UNIVERSITAT POLITÈCNICA DE CATALUNYA

DOCTORAL THESIS

Large scale Finite Element solvers for the large eddy simulation of incompressible turbulent flows

---

Author:
Oriol COLOMÉS GENÉ

Supervisor:
Prof. Santiago BADIA

A thesis submitted in fulfilment of the requirements
for the degree of Doctor of Philosophy

in the

Doctorat en Enginyeria Civil

Departament d'Enginyeria Civil i Ambiental

Barcelona, February 2016

Escola de Camins
Escola Tècnica Superior d'Enginyeria de Camins, Canals i Ports
UPC BARCELONATECH

To Raquel, my parents and my brother…

"Science is a way of life. Science is a perspective. Science is the process that takes us from confusion to understanding in a manner that's precise, predictive and reliable - a transformation, for those lucky enough to experience it, that is empowering and emotional."

Brian Greene

Abstract

In this thesis we have developed a path towards large scale Finite Element simulations of turbulent incompressible flows.

We have assessed the performance of residual-based variational multiscale (VMS) methods for the large eddy simulation (LES) of turbulent incompressible flows. We consider VMS models obtained by different subgrid scale approximations which include either static or dynamic subscales, linear or nonlinear multiscale splitting, and different choices of the subscale space. We show that VMS thought as an implicit LES model can be an alternative to the widely used physical-based models. This method is traditionally combined with equal-order velocity-pressure pairs, since it provides pressure stabilization. In this work, we also consider a different approach, based on inf-sup stable elements and convection-only stabilization. In order to do so, we define a symmetric projection stabilization of the convective term using an orthogonal subscale decomposition. The accuracy and efficiency of this method compared with residual-based algebraic subgrid scales and orthogonal subscales methods for equal-order interpolation is also assessed in this thesis.

Furthermore, we propose Runge-Kutta time integration schemes for the incompressible Navier-Stokes equations with two salient properties. First, velocity and pressure computations are segregated at the time integration level, without the need to perform additional fractional step techniques that spoil high orders of accuracy. Second, the proposed methods keep the same order of accuracy for both velocities and pressures. Precisely, the symmetric projection stabilization approach is suitable for segregated Runge-Kutta time integration schemes. This combination, together with the use of block-preconditioning techniques, lead to elasticity-type and Laplacian-type problems that can be optimally preconditioned using the balancing domain decomposition by constraints preconditioners. The weak scalability of this formulation have been demonstrated in this document.

Additionally, we also contemplate the weak imposition of the Dirichlet boundary conditions for wall-bounded turbulent flows.

Four well known problems have been mainly considered for the numerical experiments: the decay of homogeneous isotropic turbulence, the Taylor-Green vortex problem, the turbulent flow in a channel and the turbulent flow around an airfoil.

Resum

En aquesta tesi s'han desenvolupat diferents algoritmes per la simulació a gran escala de fluxos turbulents incompressibles mitjançant el mètode dels Elements Finits.

En primer lloc s'ha avaluat el comportament dels mètodes de multiescala variacional (VMS) basats en el residu, per la simulació de grans vòrtexs (LES) de fluxos turbulents. S'han considerat diferents models VMS tenint en compte diferents aproximacions de les subescales, que inclouen tant subescales estàtiques o dinàmiques, una definicó lineal o nolineal, i diferents seleccions de l'espai de les subescales. S'ha demostrat que els mètodes VMS pensats com a models LES poden ser una alternativa als models basats en la física del problema. Aquest tipus de mètode normalment es combina amb l'ús de parelles de velocitat i pressió amb igual ordre d'interpolació. En aquest treball, també s'ha considerat un enfocament diferent, basat en l'ús d'elements inf-sup estables conjuntament amb estabilització del terme convectiu. Amb aquest objectiu, s'ha definit un mètode d'estabilització amb projecció simètrica del terme convectiu mitjançant una descomposició ortogonal de les subescales. En aquesta tesi també s'ha valorat la precisió i eficiència d'aquest mètode comparat amb mètodes basats en el residu fent servir interpolacions amb igual ordre per velocitats i pressions.

A més, s'ha proposat un esquema d'integració en temps basat en els mètodes de Runge-Kutta que té dues propietats destacables. En primer lloc, el càlcul de la velocitat i la pressió es segrega al nivell de la integració temporal, sense la necessitat d'introduir tècniques de fraccionament del pas de temps. En segon lloc, els esquemes segregats de Runge-Kutta proposats, mantenen el mateix ordre de precisió tant per les velocitats com per les pressions. Precisament, els mètodes d'estabilització amb projecció simètrica són adequats per ser integrats en temps mitjançant esquemes segregats de Runge-Kutta. Aquesta combinació, juntament amb l'ús de tècniques de precondicionament en blocs, dóna lloc a problemes tipus elasticitat i Laplacià que poden ser òptimament precondicionats fent servir els anomenats *balancing domain decomposition by constraints preconditioners*. La escalabilitat dèbil d'aquesta formulació s'ha demostrat en aquest document.

Adicionalment, també s'ha contemplat la imposició de forma dèbil de les condicions de contorn de Dirichlet en problemes de fluxos turbulents delimitats per parets.

En aquesta tesi principalment s'han considerat quatre problemes ben coneguts per fer els experiments numèrics: el decaïment de turbulència isotròpica i homogènia, el problema del vòrtex de Taylor-Green, el flux turbulent en un canal i el flux turbulent al voltant d'una ala.

Acknowledgements

I would like to gratefully acknowledge the guidance, support and encouragement of my advisor, Santiago Badia. It has been a pleasure to do the Ph.D. thesis with you, Santi, and I want to thank you for your patience and for attending me all the countless times I have knocked your door.

I would also like to thank Javier Principe for his valuable advices and discussions, and also to Alberto Martín, without whom I would not have been able to meet this challenge.

I would especially like to thank Ramon Planas for his support and for acting as a mentor during the first years of my Ph.D. Also to Alba, with whom I have shared all this (long) journey. I also really appreciate the advice, help and discussions with my workmates during these years, Elisabet, Marc, Jesus, Eric, Hieu and Victor.

A special thanks to Raquel. Thank you for all the sacrifices you have done without asking anything in return, just making the things easier. I would also like to thank the support recieved from my whole family, particularly from my parents, Josep i Mercè, my brother Arnau and Carlota. Words cannot express how grateful I am to my mother, during these years you have taught me to overcome difficult situations and move forward.

Finally, I would like to acknowledge the financial support received from the Catalan Government through a FI fellowship.

Contents

List of Figures

List of Tables

List of Abbreviations

ASGS	**A**lgebraic **S**ub**G**rid **S**cale
BDDC	**B**alancing **D**omain **D**ecomposition by **C**onstraints
BDF	**B**ackward **D**ifferentiation **F**ormulas
BE	**B**ackward **E**uler
CFD	**C**omputational **F**luid **D**ynamics
CFL	**C**ourant-**F**riedrichs-**L**ewy
DAE	**D**ifferential **A**lgebraic **E**quation
DD	**D**omain **D**ecomposition
DFFT	**D**iscrete **F**ast **F**ourier **T**ransform
DHIT	**D**ecay of **H**omogeneous **I**sotropic **T**urbulence
DIRK	**D**iagonally **I**mplicit **RK**
DNS	**D**irect **N**umerical **S**imulation
DOF	**D**egree **O**f **F**reedom
EPS	**E**rror **P**er **S**tep
EPUS	**E**rror **P**er **U**nit **S**tep
FD	**F**inite **D**ifferences
FE	**F**inite **E**lement
FV	**F**inite **V**olume
HERK	**H**alf-**E**xplicit **RK**
HPC	**H**igh **P**erformance **C**omputing
ILES	**Implicit** **L**arge **E**ddy **S**imulation
IMEX	**IM**plicit-**EX**plicit
ISS	**I**nf-**S**up **S**table
LES	**L**arge **E**ddy **S**imulation
LPS	**L**ocal **P**rojection **S**tabilization
MG	**M**ulti**G**rid
MHD	**M**agneto**H**ydro**D**ynamics
MILES	**M**onotone **I**ntegrated **L**arge **E**ddy **S**imulation
MLBDDC	**M**ulti-**L**evel **BDDC**
ODE	**O**rdinary **D**ifferential **E**quation
OO	**O**bject-**O**riented
OSS	**O**rthogonal **S**ub**S**cales
PDE	**P**artial **D**ifferential **E**quation
RB	**R**esidual-**B**ased
RK	**R**unge-**K**utta

RMS	**R**oot **M**ean **S**quare
SGS	**S**ub**G**rid **S**cale
SRK	**S**egregated **R**unge-**K**utta
SV	**S**cott-**V**ogelius
TCF	**T**urbulent **C**hannel **F**low
TGV	**T**aylor-**G**reen **V**ortex flow
VMS	**V**ariational **M**ulti**S**cale
WMLES	**W**all-**M**odelled **L**arge **E**ddy **S**imulation
WRLES	**W**all-**R**esolved **L**arge **E**ddy **S**imulation

List of Symbols

Lower-case Roman

$\mathbf{a}(\mathbf{x},t)$	advection velocity field
$\mathbf{f}$	External body force
f_ν	viscous forces per unit volume
f_{in}	inertial forces per unit volume
k	wave number
k_η	Kolmogorov wave number
k_ℓ	integral scale wave number
ℓ	integral length scale
$\mathbf{n}$	unit outward normal vector
$\breve{p}$	pressure field
$p(\mathbf{x},t)$	kinematic pressure field
$\hat{p}(\mathbf{k},t)$	kinematic pressure field in the Fourier space
$\overline{p}(\mathbf{x})$	time-averaged pressure component
$p'(\mathbf{x})$	fluctuating pressure component
$\tilde{p}(\mathbf{x},t)$	pressure subscale
$\mathbf{t}$	unit vector tangential to the wall
$\mathbf{t}_N$	traction on the Neumman boundary
$\mathbf{u}(\mathbf{x},t)$	velocity field
$\hat{\mathbf{u}}(\mathbf{k},t)$	velocity field in the Fourier space
$\overline{\mathbf{u}}(\mathbf{x})$	time-averaged velocity component
$\mathbf{u}'(\mathbf{x})$	fluctuating velocity component
$\tilde{\mathbf{u}}(\mathbf{x},t)$	velocity subscale
u_x	velocity x-component
u_y	velocity y-component
u_z	velocity z-component
u	velocity RMS
u_τ	friction velocity
u_0	initial velocity
u^+	mean stream-wise velocity in wall units
$u_\perp$	velocity component perpendicular to a given surface
$u_\|$	velocity component parallel to a given surface
y^+	distance to the wall in wall units

Upper-case Roman

$\mathbb{A}_\tau$	momentum stabilization terms matrix
$\mathbb{B}_{\eta,\tau}$	velocity-projection block matrix
$\mathbb{B}_{\xi,\tau}$	pressure-projection block matrix
$\mathbf{C}$	cross stress tensor
$\mathbb{C}$	convective term matrix
$\mathbb{D}$	velocity divergence term matrix
$E(k)$	energy spectrum
$\mathbf{F}_u$	momentum force term vector
$\mathcal{F}$	implicit momentum operator
$\mathcal{G}$	explicit momentum operator
$\mathbb{G}$	pressure gradient term matrix
$\mathbf{H}$	continuity force term vector
$H^m(\Omega)$	Sovolev space $W^{m,p}(\Omega)$ for $p = 2$
$\mathbf{I}$	identity tensor
$\mathbb{I}$	identity matrix
I	moment of inertia
$\mathbb{K}$	diffusive term matrix
K	total kinetic energy
$\mathbf{L}$	Leonard stress tensor
$\mathcal{L}_\mathbf{a}$	momentum operator with advection velocity $\mathbf{a}$
$\mathcal{L}^*_\mathbf{a}$	adjoint operator of the momentum equation with advection velocity $\mathbf{a}$
$\mathbb{L}_\tau$	pressure gradient stabilization terms matrix
$L^p(\Omega)$	space of real functions on Ω with p-th power integrable
L_i	domain length on the i-direction
$\mathbb{M}$	mass matrix
$\mathbb{M}_{\eta,\tau}$, $\mathbb{M}_{\xi,\tau}$, $\mathbb{M}_\tau$	mass matrix weighted by τ_m
$\mathbf{P}$	pressure unknowns vector
$\mathcal{P}$	projection
P_D	Diagonal block preconditioner
P_U	Upper block preconditioner
P_L	Lower block preconditioner
$\mathcal{Q}$	pressure variational space
Q_{ij}	velocity correlation function
$\mathbf{R}$	SGS Reynolds stress tensor
$\mathbf{R}_u$	momentum equation residual
$\mathbb{R}^d$	Real coordinate space of d dimensions
R_p	continuity equation residual
Re	Reynolds number
Re_l	integral Reynolds number
Re_λ	Taylor-micorscale Reynolds number

Re_τ	friction Reynolds number
$R(r)$	autocovariance function
$\mathbb{S}$	Schur complement matrix
$S(r)$	skewness factor
$\mathbf{T}$	Reynolds stress tensor
$\mathcal{T}_h$	Finite Element partition
T	upper bound of the temporal domain
$\mathbf{U}$	velocity unknowns vector
$\mathcal{V}$	velocity variational space
$W^{m,p}(\Omega)$	Sovolev space
Z	enstrophy

Lower-case Greek

γ	angular distortion
δ	size of the boundary layer
δt	time step size
δ_ν	viscous length scale
∂_t	temporal partial derivative
$\partial\Omega$	domain boundary
$\boldsymbol{\varepsilon}(\mathbf{u})$	strain rate tensor
ε	mechanical energy dissipation rate
ε_h	energy transfer from Finite Element scale to subscales
$\tilde{\varepsilon}$	energy transfer from subscales to Finite Element scale
$\boldsymbol{\eta}_h$	convective term Finite Element projection
η	Kolmogorov length scale
λ	Taylor-microscale
μ	viscosity
ν	kinematic viscosity
ρ	density
$\boldsymbol{\xi}_h$	pressure gradient term Finite Element projection
$\boldsymbol{\xi}_m$	momentum residual Finite Element weighted projection
$\boldsymbol{\xi}_c$	continuity residual Finite Element weighted projection
$\boldsymbol{\sigma}(\mathbf{x}, t)$	stress tensor
τ	shear stress
τ_{xy}	shear stress xy-component
τ_w	wall shear stress
τ_m	momentum stabilization parameter
τ_c	continuity stabilization parameter
$\boldsymbol{\omega}$	vorticity field

Upper-case Greek

Γ	domain boundary ($\Gamma = \partial\Omega$)

Γ_D	Dirichlet boundary
Γ_N	Newmann boundary
$\Delta u(r)$	second-order structure function
Δx	stream-wise elemental length
Δy	wall-normal elemental length
Δz	span-wise elemental length
Π_h	projection into the Finite Element space
Π_m	projection into the Finite Element space weighted by τ_m
Π_c	projection into the Finite Element space weighted by τ_c
$\boldsymbol{\Upsilon}$	convective term projection unknowns vector
$\boldsymbol{\Xi}$	pressure gradient projection unknowns vector
Ω	bounded domain of $\mathbb{R}^d$

Superscripts and subscripts

$\tilde{(\cdot)}$	subscale component
$\dot{(\cdot)}$	time derivative
$\bar{(\cdot)}$	time-averaged component
$(\cdot)'$	fluctuating component
$(\cdot)^+$	quantity in wall units
$(\cdot)_h$	Finite Element component
$(\cdot)_{n+1}$	current time step
$(\cdot)_n$	previous time step
$(\cdot)_\Gamma$	boundary term

Chapter 1

Introduction

1.1 Motivation

The turbulent phenomena that takes place in fluid flows is one of the most fascinating and, at the same time, challenging problems of classical physics. We can find examples of turbulent flows in many situations of our daily life, from the most known and noticeable like the jets leftover by airplanes in the sky or the flow of a river in the mountains, to the most inconspicuous like the turbulent flow that we generate in the morning cup of coffee. Another peculiarity that makes turbulent flows captivating is the wide range of scales in which it appears, starting from the astrophysics with the turbulent flows developed within the stars, to the turbulence that is observed in the biological cells flow.

Despite all the efforts dedicated to understand the turbulent phenomena, this is a classical mathematical physics problem that remains hitherto unsolved. Many physicists, mathematicians and engineers have been studied this problem during the 19th and 20th centuries, and the prediction of turbulent behaviour with a certain degree of reliability is not completely understood yet. Thus, apart from the practical utility of a deep understanding of its nature, the study of turbulence is also motivated by its inherent intellectual challenge.

It is said that there are two main motivations to study turbulence, physics and engineering. From the physics point of view, the nature of turbulence must be explored to understand the behaviour of such flows at all levels. On the other hand, from the engineering perspective, there are some problems that have to be solved and a solution to them must be given with the knowledge we have now, although it might be incomplete. As an engineering thesis, the motivation of this work relies on the engineering point of view. The approach will be to contribute in developing novel techniques for the simulation of incompressible turbulent flows, making use of the current knowledge of turbulent flow phenomena.

Analytical solutions of fluid flows can only be obtained under certain restrictions and usually these restrictions cannot be satisfied in real applications. Thus, we need other approaches to obtain the solution of a fluid flow different from the analytical description. The numerical solution is an alternative to determine the behaviour of fluid flows, which basically consists on approximating the solution defined over a continuous domain by a discrete solution defined in a finite set of points. This technique is known as Computational Fluid Dynamics (CFD) and it is widely used, both in the engineering and the physics worlds.

As it is depicted in the schematic diagram shown in Figure 1.1, we can think that CFD leans on the intersection of three different topics: fluid mechanics, software engineering and physical applications. The fluid mechanics field give the basic knowledge of the physical phenomena that describes the fluid flow motion. The software engineering is needed to build codes able to simulate the fluid flow. Finally, we need the applications that give us the problem to be solved.

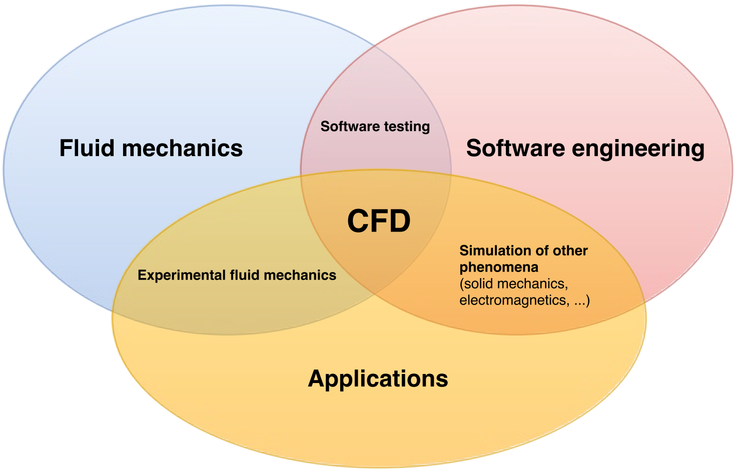


FIGURE 1.1: Topics that conform CFD.

The CFD is widely used in many disciplines and industries like in the aerospace, automotive, chemical manufacturing, power generation, petroleum exploration, medical research, meteorology or astrophysics. It is a very valuable tool since it leads to reductions in the cost of production by reducing the need of physical experimentation, improving the products or optimizing the production processes. In Table 1.1 we enumerate some more specific applications of CFD simulations that are very useful for different disciplines.

There already are CFD codes, both commercial and open source, that have the potential to solve a very broad spectrum of flow problems. However, the development of computational algorithms for the simulation of turbulent flows is still an open topic. Moreover, the improvement of accuracy, the reduction of computational time and the increase of accessibility are also ongoing objectives in CFD. It is true that CFD has some limitations, but the economic value of industrial applications has been demonstrated in a variety of industries.

The simulation of fluid flows relies on the fact that we can define a mathematical model that describes the fluid motion. Fluid dynamics are governed by the Navier-Stokes equations and the CFD basically consists on approximate the solution of these

Field	Application
Biomedical	Heart pumping Blood flow Air flow in lungs Nose and sinus flows Cell-fluid interface Artificial organ design Cardiac valve design Life support systems
Electronics	Cooling flow in electronic devices
Aerospace and automotive	Aerodynamic shape optimization Aerodynamic loads computation Turbines Propulsion systems Airbag Deployment In-Cylinder Engine Flow
Energy and power industry	Heat exchange modeling Wind turbines blade design Pulverized coal combustion Emission of NOx particles
Environmental	Impact of industrial exhausts Fire and smoke in buildings and tunnels Natural ventilation systems design Meteorology prediction
Civil	Effect of wind on structures Water flow in rivers Water management

TABLE 1.1: CFD applications.

equations. Many approximation techniques can be used to simulate fluid flows, the most common are the Finite Element (FE) method, the Finite Differences (FD) method or the Finite Volume (FV) method. All of them are methods that can approximate the solution of a fluid flow and vary in the way in which the continuous space is discretized.

The FD method is based on the application of a local Taylor expansion to approximate the governing equations, but it can only be applied on a discretization constructed by a network of topological squares or hexahedras, depending on the spatial dimensions. On the other hand, FE and FV methods are not restricted by this condition and are more extended in the CFD world. The FV method is based on the approximation of the average integral value on a reference volume. Rather than an integral average, the FE method is based on nodal approximations on a grid. In contraposition to FV, it allows the use of high-order approximations and has a strong mathematical foundation behind.

In this thesis the FE method will be considered to discretize in space the Navier-Stokes equations. In any case, it has to be highlighted that the FV method can be recovered with a particular definition of the FE method.

The increasing computing power is one of the points that make not only the CFD, but also the computational mechanics in general, more appealing to solve engineering problems with more and more complexity. In order to take advantage of the continuously growing computational power acquired with the new improvements on supercomputers, an advance in software design is imperative. New algorithms need to be designed to be used in the exascale computing environment.

We understand by exascale computing the capacity of a computing system to perform at least one exaFLOPs (10^{18} floating point operations per second). This computing capacity is still far from the current computer power, but many efforts are dedicated to achieve this objective, which is expected to be accomplished before 2020. There are many issues in the High-Performance Computing (HPC) community to be addressed in order to achieve the exascale goal. Some of them are related to the hardware development, including improvements on processors, memory size, memory bandwidth and energy consumption. Furthermore, there is the need of software improvement in such a way that exascale computing systems can be fully exploited.

Exascale computing is the key to be able to perform high-fidelity simulations of real world problems. Such simulations are a great challenge of computational physics and engineering, and can transform the computational science into a fully predictive science.

We still do not know much details about how exascale computing systems will be, but the definition and implementation of algorithms able to scale in extremely large computing environments (machines of the order of some million-cores) is a clear priority. We recall here that an optimal algorithm has the property to have a computational cost proportional to the size of the problem that is being solved. Other numerical issues that are thought to be needed to achieve exascale computations are the usage of high-order methods, which give more accurate results for a given problem size, and the development of adaptive methods, which can improve the efficiency of the simulation.

The application of a FE method to a Partial Differential Equation (PDE) leads to a matricial system of equations that has to be solved using linear algebra tools. The development of large scale FE solvers is accomplished by the use of preconditioners that improve the resolution of such matricial system. The most known algorithmically scalable preconditioners are the MultiGrid (MG) preconditioner and the Domain Decomposition (DD) based preconditioners. Nevertheless, in order to reach extreme scalabilty, the algorithmical scalability of a preconditioner is not a sufficient condition, since an efficient implementation is needed.

The development of algorithms and implementations of scalable preconditioners is a hot topic in the FE field and one of the main concerns of FEMPAR, an open source FE library whose name stands for Finite Element Multiphysics PARallel solvers. FEMPAR incorporates all the tools needed for the simulation of multiphysics problems, including the linear algebra methods used by the solver, the FE definition or the integration of the discretized (and linearized) PDEs. The resolution of linear systems in parallel with FEMPAR is based on a DD strategy.

FEMPAR is being developed within the Large Scale Scientific Computing (LSSC) group in the "Centre Internacional de Mètodes Numèrics en Enginyeria" (CIMNE), the group in which this thesis has been developed. More precisely, the FEMPAR code was born with the COMFUS project (COMputational techniques for FUSion reaction), which is a Starting Independent Research Grant awarded to Prof. Santiago Badia funded by the European Research Council and is the main financial support of the LSSC group.

1.2 Thesis objectives

Keeping the motivations described above in mind, let us define more precisely the main objectives that this thesis wants to achieve.

- **RB-VMS methods as LES models for turbulent incompressible flows**
  Since the simulation of turbulent flows is a very challenging task, if all scales of turbulent flows are wanted to be resolved, a Direct Numerical Simulation (DNS) has to be performed. This kind of techniques are extremely expensive and not feasible for practical applications. In order to reduce the computational cost of turbulent flow simulations, a technique which is gaining popularity in the CFD field is the Large Eddy Simulation (LES). LES models allow a reduction of the computational cost, since only large scales of the flow are simulated, while the smallest ones are modeled.

  The simulation of fluid flows using the standard Galerkin FE method suffer from two well-known numerical instabilities. On one hand we have the instability introduced by the convective term when convection-dominated flows are simulated, which is the case of turbulent flows. On the other hand, the velocity and pressure FE spaces need to satisfy a compatibility condition to guarantee stability. The Variational MultiScale (VMS) method is a framework for the development of stabilization techniques that overcome these two instabilities.

  One of the aims of this thesis is to assess the suitability of Residual-Based VMS (RB-VMS) methods as LES models for the simulation of incompressible turbulent flows. We will focus on the accuracy of such methods and their computational cost.

- **Mixed FE formulations for LES of turbulent incompressible flows**
  VMS methods overcome the two instabilities that arise when simulating turbulent incompressible flows using the Galerkin FE formulation, but the usage of

mixed FE formulations for the velocity and pressure spaces can also lead to stable methods in what the compatibility condition refers. Then, since when using these FE formulations the compatibility condition is satisfied by construction, only the convective instability needs to be controlled.

Based on VMS methods, a convection stabilization technique is aimed to be developed. As well as the RB-VMS methods, an assessment of its suitability as a LES model for the simulation of turbulent flows is also a target of this thesis.

- **High-order FE methods**
  One of the numerical issues to achieve exascale computations enumerated in the motivation section is the use of high-order methods. For a given number of degrees of freedom, the accuracy of the solution is improved with higher-order discretization schemes.

  It is true that using high-order methods the resulting system of equations is harder to solve due to the increase of connectivities between degrees of freedom, but the improvement on the results usually worths using them. This is one of the points that this thesis will assess.

  The development of *order-agnostic* algorithms (in the sense that arbitrary order of interpolation can be used) is also a goal that will be considered in this work.

- **High-order time integration methods**
  The accuracy in time integration is especially important when resolving turbulent flows. The multiscale behaviour of turbulent flows, also in what refers to the temporal scales, requires an accurate time integration scheme that, with sufficiently small time steps, give proper results.

  There are many high-order time integration schemes that can be used to solve the temporal flow evolution, but we will favour the so called $\theta-$methods and Runge-Kutta schemes.

  The definition of algorithms *order-agnostic* in time is also an objective. In this direction, the usage of Runge-Kutta schemes is appropriate since a parametrizable algorithm can be defined. Then, the order of the time integration scheme is given by the values of such parameters.

- **Adaptive time integration schemes**
  The fluid flow characteristics vary in space, but also in time. That means that a fluid flow can be transformed from laminar to turbulent under certain circumstances. Thus, for a continuously evolving fluid flow, it is natural to consider continuously evolving algorithms that can accommodate the flow characteristics evolution.

  When talking about the time integration, adaptive time integration schemes may be considered if one wants to adapt the time integration algorithm to the flow evolution. The use of Runge-Kutta based time integration schemes allow an easy implementation of adaptive time stepping techniques, and it is one of the reasons why this kind of methods will be considered in this thesis.

- **Segregation of velocity and pressure fields**
  The block-segregation of velocity and pressure fields can be achieved by the use of the popular pressure-correction or fractional-step methods. These kind of methods consider an auxiliar velocity or pressure extrapolated from previous time steps in order to split the computation of the problem fields. Since these techniques imply a modification of the equations to be solved, they induce the so called splitting error into the final solution.

  One of the goals of this thesis is to define a method that segregates velocity and pressure fields at the time discretization level, without introducing splitting errors.

- **Large scale and scalable FE solvers**
  As stated in the motivation, the definition and implementation of algorithms able to scale in extremely large computing environments is one of the requirements needed to have an exascale computing system. This implies to build scalable solvers that guarantee that the computational cost will not increase when more computational resources are used.

  In this thesis, the application of FE solvers able to reach extreme scalability is one of the main interests. In particular, the code that will be used in the thesis development is designed to be used in distributed computers and it incorporates DD methods optimal for large scale problems that can be combined in a natural way with block preconditioners.

  The solvers used in FEMPAR combine robust and scalable solvers for symmetric positive definite problems and block-preconditioning. This approach allows to deal with indefinite problems like the Stokes system and fluids at low Reynolds numbers. However, there is still the challenge to scale up non-symmetric and indefinite problems. This issue will be treated by the use of segregated velocity-pressure methods that lead to symmetric positive definite systems to be solved.

  When constructing block preconditioners we rely on the fact that the multifield problem (in the sense of having velocities and pressures) can be rewritten as a block system, i. e., one block for each unknown of the global problem. The non-diagonal blocks are those that couple the different subproblems. Here, the key decision is how to reorganize the system and subsequently how to define a good approximation to the original matrix, which will be used as a preconditioner.

- **Application**
  During the development of this thesis, real life applications have been considered

as a motivation of all the improvements that this work provides. All the previous goals are necessary to build faster and more accurate algorithms.

Although some of the benchmarks used in this thesis can represent a real-life problem, we also aim to apply all the techniques developed in this dissertation to a problem of interest in the industry. The turbulent flow around an airfoil could be a clear example of a problem that is being used in the areospace industry.

Moreover, all the advances induced by this work have been implemented in the FEMPAR code, and they will remain there. As FEMPAR is an open source code, all the code developed to achieve the thesis goals will be available for the society in general.

1.3 Document structure

This thesis is organized as follows:

In this first chapter (Chapter 1) we have described the main motivations in which this work has been based, as well as the main specific objectives that have been followed.

In Chapter 2, a general introduction to the FE framework is done. We begin with the description of the governing equations of the fluid flows in Section 2.1, where some details on the mathematical description of the physical phenomena of the fluid motion are given. In Section 2.2 we introduce some notation about functional spaces that will be used along the thesis. The variational formulation of the problem is given in Section 2.3, where we define the variational form of the problem at the continuous and discrete level and we define the concept of FE. After that, in Section 2.4 we describe the VMS method, which will be referenced by the core chapters of this thesis. In Section 2.5 a definition of the time integration schemes that are used in the forthcoming chapters is given. To finalize Chapter 2, a brief summary of what is contained in that chapter is given in Section 2.6.

An introduction to the turbulent phenomena is given in Chapter 3. A very basic introduction to the physical phenomena that lie behind turbulent flows is given in Section 3.1. More details about isotropic turbulence are given in Section 3.2, including some benchmarks that are used in the core chapters of this thesis. The specific details about wall-bounded turbulent flows are defined in Section 3.3. Finally, a summary of this chapter is given in Section 3.4.

The assessment of some RB-VMS methods for incompressible turbulent flows is done in Chapter 4. First, an introduction to the chapter is given in Section 4.1, where a description of the state of the art is presented. The mathematical formulation of the problem is defined in Section 4.2. Then, some energy balance statements are analyzed in Section 4.3, and the final discrete problem to be solved is described in Section 4.4. The results of the numerical experiments for different turbulent benchmarks are given in Section 4.5. Finally, the main contributions of this chapter are summarized in Section 4.6.

In Chapter 5 we propose a mixed FE formulation with convection stabilization for the simulation of turbulent incompressible flows. An introduction to the state of the art is done in Section 5.1. The problem statement is defined in Section 5.2, with a brief description of the Navier-Stokes equations and its weak formulation. After that, description of the VMS framework is introduced in Section 5.3, where apart from what we have defined in Section 2.4, the formulation of this method for mixed FE methods is analyzed. A block-preconditioning technique for the monolithic problem is defined in Section 5.4. The numerical experiments for different turbulent benchmarks are presented in Section 5.5. To sum up the chapter, some conclusions are stated in Section 5.6.

The development of novel Runge-Kutta based algorithms with velocity and pressure segregation is described in Chapter 6. We first introduce the state of the art for this chapter in Section 6.1. The problem is briefly stated in Section 6.2. After that, a detailed description of the segregated Runge-Kutta time integration schemes is given in Section 6.3. Some numerical experiments are carried out in Section 6.4 and the conclusions are stated in Section 6.5.

In Chapter 7 the method proposed in Chapter 5 together with the time integration scheme developed in Chapter 6 is assessed for the simulation of turbulent flows. After introducing the state of the art in Section 7.1, the problem statement is described in Section 7.2. A review of the relation between the proposed VMS method as an LES model is done in Section 7.3. The definition of the segregated Runge-Kutta methods, applied to the system of equations that arise from the mixed FE formulation with convection stabilization, is given in Section 7.4. Moreover, we also analyze the solvers that are used to solve the final discrete equations in Section 7.5. All this formulation is tested for two different turbulent benchmarks in Section 7.6, and, finally, some conclusions are pointed out in Section 7.7.

The formulation developed and tested in Chapter 7 is then applied to a realistic test like the turbulent flow around an airfoil, in Chapter 8. In this chapter a brief introduction to the problem is given in Section 8.1. The problem setting is defined in Section 8.2 and the numerical results are presented in Section 8.3. Finally some conclusions are stated in Section 8.4.

To conclude the thesis dissertation, in Chapter 9 we synthesize the various issues discussed in previous chapters, providing answers to the thesis research questions that we have formulated in Section 1.2. This synthesis is done in Section 9.1, where we also identify some limitations of the proposed methods. Finally, in Section 9.2 we propose some open lines of research that could improve the current work.

At the end of the document, we attach three appendices to extend some concepts explained in the body of the thesis. The first one, Appendix A gives some instructions on how to implement the energy spectra computation. In Appendix B, the implementation of the VMS methods developed in Chapter 4 is discussed. Lastly, in Appendix C the setting of the Butcher tableau's used by the SRK method are defined.

In order to improve the readability of this document, most of the chapters are self-contained. That means that some notation and definitions can be repeated in several parts of the document.

1.4 Publications and thesis development timeline

It must be emphasized that during the development of this thesis, four scientific articles have been written. Two of them have been accepted and published in peer reviewed journals, one more is under revision at the moment of submitting the thesis, and another one is about to be submitted in a short time.

To be more precise, the first contribution made during the thesis development was the article *Assessment of variational multiscale models for the large eddy simulation of turbulent incompressible flows*, published in *Computer Methods in Applied Mechanics and Engineering* journal in 2015 (see [56]). This work has motivated Chapter 4, where a set of residual-based VMS methods are assessed for turbulent incompressible flows.

With the aim of having more accurate and efficient time integration schemes, we developed the SRK method described in Chapter 6, which originated the article *Segregated Runge–Kutta methods for the incompressible Navier–Stokes equations*, published in *International Journal for Numerical Methods in Engineering* in 2016 (see [57]).

The velocity-pressure segregation introduced by the SRK methods are not applicable to the residual-based VMS methods introduced in Chapter 4. Thus, we investigated the suitability of using a mixed FE method with only convection stabilization which can be used together with SRK methods. From this study, we have written another article, *Mixed finite element methods with convection stabilization for the large eddy simulation of incompressible turbulent flows*, which is currently under revision (see [58]). Chapter 5 is devoted to this work.

Finally, in order to have efficient and scalable Finite Element solvers for the simulation of turbulent incompressible flows, we have mixed the formulation described in Chapter 5 and Chapter 6. This approach lead to the definition of *Segregated Variational Multiscale* methods, which are described in Chapter 7. Furthermore, with the aim of using this method for the simulation of wall-bounded flows, we also consider the weak imposition of Dirichlet boundary conditions. The application of the formulation described in Chapter 7 for a turbulent flow around an airfoil is studied in Chapter 8. The conjunction of Chapter 7 and Chapter 8 is the basis of the last article that is going to be submitted shortly.

Note that, although the main contributions are also reported on the cited articles, in this thesis we provide additional experiments and discussions to enrich the document.

Chapter 2

Preliminaries

2.1 The Governing equations of Fluid Mechanics

In this section we will briefly state the basic concepts that one has to know about fluid mechanics and its mathematical description, which will be used in the forthcoming chapters.

Let us start with a review of the equations that govern the fluid flow. In this thesis we will restrict to constant-property Newtonian fluids.

2.1.1 The momentum equation

From the Newton's second law, the momentum equation relates the fluid particle acceleration, $\frac{D\mathbf{u}}{Dt}$, to the external forces acting on such fluid particle. The external forces can be decomposed into the surface forces and the body forces, $\mathbf{f}$. The surface forces are described by the stress tensor $\boldsymbol{\sigma}(\mathbf{x},t)$ that is symmetric and, for constant-property Newtonian fluids, is defined by

$$\boldsymbol{\sigma} = -\breve{p}\mathbf{I} + 2\mu\boldsymbol{\varepsilon}(\mathbf{u}), \tag{2.1}$$

where $\breve{p}$ is the pressure field, $\mathbf{I}$ the identity tensor, μ the viscosity, $\mathbf{u}$ the fluid velocity and $\boldsymbol{\varepsilon}(\mathbf{u})$ the strain rate tensor, which is defined by the following expression

$$\boldsymbol{\varepsilon}(\mathbf{u}) = \frac{1}{2}\left[\nabla\mathbf{u} + (\nabla\mathbf{u})^T\right]. \tag{2.2}$$

The acceleration of the fluid particle is caused by the external forces according to the momentum equation

$$\rho\frac{D\mathbf{u}}{Dt} = \nabla\cdot\boldsymbol{\sigma} + \rho\mathbf{f}, \tag{2.3}$$

being ρ the density of the flow. Introducing the definition of the stress tensor (2.1) into (2.3), and dividing by the density ρ, which we consider to be constant, we get an alternative expression of the momentum equation

$$\frac{D\mathbf{u}}{Dt} = -\nabla p + \nabla\cdot(2\nu\boldsymbol{\varepsilon}(\mathbf{u})) + \mathbf{f}, \tag{2.4}$$

where we have used that $(1/\rho)\nabla\breve{p} = \nabla(\breve{p}/\rho) = \nabla p$, being $p := \breve{p}/\rho$ the kinematic pressure, and $\nu = \mu/\rho$ the kinematic viscosity.

2.1.2 Mass conservation

Matching the rate of change of mass in a given volume and the net mass flux across the boundary of such volume, and using the divergence theorem, we get the mass conservation or continuity equation, which is given by

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0, \tag{2.5}$$

which for constant density can be simplified to the kinematic condition that the velocity field be solenoidal or, what is the same, divergence-free:

$$\nabla \cdot \mathbf{u} = 0. \tag{2.6}$$

Equation (2.6) is also called incompressibility constraint, since it is a constraint on the fluid velocity.

2.1.3 Navier-Stokes equations

Let Ω be a bounded domain of $\mathbb{R}^d$, where $d = 2, 3$ is the number of space dimensions, $\Gamma = \partial\Omega$ its boundary and $(0, T]$ the time interval. The strong form of the steady Navier-Stokes problem that govern the fluid flow motion consists of finding the velocity field $\mathbf{u}$ and the pressure field p such that

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} - \nu \Delta \mathbf{u} + \nabla p = \mathbf{f} \quad \text{in } \Omega \times (0, T], \tag{2.7}$$

$$\nabla \cdot \mathbf{u} = 0 \quad \text{in } \Omega \times (0, T], \tag{2.8}$$

where the decomposition of the velocity material derivative into the temporal partial derivative plus the convective derivative, $\frac{D\mathbf{u}}{Dt} = \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u}$, and the fact that the velocity field is solenoidal have been used to simplify the momentum equation (2.4) into (2.7). In forthcoming sections, the temporal partial derivative $\frac{\partial(\cdot)}{\partial t}$ will also be denoted as $\partial_t(\cdot)$.

Equations (2.7) and (2.8) need to be supplied with appropriate boundary and initial conditions. The boundary Γ is divided into the Dirichlet (Γ_D) and the Neumann (Γ_N) parts such that $\Gamma_D \cup \Gamma_N = \Gamma$ and $\Gamma_D \cap \Gamma_N = \emptyset$. Then, the boundary and initial conditions can be written as

$$\mathbf{u} = \mathbf{u}_g \qquad \text{on } \Gamma_D \times (0, T], \tag{2.9}$$

$$(-p\mathbf{I} + \nu(\nabla \mathbf{u} + \nabla \mathbf{u}^T)) \cdot \mathbf{n} = \mathbf{t}_N \qquad \text{on } \Gamma_N \times (0, T], \tag{2.10}$$

$$\mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0(\mathbf{x}) \qquad \text{in } \Omega \times \{0\}, \tag{2.11}$$

$\mathbf{n}$ being the unit outward vector normal to Γ. For a solid wall, the velocity field on the Dirichlet boundary Γ_D is governed by two conditions. The first of them is the no penetration condition, the flow cannot penetrate the wall.

$$\mathbf{u} \cdot \mathbf{n} = 0 \quad \text{on } \Gamma_D \times (0, T]. \tag{2.12}$$

For the tangential velocity components we impose the so called no-slip condition, which means that there is no relative movement between the wall and the fluid.

$$\mathbf{u} \cdot \mathbf{t} = 0 \quad \text{on } \Gamma_D \times (0, T], \tag{2.13}$$

being $\mathbf{t}$ a unit vector tangential to the wall. Putting together equations (2.12) and (2.13) we have that $\mathbf{u} = \mathbf{0}$ on $\Gamma_D \times (0, T]$. If the wall is moving with a velocity $\mathbf{u}_g$ we recover the Dirichlet boundary condition (2.9).

