p and refinement factors from a starting regular mesh and up to $t_{end} = 0.1$. The chosen dt is taken into account with the maximum one allowed by the CFL time restriction for the nonlinear convection. Since we have periodic boundary condition everywhere, we have a set of solutions for the pressure given by (4.33) up to a constant. In order to verify that also the pressure field is correct, we choose c in (4.33) a posteriori according to the mean value of the resulting numerical pressure.

The resulting vorticity, pressure and streamlines are plotted in Figure 4.6 while in Table 4.1 are reported the resulting L_2 norm of the error for the steady case $\nu = 0$. We observe how the optimal order of convergence is obtained for this steady problem for the pressure, while a p order of convergence can be observed for the velocity field.

In the second test case we turn on the viscosity in order to make the problem unsteady. For this kind of problem we use the space-time DG implementation of the algorithm and we set the number of Picard iterations to $N_{pic} = p_\gamma + 1$. Unfortunately, as soon as we use an high order polynomial in time, the resulting main system loses the symmetry property and hence we have to use a slower linear solver such as the GMRES method. Since the viscosity contribution is discretized implicitly, we can take very large values for the kinematic viscosity and maintain the same CFL time restriction for the simulation. The chosen

Table 4.1: Numerical convergence results.

p	p_γ	N_e	$\epsilon(p)$	$\epsilon(\mathbf{v})$	$\sigma(p)$	$\sigma(\mathbf{v})$
1	0	7986	7.4349E-01	3.7768E-01	-	-
1	0	10368	6.2638E-01	3.1662E-01	2.0	2.0
1	0	13182	5.3318E-01	2.7046E-01	2.0	2.0
1	0	16464	4.6155E-01	2.3309E-01	2.0	2.0
2	0	7986	8.6472E-02	5.0920E-02	3.0	2.4
2	0	10368	6.7178E-02	4.1417E-02	2.9	2.4
2	0	13182	5.2651E-02	3.4271E-02	3.0	2.4
2	0	16464	4.2520E-02	2.8499E-02	2.9	2.5
3	0	7986	6.6133E-03	3.5899E-03	3.9	3.4
3	0	10368	4.7069E-03	2.6619E-03	3.9	3.4
3	0	13182	3.4219E-03	2.0294E-03	4.0	3.4
3	0	16464	2.5604E-03	1.5727E-03	3.9	3.4
4	0	6000	8.4806E-04	6.7156E-04	4.9	4.1
4	0	7986	5.3156E-04	4.5361E-04	4.9	4.1
4	0	10368	3.4667E-04	3.1585E-04	4.9	4.2
4	0	13182	2.3307E-04	2.2733E-04	5.0	4.1
5	0	4374	1.5777E-04	1.6300E-04	5.9	5.1
5	0	6000	8.4744E-05	9.4463E-05	5.9	5.2
5	0	7986	4.8228E-05	5.7433E-05	5.9	5.2
5	0	10368	2.8868E-05	3.6318E-05	5.9	5.2

viscosity for this test is $\nu = 1$ and we test it for $p = p_\gamma = 1 \dots 4$ and successively refined grids. The resulting convergence rates, as well as the L_2 errors, are shown in Table 4.2. In this case a order of $p + \frac{1}{2}$ is achieved for the pressure, while order $p + 1$ can be observed for the velocity.

Table 4.2: Numerical convergence results.

p	p_γ	N_e	$\epsilon(p)$	$\epsilon(\mathbf{v})$	$\sigma(p)$	$\sigma(\mathbf{v})$
1	1	10368	1.1713E+00	2.4695E-01	1.6	2.0
1	1	13182	1.0388E+00	2.1017E-01	1.5	2.0
1	1	16464	9.2718E-01	1.8075E-01	1.5	2.0
1	1	20250	8.3860E-01	1.5730E-01	1.5	2.0
2	2	10368	1.7339E-01	1.4475E-02	2.8	3.1
2	2	13182	1.4060E-01	1.1291E-02	2.6	3.1
2	2	16464	1.1470E-01	8.9676E-03	2.8	3.1
2	2	20250	9.5780E-02	7.2516E-03	2.6	3.1
3	3	6000	1.6219E-02	1.5469E-03	3.8	4.1
3	3	7986	1.1454E-02	1.0494E-03	3.7	4.1
3	3	10368	8.2191E-03	7.3591E-04	3.8	4.1
3	3	13182	6.1399E-03	5.3142E-04	3.6	4.1
4	4	750	4.5578E-02	3.2574E-03	4.7	4.8
4	4	1296	1.9664E-02	1.2957E-03	4.6	5.1
4	4	2058	9.3757E-03	5.9049E-04	4.8	5.1
4	4	3072	5.0553E-03	2.9738E-04	4.6	5.1

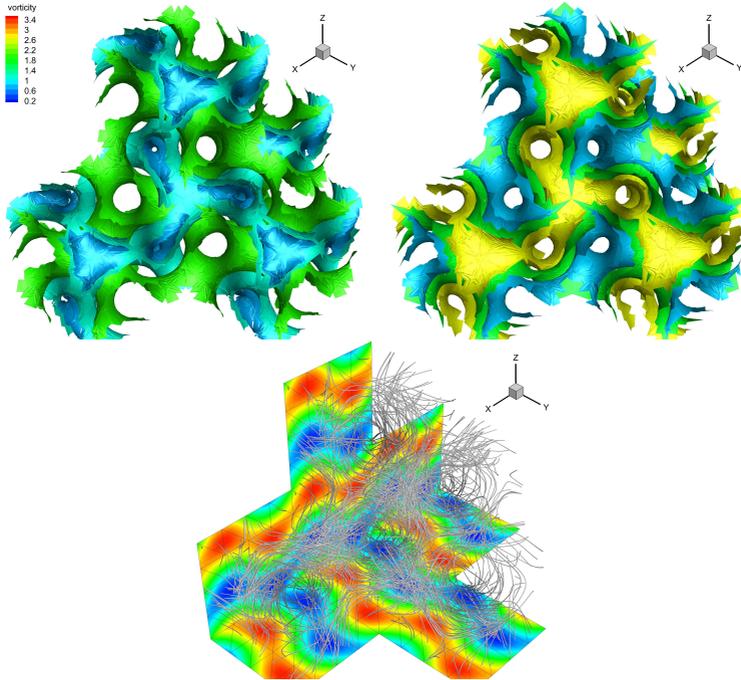


Figure 4.6: From top left to bottom right: Vorticity isolines for $vort = [0.8, 1.2, 2.0]$; pressure at $p = [-0.8, 0.0, 0.8]$ and streamlines in order to show the three-dimensionality of the considered problem.

4.3.3 Taylor Green Vortex

We investigate in this section another classical benchmark such as the Taylor Green vortex. In this test case a very simple initial analytical solution degenerates quickly to a turbulent flow with a very complex structure. We take the initial condition as given in [58]:

$$\begin{aligned}u(x, y, z, t) &= \sin(x) \cos(y) \cos(z), \\v(x, y, z, t) &= -\cos(x) \sin(y) \cos(z), \\w(x, y, z, t) &= 0, \\p(x, y, z, t) &= p_0 + \frac{1}{16} (\cos(2x) + \cos(2y)) (\cos(2z) + 2),\end{aligned}\quad (4.34)$$

in $\Omega = [\pi, \pi]^3$ and periodic boundary conditions everywhere. As numerical parameters we take $(p, p_\gamma) = (4, 0)$; $N_i = 494592$ tetrahedra; $\theta = 0.51$; dt according to the CFL time restriction; $t_{end} = 10$; and several values of ν so that the Reynolds numbers are $Re = 400$, $Re = 800$ and $Re = 1600$, respectively.

A plot of the time evolution of the pressure field, the velocity magnitude and the vorticity pattern is shown in Figure 4.7 for several times, as well as time series of the total kinematic dissipation rates compared with available DNS data given by Brachet et al in [21] in Figure 4.9. A good agreement between reference data and our numerical results can be observed. In Figure 4.7 the vorticity pattern shows a really complex behavior that appears after few seconds.

As shown in Figure 4.8 the vorticity pattern becomes very complex close to $t = 9s$. Some details of this pattern for several slice positions are shown in Figures 4.10 and 4.11 and underlining several vortical structures that are difficult to be seen from the three-dimensional plot. I underline how in this particular test, it is important to resolve well the small scales that, close to $t = 9s$, constitute the main contribution to the total kinetic energy dissipation. The typical number of iterations needed to solve the main linear system for the pressure at $Re = 1600$ and $tol = 10^{-8}$ is $I_{mean} = 290.7$. In general we observe a number of iterations in the range $I \in [93, 2516]$.

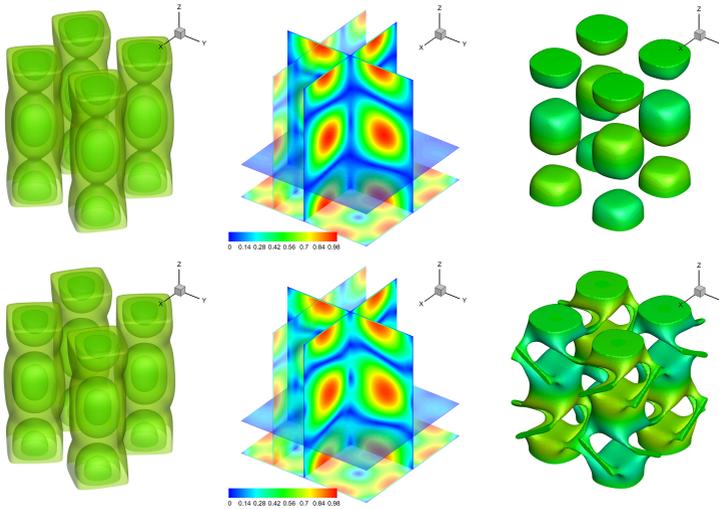


Figure 4.7: From the top to the bottom: Pressure isolines, velocity magnitude and vorticity isolines from left to right at times $t = [0.5, 1.0]$ and for $Re = 800$.

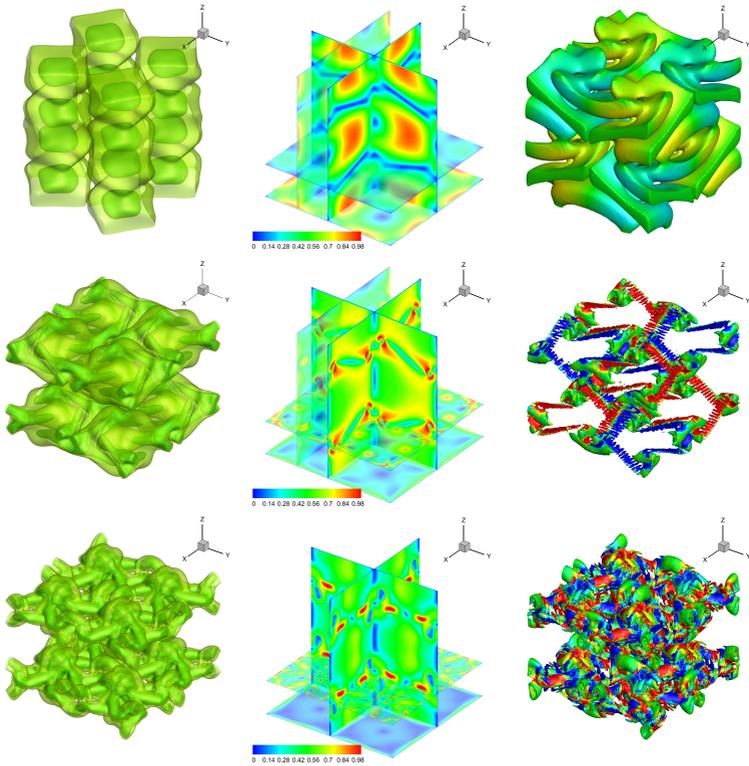


Figure 4.8: From the top to the bottom: Pressure isolines, velocity magnitude and vorticity isolines from left to right at times $t = [2.1, 4.8, 9.0]$ and for $Re = 800$.

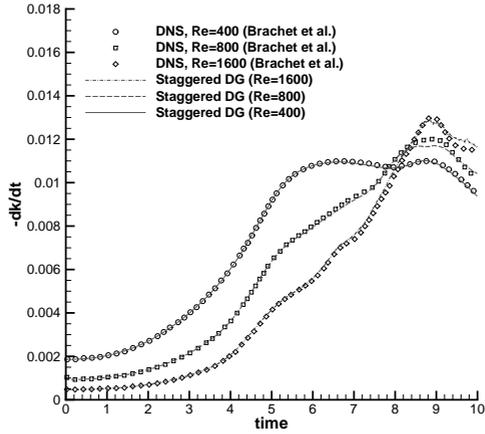


Figure 4.9: Time evolution of the kinetic energy dissipation rate $-dk/dt$ compared with DNS available data of Brachet et al [21] for $Re = 400, 800$ and $Re = 1600$.

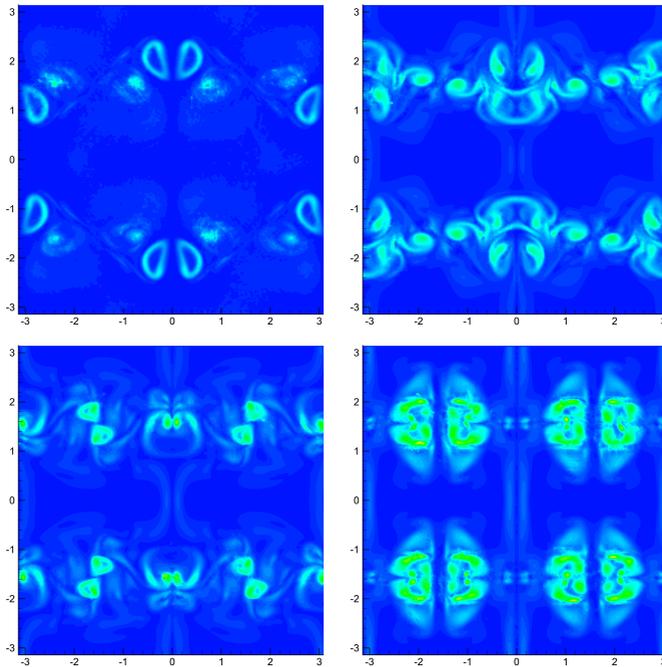


Figure 4.10: Vorticity pattern in the plane $x = 0, \frac{\pi}{8}, \frac{\pi}{4}, \frac{\pi}{2}$ from top left to bottom right for $Re = 800, t = 9s$.

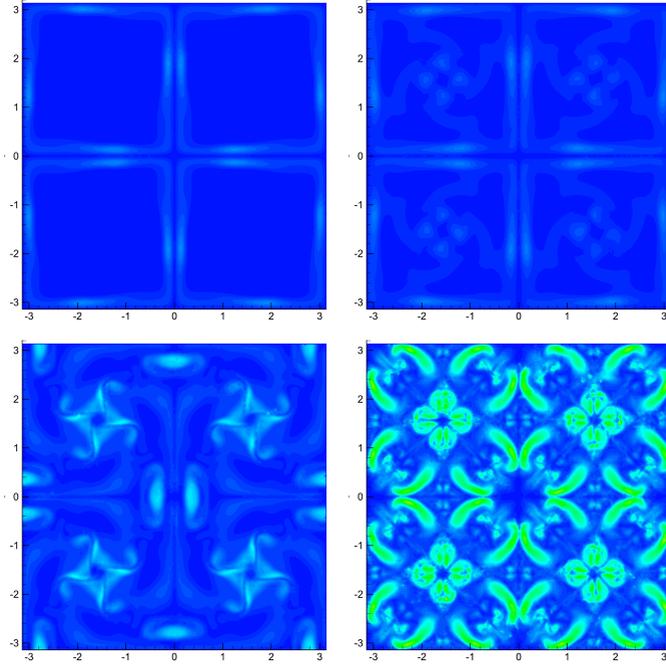


Figure 4.11: Vorticity pattern in the plane $z = 0, \frac{\pi}{8}, \frac{\pi}{4}, \frac{\pi}{2}$ from top left to bottom right for $Re = 800, t = 9s$.

4.3.4 Womersley Profile

In this section the proposed algorithm is verified against the exact solution for an oscillating flow in a rigid tube with circular cross section of length L . The unsteady flow is driven by a sinusoidal pressure gradient on the inlet and outlet boundaries

$$\frac{p_{out}(t) - p_{inlet}(t)}{L} = \frac{\tilde{p}}{\rho} e^{i\omega t}, \quad (4.35)$$

where \tilde{p} is the amplitude of the pressure gradient; ρ is the fluid density; ω is the frequency of the oscillation; i indicates the imaginary unit; p_{inlet} and p_{out} are the inlet and outlet pressures, respectively. The analytical solution was derived by Womersley in [158]. According to [158, 81] no convective contribution is considered. By imposing Eq. (4.35) at the tube ends, the resulting unsteady velocity field is uniform in the axial direction and is given by

$$u_e(x, y, t) = \frac{\tilde{p}}{\rho} \frac{1}{i\omega} \left[1 - \frac{J_0\left(\alpha\zeta i^{\frac{3}{2}}\right)}{J_0\left(\alpha i^{\frac{3}{2}}\right)} \right] e^{i\omega t}; \quad v_e(x, y, t) = 0, \quad (4.36)$$

where $\zeta = 2y/D$ is the dimensionless radial coordinate; D is the diameter of the tube; $\alpha = \frac{D}{2} \sqrt{\frac{\omega}{\nu}}$ is a constant; and J_0 is the zero-th order Bessel function of the first kind. For the present test we take Ω as a cylinder of length 4 and radius 1; $\tilde{p} = 1000$; $\rho = 1000$; $\omega = 2\pi$; and $\nu = 0.04$. The computational domain Ω is covered with a total number of $N_e = 1185$ tetrahedra and the time step size is chosen as $dt = 0.3$ that is 30% of the entire period. For this test we take $(p, p_\gamma) = (4, 3)$ in order to produce a good solution also with the chosen dt .

Due to the curved geometry of the problem we use a fully isoparametric approach to fit the cylinder. A plot of the isoparametric grid that has been used here is reported in Figure 4.14 on the left. We test our numerical solution in the cutting slice $\Gamma = \{x = 2\}$ and successively on the line given by $(x, z) = (2, 0) \in \Gamma$. Figure 4.12 shows the evolution of the velocity profile u on Γ solved in a single time cell $\Gamma^{st}(t, \vec{x}) = \Gamma(\vec{x}) \times [0.3, 0.6]$ evaluated at several intermediate times. A comparison between numerical and exact solution is reported in Figure 4.13 as well as the plot of Γ , in order to show the radial symmetry of the solution, that is not trivial for the chosen discretization. Finally, a plot of the time series of the velocity u computed in $\vec{x} = (1, 0, 0)$ and $\vec{x} = (1, 0, 0.9)$ and compared with the exact one is reported in Fig. 4.14 right. It is clear from Figures 4.13 and 4.14 that this test with the chosen time step can reproduce good results only if we use high order polynomials also in

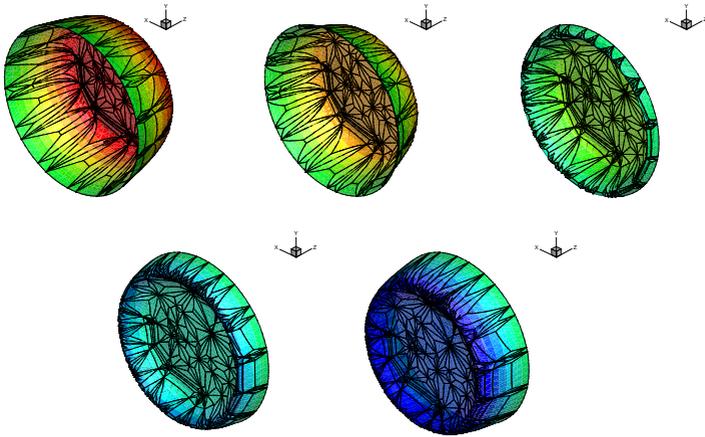


Figure 4.12: Plot of u at the middle of the tube in *one single* time control volume $T = [0.3, 0.6]$, from top left to bottom right are plotted at intermediate time levels $t = [0.3, 0.375, 0.45, 0.525, 0.6]$.

time; indeed, the solution for a first order method in time would look piecewise constant within each time step.