2.1.4 Pressure and mass conservation

Let us now focus on the role that pressure field has in the fluid flow equations. If we take the divergence of the momentum equation (2.7), assuming that the continuity equation (2.8) is not satisfied ($\nabla \cdot \mathbf{u} = \epsilon$), we have

$$\left(\frac{\partial}{\partial t} - \nu \Delta \right) \epsilon = -\Delta p - \nabla \cdot (\mathbf{u} \cdot \nabla \mathbf{u}). \tag{2.14}$$

Considering problem (2.14) with the initial condition $\epsilon_0 = 0$, we can say that the velocity field will be solenoidal ($\epsilon = 0$) if, and only if, the following Poisson problem is satisfied

$$\Delta p = -\nabla \cdot (\mathbf{u} \cdot \nabla \mathbf{u}). \tag{2.15}$$

Hence, we can state that the satisfaction of the Poisson problem (2.15) is a necessary and sufficient condition for a solenoidal velocity field to remain solenoidal, see [150]. Furthermore, for infinite domains, the solution of (2.15) using the Biot-Savart law is given by

$$p(\mathbf{x}, t) = \frac{1}{4\pi} \int_\Omega \frac{\nabla \cdot (\mathbf{u}(\mathbf{y}, t) \cdot \nabla \mathbf{u}(\mathbf{y}, t))}{|\mathbf{x} - \mathbf{y}|} d\mathbf{y}. \tag{2.16}$$

An important consequence of (2.16) is that the pressure field is non-local, that means that a fluctuation at one point $\mathbf{y}$ affects to the hole domain. A direct repercussion of the non-locality of the pressure is that the pressure waves sent from $\mathbf{y}$ induce far-field pressure forces ($-\nabla p$) that can agitate the fluid motion at large distances from that point. Then, every part of the flow feels every other part. This consequence is more relevant in the case of turbulent fluid flows, where eddies at different locations of the flow can interact each other.

2.2 The function spaces

Before describing the variational formulation of Navier-Stokes equations we need an introduction to some functional spaces. In this section we define the notation used in following sections and we give some definitions of fundamental function spaces. A deeper explanation of the concepts introduced in this section can be found in any functional analysis text, for example we refer to [175], where we find a numerical analysis for the Stokes and Navier-Stokes equations. Other texts also enhance the concepts briefly defined in this section, see [35].

Let us start considering Ω to be an open set of $\mathbb{R}^n$ with boundary Γ. Otherwise stated, we assume that the boundary of Ω is locally Lipschitz. We denote by $L^p(\Omega)$, with $1 < p < +\infty$ (or $L^\infty(\Omega)$), the space of real functions defined on Ω with the p-th power absolutely integrable (or essentially bounded real functions for the case $p = \infty$). Note that we have made an abuse of notation for p, and the reader should not confuse the p-th order with the pressure field symbol defined in the previous section. This is a Banach space with the norm

$$\|\mathbf{u}\|_{L^p(\Omega)} := \left(\int_\Omega |\mathbf{u}(\mathbf{x})|^p d\Omega \right)^{\frac{1}{p}}$$

or, for $p = \infty$,

$$\|\mathbf{u}\|_{L^\infty(\Omega)} := \operatorname*{ess\,sup}_{\Omega} |\mathbf{u}(\mathbf{x})|).$$

For $p = 2$, $L^2(\Omega)$ is a Hilbert space with the scalar product

$$(\mathbf{u}, \mathbf{v})_\Omega := \int_\Omega \mathbf{u}(\mathbf{x})\mathbf{v}(\mathbf{x}) d\Omega. \tag{2.17}$$

Henceforth, when considering the scalar product over all domain Ω we will exclude the subscript, reading $(\cdot, \cdot)$. Furthermore, in forthcoming sections, the $L^2(\Omega)$-norm will be simply denoted as $\|\cdot\|$. The Sobolev space $W^{m,p}(\Omega)$ is the space of functions in $L^p(\Omega)$ with derivatives of order less than or equal to m in $L^p(\Omega)$, being m an integer and with $1 \leq p \leq +\infty$. This is a Banach space with the norm

$$\|\mathbf{u}(x)\|_{W^{m,p}(\Omega)} := \left(\sum_{j \leq m} \|D^j \mathbf{u}(\mathbf{x})\|^p_{L^p(\Omega)} \right)^{\frac{1}{p}},$$

where D^j is the differentiation operator. When $p = 2$, $W^{m,2}(\Omega) = H^m(\Omega)$ is a Hilbert space with the scalar product

$$(\mathbf{u}, \mathbf{v})_{H^m(\Omega)} := \sum_{j \leq m} \left(D^j \mathbf{u}, D^j \mathbf{v} \right).$$

Often we are concerned about n-dimensional vector functions with components in one of the spaces defined above. In this case we use bold characters to denote a vectorial space

$$\mathbf{L}^p(\Omega) := \{L^p(\Omega)\}^n, \quad \mathbf{H}^m(\Omega) := \{H^m(\Omega)\}^n.$$

Let $\mathcal{D}(\Omega)$ be the space of $\mathcal{C}^\infty$ functions with compact support contained in Ω. The closure of $\mathcal{D}(\Omega)$ in $H^m(\Omega)$ is denoted by $H_0^m(\Omega)$. The space $H_0^m(\Omega)$ can be though as the space of functions that belong in $H^m(\Omega)$ that vanish on the boundary Γ in a general sense. Functions that belong to $H_0^m(\Omega)$ satisfy the Poincaré inequality

$$\|\mathbf{u}\| \leq c(\Omega)\|\nabla \mathbf{u}\|, \quad \forall \mathbf{u} \in \mathbf{H}_0^m(\Omega). \tag{2.18}$$

From a physical point of view, as noticed in [73], we can think on the space $\mathbf{L}^2(\Omega)$ as the space of all vector fields $\mathbf{u}$ with finite kinetic energy. Moreover, the $\mathbf{H}^1(\Omega)$ can be thought as the space of all vector fields $\mathbf{u}$ with finite enstrophy. Further explanation of the kinetic energy and enstrophy concepts is given in Chapter 3.

Let q be the dual index to p, being $1 \leq p \leq +\infty$, i.e. $\frac{1}{q} + \frac{1}{p} = 1$, and k a negative integer. The Sobolev space $W^{k,p}(\Omega)$ is defined to be the dual space $\left(W^{-k,q}(\Omega)\right)'$. For $p = 2$, $q = 2$, and the Hilbert space $H^k(\Omega)$ is the dual space of $H^{-k}(\Omega)$. We define the duality pairing

$$\langle f, v\rangle_\Omega := \int_\Omega f(x)v(x)\, d\Omega, \quad \text{for } f \in H^{-1}(\Omega) \text{ and } v \in H_0^1(\Omega).$$

For simplicity hereinafter we will omit the subscript, $\langle\cdot,\cdot\rangle$, when the integral is over the domain Ω.

Let us consider some additional spaces that are useful in the mathematical description of the Navier-Stokes equations. A possible way to deal with the incompressibility constrain (2.8) is to consider a functional space with less regularity than $\mathbf{H}^1(\Omega)$ defined as

$$\mathbf{H}(div, \Omega) := \{\mathbf{u} \in \mathbf{L}^2(\Omega) | \nabla \cdot \mathbf{u} \in \mathbf{L}^2(\Omega)\},$$

which is a Hilbert space with the norm

$$\|\mathbf{u}\|_{div} := \|\mathbf{u}\| + \|\nabla \cdot \mathbf{u}\|.$$

The closure of $\mathcal{D}(\Omega)$ in $\mathbf{H}(div, \Omega)$ is denoted by $\mathbf{H}_0(div, \Omega)$.

Let us assume that a and b are two extended real numbers, $-\infty \leq a < b \leq \infty$, and let $\mathcal{X}$ be a Banach space. For a given α, $1 \leq \alpha + \infty$, we denote as $L^\alpha(a, b; \mathcal{X})$ the space of integrable functions from $[a, b]$ into $\mathcal{X}$.

2.3 The variational formulation

2.3.1 Continuous formulation

Let us consider the strong form of the Navier-Stokes problem (2.7)-(2.11). In order to formulate the equivalent variational problem we define a set of variational spaces that incorporates the homogeneous Dirichlet boundary condition and the temporal evolution

$$\mathcal{V}_g := \left\{ \mathbf{v} \in \mathbf{H}^1(\Omega) : \mathbf{v}|_{\Gamma_D} = \mathbf{u}_g \right\} \equiv \mathbf{H}^1_g(\Omega), \tag{2.19}$$

$$\mathcal{V}_0 := \left\{ \mathbf{v} \in \mathbf{H}^1(\Omega) : \mathbf{v}|_{\Gamma_D} = 0 \right\} \equiv \mathbf{H}^1_0(\Omega), \tag{2.20}$$

$$\mathcal{Q} := \mathbf{L}^2(\Omega)/\mathbb{R}. \tag{2.21}$$

Given sufficiently smooth functions $\mathbf{v} \in \mathcal{V}_0$ and $q \in \mathcal{Q}$, we obtain the variational or weak version of the Navier-Stokes equations multiplying (2.7) by $\mathbf{v}$ and (2.8) by q, integrating over Ω and integrating by parts the second order derivatives. Then the variational Navier-Stokes problem reads: find $\mathbf{u} \in \mathbf{L}^2(0,T;\mathcal{V}_g)$,
and $p \in \mathbf{L}^1(0,T;\mathcal{Q})$ such that:

$$(\partial_t \mathbf{u}, \mathbf{v}) + \left(\nu \left(\nabla \mathbf{u} + \nabla \mathbf{u}^T\right), \nabla \mathbf{v}\right) + b(\mathbf{u},\mathbf{u},\mathbf{v}) + (\nabla p, \mathbf{v}) = \langle f, \mathbf{v} \rangle \qquad \forall \mathbf{v} \in \mathcal{V}_0, \tag{2.22}$$

$$(\nabla \cdot \mathbf{u}, q) = 0 \qquad \forall q \in \mathcal{Q}. \tag{2.23}$$

Adding up equations (2.22)-(2.23) we obtain an alternative weak form of the incompressible Navier-Stokes problem (2.7)-(2.11) consists in finding $[\mathbf{u}, p] \in L^2(0,T;\mathcal{V}_g) \times \mathcal{D}'(0,T;\mathcal{Q})$ (distributions in time with values in $\mathcal{Q}$) such that

$$(\partial_t \mathbf{u}, \mathbf{v}) + B(\mathbf{u}; [\mathbf{u}, p], [\mathbf{v}, q]) = \langle \mathbf{f}, \mathbf{v} \rangle \qquad \forall \mathbf{v} \in \mathcal{V}_0, \quad \forall q \in \mathcal{Q}, \tag{2.24}$$

satisfying the initial condition (2.11) in a weak sense. Here the form $B(\mathbf{a}; [\mathbf{u}, p], (\mathbf{v}, q))$ is defined as

$$B(\mathbf{a}; [\mathbf{u}, p], [\mathbf{v}, q]) := \nu(\nabla \mathbf{u}, \nabla \mathbf{v}) + b(\mathbf{a}, \mathbf{u}, \mathbf{v}) - (p, \nabla \cdot \mathbf{v}) + (q, \nabla \cdot \mathbf{u}) \tag{2.25}$$

where the trilinear weak form of the convective term $b(\mathbf{u}, \mathbf{v}, \mathbf{w})$ can be written in the following three equivalent ways

$$b(\mathbf{u},\mathbf{v},\mathbf{w}) = (\mathbf{u} \cdot \nabla \mathbf{v}, \mathbf{w}) \qquad \text{Non conservative,} \tag{2.26}$$

$$b(\mathbf{u},\mathbf{v},\mathbf{w}) = \frac{1}{2}(\mathbf{u} \cdot \nabla \mathbf{v}, \mathbf{w}) - \frac{1}{2}(\mathbf{v}, \mathbf{u} \cdot \nabla \mathbf{w}) \qquad \text{Skew-symmetric (type 1),} \tag{2.27}$$

$$b(\mathbf{u},\mathbf{v},\mathbf{w}) = (\mathbf{u} \cdot \nabla \mathbf{v}, \mathbf{w}) + \frac{1}{2}(\mathbf{v} \cdot \mathbf{w}, \nabla \cdot \mathbf{u}) \qquad \text{Skew-symmetric (type 2).} \tag{2.28}$$

Note that in the trilinear weak forms (2.26)-(2.28) the boundary integral terms that arise from the integration by parts have been neglected. This assumption is valid when strong Dirichlet boundary conditions are considered over all the boundary. Despite of that, this equivalence is lost at the discrete level. The skew-symmetric form (type 2) (4.10) is very common when numerical analysis are presented [15, 39, 86] but the skew-symmetric form (type 2) (4.9) has important advantages when the first argument is a discontinuous function, as will be shown in forthcoming chapters.

The well-posedness of problem (2.24) relies on the called LBB condition, which stands for the name of the authors that developed works related to that condition. See the works by Ladyzhenskaya [126], Babuŝka [10] and Brezzi [36]. The LBB condition is also called *inf-sup* condition and reads as follows: there exist a positive constant β such that,

$$\inf_{q\in\mathcal{Q}}\sup_{\mathbf{v}\in\mathcal{V}_0}\frac{(\nabla\cdot\mathbf{v},q)}{\|\mathbf{v}\|_{\mathcal{V}}\|q\|_{\mathcal{Q}/\ker A^t}}\geq\beta>0, \tag{2.29}$$

being A^t the adjoin of the operator defined as

$$A:\mathcal{V}_0\to\mathcal{Q}' \quad | \quad \langle A(\mathbf{v}),a\rangle_{\mathcal{Q}'\times\mathcal{Q}}=(\nabla\cdot\mathbf{v},q) \qquad \forall\mathbf{v}\in\mathcal{V}_0, \quad \forall q\in\mathcal{Q} \tag{2.30}$$

2.3.2 The Finite Element method

In order to approximate the solution of the variational problem (2.24), one needs to construct finite-dimensional spaces in which the solution can be computed. The approach followed in this work to construct such finite-dimensional spaces is the called Finite Element (FE) method. According to Ciarlet, see [48], we can define a FE as follows.

Let $K\subseteq\mathbb{R}^n$ be a bounded closed set with nonempty interior and piece-wise smooth boundary, the element domain. Let $\mathcal{S}$ be a finite-dimensional space of functions on K, the space of shape functions. Let $\mathcal{N}=\{\mathcal{N}_1,\mathcal{N}_2,...,\mathcal{N}_k\}$ be a basis for $\mathcal{S}'$, the set of nodal variables. Then, $(K,\mathcal{S},\mathcal{N})$ is called a FE.

We refer to Brenner et al [35] for a deeper explanation of the FE definitions.

In this thesis we will mainly use FE spaces composed by quadrilateral finite elements built from a tensor product of polynomials. For the 3D case, we consider a reference FE $(\widetilde{K},\widetilde{\mathcal{S}},\widetilde{\mathcal{N}})$ with $\widetilde{K}$ a cube defined in $[-1,1]^3$, $\widetilde{\mathcal{S}}=Q_k$ being

$$Q_k:=\left\{\sum_j c_jp_j(x)q_j(y)r_j(z):\ \text{with } p_j,\, q_j \text{ and } r_j \text{ polynomials of degree } j\leq k\right\},$$

and $\widetilde{\mathcal{N}}$ denoting the point evaluations at $\{(t_l,t_m,t_n):\ l,m,n=0,1,...,k\}$ where $\{-1=t_0<t_1<...<t_k=1\}$.

Let us now consider a FE partition $\mathcal{T}_h$ of the domain Ω composed by a set of elements $\{K_e\}_{e=1}^{ne}$, being ne the total amount of elements in the domain. Let us consider F_K a mapping from K to $\widetilde{K}$, i. e. $F_K(K)=\widetilde{K}$ with its pull-back map defined as $F_K^*(\hat{f}):=\hat{f}\circ F_K$, see [49, 35] for more details on equivalence between FEs.

Then the FE spaces for the velocity and pressure fields equivalent to (2.19)-(2.21) can be defined as

$$\mathcal{V}_h := \left\{ \mathbf{v}_h \in (\mathcal{C}^0(\Omega))^d : \mathbf{v}_h|_K = \tilde{\mathbf{v}} \circ F_K^{-1},\ \tilde{\mathbf{v}} \in (Q_{k_v})^d,\ K \in \mathcal{T}_h \right\}, \tag{2.31}$$

$$\mathcal{V}_{g,h} := \left\{ \mathbf{v}_h \in \mathcal{V}_h : \mathbf{v}_h|_{K \cap \Gamma_D} = \mathbf{u}_g \right\}, \tag{2.32}$$

$$\mathcal{V}_{0,h} := \left\{ \mathbf{v}_h \in \mathcal{V}_h : \mathbf{v}_h|_{K \cap \Gamma_D} = 0 \right\}, \tag{2.33}$$

$$\mathcal{Q}_h := \left\{ \mathcal{C}^0(\Omega) \cap \mathbf{L}^2(\Omega)/\mathbb{R} : q_h|_K = \tilde{q} \circ F_K^{-1},\ \tilde{q} \in Q_{kq},\ K \in \mathcal{T}_h \right\}. \tag{2.34}$$

Where k_v and k_q, not necessarily equal, are the degree of the polynomials used to define the interpolation space for the velocity and pressure fields, respectively. In what follows, the subindex h will denote functions related to the FE space. Note that in this work both velocity and pressure field spaces, $\mathcal{V}_h$ and $\mathcal{Q}_h$, are considered to be made by continuous functions in the same partition of the domain, $\mathcal{T}_h$.

2.3.3 Semi-discrete formulation

Let us consider a FE partition $\mathcal{T}_h$ of the domain Ω from which we can construct conforming finite dimensional spaces for the velocity $\mathcal{V}_{g,h} \subset \mathcal{V}_g$, and for the pressure $\mathcal{Q}_{0,h} \subset \mathcal{Q}_0$. The spaces $\mathcal{V}_{g,h}$ and $\mathcal{Q}_{0,h}$ are the ones defined in the previous section, equations (2.32) and (2.34) respectively.

The Galerkin FE approximation of (2.24) consists in finding $[\mathbf{u}_h, p_h] \in L^2(0,T;\mathcal{V}_{g,h}) \times \mathcal{D}'(0,T;\mathcal{Q}_h)$ such that

$$(\partial_t \mathbf{u}_h, \mathbf{v}_h) + B(\mathbf{u}_h; [\mathbf{u}_h, p_h], [\mathbf{v}_h, q_h]) = \langle \mathbf{f}, \mathbf{v}_h \rangle \qquad \forall \mathbf{v}_h \in \mathcal{V}_{0,h}, \quad \forall q_h \in \mathcal{Q}_h, \tag{2.35}$$

Problem (2.35) is well posed if the discrete *inf-sup* condition equivalent to (2.29) is satisfied. The discrete version reads: there exist a positive constant β_d, independent of h, such that,

$$\inf_{q_h \in \mathcal{Q}_h} \sup_{\mathbf{v}_h \in \mathcal{V}_{0,h}} \frac{(\nabla \cdot \mathbf{v}_h, q_h)}{\|\mathbf{v}_h\|_{\mathcal{V}_h} \|q\|_{\mathcal{Q}_h / \ker A_h^t}} \geq \beta_d > 0, \tag{2.36}$$

with A_h the equivalent operator to the one defined in (2.30).

2.4 The Variational Multiscale method

Let us consider a FE partition $\mathcal{T}_h$ of the domain Ω from which we can construct conforming finite dimensional spaces for the velocity $\mathcal{V}_{0,h} \subset \mathcal{V}_0$, and for the pressure $\mathcal{Q}_{0,h} \subset \mathcal{Q}_0$.

It is well known that the Galerkin FE approximation (2.35) has numerical instabilities for high mesh Reynolds number problems, i.e., when the nonlinear convective term dominates the viscous term. Another drawback of that formulation is the discrete *inf-sup* condition that must be satisfied by the pair $\mathcal{V}_{0,h} \times \mathcal{Q}_{0,h}$ in order to have a well-posed problem with bounded pressure. These difficulties are overcome by using the VMS approach, introduced by Hughes in [100, 101], and that is stated as follows.

Let us consider a two-scale decomposition of spaces $\mathcal{V}_0$ and $\mathcal{Q}_0$ such that

$$\mathcal{V}_0 = \mathcal{V}_{0,h} \oplus \widetilde{\mathcal{V}}_0$$

and

$$\mathcal{Q}_0 = \mathcal{Q}_{0,h} \oplus \widetilde{\mathcal{Q}}_0,$$

where $\widetilde{\mathcal{V}}_0$ and $\widetilde{\mathcal{Q}}_0$ are infinite-dimensional spaces that complete the FE spaces in $\mathcal{V}_0$ and $\mathcal{Q}_0$, respectively. Hereinafter the subscript $(\cdot)_h$ will denote the FE component and the tilde $\widetilde{(\cdot)}$ the subgrid component. Applying the two-scale decomposition to (2.24) we obtain a discrete problem

$$\begin{aligned}(\partial_t \mathbf{u}_h, \mathbf{v}_h) + (\partial_t \tilde{\mathbf{u}}, \mathbf{v}_h) + B(\mathbf{a}; [\mathbf{u}_h, p_h], [\mathbf{v}_h, q_h]) \\ + (\tilde{\mathbf{u}}, \mathcal{L}^*_{\mathbf{a}}(\mathbf{v}_h, q_h))_h - (\tilde{p}, \nabla \cdot \mathbf{v}_h) = \langle \mathbf{f}, \mathbf{v}_h \rangle ,\end{aligned} \tag{2.37}$$

where $(\cdot,\cdot)_h = \sum_{K \in \mathcal{T}_h} (\cdot,\cdot)_K$ is the sum of scalar products (2.17) over each element K of the partition $\mathcal{T}_h$, and

$$\mathcal{L}^*_{\mathbf{a}}(\mathbf{v}_h, q_h) := -\nu \nabla^2 \mathbf{v}_h - \mathbf{a} \cdot \nabla \mathbf{v}_h - \nabla q_h \tag{2.38}$$

is the formal of the adjoint operator of the momentum equation. The term involving the adjoint operator comes from an element-wise integration by parts of the terms involving the subscales, in which the boundary terms $(\mathbf{v}_h, \nu \mathbf{n} \cdot \nabla \tilde{\mathbf{u}})_{\partial h}$ and $(q_h, \mathbf{n} \cdot \tilde{\mathbf{u}})_{\partial h}$ have been neglected (the subscript ∂h is used to denote the sum over all elements of the integral on the boundary of each element). It also involves the approximation $b(\mathbf{a}, \tilde{\mathbf{u}}, \mathbf{u}_h) \approx -(\tilde{\mathbf{u}}, \mathbf{a} \cdot \nabla \mathbf{v}_h)$ which implies neglecting $(\mathbf{v}_h, \mathbf{n} \cdot \mathbf{a}\tilde{\mathbf{u}})_{\partial h}$ and $(\tilde{\mathbf{u}}, \nabla \cdot \mathbf{a}\mathbf{v}_h)$. These approximations are discussed in [55] together with the choice of $\mathbf{a}$ which defines the type of scale splitting (linear or nonlinear), also discussed below.

The discrete problem depends on $\tilde{\mathbf{u}} \in \widetilde{\mathcal{V}}_0$ and on $\tilde{p} \in \widetilde{\mathcal{Q}}_0$, $\widetilde{\mathcal{V}}_0$ and $\widetilde{\mathcal{Q}}_0$ being infinite-dimensional. Therefore, the equations for $\tilde{\mathbf{u}}$ and $\tilde{p}$ obtained after applying the two-scale decomposition cannot be directly solved, but some modeling steps are needed to obtain a feasible method. Considering the subscale as a time-dependent variable of the problem (see below) and approximating the Navier-Stokes operator by two stabilization parameters τ_m^{-1} and τ_c^{-1} (see for example [55]), the fine scale problem can be written as

$$\partial_t \tilde{\mathbf{u}} + \tau_m^{-1} \tilde{\mathbf{u}} = \mathcal{P}(\mathbf{R}_u), \tag{2.39}$$

$$\tau_c^{-1} \tilde{p} = \mathcal{P}(R_p). \tag{2.40}$$

In (2.39)-(2.40) $\mathcal{P}$ denotes the projection onto the space of subscales, which is discussed below. In turn, the vector $\mathbf{R}$ is the residual of the Navier-Stokes equations (2.7)-(2.8), defined as $\mathbf{R} = [\mathbf{R}_u, R_p]^T$, with

$$\mathbf{R}_u = \mathbf{f} - \partial_t \mathbf{u}_h - \mathcal{L}_{\mathbf{a}}(\mathbf{u}_h, p_h), \tag{2.41}$$

$$R_p = -\nabla \cdot \mathbf{u}_h. \tag{2.42}$$

where

$$\mathcal{L}_{\mathbf{a}}(\mathbf{v}_h, q_h) := -\nu \nabla^2 \mathbf{v}_h + \mathbf{a} \cdot \nabla \mathbf{v}_h + \nabla q_h \tag{2.43}$$

Finally, the expressions of the stabilization parameter τ_m is

$$\tau_m = \left(\frac{c_1 \nu}{h^2} + \frac{c_2 |\mathbf{a}|}{h} \right)^{-1}, \tag{2.44}$$

whereas we consider two possible definitions of τ_c, viz. $\tau_c = 0$ (which implies $\tilde{p} = 0$) and

$$\tau_c = \frac{h^2}{c_1 \tau_m}, \tag{2.45}$$

where h is the mesh size and c_1 and c_2 are algorithmic constants. Let us comment on expression (2.44):

- The influence of the constants c_1 and c_2 is discussed in Section 4.5.4. A theoretical way to determine them would be to impose that the numerical dissipation they introduce be equal to the molecular dissipation in turbulent regimes, as explained in [85].

- The definition of τ_m in (2.44) is not standard, in the sense that the one used often depends on the time step size of the time discretization, δt. Instead of (2.44), $\tau_m^{-1} = \frac{1}{\delta t} + \frac{c_1 \nu}{h^2} + \frac{c_2 |\mathbf{a}|}{h}$ is more often considered (see, e.g., [98, 78]). We refer to Section 4.5.5 for a more detailed discussion about this topic. Likewise, other expressions with the same asymptotic behavior in terms of h, ν and $|\mathbf{a}|$ can also be employed.

- Expression (2.44) corresponds to linear isotropic elements. If elements of order p are used (p is not the pressure, here), c_1 must be replaced by $c_1 p^4$ and c_2 by $c_2 p$. For anisotropic elements, the definition of h within each element is not obvious. A possibility is explained in [151].

In the following three sections we discuss the particular ingredients of our VMS models. A different summary can also be found in [54], together with some numerical experiments.

2.4.1 The dynamics of the subscales

Stabilized formulations were originally developed for steady convection-diffusion [38] and Stokes [68, 102] problems. As the numerical instabilities have a spatial nature, the time dependency of the subscales was not considered, and the standard choice [103, 105, 23] was to take

$$\tilde{\mathbf{u}} = \tau_m \mathcal{P}(\mathbf{R}_u), \tag{2.46}$$

that is, to neglect the temporal derivative of the subscales in (2.39). In this case, the subscales are called quasi-static in what follows.

The subscale as a time dependent variable of the problem was introduced in [51, 55]. It gives rise to important properties like commutativity of space and time discretization, stability without restrictions on the time step size [55, 13] and, combined with orthogonal subscales, to convergence towards weak solutions of the Navier-Stokes equations [15] and the possibility of predicting backscatter [54, 151].

Equation (2.39) can be analytically integrated to give

$$\tilde{\mathbf{u}}(t^*) = \tilde{\mathbf{u}}(0) + \mu^{-1}(t^*) \int_0^{t^*} \mu(t)\mathcal{P}\mathbf{R}_u \mathrm{d}t, \qquad \mu(s) = \exp \int_0^s \tau^{-1}(t)\mathrm{d}t, \tag{2.47}$$

where it is explicitly seen that the subscale is a function of the residual but also of the flow history. In practice this integration is performed numerically, as described below.

2.4.2 (Non)linear scale splitting

The original VMS formulation [100, 101] was developed having linear problems in mind and its extension to the Navier-Stokes equations was implicitly based on a "linearization", fixing the advection velocity and applying the multiscale splitting to the rest of the terms. A nonlinear scale splitting was used in [103, 105] together with an explicit resolution of the small scales in which a Smagorinsky damping was introduced. A nonlinear scale splitting with modeled subscales was used in [51, 23] and in [55], where it was shown that it leads to global conservation of momentum. We therefore consider both options

$$\mathbf{a} = \mathbf{u}_h \qquad \text{for linear subscales,} \tag{2.48}$$

$$\mathbf{a} = \mathbf{u}_h + \tilde{\mathbf{u}} \qquad \text{for nonlinear subscales.} \tag{2.49}$$

Remark 2.4.1. *When we use the nonlinear definition for the advection velocity,* $\mathbf{a} = \mathbf{u}_h + \tilde{\mathbf{u}}$*, the skew-symmetric term type 2 (2.28) in the FE equation (2.35) reads:*

$$b(\mathbf{a}, \mathbf{u}_h, \mathbf{v}_h) = ((\mathbf{u}_h + \tilde{\mathbf{u}}) \cdot \nabla \mathbf{u}_h, \mathbf{v}_h) + \frac{1}{2}(\mathbf{u}_h \cdot \mathbf{v}_h, \nabla \cdot \mathbf{u}_h) + \frac{1}{2}(\mathbf{u}_h \cdot \mathbf{v}_h, \nabla \cdot \tilde{\mathbf{u}}). \tag{2.50}$$

The last term is not well-defined, since it includes derivatives of the discontinuous subscale $\tilde{\mathbf{u}}$*. One possibility is to neglect it (as previously done with other similar terms when arriving to (2.35)), which implies*

$$b(\mathbf{a}, \mathbf{u}_h, \mathbf{u}_h) = -\frac{1}{2}(|\mathbf{u}_h|^2, \nabla \cdot \tilde{\mathbf{u}}), \tag{2.51}$$

the same result obtained when the non conservative form is used. By contrast, the skew-symmetric term type 1 in the FE equation (2.35) reads

$$b(\mathbf{a}, \mathbf{u}_h, \mathbf{v}_h) = \frac{1}{2}((\mathbf{u}_h + \tilde{\mathbf{u}}) \cdot \nabla \mathbf{u}_h, \mathbf{v}_h) - \frac{1}{2}((\mathbf{u}_h + \tilde{\mathbf{u}}) \cdot \mathbf{u}_h, \nabla \cdot \mathbf{v}_h) \tag{2.52}$$

from where

$$b(\mathbf{a}, \mathbf{u}_h, \mathbf{u}_h) = 0. \tag{2.53}$$

In Subsection 4.5.1 we will see the influence of the two forms of the convective term on the results. It is worth noting that the same approximations have been introduced in all cases to implement $b(\mathbf{a}, \tilde{\mathbf{u}}, \mathbf{u}_h)$, but these approximations are taken into account in the (usual) energy estimates of Section 4.3.

Remark 2.4.2. *At the continuous level, the different expressions of the convective term are also equivalent to the so called conservation form*

$$b(\mathbf{u}, \mathbf{v}, \mathbf{w}) = -(\mathbf{u} \otimes \mathbf{v}, \nabla \mathbf{w}).$$

In the discrete problem, the nonlinear scale splitting leads to the following terms in the momentum equation:

$$b(\mathbf{a}, \mathbf{u}_h + \tilde{\mathbf{u}}, \mathbf{v}_h) = -(\mathbf{u}_h \otimes \mathbf{u}_h, \nabla \mathbf{v}_h) - (\mathbf{u}_h \otimes \tilde{\mathbf{u}}, \nabla \mathbf{v}_h) - (\tilde{\mathbf{u}} \otimes \mathbf{u}_h, \nabla \mathbf{v}_h) - (\tilde{\mathbf{u}} \otimes \tilde{\mathbf{u}}, \nabla \mathbf{v}_h). \tag{2.54}$$

Even if this is not exactly what we get using the non-conservative or skew-symmetric forms because of the approximation error, this allows us to interpret the different contributions arising from the nonlinear scale splitting. As it is explained in [54], from (2.54) we can identify the contributions from the cross stresses, the Reynolds stresses and the subgrid scale tensor.

2.4.3 The space for the subscales

The selection of the space for the approximation of the subscales determines the projection $\mathcal{P}$ appearing in the right-hand side of (2.39) and (2.40). The first option, already considered in [103, 105, 23] and named Algebraic Subgrid Scale (ASGS) in [50] is to take the subscales in the space of the residuals, that is,

$$\mathcal{P} := \mathbf{I}. \tag{2.55}$$

Another possibility introduced in [50] is to consider the space of the subscales orthogonal to the FE space. The main motivation of the method is that a stability estimate for the projection onto the FE space of the pressure and/or the convective terms can already be obtained in the standard Galerkin method and therefore the only "missing" part is the orthogonal one. The Orthogonal Subscales (OSS) method is then characterized by the following projection definition:

$$\mathcal{P} := \Pi_h^{\perp} = \mathbf{I} - \Pi_h, \tag{2.56}$$

where Π_h is the projection onto the FE space. With this choice, the residual of the momentum equation does not depend on $\partial_t \mathbf{u}_h$. Likewise, $\mathcal{P}(\mathbf{f})$ in this case is only well defined for $\mathbf{f} \in L^2(\Omega)^d$. In the case of minimum regularity, $\mathbf{f} \in H^{-1}(\Omega)^d$, this term can be simply neglected without upsetting the accuracy of the method.

In fact, with this choice, the orthogonality between the space of subscales and the FE space is only guaranteed when the stabilization parameters are constant. If this is not the case, the method is still optimally convergent [52] but this property is lost. In order to have truly orthogonal subscales, *which guarantees a proper separation of the FE and the subgrid scale kinetic energies* (see Section 4.3) a slight modification of the projection Π_h is needed (see [52]). We will use two different weighted projections: one for the velocity subscales (Π_m) in (2.39) and another for the pressure subscales (Π_c) in (2.40). We define the weighted projections Π_m and Π_c such that given any vector $\mathbf{w} \in \mathcal{V}_0$ and any scalar $r \in \mathcal{Q}_0$ we have

$$(\tau_m \Pi_m(\mathbf{w}), \mathbf{v}_h) = (\tau_m \mathbf{w}, \mathbf{v}_h) \qquad \forall \mathbf{v}_h \in \mathcal{V}_{0,h}, \tag{2.57}$$

$$(\tau_c \Pi_c(r), q_h) = (\tau_c r, q_h) \qquad \forall q_h \in \mathcal{Q}_{0,h}. \tag{2.58}$$

These definitions guarantee the orthogonality between the FE and subscale spaces in the case of static subscales, that is, neglecting temporal derivatives in (2.39). It then follows that the term containing the temporal derivative of the subscale in the FE equation (2.37) also vanishes.

However, if the dynamic version of the method is used, the weight of the projection (2.57) must be conveniently modified to ensure the mentioned orthogonality. As it can be seen in (2.47), the definition of the weight depends on the time integration strategy, as explicitly stated in Section 4.4.

2.5 Time integration

In this section, our aim is to state some basic concepts about the time discretization. Once we have defined the semi-discrete problem, as it has been formulated in Section 2.3.3, we end up with an Ordinary Differential Equation (ODE), which has to be integrated in order to get the solution of the problem.

In the current work we only will consider the called *direct integration* methods. The *direct integration* of the transient equations rely on a numerical step-by-step procedure, where the word *direct* means that no transformation of the ODE problem is carried out a priory. Looking at the literature, many techniques can be found based in this kind of procedure. For instance in [22] the description of the following methods can be found: the central difference method, the Houbold method, the Newmark method or the θ-method. An exhaustive analysis of such methods can be found in [26].

Other commonly used time integration methods in the computational fluid dynamics field are the Backward Differentiation Formulas (BDF), the generalized-α method or the Runge-Kutta time integration schemes, see [34, 108, 65, 91] for instance.

Within the forthcoming chapters, only the θ-methods and Runge-kutta schemes are used to integrate in time the incompressible Navier-Stokes equations. Consequently, in the following subsections only these two methods are described.

2.5.1 θ-method

Let suppose that we have an initial-value problem with first order ODE of the form

$$\frac{\partial u}{\partial t} = f(t,u), \quad \in (0,T) \tag{2.59}$$

$$u(0) = u_0. \tag{2.60}$$

One of the most popular, widely used and simplest method to solve problem (2.61)-(2.60) are the so-called single step (one-step) schemes, particularly, the theta-method, which is usually denoted as θ-method. Using the notation $u(t_n) = u_n$, the θ-method is defined as

$$\begin{aligned} u_{n+1} &= u_n + h(\theta f(t_{n+1}, u_{n+1}) + (1-\theta) f(t_n, u_n)), \\ u(0) &= u_0, \end{aligned}$$

being h the time step size and $t_n = nh$ for $n = 0, ..., N$, with $N = T/h$. Here $\theta \in [0,1]$ is a fixed parameter. The θ-method is considered here as basic method since it represents the most simple Runge-Kutta method (and also linear multistep method). The case of $\theta = 0.5$ is of second order and is called Crank-Nicolson method. For $\theta = 0$ we have the so called (explicit) Forward Euler method and for $\theta = 1$ the (implicit) Backward Euler (BE) method.

2.5.2 Implicit-explicit Runge-Kutta schemes

One of the main goals of this thesis is the construction of efficient solvers for the resolution of the incompressible turbulent Navier-Stokes equations. The solver efficiency can be addressed not only by the use of efficient time integrators, but also by the application of efficient algebraic solvers for the final discrete system of equations. In this direction, the time integration scheme can help to construct smaller systems of equation segregating the different variables that appear on the problem and allowing to solve efficiently each uncoupled variable separately.

In Chapter 6 and Chapter 7 we will consider the application of Implicit-Explicit (IMEX) Runge-Kutta methods for the time integration of the Navier-Stokes equations. The aim is to take advantage of the IMEX schemes for Runge-Kutta methods to uncouple the pressure and velocity degrees of freedom when solving these equations. We also want to use the Runge-Kutta background to implement an adaptive time stepping technique to solve efficiently transient incompressible flow problems.

Given an ODE problem of the type

$$\frac{\partial u}{\partial t} = f(u) + g(u), \tag{2.61}$$

being f and g different operators which definition depends on the specific problem, an IMEX scheme consists of applying an explicit discretization for the operator f and an implicit discretization for g. This approach comes from the fact that ODEs usually are composed by operators of different nature. For instance, thinking in the convection-diffusion problem we have two different operators, which represent the convection term (let us denote it by f) and the diffusion term (which we will denote as g). As it is well known, the convection term is often nonlinear (i.e. burgers equation) while the diffusion term is generally linear and stiff. When $f \equiv 0$, problem (2.61) results in a stiff and linear system, which is natural to be solved using an implicit scheme. Otherwise, if $g \equiv 0$ the problem becomes nonlinear and it could be convenient to solve it using an explicit time integration scheme. Ascher et al. in [7] study some multistep IMEX methods for convection-diffusion problem type. Often these type of methods are used in conjunction with spectral methods, see [43, 120].

The IMEX approach can be used not only for multistep schemes, but also for Runge-Kutta time integration techniques. The multistage nature of the Runge-Kutta methods also make feasible IMEX schemes with even better properties than multistep methods, see [6] where some Runge-Kutta IMEX schemes are developed for the convection-diffusion problem.

The idea of the Runge-Kutta methods is to approximate the integral $u(t_{n+1}) = u(t_n) + \int_{t_n}^{t_{n+1}} [f(u) + g(u)]\ dt$ using a numerical quadrature with the points $c_1, ..., c_s$ and their weights $b_1, ...b_s$, which leads to

$$u(t_{n+1}) = u(t_n) + h \sum_{i=1}^{s} b_i \left(f(u(t_n + c_i h)) + g(u(t_n + c_i h)) \right) + \text{Error}. \tag{2.62}$$

Hereafter we will write t_i instead of $t_n + c_i h$. Suppose we have an approximation u_n to $u(t_n)$); to use (2.62) we also need values u_i to put in for $u(t_i)$. We compute them also by numerical quadratures on the same nodes:

$$u_i = u_n + h \sum_{j=1}^{s} a_{ij} \left(f(u_j) + g(u_j) \right). \tag{2.63}$$

In general this is a set of implicit equations, which we solve and use in (2.62) for our next value

$$u_{n+1} = u_n + h \sum_{i=1}^{s} b_i \left(f(u_i) + g(u_i) \right). \tag{2.64}$$

The formulation (2.63)-(2.64) define a Runge-Kutta method, which we designate by displaying its coefficient in the called Butcher tableau:

$$\begin{array}{c|cccc} c_1 & a_{11} & a_{12} & \dots & a_{1s} \\ c_2 & a_{21} & a_{22} & \dots & a_{2s} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_s & a_{s1} & a_{s2} & \dots & a_{ss} \\ \hline & b_1 & b_2 & \dots & b_s \end{array} \tag{2.65}$$

Runge-Kutta techniques have been widely used for a lot of ODE problems. The Navier-Stokes semidiscrete problem is not an exception and the use of Runge-Kutta methods for its time integration can be easily found in the literature, e. g. [142, 158, 159, 171]. However, like the multistep methods, the Runge-Kutta schemes need to solve several systems of equations at each time step. Further, when we use an implicit scheme all stages could be coupled, resulting a large system of equations to be solved. This drawback can be bypassed using an explicit scheme which only needs to evaluate the operators that arise from the Navier-Stokes problem. But the use of an explicit scheme, as it is well known, involve a restriction in the time-step size in order to ensure stability, see for instance the chapter IV.2 in [92].

Diagonally Implicit Runge-Kutta methods (DIRK) can be used to avoid stability problems and solving implicitly each Runge-Kutta stage uncoupledly, see [1]. This technique consists on setting all the Butcher tableau values a_{ij} in (2.65) that are above the diagonal to zero. That is, $a_{ij} = 0$ for all $j > i$. In fact, in [1] the use of the DIRK term is what in [92] is referred by Singly Diagonally Implicit Runge-Kutta methods (SDIRK) which means that all the diagonal terms are equal, $a_{ii} = \gamma$. As pointed out by Alexander in [1], the use of SDIRK methods allow to use the same LU-factorization when solving repeatedly the multistage system of equations.

An interesting issue when solving the transient Navier-Stokes problem is the decoupling of the velocity and pressure. There are several techniques to deal with this approach that consists on solving separately the velocity degrees of freedom and the pressure by approximating the coupling terms. One of them, for example, is the widely used Fractional step method, [67]. There also are some works done in this direction for multistep methods, see for instance [120]. Nikitin in [142] suggested a Runge-Kutta method which decouples pressure and velocity by using a pressure splitting technique on the last step of the scheme.

Also related with the time integration procedures, there appears the idea of using an adaptive time stepping technique. Adaptive time stepping is an interesting tool that allows to control the accuracy of the time integration, but also improves the simulation efficiency. In this direction, the Runge-Kutta method provides an excellent background to implement this computational tool since we can use the different stages to compute an error estimate at each time step. John et al. in [115] studied some time stepping control methods applied to different types of integration schemes, including the DIRK scheme. A more specific step-control analysis for explicit and implicit Runge-Kutta methods is done in [92], where a predictive controller is also proposed. More adaptive time step techniques are proposed in [83] for convection-diffusion equation and in [118] for the Navier-Stokes equations. Nikitin also includes a section dedicated to the adaptive time step in [142].

2.5.3 Stability and order of convergence

Some definitions of stability and order of convergence need to be introduced since it will be used to characterize some of the methods used in forthcoming chapters. Deeper explanations on stability and order conditions can be found in [91, 92].

When analyzing stability of ODEs, the solution of the Dahlquist's equation $u' = \lambda u$ is studied. After applying the implicit Euler method it reads

$$u_1 = u_0 + h\lambda u_1, \tag{2.66}$$

being h the step size. The solution to (2.66) is

$$u_1 = R(h\lambda)u_0, \tag{2.67}$$

where $R(z)$ is called the stability function for $z \in \mathbb{C}$. For θ-methods, the stability function reads $R(z) = \frac{1+z(1-\theta)}{1-z\theta}$.

Under these definitions we say that a method is *A-stable* if

$$z \in \mathbb{C}^-, \quad \text{with} \quad \mathbb{C}^- := \{z \in \mathbb{C} |\ Re\ z \leq 0\}. \tag{2.68}$$

It can be shown that a θ-method is A-stable for all $\theta \geq \dfrac{1}{2}$, see [127]. Furthermore, we say that a method is *L-stable* when it is A-stable and additionally it satisfies

$$\lim_{z\to\infty} R(z) = 0. \tag{2.69}$$

Let us consider a Runge-Kutta method given by equations (2.63)-(2.64). We say that a Runge-Kutta method has *order p* if

$$\|u(t_{n+1}) - u_{n+1}\| \leq Ch^{p+1}. \tag{2.70}$$

2.6 Conclusions

In this chapter we have settled, with a certain degree of detail, the formulations that will be used during the forthcoming chapters. If the reader is interested in going further on the description of the concepts stated during the chapter, more information can be found in the references that have been provided along this introductory part.

We have defined the governing equations of the flow motion, which lead to the well-known Navier-Stokes equations for incompressible flows. Some insights are also given in the first section of this chapter related to the influence of the pressure field and the incompressibility constraint in the fluid motion.

Some mathematical notation have been introduced in the second section of this chapter. This notation is a basic tool for the definition of the variational formulation of the Navier-Stokes equations, which has been stated in the third section. We have split the description of the variational formulation in three different subsections, starting with the definition of such formulation at the continuous level. After that, the definition of the FE method is stated, before the development of the semi-discrete problem.

Once we have settled the variational formulation of the Navier-Stokes equations, the Variational Multiscale method has been introduced in the fourth section. A deep description of this method is given focusing on the three different ingredients that define a particular version of such method. More precisely, we have studied the effects of the selecting the subscales to be quasi-static or dynamic, linear or nonlinear and orthogonal or non-orthogonal with respect to the FE space.

Finally, the time integration of the semi-discrete problem has been analyzed in the fifth section of this chapter. We have defined two different groups of time integration schemes. One based on the called *θ-method* and the other based on the *Runge-Kutta* schemes. Basic concepts of stability and order of convergence have been given in this last section.

Chapter 3

The turbulence phenomena

One of the main topics of this thesis is the turbulence phenomena that appear in incompressible fluid flows. In order to put the reader in context, it is mandatory to describe the main properties and definitions of such kind of flows, albeit rather briefly.

Some elementary definitions are established in the first section of this chapter. Thereafter, a short introduction to the vorticity is given, together with some characteristic properties of this field. The definition of the Reynolds stresses is stated subsequently, giving some insights of what does *turbulence modelling* mean. In forthcoming chapters, some turbulent quantities will be used to characterize the fluid flow, so we dedicate some lines in this chapter on the explanation of such quantities. Finally, particular characteristics of isotropic turbulence and wall-bounded turbulent flows are detailed. Here we introduce the turbulence tests performed during the thesis.

For a deeper explanation of turbulence phenomena we refer to [150, 62, 148, 176, 129].

3.1 Introduction to turbulence

3.1.1 Elementary concepts

In the coming chapters some elementary concepts will be used. Here we give a brief definition of the most important ones:

- **Categories of fluid flow:**
  Fluid mechanics and fluid flows are often divided into different regimes. In particular, one can make three different sub-divisions. The first division distinguishes between fluids that may be treated as inviscid or fluids in where the finite viscosity must be taken into account. The second sub-division distinguishes between laminar (organized) and turbulent (chaotic) flow. The final sub-division is between irrotational (or potential) flow and rotational flow.

  In this thesis we will mainly focus on the second sub-division, distinguishing between laminar and turbulent flows. Except when stated, the fluid will be considered to be viscous and rotational.

- **Newton's law of viscosity**
  In most fluids the shear stress is quantified using an empirical law known as Newton's law of viscosity. This law says that a shear stress, τ, is required to cause relative sliding of the fluid layers. Moreover it states that τ is directly proportional to the angular distortion rate $\frac{d\gamma}{dt}$. It can be shown that in a one-dimensional flow (being x the flow direction), $u_x(y)$, $\frac{d\gamma}{dt} = \frac{\partial u}{\partial y}$. Then the shear stress can be written as

  $$\tau = \rho\nu \frac{\partial u_x}{\partial y}, \qquad \nu = \frac{\mu}{\rho}.$$

  Where ν is called the kinematic viscosity.

  For two-dimensional flow, Newton's law of viscosity becomes

  $$\tau_{xy} = \rho\nu \left(\frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial x} \right).$$

- **Reynolds number**
  We define the inertial force of a fluid as the force due to the fluid motion and the viscous force as the force produced by the friction. The viscous forces per unit volume have a size: $f_\nu \sim \frac{\rho\nu|\mathbf{u}|}{l_\perp^2}$, where $l_\perp^2$ is a characteristic length scale normal to the streamlines. The inertial forces per unit volume are of the order of: $f_{in} \sim \frac{\rho|\mathbf{u}|^2}{l}$, where l is a typical geometric length scale.

  We can estimate the ratio between the inertial forces and the viscous forces. This ratio is called *Reynolds number*,

  $$Re = \frac{\frac{\rho|\mathbf{u}|^2}{l}}{\frac{\rho\nu|\mathbf{u}|}{l_\perp^2}} = \frac{ul}{\nu}.$$

  When Re is small, viscous forces outweight inertial forces (laminar regimes), and when Re is large, viscous forces are relatively small compared against the inertial ones (turbulent regime).

- **Boundary layers**
  Let us consider a high Reynolds number flow. Since Re is large, we might be tempted to solve the inviscid equations of motion,

  $$(\mathbf{u} \cdot \nabla)\mathbf{u} = -\nabla \left(\frac{p}{\rho} \right)$$

  subject to the inviscid boundary condition $\mathbf{u} \cdot dS = 0$ on all solid surfaces. This determines the so-called *external problem*. If the fluid satisfies the no-slip condition $\mathbf{u} = 0$ on Γ_D, there must be some region where the velocity adjust to zero. This region is the called *Boundary Layer*. In this region the only mechanical forces available to cause a drop in velocity are viscous shear stresses. Thus the viscous term must be of the same order as the other terms within the boundary layer,

$$\nu \Delta \mathbf{u} \sim (\mathbf{u} \cdot \nabla)\mathbf{u},$$

If the inertial forces are of the same order than the viscous forces in the boundary layer, we can establish the following relation

$$\begin{array}{ccc} f_{in} & \sim & f_{\nu} \\ \dfrac{\rho u^2}{l} \sim \dfrac{\rho \nu u}{\delta^2} & \Rightarrow & \dfrac{\delta}{l} \sim \left(\dfrac{ul}{\nu}\right)^{-1/2} = Re^{-1/2}, \end{array}$$

being δ the size of the boundary layer and l the characteristic domain size. Thus we see that, no matter how small we make ν, there is always some boundary layer where shear stresses are important.

Since Re is large, $\delta \ll l$. When the boundary layer is so thin, the pressure within a boundary layer is virtually the same as the pressure immediately outside the layer.

Boundary layers have another important characteristic, called *separation*, that occurs when the fluid in the boundary layer is ejected into the external flow and a turbulent wake forms. This separation is caused by the pressure forces. When the adverse pressure gradient ($\nabla p > 0$) is big enough, the flow in the boundary layer decelerates and reduces the momentum. Then, the fluid in the boundary layer has less momentum than the corresponding external flow and very quickly it comes to a halt, reverses direction and moves off into the external flow, thus forming a wake.

- **Laminar and Turbulent flow:**
  It is an empirical observation that at low values of Re flows are laminar, while at high values of Re they are turbulent (chaotic).

  A turbulent flow is characterized by the fact that, superimposed on the mean flow pattern, there is random, chaotic motion. The transition from laminar to turbulent flow occurs because, at certain value of Re, instabilities develop in the laminar flow, usually driven by the inertial forces. At low values of Re these potential instabilities are damped out by viscosity, while at high values of Re the damping is inadequate.

3.1.2 Vorticity

The vorticity is a measure of the rotation of individual fluid elements and it is defined by the following expression

$$\boldsymbol{\omega} \cdot d\mathbf{S} = \oint_C \mathbf{u} \cdot d\mathbf{l}.$$

Vorticity cannot be created within the interior of a fluid unless there are body forces present, but it spreads by diffusion and can be intensified by stretching of fluid elements. Boundary layers can be thought as diffusion layers for the vorticity generated on a surface.

Sometimes it is more fruitful to work with the vorticity field instead than the velocity field. First, because the rules governing the evolution of $\boldsymbol{\omega}$ are somewhat simpler than those governing $\mathbf{u}$. The second reason is that many flows are characterized by localized regions of intense rotation. When we are interested in rotation, it is natural to focus on angular momentum rather than linear momentum. Operating with the angular momentum, one can obtain an equation describing the vorticity motion, see [62].

$$I\frac{D\boldsymbol{\omega}}{Dt} = -\boldsymbol{\omega}\frac{DI}{Dt} + 2\nu\mathbf{T} \tag{3.1}$$

where I is the moment of inertia of a small material element that is *instantaneously* spherical and $\nu\mathbf{T}$ denotes the viscous torque acting on the sphere. The equation (3.1) suggest several results:

- $\boldsymbol{\omega}$ evolves independently of p.
- If $\boldsymbol{\omega}$ is initially zero, and the flow is inviscid ($\nu = 0$), then $\boldsymbol{\omega}$ should remain zero in each fluid particle.
- If I decreases (the vortex is stretching) in a inviscid fluid element, then the vorticity of that element should increase.

Alternatively to (3.1), introducing the identity $\nabla(\mathbf{u}^2/2) = (\mathbf{u}\cdot\nabla)\mathbf{u} + \mathbf{u}\times\boldsymbol{\omega}$ into the linear momentum equation and operating, we obtain another expression of the vorticity motion in terms of the velocity field

$$\frac{D\boldsymbol{\omega}}{Dt} = (\boldsymbol{\omega}\cdot\nabla)\mathbf{u} + \nu\Delta\boldsymbol{\omega}. \tag{3.2}$$

From (3.1) and (3.2) we can easily see that the rate of rotation of a fluid blob may increase or decrease due to changes in its moment of inertia, or changes because it is spun up or slowed down by viscous stresses. We also can state from equation (3.2) that vorticity is advected by $\mathbf{u}$ and diffused by viscous stresses.

Looking at the equation (3.2) we can see that in three-dimensional flows the first term on the right hand side is non-zero. Comparing this equation with the angular momentum equation (3.1), we can say that $(\boldsymbol{\omega}\cdot\nabla)\mathbf{u}$ represents intensification of vorticity by the stretching fluid elements, a justification of this suggestion can be fount in [150].

3.1.3 Reynolds stresses and turbulence models

It is an empirical observation that if Re is large enough a flow invariably becomes unstable and then turbulent. Suppose we have a turbulent flow in which $\mathbf{u}$ and p

consist of a time-averaged component, $\overline{\mathbf{u}}$ and $\overline{p}$, plus a fluctuating part, $\mathbf{u}'$ and p':

$$\mathbf{u} = \overline{\mathbf{u}} + \mathbf{u}', \qquad p = \overline{p} + p'.$$

Taking the x-component of the time averaged equation of motion we have that

$$\begin{aligned}(\overline{\mathbf{u}} \cdot \nabla)\overline{u}_x = -\frac{\partial}{\partial x}\left(\frac{p}{\rho}\right) + \frac{\partial}{\partial x}\left[2\nu\frac{\partial \overline{u}_x}{\partial x}\right] + \frac{\partial}{\partial y}\left[\nu\left(\frac{\partial \overline{u}_x}{\partial y} + \frac{\partial \overline{u}_y}{\partial x}\right)\right] + \\ + \frac{\partial}{\partial z}\left[\nu\left(\frac{\partial \overline{u}_x}{\partial z} + \frac{\partial \overline{u}_z}{\partial x}\right)\right] + \frac{\partial}{\partial x}[-\overline{u'_x u'_x}] + \frac{\partial}{\partial y}[-\overline{u'_x u'_y}] + \frac{\partial}{\partial z}[-\overline{u'_x u'_z}].\end{aligned} \tag{3.3}$$

We determine the laminar stresses from the Newton's law of viscosity, taking into account only the time-averaged components

$$\begin{aligned}\sigma_x &= 2\rho\nu\frac{\partial \overline{u}_x}{\partial x}, \\ \tau_{xy} &= \rho\nu\left[\frac{\partial \overline{u}_x}{\partial y} + \frac{\partial \overline{u}_y}{\partial x}\right], \\ \tau_{xz} &= \rho\nu\left[\frac{\partial \overline{u}_x}{\partial z} + \frac{\partial \overline{u}_z}{\partial z}\right].\end{aligned}$$

From (3.3) we see that the turbulent flow have additional stresses that are not considered on the laminar definition. These stresses are called *Reynolds stresses* and are determined by

$$\begin{aligned}\sigma_x^R &= -\rho\overline{u'_x u'_x}, \\ \tau_{xy}^R &= -\rho\overline{u'_x u'_y}, \\ \tau_{xz}^R &= -\rho\overline{u'_x u'_z}.\end{aligned}$$

Then, we can rewrite (3.3) in a more compact way:

$$\frac{\partial \bar{u}_x}{\partial t} + (\bar{\mathbf{u}} \cdot \nabla)\bar{u}_x = -\frac{\partial \bar{p}}{\partial x} + \nu\nabla^2\bar{u}_x + (\nabla \cdot \mathbf{T})_x, \tag{3.4}$$

being

$$\mathbf{T} = \begin{bmatrix} \sigma_x^R & \tau_{xy}^R & \tau_{xz}^R \\ \tau_{yx}^R & \sigma_y^R & \tau_{yz}^R \\ \tau_{zx}^R & \tau_{zy}^R & \sigma_z^R \end{bmatrix}$$

the Reynolds stress tensor. If we wish to make predictions from equation (3.4) we need to be able to relate the Reynolds stresses, $-\rho\overline{u'_x u'_i}$, to some quantity which we know about, such as mean velocity gradients of the type $\frac{\partial \overline{u}_x}{\partial y}$. This is the purpose of *turbulence modelling*. In effect, a turbulence model provides a means of estimating Reynolds stresses.

The Closure Problem of Turbulence

In a turbulent fluid, $\mathbf{u}$ is a chaotic field and vary from one realization of a flow to the next. But the statistical properties of $\mathbf{u}$ seem to be well behaved and perfectly reproducible. It turns out to be possible to manipulate the Navier-Stokes equations into a hierarchy of statistical equations of the form,

$$\frac{\partial}{\partial t}[\text{certain statistical properties of } \mathbf{u}] = f(\text{other statistical properties of } \mathbf{u}).$$

It can be shown that this system of equations is not closed, in the sense that, no matter how many manipulations we perform, there are always more statistical unknowns than equations relating them. This is known as the *closure problem of turbulence*, and it arises because of the non-linearity of the Navier-Stokes equations.

Reynolds stresses decomposition

Leonard (1974) introduced a decomposition of τ_{ij}^R into three component stresses.

$$\tau_{ij}^R = L_{ij} + C_{ij} + R_{ij},$$

where $L_{ij} = \overline{\overline{u}_i\overline{u}_j} - \overline{u}_i\overline{u}_j$ are the Leonard stresses. The cross stresses are $C_{ij} = \overline{\overline{u}_i u_j'} + \overline{u_i'\overline{u}_j}$ and finally, $R_{ij} = \overline{u_i'u_j'}$ are the subgrid scale (SGS) Reynolds stresses. Decomposing Reynolds stresses into Leonard, cross and SGS Reynolds stresses we can reformulate (2.7) as

$$\frac{\partial \overline{\mathbf{u}}}{\partial t} + (\overline{\mathbf{u}} \cdot \nabla)\overline{\mathbf{u}} = -\frac{\partial \overline{p}}{\partial x} + \nu\nabla^2\overline{\mathbf{u}} + \nabla \cdot \mathbf{L} + \nabla \cdot \mathbf{C} + \nabla \cdot \mathbf{R}.$$

3.1.4 Definition of some turbulent quantities

Turbulent flows are considered to be chaotic since, superimposed on the mean flow pattern, there appears a fluctuating and random, both in space and time, motion. In general, the simulation of this kind of flows requires a large amount of computational resources. This occurs because the size of spatial and temporal scales needed to accurately reproduce the flow are extremely fine. In order to simplify the simulation, instead of resolving all fluid flow scales, we can simulate the mean flow pattern, *i.e.* the largest scales, which in most engineering problems is enough, and consider how the unresolved scales interact with the largest ones taking into account their gross statistical properties. Then, if we want to ensure that the simulation reflects the correct mean flow and the effect of the small scales on the largest ones is accurately predicted we need some verification tools. These tools are some turbulent quantities that we can evaluate in our simulation and we can use to compare with analytical or more accurate results. In general, these quantities are obtained through an averaging procedure of the fluctuating fluid motion. We define the averaging procedure for a generic quantity q as

$$\langle q\rangle = \frac{1}{\Pi_{i=1}^{n_d} N_i} \sum_{j_1,\ldots,j_{n_d}} q_{j_1,\ldots,j_{n_d}},$$

that is averaged over n_d dimensions, each dimension i having N_i evaluation points.

In this section we will briefly define the turbulent quantities that are going to be used to characterize the turbulent flow simulations and that will allow us to compare the results against the references. We can find a more complete definition of the turbulent quantities that are going to be stated in this section in [148, 150, 62].

- **Velocity correlation function**
  The velocity correlation function is defined by

  $$Q_{ij} = \left\langle u_i'(\mathbf{x}) u_j'(\mathbf{x}+\mathbf{r})\right\rangle . \tag{3.5}$$

  In general Q_{ij} also depends on time. Q_{ij} tells us about the degree to which, and the manner in which, the velocity components at different points are correlated to each other. However, the velocity correlation function does not, in of itself, tell us how the kinetic energy is distributed across the different eddy sizes. Rather, we must introduce two additional quantities like the second-order structure function and the energy spectrum.

- **Second-order structure function**
  The second-order longitudinal structure function is defined in terms of the longitudinal velocity increment, $\Delta u(r) = u_x(\mathbf{x}+r\hat{\mathbf{e}}_x) - u_x(\mathbf{x})$, as follows:

  $$\left\langle [\Delta u(r)]^2\right\rangle = \left\langle [u_x(\mathbf{x}+r\hat{\mathbf{e}}_x) - u_x(\mathbf{x})]^2\right\rangle . \tag{3.6}$$

  Only eddies of size $\sim r$, or less, can make significant contribution to Δu, and so $\left\langle [\Delta u]^2\right\rangle$ is often taken as an indication of the energy per unit mass contained in the eddies of size r or less.

- **Energy spectrum**
  To determine the distribution of the kinetic energy among the eddies of different size we look at the energy spectrum function $E(k)$. Working with wave number, k, we define the energy spectrum via the transform pair

  $$E(k) = \frac{2}{\pi}\int_0^\infty R(r) kr \sin(kr) dr, \tag{3.7}$$

  $$R(r) = \int_0^\infty E(k) \frac{\sin(kr)}{kr} dk. \tag{3.8}$$

Where $R(r) = \frac{1}{2}\langle \mathbf{u}(\mathbf{x}) \cdot \mathbf{u}(\mathbf{x}+\mathbf{r})\rangle = u^2(g + f/2)$ is called the autocovariance function. The functions f and g are called the longitudinal and lateral velocity correlation coefficients and satisfy the relation $2rg = (r^2 f)'$.

In Appendix A, some tips for the implementation of the energy spectrum calculation are given.

- **Total kinetic energy**
  Taking the limit $r \to 0$, the equation (3.8) leads to

$$R(0) = \frac{1}{2}\left\langle \mathbf{u}^2 \right\rangle = \int_0^\infty E(k)dk, \tag{3.9}$$

which is nothing else than the total kinetic energy, $K := \frac{1}{2}\left\langle \mathbf{u}^2 \right\rangle$. Equation (3.9) tells us that $E(k)dk$ is the contribution to K from all eddies with wave numbers in the range $k \to k + dk$, where $k \sim \pi/r$. In fact, it is not true at all, but it works if we are on the range $\eta^{-1} < k < l^{-1}$, being η and l the Kolmogorov and integral length scales, respectively, defined in following points.

- **Enstrophy**
  The energy spectrum $E(k)$ has one further property

$$\left\langle \boldsymbol{\omega}^2 \right\rangle = \int_0^\infty k^2 E(k)dk. \tag{3.10}$$

If we define the enstrophy as $Z := \frac{1}{2}\left\langle \boldsymbol{\omega}^2 \right\rangle$, from (3.10), the quantity $k^2E(k)$ is interpreted as the contribution to the enstrophy from the range of wave numbers $k \to k + dk$.

- **Skewness factor**
  The third-order longitudinal structure function is defined, like the second-order longitudinal structure function, as

$$\left\langle [\Delta u(r)]^3 \right\rangle = \left\langle [u_x(\mathbf{x} + r\hat{\mathbf{e}}_x) - u_x(\mathbf{x})]^3 \right\rangle.$$

For further discussions, it is more useful to work with the skewness factor. This factor is a normalized version of the third-order structure function

$$S(r) = \frac{\left\langle [\Delta u(r)]^3 \right\rangle}{\left\langle [\Delta u(r)]^2 \right\rangle^{3/2}}. \tag{3.11}$$

The skewness factor for a decaying turbulent flow depends slightly on the Reynolds number, but usually has a value of around $-0,4 \pm 0,1$ for $r \to 0$ for Re up to 10^6, and decays slowly with r.

- **Root mean square of the velocity field**
  For a three-dimensional flow, the root mean square (RMS) of the velocity field is

given by

$$u = \left(\frac{2}{3} \int_0^\infty E(k) dk \right)^{1/2} .$$

- **Kolmogorov scale**
  The viscous dissipation acts most efficiently at small scales. Then, for wavenumbers greater than certain wavenumber k_η, the viscous dissipation will become important, and the energy spectrum $E(k)$ will decay more rapidly than the inertial range. It can be shown that

  $$k_\eta \sim \left(\frac{\varepsilon}{\nu^3} \right)^{1/4} ,$$

  with ε the total energy dissipation rate. The Kolmogorov scale is the inverse of this wave number $\eta := 1/k_\eta$, then

  $$\eta \sim \left(\nu^3 / \varepsilon \right)^{1/4} .$$

- **Integral length scale**
  The integral length scale is the scale of the energy-containing eddies. From the Kolmogorov spectrum (3.17) we can determine the wavenumber that defines this scale, k_ℓ, in terms of the energy dissipation rate

  $$k_\ell \sim \frac{\varepsilon}{u^3} .$$

  Then, the integral length scale, ℓ, will be $\ell \sim u^3/\varepsilon$. Taking the ratio between the integral and Kolmogorov scales we have that

  $$\frac{\ell}{\eta} \sim \frac{u^3}{\varepsilon^{3/4} \nu^{3/4}} \sim \left(\frac{u\ell}{\nu} \right)^{3/4} \sim Re_\ell^{3/4} ,$$

  where Re_l is the integral Reynolds number. Thus, the inertial range includes a set of scales growing with $Re_\ell^{3/4}$. If we want to accurately simulate the inertial range, the number grid points on an uniform 3D mesh should be $N \sim \left((Re_\ell)^{3/4} \right)^3 = (Re_\ell)^{9/4}$.

- **Taylor microscale**
  Apart from the Kolmogorov and integral length scale, there is another length scale that can characterize a turbulent flow. This is the Taylor microscale and provides a convenient estimate for the fluctuating strain rate field. It is a standard turbulent length that identifies the inertial subrange. The Taylor microscale is defined as

  $$\lambda^2 = \frac{15 u^2}{\langle \boldsymbol{\omega}^2 \rangle} = \frac{15 \nu u^2}{\varepsilon} .$$

- **Taylor-microscale Reynolds number**
  The Taylor-microscale Reynolds number is associated to the characteristic Taylor-microscale length and it is defined as

  $$Re_\lambda = \frac{u\lambda}{\nu} = \sqrt{15}\frac{u^2}{(\nu\varepsilon)^{1/2}} = \frac{1}{\nu}\sqrt{\frac{20}{3}}\frac{\int_0^\infty E(k)dk}{\left(\int_0^\infty k^2 E(k)dk\right)^{1/2}}.$$

- **Energy dissipation rate**
  For a closed domain with stationary boundaries, the total rate of energy dissipation per unit of mass is given by

  $$\varepsilon = 2\nu\frac{1}{2}\left\langle \boldsymbol{\omega}^2 \right\rangle = \frac{15\nu u^2}{\lambda^2}.$$

  See Section 3.2.1, where we derive this result.

3.2 Isotropic turbulence

We understand by isotropic turbulent flow the one in which there is no mean flow and the rotation is negligible. In this situation we do not have any phenomena that introduce anisotropy into the flow. On the other hand, we say that a flow is homogeneous if the spatial gradients of any averaged quantity are negligible. This condition means that the turbulent flow statistics do not depend on space.

3.2.1 Energy dissipation

By definition of homogeneity and isotropy, the velocity mean on the spatial domain is zero. Then, we have to characterize the 3D isotropic turbulent flows relying on the second order statistics of the velocity field, which is nothing else than the kinetic energy, $K = \frac{1}{2}\left\langle \mathbf{u}^2 \right\rangle$.

Multiplying the momentum equation (2.7) by the velocity, assuming that $\mathbf{f} = 0$, and operating we have that

$$\frac{\partial\left(\mathbf{u}^2/2\right)}{\partial t} = -\nabla\cdot\left[\left(\mathbf{u}^2/2\right)\mathbf{u}\right] - \nabla\cdot\left[(p)\,\mathbf{u}\right] + \nabla\cdot\left[(\nu(\boldsymbol{\omega}\times\mathbf{u}))\right] - \nu\boldsymbol{\omega}^2. \tag{3.12}$$

Noting that $\frac{\mathbf{u}^2}{2}$ is nothing else than the kinetic energy, equation (3.12) is giving information about how the kinetic energy is dissipated. If we integrate (3.12) over an arbitrary and fixed volume V, and using the divergence theorem, we can determine through which mechanisms the energy is dissipating:

$$\begin{aligned}
&\frac{d}{dt}\int_V \left(\mathbf{u}^2/2\right) dV = \qquad (3.13)\\
&-\text{[rate at which kinetic energy is convected across the boundary]}\\
&+\text{[rate at which the pressure forces do work on the boundary]}\\
&+\text{[rate at which the viscous forces do work on the boundary]}\\
&-\int_V \nu\boldsymbol{\omega}^2 dV.
\end{aligned}$$

Averaging (3.13) and assuming that for homogeneous and isotropic turbulence the average flux through the boundaries is negligible, we obtain a relation between the averaged total kinetic energy K with the enstrophy Z

$$\frac{dK}{dt} = -2\nu Z, \qquad (3.14)$$

which states that the rate of change of turbulent kinetic energy per unit mass is balanced by the rate of dissipation of mechanical energy per unit mass, $\varepsilon := 2\nu Z = \nu\langle\boldsymbol{\omega}^2\rangle$.

3.2.2 Kinetic energy in spectral space

One may be interested on how the energy is transferred between different scales, which is described by the energy spectrum evolution.

In order to obtain an evolutionary equation for the energy spectrum $E(k,t)$, we will first transform the Navier-Stokes equations (2.7)-(2.8) into the Fourier space. Let us consider the integral Fourier representation of the velocity field

$$\hat{\mathbf{u}}(\mathbf{k},t) = \left(\frac{1}{2\pi}\right)^3 \int \exp(-i\mathbf{k}\cdot\mathbf{x})\mathbf{u}(\mathbf{x},t)d\mathbf{x}.$$

Under this notation, the incompressibility constrain (2.8) would read

$$\mathbf{k}\cdot\hat{\mathbf{u}}(\mathbf{k},t) = 0,$$

meaning that the Fourier velocity field $\hat{\mathbf{u}}$ is perpendicular to $\mathbf{k}$. Then, the time derivative $\partial\hat{\mathbf{u}}/\partial t$ and the viscosity term $\nu k^2\hat{\mathbf{u}}$ also belong to the plane perpendicular to $\mathbf{k}$, being $k^2 = \sum_{i=1}^d k_i^2$. The pressure gradient is given by $i\hat{p}\mathbf{k}$, which is parallel to the wavevector $\mathbf{k}$. Thus, the Fourier transform of $\mathbf{u}\cdot\nabla\mathbf{u} + \nabla p$ is nothing else than the projection into the plane perpendicular to $\mathbf{k}$ of $\mathbf{u}\cdot\nabla\mathbf{u}$. The final expression of the Navier-Stokes equations in the Fourier space is given by

$$\left(\frac{\partial}{\partial t} + \nu k^2\right)\hat{u}_i(\mathbf{k},t) = -i\int_{\mathbf{p}+\mathbf{q}=\mathbf{k}} q_j\left(\delta_{i,m} - \frac{k_i p_m}{k^2}\right)\hat{u}_j(\mathbf{p},t)\hat{u}_m(\mathbf{q},t)d\mathbf{p}d\mathbf{q}. \qquad (3.15)$$

Multiplying (3.15) by the complex conjugate $\hat{\mathbf{u}}_i^*(\mathbf{k},t)$ integrating over $\mathbf{k}$, and operating we have that for isotropic turbulence the energy spectrum follows

$$\frac{\partial}{\partial t}E(k,t) = T(k,t) - 2\nu k^2 E(k,t). \tag{3.16}$$

with $T(k,t)$ the term that comprises all triad interaction terms. Integrating over all k, and using the relation between the kinetic energy and the total energy dissipation rate given by (3.14) we have that

$$\int_0^\infty T(k,t)dk = 0,$$

which means that the nonlinear interactions transfer energy between different wave numbers, but do not change the total energy.

Kolmogorov spectrum

In order to answer some key questions about turbulent flows, like *what is the size of the smallest eddies that are responsible for dissipating the energy?*, Kolmogorov stated three hypotheses in [124]. Let us remind these hypotheses:

- **Local isotropy.** At sufficiently high Reynolds number, the small-scale turbulent motions are statistically isotropic.
- **First similarity hypothesis.** For the locally isotropic turbulence the statistics of the small-scale motions are uniquely determined by the quantities ν and ε.
- **Second similarity hypothesis.** In every turbulent flow at sufficiently high Reynolds number, the statistics of the motions of scale k in the range $k_\eta \geq k \geq k_l$ have a universal form that is uniquely determined by ε, independent of ν.

It can be seen that the contribution to the dissipation rate from eddies in the range (k_a, k_b) is

$$\varepsilon_{(k_a,k_b)} = \int_{k_a}^{k_b} 2\nu k^2 E(k)dk.$$

The energy spectrum $E(k)$ can be non-dimensionalized using ε and k, leading to the following relation

$$E(k) = \varepsilon^{2/3}k^{-5/3}\Phi(k\eta),$$

where $\Phi(k\eta)$ is the compensated Kolmogorov spectrum function. According to the second similarity hypothesis, at the inertial subrange ($k_\eta \geq k \geq k_l$), the function Φ becomes independent of k and η, tending to a constant C. Then, in the inertial subrange, the energy spectrum function is

$$E(k) \sim C\varepsilon^{2/3}k^{-5/3}. \tag{3.17}$$

3.2.3 Benchmark tests

Decay of homogeneous isotropic turbulence

The Decay of Homogeneous Isotropic Turbulence (DHIT) test is one of the most used benchmarks to test the simulation accuracy of isotropic turbulent flows. Here we want to analyze the statistics of an homogeneous isotropic turbulent flow in a 3D box of size $\Omega = \{(0, 2\pi) \times (0, 2\pi) \times (0, 2\pi)\}$. The flow is initialized with an initial condition that has a predetermined energy spectrum, considering all the boundary conditions to be periodic.

In this work, we have taken the energy spectrum expression from Mansour & Wray [133].

$$E(k, 0) = \frac{q^2}{2A} k_0^{-(\sigma+1)} k^4 \exp\left(-\frac{\sigma}{2}\left(\frac{k}{k_0}\right)^2\right), \tag{3.18}$$

with k_0 the wavenumber at which $E(k, 0)$ is maximum, $\frac{q^2}{2}$ the total kinetic energy and $A = \int_0^\infty k^\sigma \exp(-\sigma k^2/2)$. In this thesis we choose $q^2 = 3$ and $\sigma = 4$.

With this setting, it can be shown that $A = \frac{3}{32}\sqrt{\frac{\pi}{2}}$. So, the initial energy spectra will be

$$E(k, 0) = 16\sqrt{\frac{2}{\pi}} k_0^{-5} k^4 \exp\left(-2\left(\frac{k}{k_0}\right)^2\right). \tag{3.19}$$

Going further, from the definitions of the total kinetic energy ,K, and the enstrophy, Z, given in (3.9) and (3.10), respectively, we can evaluate the initial value of these properties. Hence, we have

$$\begin{aligned} K &= \int_0^\infty E(k, 0) dk \qquad (3.20)\\ &= 16\sqrt{\frac{2}{\pi}} k_0^{-5}\left(-\frac{k_0^2}{4}\right)\int_0^\infty k^3\left(-\frac{4k}{k_0^2}\right)\exp\left(-2\left(\frac{k}{k_0}\right)^2\right)dk\\ &= 12\sqrt{\frac{2}{\pi}} k_0^{-3}\left(-\frac{k_0^2}{4}\right)\int_0^\infty k\left(-\frac{4k}{k_0^2}\right)\exp\left(-2\left(\frac{k}{k_0}\right)^2\right)dk\\ &= 3\sqrt{\frac{2}{\pi}} k_0^{-1}\left(\frac{k_0}{\sqrt{2}}\right)\int_0^\infty \left(\frac{\sqrt{2}}{k_0}\right)\exp\left(-2\left(\frac{k}{k_0}\right)^2\right)dk\\ &= 3\sqrt{\frac{2}{\pi}} k_0^{-1}\left(\frac{k_0}{\sqrt{2}}\right)\frac{\sqrt{2}}{2}\\ &= \frac{3}{2}, \end{aligned}$$

and

$$\begin{aligned} Z &= \int_0^\infty k^2 E(k,0)dk \qquad (3.21)\\ &= 16\sqrt{\frac{2}{\pi}}k_0^{-5}\left(-\frac{k_0^2}{4}\right)\int_0^\infty k^5\left(-\frac{4k}{k_0^2}\right)\exp\left(-2\left(\frac{k}{k_0}\right)^2\right)dk\\ &= 20\sqrt{\frac{2}{\pi}}k_0^{-3}\int_0^\infty k^4\exp\left(-2\left(\frac{k}{k_0}\right)^2\right)dk\\ &= \frac{5}{4}k_0^2\int_0^\infty E(k,0)dk\\ &= \frac{15}{8}k_0^2. \end{aligned}$$

Following Rogallo [153], we generate the initial field on the Fourier space such that it satisfies continuity and has the energy spectrum prescribed in (3.19). First, we define two complex parameters:

$$\alpha = \left(\frac{E(k,0)}{2\pi k^2}\right)^{1/2}\exp(i\theta_1)\cos(\phi),$$

$$\beta = \left(\frac{E(k,0)}{2\pi k^2}\right)^{1/2}\exp(i\theta_2)\cos(\phi).$$

Where θ_1, θ_2 and ϕ are random numbers on the interval $(0, 2\pi)$. With α and β we can construct a solenoidal field in the wave number space whose initial spectrum is equal to $E(k, 0)$.

$$\begin{aligned} \hat{u_1} &= (\alpha|k|k_2 + \beta k_1 k_3)/(|k|k_h), \qquad (3.22)\\ \hat{u_2} &= (-\alpha|k|k_2 + \beta k_3 k_2)/(|k|k_h),\\ \hat{u_3} &= (-\beta k_h^2)/(|k|k_h), \end{aligned}$$

where $k_h = \left(k_1^2 + k_2^2\right)^{1/2}$ and $|k| = \left(k_1^2 + k_2^2 + k_3^2\right)^{1/2}$.

To have the field (3.22) in the physical space we apply the inverse Fast Fourier Transform, see Appendix A for a detailed description of the implementation of the Fourier Transform.

In order to check the accuracy of the proposed methods, we will compare the results with those showed in [111]. Then, we set $k_0 = 6$ and the viscosity such that the associated Taylor-microscale Reynolds number is $Re_\lambda = 952$,

$$Re_\lambda = \sqrt{\frac{20}{3}}\frac{\int_0^\infty E(k)dk}{\nu\left(2\int_0^\infty k^2E(k)dk\right)^{1/2}} = 952$$

which results in $\nu = 3.5014006 \cdot 10^{-4}$.

In Figure 3.1 we show the initial analytical energy spectrum and the one calculated from the initial velocity field on the Fourier space (3.22) using a 128^3 linear elements mesh. We observe that the computed energy spectrum is equivalent to the analytical one, a fact that we were expecting since we use the initial field proposed by Rogallo [153], which precisely satisfies this condition.

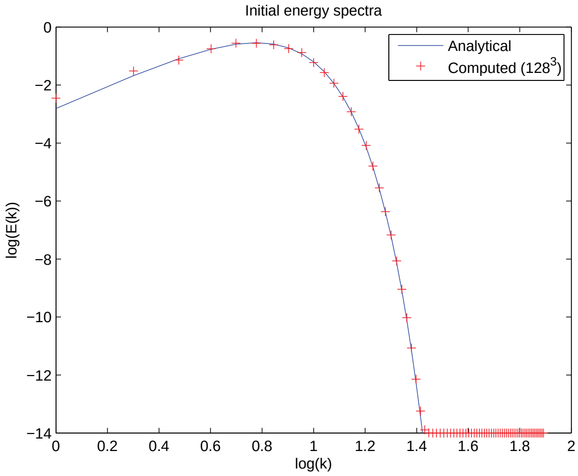


FIGURE 3.1: Analytical and computed initial energy spectrum for the DHIT test.

In Figure 3.2 we depict the characteristic structures of a fully developed homogeneous and isotropic turbulent flow, plotting the vorticity isosurfaces obtained in the DHIT test.

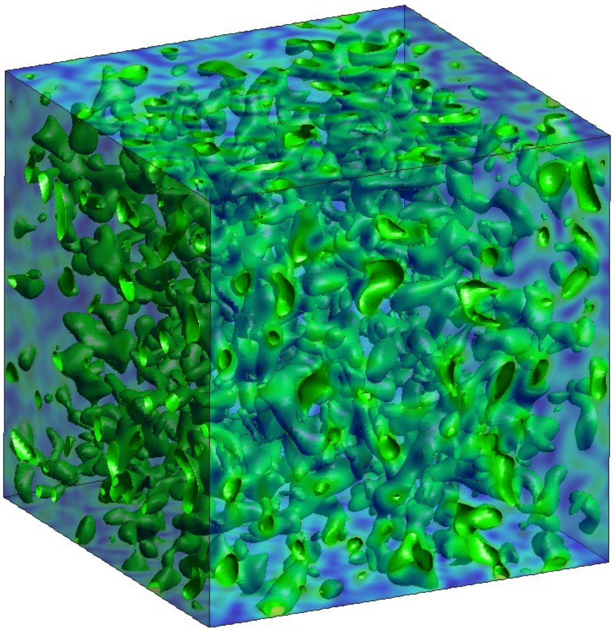


FIGURE 3.2: Vorticity isosurfaces in the DHIT test.

The initial condition has a relevant role on homogeneous isotropic turbulence. In the final periods of decay, as has been exposed by [97], the energy spectrum follows the linear decay law,

$$E(k,t) = E(k,0)\exp(-2\nu k^2 t). \tag{3.23}$$

So, a first conclusion we can point out from (3.23) is that the initial spectra, $E(k,0)$, in the final period of decay, has a direct relation with the decay exponent. The total kinetic energy can be calculated integrating (3.23) over all wave numbers. Then, when $t \to \infty$, the power law decay is determined by the shape of the initial spectrum.

Moreover, [155] showed that if the initial field is generated by random impulsive forces, a spectrum of the form

$$E(k,t) \sim k^2 \exp(-2\nu k^2 t) \tag{3.24}$$

will ensue. Then, the total kinetic energy will decay with the law $K \sim t^{-3/2}$. But, anyway, we will not consider the case of having impulsive forces acting on the flow.

Also in this direction [154] and [46], say that in the long-run, the energy should decay following the $t^{-1.4}$ law.

Taylor-Green Vortex flow

The Taylor-Green Vortex flow (TGV) problem is also a typical and widely used problem on turbulence numerical simulations. First introduced by Taylor and Green (1937) [174], this problem aims to show, in a relatively simple flow, the basic turbulence decay mechanisms like the turbulent energy cascade, the production of small eddies and the enhancement of dissipation by the stretching of vortex lines.

The initial analytical condition for this problem, unlike the DHIT problem, is defined on the physical space. Here we follow [80], where the initial solution is defined by

$$\begin{aligned}
u_x &= u_0 \cos(x)\sin(y)\sin(z), \qquad (3.25)\\
u_y &= -u_0 \sin(x)\cos(y)\sin(z),\\
u_z &= 0,\\
p &= p_0 + \frac{1}{16}\left(\cos(2x) + \cos(2y)\right)\left(\cos(2z) + 2\right).
\end{aligned}$$

With

$$u_0 = \frac{2}{\sqrt{3}}\sin\left(\gamma + \frac{2\pi}{3}\right).$$

We choose $\gamma = 0$, which gives the mean initial velocity $u_0 = 1$. The Reynolds number is defined as the inverse of the kinematic viscosity ν, noting that the length and velocity scales are of the same order. This is done according to [80] and [31], and will allow us to compare our results with those showed on these papers. For simplicity, the pressure constant parameter p_0 is chosen equal to zero.