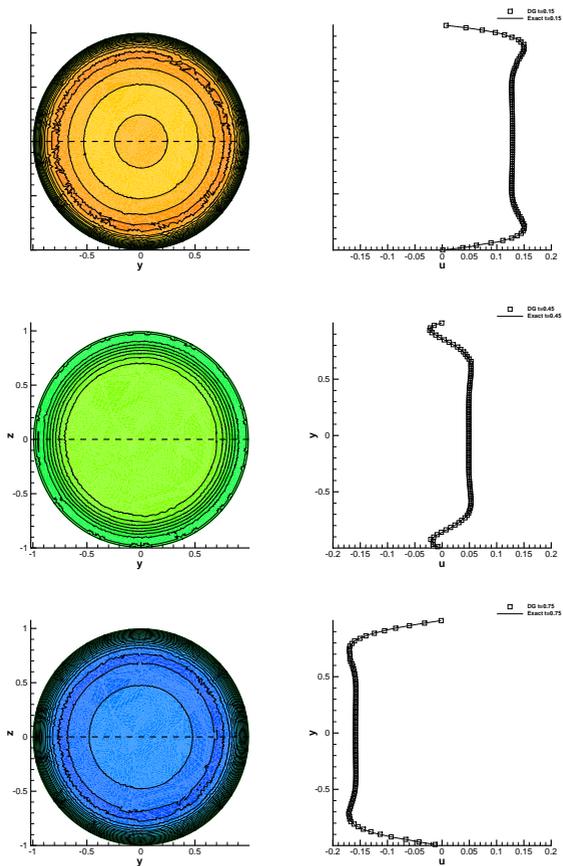


Figure 4.13: Velocity profile at the plane $x = 2$ (left column) and velocity against exact solution at $x = 2$ and $z = 0$ (right column) at times, from top to bottom, $t = [0.15, 0.45, 0.75]$.

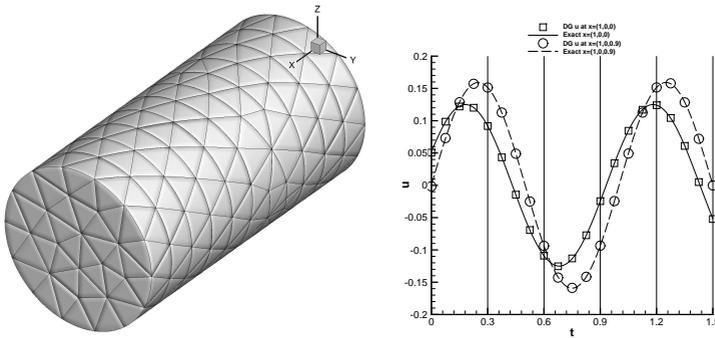


Figure 4.14: Left: Three dimensional view of the isoparametric grid used in the test case; Right: Time series of u in the plane $x = 1$, $(y, z) = (0, 0)$ and $(y, z) = (0, 0.9)$. The time subdivisions represent the time step size $dt = 0.3$.

4.3.5 Blasius boundary layer

We consider here a classical benchmark for incompressible fluids that is the Blasius boundary layer. The framework of this numerical test is the same as in the test 3.6.1.2 but it is done here in three space dimensions. In order to obtain the Blasius velocity profile in our simulations we consider a steady flow over a flat plate. As a result of the viscosity, a boundary layer appears along the no-slip wall. For the current test, we consider $\Omega = [-0.2, 0.8] \times [-0.2, 0.2]^2$. An initially uniform flow $u(x, y, z, 0) = u_\infty = 1$, $v(x, y, z, 0) = w(x, y, z, 0) = 0$ and $p(x, y, 0) = 1$ is imposed as initial condition, while an inflow boundary is imposed on the left boundary; no slip boundary condition is considered in the flat plane $\Gamma = \{(x, y, z) \mid x \geq 0, y = y_{\min}\}$; slip boundary conditions are imposed at $z = z_{\min}$ and $z = z_{\max}$; and transmissive boundary conditions are imposed at the upper face $y = y_{\max}$. We consider here an extreme case where we cover our domain Ω with a set of only $N_e = 1522$ tetrahedra whose characteristic length is $h = 0.07$. The chosen polynomial degree is $(p, p_\gamma) = (4, 0)$; $t_{end} = 10$; $\nu = 3 \cdot 10^{-4}$.

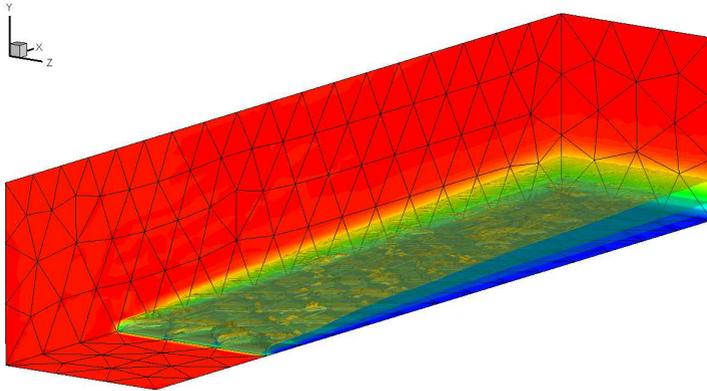


Figure 4.15: 3D plot of the domain Ω and sketch of the mesh on the boundary; The plotted iso-surfaces are corresponding to $u = 0.2, 0.4, 0.8$

The resulting Blasius velocity profile is shown in Figure 4.15 where also a sketch

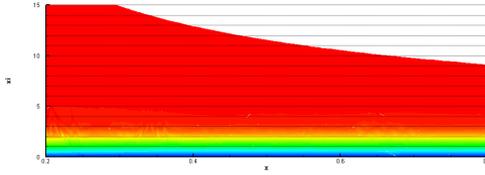


Figure 4.16: 2D plot on the symmetry plane $z = 0$ where the velocity u in the $x - \xi$ plane is plotted.

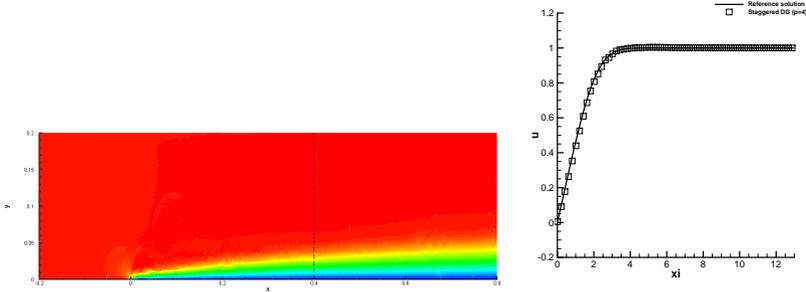


Figure 4.17: Left: 2D plot on the symmetry plane $z = 0$ where the velocity u in the $x - y$ plane is plotted and the extraction line for the comparison. Right: Numerical and reference solution taken on the line $(x, y, z) = (0.4, y, 0)$.

of the grid is reported. The profile with respect to the Blasius coordinate ξ is shown on the left of Figure 4.16 in order to verify whether the obtained solution is self-similar with respect to ξ . A comparison between the numerical results presented here and the reference solution is depicted on the right of Figure 4.16 for $(x, z) = (0.7, 0.0)$. A good agreement between numerical and reference solution can be observed. This is a very good result if we take into account the mesh size and considering that the major part of the boundary layer seems to

be relatively well solved in only one control volume.

4.3.6 Backward-facing step.

In this section, the three-dimensional numerical solution for the fluid flow over a backward-facing step is considered. For this test problem, both experimental and numerical results are available at several Reynolds numbers (see e.g. [3, 79]). In particular, it is known that two dimensional simulations are in good agreement with experimental evidence only up to $Re = 400$. Beyond this critical value, two dimensional simulations present a large secondary recirculation zone that reduces the main recirculation zone, see test in Section 3.6.1.4. On the contrary, experimental results show that this secondary vortex appears only at higher Reynolds number due to three-dimensional effects (see e.g. [3]). The used step size is of $S = 0.49$ and the ratio between the total height H and the inlet height h_{in} is of $H/h_{in} = 1.9423$. We consider here a smaller domain with respect to the experimental setup of Armaly in [3], but sufficient to see the three-dimensional effects. In particular $\frac{x}{S} \in [-10, 20]$, $y \in [-0.49, 0.51]$ and $\frac{z}{S} \in [0, 12]$. The domain is covered using $N_e = 19872$ tetrahedra and we take $(p, p_\gamma) = (4, 0)$ and $Re = 600$. We impose the exact Poiseuille profile in the y -direction at the tube inlet, transmissive boundary conditions at the tube outlet and no-slip boundary conditions otherwise. For the current test dt is taken according to the CFL time restriction and $t_{end} = 80$.

A plot of the velocity profile at several values of x/S is shown in Figure 4.18

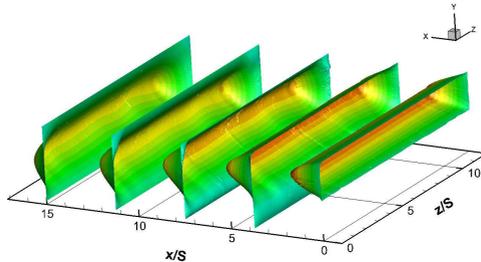


Figure 4.18: Value of u in the (y, z) -plane at $x = [0, 3.75, 7.5, 11.25, 15]$.

The resulting recirculation zones in the symmetry plane and close to the side wall $\frac{z_{max}}{S}$ are shown in Figure 4.19, as well as the equivalent in the plane $(\frac{x}{S}, \frac{z}{S})$ close to the bottom and the top wall in Figure 4.20. As we can see, no important secondary recirculation zones appear in the symmetric plane, while a couple of recirculations appear close to the side walls. The presence of these secondary recirculations seems to reduce the reattachment point for the main recirculation close to the side walls (see Figure 4.20 top). On the contrary, a larger recirculation zone can be seen in the middle of the channel. The resulting reattachment point in the symmetry plane is $\frac{x_1}{S} = 11.2$, that is really close to the one obtained in the experimental case, whose extrapolated value is $\frac{x_1}{S} = 11.24$. Remark how the two dimensional numerical simulation, such as presented in Section 3.6.1.4, leads to a reattachment point of $\frac{x_1}{S} = 9.4$ that completely underestimates the experimental one.

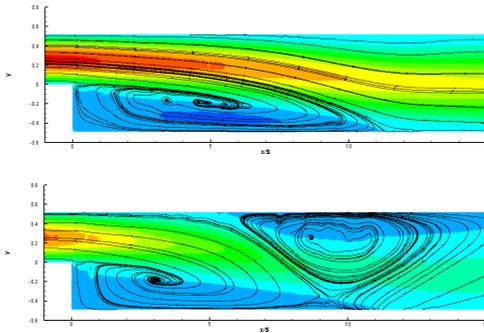


Figure 4.19: Recirculation zones in the plane $(\frac{x}{S}, y)$ in the symmetry plane (top) and close to the side wall at $\frac{z}{S} = 12$ (bottom).

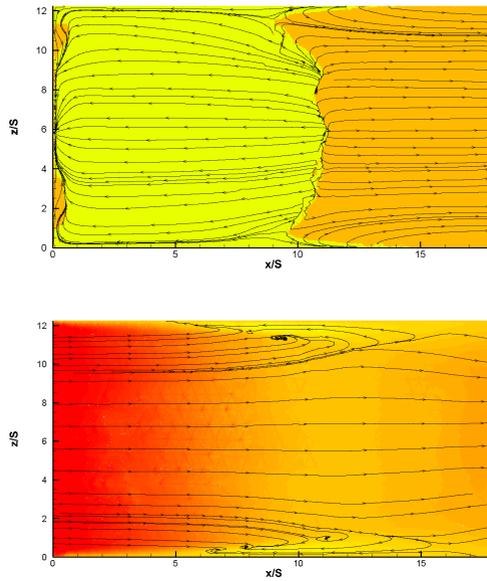


Figure 4.20: Recirculation zones in the plane $(\frac{x}{S}, \frac{z}{S})$ close to the bottom and close to the top wall.

4.3.7 Flow around a sphere

In this section we consider the flow around a sphere. In particular we take as computational domain $\Omega = \mathcal{S}_{10} \cup \mathcal{C}_{10,15} - \mathcal{S}_{0.5}$, where \mathcal{S}_r is a generic sphere with center $\vec{0}$ and radius r ; $\mathcal{C}_{r,H}$ is a cylinder with circular basis on the yz -plane, radius r and height H . We use a very coarse grid that is composed by a total number of $N_e = 14403$ tetrahedra whose characteristic length is $h = 0.2$ close to the sphere, while it is only $h = 0.8$ away from the sphere. A sketch of the grid is shown in Figure 4.21.

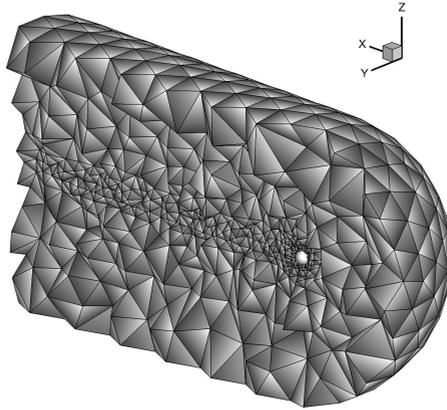


Figure 4.21: Cut view of the computational domain with $N_e = 14403$.

We start from an initial steady flow of magnitude $\vec{v}_0 = (u_\infty, 0, 0)$ with $u_\infty = 0.5$ and we impose u_∞ on $\mathcal{S}_{10} \cap \{x \leq 0\}$ as boundary condition; transmissive boundary condition on $\mathcal{C}_{10,15}$ and no-slip condition on $\mathcal{S}_{0.5}$. We use a polynomial degree $(p, p_\gamma) = (3, 0)$ and $\theta = 0.51$ using the method explained in Section 4.2.6; $Re = 300$; $t_{end} = 300$ and dt is taken according to the CFL time restriction for the convective term.

A plot of spanwise velocity contour for v is reported in Figure 4.22 at $t = 300$ and shows a very complex and three-dimensional behavior of the numerical solution. The mean number of iterations needed to update the pressure p with

a tolerance of $tol = 10^{-8}$ is $I_{mean} = 201.8$. The maximum number of iterations $I_{max} = 2552$ is observed at the beginning of the simulation, when the constant velocity initial condition has to be adjusted. Instead, the minimum $I_{min} = 62$ is observed when the Von Karman street is completely developed.

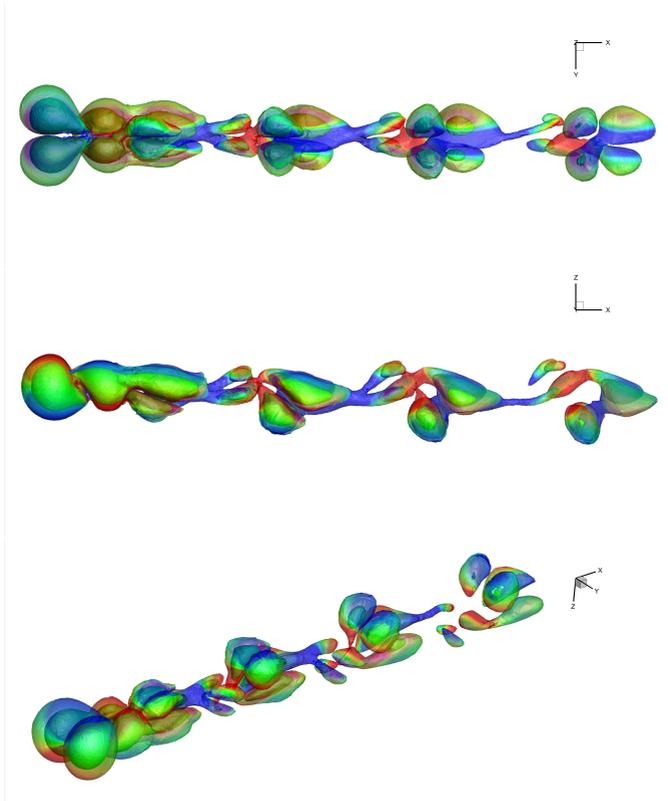


Figure 4.22: Contour isosurfaces for the spanwise velocity v in the (x, y) , (y, z) and $3D$ plot.

A side and upper view of the particle tracker is reported in Figure 4.23 at

$t = t_{end} = 300$ and looks very similar to the experimental one obtained by H.Sakamoto et al in in [139].

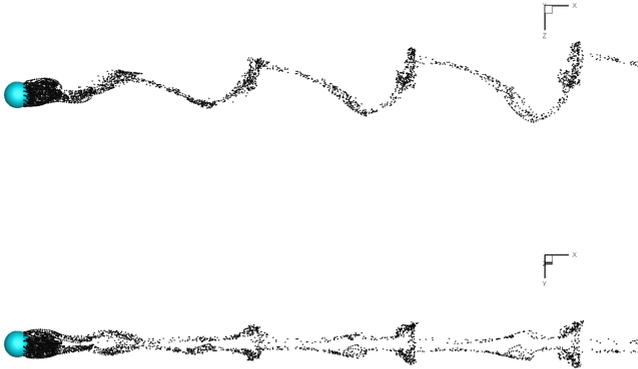


Figure 4.23: Side view and upper view of the particle path at $t = 300$.

The resulting Strouhal number for this simulation is $St = 0.145$ that is close to the experimental range obtained by H. Sakamoto in [139] of $St = 0.15 - 0.18$.

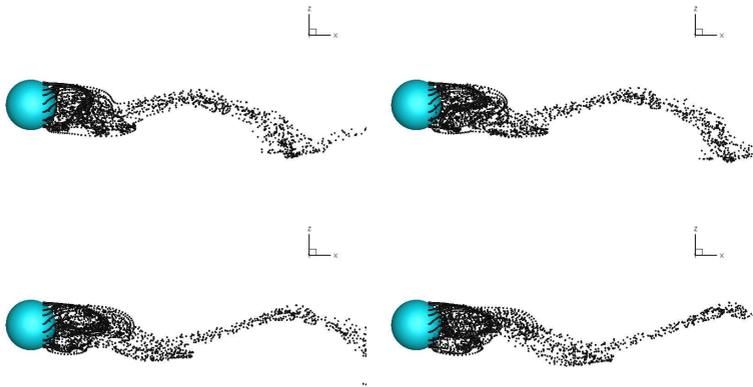


Figure 4.24: Time evolution of the particle path at times, from top left to bottom right, $t = [262.8, 265.5, 267.8, 270.2]$.

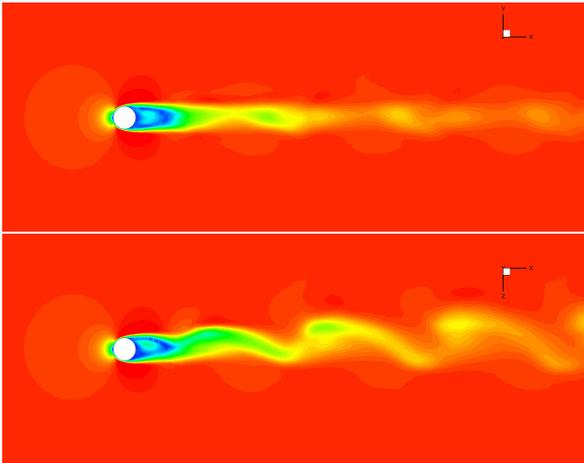


Figure 4.25: Velocity magnitude at $t = t_{end}$ in the (x, y) and (x, z) -plane.

4.3.8 Flow past a circular cylinder

In this last test case we want to treat another classical problem for the incompressible Navier-Stokes equations that is the flow around a circular cylinder. For this test, some numerical and experimental cases are available for a large range of Reynolds numbers. In particular several papers focus the attention on the formation of two instability modes characterized from large and small-scale streamwise vortex structure (see e.g. [156]) and act on the Reynold-Strouhal number relationship. We consider here the problem of the flow past a circular cylinder in a confined channel and for a Reynolds number large enough to have three-dimensional effects and small-scale streamwise vortex structures. We define the blockage ratio $\beta = d/H$ where d indicates the cylinder diameter and H is the distance separating the two walls. Rehimidi did an experimental investigation for a blockage ratio of $\beta = 1/3$ and found the $Re - St \cdot Re$ relation up to $Re = 277$ in [133]. Other numerical studies of Kanaris et al in [109] give us a numerical analysis in the case of lower blockage ratio of $\beta = 1/5$, finding a similar relation with respect to the unconfined experimental case of Williamson in [156]. We consider here two domains that are $\Omega_1 = [-10, 30] \times [-2.5, 2.5] \times [-12, 12] - C_{0.5,24}$ and $\Omega_2 = [-10, 30] \times [-1.5, 1.5] \times [-12, 12] - C_{0.5,24}$ where $C_{r,z}$ represents the cylinder of radius r and height z centered in 0 and corresponding to a blockage ratio of $\beta = 1/5$ and $\beta = 1/3$, respectively. The first domain Ω_1 is covered with a total number of $N_e = 50761$ tetrahedra and Ω_2 is covered with $N_e = 32527$ elements. A sketch of the grid used in both the cases is shown in Figure 4.26. As numerical parameters we use $(p, p_\gamma) = (3, 0)$, $\theta = 0.51$ and $t_{end} = 200$. As initial condition we take a fully developed laminar Poiseuille profile and we impose velocity boundary condition on the inlet, transmissive boundary conditions on the outlet and no slip boundary conditions otherwise. Finally we impose for the two tests $\nu_1 = 1.66667 \cdot 10^{-3}$ and $\nu_2 = 1.80505 \cdot 10^{-3}$ corresponding to $Re_1 = 300$ and $Re_2 = 277$. Furthermore, isoparametric elements are considered for both the cases in order to fit better the curved cylinder. The resulting velocity profile at t_{end} is reported in Figure 4.27, where we can observe the generation of the Von-Karman street past the cylinder as well as the three-dimensional mixing effect given by the spanwise velocity w .