With these definitions we can compute analytically the initial total kinetic energy of the problem, which is

$$\begin{aligned} K &= \frac{1}{2}\frac{1}{V}\int_0^L\int_0^L\int_0^L \mathbf{u}\cdot\mathbf{u}\,dV \\ &= \frac{1}{16\pi^3}\int_0^{2\pi}\int_0^{2\pi}\int_0^{2\pi}\left(u_x^2+u_y^2+u_z^2\right)dxdydz \\ &= \frac{u_0^2}{16\pi^3}\int_0^{2\pi}\int_0^{2\pi}\int_0^{2\pi}\left(\cos^2(x)\sin^2(y)+\sin^2(x)\cos^2(y)\right)\sin^2(z)dxdydz \\ &= \frac{u_0^2}{8}. \end{aligned} \tag{3.26}$$

We also can calculate the initial enstrophy of the problem, given by the following expression

$$\begin{aligned} Z &= \frac{1}{2V}\int_0^L\int_0^L\int_0^L \nabla\mathbf{u}:\nabla\mathbf{u}\,dV \\ &= \frac{1}{16\pi^3}\int_0^{2\pi}\int_0^{2\pi}\int_0^{2\pi}\left(\sum_{i=1}^{3}\sum_{j=1}^{3}\left(\frac{\partial u_i}{\partial x_j}\right)^2\right)dxdydz \\ &= \frac{u_0^2}{16\pi^3}\int_0^{2\pi}\int_0^{2\pi}\int_0^{2\pi}\left[2\sin^2(z)\left(\sin^2(x)\sin^2(y)+\cos^2(x)\cos^2(y)\right)\right. \\ &+ \left.\cos^2(z)\left(\cos^2(x)\sin^2(y)+\sin^2(x)\cos^2(y)\right)\right]dxdydz \\ &= \frac{3}{8}u_0^2. \end{aligned} \tag{3.27}$$

Another interesting result coming from the definition of the initial condition (3.25) is its representation on the Fourier space. According to Fauconnier et al, [71], the initial velocity field on the Fourier space corresponds to eight Fourier modes located at $\mathbf{k} = (\pm1,\pm1,\pm1)$. It means that the initial flow generates a single vortex scale.

One of the peculiarities about the TGV test is that the initial condition has two-dimensional streamlines, on the $x-y$ plan, but the flow is three dimensional (the initial velocity field also depends on the z direction).

It is also important to be said that the initial flow is highly symmetric. This symmetry makes that, as stressed by Brachet et al. [32], for all times, no fluid crosses any plan such that x, y or $z = n\pi$, being n an integer. Taking into account these flow properties we can define the region $0 \leq x,y,z \leq \pi$ as the *impermeable box*, since there is a confination of the flow inside this domain. The whole region, $0 \leq x,y,z \leq 2\pi$, can be called the *periodic box*. Finally, due to the flow symmetries, there is a region from which we can determine the flow at any point in the space. This region is called *fundamental box* and is generated by the box $0 \leq x,y,z \leq \frac{1}{2}\pi$.

It can be seen that the perpendicular velocity component, $u_\perp$, and the normal derivative of the parallel velocity component, $\frac{\partial u_\parallel}{\partial n}$, for each impermeable box face vanish on this face. That is, given a face Γ_i

$$u_\perp|_{\Gamma_i} = 0, \quad \text{and} \quad \frac{\partial u_\parallel}{\partial n}|_{\Gamma_i} = 0. \tag{3.28}$$

Equation (3.28) has a direct implication to the voritcity field, $\boldsymbol{\omega}$. For a given impermeable box face, the vorticity on this face can be defined in terms of perpendicular and parallel components as

$$\begin{aligned} \omega_{\parallel_1} &= \left(\frac{\partial u_{\parallel_2}}{\partial x_\perp} - \frac{\partial u_\perp}{\partial x_{\parallel_2}} \right) = 0, \\ \omega_{\parallel_2} &= \left(\frac{\partial u_{\parallel_1}}{\partial x_\perp} - \frac{\partial u_\perp}{\partial x_{\parallel_1}} \right) = 0, \\ \omega_\perp &= \left(\frac{\partial u_{\parallel_1}}{\partial x_{\parallel_2}} - \frac{\partial u_{\parallel_2}}{\partial x_{\parallel_1}} \right) \neq 0. \end{aligned} \tag{3.29}$$

Which means that the vorticity is perpendicular to each impermeable box face.

We solve the TGV problem using a Reynolds number $Re = 1600$. The most common Reynolds numbers available in the literature are $\mathrm{Re} = 800$, $\mathrm{Re} = 1600$ and $\mathrm{Re} = 3000$ (see, e.g., [2, 71, 80, 109]).

The TGV test is characterized by its laminar evolution at the initial time steps, when the flow is strongly anisotropic due to the structured large-scale vortexes directly related to the initial condition. If the Reynolds number is large enough, there appears the vortex-stretching process, which introduce the energy cascade effect, transferring energy from large to small-scales. Because of this procedure, the flow becomes unstable and, therefore, turbulent. According to Brachet et al. [32], the flow becomes nearly isotropic for $Re \geq 1000$.

In Figure 3.3 we depict a vorticity isosurface image computed using a 128^3 trilinear hexahedral elements mesh. In this picture we can see the symmetry plans stated before.

3.3 Turbulence in wall-bounded flow

The vast majority of turbulent flows that we can find in the nature are bounded by, at least, one solid surface. Some examples of bounded flows can be the flow through a pipe or a duct, the flow in a channel or a river, the flow around an aircraft or a ship, or even the atmospheric flow that is bounded by the terrain. In a turbulent wall-bounded flow, the turbulence is generated through the viscous forces near the wall and then propagated to the outer layer. In this kind of flows, we are interested on defining the mean velocity profiles as well as the friction laws, which will allow us to determine the forces that the fluid flow exerts on the the solid wall.

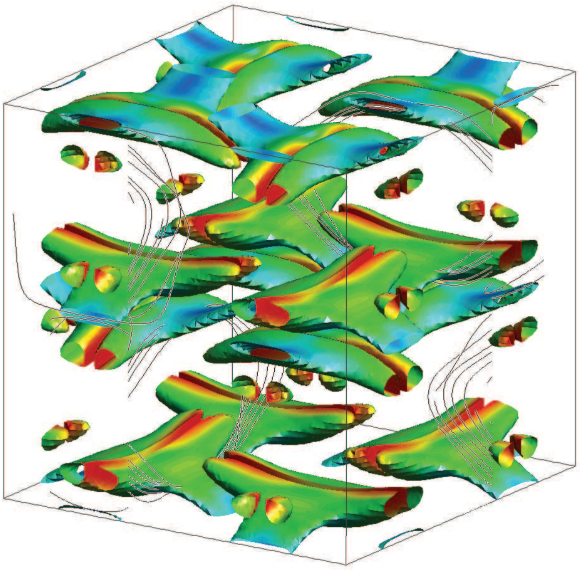


FIGURE 3.3: Vorticity isosurfaces at $t = 4,0$ with streamlines for the TGV test.

3.3.1 Mean flow

Shear stress

Let us assume that within a certain region near the wall the mean velocity is parallel, or almost parallel, to that wall. We consider that we have a wall on the plane $(x - z)$ where the mean flow is predominantly in the x-direction, which we will denote as the stream-wise direction and statistically independent of z. The y-direction will be denoted by the wall-normal direction and the z-direction the span-wise direction. Under this assumptions, $\langle u_z \rangle = 0$ and the $\langle u_x \rangle$ is independent of x, leading to a mean continuity equation

$$\frac{d \langle u_y \rangle}{dy} = 0. \tag{3.30}$$

From the mean momentum equation in the wall-normal direction we can conclude that the pressure is uniform across the flow, satisfying

$$\frac{\partial \langle p \rangle}{\partial x} = \frac{dp_w}{dx},$$

being $p_w(x)$ the pressure value at the wall. Then, from the axial mean momentum equation we have that the shear total shear stress $\tau(y)$ follows

$$\frac{d\tau}{dy} = \frac{dp_w}{dx}, \tag{3.31}$$

with

$$\tau(y) := \nu \frac{d \langle u_x \rangle}{dy} - \langle u_x u_y \rangle , \tag{3.32}$$

see [150] for a more detailed deduction. We will denote the wall shear stress as $\tau_w := \tau(0)$. It is seen that the shear stress is the sum of the viscous stress, $\nu \frac{d\langle u_x \rangle}{dy}$, and the Reynolds stress $-\langle u_x u_y \rangle$. At the wall, the no-slip boundary condition, $\mathbf{u} = 0$, implies that

$$\tau_w = \nu \left. \frac{d\langle u_x \rangle}{dy} \right|_{y=0}. \tag{3.33}$$

Friction quantities

From (3.33) one can observe that the viscosity ν is an important parameter in turbulent wall-bounded flows, particularly in the region near the wall. Hence, the mean velocity profile will depend on the Reynolds number. Moreover, in the region near the wall we can define the appropriate viscous velocity and length scales. We will call friction velocity the velocity scale in the near-wall region, which is defined as

$$u_\tau := \sqrt{\frac{\tau_w}{\rho}}, \tag{3.34}$$

and the viscous length scale as

$$\delta_\nu := \nu \sqrt{\frac{\rho}{\tau_w}} = \frac{\nu}{u_\tau}. \tag{3.35}$$

Note that the Reynolds number based on the viscous scales is equal to the unity, $Re_\nu = \frac{u_\tau \delta_\nu}{\nu} = 1$. An alternative is the so called friction Reynolds number, which is defined as

$$Re_\tau := \frac{u_\tau \delta}{\nu}, \tag{3.36}$$

being δ the boundary layer thickness.

For wall-bounded turbulent flows it is a common practise to use the quantities in terms of wall units. We will define the distance from the wall in wall units as $y^+ := \frac{y}{\delta_\nu} = \frac{u_\tau y}{\nu}$, and the mean stream-wise velocity in wall units as $u^+ := \frac{\langle u_x \rangle}{u_\tau}$.

Velocity profile

It has been shown that in the region near the wall, the mean stream-wise velocity can be defined by the distance to the wall. This is known as the law of the wall, which can be expressed in wall units as $u^+ = f_w(y^+)$. In the inner layer, where $y/\delta \leq 0.1$, the law of the wall has been shown to follow the following expression

$$u^+ = \frac{1}{\kappa} \ln y^+ + B, \tag{3.37}$$

with $\kappa = 0.41$ and $B = 5.2$.

3.3.2 Benchmark tests

Turbulent Channel Flow

According to [150], most turbulent flows are bounded by one or more solid surfaces, giving wall-bounded flows. There are several turbulent wall-bounded tests, e.g. fully developed channel flow, fully developed pipe flow and the flat-plate boundary layer. We will test just one of them, in particular the fully developed Turbulent Channel Flow (TCF), which is a widely used wall-bounded turbulent benchmark.

An evidence of the popularity of this test is the large amount of experimental and computational works done about it. This test consists of a flux that flows between two parallel walls driven by an imposed pressure gradient, where the mean velocity vector is parallel to the wall. The most important issues to consider when characterizing the flow are the mean velocity profiles, velocity fluctuations and the shear stress carried by the fluid on the wall. It has to be noticed that we will concern only on the fully developed flow, which means that the turbulent regime has been reached at the time when we start our flow characterization.

In this test, like in most of turbulent flow tests, the Reynolds number has an important role when defining the test parameters. In particular, in the turbulent channel flow test, the most characteristic parameter is the Reynolds number based on the wall shear velocity, Re_τ. In the literature, there appear mostly three different values of this parameters: $Re_\tau = 590$, $Re_\tau = 395$ and $Re_\tau = 180$, see [23, 40, 78, 82, 87, 105, 113, 121, 136, 140]. We will restrict our work to $Re_\tau = 180$ and $Re_\tau = 395$ cases.

We consider a computational domain defined by a box of length $(L_x \times L_y \times L_z)$, which depends on the Reynolds number. For $\mathrm{Re}_\tau = 180$ the domain size is $(4\pi\delta \times 2\delta \times 4/3\pi\delta)$ while for $\mathrm{Re}_\tau = 395$ the size of the channel is $(2\pi\delta \times 2\delta \times 2/3\pi\delta)$. The x-direction is the flow direction, also called stream-wise direction, the y-direction is the wall-normal direction, and the z-direction is the span-wise direction. In Fig. 3.4 we show schematically the geometry of the computational domain for this test.

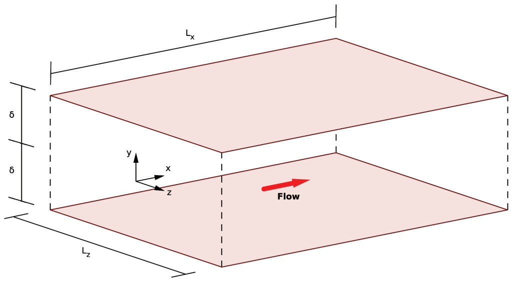


FIGURE 3.4: Channel computational domain.

In the wall-normal direction boundaries ($y = -\delta$ and $y = \delta$) we impose a non-slip condition. The stream-wise and span-wise directions are assumed to be homogeneous, so we use periodic boundary conditions in these directions. According to Kim *et al.* [121], the use of periodic boundary conditions in the homogeneous directions can be justified when the computational box is such that the largest eddies in the flow fit in the computational domain. This is the case considered here.

We solve the problem using different mesh discretizations, all of them uniform in the homogeneous directions and refined near the walls in the wall-normal direction. The refinement in the wall-normal direction follows a hyperbolic function, also used in [40, 78, 82, 87, 136], defined as

$$y_i = \frac{\tanh\left(\gamma\left(\frac{2i}{np_y} - 1\right)\right)}{\tanh(\gamma)},$$

where $i = 1, ..., np_y$ with np_y the total amount of nodes in the wall-normal direction. Here, γ is chosen to be equal to 2.75 for both $\mathrm{Re}_\tau = 180$ and $\mathrm{Re}_\tau = 395$. We refer the reader to [8] for a complete study of the influence of the discretization in the results of the TCF.

3.4 Conclusions

With the aim of having a self-contained thesis, in this chapter we have described the basic concepts of turbulent flows. Although there are not any innovative concepts in this chapter, all the definitions given in it are basic for the development and interpretation of further investigations. In any case, all the information provided in this chapter have been properly referenced.

We have started the chapter with some basic definitions that will be used during al this thesis. We also have analyzed the vorticity field and its behaviour in a turbulent flow. The definition of the Reynolds stresses and some turbulent quantities have also been described in the first section of this chapter.

Some particular insights about isotropic turbulence have been given in the second section of the chapter. The energy dissipation of this kind of turbulent flows have been analyzed in both real and spectral spaces. We have stated the relation between the energy dissipation and the enstrophy of the flow, and we have given a formula for the energy spectra in terms of the wave length and energy dissipation rate. Finally, two different isotropic turbulent benchmarks have been defined in the second section, the decaying of homogeneous isotropic turbulence and the Taylor-Green vortex flow.

In the third section of this chapter, the particular characteristics of wall-bounded turbulent flows have been defined. First, the description of the mean flow of such problems is studied. Then, the definition of the turbulent channel flow test is given as a benchmark for this kind of turbulent flows.

Chapter 4

Residual-based VMS methods

4.1 Introduction

LES techniques for the numerical simulation of turbulent flows [157] are based on a scale separation that permits to reduce the computational cost with respect to direct numerical simulation (DNS). Such scale separation is traditionally achieved by filtering the original Navier-Stokes equations, which leads to an extra forcing term defined by a physical (functional or structural) model. This widely used approach is usually referred to as explicit LES [157].

By contrast, implicit LES techniques (ILES) rely on purely numerical artifacts without any modification of the continuous problem. This approach was seldom followed, the MILES (Monotone Integrated LES) approach [28, 76, 84] being the main exception, until the VMS method was introduced [100, 101] and subsequently proposed as an ILES method (see below). ILES techniques are usually considered to be based on the addition of purely dissipative numerical terms, see [157, Section 5.3.4]. It is worth to emphasize that this is not the case of some particular VMS models, as it is shown in [151] and discussed below.

VMS was introduced in [100, 101] as a framework for the motivation and development of stabilization techniques, which aim to overcome numerical difficulties encountered when using the standard Galerkin method. On the one hand, the velocity and pressure finite element (FE) spaces need to satisfy the *inf-sup* compatibility condition that guarantees pressure stability and precludes the use of equal order interpolation. Mixed methods satisfying this condition can be used and their finite volume counterpart, based on staggered grids, are common in the LES community. Stabilization techniques that permit the use of equal order interpolation were proposed, e.g., in [68, 102]. On the other hand, global nonphysical oscillations appear in the convection dominated regime, when the mesh is not fine enough, that is, for high mesh Reynolds number flows. The only way to overcome this problem is through the addition of some form of dissipation which was recognized in the early development of stabilized methods [38]. Let us note that the common practice in the LES community is to rely on the explicit extra term introduced by the physical model using high order approximations of the convective term. [1]

The first attempts to perform LES using VMS concepts, presented in [103, 105, 104, 125, 113], were performed introducing explicit subgrid modeling. The VMS models used in these works split resolved scales into large and small, introducing an explicit LES model to account for the small scales stress tensor, e.g., a Smagorinsky-type dissipative term acting on the small scales only. As a result, an important fraction of the degrees of freedom are used for the small resolved scales whereas consistency is retained in the large resolved scales only.

ILES using a VMS approach with resolved and unresolved subgrid scales (the setting that permits to recover stabilized formulations) was suggested in [51] and performed in [42, 23, 144]. Excellent results were first presented in [23], but using isogeometric analysis for the space approximation [99]. Compared to classical LES based on filtering, the VMS approach does not face difficulties associated to inhomogeneous non-commutative filters in wall-bounded flows. Further, it retains numerical consistency in the FE equations and optimal convergence up to the interpolation order whereas, e.g., Smagorinsky models introduce a consistency error of order $h^{4/3}$ (see [103, 105, 23]).

Scale separation is achieved in the VMS formalism by a variational projection. The continuous unknown is split into a resolvable FE component and an unresolvable subgrid or subscale component. The action of the subscales onto the FE scales can be approximated in different ways, leading to different VMS models but in all cases these models are *residual based* (no eddy viscosity is introduced), which permits to retain consistency. Among the modeling possibilities is the choice of the subscale space, first discussed in [50], where it was enforced to be L^2-orthogonal to the FE space. Another modeling ingredient is the possibility of considering time-dependent subscales and to keep the VMS decomposition in all the nonlinear terms, which was studied in [51, 55]. Clear improvements have been observed when using dynamic and fully nonlinear models for the simulation of laminar flows [55, 9].

In this chapter we assess implicit VMS models for the numerical simulation of turbulent flows. We refer to the original references for a comprehensive treatment of the assumptions of the formulations and their numerical analysis. Our intention here is to compare the different VMS schemes in terms of quality of the results and computational cost, and discuss some implementation aspects that we find particularly relevant for the simulation of turbulent flows. Our main motivation is to compare the influence of using orthogonal subscales, in order to enrich current comparisons on VMS techniques for large-eddy simulations, such as [78], where only non-projected subscales are considered. We present a detailed numerical experimentation for three well known

---

[1] It is worth to point out that both problems (convection instability and compatibility conditions) are also present in the *linear* Oseen problem. One of the inconsistencies of an explicit LES approach without a numerical dissipation term is that convection is stabilized by a term that comes from the physical model of the nonlinear Navier Stokes equations and such a term is not present when the linear Oseen problem is considered.

problems: the decay of homogeneous isotropic turbulence (DHIT), the Taylor-Green vortex (TGV) and the turbulent flow in a channel (TCF), described in Section 3. Thus, both unbounded and wall-bounded flows are considered; only wall-bounded tests are performed in [78]. Some other differences with respect to [78] are: 1) we consider a nonlinear sub-scale equation; 2) we do not include the time step size in the stabilization term; 3) we have analyzed the effect of the skew-symmetric term.

The first implementation aspect we discuss is the treatment of the convective term. As it is well-known, the numerical analysis [15, 39, 86] requires a skew-symmetric form of this term in order to avoid any positive contribution to the energy estimates that cannot be properly controlled. The construction of numerical schemes that preserve the skew-symmetry of the convective term at the discrete level has been long studied in the finite difference and finite volume contexts [4, 179, 79, 177, 64, 172]. However, FE formulations used to perform VMS-based LES use either the conservative form [103, 125, 23, 40, 42, 78, 82, 136] or the non-conservative one [113, 55, 151]. The approach we follow here is similar to that in [175] and is based on a split of the convective term into conservative and non-conservative terms. In the FE (variational) context, this simple approach guarantees the preservation of skew-symmetry at the discrete level. We remark that in the nonlinear VMS models the convective velocity is discontinuous (due to the subscale contribution), which prevents us to use some popular skewsymmetric forms. We also show numerically that a positive energy contribution actually appears if a non skew-symmetric form is used.

The second point that we address is the use of weighted (by the stabilization parameter) projections and consistent mass matrices when orthogonal subscales are considered. Even though it is cheaper to use non-weighted projections and lumped mass matrices, only the use of consistent projections guarantees exact L^2 orthogonality. An alternative is the use of Scott-Zhang projections recently proposed in [11], although we do not consider this approach here.

We also discuss the influence of the algorithmic constants of the stabilization parameters in the numerical results. In particular, we show that the choice of the stabilization parameter multiplying the div-div term has a strong influence on the numerical results while it is not essential for stability and convergence of the methods. We further analyze the behavior of the VMS formulation as the time step size is reduced. These two facts are actually related by the way the stabilization parameters are usually defined (see [78, 98]).

Finally, we compare the results obtained using VMS models against those obtained using classical LES based on filtering and the dynamic Smagorinsky closure [71], and another implicit LES method, the adaptive local deconvolution presented in [96].

The chapter is organized as follows. In Section 4.2 we present the VMS formulation, how to compute truly orthogonal subscales and the different models we aim at analyzing, whereas in Section 4.3 we discuss energy conservation statements and how they are influenced by the choice of the VMS method and the definition of the convective term. Sections 4.5.1, 4.5.2, 4.5.3 are devoted to the numerical approximation of the DHIT, the TGV, and the TCF problems, respectively. Sections 4.5.4 and 4.5.5 discuss the effect of the algorithmic constants on the results and the behavior of the different schemes in the small time step limit. Some remarks close the article in Section 4.6.

4.2 Problem statement

4.2.1 The Galerkin semi-discrete problem

In order to improve the readability of this chapter we refresh the Navier-Stokes formulation that was already stated in Chapter 2.

Let Ω be a bounded domain of $\mathbb{R}^d$, where $d = 2, 3$ is the number of space dimensions, $\Gamma = \partial\Omega$ its boundary and $[0, T]$ the time interval. The strong form of the incompressible Navier-Stokes problem consists of finding the velocity field $\mathbf{u}$ and the pressure field p such that

$$\partial_t \mathbf{u} - \nu\Delta\mathbf{u} + \mathbf{u}\cdot\nabla\mathbf{u} + \nabla p = \mathbf{f} \qquad \text{in } \Omega \times (0, T), \tag{4.1}$$

$$\nabla \cdot \mathbf{u} = 0 \qquad \text{in } \Omega \times (0, T), \tag{4.2}$$

with $\mathbf{f}$ the force vector and ν the kinematic viscosity.

Equations (4.1)-(4.2) have to be supplied with appropriate boundary and initial conditions. The boundary Γ is divided into the Dirichlet (Γ_D) and the Neumann (Γ_N) parts such that $\Gamma_D \cup \Gamma_N = \Gamma$ and $\Gamma_D \cap \Gamma_N = \emptyset$. Then, the boundary and initial conditions can be written as

$$\mathbf{u} = \mathbf{u}_g \qquad \text{on } \Gamma_D \times (0, T], \tag{4.3}$$

$$(-p\mathbf{I} + \nu(\nabla\mathbf{u} + \nabla\mathbf{u}^T)) \cdot \mathbf{n} = \mathbf{t}_N \qquad \text{on } \Gamma_N \times (0, T], \tag{4.4}$$

$$\mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0(\mathbf{x}) \qquad \text{in } \Omega \times \{0\}, \tag{4.5}$$

$\mathbf{n}$ being the unit outward vector normal to Γ. To simplify the exposition, we will consider $\mathbf{u}_g = \mathbf{0}$ and $\Gamma_D = \Gamma$ in what follows.

The weak form of the incompressible Navier-Stokes problem (4.1)-(4.5) consists, e.g., in finding $[\mathbf{u}, p] \in L^2(0, T; \mathcal{V}_0) \times \mathcal{D}'(0, T; \mathcal{Q}_0)$ (distributions in time with values in $\mathcal{Q}_0$) such that

$$(\partial_t \mathbf{u}, \mathbf{v}) + B(\mathbf{u}; [\mathbf{u}, p], [\mathbf{v}, q]) = \langle \mathbf{f}, \mathbf{v} \rangle \qquad \forall \mathbf{v} \in \mathcal{V}_0, \quad \forall q \in \mathcal{Q}_0, \tag{4.6}$$

satisfying the initial condition (4.5) in a weak sense. Here $\mathcal{V}_0 := H_0^1(\Omega)^d$, $\mathcal{Q}_0 := L^2(\Omega)/\mathbb{R}$ and the form $B(\mathbf{a}; [\mathbf{u}, p], (\mathbf{v}, q))$ is defined as

$$B(\mathbf{a}; [\mathbf{u}, p], [\mathbf{v}, q]) := \nu(\nabla\mathbf{u}, \nabla\mathbf{v}) + b(\mathbf{a}, \mathbf{u}, \mathbf{v}) - (p, \nabla \cdot \mathbf{v}) + (q, \nabla \cdot \mathbf{u}) \tag{4.7}$$

where the trilinear weak form of the convective term $b(\mathbf{u},\mathbf{v},\mathbf{w})$ can be written in the following three equivalent ways

$$b(\mathbf{u},\mathbf{v},\mathbf{w}) = (\mathbf{u}\cdot\nabla\mathbf{v},\mathbf{w}) \qquad \text{Non conservative,} \tag{4.8}$$

$$b(\mathbf{u},\mathbf{v},\mathbf{w}) = \frac{1}{2}(\mathbf{u}\cdot\nabla\mathbf{v},\mathbf{w}) - \frac{1}{2}(\mathbf{v},\mathbf{u}\cdot\nabla\mathbf{w}) \qquad \text{Skew-symmetric (type 1),} \tag{4.9}$$

$$b(\mathbf{u},\mathbf{v},\mathbf{w}) = (\mathbf{u}\cdot\nabla\mathbf{v},\mathbf{w}) + \frac{1}{2}(\mathbf{v}\cdot\mathbf{w},\nabla\cdot\mathbf{u}) \qquad \text{Skew-symmetric (type 2).} \tag{4.10}$$

This equivalence is lost at the discrete level. The skew-symmetric form (type 2) (4.10) is very common when numerical analysis are presented [15, 39, 86] but the skew-symmetric form (type 2) (4.9) has important advantages when the first argument is a discontinuous function, as will be shown below.

Let us now consider a FE partition $\mathcal{T}_h$ of the domain Ω from which we can construct conforming finite dimensional spaces for the velocity $\mathcal{V}_{0,h} \subset \mathcal{V}_0$, and for the pressure $\mathcal{Q}_{0,h} \subset \mathcal{Q}_0$.

The Galerkin FE approximation of (4.6) consists in finding $[\mathbf{u}_h, p_h] \in L^2(0,T;\mathcal{V}_{0,h}) \times \mathcal{D}'(0,T;\mathcal{Q}_{0,h})$ such that

$$(\partial_t\mathbf{u}_h,\mathbf{v}_h) + B(\mathbf{u}_h;[\mathbf{u}_h,p_h],[\mathbf{v}_h,q_h]) = \langle\mathbf{f},\mathbf{v}_h\rangle \qquad \forall\mathbf{v}_h\in\mathcal{V}_{0,h}, \forall q_h\in\mathcal{Q}_{0,h}. \tag{4.11}$$

4.2.2 VMS framework

As it was pointed out in Section 2.4, the Galerkin FE formulation (4.11) suffers from numerical instabilities for high mesh Reynolds number problems, i.e., convection dominated flows. The discrete *inf-sup* condition that must be satisfied by the pair $\mathcal{V}_{0,h}\times\mathcal{Q}_{0,h}$ in order to have a well-posed problem with bounded pressure is also a problem that arise in the Galerkin FE formulation. The VMS method described in Section 2.4 overcomes these two instabilities.

After the space splitting into the FE space and the subscale space $\mathcal{V}_0 = \mathcal{V}_{0,h}\oplus\widetilde{\mathcal{V}}_0$ and $\mathcal{Q} = \mathcal{Q}_{0,h}\oplus\widetilde{\mathcal{Q}}_0$, where $\widetilde{\mathcal{V}}_0$ and $\widetilde{\mathcal{Q}}_0$ are infinite-dimensional spaces that complete the FE spaces in $\mathcal{V}_0$ and $\mathcal{Q}_0$, respectively. The VMS semi-discrete problem reads: find $[\mathbf{u}_h, p_h] \in L^2(0,T;\mathcal{V}_{0,h}) \times \mathcal{D}'(0,T;\mathcal{Q}_{0,h})$ such that

$$(\partial_t\mathbf{u}_h,\mathbf{v}_h) + (\partial_t\tilde{\mathbf{u}},\mathbf{v}_h) + B(\mathbf{a};[\mathbf{u}_h,p_h],[\mathbf{v}_h,q_h]) + (\tilde{\mathbf{u}},\mathcal{L}^*_{\mathbf{a}}(\mathbf{v}_h,q_h))_h - (\tilde{p},\nabla\cdot\mathbf{v}_h) = \langle\mathbf{f},\mathbf{v}_h\rangle\,. \tag{4.12}$$

Where $\mathcal{L}^*_{\mathbf{a}}(\mathbf{v}_h,q_h)$ is the adjoint operator defined in (2.38). The fine scale problem is approximated by

$$\partial_t\tilde{\mathbf{u}} + \tau_m^{-1}\tilde{\mathbf{u}} = \mathcal{P}(\mathbf{R}_u), \tag{4.13}$$

$$\tau_c^{-1}\tilde{p} = \mathcal{P}(R_p). \tag{4.14}$$

With $\mathbf{R}_u$ the momentum equation residual (2.41), R_p the continuity equation residual (2.42). τ_m and τ_c are the momentum and continuity equations stabilization parameters, respectively, which definition is given by

$$\tau_m = \left(\frac{c_1\nu}{h^2} + \frac{c_2|\mathbf{a}|}{h}\right)^{-1}, \tag{4.15}$$

$$\tau_c = \frac{h^2}{c_1\tau_m}. \tag{4.16}$$

The particular VMS method will result from the combination of the three particular ingredients discussed in Sections 2.4.1-2.4.3, the dynamics of the subscale velocity, the nonlinearity of the advection velocity and the definition of the projection.

4.3 Energy balance statements

In this section we revisit global energy conservation statements of the method. As shown in [151], similar statements can be obtained locally (in a volume $\omega \subset \Omega$).

Taking $\mathbf{v}_h = \mathbf{u}_h$ and $q_h = p_h$ in (4.12) we have the energy balance on the FE component

$$\begin{aligned}&\underbrace{\frac{1}{2}d_t\|\mathbf{u}_h\|^2}_{I} + \underbrace{\nu\|\nabla\mathbf{u}_h\|^2}_{II} + \underbrace{b(\mathbf{a},\mathbf{u}_h,\mathbf{u}_h)}_{III} \\ &\quad + \underbrace{(\partial_t\tilde{\mathbf{u}},\mathbf{u}_h) + (\tilde{\mathbf{u}},\mathcal{L}^*_{\mathbf{a}}(\mathbf{u}_h,p_h))_h - (\tilde{p},\nabla\cdot\mathbf{u}_h)}_{IV} = \underbrace{\langle\mathbf{f},\mathbf{u}_h\rangle}_{V},\end{aligned} \tag{4.17}$$

In equation (4.17) we group the terms as

I)	FE kinetic energy variation:	$\frac{1}{2}d_t\|\mathbf{u}_h\|^2$
II)	FE viscous dissipation:	$\nu\|\nabla\mathbf{u}_h\|^2$
III)	FE convective term:	$b(\mathbf{a},\mathbf{u}_h,\mathbf{u}_h)$
IV)	FE to SGS energy transfer:	$\varepsilon_h = (\partial_t\tilde{\mathbf{u}},\mathbf{u}_h) + (\tilde{\mathbf{u}},\mathcal{L}^*_{\mathbf{a}}(\mathbf{u}_h,p_h))_h - (\tilde{p},\nabla\cdot\mathbf{u}_h)$
V)	FE component of external power:	$\langle\mathbf{f},\mathbf{u}_h\rangle$

Multiplying (4.13) by $\tilde{\mathbf{u}}$ and (4.14) by $\tilde{p}$, integrating over the domain and decomposing the residual of the momentum equation as $\mathbf{R}_u = \mathbf{f} - \partial_t\mathbf{u}_h - \mathcal{L}_{\mathbf{a}}(\mathbf{u}_h,p_h)$, we obtain the global energy balance on the fine scale

$$\begin{aligned}&\underbrace{\frac{1}{2}d_t\|\tilde{\mathbf{u}}\|^2}_{I} + \underbrace{\tau_m^{-1}\|\tilde{\mathbf{u}}\|^2}_{II} + \underbrace{\tau_c^{-1}\|\tilde{p}\|^2}_{III} \\ &\quad + \underbrace{(\mathcal{P}(\partial_t\mathbf{u}_h),\tilde{\mathbf{u}}) + (\mathcal{P}(\mathcal{L}_{\mathbf{a}}(\mathbf{u}_h,p_h)),\tilde{\mathbf{u}})_h + (\mathcal{P}(\nabla\cdot\mathbf{u}_h),\tilde{p})}_{IV} = \underbrace{(\mathcal{P}(\mathbf{f}),\tilde{\mathbf{u}})}_{V}.\end{aligned} \tag{4.18}$$

We group the terms in (4.18) as

I) SGS kinetic energy variation: $\frac{1}{2}d_t\|\tilde{\mathbf{u}}\|^2$

II) SGS velocity dissipation: $\tau_m^{-1}\|\tilde{\mathbf{u}}\|^2$

III) SGS pressure dissipation: $\tau_c^{-1}\|\tilde{p}\|^2$

IV) SGS to FE energy transfer: $\tilde{\varepsilon} = (\mathcal{P}(\partial_t \mathbf{u}_h), \tilde{\mathbf{u}}) + (\tilde{\mathbf{u}}, \mathcal{P}(\mathcal{L}_{\mathbf{a}}(\mathbf{u}_h, p_h)))_h + (\tilde{p}, \mathcal{P}(\nabla \cdot \mathbf{u}_h))$

V) SGS component of external power: $(\mathcal{P}(\mathbf{f}), \tilde{\mathbf{u}})$

Finally, adding up equations (4.17) and (4.18) we obtain an equation for the total kinetic energy

$$
\begin{aligned}
&\frac{1}{2}d_t\|\mathbf{u}_h\|^2 + \frac{1}{2}d_t\|\tilde{\mathbf{u}}\|^2 + \nu\|\nabla\mathbf{u}_h\|^2 + b(\mathbf{a}, \mathbf{u}_h, \mathbf{u}_h) + \tau_m^{-1}\|\tilde{\mathbf{u}}\|^2 + \tau_c^{-1}\|\tilde{p}\|^2 \\
&\quad + (\partial_t\tilde{\mathbf{u}}, \mathbf{u}_h) + (\mathcal{P}(\partial_t\mathbf{u}_h), \tilde{\mathbf{u}}) + (\mathcal{P}(\mathcal{L}_{\mathbf{a}}(\mathbf{u}_h, p_h)) + \mathcal{L}^*_{\mathbf{a}}(\mathbf{u}_h, p_h), \tilde{\mathbf{u}})_h \\
&\qquad + (\mathcal{P}(\nabla\cdot\mathbf{u}_h) - \nabla\cdot\mathbf{u}_h, \tilde{p}) = \langle \mathbf{f}, \mathbf{u}_h\rangle + ((\mathcal{P}(\mathbf{f}), \tilde{\mathbf{u}})\,.
\end{aligned}
\tag{4.19}
$$

Let us note the presence of $b(\mathbf{a}, \mathbf{u}_h, \mathbf{u}_h)$, which is zero only when the skew-symmetric type 1 form is considered. Other choices could result in a spurious positive contribution to the FE kinetic energy as it is actually observed in the DHIT problem, and could result in a loss of stability, although that was not observed.

4.3.1 Static subscales

In this case the energy balance for the subscale is meaningless because there are explicit expressions for the subscales (2.46) and (4.14). When (2.46) and (4.14) are used in (4.17), we obtain

$$
\begin{aligned}
&\frac{1}{2}d_t\|\mathbf{u}_h\|^2 + \nu\|\nabla\mathbf{u}_h\|^2 + b(\mathbf{a}, \mathbf{u}_h, \mathbf{u}_h) + \left(\tau_m\mathcal{P}\left(\partial_t\mathbf{u}_h\right), \mathcal{L}^*_{\mathbf{a}}(\mathbf{u}_h, p_h)\right)_h \\
&\quad + \left(\tau_m\mathcal{P}\left(\mathcal{L}_{\mathbf{a}}(\mathbf{u}_h, p_h)\right), \mathcal{L}^*_{\mathbf{a}}(\mathbf{u}_h, p_h)\right)_h + \tau_c\left\|\mathcal{P}(\nabla\cdot\mathbf{u}_h)\right\|^2 \\
&\quad = \langle \mathbf{f}, \mathbf{u}_h\rangle + \left(\tau_m\mathcal{P}(\mathbf{f}), \mathcal{L}^*_{\mathbf{a}}(\mathbf{u}_h, p_h)\right)_h\,.
\end{aligned}
\tag{4.20}
$$

In the case of the ASGS method, where $\mathcal{P} := \mathbf{I}$, the fourth term on the left hand side is a source of problems. One the one hand, it cannot be neglected because it is needed to make the method consistent. On the other hand, it can only be controlled by the dissipation of the time integration scheme and is therefore responsible for the introduction of a restriction on the time step size. As a side problem, it is very inconvenient for an implementation if any explicit (operator splitting) time integration is chosen as it results in a non-symmetric mass matrix. This term is not present if the OSS method is chosen using the projection $\mathcal{P} := \mathbf{I} - \Pi_h$. Stability of both the fully discrete and the semidiscrete Stokes problem have been proven in [13].

The important term is the fifth one, which permits to control $\tau_m\left\|\mathcal{P}\left(\mathbf{a}\cdot\nabla\mathbf{u}_h + \nabla p_h\right)\right\|^2$; the FE part in the OSS formulation is readily controlled using inverse estimates. It therefore provides the essential numerical stability. The last term acts as a penalty on the divergence constraint, adding volumetric diffusion and provides (extra, non-essential) numerical stability.

For the OSS method, it is proved in [85] that the dissipative structure of the discrete problem has the same statistical behavior in fully developed turbulence than the continuous problem, in the sense that this dissipation has the same estimates as the molecular one. Both dissipations could be made equal by a proper choice of the stabilization parameters in (4.15). This, however, requires a small change in the advection velocity of this expression, which depends on an integral length of the problem. See [85] for details.

4.3.2 Dynamic subscales

In this case, the time derivative of both the FE and subgrid components have to be considered and an estimator for the kinetic energy variation of both the FE and subgrid velocity can be obtained. The stability of the subgrid scale velocity can then be used to obtain a stability estimate of the FE component in a norm that includes the convective and pressure terms [55, 13, 14].[2] Therefore, the numerical dissipation of the method is actually given by the energy transfer ε_h from the FE to the subscale component. Using (4.13)-(4.14), we get:

$$\begin{aligned}\varepsilon_h = (\partial_t \tilde{\mathbf{u}}, \mathbf{u}_h) &- (\tau_m \partial_t \tilde{\mathbf{u}}, \mathcal{L}^*_{\mathbf{a}}(\mathbf{u}_h, p_h))_h - (\tau_m \mathcal{P}\left(\partial_t \mathbf{u}_h\right), \mathcal{L}^*_{\mathbf{a}}(\mathbf{u}_h, p_h))_h \\ &- (\tau_m \mathcal{P}\left(\mathcal{L}_{\mathbf{a}}(\mathbf{u}_h, p_h)\right), \mathcal{L}^*_{\mathbf{a}}(\mathbf{u}_h, p_h))_h + \tau_c \left\| \mathcal{P} \nabla \cdot \mathbf{u}_h \right\|^2 .\end{aligned} \tag{4.21}$$

Except from the viscous contribution, the last two terms in (4.21) are positive, providing dissipation of the FE energy, but the first three could be negative, providing these models with a mechanism to predict a backward energy transfer, not frequently found in classical LES models [157]. It is justified in [54] that even if the first three terms may be negative at a certain time instant, their averaged contribution in a time window greater than the largest period needs to be positive, which is the behavior expected of backscatter from a physical point of view.

For the ASGS method, i.e., $\mathcal{P} := \mathbf{I}$, the last term in the left hand side of (4.19) vanishes and the previous one reads

$$(\mathcal{L}_{\mathbf{a}}(\mathbf{u}_h, p_h) + \mathcal{L}^*_{\mathbf{a}}(\mathbf{u}_h, p_h), \tilde{\mathbf{u}})_h = -2\left(\nu \Delta \mathbf{u}_h, \tilde{\mathbf{u}}\right)_h . \tag{4.22}$$

In turn, the time derivatives of the FE and subscale velocities can be combined as

$$\frac{1}{2} d_t \|\mathbf{u}_h\|^2 + \frac{1}{2} d_t \|\tilde{\mathbf{u}}\|^2 + (\partial_t \tilde{\mathbf{u}}, \mathbf{u}_h) + (\partial_t \mathbf{u}_h, \tilde{\mathbf{u}}) = \frac{1}{2} d_t \|\mathbf{u}_h + \tilde{\mathbf{u}}\|^2 \tag{4.23}$$

---

[2] However, it should be kept in mind that the numerical solution of the problem is the FE component. There is no reason to add the subscale to the final solution as the approximation is limited by the interpolation order, see [55, Remark 10].

to rewrite (4.19) as

$$\begin{aligned}\frac{1}{2}d_t\|\mathbf{u}_h+\tilde{\mathbf{u}}\|^2+\nu\|\nabla\mathbf{u}_h\|^2+b(\mathbf{a},\mathbf{u}_h,\mathbf{u}_h)& \\ +\tau_m^{-1}\|\tilde{\mathbf{u}}\|^2+\tau_c^{-1}\|\tilde{p}\|^2-2\left(\nu\Delta\mathbf{u}_h,\tilde{\mathbf{u}}\right)&=\langle\mathbf{f},\mathbf{u}_h\rangle+\langle\mathbf{f},\tilde{\mathbf{u}}\rangle\,.\end{aligned}\tag{4.24}$$

From this equation, a stability estimate for $\|\mathbf{u}_h+\tilde{\mathbf{u}}\|$ can be obtained as the last term on the left hand side can be controlled using the second one (see [13, Remark 4.7]).