In Figure 4.28 the isosurfaces are reported for of the three-dimensional pattern of the spanwise velocity v and w at $t = t_{end}$. The resulting Strouhal number for the first case is $St = 0.198$ which is in good agreement with the numerical one $St = 0.1989$ of Kanaris in [109] and the experimental one of Williamson in [156]. In the second case the obtained Strouhal number is $St = 0.2414$ which

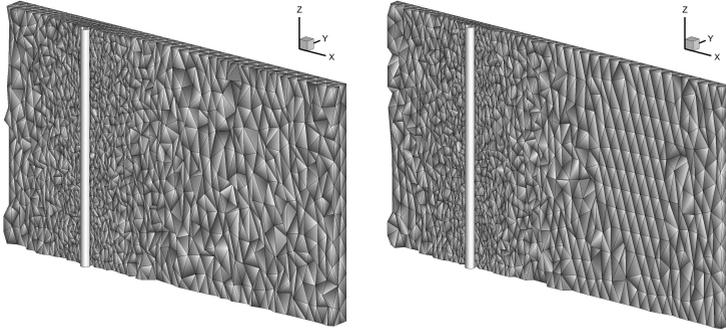


Figure 4.26: Half grid plot of Ω_1 and Ω_2 , respectively from left to right.

corresponds to a value of $St \cdot Re = 66.877$ that is in line with the experimental one of Rehimí et al in [133], whose extrapolated value is $St \cdot Re = 66.929$. This confirms the suggestion given in [133] that the Strouhal number increases with increasing blockage.

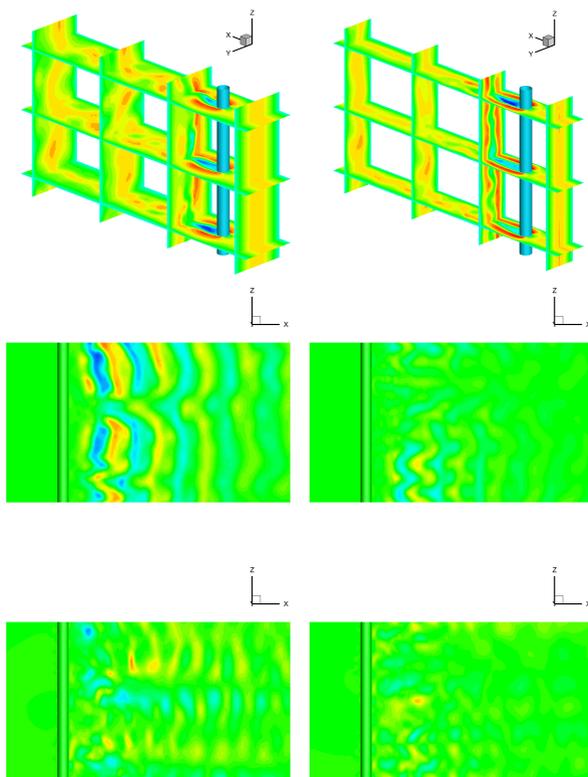


Figure 4.27: Velocity profile u , v and w from top to bottom past the circular cylinder in the sub-domain $\Omega = [-5, 20] \times [-\frac{r}{\beta}, \frac{r}{\beta}] \times [-7, 7]$ for $(Re, \beta) = (300, \frac{1}{5})$ and $(Re, \beta) = (277, \frac{1}{3})$ in the left and right column, respectively.

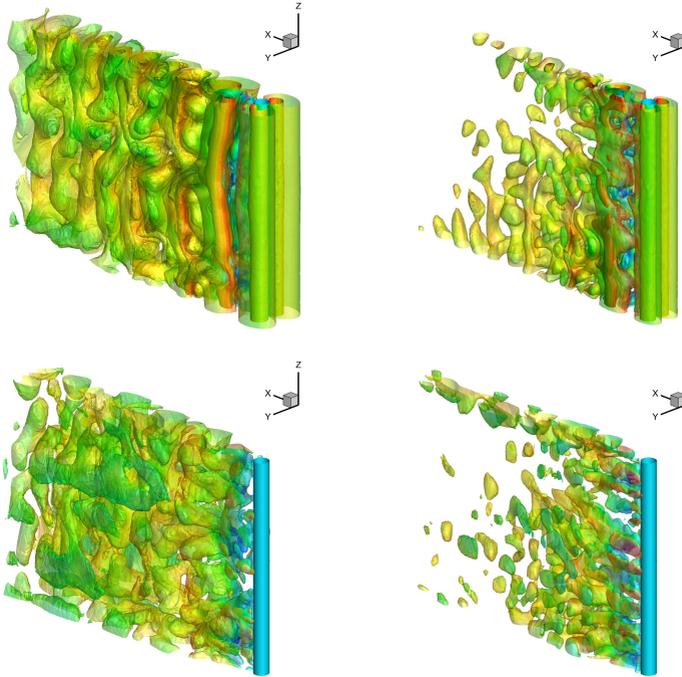


Figure 4.28: Isosurfaces of spanwise velocity $v = [\pm 0.1, \pm 0.03]$ and $w = \pm 0.03$ from top to bottom for the case $(Re, \beta) = (300, \frac{1}{5})$ (left column) and $(Re, \beta) = (277, \frac{1}{3})$ (right column).

4.4 A discussion about the dual convective-viscous computation

In this section we want to investigate the effects of using the natural extension of the convective-viscous term on the dual grid against the procedure on the main grid explained in Section 4.2.4.1. In particular we will see some comparisons regarding the three dimensional lid-driven cavity such as reported in example 4.3.1.

First of all we redo the computation using the natural extension of the nonlinear convective-viscous term on the dual grid. The obtained results are reported in Figures 4.29 for $Re = 400$ and 4.30 for $Re = 1000$; a really good agreement between our numerical solution and the reference one of [2] is achieved also in this case.

We have chosen this test due to the presence, on the upper frame, of a strong discontinuity directly on the boundary conditions, i.e. $\vec{v} = (1, 0, 0)$ on the upper face and $\vec{v} = \vec{0}$ on the other boundaries. For any coverage of Ω , the dual hexahedral mesh contains intrinsically a discontinuity between different faces by construction. This means that if we compute the convective-viscous term on the dual grid, it contains, close to the critical frame, a natural discontinuity also at the discrete level. In particular, for those tetrahedra for which an edge lies on the upper boundary where $\vec{v} = (1, 0, 0)$ and the others lies on the no slip boundary, we can take advantage from the presence of a discontinuity fixed also in the numerical solution. On the contrary, performing the alternative procedure such as described in 4.2.4.1 means that the systems for the convective-viscous contribution are solved at the main grid level. Indeed DoFs are connected in a continuous way inside boundary elements of this kind. As a consequence, if we use a coarse grid such as the considered one as well as a relatively small value of p , then the numerical solution inside these elements tends to generate a small recirculation region due to the Gibbs phenomenon. In order to achieve the natural discontinuity close to the upper frame we force tetrahedra on the line $\{y = 0.5 \text{ and } x = \pm 0.5\}$ to have only faces with one single kind of boundary condition. The grid so constructed will be called *corrected* while the standard one will be called *non corrected*. A detail of the velocity on the symmetry plane $\{z = 0\}$ close to the critical point $(x, y) = (0.5, 0.5)$ is reported in figure 4.31.

We can see how, in the non corrected case, there is an additional recirculation region close to the critical point. This acts as a buffer zone and hence reduces the vertical velocity close to the right face of the cavity (see Fig. 4.31). Con-

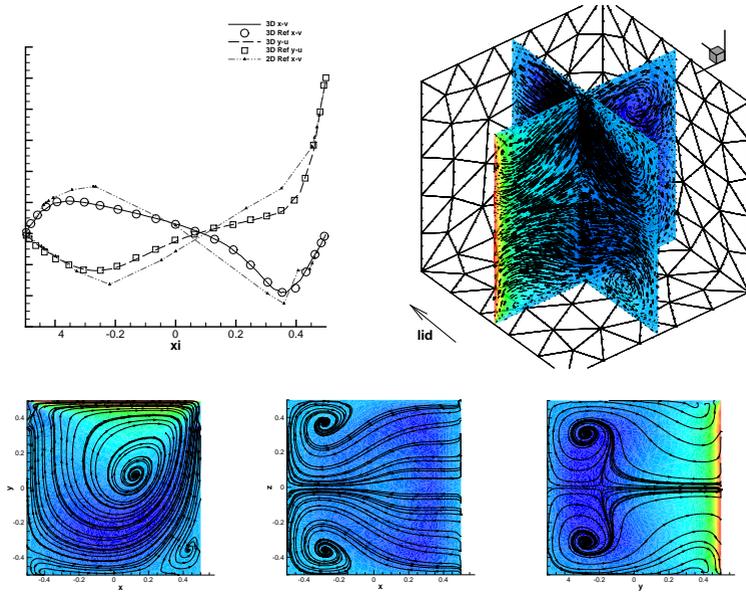


Figure 4.29: Comparison between our numerical results, the one obtained by Albensoeder et al in [2], and the two dimensional data from Ghia et al [89] for $Re = 400$; three-dimensional plot of the two secondary slices and computational grid; streamlines and magnitude of u on slices $x - y$, $x - z$ and $y - z$.

sequently, the resulting numerical solution does not fit the reference close to the right face, see comparison in Figure 4.32. Since the computation of the convective-viscous term on the dual grid by construction does not have this problem, we achieve a good solution also in the case of non corrected grids (Fig. 4.32). This means that once we have a discontinuity at the boundary level we have to ensure it also at the discrete level in order to take advantages of a DG scheme. We do it when we generate the grid and then use this grid without any further modification of the code. In terms of computational effort, the resulting typical CPU time for a single update in time to compute the

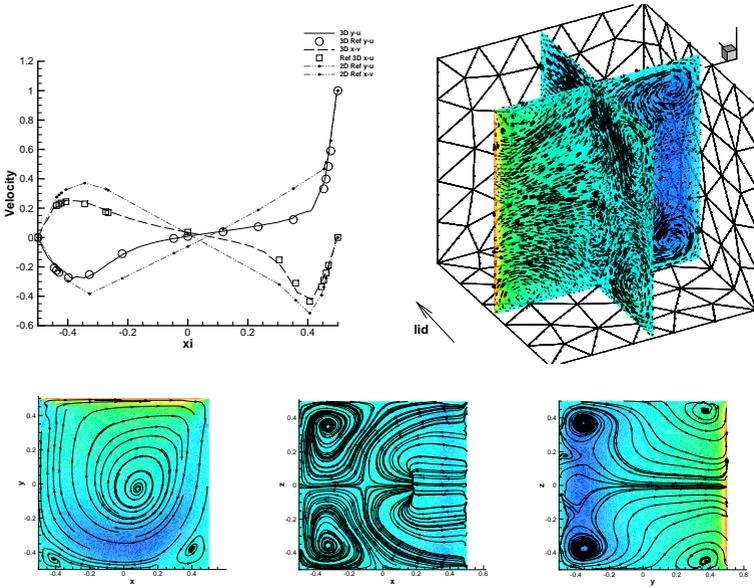


Figure 4.30: Comparison between our numerical results, the one obtained by Albensoeder et al in [2], and the two dimensional data from Ghia et al [89] for $Re = 1000$; three-dimensional plot of the two secondary slices and computational grid; streamlines and magnitude of u on slices $x - y$, $x - z$ and $y - z$.

convective-viscous contribution for the current test on the corrected primary grid is $t_{CPU}^{Main} = 0.65sec$ instead of $t_{CPU}^{Dual} = 3.09sec$ using 4 cores of an $i7 - 3770$ $3.40GHz$ so that the reduction factor for this test case is 4.75. This underlines how the use of this new procedure significantly reduces the computational effort of the nonlinear convective-viscous term.

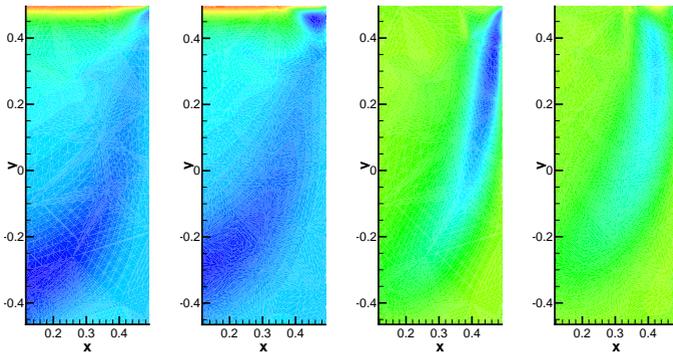


Figure 4.31: Detail of u (first two figures) and v (last two) close to the point $\vec{x} = (0.5, 0.5, 0)$ in the corrected case (left) and non corrected one (right).

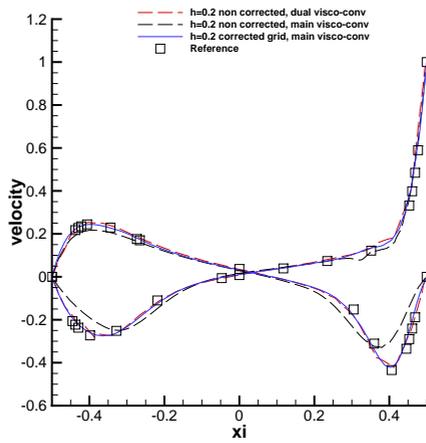


Figure 4.32: Comparison with the reference solution of the three considered cases.

5 Conclusions and Outlook

In this thesis, a new family of arbitrary high order space-time discontinuous Galerkin (DG) schemes on unstructured staggered grids for the numerical solution of the incompressible Navier-Stokes and shallow water equations has been developed. The proposed schemes have been validated against several classical benchmarks, exact solutions and available experimental results. We have first developed the method for the simple two-dimensional shallow water equations and then extended it for the numerical solution of two and three-dimensional incompressible Navier-Stokes equations. In the last two cases the method is derived in such a fashion that we can always recover the low order in time method as a particular case for which we recover the symmetry and in general the semi-positive definiteness properties of the main matrix for the pressure. Furthermore, we have implemented a classical semi-implicit representation that maintains the symmetry of the matrix and involves also the Crank-Nicolson procedure in order to recover some extra precision in time compared to the simple backward Euler scheme. The possible use of general unstructured meshes and the ability to curve the elements allow to adapt the mesh easily to the real geometry of the problem. Thanks to the use of staggered grids, each main element involves only its direct neighbors and consequently the resulting main matrices for the pressure (respectively free surface) become a four-point block system for the two dimensional cases and a five-point block system in three space dimensions. This allows to solve the system very efficiently even if no preconditioning has been used, which is a very interesting feature of our staggered space-time DG scheme compared to other implicit and semi-implicit DG schemes that exist in the literature. In addition, all the matrices and tensors used in the algorithm can be precomputed and stored in a preprocessing step. In this way also the extension to high order isoparametric geometry does not affect the run time part since they are used to compute only the matrices and tensors.

In the incompressible Navier-Stokes case it was shown that the stability of the scheme is given by the stability of the nonlinear convective contribution. For the convective contribution we have used a classical explicit upwind discretization for DG schemes that is affected by a *CFL* stability restriction based on the flow

velocity. We have also discussed about a different procedure to compute the nonlinear convective-viscous contribution for three dimensional flows. In this case a better conditioned matrix that results symmetric and positive definite in the case $p_\gamma = 0$ is obtained for the computation of the viscous contribution. This leads to a significant saving in terms of computational effort.

We have finally discussed about some technical issues concerning isoparametric and parallel MPI implementation.

Future developments will concern the extension of the presented methods to the compressible Navier-Stokes equations as well as to the ideal and viscous and resistive MHD equations, where the divergence free condition of the magnetic field can be naturally incorporated at the staggered level. Furthermore, since the *CFL* condition affecting the nonlinear convective terms is based on the local convective speed rather than wave velocity speed, it allowed us to use large time steps for the low Froude number, shallow water case. Similarly, a semi-implicit discretization should provide significant advantages for low Mach number compressible flows, such as they appear in computational meteorology. In this framework, the method developed in this thesis is a good candidate for the high order extension of the semi-implicit finite volume method presented very recently by Dumbser and Casulli in [71] for the simulation of all Mach number compressible flows. Further work will consider the application of the present method to large scale simulations of realistic geophysical flows.

A Appendices

A.1 Further notes about 2D shallow water

A.1.1 Compute the nonlinear convection term

In this section we give some details about the computation of the nonlinear convective contribution. A weak formulation of Eq. (2.35) reads, for each component of the velocity, as

$$\begin{aligned} \left(\int_{\mathbf{R}_j} \psi_k \psi_l d\mathbf{x} \right) \frac{\hat{\mathbf{F}}\mathbf{U}_{l,j}^n - \hat{\mathbf{U}}_{l,j}^n}{\Delta t} &= \left(\int_{\mathbf{R}_j} \nabla \psi_k \psi_l d\mathbf{x} \right) \hat{\mathbf{F}}_{l,j}^x - \oint_{\partial \mathbf{R}_j} \psi_k \left(\vec{F}^x \cdot \vec{n}_{e,j} \right) ds, \\ \left(\int_{\mathbf{R}_j} \psi_k \psi_l d\mathbf{x} \right) \frac{\hat{\mathbf{F}}\mathbf{V}_{l,j}^n - \hat{\mathbf{V}}_{l,j}^n}{\Delta t} &= \left(\int_{\mathbf{R}_j} \nabla \psi_k \psi_l d\mathbf{x} \right) \hat{\mathbf{F}}_{l,j}^y - \oint_{\partial \mathbf{R}_j} \psi_k \left(\vec{F}^y \cdot \vec{n}_{e,j} \right) ds, \end{aligned} \quad (\text{A.1})$$

where $\vec{F}^x = \frac{1}{H}(U^2, UV)$; $\vec{F}^y = \frac{1}{H}(UV, U^2)$; and $\widehat{\mathbf{F}}\vec{v} = (\hat{\mathbf{F}}\mathbf{U}, \hat{\mathbf{F}}\mathbf{V})$. Defining

$$\mathbf{V}_{\mathcal{CF}_j}(k, l) = \int_{\mathbf{R}_j} \nabla \psi_k \psi_l d\mathbf{x} = \begin{pmatrix} \mathbf{V}_{\mathcal{CF}_j^1} \\ \mathbf{V}_{\mathcal{CF}_j^2} \end{pmatrix}, \quad (\text{A.2})$$

we obtain directly the volume contribution to the nonlinear part, explicitly given by $\mathbf{V}_{\mathcal{CF}_j} \cdot \hat{\mathbf{F}}_j^n$. For the edge contribution in Eqs. (A.1) we use a Rusanov numerical flux defined such as

$$\begin{aligned} \vec{F}^x \cdot \vec{n}_{e,j} &= \frac{1}{2} \left[\left(\vec{F}^x \right)^+ + \left(\vec{F}^x \right)^- \right] \cdot \vec{n}_{e,j} - \frac{1}{2} s_{max} [(U)^+ - (U)^-], \\ \vec{F}^y \cdot \vec{n}_{e,j} &= \frac{1}{2} \left[\left(\vec{F}^y \right)^+ + \left(\vec{F}^y \right)^- \right] \cdot \vec{n}_{e,j} - \frac{1}{2} s_{max} [(V)^+ - (V)^-], \end{aligned} \quad (\text{A.3})$$

and then, the edge contribution is given, for the first equation as

$$\begin{aligned}
 \oint_{\partial \mathbf{R}_j} \psi_k \vec{F}^x \cdot \vec{n}_{e,j,l} ds &= \sum_{l \in S_j} \oint_{\Gamma_l} \psi_k \left(\vec{F}^x \cdot \vec{n}_{e,j} ds \right) \\
 &= \sum_{l \in S_j} \oint_{\Gamma_l} \psi_k^{(j)} \frac{1}{2} \left[\left(\vec{F}^x \right)^+ + \left(\vec{F}^x \right)^- \right] \cdot \vec{n}_{e,j,l} ds \\
 &\quad - \sum_{l \in S_j} \oint_{\Gamma_l} \psi_k^{(j)} \frac{1}{2} s_{max} [(U)^+ - (U)^-] ds \\
 &= \frac{1}{2} \sum_{l \in S_j} \left\{ \oint_{\Gamma_l} \psi_k^{(j)} \psi_r^{(\varphi(j,l))} ds \vec{n}_{e,j,l} \hat{\mathbf{F}}_{\varphi(j,l)}^x \right. \\
 &\quad \left. + \oint_{\Gamma_l} \psi_k^{(j)} \psi_r^{(j)} ds \vec{n}_{e,j,l} \hat{\mathbf{F}}_j^x \right\} \\
 &\quad - \frac{1}{2} s_{max} \sum_{l \in S_j} \left\{ \oint_{\Gamma_l} \psi_k^{(j)} \psi_r^{(\varphi(j,l))} ds \hat{\mathbf{U}}_{\varphi(j,l)} \right. \\
 &\quad \left. - \oint_{\Gamma_l} \psi_k^{(j)} \psi_r^{(j)} ds \hat{\mathbf{U}}_j \right\}. \tag{A.4}
 \end{aligned}$$

Also in this case it is convenient to denote the matrices of the contribution for a quadrilateral element j and edge $l \in S_j$ such as

$$\begin{aligned}
 \mathcal{E}_{\mathcal{F}_{j,l}^c}(k, r) &= \oint_{\Gamma_l} \psi_k^{(j)} \psi_r^{(j)} ds, \\
 \mathcal{E}_{\mathcal{F}_{j,l}^o}(k, r) &= \oint_{\Gamma_l} \psi_k^{(j)} \psi_r^{(\varphi(j,l))} ds, \tag{A.5}
 \end{aligned}$$

where the indexes c and o are used to distinguish the contribution from the current quadrilateral element and from the other quadrilateral element with

respect to the edge l . Eq. (A.4) is then rewritten in a compact form as

$$\oint_{\partial \mathbf{R}_j} \psi_k \left(\vec{F}^x \cdot \vec{n}_{e,j,l} \right) ds = \frac{1}{2} \sum_{l \in S_j} \left\{ \mathcal{E} \mathcal{F}_{j,l}^o \vec{n}_{e,j,l} \hat{\mathbf{F}}_{\varphi(j,l)}^x + \mathcal{E} \mathcal{F}_{j,l}^c \vec{n}_{e,j,l} \hat{\mathbf{F}}_j^x - s_{max} \left[\mathcal{E} \mathcal{F}_{j,l}^o \hat{\mathbf{U}}_{\varphi(j,l)} - \mathcal{E} \mathcal{F}_{j,l}^c \hat{\mathbf{U}}_j \right] \right\} \quad (\text{A.6})$$

and in the same way for the other component \vec{F}^y . The maximum convection speed s_{max} is given by the maximum local normal velocity, computed on the edge l . Substituting of Eq. (A.1) into (A.6) one obtains an explicit formulation for the convective terms in the \vec{x} and \vec{y} direction:

$$\begin{aligned} \frac{1}{\Delta t} \left(\hat{\mathbf{F}} \mathbf{U}_j^n - \hat{\mathbf{U}}_j^n \right) &= \mathcal{V} \mathcal{C} \mathcal{F}_j \hat{\mathbf{F}}_{l,j}^x \\ &\quad - \frac{1}{2} \sum_{l \in S_j} \left\{ \mathcal{E} \mathcal{F}_{j,l}^o \vec{n}_{e,j,l} \hat{\mathbf{F}}_{\varphi(j,l)}^x + \mathcal{E} \mathcal{F}_{j,l}^c \vec{n}_{e,j,l} \hat{\mathbf{F}}_j^x \right. \\ &\quad \left. - s_{max} \left[\mathcal{E} \mathcal{F}_{j,l}^o \hat{\mathbf{U}}_{\varphi(j,l)} - \mathcal{E} \mathcal{F}_{j,l}^c \hat{\mathbf{U}}_j \right] \right\} \\ &= \mathcal{S}^x(\hat{\mathbf{V}}), \end{aligned} \quad (\text{A.7})$$

and

$$\begin{aligned} \frac{1}{\Delta t} \left(\hat{\mathbf{F}} \mathbf{V}_j^n - \hat{\mathbf{V}}_j^n \right) &= \mathcal{V} \mathcal{C} \mathcal{F}_j \hat{\mathbf{F}}_{l,j}^y \\ &\quad - \frac{1}{2} \sum_{l \in S_j} \left\{ \mathcal{E} \mathcal{F}_{j,l}^o \vec{n}_{e,j,l} \hat{\mathbf{F}}_{\varphi(j,l)}^y + \mathcal{E} \mathcal{F}_{j,l}^c \vec{n}_{e,j,l} \hat{\mathbf{F}}_j^y \right. \\ &\quad \left. - s_{max} \left[\mathcal{E} \mathcal{F}_{j,l}^o \hat{\mathbf{V}}_{\varphi(j,l)} - \mathcal{E} \mathcal{F}_{j,l}^c \hat{\mathbf{V}}_j \right] \right\} \\ &= \mathcal{S}^y(\hat{\mathbf{V}}). \end{aligned} \quad (\text{A.8})$$

In order to get stability, a TVD Runge Kutta of the third order is used for the operators \mathcal{S} . The convective nonlinear term is then computed first by setting:

$$\begin{aligned} k_1 &= \mathcal{S} \left(\hat{\mathbf{v}}^{\cdot,n} \right), \\ k_2 &= \mathcal{S} \left(\hat{\mathbf{v}}^{\cdot,n} + \Delta t k_1 \right), \\ k_3 &= \mathcal{S} \left(\hat{\mathbf{v}}^{\cdot,n} + \frac{1}{4} \Delta t (k_1 + k_2) \right), \end{aligned} \quad (\text{A.9})$$

and finally

$$\begin{aligned}\hat{\mathbf{F}}\mathbf{U}_j^n &= \hat{\mathbf{U}}_j^n + \Delta t \frac{1}{6} (k_1^x + k_2^x + 4k_3^x), \\ \hat{\mathbf{F}}\mathbf{V}_j^n &= \hat{\mathbf{V}}_j^n + \Delta t \frac{1}{6} (k_1^y + k_2^y + 4k_3^y).\end{aligned}\tag{A.10}$$

A.2 High Performance Computing

In this part of the thesis we want to introduce some technical issues about high performance computing (HPC) using a message passing interface (MPI). Contrary to what happens using OpenMP, where we have a unique memory and several threads, in MPI programming we have several processors with their own memory. The passage of information about computational domains handled by different CPUs has to pass through direct communications or messages between processors. This philosophy allows to distribute the running on a huge amount of processors that can be also physically separated, as it is typical in the major supercomputer structures. In particular, the fact that each processor has its own memory means that each processor sees only a part of the computational domain and so we have all the quantities defined in Section 2.2.1 that have to be computed locally. Furthermore, the communications have to be handled in a proper way. In this section we will focus our attention on some particular aspects regarding MPI implementation using staggered unstructured meshes. Furthermore we will discuss these issues in a two dimensional framework since it is easier to visualize, but the same reasoning applies readily to the complete three-dimensional case.

A.2.1 Some details about MPI parallel implementation

We report here some technical details regarding the high performance computing performed using MPI. In particular, we will see some details about the treatment of the staggered grid on the MPI boundary (MPIb). In this direction, the simplest solution is to duplicate the dual elements on the MPI boundary. This solution leads to an overhead of information since the boundary dual elements will be computed twice; on the other hand, we would guarantee a direct and natural assignment between edges and the dual elements in the local connectivity system. This means that in problems where the MPI boundary is an important part (i.e. three dimensional problems), this overhead reduces the maximum efficiency of the parallelized code. Taking into account this, we will assign the MPI boundary dual element to one of the two neighbor processors,

for instance the left processor. In this way we have solved the overhead problem, but we have introduced an additional issue: let us consider in figure A.1

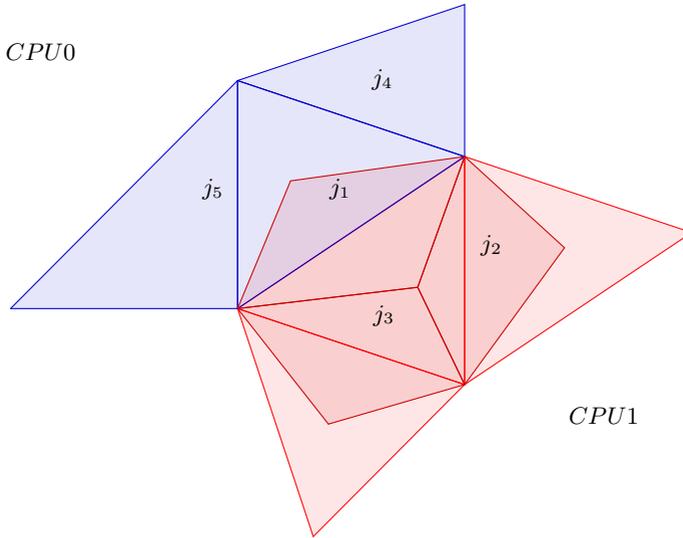


Figure A.1: MPI boundary dual mesh issue.

the boundary element j_1 assigned to $CPU1$. In a local connectivity system processor 1 will assign:

$$CPU1 : \quad j_1 \rightarrow j_1^1 \quad , \quad j_2 \rightarrow j_2^1 \quad , \quad j_3 \rightarrow j_3^1$$

while processor 0 will assign

$$CPU0 : \quad j_1 \rightarrow j_1^0 \quad , \quad j_4 \rightarrow j_4^0 \quad , \quad j_5 \rightarrow j_5^0$$

The difference in this case is that processor 1 can associate each edge in the local connectivity system to a dual element, while processor 0 has the information about the edge, since it composes a triangle in the region of CPU 0, but it

cannot associate a corresponding dual element. In addition if we take a look on the halo zone, we get that for the main grid the halo zone is composed by the direct neighbor of each main element on the MPIb, while for the dual grid it is necessary, in general, to consider the neighbor of the neighbor (see e.g. Figure A.2). Of course this information is needed only to reconstruct the main

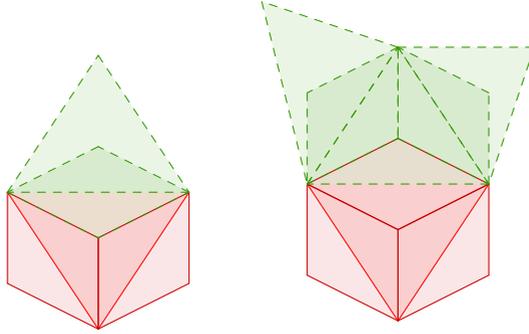


Figure A.2: Information needed to reconstruct the communicated informations.

matrices. Once we have computed all the contributions we can readily use the halo information sent from the neighbor CPUs. We will call this extended stencil for the MPI boundary shadow zone since, in general it is different from the halo zone. Through this work we will use the convention of negative indexes on the shadow and halo elements. This means that shadow or halo triangles, edges and dual elements will be negative while internal elements for the main and dual grid will have a positive numbering. As explained above, we have a potential problem in the local connectivity system for the dual grid. We cannot guarantee that the numbering used for the edges is the same used for the dual grid, but we can construct the shadow zone in such a fashion that there exists a bijection between the edges and the dual grid. In this way we can renumber the edges according to the dual grid. This is simply done by considering only the direct neighbor if the corresponding dual element on the MPI boundary is associated to the neighbor CPU and considering the extended shadow (involving also the neighbor of the neighbor) if the dual element is an element of the current CPU. In addition we will store only the edges that has a

left and a right element as well as the real boundary elements (i.e. $j \in \mathcal{B}(\Omega)$). Let take for example the figure A.1. If the original numbering is taken to be $\{j_i\}_{i=1\dots 5} = 1\dots 5$ then the local connectivity for the edges ($j_{glob} \rightarrow j_{edges}$) becomes

$$\begin{aligned} CPU0 : & \quad 1 \rightarrow 1 \quad , \quad 4 \rightarrow 2 \quad , \quad 5 \rightarrow 3 \\ CPU1 : & \quad 1 \rightarrow 1 \quad , \quad 2 \rightarrow 2 \quad , \quad 3 \rightarrow 3 \quad , \quad 4 \rightarrow -1 \quad , \quad 5 \rightarrow -2 \end{aligned}$$

while the assignation between the edges and the dual elements becomes

$$\begin{aligned} CPU0 : & \quad 1 \rightarrow 1 \quad , \quad 2 \rightarrow 2 \quad , \quad 3 \rightarrow -1 \\ CPU1 : & \quad 1 \rightarrow 1 \quad , \quad 2 \rightarrow 2 \quad , \quad 3 \rightarrow 3 \quad , \quad -1 \rightarrow -1 \quad , \quad -2 \rightarrow -2 \end{aligned}$$

We can remark for example how, for CPU 0, the physical edge 3 is associated to a shadow dual element -1 , while for CPU 1 all the physical edges are associated to physical dual elements as well as shadow edges are associated to shadow dual elements.

In this way we can renumber the edges and so obtain again a bijective map between the edges and the dual elements in this local index system.

A.2.2 Periodic boundary conditions using MPI

In this part of the section about HPC we want to introduce a way to implement the periodic boundary conditions with MPI. In a classical framework of DG schemes on non-staggered grids this represents a really trivial issue since it is sufficient to geometrically connect the edges where there is a periodic boundary (PB). Similarly, the use of a discretization of the nonlinear convective-viscous contribution on the main grid such as described in Section 4.2.4.1 moves the MPI communications from the dual to the main grid and so a classical implementation of MPI code can be done.

On the contrary, as soon as we use staggered grids we have to consider also the staggered elements that lie on the periodic boundary. In order to compute the main matrices, these elements can be taken into account by duplicating one single node (i.e. the external vertex) for each element. This node can be reconstructed using a rigid body translation of the physical node placed in the opposite PB. In addition, we have to choose an element that will be the final physical element. We have chosen to keep the elements that lead on the left boundary in the direction where the PB is applied (with respect to the element couple). For each $j_p \in \mathcal{B}(\Omega)$ that is a periodic edge, we have a couple of associated elements $PBE_{lr}(j_p) = \{j_l, j_r\} \in \mathcal{B}(\Omega)^2$ so that $j_l = j_r$

once the geometry will be connected. On the staggered level we have the two associated dual elements $\Omega_{j_l} = \{\ell(j_l), \Gamma_{j_l}\}$ and $\Omega_{j_r} = \{\ell(j_r), \Gamma_{j_r}\}$ that are some triangle elements because of $\{j_l, j_r\} \in \mathcal{B}(\Omega)^2$. We can easily develop the final elements that will be $\Omega_{j_l} = \{\ell(j_l), p_l, \Gamma_{j_l}\}$ and $\Omega_{j_r} = \{\ell(j_r), p_r, \Gamma_{j_r}\}$, where $\vec{x}_{p_l} = U(\vec{x}_{\ell(j_l)})$ is a new virtual point that is given by a proper rigid translation of the point $\ell(j_l)$ and same reasons for j_r . At the end of the connection only j_l will be a physical element and in this context we can call j_r a virtual element. Furthermore, since we are considering also high order, we have to compute in a pre-processing step some flux matrices for each $j \in S_i$, $l \in S_j$ and j a PBE. In order to do this it is convenient to keep also the virtual elements in order to compute the fluxes (two fluxes from one side and two from the other) on each $l \in S_j$. After having done this we can connect the correct fluxes and discard the virtual elements. This procedure results convenient if we simply consider a serial code. In the case of a parallel MPI program, we have also a halo zone as well as a shadow zone and, in general, the virtual one may or may not be contained in the shadow one. In addition we have to be careful about the communication order.

For example let us take a periodic square where each processor has a corner, so that CPU0 will have the top left corner, CPU1 will have the top right and so on (see fig. A.3). About halo communication on the main grid we have no special issues since, if we take in the worst case a corner element of CPU1, then it communicates with CPU0 through the right edge and CPU3 through the top edge. From a staggered point of view, the same element communicates also with CPU2. Using the same example but with only two processors where CPU0 takes the left area and CPU1 takes the right one, if we consider the main grid and the same corner element we do not need apparently any communication since the neighbor element lies on the same CPU, but for the dual staggered equivalent we need a communication between the two processors. This means that in this case, for this element, we do not have a main halo zone but we need to take into account a corresponding shadow element.

A.2.3 Some 2D examples

In this section we want to show some examples of applicability of the MPI implementation in two space dimensions. As a first example we take the double shear layer test such as in section 3.6.2.4 but we use 40 CPUs and a finest grid whose characteristic size is $h = 0.05$. A sketch of the grid and local indexes for each CPU is shown in figure A.4.

We perform the computation such as in 3.6.2.4 but with a $\nu = 5 \cdot 10^{-5}$ corre-

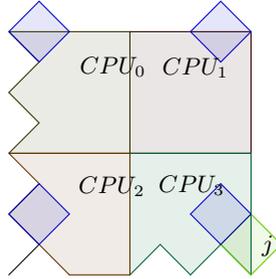


Figure A.3: Periodic Boundary conditions and MPI shadow issue.

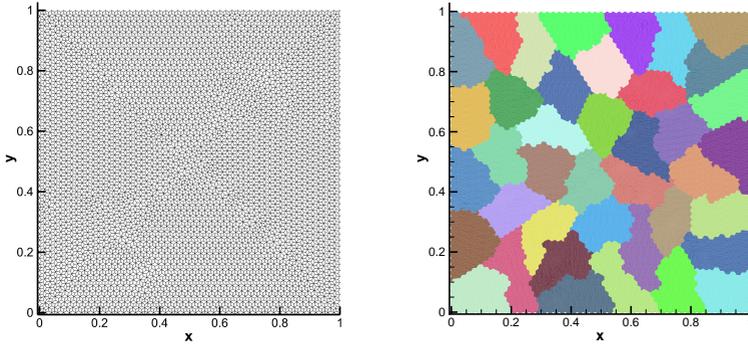


Figure A.4: Dual grid plot and MPI decomposition of the computational domain.

sponding to $Re = 20.000$, $(p, p_\gamma) = (4, 0)$ and $\theta = 0.51$. The resulting vorticity profile is reported in figure A.5 for several times and showing the formation of several thin vorticity layers that are well solved also at time $t = 1.8s$.

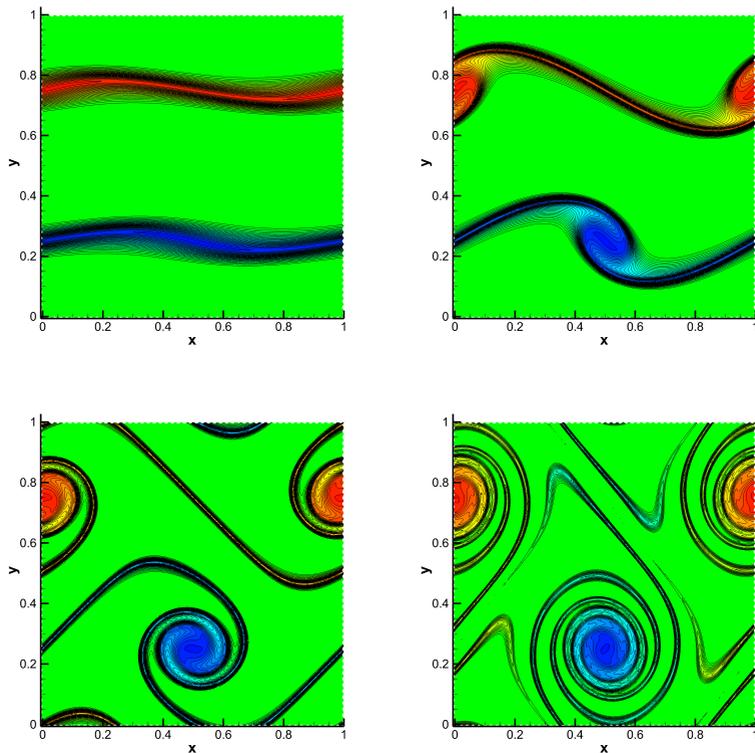


Figure A.5: Vorticity pattern for the double shear layer test at times, from top left to bottom right, $t = 0.4$; $t = 0.8$; $t = 1.2$; $t = 1.8$.

A second $2D$ application of the MPI algorithm is the cavity flow presented in Section 3.6.1.3 but at $Re = 10.000$. We use the same grid such as shown in Figure A.4 but without periodic boundary. In Figure A.6 is reported the time evolution of the vorticity. Remark how small structures can be observed as well as Kelvin Helmholtz instability appearing close to the internal corners.