Another important point of (4.24) is that it immediately shows that when the mesh is fine enough, i.e.,

$$\frac{|\mathbf{a}|\,h}{\nu}\ll 1,$$

the dissipation of the total energy depends only on the viscosity. Therefore, the dissipative structure is correctly predicted when a laminar flow is considered or when the discretization is fine enough to resolve all scales of the flow, an important advantage over other LES techniques.

On the other hand, for the OSS method, the FE and subgrid kinetic energy can be summed to obtain the total one

$$\frac{1}{2}d_t\|\mathbf{u}_h\|^2+\frac{1}{2}d_t\|\tilde{\mathbf{u}}\|^2=\frac{1}{2}d_t\|\mathbf{u}_h+\tilde{\mathbf{u}}\|^2.\tag{4.25}$$

since $(\partial_t\tilde{\mathbf{u}},\mathbf{u}_h)=(\partial_t\mathbf{u}_h,\tilde{\mathbf{u}})=0$ as soon as we enforce the subscale to be orthogonal to the FE space. This property also guarantees that

$$(\Pi_m(\mathcal{L}_{\mathbf{a}}(\mathbf{u}_h,p_h)),\tilde{\mathbf{u}})=0\tag{4.26}$$

$$(\Pi_c(\nabla\cdot\mathbf{u}_h),\tilde{p})=0\tag{4.27}$$

which implies that the last term on the left hand side of (4.19) vanishes and that the previous one can be written as

$$(\mathcal{P}(\mathcal{L}_{\mathbf{a}}(\mathbf{u}_h,p_h))+\mathcal{L}^*_{\mathbf{a}}(\mathbf{u}_h,p_h),\tilde{\mathbf{u}})_h=((\mathcal{L}_{\mathbf{a}}(\mathbf{u}_h,p_h)+\mathcal{L}^*_{\mathbf{a}}(\mathbf{u}_h,p_h),\tilde{\mathbf{u}})_h=-2\left(\nu\Delta\mathbf{u}_h,\tilde{\mathbf{u}}\right)_h\tag{4.28}$$

as in the ASGS case. Let us note that the Laplacian term can be eliminated without affecting the convergence properties of the method. Then, the global energy balance equation (4.19) reads

$$\begin{aligned}\frac{1}{2}d_t\|\mathbf{u}_h\|^2+\frac{1}{2}d_t\|\tilde{\mathbf{u}}\|^2+\nu\|\nabla\mathbf{u}_h\|^2+b(\mathbf{a},\mathbf{u}_h,\mathbf{u}_h)& \\ +\tau_m^{-1}\|\tilde{\mathbf{u}}\|^2+\tau_c^{-1}\|\tilde{p}\|^2&=\langle\mathbf{f},\mathbf{u}_h\rangle+(\mathcal{P}(\mathbf{f}),\tilde{\mathbf{u}})\,,\end{aligned}\tag{4.29}$$

which is exactly (4.24) except for the projection of the force in the last term. Stability and convergence of this formulation have been proved in [14, 15].

4.4 Final discrete problem

Applying a time integration algorithm to (4.12)-(4.13)-(4.14) we get the fully discrete problem. The final implementation of the discrete problem is written here considering a Picard linearization of the convective term and the Backward Euler (BE) scheme for the time discretization. It can be straightforwardly modified to consider the Crank-Nicolson time integration scheme; this last scheme is the one used in the numerical examples of Sections 4.5.1, 4.5.2 and 4.5.3.

Some tips and comments on the implementation of the methods described hereinafter are given in Appendix B.

4.4.1 Algebraic Subgrid Scales (ASGS)

Taking the nonlinear advection velocity definition (2.49) and considering the time derivative in the fine scales, we have the Dynamic and Nonlinear ASGS method, hereinafter Dyn-Nl-ASGS. At time step n and nonlinear iteration i, given $\mathbf{u}_h^{n,i-1}$, $\mathbf{u}_h^{n-1}$, $\tilde{\mathbf{u}}^{n,i-1}$ and $\tilde{\mathbf{u}}^{n-1}$ we compute $\mathbf{u}_h^{n,i}$ and $p_h^{n,i}$ such that

$$\begin{aligned} &\frac{1}{\delta t}(\mathbf{u}_h^{n,i}, \mathbf{v}_h) + B(\mathbf{a}^{n,i-1}; [\mathbf{u}_h^{n,i}, p_h^{n,i}], [\mathbf{v}_h, q_h]) \\ &\qquad + \left(\tau_{m,t}\left[\frac{1}{\delta t}\mathbf{u}_h^{n,i} + \mathcal{L}_{\mathbf{a}^{n,i-1}}(\mathbf{u}_h^{n,i}, p_h^{n,i})\right], \mathcal{L}^*_{\mathbf{a}^{n,i-1}}(\mathbf{v}_h, q_h)\right)_h \\ &\qquad + \left(\tau_c \nabla \cdot \mathbf{u}_h^{n,i}, \nabla \cdot \mathbf{v}_h\right) - \frac{1}{\delta t}\left(\tau_{m,t}\left[\frac{1}{\delta t}\mathbf{u}_h^{n,i} + \mathcal{L}_{\mathbf{a}^{n,i-1}}(\mathbf{u}_h^{n,i}, p_h^{n,i})\right], \mathbf{v}_h\right)_h \\ &\qquad = \langle \mathbf{v}_h, \mathbf{f} \rangle + \frac{1}{\delta t}(\mathbf{u}_h^{n-1}, \mathbf{v}_h) + \frac{1}{\delta t}(\tilde{\mathbf{u}}^{n-1}, \mathbf{v}_h) \\ &\qquad - \frac{1}{\delta t}\left(\tau_{m,t}\left[\frac{1}{\delta t}\mathbf{u}_h^{n-1} + \mathbf{f} + \frac{1}{\delta t}\tilde{\mathbf{u}}^{n-1}\right], \mathbf{v}_h\right) \\ &\qquad + \left(\tau_{m,t}\left[\frac{1}{\delta t}\mathbf{u}_h^{n-1} + \mathbf{f} + \frac{1}{\delta t}\tilde{\mathbf{u}}^{n-1}\right], \mathcal{L}^*_{\mathbf{a}^{n,i-1}}(\mathbf{v}_h, q_h)\right)_h , \end{aligned} \tag{4.30}$$

where $\tau_{m,t} = \left(\delta t^{-1} + \tau_m^{-1}\right)^{-1}$ and $\mathbf{a}^{n,i-1} = \mathbf{u}_h^{n,i-1} + \tilde{\mathbf{u}}^{n,i-1}$.

In turn, $\tilde{\mathbf{u}}^{n,i}$ is computed by solving the discretization of the fine scale problem (4.13). Note that in the nonlinear version of the algorithm, the stabilization parameter $\tau_{m,t}$ depends on the subscale itself through $\mathbf{a}$ in (4.15), making the fine scale equation also nonlinear, although it is local and does not increase the size of the global linear system to be solved. At each integration point of each element we iteratively solve

$$\tilde{\mathbf{u}}^{n,i,k} = \tau_{m,t}^{k-1}\frac{1}{\delta t}\tilde{\mathbf{u}}^{n-1} + \tau_{m,t}^{k-1}\left[\mathbf{f} - \frac{(\mathbf{u}_h^{n,i} - \mathbf{u}_h^{n-1})}{\delta t} - \mathcal{L}_{\mathbf{a}^{n,i,k-1}}(\mathbf{u}_h^{n,i}, p_h^{n,i})\right]. \tag{4.31}$$

where $\mathbf{a}^{n,i,k-1} = \mathbf{u}_h^{n,i} + \tilde{\mathbf{u}}^{n,i,k-1}$ is used in (4.15) to obtain $\tau_{m,t}^{k-1}$.

Alternatively, one can send the corresponding fine scale convective term $\tilde{\mathbf{u}} \cdot \nabla \mathbf{u}_h$ to the left-hand side, improving the convergence of the iterative process as

$$\tilde{\mathbf{u}}^{n,i,k} + \tilde{\mathbf{u}}^{n,i,k} \cdot \nabla \mathbf{u}_h^{n,i} = \tau_{m,t}^{k-1} \frac{1}{\delta t} \tilde{\mathbf{u}}^{n-1} + \tau_{m,t}^{k-1} \left[\mathbf{f} - \frac{(\mathbf{u}_h^{n,i} - \mathbf{u}_h^{n-1})}{\delta t} - \mathcal{L}_{\mathbf{u}_h^{n,i}}(\mathbf{u}_h^{n,i}, p_h^{n,i}) \right]. \tag{4.32}$$

This is a simple fixed-point iterative scheme that we have found efficient and robust for the numerical simulations presented in this paper, although in other situations we have found more convenient to use a conventional Newton-Raphson scheme to solve the nonlinear subscale equation [9].

For the simplest ASGS scheme we do not consider the time derivative of the fine scale, we consider them quasi-static, i.e., $(\partial_t \tilde{\mathbf{u}}, \mathbf{v}_h) = 0$. Note that in any case the subscales will depend on time through the FE residual and the stabilization parameter. On the other hand, the advection velocity is considered to be linear as indicated in (2.48). We label this method as Static Linear ASGS (Sta-Lin-ASGS). Note that the Sta-Lin-ASGS method does not need to explicitly compute $\tilde{\mathbf{u}}$; invoking (4.15) and (4.16) in (4.12) we get a discrete equation only in terms of the FE component.

We can readily define the rest of possible combinations of time and nonlinear treatment considering the linear advection velocity definition and the time-dependence in the subscales (Dyn-Lin-ASGS method) or keeping the static definition of the subscales with the nonlinear choice for the advection velocity (Sta-Nl-ASGS method).

4.4.2 Orthogonal Subscales (OSS)

Let us state the Dynamic and Nonlinear OSS (Dyn-Nl-OSS) method, which means to take into account the nonlinearity of the advection velocity (2.49) and the time derivative of the subscales. At time step n and nonlinear iteration i, given $\mathbf{u}_h^{n,i-1}$, $\mathbf{u}_h^{n-1}$, $\tilde{\mathbf{u}}^{n,i-1}$ and $\tilde{\mathbf{u}}^{n-1}$ we compute $\mathbf{u}_h^{n,i}$ and $p_h^{n,i}$ by solving

$$\begin{aligned}
&\frac{1}{\delta t}(\mathbf{u}_h^{n,i}, \mathbf{v}_h) + B(\mathbf{a}^{n,i-1}; [\mathbf{u}_h^{n,i}, p_h^{n,i}], [\mathbf{v}_h, q_h]) \\
&\quad + \left(\tau_{m,t} \left[\frac{1}{\delta t} \mathbf{u}_h^{n,i} + \mathcal{L}_{\mathbf{a}^{n,i-1}}(\mathbf{u}_h^{n,i}, p_h^{n,i}) \right], \mathcal{L}^*_{\mathbf{a}^{n,i-1}}(\mathbf{v}_h, q_h) \right) \\
&\quad + \left(\tau_c \nabla \cdot \mathbf{u}_h^{n,i}, \nabla \cdot \mathbf{v}_h \right) \\
&\quad = \langle \mathbf{v}_h, \mathbf{f} \rangle + \frac{1}{\delta t}(\mathbf{u}_h^{n-1}, \mathbf{v}_h) \\
&\quad + \left(\tau_{m,t} \left[\mathbf{f} + \frac{1}{\delta t} \tilde{\mathbf{u}}^{n-1} - \boldsymbol{\xi}_m^{n,i-1} \right], \mathcal{L}^*_{\mathbf{a}^{n,i-1}}(\mathbf{v}_h, q_h) \right) \\
&\quad - \left(\tau_c \xi_c^{n,i-1}, \nabla \cdot \mathbf{v}_h \right),
\end{aligned} \tag{4.33}$$

where $\boldsymbol{\xi}_m$ and ξ_c are the weighted projections of the residuals $\mathbf{R}_u$ and R_p (see below) evaluated at the corresponding time step and nonlinear iteration.

Like the Dyn-NL-ASGS method, we also need to compute the subscale velocity $\tilde{\mathbf{u}}$ explicitly. We compute the discrete subscale problem with the OSS counterpart of (4.31) or (4.32), viz.

$$\begin{aligned}\tilde{\mathbf{u}}^{n,i,k} + \tilde{\mathbf{u}}^{n,i,k} \cdot \nabla \mathbf{u}_h^{n,i} &= \tau_{m,t}^{k-1} \frac{1}{\delta t} \tilde{\mathbf{u}}^{n-1} \\ &+ \tau_{m,t}^{k-1} \left[\mathbf{f} - \frac{(\mathbf{u}_h^{n,i} - \mathbf{u}_h^{n-1})}{\delta t} - \mathcal{L}_{\mathbf{u}_h^{n,i}}(\mathbf{u}_h^{n,i}, p_h^{n,i}) \right] - \tau_{m,t}^{k-1} \boldsymbol{\xi}_m^{n,i-1}.\end{aligned} \tag{4.34}$$

Note that $\boldsymbol{\xi}_m^{n,i}$ actually depends on $\tilde{\mathbf{u}}^{n,i}$ via the advection velocity of the convective term $\mathbf{a}^{n,i} = \mathbf{u}_h^{n,i} + \tilde{\mathbf{u}}^{n,i}$. In order to simplify the fine scale computation (4.34) we use the projection at the previous nonlinear iteration, i.e., $\boldsymbol{\xi}_m^{n,i-1}$.

For the dynamic OSS case we should introduce some modifications in the computation of the projection Π_m. At the fully discrete level, in order for $\tilde{\mathbf{u}}^{n+1}$ to be L^2 orthogonal to $\mathcal{V}_h$, we must add to the FE residual the subscale time derivative contribution from the previous time step and use $\tau_{m,t}$ instead of τ_m in the computation of the projections. Finally, the projections of the residuals onto the FE spaces $\boldsymbol{\xi}_m^{n,i}$ and $\xi_c^{n,i}$ are such that

$$(\tau_{m,t}\boldsymbol{\xi}_m^{n,i}, \mathbf{v}_h) = (\tau_{m,t}(\mathbf{R}_u^{n,i} + \frac{1}{\delta t}\tilde{\mathbf{u}}^{n-1}), \mathbf{v}_h) \qquad \forall \mathbf{v}_h \in \mathcal{V}_{0,h}, \tag{4.35}$$

$$(\tau_c \xi_c^{n,i}, q_h) = (\tau_c R_p^{n,i}, q_h) \qquad \forall q_h \in \mathcal{Q}_{0,h}, \tag{4.36}$$

where the residuals $\mathbf{R}_u^{n,i}$ and $R_p^{n,i}$ are evaluated using (2.41) and (2.42) with $\mathbf{a}^{n,i} = \mathbf{u}_h^{n,i} + \tilde{\mathbf{u}}^{n,i}$, $\mathbf{u}_h^{n,i}$, $\mathbf{u}_h^{n-1}$ and $p_h^{n,i}$. Note that when convergence of the nonlinear iteration is achieved, (4.34) and (4.35) guarantee that $(\mathbf{v}_h, \tilde{\mathbf{u}}^n) = 0$ for any $\mathbf{v}_h \in \mathcal{V}_{0,h}$.

4.5 Numerical experiments

4.5.1 Decay of homogeneous isotropic turbulence

This problem, one of the most used benchmarks to test LES models, consists of analyzing the statistics of the turbulent flow in a 3D box of size $\Omega = (0, 2\pi) \times (0, 2\pi) \times (0, 2\pi)$ with periodic boundary conditions in all directions, which is started with a field having a predetermined energy spectrum. A detailed description of the computational domain, initial conditions, and problem setting is given in [133]. Furthermore, the main characteristics of this test are stated in Section 3.2.3.

We recall that the initial energy spectra is:

$$E(k, 0) = 16\sqrt{\frac{2}{\pi}} k_0^{-5} k^4 \exp\left(-2\left(\frac{k}{k_0}\right)^2\right). \tag{4.37}$$

The initial energy spectrum defined in (4.37) has a peak on the wavenumber $k_0 = 6$ which supposes that the total kinetic energy at the initial time step should be $1/2q^2 = 1.5$ and the enstrophy $\frac{1}{2} < \boldsymbol{\omega}^2 >= 67.5$.

Setting

We solve the DHIT test using different meshes (32^3, 64^3 and 128^3 Q_1/Q_1 elements).

We test the ASGS method and the OSS method reported in Section 4.4. The problem is solved considering three different cases for both methods, depending on the definition of the advection velocity and the tracking of the subscales. The advection velocity $\mathbf{a}$ can be linear or nonlinear and the subscales can be dynamic or static (see Subsection 4.2.2). Table 4.1 collects all the VMS combinations to be compared for the different simulations.

Case Id.	Label	Method	Advection velocity ($\mathbf{a}$)	Subscales tracking
1	Sta-Lin-ASGS	ASGS	Linear ($\mathbf{a} = \mathbf{u}_h$)	Static ($\partial_t \tilde{\mathbf{u}} = \mathbf{0}$)
2	Dyn-Lin-ASGS	ASGS	Linear ($\mathbf{a} = \mathbf{u}_h$)	Dynamic ($\partial_t \tilde{\mathbf{u}} \neq \mathbf{0}$)
3	Dyn-Nl-ASGS	ASGS	Nonlinear ($\mathbf{a} = \mathbf{u}_h + \tilde{\mathbf{u}}$)	Dynamic ($\partial_t \tilde{\mathbf{u}} \neq \mathbf{0}$)
4	Sta-Lin-OSS	OSS	Linear ($\mathbf{a} = \mathbf{u}_h$)	Static ($\partial_t \tilde{\mathbf{u}} = \mathbf{0}$)
5	Dyn-Lin-OSS	OSS	Linear ($\mathbf{a} = \mathbf{u}_h$)	Dynamic ($\partial_t \tilde{\mathbf{u}} \neq \mathbf{0}$)
6	Dyn-Nl-OSS	OSS	Nonlinear ($\mathbf{a} = \mathbf{u}_h + \tilde{\mathbf{u}}$)	Dynamic ($\partial_t \tilde{\mathbf{u}} \neq \mathbf{0}$)

TABLE 4.1: DHIT test cases.

In terms of the numerical parameters of the methods, we use the skew-symmetric convective term *type 1* defined in Subsection 4.2.1. The stabilization parameter τ_c is set equal to zero and the algorithmic constants in τ_m are $c_1 = 12$ and $c_2 = 2$ (see Section 4.5.4). Further, we use linear, quadratic, and cubic FEs.

The time integration has been performed using the Crank-Nicolson scheme with an adaptive time step. The initial time step is set to $\delta t_0 = 5.0 \cdot 10^{-3}$ and it is increased at each step multiplying it by an amplification factor. For this test the amplification is equal to 1.1, reading $\delta t_i = 1.1 \cdot \delta t_{i-1}$. The time step is increased step by step until it reaches a predefined threshold, e.g., 0.1 s. If convergence is not attained at either the nonlinear or solver loops, we apply a reduction factor (5.0 for this test) and recalculate the solution using the reduced time step, i.e., $\delta t_i = \frac{1}{5.0}\delta t_{i-1}$. The amplification of the time step when the solution converges allows one to decrease the total amount of time steps needed for the computation. At each time step the nonlinear system is solved as described in Section 4.4.

It has been seen that for a coarse mesh like the 32^3 linear hexahedral elements mesh that we are using for this test, the total amount of energy at the initial time step (computed after the Inverse Fast Fourier Transform of the velocity field is far from the theoretical starting point $E_0 = 1.5$. This error is due to the projection from the Fourier space onto the FE space, which obviously depends on the mesh size. As we refine the mesh, the initial value of total energy tends to 1.5, as can be seen in Figure 4.1.

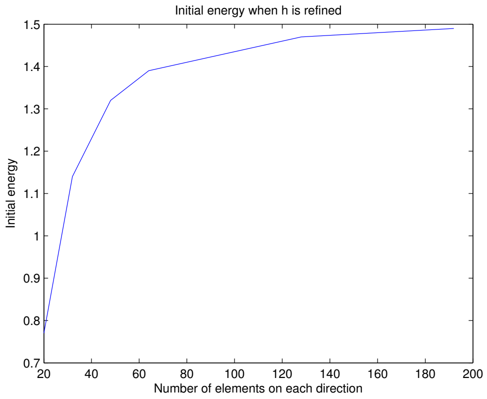


FIGURE 4.1: Initial energy evolution refining the mesh

Knowing this behavior of the initial condition, it is difficult to draw conclusions comparing the simulations with a DNS like AGARD database, [111]. In order to avoid this problem, we scale the initial condition in a such way that its initial energy achieve the desired value. This has been done multiplying the initial condition by a factor defined as

$$\mathbf{u}_{0_{new}} = \sqrt{\frac{E_{0_{new}}}{E_{0_{old}}}} \mathbf{u}_{0_{old}}, \qquad \text{where } E_{0_{new}} = 1.5.$$

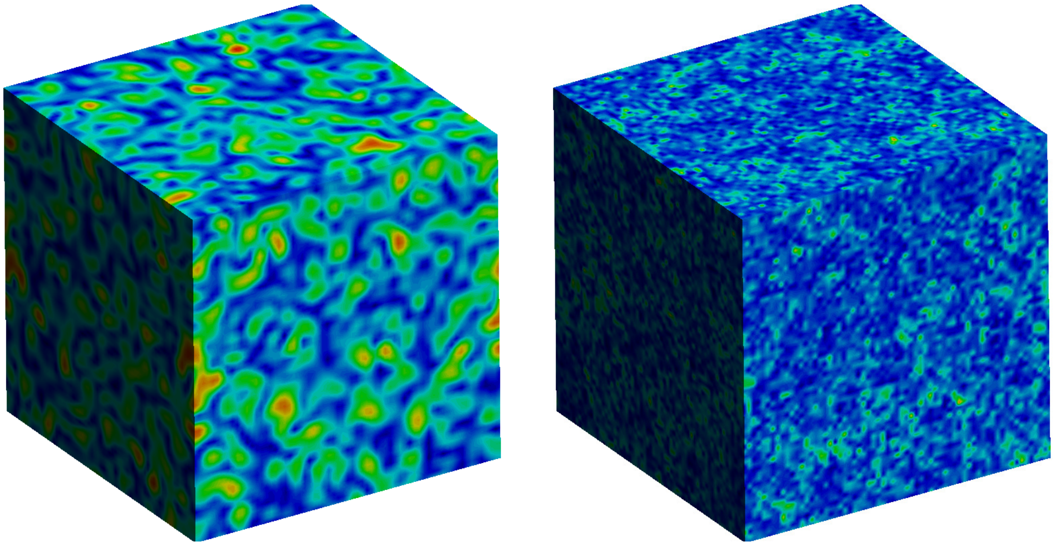
Vorticity

As an introductory result, we show the vorticity contour at two different time steps, $t = 5.0 \cdot 10^{-3}$ and $t = 2.0$. These results are obtained considering the nonlinear definition of $\mathbf{a}$ and the dynamic subscales tracking for the OSS method with a 64^3 hexahedral trilinear elements mesh.

In Figure 4.2 we see that at the earlier time step the structures are much bigger than at $t = 2.0$. As we will see in forthcoming sections, when the energy of the problem is dissipated, the energy containing eddies become smaller, reducing its vorticity. This mechanism induces a flattening of the energy spectra, as we will discuss later.

Energy Conservation

In this section we present results of the energy budget described in Section 4.3 obtained in a 32^3 elements mesh for the ASGS and OSS methods using the dynamic and nonlinear cases. Fig. 4.3 depicts the energy balance evolution for the mean flow equation (4.17) and the subscale equation (4.18) separately for the Dyn-Nl-ASGS case. It can be seen that the variation of kinetic energy shown by the FE component in Fig. 4.3(a) is offset in a large part by the transfer of energy to the subscales, while remaining energy on the mean flow balance is offset by the viscous term. On the other side, Fig. 4.3(b)

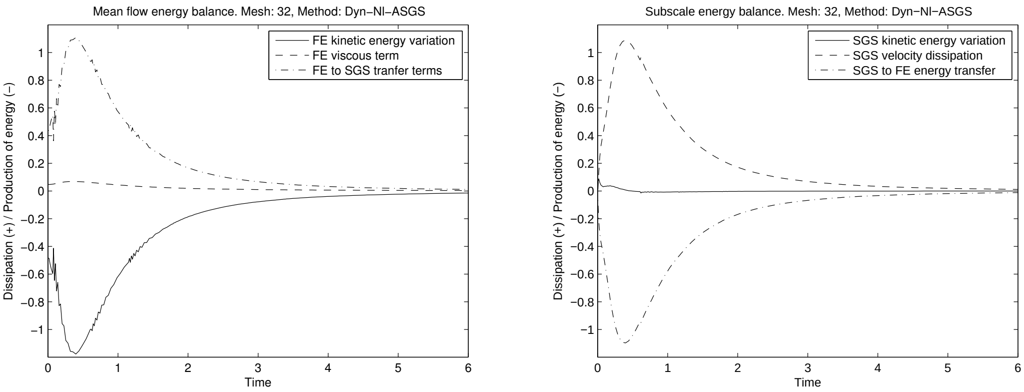


(a) $t = 5.0 \cdot 10^{-3}$

(b) $t = 2.0$

FIGURE 4.2: Vorticity contour

shows that the energy transferred from the FE equation is mainly dissipated by the subscale velocity term. There is an small variation of the kinetic energy of the subscale at the beginning of the simulation. Note that since the viscosity is small, so are the viscous effects compared to the dissipation introduced by the subscale velocity. As we use a skew-symmetric form of the convective term, this term does not affect the energy balance and is not plotted in Fig. 4.3(a). Since $\tau_c = 0$, the pressure subscale term $\tau_c^{-1}\|\tilde{p}\|^2 = \tau_c\|\mathcal{P}(\nabla \cdot \mathbf{u}_h)\|^2$ is also zero and does not appear in Fig. 4.3(b).

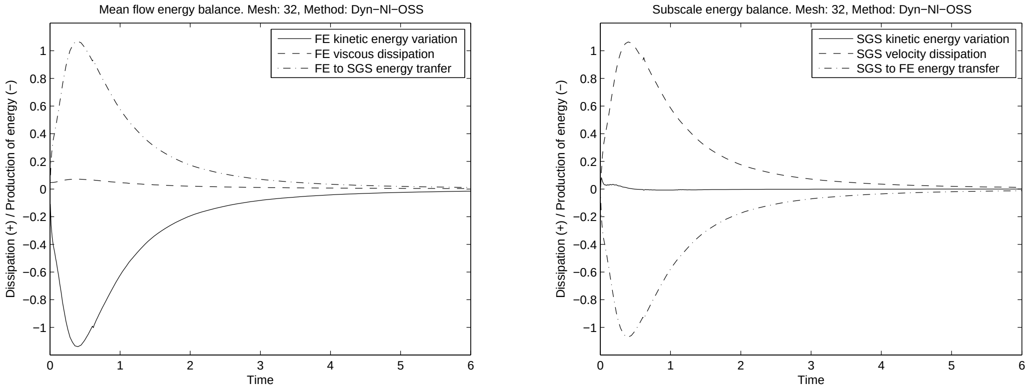


(a) Mean flow equation energy balance

(b) Subscale equation energy balance

FIGURE 4.3: Mean flow and subscale energy balances for the Dyn-Nl-ASGS method.

The energy balance evolution for the mean flow and the subscales equations in (4.17)-(4.18) for the Dyn-Nl-OSS case are shown in Fig. 4.4. Fig. 4.4(a) depicts the energy balance evolution for the mean flow equation. Like for the ASGS method, the loss of kinetic energy is balanced by the FE scales to subscales energy transfer terms. The FE viscous term also has a very little impact on the dissipation of energy. On the other side, the subscales energy balance shown in Fig. 4.4(b) shows that almost all the energy transferred by the FE to the subscales is offset by the subscale velocity term, again like in the ASGS method. The only important difference between both methods is that *no oscillations are observed in the FE kinetic energy evolution when the OSS method is used.*



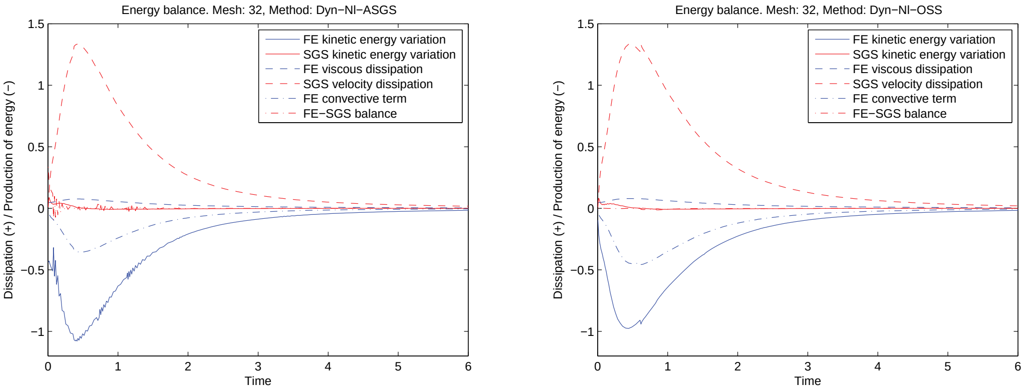
(a) Mean flow equation energy balance (b) Subscale equation energy balance

FIGURE 4.4: Mean flow and subscale energy balances the Dyn-Nl-OSS method.

The global energy balance terms obtained solving the problem with the skew-symmetric convective term *type 2* for the Dyn-Nl-ASGS and Dyn-Nl-OSS cases are shown in Fig. 4.5. We note that the loss of skew-symmetry in the convective term has a non-negligible effect (see Figs. 4.5(a) and 4.5(b)). In particular, this term introduces negative dissipation (production of energy) into the problem. This fact implies that the method is less dissipative and the energy decays at a slower rate than using the convective term *type 1* and the method seems to be less diffusive. This negative contribution, however, is not predictable and could result in a blow up of the calculation. We refer to Section 4.5.4 for further comments about numerical instabilities associated to the *type 2* convective term.

Computational cost analysis

The actual implementation in the parallel FE multiphysics code FEMPAR [16] is based on a classical domain decomposition strategy. At each nonlinear iteration the monolithic linear system is solved using a classical GMRES method applied to the Schur complement over the interfaces of the subdomains. This iterative procedure is preconditioned using a balancing Neumann-Neumann method applied to the monolithic system. The cost of each iteration is that of local Dirichlet solves for the Schur complement



(a) Global energy balance for Dyn-Nl-ASGS

(b) Global energy balance for Dyn-Nl-OSS

FIGURE 4.5: Global energy balance using skew-symmetric convective term *type* 2.

application and a local Neumann solve and a global solve for the preconditioner application (see [132, 66, 19]). All local systems are solved using the sparse direct solvers in PARDISO library [162, 163].

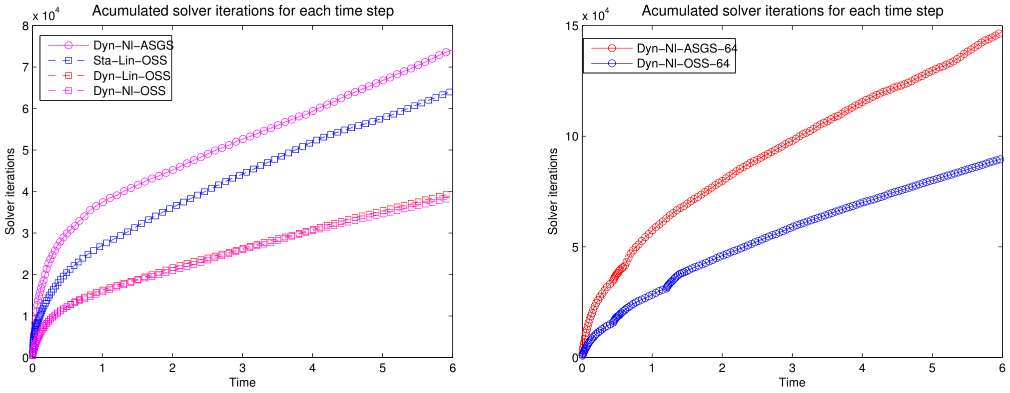
An important issue when comparing different computational methods is their corresponding computational cost. In order to characterize the performance of the different VMS methods introduced in Section 4.2, we analyze some quantities that define the computational cost of each method, viz. nonlinear iterations, iterative solver iterations, and the adaptive time step evolution.

The cases compared here have been solved using 32^3 and 64^3 linear hexahedral element meshes. The 32^3 discretization is very coarse but it allows us to stress the differences between the proposed methods. In fact, due to this discretization, the linear and static ASGS case (Sta-Lin-ASGS) and the dynamic and linear ASGS case (Dyn-Lin-ASGS) do not converge at $t = 0.0$ and $t = 0.123$, respectively; the nonlinear iterations diverge even reducing the time step size. Anyway, all the methods converge as $h \to 0$.

The number of nonlinear iterations needed at each time step by the ASGS method is smaller than the one required by the OSS method in all cases. This is due to the evaluation of the projections at the previous nonlinear iteration $i-1$; the implicit treatment of the projection is carried out by the nonlinear loop. Alternatively, since the projection is a linear operation, it can be performed together with the linear system [52], although a more involved implementation is required. Referring to the OSS method, we observe that the dynamic cases, both linear and nonlinear, need less iterations to achieve convergence without any significant difference between each other.

However, the number of nonlinear iterations is not the most relevant measure of the computational cost as the cost of each iteration is not fixed when iterative linear solvers are considered. Fig. 4.6 shows the accumulated number of solver iterations for each time step for the methods that have attained convergence with the 32^3 mesh (Fig.

4.6(a)) and for the dynamic versions with the 64^3 mesh (Fig. 4.6(b)). Unlike the nonlinear iterations, here we see that the ASGS method requires more solver iterations than OSS. The maximum solver iterations at each time step for the dynamic and nonlinear ASGS case is variable, starting from near 600, dropping to 200 and rising to around 300 iterations at the end of the computation. Meanwhile, all cases of the OSS method remain almost constant, around 60 iterations in the dynamic cases and around 40 iterations in the static one. The relation between time step size and solver iterations for each method is analyzed in Section 4.5.5.



(a) 32^3 mesh

(b) 64^3 mesh

FIGURE 4.6: Accumulated solver iterations.

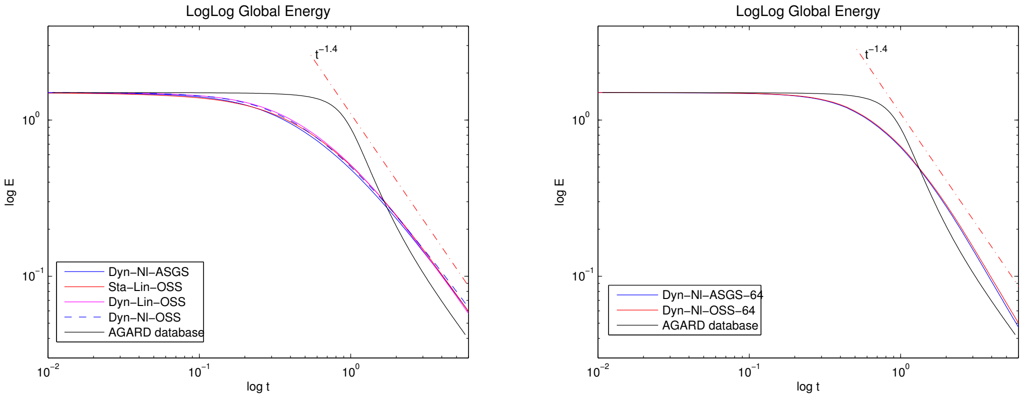
The adaptive time stepping described in the previous subsection has an important role on the computational cost, as mentioned earlier. If the time step is reduced in order to ensure convergence, the global computational cost is increased. Then, we are looking for those methods that do not require time step reductions, consequently reducing the total amount of time step evaluations. In this case, any of the methods shown in Fig. 4.6(a) need to reduce the time step. Since we do not have any time step reduction and the number of solver iterations per step is stabilized after $t = 1$ for the 32^3 mesh and $t = 1.5$ for the 64^3 mesh, the total amount of accumulated solver iterations (in nonlinear and time loops) shown in Fig. 4.6 increases almost linearly. We see in this figure that the ASGS scheme performs worse than OSS in this aspect, with a steeper slope in both the 32^3 and the 64^3 meshes. With respect to the OSS method, we see that the number of nonlinear iterations needed by the static version of this method results in a steeper slope of the accumulated solver iterations. No significant differences appear between the dynamic linear and nonlinear definitions of the OSS method.

Summarizing, ASGS methods need less nonlinear iterations (due to the treatment of the projections in the OSS method), but on the other hand OSS methods need less solver iterations. Furthermore, *ASGS formulations are prone to instabilities; linear formulations diverge and the nonlinear dynamic formulation requires much more solver iterations.*

We can clearly state that the most efficient method for this setting, in terms of computational cost, is the dynamic (both linear and nonlinear) OSS method; all OSS cases are below ASGS. It has to be said that the dynamic nonlinear OSS case requires less nonlinear iterations in some of the time step computations.

Total energy evolution

In this section we present the total energy evolution of the resolved scales, i.e., the FE component. The results are shown in Fig. 4.7 for the 32^3 and the 64^3 grids. We observe that all methods have a very similar accuracy for this test case, still far from the DNS result. The difference between the methods becomes even smaller when the mesh is refined and they are all closer to the DNS solution. Note that we do not plot the non-converged results from the ASGS static cases.



(a) 32^3 mesh (b) 64^3 mesh

FIGURE 4.7: Total energy evolution for the 32^3 and 64^3 elements meshes with the scaled initial condition.

Remark 4.5.1. *The results shown in Figure 4.7 have been obtained using the skewsymmetric convective term type 1,* $b(\mathbf{a}, \mathbf{u}_h, \mathbf{v}_h) = \frac{1}{2}(\mathbf{a} \cdot \nabla \mathbf{u}_h, \mathbf{v}_h) - \frac{1}{2}(\mathbf{a} \cdot \mathbf{u}_h, \nabla \mathbf{v}_h)$*. But what happens if we choose another definition for this term? We know that the nonskewsymmetric term (off) can involve convergence problems. So we tested the influence of this term by solving the same problem with the convective skewsymmetric term (type 2). We did that for the dynamic and nonlinear cases for both methods, ASGS and OSS. The results obtained for this test are shown in Figure 4.8.*

We can clearly see in Figure 4.8 that the solution using the type 2 term is above the obtained with term type 1. Then, we can say that the solution is less dissipative when we use this convective term. It has to be said that in Figure 4.8 we only represent the cases which exhibit more differences. In other words, in order to see the effect of the convective term definition, we only show the dynamic and nonlinear cases for ASGS and OSS methods. On the other hand, here we also have to take into account what we said on Remark 2.4.1. Introducing the nonlinearity on $\mathbf{a}$ *we are increasing its weight on the convective term and changing its correct definition, therefore, it is normal that the differences between each definition increase.*

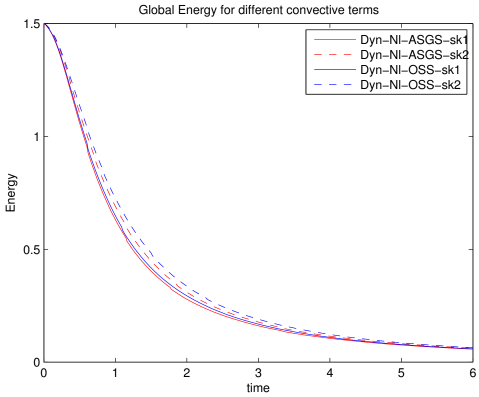


FIGURE 4.8: Total energy evolution. Comparison between different definitions for the convective term

Energy spectra

According to [133], the resolution of the small scales in isotropic decaying turbulence is judged by the shape of the energy spectra at high wave numbers, and requires $k_{max}\eta \approx 1$, $\eta = (\nu^3/\epsilon)^{1/4}$ being the Kolmogorov length scale and k_{max} the maximum wave number required. In this case $k_{max} \approx 182$, which means at least a 300^3 FE mesh for a DNS computation, with a high computational cost. In this section we evaluate the capability of the VMS method to represent the energy of the eddies at the inertial subrange without solving the small scales and compare the results against Kolmogorov's law prediction

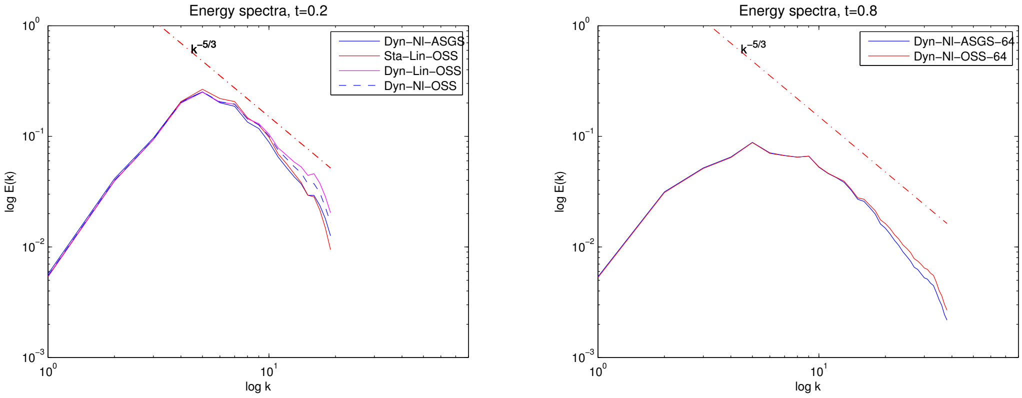
$$E(k) \propto \epsilon^{2/3} k^{-5/3},$$

E being the turbulent kinetic energy.

In Fig. 4.9 the energy spectra for the different cases described in Table 4.1, using 32^3 and 64^3 linear hexahedral element mesh are presented. We can see in Fig. 4.9(a) that the energy spectra at $t = 0.2$ decays with a different slope depending on the VMS method used. Although the differences are small and only appear at large wavenumbers, we see that the dynamic OSS models are less dissipative than the Dyn-Nl-ASGS and Sta-Lin-OSS ones. For the finer 64^3 mesh the difference between the spectra obtained using Dyn-Nl-ASGS and Dyn-Lin-OSS are even smaller, as shown in Fig. 4.9(b); OSS is again less dissipative.

h-p refinement

The energy decay computed using 32^3 and 64^3 linear FE meshes is far from the one obtained using DNS [111], as shown in Fig. (4.7) and discussed above. To make clear that these poor results are due to this crude discretization, we present a mesh refinement analysis, both reducing the element length h and increasing the interpolation order p. We choose the Dynamic and Nonlinear OSS method (Dyn-Nl-OSS), which is the one that shows the lowest slope in the accumulated iterations evolution (Fig. 4.6) for the 32^3



(a) 32^3 mesh (b) 64^3 mesh

FIGURE 4.9: Energy spectra at $t = 0.2$ (32^3 mesh) and $t = 0.8$ (64^3 mesh).