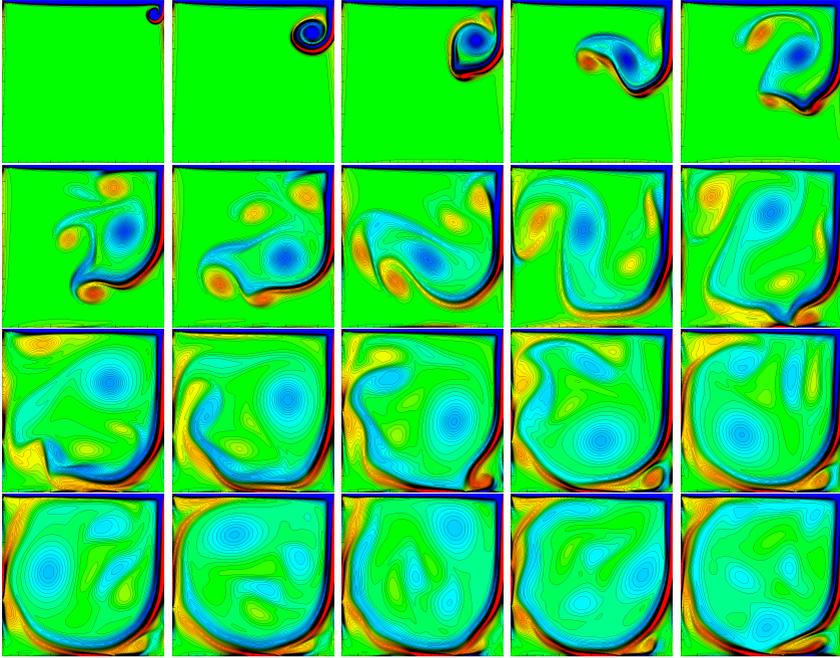


Figure A.6: Vorticity pattern for the cavity at times from $t = 1$ and taken each $2sec$ from top left to bottom right.

A.2.4 A three dimensional test

It remains to check what is the efficiency of the parallel implementation. In general, we have a code that runs in parallel and a certain time that is given by some communication and synchronization procedures. In our simulations we can always measure the total CPU time, which in general can be written as

$$T = T^{par} + T^{sync}$$

where T^{sync} is the total synchronization time computed as the maximum waiting time over all the processors; T^{par} is the time to compute the parallel code

with no communications. Observe how T^{sync} contains the effective time of communications as well as bad balancing problems that intrinsically increase the synchronization time. The efficiency is in general defined as

$$s = \frac{T_1}{nT_n}.$$

We can assume that, for an optimal parallel code where no communications appear (or the communication time is zero), the efficiency is of 100%, so that

$$\alpha = \frac{T_1}{nT_n^{par}}$$

with $\alpha = 1$. Remark also that $T_1^{sync} = 0$ since there is only one processor. This means that $T_1 = T_1^{par}$. This is a really strong assumption since, in applications, we have several memory issues that tend to change the value of α . As example, a memory buffer saturation would reduce the value of α while a cache phenomenon would increase this value. We can in any case manipulate the previous formula and so obtain:

$$\alpha = \frac{T_1}{nT_n^{par}} = \frac{T_1}{nT_n - nT^{sync}} = \frac{T_1}{nT_n} \frac{1}{1 - \frac{T^{sync}}{T_n}} = s \frac{1}{1 - \frac{T^{sync}}{T_n}}, \quad (\text{A.11})$$

and finally

$$s = \alpha \left(1 - \frac{T^{sync}}{T_n} \right), \quad (\text{A.12})$$

so the efficiency is directly a function of the percentage of synchronization time with respect to the total time. This means that we can estimate, neglecting all the memory issues, the efficiency simply measuring the total time needed for the communications/synchronization procedures. A more practical test that, on the contrary, requires more effort to be done, is a classical estimation of the efficiency, using different number of CPUs and measuring the entire CPU time. For the strong scaling test we fix the total number of tetrahedra and we increase the number of CPUs. In this sense we have done a simple three-dimensional Taylor-Green vortex at $Re = 400$ such as described in Section 4.3.3 using $N_e = 331338$ tetrahedra up to $t = 1s$. The resulting computational time and efficiency is reported in Table A.1 where the local efficiency is reported and compared between two successively simulations. For this test we have used the fat nodes of SuperMUC system of LRZ. In particular, each node consists in

Table A.1: Strong scaling efficiency test for the Taylor-Green vortex on SuperMUC.

N_{Nodes}	Submit Class	N_{CPU}	Comp. Time	Loc. Eff
40	General	1120	2750.39	-
80	General	2240	1386.42	99.2%
100	General	2800	1178.88	94.1%
120	General	3360	1058.52	92.8%

four Intel Xeon E7-4870, 2.40Ghz with 40 cores per node and 6.4GB of RAM memory per core.

Figure A.7 shows the details about computational time of different part of the program. In particular are reported the CPU time for the nonlinear convective-viscous part, the solution of the main system and the total time for each time update.

We perform now a weak scaling test. In this case we maintain constant the load for CPU to $N_e = 138$ tetrahedra with $p = 4$. Furthermore we use a fixed $dt = 10^{-3}$ for all the simulations. For this test we use the new Phase 2 Haswell nodes on SuperMUC. Each node consists in two Intel Xeon E5-2697 v3 with 2.60Ghz, 28 cores per node and 2.3GB of RAM memory per core.

Table A.2: Weak scaling efficiency test for the Taylor-Green vortex on SuperMUC.

N_{Nodes}	Submit Class	N_{CPU}	N_e	Comp. Time	Loc. Eff
2	Micro	56	7728	709.70	-
16	Micro	448	61824	1007.74	70.4%
54	General	1512	208656	1260.88	79.9%
128	General	3584	494592	1649.90	76.4%
250	General	7000	966000	2509.50	65.7%

The resulting efficiency is reported in Table A.2 where a worse efficiency can

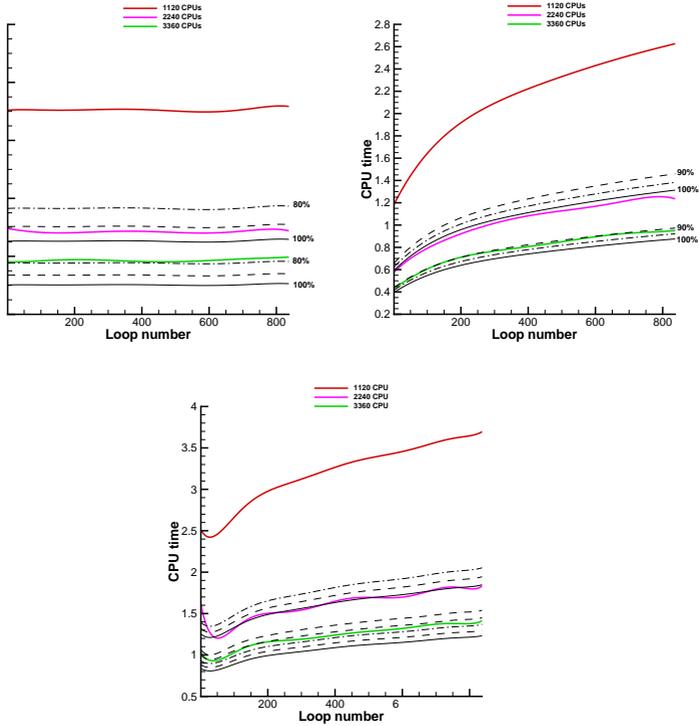


Figure A.7: Computational time plot, from left to right, for nonlinear convective-viscous term, main linear system and total time.

be observed with respect to the strong scaling. However, since we are using iterative linear solvers, the total number of iterations usually increases with the system dimension. Consequently, the computational effort of each processor cannot be considered constant in this case.

A.3 Solving the Laplace equation for 3D isoparametric elements

When we use isoparametric elements we have to be careful since it is extremely easy to generate nonconforming meshes when relatively coarse grids are used to represent complex domains. Let us consider first a simple convex domain. In this case it would be sufficient to project the elements that share at least a one-dimensional boundary element on the physical boundary. In order to avoid confusion we introduce some definitions regarding boundary and near-boundary elements:

Definition 1 *Let d the dimension of the problem. We say that an element T is a boundary element if it contains an element of dimension $(d - 1)$ that is on the boundary. It is said to be a near-boundary element if it is not a boundary element and contains an element of dimension $(d - 2)$ that is on the computational boundary.*

For the two dimensional case and convex domains we have eventually to modify only the boundary elements since the near-boundary triangles have only points that lie on the boundary and hence do not need to be modified. For the three dimensional case if the domain is convex we can modify only boundary and near-boundary elements. Remark how since the NB elements have lines that lie on the boundary, they have to be modified according to the physical domain. In the general case of non-convex domains it is not possible to project locally since it is very easy to generate intersections or leave some DoF out of the physical domain. In addition it is still not sufficient, in general, to modify in an elastic way the internal DoF of boundary and near-boundary elements since the curvature can completely invalidate some boundary elements. So we have to move the entire grid. In order to do this we mount a tetrahelization that involves all the DoFs and we solve an elastic Laplace equation for the displacement with a classical continuous finite element methods using P1 elements and imposing the projection on the physical domain as boundary conditions. A sketch of the displacement is shown in figure A.8 for the flow past sphere test case where, on the left, is reported the grid before the elastic transformation as well as the displacement of all DoFs in several planes. On the right is plotted the same grid after the elastic transformation. Remark how the original mesh is too coarse to represent the sphere using subparametric elements and to move only boundary and near-boundary elements. On the contrary we have now that the entire grid is curved also for internal elements. Fortunately since

the isoparametric approach involves only the geometry, it does not affect the computational effort during the runtime but only the preprocessing step.

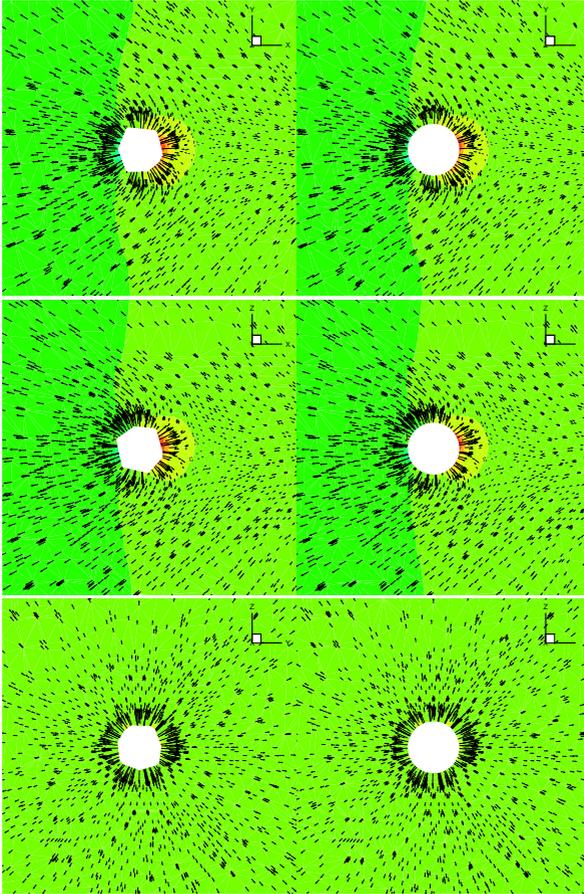


Figure A.8: Really coarse grid with $p = 4$, (Left) before elastic transformation, (Right) after elastic transformation. From top to bottom: xy , xz , yz plane.

A.4 Isoparametric Hexahedral characterization

From the previous section we have now our generalized isoparametric elements $\{\Omega_i^{iso}\}_i$.

It is convenient, in order to compute the integrals, to represent each hexaedral such as the union of two tetrahedra, namely $\Omega_j = \Omega_{\ell(j),j} \cup \Omega_{r(j),j}$. The purpose of this section is to generate, starting from a given main mesh $\{\Omega_i^{iso}\}_i$ and $j \in [1, N_d]$ a consistent dual element Ω_j^{iso} . This reduces to found an element $\Omega_{i,j}^{iso}$ for each $i, j \in S_i$. The trick here is to use the same isoparametric mapping used on the main grid. We will first recognize the correct sub-tetrahedron in a first step reference space and then we will reconstruct the final nodes from a second step reference space. Let $\{p_1, p_2, p_3\}$ be the three nodal point of a standard triangle (i.e. $p_1 = (0, 0), p_2 = (0, 1), p_3 = (1, 0)$). The three points that characterize in a subparametric mean the edge j are given by

$$p^{ph} = T_{tri}^{-1,3D} \left(p \cdot |\Gamma_j^{iso} \right)$$

and the corresponding characteristic points in the first step reference space are

$$p^r = T^{iso} \left(p^{ph} | \Omega \right) = T^{iso} \left(T_{tri}^{-1,3D} \left(p \cdot |\Gamma_j^{iso} \right) | \Omega \right)$$

Now we can add the internal mean point in the reference space that is given by $m^r = (0.25, 0.25, 0.25)$ and use the resulting four points $T_{ref}^{dual} = \{m^r, p_1^r, p_2^r, p_3^r\}$ to develop the final second step reference space. In this way we can use a set of characterization points $\{\vec{x}_k = (x, y, z)_k\}_{k=1 \dots N_\phi}$ on the standard tetrahedron. The resulting characterization points in the physical space will be given by the composition of the first and second step reference maps such as:

$$\vec{x}_k^{ph} = T^{-1,iso} \left(T^{-1} \left(\vec{x}_k | T_{ref}^{dual} \right) | \Omega \right)$$

A diagram of the used maps is given in figure A.9.

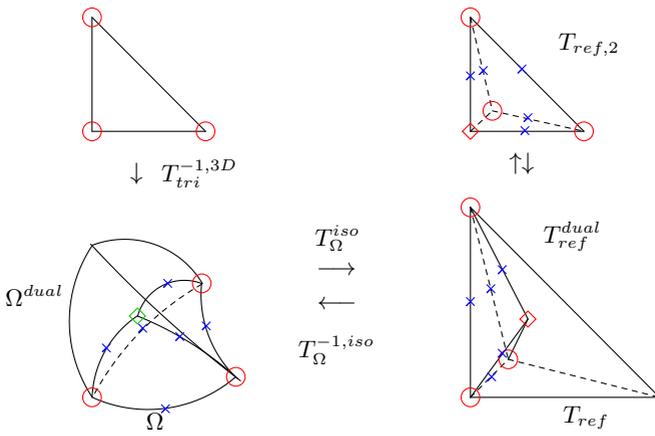


Figure A.9: Construction of a general dual isoparametric element.

A.5 A brief discussion about optimal order estimation

We want to give here a really brief discussion about the advantages and disadvantages of using an high order method instead of a low order one. In order to do this we will consider a very simplified framework and we perform several analysis in order to answer the question: when is it convenient to use high order methods or not? At the base of this question there is the fact that, from one side, high order methods produce a very fast reduction of the error but, from the other side, they have a cost greater than a low order method. Let take a look at the Table 2.1 in Chapter 2. We see how, taking the same number of elements, the high order method costs much more than the low order one (i.e. $p = 0$). On the contrary, since we need few elements to get a good solution, it seems that fixing an error, it is convenient to use high order methods. Another question is if there exists an optimal polynomial degree for a given error and complexity of the algorithm.

For high order methods we usually estimate the error norm E as a function of the number of elements n and polynomial degree p as follows:

$$E(p, n) = C(p)H(n)^{p+\alpha},$$

where $C(p)$ is a constant that does not depend on n ; α is an optimality parameter; $H(n)$ is a characteristic size, which for square domains can be written such as

$$H \approx \frac{\sqrt{d}|\Omega|}{n^{\frac{1}{d}}},$$

here d represents the dimension of the problem. Formally proofs applied to central DG schemes lead to an optimality parameter of $\alpha = 0$, while Shu showed that it is at least $\alpha = 0.5$ for the two dimensional unstructured case. In practice the observed value of α is 1. We consider now $n = 1$, the resulting error $E(p, 1)$ represents the L_2 -error on the subspace

$$V_{\Omega}^p = \{P(x) \mid P \text{ is a polynomial of degree } p \text{ on } \Omega\}.$$

Since $V_{\Omega}^p \subset V_{\Omega}^{\bar{p}} \forall \bar{p} > p$ we have that $E(\bar{p}, 1) \leq E(p, 1)$. We consider now the normalized error with respect to this projection error $E(p, 1)$:

$$e(p, n) = \frac{E(p, n)}{E(p, 1)} = h(n)^{p+\alpha} \tag{A.13}$$

where $h(n) = n^{-\frac{1}{d}}$. In the rest of this section we will hence suppose $n > 1$ and $e \in [0, 1)$. We need now to write an equation for the computational effort.

The particular structure of the proposed method allows us to write everything in terms of *atomic operations* and the corresponding *weight* of each atomic operation. Let indicate with $w(p)$ this weight, the computational effort for the main operations can be schematized in:

- Element Element multiplication (EE) operation $\rightarrow w(p)$ operations,
- Matrix Element multiplication (ME) operation $\rightarrow w(p)^2$ operations,
- Matrix Matrix Element multiplication (MME) operation $\rightarrow 2w(p)^2$ operations.

Usually the biggest part of the algorithm consists in (ME) and (MME) operations and hence we will neglect (EE) and initialization cost in this analysis. For an iterative linear solver we get the following computational cost in terms of atomic operations:

$$I = (n \cdot \text{Stencil} \cdot ME) \cdot \text{Loops}(n, p, \text{tol}). \quad (\text{A.14})$$

We further suppose that the number of loops for the linear solver depends on the number of elements but not on the size of the polynomial space, i.e. $\text{Loops}(N(n), p, \text{tol}) = \text{Loops}(N(n), \text{tol})$. In this way we are able to write a general expression for the computational cost such as

$$\begin{aligned} O(p, n) &= O^m(p, n) + O^d(p, n) \\ &= f(n) (w^m(p))^2 + g(n) (w^d(p))^2, \end{aligned} \quad (\text{A.15})$$

where f indicates the number of only atomic operations on the main grid, while g is the equivalent for the dual staggered one. It is natural to suppose that f and g are monotone increasing functions. A particular way to represent f and g is to use polynomials for which we have the following result:

Lemma 4 *Let $f(x) = \sum_{k=0}^N \alpha_k x^k$ with $\alpha_k \geq 0 \forall k \in [0, N]$, $N > 0$ a monotonically increasing function of $x > 1$. Then:*

1. $f(x) > 0$ and $f'(x) > 0 \forall x > 1$,
2. $\frac{f(x)}{x f'(x)} > 0 \forall x > 1$,
3. $\left(\frac{f(x)}{x f'(x)} \right)' < 0 \forall x > 1$.

A.5.1 Minimize the computational cost

We consider now a fixed error $\epsilon \in (0, 1)$. From Eq. (A.13) there exists a series of combinations $(n, p)(\epsilon)$ so that $e(n(\epsilon), p(\epsilon)) = \epsilon$ and once we fix n or p then the other is fixed. We will hence write $\mathbf{p} = p(n)$ as function of n so that $e(n(\epsilon), \mathbf{p}(n(\epsilon))) = \epsilon$. We easily derive from Eq. (A.13) that

$$\mathbf{p}(n) = \frac{d|\log \epsilon| - \alpha \log n}{\log n}. \quad (\text{A.16})$$

The previous equation is valid for $n \in (1, n_{lim}]$ where $n_{lim} = \epsilon^{-\frac{d}{\alpha}}$ is the solution of $\mathbf{p}(n_{lim}) = 0$. Furthermore we observe that

$$\mathbf{p}(n) \geq 0 \quad \forall n \in (1, n_{lim}], \quad (\text{A.17})$$

and

$$\mathbf{p}(n)' = \left(-\frac{1}{n \log n} \right) (\mathbf{p}(n) + \alpha) < 0 \quad \forall n \in (1, n_{lim}]. \quad (\text{A.18})$$

We consider now the case of one, two and three dimensional problems separately, since the weight function $w(p)$ has different shape depending on the problem dimension.

A.5.1.1 One dimensional case

In the one dimensional framework $w = w^m(p) = w^d(p) = p + 1$. In this way Eq. (A.15) is rewritten such as

$$O(p, n) = f(n)w(p)^2, \quad (\text{A.19})$$

where f groups both the main and the dual number of atomic operations. Let $o(n) = O(\mathbf{p}(n), n)$ the computational cost associated to $(\epsilon, n(\epsilon))$. We want to minimize this quantity with respect to n , so we consider the quantity $o'(n)$ that reads:

$$\begin{aligned} o'(n) &= f'(n) (\mathbf{p}(n) + 1)^2 + 2f(n) (\mathbf{p}(n) + 1) \mathbf{p}'(n) \\ &= f'(n) (\mathbf{p}(n) + 1)^2 + 2f(n) (\mathbf{p}(n) + 1) \left(-\frac{1}{n \log n} \right) (\mathbf{p}(n) + \alpha) \\ &= f'(n) (\mathbf{p}(n) + 1)^2 \left[1 - 2 \frac{f(n)}{n f'(n)} \frac{1}{\log n} \frac{\mathbf{p}(n) + \alpha}{\mathbf{p}(n) + 1} \right] \\ &= f'(n) (\mathbf{p}(n) + 1)^2 S(n). \end{aligned} \quad (\text{A.20})$$

Since $f'(n) > 0$ and $\mathbf{p}(n) + 1 > 0$ we have characterized the minimum such as the zeros of S . Since the range is limited by n_{lim} and $\lim_{n \rightarrow 1} o(n)' = -\infty$ there exists at least a minimum, which in the worst case is achieved by choosing $n = n_{lim}$. Observe how this last case would mean that it is convenient to take $p = 0$ and hence a low order method. The general form of f does not allow us to characterize better the minimum but we can still proof that, for polynomial shape of f , the minimum is unique. Let take first $\alpha = 1$ so that

$$S(n) = 1 + 2 \frac{f(n)}{nf'(n)} \left(-\frac{1}{\log n} \right),$$

and hence, using Lemma 4,

$$\begin{aligned} S'(n) &= 2 \left(\frac{f(n)}{nf'(n)} \right)' \left(-\frac{1}{\log n} \right) + 2 \frac{f(n)}{nf'(n)} \left(-\frac{1}{\log n} \right)' \\ &= 2 \underbrace{\left(\frac{f(n)}{nf'(n)} \right)'}_{<0} \underbrace{\left(-\frac{1}{\log n} \right)}_{<0} + 2 \underbrace{\frac{f(n)}{nf'(n)}}_{>0} \underbrace{\left(-\frac{1}{n \log^2 n} \right)}_{>0}, \end{aligned} \quad (\text{A.21})$$

and so $S'(n) > 0$ for $n \in (1, n_{lim}]$. Since the minimum are characterized only by S , it means that the minimum is unique. In the general case of $\alpha \in [0, 1]$ we have:

$$\begin{aligned} S'(n) &= 2 \left(\frac{f(n)}{nf'(n)} \right)' \left(-\frac{1}{\log n} \right) \frac{\mathbf{p}(n) + \alpha}{\mathbf{p}(n) + 1} \\ &\quad + 2 \frac{f(n)}{nf'(n)} \left(-\frac{1}{\log n} \right)' \frac{\mathbf{p}(n) + \alpha}{\mathbf{p}(n) + 1} \\ &\quad + 2 \frac{f(n)}{nf'(n)} \left(-\frac{1}{\log n} \right) \left(\frac{\mathbf{p}(n) + \alpha}{\mathbf{p}(n) + 1} \right)', \end{aligned} \quad (\text{A.22})$$

and thanks to

$$\left(\frac{\mathbf{p}(n) + \alpha}{\mathbf{p}(n) + 1} \right)' = \left(1 - \frac{1 - \alpha}{\mathbf{p}(n) + 1} \right)' = \frac{(1 - \alpha)\mathbf{p}'(n)}{(\mathbf{p}(n) + 1)^2} < 0, \quad (\text{A.23})$$

we still conclude that the minimum is unique. Furthermore, if \bar{n} is the minimum (i.e. $S(\bar{n}) = 0$), then $o'(n) < 0 \forall n < \bar{n}$ and $o'(n) > 0 \forall n > \bar{n}$, because $f'(n) > 0$, $\mathbf{p}(n) + 1 > 0$ and $\lim_{n \rightarrow 1} S(n) = -\infty$. This allows us to investigate

in which cases it is convenient to use $p = 0$ just checking the sign of $o'(n_{lim})$. After some computations we obtain that it is convenient to use $p > 0$ if

$$o'(n_{lim}) > 0 \Leftrightarrow |\log \epsilon| > 2 \frac{\alpha}{d} \epsilon^{\frac{d}{\alpha}} \frac{f\left(\epsilon^{-\frac{d}{\alpha}}\right)}{f'\left(\epsilon^{-\frac{d}{\alpha}}\right)}. \quad (\text{A.24})$$

From previous Eq. (A.24) and assuming f a polynomial function such as in Lemma 4, we readily obtain that passing through the limit of $\epsilon \rightarrow 0$ then $\log \epsilon \rightarrow \infty$ and

$$\epsilon^{\frac{d}{\alpha}} \frac{f\left(\epsilon^{-\frac{d}{\alpha}}\right)}{f'\left(\epsilon^{-\frac{d}{\alpha}}\right)} \rightarrow M < \infty$$

thanks to Lemma 4, so Eq. (A.24) is always satisfied. On the contrary, if we pass through the limit $\epsilon \rightarrow 1$ we get

$$0 > 2 \frac{\alpha}{d} \frac{f(1)}{f'(1)} > 0$$

which is a contradiction. In addition, the function on the left is a monotonic decreasing function of ϵ and the right one is a monotonic increasing function of ϵ and hence we recover the following results:

Lemma 5 *Let f a polynomial function such as in Lemma 4, then*

- $\forall \epsilon \in (0, 1) \exists !(\bar{n}, \mathbf{p}(\bar{n}))$ that minimizes the computational cost O ;
- $\exists \epsilon_{lim} \in (0, 1)$ such that $\forall \epsilon > \epsilon_{lim}$ it is convenient to take $p = 0$, otherwise it is convenient to take $p > 0$;
- For $\epsilon \rightarrow 0$ it is always convenient to use $p > 0$, while for $\epsilon \rightarrow 1$ it is always convenient to use $p = 0$.

A.5.1.2 Two dimensional case

In the two dimensional case the weights for the main and staggered grid are:

$$\begin{aligned} w^m(p) &= \frac{(p+1)(p+2)}{2}, \\ w^d(p) &= (p+1)^2. \end{aligned} \quad (\text{A.25})$$

Since in general the computational cost on the main grid is higher with respect to the one at the staggered level, we look at the minimum of the two contributions separately. Indeed, using the same reasoning of previous the section we obtain:

$$\begin{aligned} o^m(n)' &= \frac{1}{4} f'(n) (\mathbf{p}(n) + 1)^2 (\mathbf{p}(n) + 2)^2 S^m(n), \\ o^d(n)' &= g'(n) (\mathbf{p}(n) + 1)^4 S^d(n), \end{aligned} \tag{A.26}$$

where

$$\begin{aligned} S^m(n) &= 1 + 2 \frac{f(n)}{n f'(n)} \left(-\frac{1}{\log n} \right) \frac{\mathbf{p}(n) + \alpha}{\mathbf{p}(\alpha) + 1} + 2 \frac{f(n)}{n f'(n)} \left(-\frac{1}{\log n} \right) \frac{\mathbf{p}(n) + \alpha}{\mathbf{p}(\alpha) + 2}, \\ S^d(n) &= 1 + 4 \frac{g(n)}{n g'(n)} \left(-\frac{1}{\log n} \right) \frac{\mathbf{p}(n) + \alpha}{\mathbf{p}(\alpha) + 1}. \end{aligned} \tag{A.27}$$

About S^m and S^d we reach the same conclusions such as in the previous section. In this case we potentially have two minimum values \bar{n} and \hat{n} , one for the main and another one for the dual grid. In any case $o'(n) < 0 \forall n < \min(\bar{n}, \hat{n})$ and $o'(n) > 0 \forall n > \max(\bar{n}, \hat{n})$ and in the range $\min(\bar{n}, \hat{n}) \leq n \leq \max(\bar{n}, \hat{n})$ we have at least a minimum. A typical shape of the function $o(n)$ can be observed in Figure A.10 left for f polynomial of third order and g polynomial of second order. Figure A.10 right shows the behavior of $\mathbf{p}(n)$ for $\epsilon = 10^{-6}$, while Figure A.11 shows the value of $o'(n_{lim})$ for several kind of f (in particular, from right to left, for f polynomial of degree 3,2,1 and 0).

Regarding the three-dimensional case, using the L_2 -mean procedure such as described in Section 4.2.4.1, then $w(p) = w^d(p) = w^m(p) = \frac{(p+1)(p+2)(p+3)}{6}$ and, after some computations, we achieve the same conclusions of the Lemma 4.

This simple analysis in a very simplified framework will be now compared with the practical results achieved for the shallow water equations.

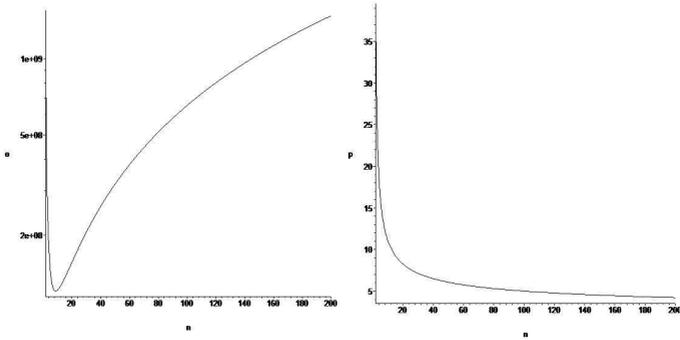


Figure A.10: Typical shape for $o(n)$ and $p(n)$.

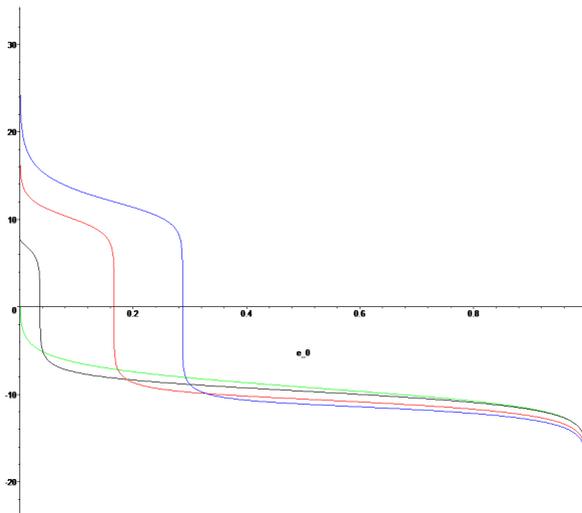


Figure A.11: O' as function of n for several kind of f , $g = 0$.

A.6 Experimental estimation of the optimal polynomial degree

We want here to estimate which is the optimal polynomial degree that minimizes the computational cost. In order to do this we take the convergence results given in Section 2.3.1.

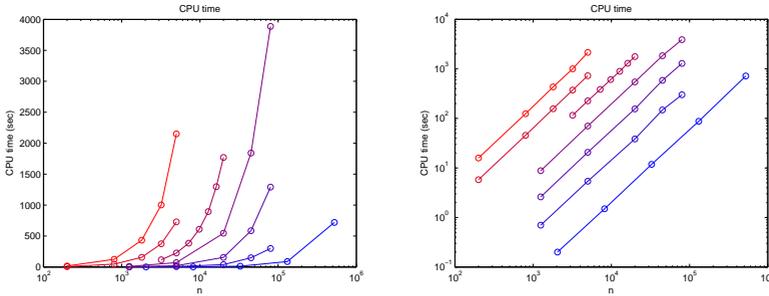


Figure A.12: Experimental values of CPU time for several n and polynomial degrees.

Figure A.12 shows the resulting CPU time and number of elements chart in semi and total logarithmic scale, colored using polynomial degree p . We observe how, in the fully logarithmic scale (log-log), the CPU time behaves linearly with respect to n . In addition, the gradient in the log-log space seems to be constant. This means that the computational effort $O(n, p)$ can be estimated as the multiplication of two parts: one dependent only on p and the second one dependent only on n :

$$O(n, p) \approx n^a e^b, \tag{A.28}$$

where $a = 1.4884$ and $b = b(p)$ is an increasing function of p . Figure A.13 shows the behavior of $e^{b(p)}$ and suggests to take a polynomial type extrapolation curve. From Figure A.13, a $p4$ extrapolation curve seems to be a good approximation, but practice shows that we would take also $p5$ in order to approximate better $b(p)$ for low values of p .

In the same way we can estimate the error norm as function of the number of elements and the polynomial degree, using $p1$ for the gradient and $p2$ extrapolation polynomials for the constant. The resulting approximations compared

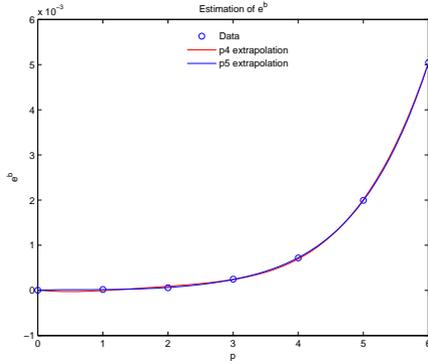


Figure A.13: Behavior of $e^{b(p)}$.

to the data are reported in Fig. A.14. In order to finish our analysis we need

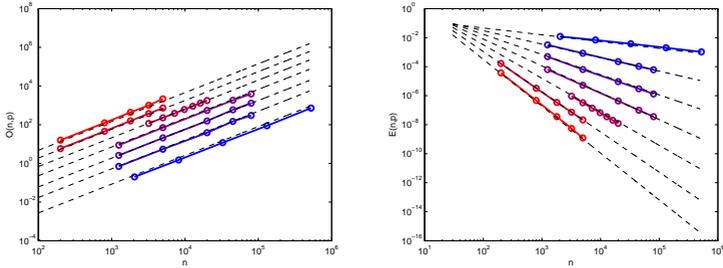


Figure A.14: Data and estimated value for the CPU time O and the error E .

to know, for a fixed ϵ and \bar{n} , what is the combination so that $E(\bar{n}, \bar{p}) = \epsilon$ and so what is $\bar{p} = p(\bar{n}, \epsilon)$. In order to do this we have first written the nonlinear functional $f(n, p) = E(n, p) - \epsilon$ and then solved it using the secant method. Once we have $p(\bar{n}, \epsilon)$ we further compute $O(n, \epsilon) = O(n(\epsilon), p(n(\epsilon)))$. Figure

A.15 shows the resulting $O(n, \epsilon)$ using $\epsilon = 10^{-5}$ as well as the corresponding value of $p(n, \epsilon)$.

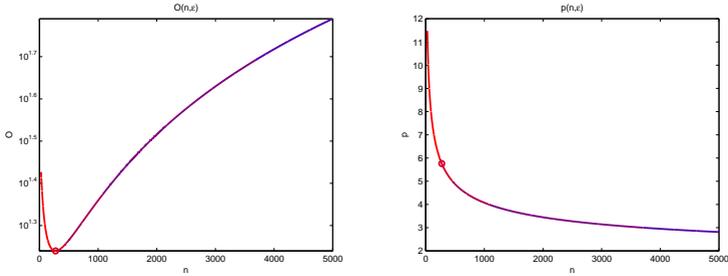


Figure A.15: CPU time and polynomial degree as a function of n for $\epsilon = 10^{-5}$.

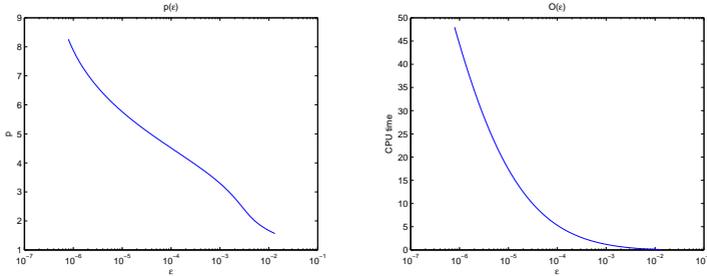


Figure A.16: $p(\epsilon)$ and $O(\epsilon)$ for several values of ϵ .

We finally track the position of the minimum for several values of ϵ . In this way $O(\epsilon) = O(\bar{n}(\epsilon), p(\bar{n}(\epsilon)))$. We report in Fig. A.16 the graph for $n(\epsilon)$, $p(\epsilon)$, and $O(\epsilon)$. As expected, the optimal polynomial order tends to increase for $\epsilon \rightarrow 0$, confirming the theoretical analysis given in the previous section. Also the typical behavior of the computational cost acts similar to the one observed in the previous section, where a very simplified framework was considered.

Bibliography

- [1] C. K. Aidun, N.G. Triantafillopoulos, and J.D. Benson. Global stability of a lid-driven cavity with throughflow: Flow visualization studies. *Physics of Fluids*, 3:2081–2091, 1991.
- [2] S. Albensoeder and H.C. Kuhlmann. Accurate three-dimensional lid-driven cavity flow. *Journal of Computational Physics*, 206:536–558, 2005.
- [3] B.F. Armaly, F. Durst, J.C.F. Pereira, and B. Schonung. Experimental and theoretical investigation on backward-facing step flow. *Journal of Fluid Mechanics*, 127:473–496, 1983.
- [4] V.I. Arnold. Sur la topologie des écoulements stationnaires des fluides parfaits. *Comptes Rendus Hebdomadaires des Séances de l'Académie des Sciences*, 261:17–20, 1965.
- [5] E. Audusse, F. Bouchut, M.O. Bristeau, R. Klein, and B. Perthame. A fast and stable well-balanced scheme with hydrostatic reconstruction for shallow water flows. *SIAM Journal on Scientific Computing*, 25:2050–2065, 2004.
- [6] T. Barth and Pierre Charrier. Energy stable flux formulas for the discontinuous galerkin discretization of first-order nonlinear conservation laws. Technical Report NAS-01-001, NASA, 2001.
- [7] F. Bassi, A. Crivellini, D.A. Di Pietro, and S. Rebay. On a robust discontinuous galerkin technique for the solution of compressible flow. *Journal of Computational Physics*, 218:208–221, 2006.
- [8] F. Bassi, A. Crivellini, D.A. Di Pietro, and S. Rebay. An implicit high-order discontinuous Galerkin method for steady and unsteady incompressible flows. *Computers and Fluids*, 36:1529–1546, 2007.
- [9] F. Bassi, A. Crivellini, D.A. Di Pietro, and S. Rebay. An implicit high-order discontinuous Galerkin method for steady and unsteady incompressible flows. *Computers and Fluids*, 36:1529–1546, 2007.

- [10] F. Bassi, N. Franchina, A. Ghidoni, and S. Rebay. Spectral p-multigrid discontinuous Galerkin solution of the Navier-Stokes equations. *International Journal for Numerical Methods in Fluids*, 67:1540–1558, 2011.
- [11] F. Bassi, A. Ghidoni, S. Rebay, and P. Tesini. High-order accurate p-multigrid discontinuous Galerkin solution of the Euler equations. *International Journal for Numerical Methods in Fluids*, 60:847–865, 2009.
- [12] F. Bassi and S. Rebay. A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations. *Journal of Computational Physics*, 131:267–279, 1997.
- [13] F. Bassi and S. Rebay. High-order accurate discontinuous finite element solution of the 2D Euler equations. *Journal of Computational Physics*, 138:251–285, 1997.
- [14] J. B. Bell, P. Coletta, and H. M. Glaz. A second-order projection method for the incompressible navier-stokes equations. *Journal of Computational Physics*, 85:257–283, 1989.
- [15] A. Bermudez, A. Dervieux, J.A. Desideri, and M.E. Vazquez. Upwind schemes for the two-dimensional shallow water equations with variable depth using unstructured meshes. *Computer Methods in Applied Mechanics and Engineering*, 155:49–72, 1998.
- [16] A. Bermúdez and M. E. Vázquez. Upwind methods for hyperbolic conservation laws with source terms. *Computers and Fluids*, 23:1049–1071, 1994.
- [17] F. Biagio. Calculation of laminar flows with second-order schemes and collocated variable arrangement. *International Journal for Numerical Methods in Fluids*, 26:887–905, 1998.
- [18] P. Birken, G. Gassner, M. Haas, and C.D. Munz. Preconditioning for modal discontinuous Galerkin methods for unsteady 3D Navier-Stokes equations. *Journal of Computational Physics*, 240:20–35, 2013.
- [19] H. Blasius. Grenzschichten in Flüssigkeiten mit kleiner Reibung. *Z. Math. Physik*, 56:1–37, 1908.

- [20] W. Boscheri, M. Dumbser, and M. Righetti. A semi-implicit scheme for 3d free surface flows with high order velocity reconstruction on unstructured voronoi meshes. *International Journal for Numerical Methods in Fluids*, 72:607–631, 2013.
- [21] M.E. Brachet, D.I. Meiron, and S.A. Orszag. Small-scale structure of the taylor-green vortex. *Journal of Fluid Mechanics*, 130:411–452, 1983.
- [22] A.N. Brooks and T.J.R. Hughes. Stream-line upwind/Petrov Galerkin formulation for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equation. *Computer Methods in Applied Mechanics and Engineering*, 32:199–259, 1982.
- [23] P. Brufau, M.E. Vazquez-Cendon, and P. Garcia-Navarro. A numerical model for the flooding and drying of irregular domains. *International Journal for Numerical Methods in Fluids*, 39:247–275, 2002.
- [24] L. Brugnano and V. Casulli. Iterative solution of piecewise linear systems. *SIAM Journal on Scientific Computing*, 30:463–472, 2007.
- [25] L. Brugnano and A. Sestini. Iterative solution of piecewise linear systems for the numerical solution of obstacle problems. *Journal of Numerical Analysis, Industrial and Applied Mathematics*, 6:67–82, 2012.
- [26] A. Canestrelli, M. Dumbser, A. Siviglia, and E. F. Toro. Well-balanced high-order centered schemes on unstructured meshes for shallow water equations with fixed and mobile bed. *Advances in Water Resources*, 33:291–303, 2010.
- [27] M. J. Castro, J. M. Gallardo, and C. Parés. High-order finite volume schemes based on reconstruction of states for solving hyperbolic systems with nonconservative products. applications to shallow-water systems. *Mathematics of Computations*, 75:1103–1134, 2006.
- [28] M. J. Castro, P. G. LeFloch, M. L. Muñoz-Ruiz, and C. Parés. Why many theories of shock waves are necessary: Convergence error in formally path-consistent schemes. *Journal of Computational Physics*, 227:8107–8129, 2008.
- [29] M.J. Castro, J.A. López-García, and C. Parés. High order exactly well-balanced numerical methods for shallow water systems. *Journal of Computational Physics*, 246:242–264, 2013.