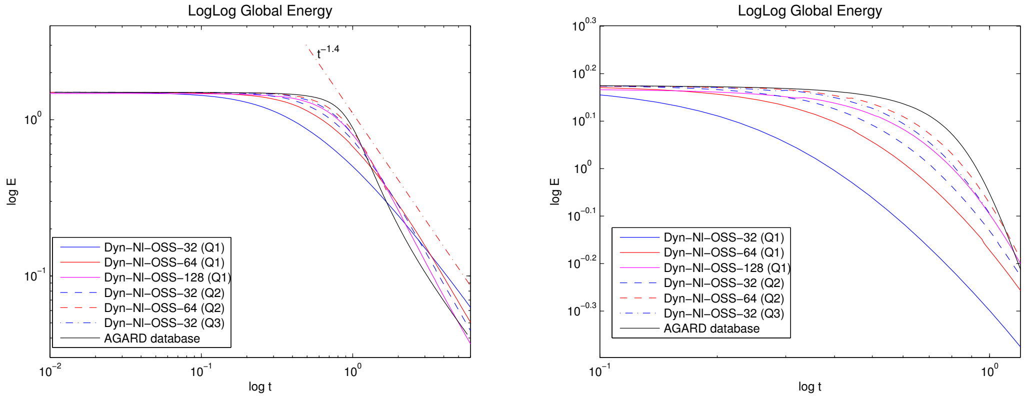
and 64^3 linear elements mesh. We solve the problem using the discretizations exposed in Table 4.2.

Label	Mesh elements	Element type
32 ($Q1$)	32^3	hexahedral linear (Q_1)
64 ($Q1$)	64^3	hexahedral linear (Q_1)
128 ($Q1$)	128^3	hexahedral linear (Q_1)
32 ($Q2$)	32^3	hexahedral quadratic (Q_2)
64 ($Q2$)	64^3	hexahedral quadratic (Q_2)
32 ($Q3$)	32^3	hexahedral cubic ($Q3$)

TABLE 4.2: h-p refinement cases.

In Fig. (4.10) we show the total kinetic energy evolution obtained using the discretizations defined in Table 4.2. Reducing the mesh size h and/or increasing the polynomial order p (not to be confused with the pressure) the result becomes closer to the DNS, as expected. In Fig. 4.10(b) three groups can be clearly observed, namely 32 ($Q1$), 32 ($Q2$) and 64 ($Q1$) and the remaining three. The best results are obtained using $Q2$ elements although the difference is really small.

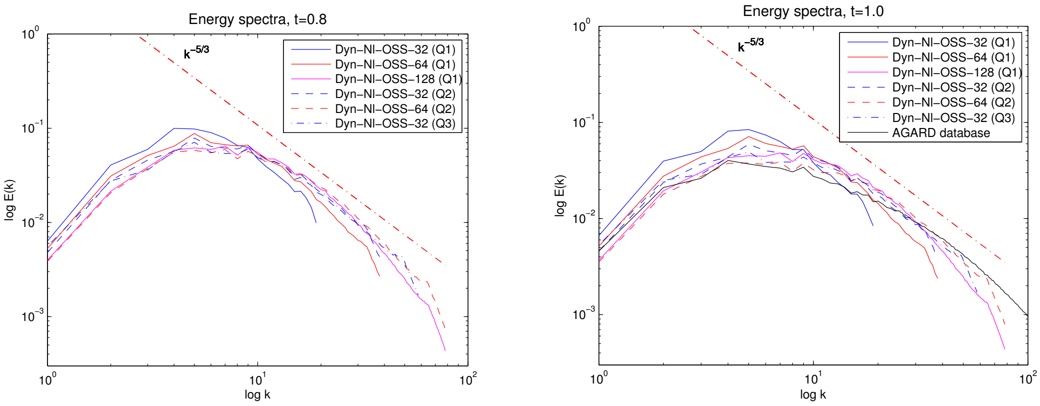
Given the differences in the total energy evolution the time at which the $k^{-5/3}$ law is achieved differs for the different methods. We show the energy spectra at time $t = 0.8$ and $t = 1.0$ for the different cases presented before in Fig. 4.11. As it can be observed in Fig. 4.11(a), at $t = 0.8$, only the energy spectra obtained using the 32 (Q_1) and 64 (Q_1) have a steeper slope, while the other cases are almost parallel to the $k^{-3/5}$ line. This is what was expected since the kinetic energy decay occurs earlier in the coarser cases. In Fig. 4.11(b) we show the energy spectra at $t = 1.0$, and compare it against the DNS spectrum from [111] at the same time step. It can be observed that the results tend to the DNS one as we increase resolution, being the $64(Q_2)$ case the most accurate one.



(a) Total evolution (b) Close up view

FIGURE 4.10: Total kinetic energy evolution.

Note that the DNS spectra is not present in Fig. 4.11(a) because it is not available in the database [111] at this time step.



(a) Energy spectra at $t = 0.8$ (b) Energy spectra at $t = 1.0$

FIGURE 4.11: Energy spectra at $t = 0.8$ and $t = 1.0$ for the h-p refinement defined in Table 4.2.

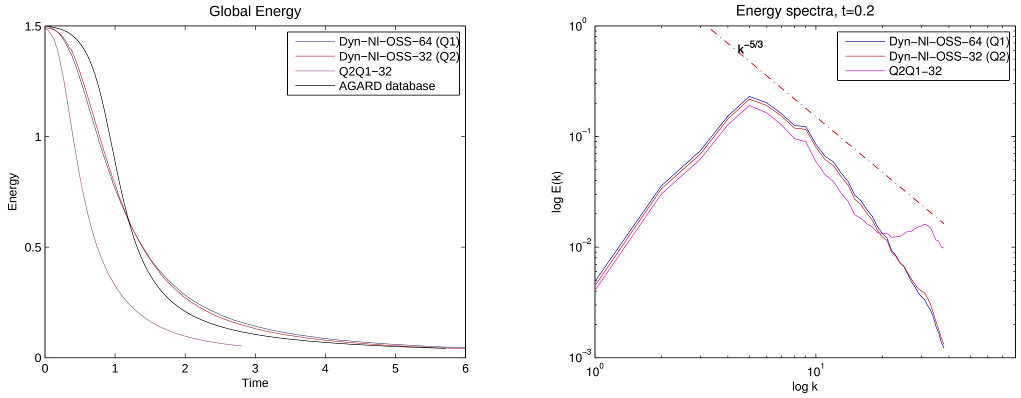
Comparison with a non-stabilized method

We use the classical static Smagorinsky model, consisting in adding a turbulent viscosity ν_t that depends on the velocity gradient and the characteristic element length h. This additional viscosity also acts as stabilization of convection, as usual in standard LES simulations. Then, we have to solve the standard Galerkin problem using Taylor-Hood Q_2/Q_1 elements and introducing a modified viscosity defined as

$$\nu = \nu_l + \nu_t, \tag{4.38}$$

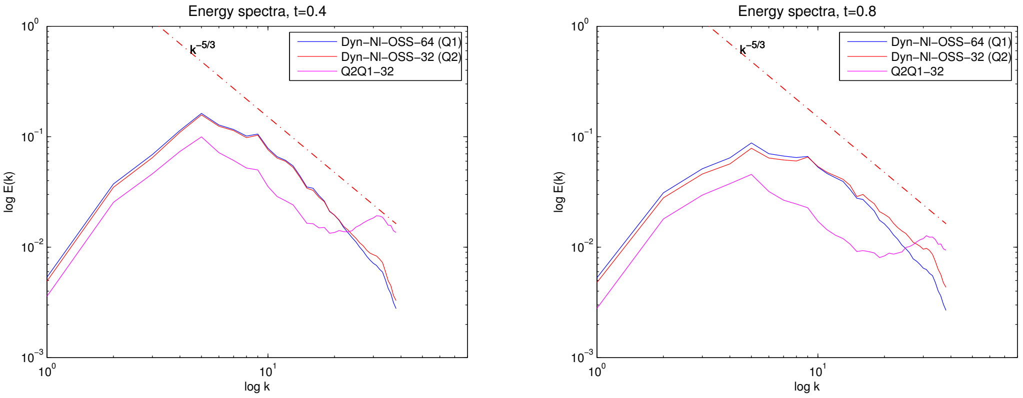
where ν_l is the real flow viscosity and $\nu_t = (C_s h)^2 |\nabla^s \mathbf{u}|$. C_s is the Smagorinsky constant, which we set equal to 0.15.

We can see in Fig. 4.12(a) that the total kinetic energy is decaying faster for the non-stabilized method than for the OSS method. This behavior is directly related to the shape of the energy spectra in Figs. 4.12(b)-4.12(d), where we can see that the Smagorinsky method presents lower values of energy at $t = 0.4$. It is important to point out the pile-up that appears in the Smagorinsky spectra, denoting that small scales are not dissipating energy properly.



(a) Energy evolution

(b) Energy spectra at $t = 0.2$



(c) Energy spectra at $t = 0.4$

(d) Energy spectra at $t = 0.8$

FIGURE 4.12: Total kinetic energy evolution and energy spectra using OSS and non-stabilized method with an inf-sup stable Q_2/Q_1 element

4.5.2 Taylor-Green Vortex

This problem aims to show, in a relatively simple flow, the basic turbulence decay mechanisms like the turbulent energy cascade, the production of small eddies and the enhancement of dissipation by the stretching of vortex lines. We refer to Section 3.2.3 for a deeper description of this problem.

The computational domain is the unit cube with periodical boundary conditions. The initial analytical condition is defined in the physical space (see, e.g., [80]), and given by

$$\begin{aligned} u_x &= u_0 \cos(x)\sin(y)\sin(z), \\ u_y &= -u_0 \sin(x)\cos(y)\sin(z), \\ u_z &= 0, \\ p &= p_0 + \frac{1}{16}\left(\cos(2x)+\cos(2y)\right)\left(\cos(2z)+2\right), \end{aligned} \tag{4.39}$$

with

$$u_0 = \frac{2}{\sqrt{3}} \sin\left(\gamma + \frac{2\pi}{3}\right).$$

We choose $\gamma = 0$, which gives the mean initial velocity $u_0 = 1$. The pressure constant parameter p_0 is chosen equal to zero.

Setting

We solve the TGV problem using a Reynolds number $\mathrm{Re} = 1600$. The most common Reynolds numbers available in the literature are $\mathrm{Re} = 800$, $\mathrm{Re} = 1600$ and $\mathrm{Re} = 3000$ (see, e.g., [2, 71, 80, 109]). We use the same VMS methods as for the DHIT problem defined in Section 4.5.1 to solve this test, namely the ASGS and OSS methods, both with linear and nonlinear definitions of the convective term and static or dynamic tracking in time of the subscales, as it is summarized in Table 4.1. The stabilization parameters for each method are the same as those chosen for the DHIT test, see Subsection 4.5.1, and discussed in Section 4.5.4.

Initially we consider a mesh of 32^3 hexahedral linear elements (Q_1), but we will redefine this discretization to analyze the method performance when we refine the mesh, decreasing the element size h or increasing the degree of the interpolation polynomial p. It implies to solve the problem on meshes with 64^3 and 128^3 linear (Q_1), quadratic (Q_2) or cubic (Q_3) hexahedral elements. We also use a $20^3(Q_3)$ discretization to compare against other authors results.

Vorticity

The TGV test is characterized by its laminar evolution at the initial time steps, when the flow is strongly anisotropic due to the structured large-scale vortices directly related to the initial condition. If the Reynolds number is large enough, the vortex-stretching process, which activates the energy cascade effect, transfers energy from large to small-scales and the flow becomes unstable and turbulent. According to Brachet *et al.* [32], the flow becomes nearly isotropic for $\mathrm{Re} \geq 1000$.

In Fig. 4.13 we present some vorticity isosurface images showing this process for a 128^3 linear hexahedral elements mesh, for the dynamic and nonlinear OSS method.

Note that the initial condition (Fig. 4.13(a)) consists in eight vortices with the same scale corresponding to the eight Fourier modes located at $\mathbf{k} = (\pm 1, \pm 1, \pm 1)$, as it has been pointed out previously.

Comparison between VMS methods

In order to compare the different VMS methods defined previously and to test their performance as LES models we solve the TGV test on a 32^3 and 64^3 linear elements mesh with a Reynolds number $\mathrm{Re} = 1600$.

We want to show the amount of numerical dissipation, the energy cascade in the spectra and the enstrophy evolution (compared to DNS) in all cases. We compare first the kinetic energy evolution with the kinetic energy evolution obtained by Brachet *et al.* [32], (Fig. 4.14(a)). We also present the energy spectra at $t = 9$, when the flow is supposed to be nearly isotropic at large wave numbers, (Fig. 4.14(b)).

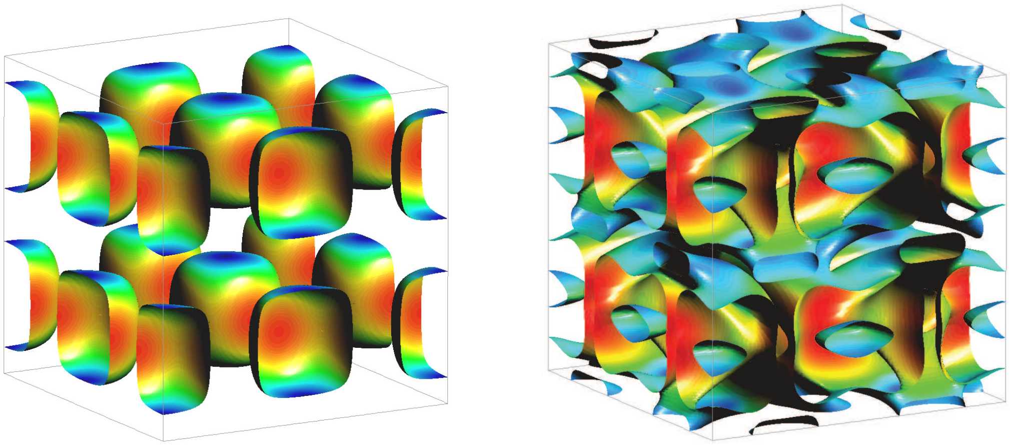
In Fig. 4.14(a), we can see that for a 32^3 trilinear hexahedral elements mesh all methods show a premature decay of energy. We recognize the same behavior observed for the same mesh in the DHIT test, see Subsection 4.5.1. For this mesh, it is clear that the methods are not able to simulate properly the transition to turbulence. The energy spectra at $t = 9.0$ shows us that the flow is isotropic at large wave numbers since it is decaying following the $k^{-5/3}$ Kolmogorov law.

As in the DHIT test, the cases with nonlinear and dynamic definitions of the subscales, using either ASGS or OSS methods, seem to be slightly less dissipative. Furthermore, OSS is a little bit less dissipative than ASGS, but the differences are not important.

As for the DHIT test, the results obtained using the different methods listed in Table 4.1 are very similar for these coarser discretizations. The only point that is worth to note is that the linear and static ASGS case (Sta-Lin-ASGS) and the dynamic and linear ASGS case (Dyn-Lin-ASGS) diverge at some time step before $t = 9$. Anyway, all the methods converge as $h \to 0$ and the accuracy depends much more on the mesh size than on the choice of the method. In turn, similar trends for the computational cost analyzed in the previous section have been observed.

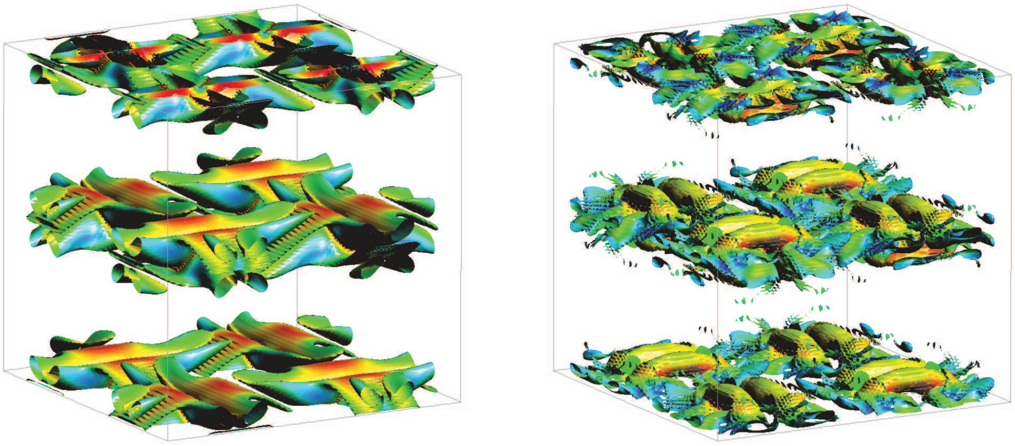
h-p refinement

As in the DHIT problem, we perform a refinement study reducing h and/or increasing p using Dyn-Nl-OSS. The global energy evolution and the energy spectra are shown in Fig. 4.15. Fig. 4.15(a) displays the total kinetic energy evolution compared with the DNS [32]. The results show that all cases, excluding the 32^3 and 64^3 linear hexahedral mesh, follow almost perfectly the line defined by the DNS result points. On the other hand, Fig. 4.15(b) displays the energy spectra at $t = 9$, when the dissipation is maximum and the flow is evolving to turbulence. We compare the energy spectra obtained solving all the cases considered before with the DNS computed by [80], using the same Reynolds number ($\mathrm{Re} = 1600$) at the same time.



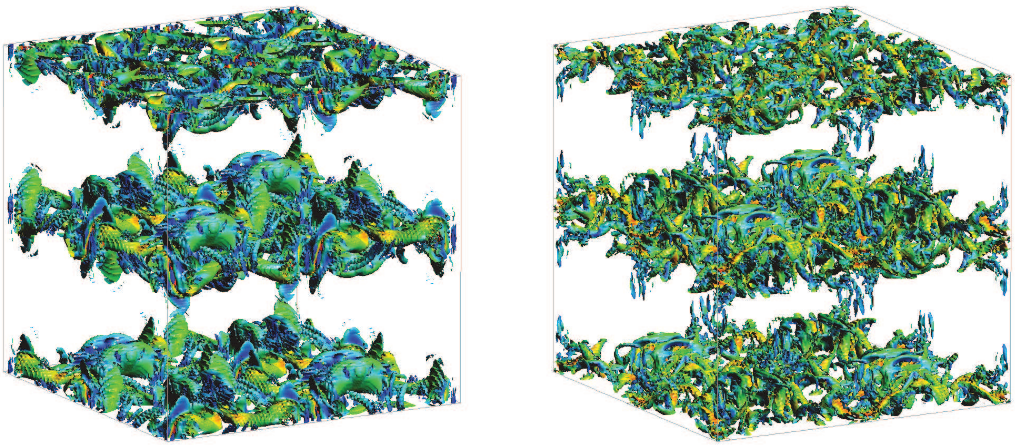
(a) Isosurface for $|\omega| = 1.0$ at $t = 0.0$

(b) Isosurface for $|\omega| = 1.0$ at $t = 2.0$



(c) Isosurface for $|\omega| = 2.5$ at $t = 4.1$

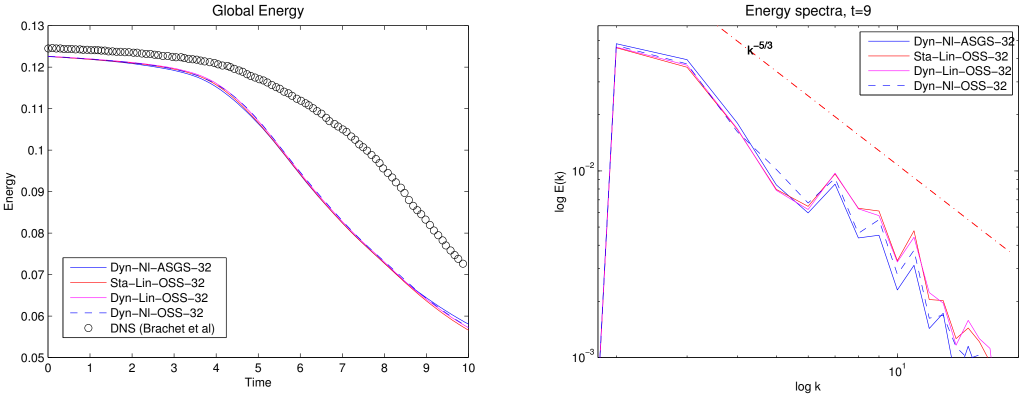
(d) Isosurface for $|\omega| = 5.0$ at $t = 6.1$



(e) Isosurface for $|\omega| = 8.0$ at $t = 8.2$

(f) Isosurface for $|\omega| = 9.0$ at $t = 10.2$

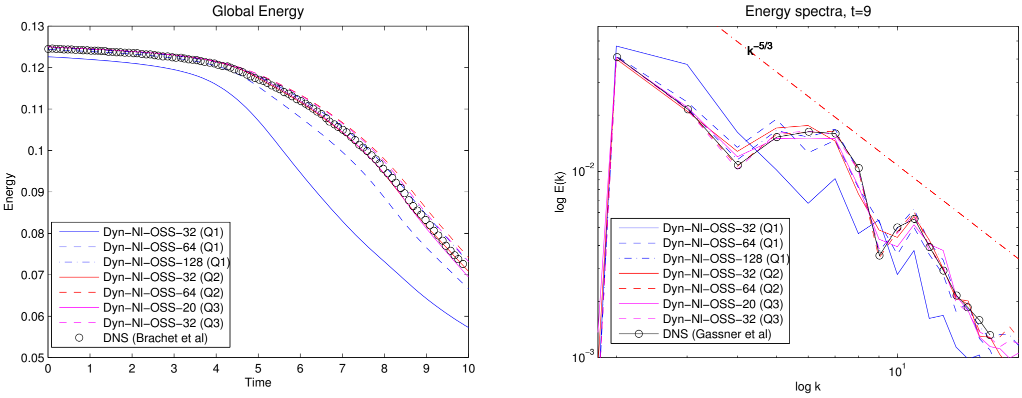
FIGURE 4.13: Vorticity isosurfaces with velocity contour at different time steps.



(a) Total kinetic energy evolution

(b) Energy spectra at $t = 9.0$

FIGURE 4.14: Total kinetic energy evolution and energy spectra



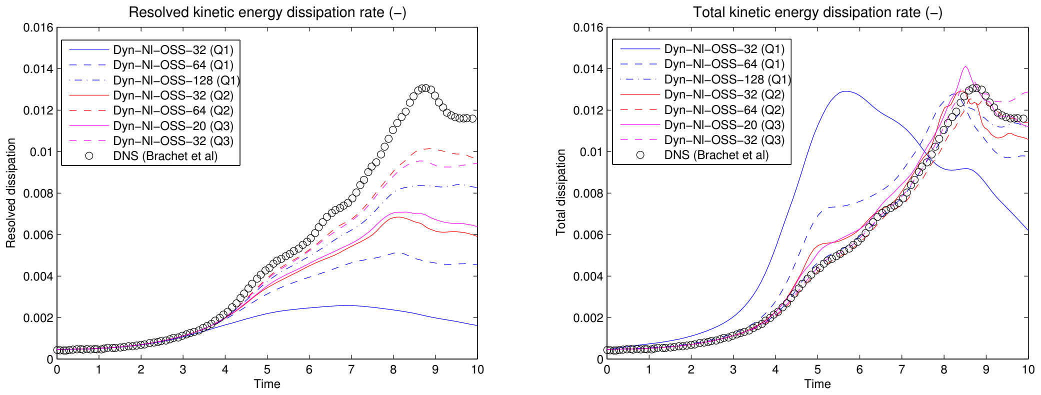
(a) Total kinetic energy evolution

(b) Energy spectra at $t = 9$

FIGURE 4.15: Total kinetic energy evolution and energy spectra for the $h - p$ refinement cases.

In Fig. 4.16 we show the dissipation rate of the problem, compared to the DNS results. The dissipation rate is directly related to the enstrophy of the problem, $\epsilon = 2\nu\left(\frac{1}{2}\left\langle|\omega|^2\right\rangle\right)$, where $|\omega|$ is the modulus of the vorticity. At the continuous level, it determines the kinetic energy decay which, at the discrete level, is also influenced by the numerical dissipation (see equation (4.17)). When an explicit model is used, the dissipation introduced by the subgrid model also needs to be included. The FE viscous dissipation $\nu\|\nabla\mathbf{u}_h\|^2$, is shown in Fig. 4.16(a) whereas the total dissipation rate $\nu\|\nabla\mathbf{u}_h\|^2 + \varepsilon_h$ defined by equation (4.17) is shown in Fig. 4.16(b).

As in the DHIT problem, the results obtained using the coarser $32^3(Q_1)$ and $64^3(Q_1)$ meshes are not accurate, the FE viscous dissipation being far from the exact viscous dissipation, as shown in Fig. 4.16(a). The total dissipation introduced by the method is too large and, especially for the $32^3(Q_1)$, peaked at earlier times, i.e., the energy decays faster and earlier than it should (see Fig. 4.16(b)). When finer resolutions are used, the flows dynamics are much better predicted. Even when the resolution is not enough



(a) Resolved (b) Total

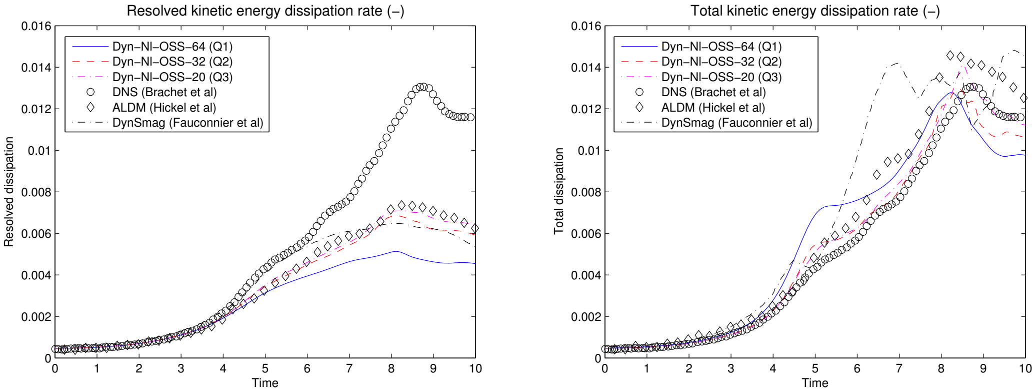
FIGURE 4.16: Dissipation rate evolution for the h-p refinement cases.

to completely capture the viscous dissipation, the total dissipation compares very well with the exact one, as shown in Fig. 4.16. This is a *clear illustration of the very good performance of the method, which adds the right amount of dissipation when the gradients are not captured by the resolution.*

Comparison with a non-stabilized method

All the results presented up to this point have been computed using a VMS method, either ASGS or OSS. But, what would be the result using other methods? Are the methods presented here, comparable to classical LES methods? Which methods perform better? To answer all these questions, we compare the results obtained here against those obtained using the dynamic Smagorinsky model [71] and the adaptive local deconvolution method [96] specifically designed as an implicit LES model. The former has been obtained with a filter of size $2\pi/64$ and spectral resolution up to $2\pi/256$, thus not having numerical but only modeling error. In turn, the latter has been obtained using a 64^3 grid without explicit subgrid model, an explicit third-order Runge-Kutta scheme for the time discretization, a fourth order spatial approximation of the symmetric terms and its particular approximation of the convective term which is based on the (forth order) five-point central stencils approximation of the convective term [96]. To make the comparison as fair as possible we select those combinations of h and p that result in a similar number of degrees of freedom, which are $64^3(Q_1)$ (second order), $32^3(Q_2)$ (third order) and $20(Q3)$ (fourth order) meshes (the last one having actually a bit less degrees of freedom).

The FE viscous dissipation is shown in Fig. 4.17(a) compared to the resolved dissipation obtained using the dynamic Smagorinsky model [71] and the "molecular dissipation" of [96] (the one computed using the molecular viscosity and the approximated solution, equivalent to our FE viscous dissipation but in the finite volume context). The total dissipations of the three methods are compared in Fig. 4.17(b).



(a) Resolved (b) Total

FIGURE 4.17: Dissipation rate evolution compared to the dynamic Smagorinsky [71] and ALDM models [96].

It can be observed in Fig. 4.17(a) that all the methods produce similar results, The dynamic Smagorinsky is more accurate in predicting the resolved dissipation at earlier times (up to $t \approx 6$) but less accurate at later times (see Fig. 4.17(a)). We plot the total dissipation in Fig. 4.17(b). We can see that the excellent job our implicit LES model does when the $20(Q3)$ mesh is used, which would result in an excellent prediction of the resolved kinetic energy decay (which is not available in [96]).

4.5.3 Turbulent channel flow

After studying the performance of VMS in the LES of homogeneous flows we turn our attention to wall-bounded turbulent flow and present results of fully developed turbulent flow in a channel.

This test consists of a fluid that flows between two parallel walls driven by an imposed pressure gradient which is defined by the Reynolds number based on the wall shear velocity, Re_τ. In the important amount of literature devoted to this problem, the usual Reynolds numbers are: $\mathrm{Re}_\tau = 590$, $\mathrm{Re}_\tau = 395$ and $\mathrm{Re}_\tau = 180$ (see [23, 40, 78, 82, 87, 105, 113, 121, 136, 140]). We will restrict our attention to $\mathrm{Re}_\tau = 180$ and $\mathrm{Re}_\tau = 395$. See Section 3.3.2 for an extended description of this test.

Setting

We solve the problem using the coarsest mesh from previous tests, 32^3 linear hexahedral (Q_1) elements. The refinement in the wall-normal direction follows a hyperbolic function, also used in [40, 78, 82, 87, 136], defined as

$$y_i = \frac{\tanh\left(\gamma\left(\frac{2i}{np_y} - 1\right)\right)}{\tanh(\gamma)},$$

where $i = 1, ..., np_y$ with np_y the total amount of nodes in the wall-normal direction. Here, γ is chosen to be equal to 2.75 for both $\mathrm{Re}_\tau = 180$ and $\mathrm{Re}_\tau = 395$. We refer the reader to [8] for a complete study of the influence of the discretization in the results of the TCF.

As it has been said above, we solve the problem using two different friction Reynolds numbers, $\mathrm{Re}_\tau = 180$ and $\mathrm{Re}_\tau = 395$. We compare our results against those obtained by DNS in [140, 121] and we choose our parameters accordingly. We take the bulk mean velocity and the half channel height equal to one, $\bar{U} = 1$ and $\delta = 1$. The viscosity is computed from the estimated Reynolds number based on the bulk mean velocity Re. Then, from the friction Reynolds number Re_τ we compute the friction velocity (u_τ), the wall shear stress (τ_w) and a driving force equivalent to a pressure gradient (f_x), given by [150]:

$$u_\tau = \frac{\nu \mathrm{Re}_\tau}{\delta}, \qquad \tau_w = \rho u_\tau^2, \qquad f_x = \frac{\tau_w}{\delta}.$$

We use the Crank-Nicolson time integration scheme with a constant time step. Ham *et al.* test in [93] the influence of the time step for a fully implicit Finite Difference midpoint method, equivalent to Crank-Nicolson, on the statistics of a TCF DNS. They found little variation in statistical turbulence quantities up to $\delta t^+ = 1.6$. Following Gravemeier *et al.* [82], we define a time step in wall units $\delta t^+ = \frac{\delta t u_\tau^2}{\nu} \approx 0.69$, which, according to [93], should not affect the turbulent quantity statistics. The same authors performed 25000 time steps in order to allow the flow to develop and they collected the statistics during another 5000 time steps. A total averaging time about $500\delta/U_0$ is used in [47] once the statistically stable regime is achieved.

In Table 4.3 we present the value of the different parameters defined above for the two different friction Reynolds numbers. For the initial condition we impose a parabolic profile obtained solving the stationary Stokes problem with the driving force and viscosity defined above. Additionally, with the aim to achieve a fully developed flow earlier, we introduce a perturbation with a maximum value of 10% the bulk velocity.

Re_τ	180	395
ν	$3.5714 \cdot 10^{-4}$	$1.4545 \cdot 10^{-4}$
u_τ	$6.4286 \cdot 10^{-2}$	$5.7455 \cdot 10^{-2}$
τ_w	$4.1327 \cdot 10^{-3}$	$3.3010 \cdot 10^{-3}$
f_x	$4.1327 \cdot 10^{-3}$	$3.3010 \cdot 10^{-3}$
δt	0.06	0.03

TABLE 4.3: Test parameters for the different friction Reynolds number.

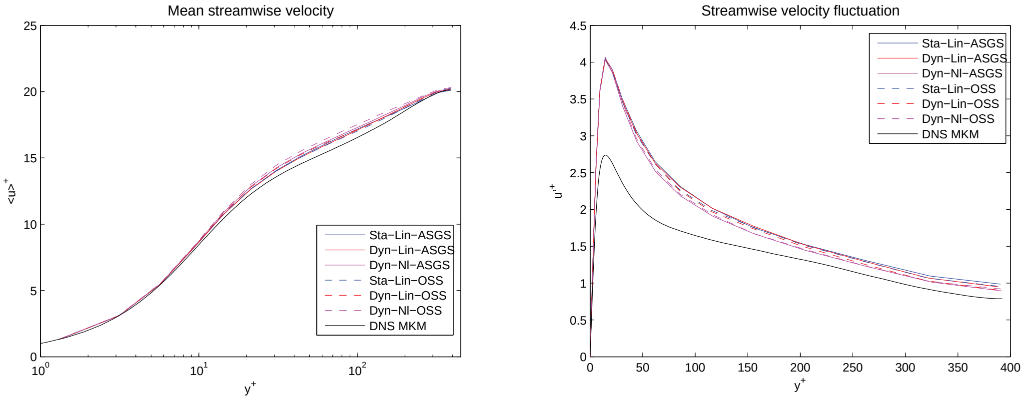
Our purpose is to check the VMS methods defined in Subsection 4.2.2 for a wall-bounded flow. Following the computations performed for the previous tests, we solve the problem using the same cases defined in Table 4.1 and the numerical parameters $\tau_c = 0$ and τ_m are defined in the same way, now with the algorithmic constants $c_1 = 12$

and $c_2 = 8$ (see Subsection 4.5.4) and the characteristic length, h, is chosen to be the minimum element length.

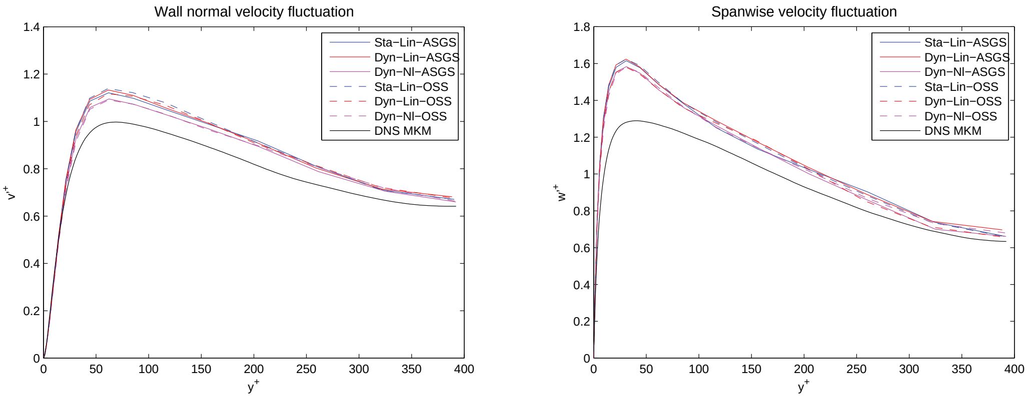
Velocity profiles

We first present the mean stream-wise velocity profile scaled by the wall shear stress velocity, $\langle u\rangle^+ = \frac{\langle u\rangle}{u_\tau}$ for all cases defined in Table 4.1, where $\langle\cdot\rangle$ denotes the mean value in stream-wise and span-wise direction and in time, as a function of $y^+ = \frac{yu_\tau}{\nu}$.

In Fig. 4.18(a) we show the mean stream-wise velocity normalized by the wall-shear velocity, u_τ, obtained for all cases considered in Table 4.1 in a 32^3 linear elements mesh for the $\mathrm{Re}_\tau = 395$ case. We compare the results with the DNS one obtained in [140]. We can observe in Fig. 4.18(a) that all methods perform quite similar and are very close to the DNS result. Fig. 4.18 also depicts the stream-wise, span-wise and wall-normal root mean square (rms) velocity fluctuation components normalized by the wall-shear stress velocity.



(a) Mean stream-wise velocity (b) Rms stream-wise velocity fluctuation



(c) Rms wall-normal velocity fluctuation (d) Rms span-wise velocity fluctuation

FIGURE 4.18: Mean stream-wise velocity and rms velocity fluctuations for $\mathrm{Re}_\tau = 395$ case using a 32^3 $Q1$ mesh.

Reynolds shear stress

Another turbulent quantity widely used in the TCF test is the Reynolds shear stress. At the continuous level the Reynolds shear stress is defined as

$$R_{xy} = -\left\langle u'v' \right\rangle + \nu \frac{\partial \left\langle u \right\rangle}{\partial y}, \tag{4.40}$$

being u and v the velocity in the streamwise direction and wall-normal direction, respectively, and the prime denoting the fluctuations, i.e., the variable minus the mean.

It can be seen that for the discrete equation (4.12), one can obtain the Reynolds shear stress defined as follows:

$$R_{xy} = -\left\langle a_x' a_y' \right\rangle + \nu \frac{\partial \left\langle u_h \right\rangle}{\partial y} = -\underbrace{\left\langle u_h' v_h' \right\rangle}_{I} - \underbrace{\left\langle u_h' \tilde{v}' \right\rangle - \left\langle \tilde{u}' v_h' \right\rangle - \left\langle \tilde{u}' \tilde{v}' \right\rangle}_{II} + \underbrace{\nu \frac{\partial \left\langle u_h \right\rangle}{\partial y}}_{III}. \tag{4.41}$$

being a_i the i-th component of the advection velocity. In (4.41) we have used the nonlinear definition of the advection velocity defined in (2.49).

The first term on the second part of (4.41) (term I) is the contribution of the resolved scales (FE component) to the cross term $\left\langle a_x' a_y' \right\rangle$. Term II denotes the contribution of the subgrid scales and their interaction with the FE components, that is, the unresolved part of the equation. Finally, term III accounts for the viscous portion of the Reynolds shear stress. Note that the derivatives of the approximated subscales are not computable, since these approximated subscales are discontinuous and have been designed to approximate the effect of the exact subscales on the FE scales element-wise.

For a fully developed and statistically stable turbulent flow, the Reynolds shear stress along the wall-normal direction has a linear shape (see [121]). Normalized by the viscous term III value at the wall, the total Reynolds shear stress in terms of y/δ should have the following expression: $R_{xy}(y/\delta) = (-y/\delta)$. Fig. 4.19 depicts the absolute value of the Reynolds shear stress along the upper half channel ($y > 0$), with the different terms appearing in (4.41) and compared with the DNS in [140], for the Dyn-Nl-OSS case with $\mathrm{Re}_\tau = 395$. The computed results are almost identical to the DNS ones. It has to be noted that the computed results are evaluated at the integration points due to the presence of the derivative in the Reynolds shear stress, which using linear FEs is constant at each element. Then, using two integration points per direction for the numerical integration, term III will be constant for those two integration points being in the same element. This behavior is observed in Fig. 4.19, where the viscous term is pairwise constant. This last fact also affects the total Reynolds shear stress. Since the resolved term has different values at each element Gauss point, the sum of terms I and III results in an oscillatory shape near the wall, where the viscous term is more relevant. It is also seen that the unresolved term II does not contribute to the Reynolds shear stress, which is a good property of the tested VMS methods. The results for the remaining cases in Table 4.1 are similar to those presented in Fig. 4.19 and have not been reported.

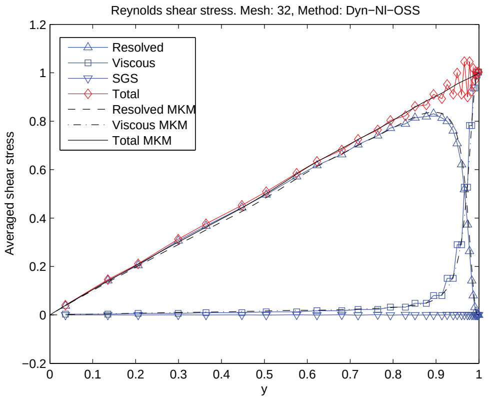


FIGURE 4.19: Reynolds stress of the Dyn-Nl-OSS case.

4.5.4 Sensitivity with respect to the stabilization parameters

All the VMS models considered herein depend on the stabilization parameters τ_m and τ_c, which contain constants c_1 and c_2 whose value is chosen from numerical experiments. However, we can infer from (4.20) or (4.21) how this dependency will be. As mentioned before, the last two terms in (4.21) are dissipative and therefore, increasing τ_m and/or τ_c we obtain a more dissipative method. From (4.15)-(4.16), increasing τ_m results in a reduction of τ_c. More precisely

$$\tau_c = \nu + \frac{c_2}{c_1}|\mathbf{a}|h \tag{4.42}$$

from where we see that increasing c_1 reduces both τ_m and τ_c but increasing c_2 reduces τ_m but increases τ_c. On the other hand, only the fourth term in (4.21) is essential to control $\tau_m \left\| \mathcal{P}\left(\mathbf{a} \cdot \nabla \mathbf{u}_h + \nabla p_h\right) \right\|^2$ and it is possible to choose $\tau_c = 0$.

The results presented above have been obtained using different settings of the numerical stabilization parameters τ_m and τ_c. In DHIT and TGV tests, we take the algorithmic constants $c_1 = 12$ and $c_2 = 2$ for τ_m and we set $\tau_c = 0$, while for the TCF test we have used $c_1 = 12$ and $c_2 = 8$ for τ_m and also $\tau_c = 0$. In this section we analyze the influence of these parameters on the numerical results and justify our choice of the constants for the large eddy simulation of turbulent flows.