- [30] M.J. Castro, C. Parés, G. Puppo, and G. Russo. Central schemes for non-conservative hyperbolic systems. *SIAM Journal on Scientific Computing*, 34:B523–B558, 2012.
- [31] V. Casulli. On eulerian-lagrangian methods for the navier-stokes equations at high reynolds number. *Journal for numerical methods in fluids*, 8:1349–1360, 1988.
- [32] V. Casulli. Semi-implicit finite difference methods for the two-dimensional shallow water equations. *Journal of Computational Physics*, 86:56–74, 1990.
- [33] V. Casulli. A semi-implicit finite difference method for non-hydrostatic free-surface flows. *International Journal for Numerical Methods in Fluids*, 30:425–440, 1999.
- [34] V. Casulli. A high-resolution wetting and drying algorithm for free-surface hydrodynamics. *International Journal for Numerical Methods in Fluids*, 60:391–408, 2009.
- [35] V. Casulli. A semi-implicit numerical method for the free-surface Navier-Stokes equations. *International Journal for Numerical Methods in Fluids*, 74:605–622, 2014.
- [36] V. Casulli and E. Cattani. Stability, accuracy and efficiency of a semi-implicit method for three-dimensional shallow water flow. *Computers & Mathematics with Applications*, 27:99–112, 1994.
- [37] V. Casulli and R. T. Cheng. Semi-implicit finite difference methods for three-dimensional shallow water flow. *International Journal for Numerical Methods in Fluids*, 15:629–648, 1992.
- [38] V. Casulli, M. Dumbser, and E. F. Toro. Semi-implicit numerical modeling of axially symmetric flows in compliant arterial systems. *International Journal for Numerical Methods in Biomedical Engineering*, 28:257–272, 2012.
- [39] V. Casulli and G. S. Stelling. Semi-implicit subgrid modelling of three-dimensional free-surface flows. *International Journal for Numerical Methods in Fluids*, 67:441–449, 2011.

- [40] V. Casulli and R. A. Walters. An unstructured grid, three-dimensional model based on the shallow water equations. *International Journal for Numerical Methods in Fluids*, 32:331–348, 2000.
- [41] V. Casulli and P. Zanolli. Semi-implicit numerical modeling of nonhydrostatic free-surface flows for environmental problems. *Mathematical and Computer Modelling*, 36:1131–1149, 2002.
- [42] V. Casulli and P. Zanolli. High resolution methods for multidimensional advection-diffusion problems in free-surface hydrodynamics. *Ocean Modelling*, 10:137–151, 2005.
- [43] V. Casulli and P. Zanolli. A nested newton-type algorithm for finite volume methods solving richards' equation in mixed form. *SIAM Journal on Scientific Computing*, 32:2255–2273, 2009.
- [44] V. Casulli and P. Zanolli. Iterative solutions of mildly nonlinear systems. *Journal of Computational and Applied Mathematics*, 236:3937–3947, 2012.
- [45] V. Chatzi and F.P. Preparata. Using pyramids in mixed meshes - point placement and basis functions. *Brown University Providence*, 2000.
- [46] S. Childress. New solutions of the kinematic dynamo problem. *Journal of Mathematical Physics*, 11:3063–3076, 1970.
- [47] A.J. Chorin. A numerical method for solving incompressible viscous flow problems. *Journal of Computational Physics*, 2:12–26, 1967.
- [48] A.J. Chorin. Numerical solution of the Navier-Stokes equations. *Mathematics of Computation*, 23:341–354, 1968.
- [49] E. T. Chung and C. S. Lee. A staggered discontinuous Galerkin method for the convection-diffusion equation. *Journal of Numerical Mathematics*, 20:1–31, 2012.
- [50] E.T. Chung, P. Ciarlet, and T.F. Yu. Convergence and superconvergence of staggered discontinuous Galerkin methods for the three-dimensional Maxwell's equations on Cartesian grids. *Journal of Computational Physics*, 235:14–31, 2013.

- [51] E.T. Chung and B. Engquist. Optimal discontinuous Galerkin methods for wave propagation. *SIAM Journal on Numerical Analysis*, 44:2131–2158, 2006.
- [52] E.T. Chung and B. Engquist. Optimal discontinuous Galerkin methods for the acoustic wave equation in higher dimensions. *SIAM Journal on Numerical Analysis*, 47:3820–3848, 2009.
- [53] B. Cockburn, S. Hou, and C. W. Shu. The Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: the multidimensional case. *Mathematics of Computation*, 54:545–581, 1990.
- [54] B. Cockburn, S. Y. Lin, and C. W. Shu. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: one dimensional systems. *Journal of Computational Physics*, 84:90–113, 1989.
- [55] B. Cockburn and C. W. Shu. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws II: general framework. *Mathematics of Computation*, 52:411–435, 1989.
- [56] B. Cockburn and C. W. Shu. The Runge-Kutta local projection P1-Discontinuous Galerkin finite element method for scalar conservation laws. *Mathematical Modelling and Numerical Analysis*, 25:337–361, 1991.
- [57] B. Cockburn and C. W. Shu. The Runge-Kutta discontinuous Galerkin method for conservation laws V: multidimensional systems. *Journal of Computational Physics*, 141:199–224, 1998.
- [58] Oriol Colomes, Santiago Badia, Ramon Codina, and Javier Principe. Assessment of variational multiscale models for the large eddy simulation of turbulent incompressible flows. *Computer Methods in Applied Mechanics and Engineering*, 285:32–63, 2015.
- [59] R. Courant, K. Friedrichs, and H. Lewy. Über die partiellen Differenzgleichungen der mathematischen Physik. *Mathematische Annalen*, 100:32–74, 1928.
- [60] J. Crank and P. Nicolson. A practical method for numerical evaluation of solutions of partial differential equations of the heat-conduction type.

- Mathematical Proceedings of the Cambridge Philosophical Society*, 43:50–67, 1947.
- [61] A. Crivellini and F. Bassi. An implicit matrix-free Discontinuous Galerkin solver for viscous and turbulent aerodynamic simulations. *Computers and Fluids*, 50:81–93, 2011.
- [62] A. Crivellini, V. D’Alessandro, and F. Bassi. High-order discontinuous Galerkin solutions of three-dimensional incompressible RANS equations. *Computers and Fluids*, 81:122–133, 2013.
- [63] V. Dolejsi. Semi-implicit interior penalty discontinuous galerkin methods for viscous compressible flows. *Communications in Computational Physics*, 4:231–274, 2008.
- [64] V. Dolejsi and M. Feistauer. A semi-implicit discontinuous galerkin finite element method for the numerical solution of inviscid compressible flow. *Journal of Computational Physics*, 198:727–746, 2004.
- [65] V. Dolejsi, M. Feistauer, and J. Hozman. Analysis of semi-implicit dgfm for nonlinear convection-diffusion problems on nonconforming meshes. *Computer Methods in Applied Mechanics and Engineering*, 196:2813–2827, 2007.
- [66] T. Dombre, U. Frisch, J. M. Greene, M. Hnon, A. Mehr, and A. M. Soward. Chaotic streamlines in the abc flows. *Journal of Fluid Mechanics*, 167:353–391, 6 1986.
- [67] M. Dumbser. Arbitrary high order PNPM schemes on unstructured meshes for the compressible Navier–Stokes equations. *Computers & Fluids*, 39:60–76, 2010.
- [68] M. Dumbser, D. S. Balsara, E. F. Toro, and C. D. Munz. A unified framework for the construction of one-step finite-volume and discontinuous Galerkin schemes. *Journal of Computational Physics*, 227:8209–8253, 2008.
- [69] M. Dumbser, M. Castro, C. Parés, and E.F. Toro. ADER schemes on unstructured meshes for non-conservative hyperbolic systems: Applications to geophysical flows. *Computers and Fluids*, 38:1731–1748, 2009.

- [70] M. Dumbser and V. Casulli. A staggered semi-implicit spectral discontinuous galerkin scheme for the shallow water equations. *Applied Mathematics and Computation*, 219(15):8057–8077, 2013.
- [71] M. Dumbser and V. Casulli. A conservative, weakly nonlinear semi-implicit finite volume scheme for the compressible navier-stokes equations with general equation of state. *Applied Mathematics and Computation*, 272:479–497, 2015.
- [72] M. Dumbser and M. Facchini. A space-time discontinuous Galerkin method for Boussinesq-type equations. *Applied Mathematics and Computation*, 272:336–346, 2015. DOI: 10.1016/j.amc.2015.06.052.
- [73] M. Dumbser, A. Hidalgo, M. Castro, C. Parés, and E. F. Toro. FORCE schemes on unstructured meshes II: Non-conservative hyperbolic systems. *Computer Methods in Applied Mechanics and Engineering*, 199:625–647, 2010.
- [74] M. Dumbser and M. Käser. Arbitrary high order non-oscillatory finite volume schemes on unstructured meshes for linear hyperbolic systems. *Journal of Computational Physics*, 221:693–723, 2007.
- [75] M. Dumbser, M. Käser, V. A Titarev, and E. F. Toro. Quadrature-free non-oscillatory finite volume schemes on unstructured meshes for nonlinear hyperbolic systems. *Journal of Computational Physics*, 226:204–243, 2007.
- [76] M. Dumbser and E. F. Toro. On universal Osher-type schemes for general nonlinear hyperbolic conservation laws. *Communications in Computational Physics*, 10:635–671, 2011.
- [77] M. Dumbser and E. F. Toro. A simple extension of the Osher Riemann solver to non-conservative hyperbolic systems. *Journal of Scientific Computing*, 48:70–88, 2011.
- [78] S.V. Ershkov. Non-stationary helical flows for incompressible 3d navier-stokes equations. *Applied Mathematics and Computation*, 274:611–614, 2015.
- [79] E. Erturk. Numerical solutions of 2d steady incompressible flow over a backward-facing step, part i: High reynolds number solutions. *Computers and Fluids*, 37:633–655, 2008.

-
- [80] E. Erturk, T.C. Corke, and C. Gokcol. Numerical solutions of 2-d dteady incompressible driven cavity flow at high reynolds numbers. *International Journal for numerical Methods in Fluids*, 48:747–774, 2005.
- [81] F. Fambri, M. Dumbser, and V. Casulli. An Efficient Semi-Implicit Method for Three-Dimensional Non-Hydrostatic Flows in Compliant Arterial Vessels. *International Journal for Numerical Methods in Biomedical Engineering*, 30:1170–1198, 2014.
- [82] M. Feistauer and V. Kucera. On a robust discontinuous galerkin technique for the solution of compressible flow. *Journal of Computational Physics*, 224:208–221, 2007.
- [83] E. Ferrer and R.H.J. Willden. A high order discontinuous galerkin finite element solver for the incompressible navierstokes equations. *Computer and Fluids*, 46:224–230, 2011.
- [84] M. Fortin. Old and new finite elements for incompressible flows. *International Journal for Numerical Methods in Fluids*, 1:347–364, 1981.
- [85] L. Fraenkel. On corner eddies in plane inviscid shear flow. *Journal of Fluid Mechanics*, 11:400–406, 1961.
- [86] P. Garcia-Navarro and M.E. Vázquez-Cendón. On numerical treatment of the source terms in the shallow water equations. *Computers & Fluids*, 29:951–979, 2000.
- [87] G. Gassner and D.A. Kopriva. A comparison of the dispersion and dissipation errors of gauss and gauss-lobatto discontinuous galerkin spectral element methods. *SIAM Journal on Scientific Computing*, 33:2560–2579, 2011.
- [88] G. Gassner, F. Lörcher, and C. D. Munz. A contribution to the construction of diffusion fluxes for finite volume and discontinuous Galerkin schemes. *Journal of Computational Physics*, 224:1049–1063, 2007.
- [89] U. Ghia, K. N. Ghia, and C. T. Shin. High-re solutions for incompressible flow using navier-stokes equations and multigrid method. *Journal of Computational Physics*, 48:387–411, 1982.
- [90] F. X. Giraldo, J. S. Hesthaven, and T. Warburton. Nodal high-order discontinuous galerkin methods for the spherical shallow water equations. *Journal of Computational Physics*, 181:499–525, 2002.

- [91] F. X. Giraldo and M. Restelli. High-order semi-implicit time-integrators for a triangular discontinuous galerkin oceanic shallow water model. *International Journal for Numerical Methods in Fluids*, 63:1077–1102, 2010.
- [92] L. Gosse. A well-balanced flux-vector splitting scheme designed for hyperbolic systems of conservation laws with source terms. *Comput. Math. Appl.*, 39:135–159, 2000.
- [93] L. Gosse. A well-balanced scheme using non-conservative products designed for hyperbolic systems of conservation laws with source terms. *Math Models Methods Appl Sci*, 11:339–365, 2001.
- [94] J. M. Greenberg and A. Y. Le Roux. A wellbalanced scheme for the numerical processing of source terms in hyperbolic equations. *SIAM Journal on Numerical Analysis*, 33:1–16, 1996.
- [95] F.H. Harlow and J.E. Welch. Numerical calculation of time-dependent viscous incompressible flow of fluid with a free surface. *Physics of Fluids*, 8:2182–2189, 1965.
- [96] R. Hartmann and P. Houston. Symmetric interior penalty DG methods for the compressible navier–stokes equations I: Method formulation. *Int. J. Num. Anal. Model.*, 3:1–20, 2006.
- [97] R. Hartmann and P. Houston. An optimal order interior penalty discontinuous galerkin discretization of the compressible navierstokes equations. *Journal of Computational Physics*, 227:9670–9685, 2008.
- [98] M. R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49:409–436, 1952.
- [99] J.S. Hesthaven and T. Warburton. Nodal high-order methods on unstructured grids. I. Time-domain solution of Maxwell’s equations. *Journal of Computational Physics*, 181:186–221, 2002.
- [100] J.S. Hesthaven and T. Warburton. *Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications*. Springer, 2008.
- [101] J. G. Heywood and R. Rannacher. Finite element approximation of the nonstationary Navier-Stokes Problem. I. Regularity of solutions and second order error estimates for spatial discretization. *SIAM Journal on Numerical Analysis*, 19:275–311, 1982.

- [102] J. G. Heywood and R. Rannacher. Finite element approximation of the nonstationary Navier-Stokes Problem. III. Smoothing property and higher order error estimates for spatial discretization. *SIAM Journal on Numerical Analysis*, 25:489–512, 1988.
- [103] C. W. Hirt and B. D. Nichols. Volume of fluid (VOF) method for dynamics of free boundaries. *Journal of Computational Physics*, 39:201–225, 1981.
- [104] P. Houston, C. Schwab, and E. Süli. Discontinuous hp–finite element methods for advection–diffusion–reaction problems. *SIAM Journal on Numerical Analysis*, 39:2133–2163, 2002.
- [105] P. Houston, B. Senior, and E. Süli. hp–discontinuous Galerkin finite element methods for hyperbolic problems: error analysis and adaptivity. *International Journal for Numerical Methods in Fluids*, 40:153–169, 2002.
- [106] P. Houston and E. Süli. hp–adaptive discontinuous Galerkin finite element methods for first–order hyperbolic problems. *SIAM Journal on Scientific Computing*, 23:1226–1252, 2002.
- [107] T.J.R. Hughes, M. Mallet, and M. Mizukami. A new finite element formulation for computational fluid dynamics: II. Beyond SUPG. *Computer Methods in Applied Mechanics and Engineering*, 54:341–355, 1986.
- [108] G. Jiang and C. W. Shu. On a cell entropy inequality for discontinuous Galerkin methods. *Mathematics of Computation*, 62:531–538, 1994.
- [109] N. Kanaris, D. Grigoriadis, and S. Kassinos. Three dimensional flow around a circular cylinder confined in a plane channel. *Physics of Fluids*, 23:064106, 2011.
- [110] H. Khurshid and K. A Hoffmann. A high order numerical scheme for incompressible navier-stokes equations. *The Arabian Journal for Science and Engineering*, 0:0–0, 2014.
- [111] B. Klein, F. Kummer, and M. Oberlack. A SIMPLE based discontinuous Galerkin solver for steady incompressible flows. *Journal of Computational Physics*, 237:235–250, 2013.
- [112] D.A. Kopriva. Metric identities and the discontinuous spectral element method on curvilinear meshes. *Journal of Scientific Computing*, 26:301–327, 2006.

- [113] D.A. Kopriva and G. Gassner. On the quadrature and weak form choices in collocation type discontinuous galerkin spectral element methods. *Journal of Scientific Computing*, 44:136–155, 2010.
- [114] S. C. Kramer and G. S. Stelling. A conservative unstructured scheme for rapidly varied flows. *International Journal for Numerical Methods in Fluids*, 58:183–212, 2008.
- [115] Hwar C. Ku, Richard S. Hirsh, and Thomas D. Taylor. A pseudospectral method for solution of the three-dimensional incompressible navier-stokes equations. *Journal of Computational Physics*, 70:439–462, 1987.
- [116] R. J. LeVeque. Balancing source terms and flux gradients in high-resolution godunov methods: The quasi-steady wavepropagation algorithm. *Journal of Computational Physics*, 146:346–365, 1998.
- [117] Y. J. Liu, C. W. Shu, E. Tadmor, and M. Zhang. Central discontinuous galerkin methods on overlapping cells with a non-oscillatory hierarchical reconstruction. *SIAM Journal on Numerical Analysis*, 45:2442–2467, 2007.
- [118] Y. J. Liu, C. W. Shu, E. Tadmor, and M. Zhang. L2-stability analysis of the central discontinuous galerkin method and a comparison between the central and regular discontinuous galerkin methods. *Mathematical Modeling and Numerical Analysis*, 42:593–607, 2008.
- [119] G. Dal Maso, P. G. LeFloch, and F. Murat. Definition and weak stability of nonconservative products. *J. Math. Pures Appl.*, 74:483–548, 1995.
- [120] H.F. Meier, J.J.N. Alves, and M. Mori. Comparison between staggered and collocated grids in the finite-volume method performance for single and multi-phase flows. *Computers and Chemical Engineering*, 23:247–262, 1999.
- [121] M.L. Munoz-Ruiz and C. Parés. On the convergence and well-balanced property of path-conservative numerical schemes for systems of balance laws. *Journal of Scientific Computing*, 48:274–295, 2011.
- [122] N.C. Nguyen, J. Peraire, and B. Cockburn. An implicit high-order hybridizable discontinuous galerkin method for the incompressible navier-stokes equations. *Journal of Computational Physics*, 230:1147–1170, 2011.

- [123] A. Nigro, S. Renda, C. De Bartolo, R. Hartmann, and F. Bassi. A high-order accurate discontinuous Galerkin finite element method for laminar low Mach number flows. *International Journal for Numerical Methods in Fluids*, 72:43–68, 2013.
- [124] C. Parés. Numerical methods for nonconservative hyperbolic systems: a theoretical framework. *SIAM Journal on Numerical Analysis*, 44:300–321, 2006.
- [125] V.S. Patankar. *Numerical Heat Transfer and Fluid Flow*. Hemisphere Publishing Corporation, 1980.
- [126] V.S. Patankar and B. Spalding. A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. *International Journal of Heat and Mass Transfer*, 15:1787–1806, 1972.
- [127] P.O. Persson, J. Bonet, and J. Peraire. Discontinuous Galerkin solution of the Navier-Stokes equations on deformable domains. *Computer Methods in Applied Mechanics and Engineering*, 198:1585–1595, 2009.
- [128] P.O. Persson and J. Peraire. Newton-GMRES preconditioning for discontinuous Galerkin discretizations of the Navier-stokes equations. *SIAM Journal on Scientific Computing*, 30(6):2709–2733, 2008.
- [129] O. Podvigina, P. Ashwin, and D. Hawker. Modelling instability of flow using a mode interaction between steady and hopf bifurcations with rotational symmetries of the cube. *Physica D: Nonlinear Phenomena*, 215(1):62 – 79, 2006.
- [130] Olga M. Podvigina. Spatially-periodic steady solutions to the three-dimensional navierstokes equation with the abc-force. *Physica D: Nonlinear Phenomena*, 128(24):250 – 272, 1999.
- [131] L. Qu, C. Norberg, L. Davidson, S.H. Peng, and F. Wang. Quantitative numerical analysis of flow past a circular cylinder at reynolds number between 50 and 200. *Journal of Fluids and Structures*, 39:347–370, 2013.
- [132] W. H. Reed and T. R. Hill. Triangular mesh methods for the neutron transport equation. Technical Report LA-UR-73-479, Los Alamos Scientific Laboratory, 1973.

- [133] F. Remihi, F. Aloui, S. B. Nasrallah, L. Doubriez, and J. Legrand. Experimental investigation of a confined flow downstream of a circular cylinder centred between two parallel walls. *Journal of Fluids and Structures*, 24:855–882, 2008.
- [134] S. Rhebergen, O. Bokhove, and J. J. W. van der Vegt. Discontinuous Galerkin finite element methods for hyperbolic nonconservative partial differential equations. *Journal of Computational Physics*, 227:1887–1922, 2008.
- [135] S. Rhebergen and B. Cockburn. A spacetime hybridizable discontinuous Galerkin method for incompressible flows on deforming domains. *Journal of Computational Physics*, 231:4185–4204, 2012.
- [136] S. Rhebergen, B. Cockburn, and Jaap J.W. van der Vegt. A space-time discontinuous Galerkin method for the incompressible NavierStokes equations. *Journal of Computational Physics*, 233:339–358, 2013.
- [137] V. V. Rusanov. Calculation of Interaction of Non–Steady Shock Waves with Obstacles. *J. Comput. Math. Phys. USSR*, 1:267–279, 1961.
- [138] Y. Saad and M.H. Schultz. GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 7:856–869, 1986.
- [139] H. Sakamoto and H. Haniu. A study on vortex shedding from spheres in a uniform flow. *Journal of Fluids Engineering*, 112:386–392, 1990.
- [140] K. Shahbazi, P. F. Fischer, and C. R. Ethier. A high-order discontinuous galerkin method for the unsteady incompressible navier-stokes equations. *Journal of Computational Physics*, 222:391–407, 2007.
- [141] G. S. Stelling and S. P. A. Duynmeyer. A staggered conservative scheme for every froude number in rapidly varied shallow water flows. *International Journal for Numerical Methods in Fluids*, 43:1329–1354, 2003.
- [142] A. H. Stroud. *Approximate Calculation of Multiple Integrals*. Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1971.
- [143] M. Tavelli and M. Dumbser. A high order semi-implicit discontinuous galerkin method for the two dimensional shallow water equations on staggered unstructured meshes. *Applied Mathematics and Computation*, 0:0–0, 2014.

- [144] M. Tavelli and M. Dumbser. A staggered arbitrary high order semi-implicit discontinuous galerkin method for the two dimensional incompressible navier-stokes equations. *Computer and Fluids*, 119:235–249, 2015.
- [145] M. Tavelli, M. Dumbser, and V. Casulli. High resolution methods for scalar transport problems in compliant systems of arteries. *Applied Numerical Mathematics*, 74:62–82, 2013.
- [146] C. Taylor and P. Hood. A numerical solution of the Navier-Stokes equations using the finite element technique. *Computers and Fluids*, 1:73–100, 1973.
- [147] E. F. Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics, Second Edition*. Springer-Verlag, 1999.
- [148] E. F. Toro. *Shock-Capturing Methods for Free-Surface Shallow Flows*. Wiley, 2001.
- [149] E. F. Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics*. Springer, third edition, 2009.
- [150] E. F. Toro, A. Hidalgo, and M. Dumbser. FORCE schemes on unstructured meshes I: Conservative hyperbolic systems. *Journal of Computational Physics*, 228:3368–3389, 2009.
- [151] G. Tumolo, L. Bonaventura, and M. Restelli. A semi-implicit, semi-Lagrangian, p-adaptive discontinuous Galerkin method for the shallow water equations . *Journal of Computational Physics*, 232:46–67, 2013.
- [152] J. van Kan. A second-order accurate pressure correction method for viscous incompressible flow. *SIAM Journal on Scientific and Statistical Computing*, 7:870–891, 1986.
- [153] R. Verfürth. Finite element approximation of incompressible Navier-Stokes equations with slip boundary condition II. *Numerische Mathematik*, 59:615–636, 1991.
- [154] R. A. Walters and V. Casulli. A robust finite element model for hydrostatic surface water flows. *Communications in Numerical Methods in Engineering*, 14:931–940, 1998.

- [155] T. Warburton and T. Hagstrom. Taming the CFL number for discontinuous galerkin methods on structured meshes. *SIAM Journal on Numerical Analysis*, 46:3151–3180, 2008.
- [156] C.H.K. Williamson. The existence of two stages in the transition to three-dimensionality of a cylinder wake. *Physics of Fluids*, 24:855–882, 1988.
- [157] C.H.K. Williamson and G.L. Brown. A series in $1/\sqrt{Re}$ to represent the strouhal-reynolds number relationship of the cylinder wake. *Journal of Fluids and Structures*, 12:1073–1085, 1998.
- [158] J. Womersley. Method for the calculation of velocity, rate of flow and viscous drag in arteries when the pressure gradient is known. *Journal of Physiology*, 127:553–563, 1955.

List of Tables

2.1	Numerical convergence results for $p = 0, 1, 2, 3$. " <i>Pre.T</i> " and " <i>Sim.T</i> " indicate the wallclock time expressed in seconds that was needed for the preprocessing stage and for the simulation, respectively.	30
2.2	Numerical convergence results for $p = 4, 5, 6$. " <i>Pre.T</i> " and " <i>Sim.T</i> " indicate the wallclock time expressed in seconds that was needed for the preprocessing stage and for the simulation, respectively.	31
2.3	Comparison between third and sixth order staggered unstructured semi-implicit DG schemes and explicit cell-centered unstructured path-conservative $P_N P_M$ schemes for the smooth vortex problem. The third order simulations use a mesh with $N_e = 20000$ and the sixth order simulations were done on the grid with $N_e = 3200$	32
2.4	Numerical verification of the exact C -property (well-balanced property) of the unstructured staggered semi-implicit DG scheme ($p = 4$) for different machine precisions at time $t = 0.1$. The L_2 and L_∞ errors refer to the free surface elevation η	33
2.5	Comparison between fourth order staggered unstructured semi-implicit DG scheme and unstructured path-conservative ADER-WENO finite volume method for the circular dambreak problem over a bottom step.	40
3.1	Numerical convergence results for $p = 0$ and $p = 1$	79
3.2	Numerical convergence results for $p = 2$ and $p = 3$	80
3.3	Numerical convergence results for the manufactured solution test problem with polynomial degrees $p = 1$ to $p = 4$ in space and time.	94
3.4	Numerical convergence results for the planar Womersley problem.	97
3.5	Numerical convergence results for the velocity vector field of the Taylor-Green vortex.	99

List of Tables

4.1	Numerical convergence results.	123
4.2	Numerical convergence results.	124
A.1	Strong scaling efficiency test for the Taylor-Green vortex on SuperMUC.	169
A.2	Weak scaling efficiency test for the Taylor-Green vortex on SuperMUC.	169

List of Figures

2.1	Example of a triangular mesh element with its three neighbors and the associated staggered edge-based dual control volumes, together with the notation used throughout the chapter.	11
2.2	Jumps of η on the main grid (top) and of \vec{V} on the dual grid (bottom) highlighted by the bold red lines.	13
2.3	Volume (top) and line (bottom) integration paths for $\mathcal{D}_{i,j}$. . .	23
2.4	Integration for $\mathcal{Q}_{i,j}$, the quadrature points (red) are transformed using the subparametric maps T	24
2.5	Subparametric approach (left) and Isoparametric approach (right) for $p = 2$	27
2.6	The thick solid lines indicate the primary triangular grid (top) and the dual quadrilateral grid (bottom) used for the small surface perturbation test of Leveque. The thin solid lines denote the subgrid associated with the nodal degrees of freedom of the high order isoparametric DG finite element scheme.	34
2.7	Numerical solution for the free surface η at times $t = 0.12$, $t = 0.24$, $t = 0.36$ and $t = 0.48$ for the 2D surface perturbation test problem ($\epsilon = 10^{-2}$) of LeVeque [116]. An unstructured staggered semi-implicit DG scheme with piecewise polynomials of degree $p = 4$ has been used, together with a mesh size of $h = 0.1$ and a time step of $\Delta t = 0.001$. 80 equidistant contour lines in the interval $\eta \in [0.99, 1.01]$ are shown.	35
2.8	Comparison between numerical and reference solution at times $t = 0.05$, $t = 0.1$ and $t = 0.15$ for the free surface elevation (left) and the velocity u (right) in the x direction.	36
2.9	Three-dimensional plot of the free surface elevation at times $t = 0$, $t = 0.05$, $t = 0.1$ and $t = 0.15$	36
2.10	Comparison between numerical and reference solution at time $t_{end} = 0.20$: free surface elevation η and $-b$ on the left; velocity u in the x direction on the right.	38

2.11	Main triangular grid with nodal subgrid (top left); dual grid with nodal subgrid (top right) and both grids (bottom) for the circular cylinder testcase; the blue triangles define the primary mesh, the red quadrilaterals define the dual mesh and the green triangular elements are the boundary elements for the dual grid.	39
2.12	Differences between the grid obtained with the curved isoparametric approach (left) and the non-curved approach (right). . .	40
2.13	Profile of η at time $t = t_{end}$ using the isoparametric staggered DG approach.	42
2.14	Streamlines and pressure contours using the isoparametric staggered semi-implicit DG scheme for the circular cylinder problem on an unstructured triangular mesh.	43
2.15	Comparison of the numerical solution against the exact solution for $\theta = [-\pi, \pi]$ and several radii. From the top to the bottom we have plotted the profile of η , u and v	44
2.16	Comparison between the velocity u for the curved approach and the non-curved approach at $r = 0.5$, in this last case the convection term is taken such as $F\vec{v} = \vec{v}$	45
3.1	Example of space-time elements \mathbf{T}_i^{st} (red) and \mathbf{R}_j^{st} (green) with $j \in S_i$	49
3.2	Computational domain used for the simulation of the Blasius boundary layer. The colors represent the horizontal velocity u .	80
3.3	Velocity profile with respect to the Blasius coordinate ξ	81
3.4	Numerical and reference solution for the Blasius boundary layer at $x = 0.4$ and $x = 0.6$	81
3.5	Main and dual grid used for the lid-driven cavity problem for $Re = 100, 400, 1000$	82
3.6	Velocity profiles (left) and streamlines (right) at $Re = 100$ and $Re = 400$ for the lid-driven cavity problem.	83
3.7	Velocity profiles (left) and streamlines (right) at $Re = 1000$ and $Re = 3200$ for the lid-driven cavity problem.	84
3.8	Grid and main notation used for the backward-facing step problem.	85
3.9	Comparison of the experimental data of Armaly et al. [3] with the numerical results obtained with the present semi-implicit staggered DG scheme and the numerical solution obtained in [79] for the reattachment point X1 in the backward-facing step problem.	85

3.10	Streamlines at Reynolds numbers $Re = 44, 113$ and 250 from top to bottom.	86
3.11	Streamlines at Reynolds numbers $Re = 316, 426$ and 633 from top to bottom.	87
3.12	Rotational inviscid flow past a circular half-cylinder. Left: Streamlines. Right: Analytical and numerical results for $r = 0.5$. The vertical lines show the dimension of the six curved elements that cover the half-cylinder.	88
3.13	Steady flow of an inviscid incompressible fluid around a circular cylinder. On the top left: streamlines and pressure contours at $t_{end} = 10$; Numerical and exact solution at $r = 1.0$, $r = 1.5$ and $r = 2.0$ for the velocity components u , v and pressure p from top right to the bottom right, respectively.	89
3.14	Strouhal-Reynolds number relationship for the present method, the method of Qu et al. [131] and experimental data of Williamson and Brown [157].	90
3.15	Dual mesh and vorticity contours of the von Karman vortex street generated at time $t = 500$ for $Re = 50$ (top) and $Re = 125$ (bottom).	91
3.16	Temporal evolution of the vorticity profile for $t = 15$, $t = 30$, $t = 50$, $t = 75$ from top left to bottom right at $Re = 200$	91
3.17	Left: comparison between the CPU time for the GMRES method and the CG method. Right: total number of iterations for the GMRES against the number of iterations for the CG method.	92
3.18	Time series for the axial velocity u and the pressure p computed at $(x, y) = (-0.5, 0)$ for the coarsest grid $N_e = 46$ and $N_t = 6$	95
3.19	Radial velocity profiles for $x = -0.2$ and $x = 0.3$ at times, from left to right, $t = [0.75, 0.5, 0.875, 1.0, 0.125]$. Comparison between exact and numerical solution.	96
3.20	Velocity field of the Taylor-Green vortex on the coarse grid $N_e = 40$ with $p = 4$. The edge-based dual grid is shown.	98
3.21	Vorticity pattern for the double shear layer test at times, from top left to bottom right, $t = 0.4; t = 0.8; t = 1.2; t = 1.8$	100
3.22	Streamlines along the circular cylinder at times, from top left to bottom right, $t = 25, 50, 100$ and $t = 200$	102
3.23	Laminar viscous flow past a circular cylinder. Profile for the vorticity and horizontal velocity u at time $t = 300$	103

4.1	On the left: main tetrahedral mesh with $S_i = \{j_1, j_2, j_3, j_4\}$. On the right: standard orientation used through this chapter. . . .	107
4.2	An example of dual hexahedral element based on the side Γ_j . . .	108
4.3	Isoparametric dual element example for $p = 3$. On the left a dual element with a $2D$ face on the curved boundary; On the right, an internal dual element, but with a $1D$ edge on the curved boundary.	119
4.4	From top left to bottom right: Comparison between our numerical results, the one obtained by Albensoeder et al in [2], and the two dimensional data from Ghia et al [89] for $Re = 400$; three-dimensional plot of the two secondary slices and grid space; streamlines and magnitude of u on slices $x - y$, $x - z$ and $y - z$.	120
4.5	From top left to bottom right: Comparison between our numerical results, the one obtained by Albensoeder et al in [2], and the two dimensional data from Ghia et al [89] for $Re = 1000$; three-dimensional plot of the two secondary slices and grid space; streamlines and magnitude of u on slices $x - y$, $x - z$ and $y - z$.	121
4.6	From top left to bottom right: Vorticity isolines for $vort = [0.8, 1.2, 2.0]$; pressure at $p = [-0.8, 0.0, 0.8]$ and streamlines in order to show the three-dimensionality of the considered problem.	125
4.7	From the top to the bottom: Pressure isolines, velocity magnitude and vorticity isolines from left to right at times $t = [0.5, 1.0]$ and for $Re = 800$	127
4.8	From the top to the bottom: Pressure isolines, velocity magnitude and vorticity isolines from left to right at times $t = [2.1, 4.8, 9.0]$ and for $Re = 800$	128
4.9	Time evolution of the kinetic energy dissipation rate $-dk/dt$ compared with DNS available data of Brachet et al [21] for $Re = 400, 800$ and $Re = 1600$	129
4.10	Vorticity pattern in the plane $x = 0$, $\frac{\pi}{8}$, $\frac{\pi}{4}$, $\frac{\pi}{2}$ from top left to bottom right for $Re = 800$, $t = 9s$	130
4.11	Vorticity pattern in the plane $z = 0$, $\frac{\pi}{8}$, $\frac{\pi}{4}$, $\frac{\pi}{2}$ from top left to bottom right for $Re = 800$, $t = 9s$	131
4.12	Plot of u at the middle of the tube in <i>one single</i> time control volume $T = [0.3, 0.6]$, from top left to bottom right are plotted at intermediate time levels $t = [0.3, 0.375, 0.45, 0.525, 0.6]$	133
4.13	Velocity profile at the plane $x = 2$ (left column) and velocity against exact solution at $x = 2$ and $z = 0$ (right column) at times, from top to bottom, $t = [0.15, 0.45, 0.75]$	134

4.14	Left: Three dimensional view of the isoparametric grid used in the test case; Right: Time series of u in the plane $x = 1$, $(y, z) = (0, 0)$ and $(y, z) = (0, 0.9)$. The time subdivisions represent the time step size $dt = 0.3$	135
4.15	3D plot of the domain Ω and sketch of the mesh on the boundary; The plotted iso-surfaces are corresponding to $u = 0.2, 0.4, 0.8$. . .	136
4.16	2D plot on the symmetry plane $z = 0$ where the velocity u in the $x - \xi$ plane is plotted.	137
4.17	Left: 2D plot on the symmetry plane $z = 0$ where the velocity u in the $x - y$ plane is plotted and the extraction line for the comparison. Right: Numerical and reference solution taken on the line $(x, y, z) = (0.4, y, 0)$	137
4.18	Value of u in the (y, z) -plane at $x = [0, 3.75, 7.5, 11.25, 15]$	138
4.19	Recirculation zones in the plane $(\frac{x}{S}, y)$ in the symmetry plane (top) and close to the side wall at $\frac{z}{S} = 12$ (bottom).	139
4.20	Recirculation zones in the plane $(\frac{x}{S}, \frac{z}{S})$ close to the bottom and close to the top wall.	140
4.21	Cut view of the computational domain with $N_e = 14403$	141
4.22	Contour isosurfaces for the spanwise velocity v in the (x, y) , (y, z) and 3D plot.	142
4.23	Side view and upper view of the particle path at $t = 300$	143
4.24	Time evolution of the particle path at times, from top left to bottom right, $t = [262.8, 265.5, 267.8, 270.2]$	144
4.25	Velocity magnitude at $t = t_{end}$ in the (x, y) and (x, z) -plane.	145
4.26	Half grid plot of Ω_1 and Ω_2 , respectively from left to right.	147
4.27	Velocity profile u , v and w from top to bottom past the circular cylinder in the sub-domain $\Omega = [-5, 20] \times [-\frac{r}{\beta}, \frac{r}{\beta}] \times [-7, 7]$ for $(Re, \beta) = (300, \frac{1}{5})$ and $(Re, \beta) = (277, \frac{1}{3})$ in the left and right column, respectively.	148
4.28	Isosurfaces of spanwise velocity $v = [\pm 0.1, \pm 0.03]$ and $w = \pm 0.03$ from top to bottom for the case $(Re, \beta) = (300, \frac{1}{5})$ (left column) and $(Re, \beta) = (277, \frac{1}{3})$ (right column).	149
4.29	Comparison between our numerical results, the one obtained by Albensoeder et al in [2], and the two dimensional data from Ghia et al [89] for $Re = 400$; three-dimensional plot of the two secondary slices and computational grid; streamlines and magnitude of u on slices $x - y$, $x - z$ and $y - z$	151

4.30	Comparison between our numerical results, the one obtained by Albensoeder et al in [2], and the two dimensional data from Ghia et al [89] for $Re = 1000$; three-dimensional plot of the two secondary slices and computational grid; streamlines and magnitude of u on slices $x - y$, $x - z$ and $y - z$	152
4.31	Detail of u (first two figures) and v (last two) close to the point $\vec{x} = (0.5, 0.5, 0)$ in the corrected case (left) and non corrected one (right).	153
4.32	Comparison with the reference solution of the three considered cases.	154
A.1	MPI boundary dual mesh issue.	161
A.2	Information needed to reconstruct the communicated informations.	162
A.3	Periodic Boundary conditions and MPI shadow issue.	165
A.4	Dual grid plot and MPI decomposition of the computational domain.	165
A.5	Vorticity pattern for the double shear layer test at times, from top left to bottom right, $t = 0.4$; $t = 0.8$; $t = 1.2$; $t = 1.8$	166
A.6	Vorticity pattern for the cavity at times from $t = 1$ and taken each $2sec$ from top left to bottom right.	167
A.7	Computational time plot, from left to right, for nonlinear convective-viscous term, main linear system and total time.	170
A.8	Really coarse grid with $p = 4$, (Left) before elastic transformation, (Right) after elastic transformation. From top to bottom: xy , xz , yz plane.	173
A.9	Construction of a general dual isoparametric element.	175
A.10	Typical shape for $o(n)$ and $\mathbf{p}(n)$	182
A.11	O' as function of n for several kind of f , $g = 0$	182
A.12	Experimental values of CPU time for several n and polynomial degrees.	183
A.13	Behavior of $e^{b(p)}$	184
A.14	Data and estimated value for the CPU time O and the error E	184
A.15	CPU time and polynomial degree as a function of n for $\epsilon = 10^{-5}$	185
A.16	$p(\epsilon)$ and $O(\epsilon)$ for several values of ϵ	185