We have performed a sensitivity analysis of the VMS schemes with respect to the value of c_1 and c_2. To see the effect of such algorithmic constants on τ_m and τ_c independently, we define a new constant c_c which allows us to redefine (4.42) as

$$\tau_c = c_c \left(\nu + \frac{c_2}{c_1}|\mathbf{a}|h \right) \tag{4.43}$$

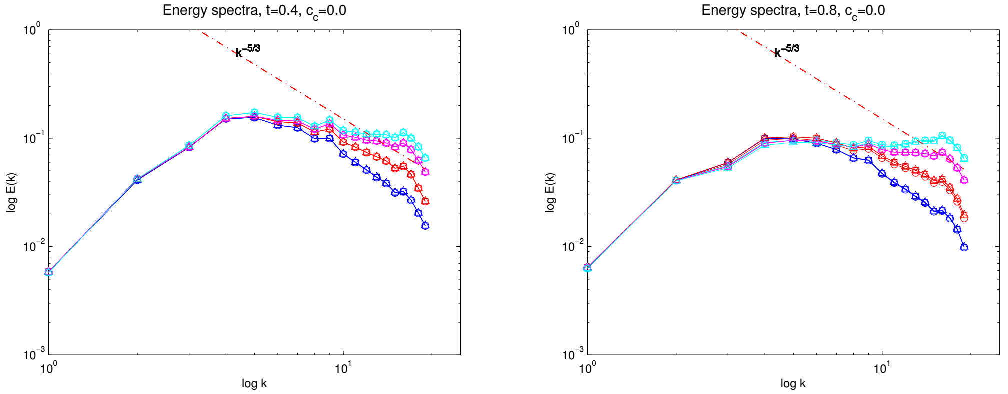
These experiments have been done for the DHIT test using the Dyn-Nl-OSS case in a 32^3 $Q1$ mesh and the results are depicted in Fig. 4.20. They show important changes in the dissipation the VMS methods introduce when constants are changed. It is known that the decay rate of kinetic energy in isotropic turbulence is driven by large scales (of the order of the integral scale) (see, e.g., [59]). As we have observed, the subgrid model has only influence when a very coarse grid is used.

In particular, for high Reynolds number problems, the constant c_1 does not have so much influence on τ_m, but it does on τ_c. With respect to c_2, we observe that it influences the energy dissipation of the method, which is increased when the value of this constant is decreased. When τ_c is activated $(c_c = 1)$, we observe a growth of the energy dissipation when the coefficient c_2/c_1 increases. This behavior is what we are expecting since the method becomes more diffusive when τ_c is increased due to the last term in (4.21).

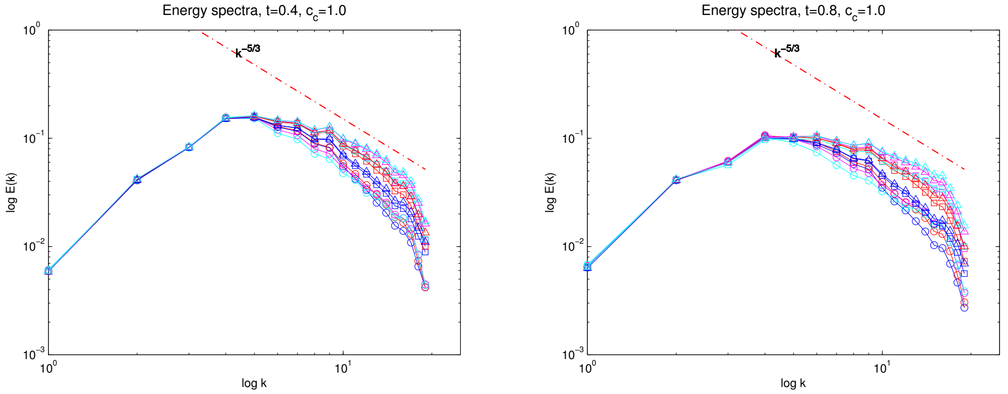
Concerning the energy spectra, it is also shown in Fig. 4.20(a) and Fig. 4.20(b) that the only constant that influences the result when $\tau_c = 0$ is c_2. In these figures we can see that when we increase c_2 the method is less dissipative, resulting in an inappropriate slope of the energy spectra. We can observe that with $c_2 = 2$ the decay of the energy behaves correctly, keeping the $k^{-5/3}$ law. For the largest values of c_2 the energy at small scales is not properly dissipated. Note that for $c_2 = 2$ the slope of the energy spectrum is kept almost constant along the time, which does not happen in the other cases. When we activate τ_c (see Figs. 4.20(c) and 4.20(d)) we are introducing additional dissipation into the system that eliminates the pile up of the energy spectra for all the cases, but generally results in steeper slopes. Here we also have to note that the energy spectra slope is time dependent for all cases except for $c_2 = 2$. This analysis led us to choose $c_1 = 12$ and $c_2 = 2$ for τ_m and set $\tau_c = 0$ for homogeneous turbulence, i.e., DHIT and TGV tests.

In order to go in depth on the effect of the algorithmic constants c_1 and c_2 and the stabilization parameter τ_c of the incompressibility equation, we compare the results for the TCF problem with a friction Reynolds number $\mathrm{Re}_\tau = 180$ using the same choice made for homogeneous turbulence ($c_1 = 12$, $c_2 = 2$ and $\tau_c = 0$) against the setting of the incompressible case in [8] ($c_1 = 12$, $c_2 = 2$ and τ_c as in (4.16)) and a less dissipative setting with $c_1 = 12$, $c_2 = 8$ and $\tau_c = 0$. These tests have been done using the Dyn-Nl-OSS case in a 32^3 Q_1 mesh.

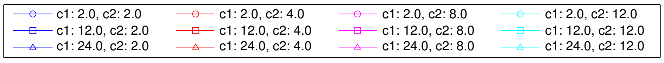
In Fig. 4.21(a) the mean velocity in the streamwise direction is shown. As in the case of homogeneous turbulence, some differences between the three cases can be observed, the choice used in section 4.5.3 being the most accurate one. The effect of the algorithmic constant c_2 and the stabilization parameter τ_c in the problem solution can be clearly observed, i.e., the less dissipative choice gives the best results. Figs. 4.21(b), 4.21(c) and 4.21(d) depict the rms velocity fluctuations in all directions. The fluctuations in the streamwise direction are better predicted using ($c_1 = 12$, $c_2 = 8$ and $\tau_c = 0$) but the span-wise and wall-normal directions are not.



(a) Energy spectra at $t = 0.4$ with $c_c = 0$ (b) Energy spectra at $t = 0.8$ with $c_c = 0$



(c) Energy spectra at $t = 0.4$ with $c_c = 1$ (d) Energy spectra at $t = 0.8$ with $c_c = 1$



(e) Legend

FIGURE 4.20: Comparison of energy spectra for different c_1, c_2 and c_c in the DHIT test.

4.5.5 Behavior in the small time step limit

Small time step instabilities for VMS LES simulations of turbulent flows have been reported in [98, 78]. In these references, the VMS models differ from the ones in this work. Instead of the definition of τ_m in (4.15), a time step dependent stabilization parameter τ_m

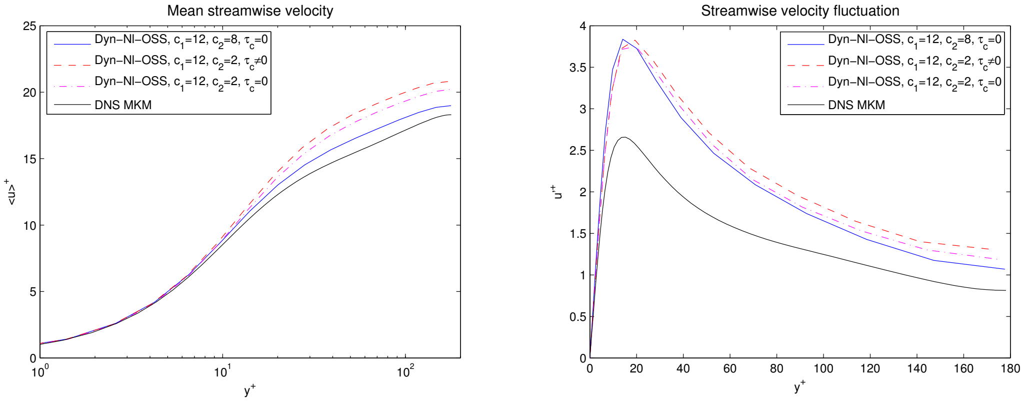
$$\tau_m = \left(\frac{1}{\delta t} + \frac{c_1 \nu}{h^2} + \frac{c_2 |\mathbf{a}|}{h} \right)^{-1},$$

is considered in all cases.[3] The plain introduction of a time step dependency in τ_m faces serious difficulties:

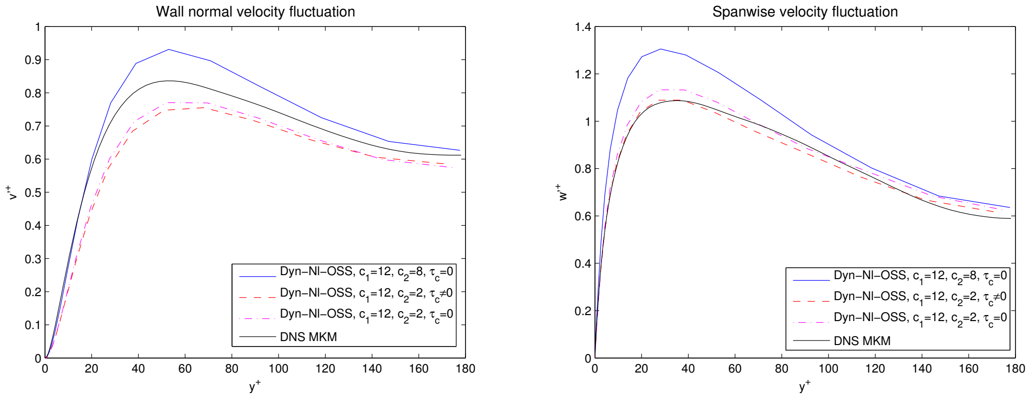
- The method becomes unstable in the small time step limit since it converges to the unstable Galerkin formulation.

---

[3] The parameter $\tau_{m,t}$ for the dynamic subscales model also scales with δt, as discussed in Section (4.4). *However*, this dependence comes from a consistent time integration of the subscale time derivative (see also [55, Section 3.2]).



(a) Mean streamwise velocity (b) Rms streamwise velocity fluctuation



(c) Rms wall-normal velocity fluctuation (d) Rms span-wise velocity fluctuation

FIGURE 4.21: Comparison of mean streamwise velocity and rms velocity fluctuations for $\mathrm{Re}_\tau = 180$ case using a 32^3 $Q1$ mesh.

- If τ_c is computed from (4.16) (as it is usually done, see, e.g., [23, 98, 78, 82]), $\tau_m \sim \delta t$ and $\tau_c \sim \delta t^{-1}$ in the small time step limit. If this approach is followed , the essential numerical dissipation given by the fifth term in (4.20) is reduced as $\delta t \to 0$, whereas the numerical dissipation introduced by the last term in the left hand side (a incompressibility penalty term) of (4.20) is increased. It has a compensating effect in practice, but the penalty term does not properly act as a turbulence model.

Let us perform a test to study the small time step behavior of the VMS methods presented in Section 4.2, using the skew-symmetric *type 1* form of the convective term, as in previous numerical experiments. We also include a combination we do not advocate here, static subscales and nonlinear splitting, an approach followed in [42, 23, 98, 78, 82]. The behavior of all the methods for the TCF test with $\delta t = 0.002$ is summarized in Table 4.4, where YES means that the simulation was successful, NO means that the simulation diverged and $\delta t \downarrow$ means that the simulation was successful only when the adaptive time step strategy described in Section 4.5.1 was used.

It is important to note that the static and nonlinear ASGS formulation used in [23, 98, 78, 82] with the convective term *type 2* becomes unstable after some time, as also reported in these works, even for the time step size defined in section 4.5.3. However, using the the skew-symmetric *type 1* form of the convective term, which exactly conserves energy, the simulation ended successfully for the time step defined in section 4.5.3, but failed to converge with the small one. This result is a numerical evidence of the fact that the use of convective terms without the skew-symmetric property produce energy (see also Section 4.5.1) that can make simulations unstable. Further, *these results evidence once again that it is a good choice to stick to provably unconditionally stable formulations, i.e., the dynamic formulations and/or orthogonal subscales formulations with a skew-symmetric convective term*. Similar results have been reported in the finite difference context in [179], where it is shown that stable simulations of the TCF can be performed using an energy-preserving skew-symmetric formulation.

Method	ASGS				OSS			
Tracking	Static		Dynamic		Static		Dynamic	
Advection	Linear	Nonlinear	Linear	Nonlinear	Linear	Nonlinear	Linear	Nonlinear
Converged	Yes	No	Yes	Yes	$\delta t \downarrow$	$\delta t \downarrow$	Yes	Yes

TABLE 4.4: Small time step convergence analysis.

To the best of our knowledge, the stability (or instability) of dynamic ASGS methods has not been proved. In our numerical experiments the static and dynamic linear versions fail to converge in some problems (e.g DHIT) but we have not found these problems with the Dyn-Nl-ASGS method. Nevertheless, we have found an important increase in the computational cost when the time step is reduced. This behavior is explained in Fig. 4.22, where the number of solver iterations at the first time step is plotted against the time step size for the dynamic and nonlinear cases of ASGS and OSS methods, with 32^3 and 64^3 Q_1 mesh, for the DHIT test case. *The number of required solver iterations (and as a result the condition number of the system matrix) blows up exponentially for the ASGS method as we reduce the time step size, whereas it remains constant for the OSS method.* This important observation explains the computational cost trends observed in the previous section.

4.6 Conclusions

In this paper we have assessed the performance of the numerical formulations previously developed in our group [51, 55, 54, 151] for turbulent incompressible flow problems.

The methods proposed are different to those whose testing in turbulent regimes has been published before, the closest ones being those reported in [23, 78]. First, we consider orthogonal subscales formulations. Further, in [23] the ASGS method with

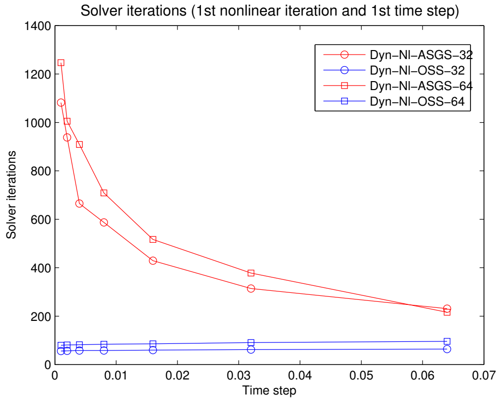


FIGURE 4.22: Solver iterations at the first time step for DHIT test.

quasi-static subscales is used (in the isogeometrical analysis context) but the time step dependency is included in the stabilization parameter (with the inconsistencies and problems discussed in section 4.5.5) and the nonlinear scale splitting is applied in the FE equation only (not in the subscale equation). Time dependent subscales are used in [78], but the authors consider a linear scale splitting. Furthermore, in both works $\tau_c \neq 0$.

First, we have discussed some theoretical aspects, such as the dissipative structure of the methods and the way energy is conserved, which we have numerically verified. Related to this point, we analyze the effect of using different skew-symmetric forms of the convective term, and its impact on energy conservation; if a skew symmetric form is not used, negative energy dissipation can be introduced to the scheme, which may be a source of instability.

However, the most important conclusions come from the different problems that we have solved numerically. Overall, OSS and ASGS yield similar results, all displaying the features of turbulent flows, reproducing appropriately global outputs such as energy spectra. The methods are stable and converge to reference solutions, both when the mesh is refined and when the polynomial order is increased.

On the other hand, we have thoroughly analyzed the effect of the algorithmic constants for isotropic turbulence and wall-bounded turbulent flows, and chosen them based on this sensitivity analysis. An important observation in this line is the fact that all the methods considered in this work are certainly sensitive to the algorithmic constants and they have to be properly chosen in order to simulate turbulent flows on coarse meshes. In fact, the differences in the numerical results are much more influenced by the algorithmic constants than by the choice of the VMS formulation itself. This strong influence seems to be a characteristic feature of turbulence, since in our experience it is not so important in laminar flows. VMS methods is something that needs further research.

Further, we have analyzed the effect of small time steps when the stabilization parameters depend on them.

Apart from the quality of the results, the OSS method with dynamic subscales is convenient in terms of numerical performance. It requires more nonlinear iterations than ASGS, but less iterations of the linear solver, altogether leading to lower computational cost. In both formulations, ASGS and OSS, the use of dynamic subscales has been found to be crucial for nonlinear convergence. In fact, in some cases quasi-static subscales failed to converge. We have explained these facts by plotting the number of solver iterations required to converge as we reduce the time step size, for a fixed mesh in space. The number of iterations (and as a result the condition number of the system matrix) blows up exponentially for ASGS whereas it remains bounded for OSS.

Chapter 5

Mixed FE methods with convection stabilization

5.1 Introduction

The VMS method for incompressible turbulent flows has been analyzed previously in Chapter 4. As stated there, the VMS method introduced by Hughes in [100, 101] is a framework to develop stable and accurate numerical approximations of partial differential equations, preventing numerical instabilities that arise when the standard Galerkin FE method is used.

We also recall here that the use of VMS method as an ILES method was firstly suggested in [104, 105, 51] and, since then, several VMS methods have been developed and used as ILES. We can distinguish between those that introduce a three scale decomposition into resolved large and small scales and unresolved scales [125, 113, 114, 41], with a Smagorinsky type model for the influence of unresolved scales onto the small resolved ones, and those that introduce a two scale decomposition into resolved and unresolved ones [23, 56] using a residual based or projection based model of the unresolved scales to account for their influence into the resolved ones.

In [50], a two scale VMS approach through OSS was firstly introduced. The main idea of the OSS method is to select the space of small scales orthogonal to the FE space, in contrast to the traditional choice of taking the subscales proportional to the residual, which is called ASGS in [50]. Apart from the choice of the space of subscales, their time dependency and the VMS splitting of nonlinear terms was studied in [55]. Several combinations of these modelling possibilities where exhaustively assessed for homogeneous and wall bounded turbulent flows in [56][1]. In that work, an explicit algorithm to compute the orthogonal projections was used and the projection of the whole residual was considered.

An alternative definition of the OSS method was proposed in [52] using a term-by-term stabilization that does not involve the full residual. A similar term-by-term stabilization approach was followed in [30], where the Local Projection Stabilization (LPS) method was introduced. This type of techniques are also known as symmetric

---

[1] Chapter 4 collects most of the work exposed in [56], which has been extended and adapted to fit in the argumentation of this thesis.

projection stabilization, and the key ingredient that leads to different schemes is the definition of such projection [30, 11]. One of the main interests of a term-by-term stabilization is that one can avoid the addition of the pressure gradient stabilization term when using *inf-sup* stable (ISS) velocity-pressure pairs. Another advantage of the *term-by-term* stabilization methods is that the projection can be easily treated as implicit, without having all the residual terms coupled. The use of equal-order or ISS pairs with the LPS method was assessed for laminar flows in [77], concluding that the grad-div stabilization term is more relevant than the convective stabilization term when ISS FEs are used. The same conclusion was pointed out in [114], where the turbulent channel flow is studied using a projection-based and a bubble-based FE VMS method introducing a Smagorinsky type model to stabilize convection.

The main goal of this chapter is to assess for the first time the accuracy and efficiency of convection-stabilized ISS schemes as ILES methods, where symmetric projection stabilization is used. In particular, we analyze the term-by-term OSS method with implicit treatment of the projection for turbulent flows. We also analyze the influence of the grad-div stabilization on the accuracy of the method. For ISS discretizations, the influence of this term on the mass conservation is well known [130] but it also influences the computational cost of the linear solvers [146, 95]. In this respect, we present a block preconditioning strategy that makes use of recursive block factorizations [18] to deal with the implicit projections and with the saddle point structure of the velocity-pressure coupling, which can also be applied to equal order interpolation with pressure stabilization. The comparison of the results is made with respect to those obtained using the residual-based ASGS for which we also use a block preconditioning strategy.

This chapter is organized as follows. The Navier-Stokes equations together with some notation used in the paper are stated in Section 5.2. The VMS framework is introduced in Section 5.3, which includes the final discrete formulation of the ASGS method, given in Section 5.3.1, the term-by-term OSS given in Section 5.3.2, the term-by-term OSS with ISS elements given in Section 5.3.3 and also a brief discussion of known properties of the grad-div stabilization given in Section 5.3.4. The recursive block iterative strategy proposed to solve the linear system of the monolithic problem is presented in Section 5.4. The numerical results are shown in Section 5.5, where two different turbulent tests are analyzed: the Taylor-Green Vortex flow in Section 5.5.2 and the Turbulent Channel Flow in Section 5.5.3. Finally, some conclusions are pointed out in Section 5.6.

5.2 Problem statement

For the sake of chapter completeness, let us retrieve the problem definition described in Chapter 2, albeit rather briefly.

Let Ω be a bounded domain of $\mathbb{R}^d$, where $d = 2, 3$ is the number of space dimensions, $\Gamma = \partial\Omega$ its boundary and $(0, T]$ the time interval. The strong form of the transient

Navier-Stokes problem consists in finding the velocity field $\mathbf{u}$ and the pressure field p such that

$$\partial_t \mathbf{u} - \nu\Delta\mathbf{u} + \mathbf{u}\cdot\nabla\mathbf{u} + \nabla p = \mathbf{f} \qquad \text{in } \Omega\times(0,T], \tag{5.1}$$

$$\nabla\cdot\mathbf{u} = 0 \qquad \text{in } \Omega\times(0,T], \tag{5.2}$$

with $\mathbf{f}$ the force vector and ν the kinematic viscosity. Equations (5.1) and (5.2) need to be supplied with appropriate boundary and initial conditions. The boundary Γ is divided into the Dirichlet (Γ_D) and the Neumann (Γ_N) parts such that $\Gamma_D \cup \Gamma_N = \Gamma$ and $\Gamma_D \cap \Gamma_N = \emptyset$. Then, the boundary and initial conditions can be written as

$$\mathbf{u} = \mathbf{u}_g \qquad \text{on } \Gamma_D\times(0,T], \tag{5.3}$$

$$(-p\cdot\mathbf{I} + \nu(\nabla\mathbf{u} + \nabla\mathbf{u}^T))\cdot\mathbf{n} = \mathbf{t}_N \qquad \text{on } \Gamma_N\times(0,T], \tag{5.4}$$

$$\mathbf{u}(\mathbf{x},0) = \mathbf{u}_0(\mathbf{x}) \qquad \text{in } \Omega\times\{0\}, \tag{5.5}$$

$\mathbf{n}$ being the unit outward vector normal to Γ and $\mathbf{u}_0(\mathbf{x})$ satisfying $\nabla\cdot\mathbf{u}_0 = 0$.

From equations (5.1)-(5.5) one can derive the weak form of the problem, which consists in finding $[\mathbf{u},p] \in \mathbf{L}^2(0,T;\mathcal{V}_g)\times L^1(0,T;\mathcal{Q}_0)$ such that

$$(\partial_t\mathbf{u},\mathbf{v}) + B(\mathbf{u},(\mathbf{u},p),(\mathbf{v},q)) = \langle\mathbf{f},\mathbf{v}\rangle \qquad \forall\mathbf{v}\in\mathcal{V}_0, \quad \forall q\in\mathcal{Q}_0, \tag{5.6}$$

satisfying the initial condition (5.5) in a weak sense. Here $\mathcal{V}_0 := \mathbf{H}_0^1(\Omega)$, $\mathcal{V}_g := \mathbf{H}_g^1(\Omega)$ and $\mathcal{Q}_0 := L^2(\Omega)/\mathbb{R}$ and the form $B(\mathbf{u},(\mathbf{u},p),(\mathbf{v},q))$ is defined as

$$B(\mathbf{u},(\mathbf{u},p),(\mathbf{v},q)) := \nu(\nabla\mathbf{u},\nabla\mathbf{v}) + b(\mathbf{u},\mathbf{u},\mathbf{v}) - (p,\nabla\cdot\mathbf{v}) + (q,\nabla\cdot\mathbf{u}) \tag{5.7}$$

with the trilinear form of the convective term $b(\mathbf{u},\mathbf{v},\mathbf{w})$ defined in its skew symmetric version

$$b(\mathbf{u},\mathbf{v},\mathbf{w}) = \frac{1}{2}(\mathbf{u}\cdot\nabla\mathbf{v},\mathbf{w}) - \frac{1}{2}(\mathbf{v},\mathbf{u}\cdot\nabla\mathbf{w}) + \frac{1}{2}(\mathbf{v},(\mathbf{u}\cdot\mathbf{n})\mathbf{w})_{\Gamma_N}. \tag{5.8}$$

5.3 VMS framework

Let us consider a FE partition $\mathcal{T}_h$ of the domain Ω from which we can construct conforming finite dimensional spaces for the velocity $\mathcal{V}_{0,h}\subset\mathcal{V}_0$, $\mathcal{V}_{g,h}\subset\mathcal{V}_g$, and for the pressure $\mathcal{Q}_{0,h}\subset\mathcal{Q}_0$. The Galerkin FE approximation of the problem (5.6) consists in finding $[\mathbf{u}_h,p_h]\in\mathbf{L}^2(0,T;\mathcal{V}_{g,h})\times L^1(0,T;\mathcal{Q}_{0,h})$ such that

$$(\partial_t\mathbf{u}_h,\mathbf{v}_h) + B(\mathbf{u}_h,(\mathbf{u}_h,p_h),(\mathbf{v}_h,q_h)) = \langle\mathbf{f},\mathbf{v}_h\rangle \qquad \forall\mathbf{v}_h\in\mathcal{V}_{0,h}, \forall q_h\in\mathcal{Q}_{0,h}. \tag{5.9}$$

In order to overcome the numerical instabilities and, eventually, to bypass the *inf-sup* condition that arises when problem (5.9) is solved, we use the VMS approach [100, 101] which consists in a two-scale decomposition of spaces $\mathcal{V}_0$, $\mathcal{V}_g$ and $\mathcal{Q}_0$ as $\mathcal{V}_0 = \mathcal{V}_{0,h}\oplus\widetilde{\mathcal{V}}_0$,

$\mathcal{V}_g = \mathcal{V}_{g,h} \oplus \widetilde{\mathcal{V}}_g$ and $\mathcal{Q} = \mathcal{Q}_{0,h} \oplus \widetilde{\mathcal{Q}}_0$, where $\widetilde{\mathcal{V}}_0$, $\widetilde{\mathcal{V}}_g$ and $\widetilde{\mathcal{Q}}_0$ are infinite-dimensional spaces that complete the FE spaces in $\mathcal{V}_0$, $\mathcal{V}_g$ and $\mathcal{Q}_0$, respectively. Hereinafter the subscript $(\cdot)_h$ will denote the FE component and the tilde $\widetilde{(\cdot)}$ the subgrid component. Applying the two-scale decomposition to (5.6) we obtain

$$(\partial_t \mathbf{u}_h, \mathbf{v}_h) + (\partial_t \tilde{\mathbf{u}}, \mathbf{v}_h) + B(\mathbf{a}; [\mathbf{u}_h, p_h], [\mathbf{v}_h, q_h]) - (\tilde{\mathbf{u}}, \nu\Delta\mathbf{v}_h + \mathbf{a}\cdot\nabla\mathbf{v}_h + \nabla q_h)_h \\ - (\tilde{p}, \nabla\cdot\mathbf{v}_h)_h = \langle \mathbf{f}, \mathbf{v}_h \rangle , \tag{5.10}$$

where $(\cdot,\cdot)_h = \sum_{K\in\mathcal{T}_h}(\cdot,\cdot)_K$ is the sum of scalar products (2.17) over each element K of the partition $\mathcal{T}_h$. The terms involving the subscales come from an element-wise integration by parts, in which the boundary terms $(\mathbf{v}_h, \nu\mathbf{n}\cdot\nabla\tilde{\mathbf{u}})_{\partial h}$ and $(q_h, \mathbf{n}\cdot\tilde{\mathbf{u}})_{\partial h}$ have been neglected (the subscript ∂h is used to denote the sum over all elements of the integral on the boundary of each element). It also involves the approximation $b(\mathbf{a}, \tilde{\mathbf{u}}, \mathbf{u}_h) \approx -(\tilde{\mathbf{u}}, \mathbf{a}\cdot\nabla\mathbf{v}_h)$ which implies neglecting $(\mathbf{v}_h, \mathbf{n}\cdot\mathbf{a}\tilde{\mathbf{u}})_{\partial h}$ and $(\tilde{\mathbf{u}}, \nabla\cdot\mathbf{a}\,\mathbf{v}_h)$. These approximations are discussed in [55] together with the choice of the advection velocity $\mathbf{a}$ which defines the type of scale splitting (linear or nonlinear), see also [56].

Problem (5.10) depends on $\tilde{\mathbf{u}} \in \widetilde{\mathcal{V}}_0$ and on $\tilde{p} \in \widetilde{\mathcal{Q}}_0$, $\widetilde{\mathcal{V}}_0$ and $\widetilde{\mathcal{Q}}_0$ being infinite-dimensional. Therefore, the equations for $\tilde{\mathbf{u}}$ and $\tilde{p}$ obtained after applying the two-scale decomposition cannot be directly solved, but some modelling steps are needed to obtain a feasible method. In this work we consider the velocity subscale as linear ($\mathbf{a} = \mathbf{u}_h$) and quasi-static, while the pressure subscale is neglected when equal order interpolation is used. Different approaches could be used for the definition of the subscales, see Chapter 4 for a deep explanation of the different choices and their numerical evaluation with equal order approximation. After the approximation of the Navier-Stokes operators by the stabilization parameters τ_m^{-1} and τ_c^{-1} (see for example [55]), and introducing the fine scales definitions into (5.10) we get the final discrete problem

$$(\partial_t \mathbf{u}_h, \mathbf{v}_h) + B_h(\mathbf{u}_h, (\mathbf{u}_h, p_h), (\mathbf{v}_h, q_h)) = L_h(\mathbf{v}_h, q_h) \qquad \forall \mathbf{v}_h \in \mathcal{V}_{0,h}, \forall q_h \in \mathcal{Q}_{0,h}, \tag{5.11}$$

The bilinear form B_h and the linear form L_h depend on the particular VMS method as discussed below.

5.3.1 Residual-based ASGS

The space for the subscales $\widetilde{\mathcal{V}}_0$ is determined by the definition of the projection $\mathcal{P}$ appearing in the right-hand side of (2.39)-(2.40). The ASGS method is obtained taking the subscales in the space of the residuals, that is, $\mathcal{P}_{\widetilde{\mathcal{V}}} := \mathbf{I}$ and $\mathcal{P}_{\widetilde{\mathcal{Q}}} := I$. The final discrete is given by (5.11) with $B_h = B_{asgs}$ and $L_h = L_{asgs}$ given by

$$B_{asgs}(\mathbf{u}_h, (\mathbf{u}_h, p_h), (\mathbf{v}_h, q_h)) := B(\mathbf{u}_h, (\mathbf{u}_h, p_h), (\mathbf{v}_h, q_h)) \tag{5.12}$$

$$+ (\tau_m(\partial_t \mathbf{u}_h - \nu\Delta\mathbf{u}_h + \mathbf{u}_h\cdot\nabla\mathbf{u}_h + \nabla p_h), \nu\Delta\mathbf{v}_h + \mathbf{u}_h\cdot\nabla\mathbf{v}_h + \nabla q_h),$$

$$L_{asgs}(\mathbf{v}_h, q_h) := \langle \mathbf{f}, \mathbf{v}_h \rangle + (\tau_m \mathbf{f}, \nu\Delta\mathbf{v}_h + \mathbf{u}_h\cdot\nabla\mathbf{v}_h + \nabla q_h). \tag{5.13}$$

Note that the pressure subscale term has been neglected in (5.12), that is we have taken $\tau_c = 0$. We have observed in [56] (using equal order interpolation) that this term introduces extra dissipation and does not in general improve the solution significantly.

5.3.2 Term by term OSS

Another possibility introduced in [50] is to consider the space of the subscales orthogonal to the FE space. This method is mainly motivated by the fact that a stability estimate for the projection onto the FE space of the pressure and the convective terms can already be obtained in the standard Galerkin method and therefore the only "missing" part is the orthogonal one. The Orthogonal Subscales (OSS) method is then obtained taking $\mathcal{P}_{\widetilde{\mathcal{V}}} := \Pi_h^{\perp} = \mathbf{I} - \Pi_h$ where Π_h is a projection onto the FE space. The L^2 orthogonality between the FE and subscale spaces is guaranteed considering the τ_m-weighted projection

$$(\tau_m \Pi_h(\mathbf{w}), \mathbf{v}_h) = (\tau_m \mathbf{w}, \mathbf{v}_h) \qquad \forall \mathbf{v}_h \in \mathcal{V}_{0,h}. \tag{5.14}$$

which requires the solution of a linear system defined by a τ_m-weighted mass matrix. Note that for this choice, the residual of the momentum equation does not depend on $\partial_t \mathbf{u}_h$. Likewise, $\mathcal{P}(\mathbf{f})$ in this case is only well defined for $\mathbf{f} \in L^2(\Omega)^d$. In the case of minimum regularity, $\mathbf{f} \in H^{-1}(\Omega)^d$, this term can be simply neglected without upsetting the accuracy of the method.

Remark 5.3.1. *Alternatively we can consider the standard L^2 projection*

$$(\Pi_h(\mathbf{w}), \mathbf{v}_h) = (\mathbf{w}, \mathbf{v}_h) \qquad \forall \mathbf{v}_h \in \mathcal{V}_{0,h}, \tag{5.15}$$

modifying the model of the subscales as

$$\tilde{\mathbf{u}} = \mathcal{P}_{\widetilde{\mathcal{V}}}(\tau_m \mathbf{R}_u), \tag{5.16}$$

$$\tilde{p} = \mathcal{P}_{\widetilde{\mathcal{Q}}}(\tau_c R_p). \tag{5.17}$$

With this modification the orthogonality between the FE and subscale spaces is guaranteed (and thus $(\partial_t \mathbf{u}_h, \tilde{\mathbf{u}}) = 0$) and standard mass matrices are used. This is an advantage from which we can take profit to build more efficient solvers. In the numerical tests we will favour this alternative.

Neglecting the pressure subscales the OSS method is given by (5.11) with $B = B_{oss}$ and $L_h(\mathbf{v}_h, q_h) = \langle \mathbf{f}, \mathbf{v}_h \rangle$ where

$$\begin{aligned} B_{oss}(\mathbf{u}_h, (\mathbf{u}_h, p_h), (\mathbf{v}_h, q_h)) &= B(\mathbf{u}_h, (\mathbf{u}_h, p_h), (\mathbf{v}_h, q_h)) \\ &+ (\tau_m(-\nu\Delta\mathbf{u}_h + \mathbf{u}_h \cdot \nabla \mathbf{u}_h + \nabla p_h), \nu\Delta\mathbf{v}_h + \mathbf{u}_h \cdot \nabla \mathbf{v}_h + \nabla q_h) \\ &- (\tau_m \boldsymbol{\eta}_h, \nu\Delta\mathbf{v}_h + \mathbf{u}_h \cdot \nabla\mathbf{v}_h + \nabla q_h). \end{aligned} \tag{5.18}$$

where $\boldsymbol{\eta}_h := \Pi_h(\mathbf{R}_u)$ is computed solving

$$(\tau_m \boldsymbol{\eta}_h, \boldsymbol{\kappa}_h) + (\tau_m(-\nu\Delta\mathbf{u}_h + \mathbf{u}_h \cdot \nabla\mathbf{u}_h + \nabla p_h), \boldsymbol{\kappa}_h) = (\tau_m \mathbf{f}, \boldsymbol{\kappa}_h) \qquad \forall \boldsymbol{\kappa}_h \in \mathcal{V}_{h,0}, \quad (5.19)$$

An implicit implementation of this method would require the introduction of an extra variable (the projection $\boldsymbol{\eta}_h$) and, more importantly, this variable would be coupled with both velocity and pressure. Due to the change on the sign of the Laplacian terms of the adjoint operator, the velocity-projection coupling is non-symmetric, as it can be seen in (5.18) and (5.19).

In order to reduce the coupling between variables we consider the *term by term* OSS method proposed in [52]. The main goal of this alternative is to stabilize separately the convective term and the pressure gradient term by two uncoupled orthogonal projections. As noted in [52] the *term by term* OSS has better stability properties than the classical one. Considering quasi-static and linear subscales the discrete problem is given by (5.11) with $B = B_{tbt\_oss}$ and $L_h(\mathbf{v}_h, q_h) = \langle \mathbf{f}, \mathbf{v}_h \rangle$ where

$$\begin{aligned} B_{tbt\_oss}(\mathbf{u}_h, (\mathbf{u}_h, p_h), (\mathbf{v}_h, q_h)) = B(&\mathbf{u}_h, (\mathbf{u}_h, p_h), (\mathbf{v}_h, q_h)) \\ &+ (\tau_m \mathbf{u}_h \cdot \nabla\mathbf{u}_h, \mathbf{u}_h \cdot \nabla\mathbf{v}_h) + (\tau_m \nabla p_h, \nabla q_h) \\ &- (\tau_m \boldsymbol{\eta}_h, \mathbf{u}_h \cdot \nabla\mathbf{v}_h) - (\tau_m \boldsymbol{\xi}_h, \nabla q_h). \end{aligned} \quad (5.20)$$

and $\boldsymbol{\eta}_h := \Pi_h(\mathbf{u}_h \cdot \nabla\mathbf{u}_h)$ and $\boldsymbol{\xi}_h := \Pi_h(\nabla p_h)$ are computed solving

$$(\tau_m \boldsymbol{\eta}_h, \boldsymbol{\kappa}_h) = (\tau_m \mathbf{u}_h \cdot \nabla\mathbf{u}_h, \boldsymbol{\kappa}_h) \qquad \forall \boldsymbol{\kappa}_h \in \mathcal{V}_{h,0}, \quad (5.21)$$

$$(\tau_m \boldsymbol{\xi}_h, \boldsymbol{\zeta}_h) = (\tau_m \nabla p_h, \boldsymbol{\zeta}_h) \qquad \forall \boldsymbol{\zeta}_h \in \mathcal{V}_{h,0}. \quad (5.22)$$

Note that with the *term by term* OSS method with implicit FE projections, there are $3d + 1$ unknowns per node, while for ASGS the number of unknowns is $d + 1$ per node. A priory it seems that such increase of unknowns make the former method not appealing in front of ASGS, but we will see later that the increase of computational cost is not linear with the increase of unknowns in this case. Furthermore, an optimal block preconditioning technique can be used to solve problem (5.11)-(5.20)-(5.21)-(5.22) taking advantage of its block structure. This point is further discussed in Section 5.4.

5.3.3 Term by term OSS with ISS elements

For equal order interpolation the pressure stabilization is mandatory but for ISS FE spaces pressure stability is guaranteed. Then, we could define a *term by term* OSS method for ISS FE that only stabilizes the convective term by an orthogonal FE projection. This approach would reduce the number of unknowns per node, keeping stability properties and the better conditioned matrix than ASGS. The definition of the *term by term* OSS-ISS method can be given by equation (5.11) with $B = B_{tbt\_oss\_iss}$ and

$L_h(\mathbf{v}_h, q_h) = \langle \mathbf{f}, \mathbf{v}_h \rangle$ where

$$\begin{aligned} B_{tbt\_oss\_iss}(\mathbf{u}_h, (\mathbf{u}_h, p_h), (\mathbf{v}_h, q_h)) &= B(\mathbf{u}_h, (\mathbf{u}_h, p_h), (\mathbf{v}_h, q_h)) \\ &+ (\tau_m \mathbf{u}_h \cdot \nabla \mathbf{u}_h, \mathbf{u}_h \cdot \nabla \mathbf{v}_h) - (\tau_m \boldsymbol{\eta}_h, \mathbf{u}_h \cdot \nabla \mathbf{v}_h) \\ &+ (\tau_c \nabla \cdot \mathbf{u}_h, \nabla \cdot \mathbf{v}_h), \end{aligned} \tag{5.23}$$

complemented by the FE projection of the convective term defined in (5.21). Note that in this case we include the effect of the pressure subscales, that is, the grad-div stabilization. However (for constant stabilization parameters) the projection of the divergence required to implement (2.40) needs not to be computed because it vanishes, which is implied by the discrete mass conservation equation. This is not the case when pressure stabilization is used. When only the projection of the convective term is considered, we neglect this projection to reduce the computational cost. Note, however, that this approximation does not introduce any consistency error.

5.3.4 The grad-div stabilization

It is important to highlight here the presence of the pressure subscale term, $\tau_c(\nabla \cdot \mathbf{u}_h, \nabla \cdot \mathbf{v}_h)$ in (5.23), first introduced in [75]. Even though it is usually included in VMS formulations it is sometimes neglected in practice, particularly when equal order interpolations with pressure stabilization are considered [56]. As it is shown in [77], using the grad-div stabilization in the simulation of laminar flows produces a *small improvement* of the results obtained with equal order interpolation but a *clear improvement* of the results obtained with ISS elements.

In the VMS decomposition, this term comes from the residual of the incompressibility constraint and its addition improves the conservation of mass as well as the effect that the error on the pressure field produces on the velocity field. In [145, 81] the authors assessed the use of ISS elements for the incompressible Navier-Stokes equations in the laminar regime, highlighting the importance of this term. The optimal choice of the parameter τ_c, discussed in [110], depends on the relative norms of the velocity and pressure (is therefore problem dependent) and can be of order one but also much bigger. On the other hand, in [44] it is proved that on a regular mesh, the Taylor-Hood approximations converge to a point-wise divergence-free solution, the one obtained using Scott-Vogelius (SV) elements [165], as $\tau_c \to \infty$. Then, it is seen that the optimal value of τ_c is an open question and, as stated in [145], *we may consider the search of optimal parameters as a trade-off between mass and momentum balance in the FE system.* In this work we will try to evaluate the importance of such term for turbulent incompressible flows when ISS elements are used. Thus, a detailed discussion of which are the values that should take τ_c is considered for each numerical test in further sections.

Apart from its influence on mass conservation, this term is also known to introduce numerical dissipation both when equal order [56] or ISS elements are used [145]. An energy balance of the term-by-term OSS is obtained taking $v_h = u_h$ and $q_h = p_h$ in

(5.11) and using (5.23)

$$\frac{1}{2}\frac{d}{dt}\|u_h\|^2 + \nu\|\nabla \mathbf{u}_h\|^2 + \|\tau_m^{1/2}(\mathbf{u}_h \cdot \nabla \mathbf{u}_h - \boldsymbol{\eta}_h)\|^2 + \|\tau_c^{1/2}\nabla \cdot \mathbf{u}_h\|^2 = \langle \mathbf{f}, \mathbf{u}_h \rangle \,. \tag{5.24}$$

Apart from the viscous dissipation coming from the Galerkin method, which is negligible in turbulent flows, we get extra dissipation that comes from the control of the orthogonal projection of the convective term and the dissipation that comes from the grad-div stabilization. When equal order interpolations are used this extra dissipation is not necessary and very good results are obtained taking $\tau_c = 0$ [56]. As discussed above, for ISS discretizations, the numerical dissipation introduced by the grad-div term is crucial to obtain accurate solutions. For instance, given a velocity space, the Taylor-Hood element has one order less pressure space than the stabilized equal order counterpart, leading to a poorer approximation of the mass conservation equation. Note that when τ_c goes to zero no dissipation is introduced but the accuracy in the satisfaction of mass conservation is poor. On the other hand taking τ_c large results in a strong imposition of the mass conservation (it acts as a penalty term) giving, for the Taylor-Hood pair on regular meshes, exact (point-wise) zero divergence in the limit [44]. In this case it turns out that the extra dissipation also vanishes as $\|\tau_c^{1/2}\nabla\cdot\mathbf{u}_h\| \longrightarrow 0$ in the limit of $\tau_c \longrightarrow \infty$ as $\tau_c^{-1/2}$ [131]. In any case it is important to keep in mind that when τ_c is changed the velocity field changes and the other dissipative terms in (5.24) also change. We cannot therefore conclude whether the method is more or less dissipative looking only at $\|\tau_c^{1/2}\nabla \cdot \mathbf{u}_h\|^2$. It is possible to arrive to such a conclusion when this parameter does not influence very much the solution, as in the case of equal order interpolation. This point is discussed in detail when presenting the numerical results in Section 5.5.

Finally, the grad-div term also has a strong influence on the conditioning of the linear system and therefore on the convergence of iterative solvers. When the monolithic system is considered, this term acts as an augmented Lagrangian term improving the convergence of block iterative schemes but it is also known that it introduces stiffness in the velocity block [146, 95].

An alternative to the parameter *tuning* needed for, e.g., Taylor Hood elements, is the use of divergence-free FEs that also satisfy the inf-sup condition (obviously, for these elements the grad-div stabilization vanishes). One of this group of elements is the Scott-Vogelius pair [165] which is given by the triangular/tetrahedral elements P_k/P_{k-1}^{disc}. This element is similar to the Taylor-Hood element except that the pressure space is discontinuous, which implies the property of point-wise divergence-free (taking the pressure test function as the divergence of the velocity). The Scott-Vogelius element is ISS under certain assumptions on the mesh, e.g., the order of the interpolation $k \geq d$ and the mesh is a barycentre-refinement of a regular mesh [131]. For quadrilateral meshes, Zhang introduced a new divergence-free ISS element in [184].

This work proposed a new family of quadrilateral elements that have a different interpolation space for each velocity component, which for the 3D case can be stated as $Q_{k+1,k,k} \times Q_{k,k+1,k} \times Q_{k,k,k+1}/Q_k^{disc}$. In this case, the pressure field is also discontinuous with spurious modes filtered. We do not consider these approaches here.

5.4 Block preconditioning for the monolithic problem

A common approach when the OSS method is used is to treat the projection explicitly within a Picard linearization scheme. That means to compute its value after the resolution of the velocity-pressure system and iterating until the solution converges. Although the resulting matrix has a better condition number than the ASGS method, the increase of nonlinear iterations due to the explicit treatment of the orthogonal projection may cause a lose of efficiency of this method in many cases. In [56] there is a computational cost analysis of these methods for turbulent incompressible flows where this effect can be shown.

Alternatively, the implicit approach of the OSS method increases the number of unknowns of the problem, not only having the usual velocity and pressure unknowns, but also the FE projection. Treating the FE projection as a new unknown, the system of equations to be solved is increased with equation (5.14) or (5.15). This projection unknown is coupled with both the velocity and the pressure and these blocks are nonsymmetric which makes the application of the block preconditioning technique more difficult. As mentioned, this is not the case when the term by term OSS or the term by term OSS with ISS elements are considered.

In order to present our recursive block-preconditioning technique, let us assume that we solve the Navier-Stokes problem using (to fix ideas) a backward Euler time integration with a monolithic approach, i.e., without any velocity-pressure segregation algorithm. We consider the implicit term-by-term OSS stabilization given by the bilinear form (5.20) as this is the more general case when talking about number of unknowns that appear in the system, since there are the velocity, pressure and two projections. The case of the term by term OSS with ISS elements and the ASGS are obtained just eliminating rows and columns of this system.

Assuming that $\mathbf{u}_h$, p_h, $\boldsymbol{\eta}_h$ and $\boldsymbol{\xi}_h$ are defined by a FE interpolation from the nodal values $\{\mathbf{U}^a\}_{a=1,\dots,N_u}$, $\{P^b\}_{b=1,\dots,N_p}$, $\{\boldsymbol{\Upsilon}^l\}_{l=1,\dots,N_\eta}$ and $\{\boldsymbol{\Xi}^m\}_{m=1,\dots,N_\xi}$, the FE approximation of the velocity, pressure and projection fields can be written as

$$\mathbf{u}_h(\mathbf{x}) = \sum_{a=1}^{N_u} \boldsymbol{\phi}_a(\mathbf{x})\mathbf{U}^a, \quad p_h(\mathbf{x}) = \sum_{b=1}^{N_p} \psi_b(\mathbf{x})P^b,$$

$$\boldsymbol{\eta}_h(\mathbf{x}) = \sum_{l=1}^{N_\eta} \boldsymbol{\phi}_\eta(\mathbf{x})\boldsymbol{\Upsilon}^l, \quad \boldsymbol{\xi}_h(\mathbf{x}) = \sum_{m=1}^{N_\xi} \boldsymbol{\phi}_\xi(\mathbf{x})\boldsymbol{\Xi}^m,$$

where $\{\boldsymbol{\phi}_{a,i}\}_{a=1,...,N_u;i=1,...,d}$, $\{\psi_b\}_{b=1,...,N_p}$, $\{\boldsymbol{\phi}_{l,i}\}_{l=1,...,N_\eta;i=1,...,d}$ and $\{\boldsymbol{\phi}_{m,i}\}_{m=1,...,N_\xi;i=1,...,d}$ are the Lagrangian basis associated to $\mathcal{V}_h$ and $\mathcal{Q}_h$. N_u, N_p, N_η and N_ξ are the total amount of nodes for the velocity, pressure and projection fields.

The matrix form of the problem (5.11) with the bilinear form (5.20) and, the projections (5.21)-(5.22) can be written as

$$\begin{bmatrix} \frac{1}{\delta t}\mathbb{M}+\mathbb{K}+\mathbb{C}+\mathbb{A}_\tau & \mathbb{G} & \mathbb{B}_{\eta,\tau} & 0 \\ \mathbb{D} & \mathbb{L}_\tau & 0 & \mathbb{B}_{\xi,\tau} \\ -\mathbb{B}_{\eta,\tau}^T & 0 & \mathbb{M}_{\eta,\tau} & 0 \\ 0 & -\mathbb{B}_{\xi,\tau}^T & 0 & \mathbb{M}_{\xi,\tau} \end{bmatrix} \begin{bmatrix} \mathbf{U} \\ \mathbf{P} \\ \boldsymbol{\Upsilon} \\ \boldsymbol{\Xi} \end{bmatrix} = \begin{bmatrix} \mathbf{F}_u \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \tag{5.25}$$

where $\mathbb{M}$, $\mathbb{K}$, $\mathbb{C}$, $\mathbb{D}$ and $\mathbb{G}$ are the matrices that arise from the Galerkin integration of the mass, diffusive, convective, velocity divergence and pressure gradient terms, respectively. The definition of the remaining terms are given by

$$\begin{aligned}
\mathbb{A}_\tau^{ab} &:= (\tau_m \mathbf{u}\cdot\nabla\boldsymbol{\phi}_a, \mathbf{u}\cdot\nabla\boldsymbol{\phi}_b) + (\tau_c \nabla\cdot\boldsymbol{\phi}_a, \nabla\cdot\boldsymbol{\phi}_b), & a,b &= 1,...,N_u,\\
\mathbb{L}_\tau^{ab} &:= (\tau_m \nabla\psi_a, \nabla\psi_b), & a,b &= 1,...,N_p,\\
\mathbb{B}_{\eta,\tau}^{ab} &:= -(\tau_m \mathbf{u}\cdot\nabla\boldsymbol{\phi}_a, \boldsymbol{\phi}_b), & a = 1,...,N_u, b &= 1,...,N_\eta,\\
\mathbb{M}_{\eta,\tau}^{ab} &:= -(\tau_m \boldsymbol{\phi}_a, \boldsymbol{\phi}_b), & a,b &= 1,...,N_\eta,\\
\mathbb{B}_{\xi,\tau}^{ab} &:= -(\tau_m \nabla\psi_a, \boldsymbol{\phi}_b), & a = 1,...,N_p, b &= 1,...,N_\xi,\\
\mathbb{M}_{\xi,\tau}^{ab} &:= -(\tau_m \boldsymbol{\phi}_a, \boldsymbol{\phi}_b), & a,b &= 1,...,N_\xi,
\end{aligned}$$

being a and b the node identification. Note that $\mathbb{D} = -\mathbb{G}^T$, when Dirichlet boundary conditions are considered. In general, $N_\xi = N_\eta$, then, $\mathbb{M}_{\xi,\tau} = \mathbb{M}_{\eta,\tau}$.

For the ASGS method, only the first two rows and columns of the matricial system (5.25) are present, with different definitions of $\mathbb{A}_\tau$ and $\mathbb{L}_\tau$ that are straight forward from the bilinear form (5.12). In the case of OSS-ISS method, only the last row and column disappear, keeping the same definition for the remaining terms.

To solve the system (5.25) we use a recursive block-preconditioning technique. This methodology was used in [18] for a multiphysics problem like the thermally coupled inductionless magnetohydrodynamics (MHD). The idea is to construct recursively block preconditioners of size 2×2 from an incomplete block factorization of the original 2×2 block matrices.

Let us consider a block system equivalent to (5.25) defined as

$$\begin{bmatrix} \tilde{\mathbb{M}}_\tau & -\tilde{\mathbb{B}}_\tau^T \\ \tilde{\mathbb{B}}_\tau & \tilde{\mathbb{K}}_\tau \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{\Xi}} \\ \tilde{\mathbf{U}} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \tilde{\mathbf{F}}_u \end{bmatrix}, \tag{5.26}$$

with $\tilde{\mathbb{K}}_\tau := \begin{bmatrix} \frac{1}{\delta t}\mathbb{M}+\mathbb{K}+\mathbb{C}+\mathbb{A}_\tau & \mathbb{G} \\ \mathbb{D} & \mathbb{L}_\tau \end{bmatrix}$, $\tilde{\mathbb{B}}_\tau := \begin{bmatrix} \mathbb{B}_{\eta,\tau} & 0 \\ 0 & \mathbb{B}_{\xi,\tau} \end{bmatrix}$, $\tilde{\mathbb{M}}_\tau := \begin{bmatrix} \mathbb{M}_{\eta,\tau} & 0 \\ 0 & \mathbb{M}_{\xi,\tau} \end{bmatrix}$, $\tilde{\boldsymbol{\Xi}} := \begin{bmatrix} \boldsymbol{\Upsilon} \\ \boldsymbol{\Xi} \end{bmatrix}$, $\tilde{\mathbf{U}} := \begin{bmatrix} \mathbf{U} \\ \mathbf{P} \end{bmatrix}$ and $\tilde{\mathbf{F}}_u := \begin{bmatrix} \mathbf{F}_u \\ 0 \end{bmatrix}$.

The matrix that appear in the system of equations (5.26) can be factorized into an exact LU matrix product as follows

$$\begin{aligned} \tilde{\mathbb{A}} := \begin{bmatrix} \tilde{\mathbb{M}}_\tau & -\tilde{\mathbb{B}}_\tau^T \\ \tilde{\mathbb{B}}_\tau & \tilde{\mathbb{K}}_\tau \end{bmatrix} &= \begin{bmatrix} \tilde{\mathbb{M}}_\tau & 0 \\ \tilde{\mathbb{B}}_\tau & \tilde{\mathbb{S}} \end{bmatrix} \begin{bmatrix} \mathbb{I} & -\tilde{\mathbb{M}}_\tau^{-1}\tilde{\mathbb{B}}_\tau^T \\ 0 & \mathbb{I} \end{bmatrix} \\ &= \begin{bmatrix} \mathbb{I} & 0 \\ \tilde{\mathbb{B}}_\tau\tilde{\mathbb{M}}_\tau^{-1} & \mathbb{I} \end{bmatrix} \begin{bmatrix} \tilde{\mathbb{M}}_\tau & -\tilde{\mathbb{B}}_\tau^T \\ 0 & \tilde{\mathbb{S}} \end{bmatrix}, \end{aligned} \tag{5.27}$$

being $\tilde{\mathbb{S}} := \tilde{\mathbb{K}}_\tau + \tilde{\mathbb{B}}_\tau^T\tilde{\mathbb{M}}_\tau^{-1}\tilde{\mathbb{B}}_\tau$ the Schur complement with respect to $\mathbf{U}$. In order to construct a preconditioner to the system (5.26) we build an inexact factorization of $\tilde{\mathbb{A}}$ such that each diagonal block is recursively preconditioned by another block preconditioner. We consider three different preconditioners of $\tilde{\mathbb{A}}$, which try to approximate the LU decompositions (5.27)

Diagonal preconditioner (D):
$$P_D(\tilde{\mathbb{A}}) = \begin{bmatrix} \tilde{\mathbb{M}}_\tau & 0 \\ 0 & \tilde{\mathbb{S}} \end{bmatrix}^{-1} = \begin{bmatrix} \tilde{\mathbb{M}}_\tau^{-1} & 0 \\ 0 & \tilde{\mathbb{S}}^{-1} \end{bmatrix}, \tag{5.28}$$

Upper preconditioner (U):
$$P_U(\tilde{\mathbb{A}}) = \begin{bmatrix} \tilde{\mathbb{M}}_\tau & -\tilde{\mathbb{B}}_\tau^T \\ 0 & \tilde{\mathbb{S}} \end{bmatrix}^{-1} = \begin{bmatrix} \tilde{\mathbb{M}}_\tau^{-1} & \tilde{\mathbb{M}}_\tau^{-1}\tilde{\mathbb{B}}_\tau^T\tilde{\mathbb{S}}^{-1} \\ 0 & \tilde{\mathbb{S}}^{-1} \end{bmatrix}, \tag{5.29}$$

Lower preconditioner (U):
$$P_L(\tilde{\mathbb{A}}) = \begin{bmatrix} \tilde{\mathbb{M}}_\tau & 0 \\ \tilde{\mathbb{B}}_\tau & \tilde{\mathbb{S}} \end{bmatrix}^{-1} = \begin{bmatrix} \tilde{\mathbb{M}}_\tau^{-1} & 0 \\ -\tilde{\mathbb{S}}^{-1}\tilde{\mathbb{B}}_\tau\tilde{\mathbb{M}}_\tau^{-1} & \tilde{\mathbb{S}}^{-1} \end{bmatrix}. \tag{5.30}$$

To apply (5.28)-(5.30) we need to compute the action of $\tilde{\mathbb{M}}_\tau^{-1}$ and $\tilde{\mathbb{S}}^{-1}$. In the first case, $\tilde{\mathbb{M}}_\tau^{-1}$ is block diagonal thus requiring the inverse of each projection mass matrix, $\mathbb{M}_{\eta,\tau}^{-1}$ and $\mathbb{M}_{\xi,\tau}^{-1}$. In a serial computation, they are applied by a direct method whereas in a parallel computation they are approximated by one application of a diagonal DD preconditioner constructed with the values of the diagonal of $\mathbb{M}_{\eta,\tau}$ and $\mathbb{M}_{\xi,\tau}$, respectively. In the second case we approximate the Schur complement inverse as $\tilde{\mathbb{S}} \approx \tilde{\mathbb{K}}_\tau$, that is, we neglect the contribution from the projections $\tilde{\mathbb{B}}_\tau^T\tilde{\mathbb{M}}_\tau^{-1}\tilde{\mathbb{B}}_\tau$ in the preconditioner. This approximation corresponds to use the solution of a system arising from a non-consistent formulation in which artificial pressure and streamline diffusion are added. Stability of the preconditioner is therefore guaranteed.

For the application of $\tilde{\mathbb{K}}_\tau^{-1}$ we consider other three preconditioners equivalent to (5.28)-(5.30). Using the notation $\mathbb{K}_\tau := \frac{1}{\delta t}\mathbb{M}+\mathbb{K}+\mathbb{C}+\mathbb{A}_\tau$ for the velocity block matrix and $\mathbb{S} : \mathbb{L}_\tau - \mathbb{D}\mathbb{K}_\tau^{-1}\mathbb{G}$ for the Schur complement associated to the pressure field, we have the following block preconditioners

$$\text{Diagonal preconditioner (D):} \qquad P_D(\tilde{\mathbb{K}}_\tau) = \begin{bmatrix} \mathbb{K}_\tau & 0 \\ 0 & \mathbb{S} \end{bmatrix}^{-1} = \begin{bmatrix} \mathbb{K}_\tau^{-1} & 0 \\ 0 & \mathbb{S}^{-1} \end{bmatrix}, \tag{5.31}$$

$$\text{Upper preconditioner (U):} \qquad P_U(\tilde{\mathbb{K}}_\tau) = \begin{bmatrix} \mathbb{K}_\tau & \mathbb{G} \\ 0 & \mathbb{S} \end{bmatrix}^{-1} = \begin{bmatrix} \mathbb{K}_\tau^{-1} & -\mathbb{K}_\tau^{-1}\mathbb{G}\mathbb{S}^{-1} \\ 0 & \mathbb{S}^{-1} \end{bmatrix}, \tag{5.32}$$

$$\text{Lower preconditioner (U):} \qquad P_L(\tilde{\mathbb{K}}_\tau) = \begin{bmatrix} \mathbb{K}_\tau & 0 \\ \mathbb{D} & \mathbb{S} \end{bmatrix}^{-1} = \begin{bmatrix} \mathbb{K}_\tau^{-1} & 0 \\ -\mathbb{S}^{-1}\mathbb{D}\mathbb{K}_\tau^{-1} & \mathbb{S}^{-1} \end{bmatrix}. \tag{5.33}$$

In this case, the inverse of the Schur complement is approximated by a pressure Laplacian matrix, $\mathbb{S}^{-1} \sim \delta t \mathbb{L}_p^{-1}$. In regimes where the viscous term becomes dominant, one can use the Cahouet-Chabard preconditioner to approximate the inverse of the Schur complement, i.e., $\mathbb{S}^{-1} \sim (\nu + \tau_c)\mathbb{M}_p^{-1} + \delta t \mathbb{L}_p^{-1}$, see [95, 147]. For the serial case, the Laplacian matrix is inverted using a direct method, but in a parallel context we approximate the inverse of $\mathbb{L}_p$ by one application of a Balancing Domain Decomposition by Constraints (BDDC) preconditioner over such matrix. Something similar is done for the inverse of the velocity block matrix $\mathbb{K}_\tau$. Here we use a direct method for the serial case and one application of BDDC preconditioner when parallel solvers are treated. We refer to [183] for the application of the BDDC preconditioner to nonsymmetric problems. For a more detailed description of the implementation and algorithms used for the recursive block-preconditioning technique we refer to [18]. Besides monolithic approaches, splitting velocity-pressure techniques are commonly used for high Reynolds turbulent flows, which can be interpreted as one application of a block-preconditioner [69, 12]. High order time integration schemes that segregate velocity and pressure computation have recently been proposed in [57]. Since the stabilized inf-sup stable FEs proposed herein do not introduce any term at the pressure-pressure block, they are particularly well-suited for this type of schemes, because it keeps the index-2 differential-algebraic nature of the problem.

5.5 Numerical experiments

In this section we provide some numerical results for turbulent incompressible flow simulations using all the methods stated above. First, we do a comparative analysis between the three stabilization methods developed in Section 5.3. The parameter election for the pressure subscale term is assessed in all numerical tests performed in this section. But, first of all, we show some results about the computational cost of the VMS methods considered in this work. To check the efficiency of each method we solve a simple 2D steady problem with analytical solution.

5.5.1 Analytical colliding flow

In this section we solve a problem with analytical solution that models a colliding flow. This test have been used in [69] to discuss error estimates for the Stokes and Navier-Stokes problems using ISS FE velocity-pressure pairs. It has an analytical solution with the expression:

$$\mathbf{u}(x,y,t) = \begin{bmatrix} 20xy^3 \\ 5x^4 - 5y^4 \end{bmatrix}, \tag{5.34}$$

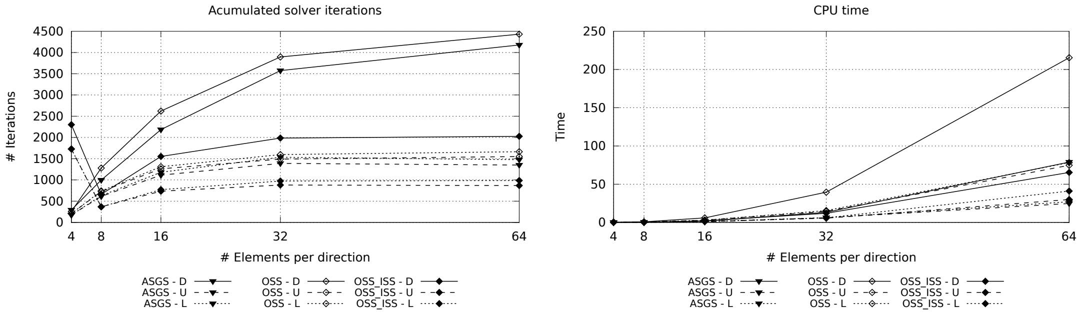
$$p(x,y) = 60x^2y - 20y^3 + 40. \tag{5.35}$$

The problem is solved in the square domain $[-1,1] \times [-1,1]$ with a Reynolds number $Re = 25$. Here we analyze the solver iterations convergence in h for each preconditioner defined in Section 5.4 as well as the elapsed CPU time and the error on the velocity and pressure fields. Due to the simplicity of the test, we will solve the problem in serial using Q_1/Q_1 elements meshes for the ASGS and term-by-term OSS, and Q_2/Q_1 elements mesh for the OSS-ISS method.

We only show the results when using the lower-triangular preconditioner for the global matrix $P_L(\tilde{\mathbb{A}})$, defined in (5.30). The diagonal version $P_D(\tilde{\mathbb{A}})$ defined in (5.28) give poorer performance and same results are obtained with the upper-triangular version $P_U(\tilde{\mathbb{A}})$ (5.29). For the velocity-pressure block, all preconditioners defined in (5.31)-(5.33) are considered.

In Figure 5.1 we depict the solver iterations needed to solve the problem as well the total elapsed CPU time for different meshes composed by $4, 8, 16, 32$ or 64 elements per direction. When the OSS-ISS method is considered, as we use Q_2/Q_1 elements, the number of elements per direction is divided by two. The amount of solver iterations shown in Figure 5.1(a) is computed adding all solver iterations needed for all nonlinear iterations. We see in this figure that the OSS-ISS method has an increase of solver iterations for the coarser mesh, compared with the other two methods. This behaviour is produced by the fact that for this case the mesh is too coarse, 2×2 Q_2/Q_1 elements, and the nonlinear iterations suffer a drastic increase. For finer meshes, the OSS-ISS method needs less solver iterations than the other two. We see that for the diagonal preconditioner $P_D(\mathbb{K}_\tau)$ the OSS-ISS is the only that scales when we refine the mesh. The upper and lower preconditioners, $P_U(\mathbb{K}_\tau)$ and $P_L(\mathbb{K}_\tau)$, have similar results with a slightly better performance of the upper version for all methods. The ASGS and OSS methods also need a similar number of iterations to solve the problem, being the OSS method a little bit over the ASGS.

Looking at the elapsed computational time, Figure 5.1(b), we see that the OSS method is clearly more expensive, in terms of consumed time, than the other two methods. The ASGS method is slightly faster than the OSS-ISS, but the differences are almost negligible for the $P_U(\mathbb{K}_\tau)$ preconditioner. The fact that the most expensive method is the OSS method is justified by the number of Degrees Of Freedom (DOFs) that appear in the system of equations. As stated in Section 5.4, the OSS method has two vectorial unknowns more than ASGS method. On the other hand, the OSS-ISS method also has an additional vectorial unknown than ASGS, but the pressure field is approximated with half nodes. Without taking into account the boundary conditions, for the 4×4 elements mesh (2×2 for the OSS-ISS case), we would have 175 DOFs for the OSS method, 75 DOFs for the ASGS method and 109 DOFs for the OSS-ISS method. Then, we see that, although it has more DOFs, the OSS-ISS method is comparable in terms of computational time with the ASGS. Further, the number of DOFs is a wrong measure of the CPU cost. The most expensive problem in such simulations is the pressure Poisson equation, which in this case is smaller (one order less) for ISS elements. Further, the projection DOFs are almost for free, since they just involve a mass matrix solve. Further, the ISS method exhibit a clearly lower number of iterations.

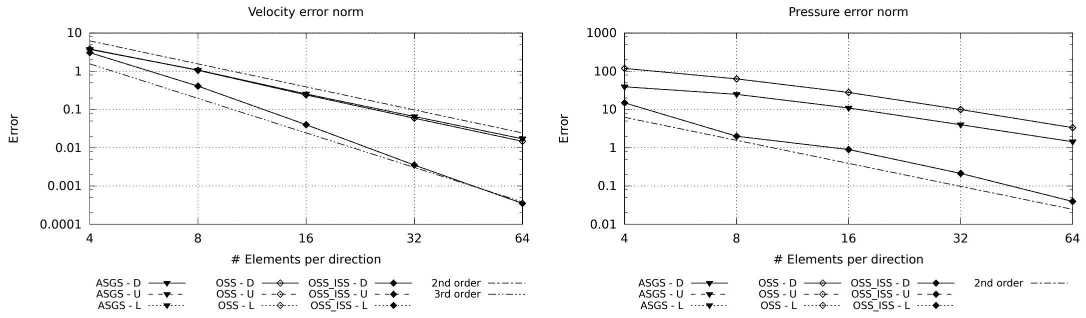


(a) Solver iterations

(b) Elapsed CPU time

FIGURE 5.1: Colliding flow solver iterations and elapsed CPU time using $P_U(\tilde{\mathbb{A}})$ for the global matrix.

Let us now focus on the error in the velocity and pressure fields. Since we are solving a problem with analytical solution, we can evaluate exactly the error of the FE approximation, $e_u := \|\mathbf{u}_h - \mathbf{u}\|$ and $e_p := \|p_h - p\|$. Figure 5.2 depicts the convergence of both errors when refining the mesh. In Figure 5.2(a) we see that the velocity error converges as expected, with a 2nd order rate for the ASGS and OSS methods, which are approximated by Q_1/Q_1 elements, and with a 3rd order rate for the OSS-ISS method, which is approximated by a Q_2/Q_1 element. Looking at Figure 5.2(b), it is seen that the pressure error norm converges with a 2nd order rate for all the methods, as expected. Note that there is no difference between the results changing the preconditioner, since the solution is the same for all cases because we are not modifying the system that is solved.

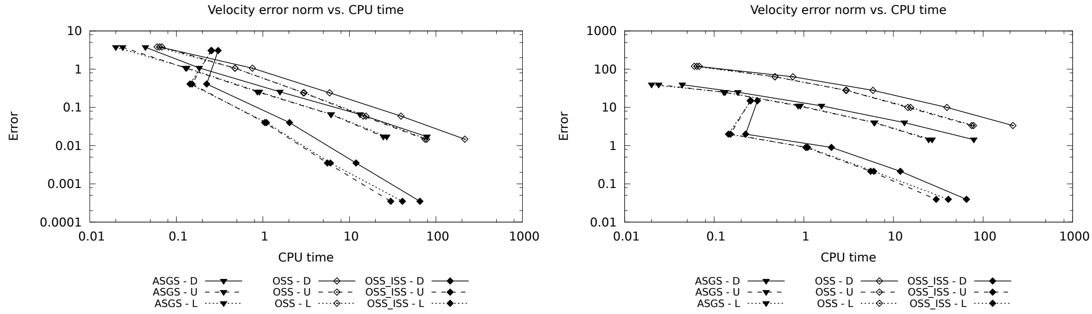


(a) Velocity error

(b) Pressure error

FIGURE 5.2: Colliding flow error convergence using $P_U(\tilde{\mathbb{A}})$ for the global matrix.

In order to check the efficiency of each method we compare the error norm with the elapsed computational time. In this way we have an idea of the time needed for a given method to achieve certain solution accuracy. Figure 5.3 shows this comparison for velocity (Figure 5.3(a)) and pressure (Figure 5.3(b)) fields, where we see that, excluding the coarser mesh, the OSS-ISS method is much more efficient than the other two. Furthermore, the most efficient preconditioner is the upper version $P_U(\mathbb{K}_\tau)$.



(a) Velocity error vs. CPU time

(b) Pressure error vs. CPU time

FIGURE 5.3: Colliding flow error vs. CPU time using $P_U(\tilde{\mathbb{A}})$ for the global matrix.

5.5.2 Taylor Green Vortex flow

It has been shown by many authors that VMS stabilization terms can act as a LES model for turbulent flows, introducing the appropriate dissipation of the small scales that are not captured by the coarse solution. In particular, ASGS and OSS methods were assessed in [56], showing that these methods are capable to perform a good LES simulation of different turbulent benchmark problems. In order to check the performance of the convection-only OSS stabilization of ISS elements for a LES simulation, we analyze its behaviour in the TGV flow, extensively described in Section 3.2.3.

We recall that this test is solved in a computational domain given by the cube $(0, 2\pi)^3$ with periodical boundary conditions and the initial condition is given by an analytical field (see, e.g., [31])

$$\mathbf{u}(x,y,z,0) = \begin{pmatrix} u_x \\ u_y \\ u_z \end{pmatrix} = \begin{pmatrix} u_0 \cos(x)\sin(y)\sin(z) \\ -u_0 \sin(x)\cos(y)\sin(z) \\ 0 \end{pmatrix} \tag{5.36}$$

$$p(x,y,z,0) = p_0 + \frac{1}{16}\left(\cos(2x)+\cos(2y)\right)\left(\cos(2z)+2\right),$$

with

$$u_0 = \frac{2}{\sqrt{3}} \sin\left(\gamma + \frac{2\pi}{3}\right).$$

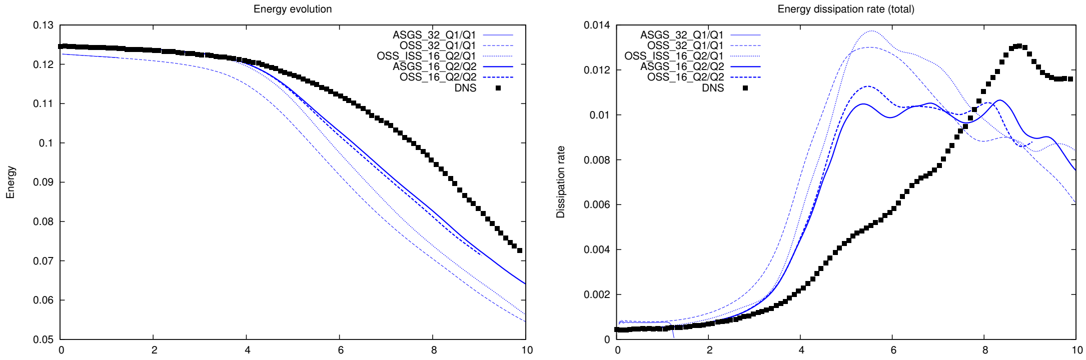
Being $\gamma = 0$, resulting in a mean initial velocity $u_0 = 1$. We consider the TGV problem with a Reynolds number $\mathrm{Re} = 1600$.

Setting

The problem is solved from $t = 0.0$ to $T = 10.0$ with a fixed time step size of $\delta t = 5.0 \cdot 10^{-2}$ using a Crank-Nicolson time integration scheme, and the results are compared against a DNS by Brachet et al. [31]. We discretize the domain using different choices of the number of elements and the order of approximation, having two different group of discretizations; one with 32^3 velocity DOFs and another with 64^3 velocity DOFs (for each velocity component). The former will be composed by the following meshes: 32^3 Q_1/Q_1, 16^3 Q_2/Q_2 elements or 16^3 Q_2/Q_1 elements when we use ISS discretization. The second group of meshes is made by: 64^3 Q_1/Q_1, 32^3 Q_2/Q_2 or 32^3 Q_2/Q_1 elements. For the stabilized formulations, ASGS and OSS, the algorithmic constants are $c_1 = 12$ $c_2 = 2$ and $c_c = 0.0$, and for the OSS-ISS method the same c_1 and c_2 algorithmic constants are used, but $c_c = 4.0$ unless noted otherwise. This choice of c_c for the OSS-ISS method is assessed in a following subsection.

Comparison between VMS methods

In Figure 5.4 we show the energy evolution and the energy dissipation rate for the ASGS, the OSS, and the OSS-ISS methods. A first thing that we have to state at this point is that the ASGS method with 32^3 Q_1/Q_1 has failed to converge at early stages of the problem, a behaviour also observed in [56]. Looking at Figure 5.4(a) it is clear that the degree of interpolation makes a great difference on the solution, even with the same number of DOFs, the solution is more accurate when a higher order of interpolation is used. In the same figure we see that the loss of precision in the pressure for the ISS elements Q_2/Q_1 does affect the solution, giving a result between the Q_1/Q_1 and Q_2/Q_2 solutions. At Figure 5.4(b) we see that the OSS-ISS method is more dissipative than the others. This behaviour is caused by the lack of accuracy in the pressure field, which in turn affects the conservation of mass of the problem. A more exhaustive analysis of the effect of this term is done below.



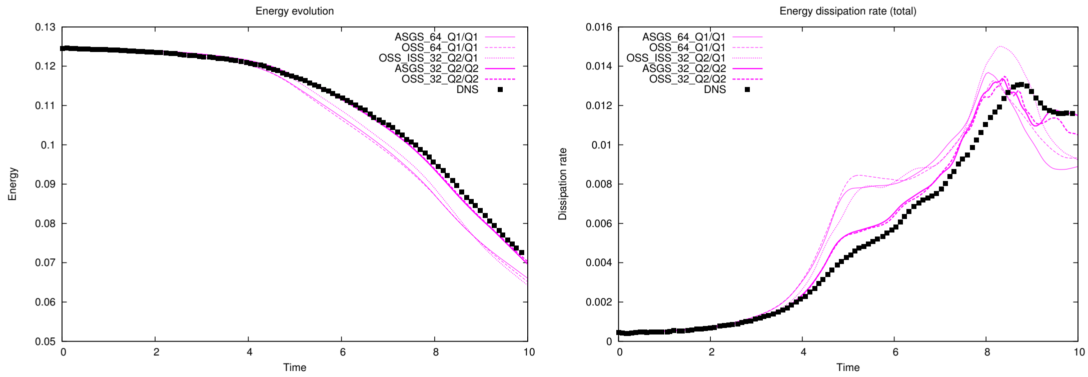
(a) Energy evolution

(b) Total energy dissipation rate

FIGURE 5.4: Energy and Total energy dissipation rate evolution with 32^3 velocity DOFs

Although an important improvement of the solution is achieved by increasing the order of interpolation, 32^3 velocity DOFs is still a very coarse mesh and the results shown in Figure 5.4 are far from the DNS ones. Then, the same problem is solved in a finer mesh, with the double of velocity DOFs per direction, with the results shown in Figure 5.5. In this case, all methods converge at all time steps. Figure 5.5(a) depicts the energy evolution and it is also seen that the increase on the degree of interpolation results in a more accurate solution. Both ASGS and OSS methods with 32^3 Q_2/Q_2 elements have very accurate results, providing a solution almost on top of the DNS. There are very little differences between stabilization methods when the same discretization is used. Furthermore, the results of the OSS-ISS method are closer to the Q_1/Q_1 discretization than to the Q_2/Q_2 one. The total energy dissipation rate shown in Figure 5.5(b) denote a very good agreement of the Q_2/Q_2 solution with the DNS, while the Q_1/Q_1 discretization for both ASGS and OSS methods are still more diffusive. Note that the OSS-ISS method for this discretization has more or less the same energy dissipation as the Q_1/Q_1 discretizations.

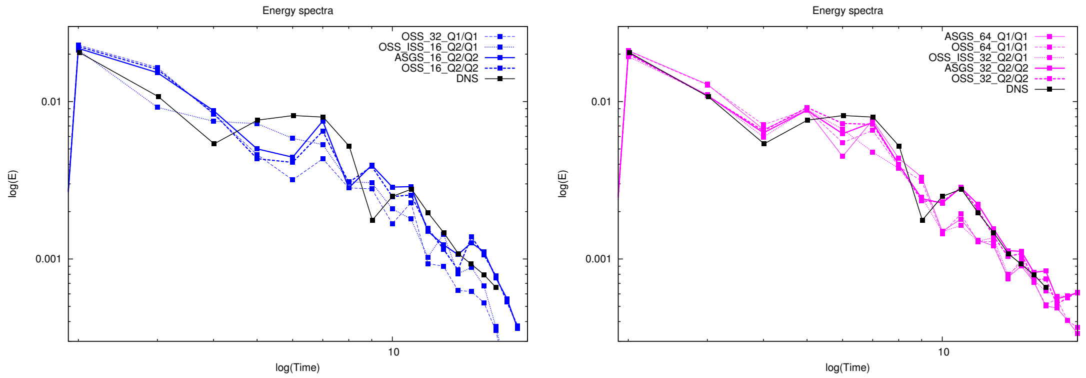
When analyzing the suitability of a LES model, a very important turbulent quantity to take into account is the energy spectra. It gives us information about how the energy is distributed among the scales of the problem. In order to assess the behaviour of the proposed methods in that aspect, we compare our results by the DNS by Gassner et al. [80]. In Figure 5.6 the energy spectra at $t = 9.0$ is depicted for both discretization groups. The coarser cases shown in Figure 5.6(a) are all far from the DNS result, but follow the same pattern, with most of the energy on the greatest scales and little energy on the small scales, without any pileup of energy on the small scales. When the mesh is refined, see Figure 5.6(b), the computed energy spectra tends to the DNS one. Note that also in this plot, Q_2/Q_2 discretization have better agreement with the DNS, specially



(a) Energy evolution (b) Total energy dissipation rate

FIGURE 5.5: Energy and Total energy dissipation rate evolution with 64^3 velocity DOFs

on the small scales, where the influence of the enrichment of the interpolation space is patent.



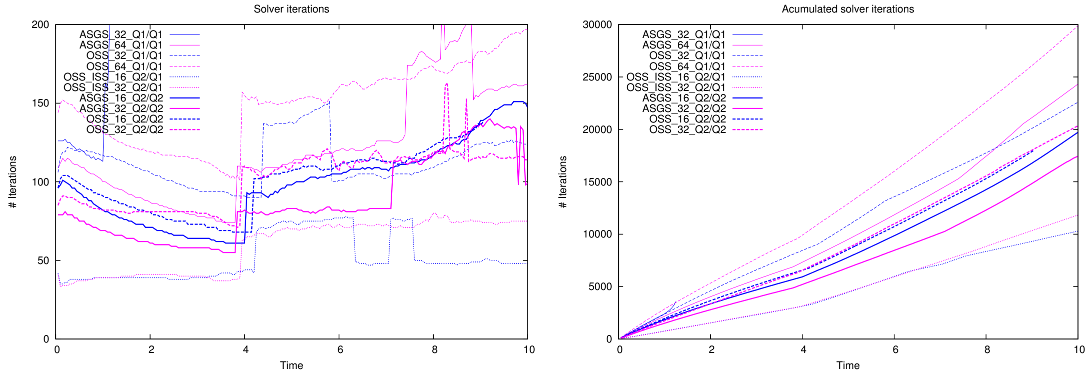
(a) 32^3 velocity DOFs (b) 64^3 velocity DOFs

FIGURE 5.6: Energy spectra at $t = 9.0$

Computational cost

It is clear that a good LES model has to reproduce as accurately as possible all the turbulent quantities, even for coarse meshes. The results presented until now show that ASGS and OSS methods perform better than the OSS-ISS when Q_2/Q_2 elements are used. Additionally, it has been seen that all methods converge to the DNS results when the mesh is refined. But a crucial point that has to be always taken into account when we talk about numerical simulations is the computational cost. At the end, we are looking for the cheapest method that allow us to reproduce accurately the physical phenomena that takes place in a turbulent flow. So in this subsection we will discuss the computational cost associated to each method and their efficiency when solving this kind of flows.

To check the computational cost we look at the number of solver iterations needed for each method, shown in Figure 5.7. In particular, Figure 5.7(a) depicts the total amount of solver iterations at each time step, adding up all nonlinear iterations. Note that the change of the nonlinear iterations along the time can be clearly noticed by the jumps on the curves. We see that the cheapest method is the OSS-ISS for both discretizations, 32^3 and 64^3 velocity DOFs, with much less solver iterations per time step than the other methods. It is also seen that ASGS is a little bit cheaper than the OSS method, which can be caused by the size of the system, much bigger for the OSS case due to the implicit treatment of the projections. An interesting result also seen in 5.7(a) is the improvement on the computational cost when we go from Q_1/Q_1 to Q_2/Q_2 discretization, keeping constant the number of DOFs. Looking at 5.7(b) we realize that at the end of the computation, the total amount of solver iterations needed by OSS-ISS method is around 1.5 times less than the ASGS method, for the finer mesh, and half of them for the coarse mesh. This result indicates that the OSS-ISS is a very good approach as a LES model for turbulent problems since, although it has been seen that is not the most accurate when looking to the turbulent quantities, it is much cheaper and we can refine the mesh in order to get better results remaining competitive with the other methods. Further, let us remark that the pressure Poisson solvers are the most computationally intensive, and ISS methods involve Q_1 pressure spaces for Q_2 ones for stabilized methods.



(a) Solver iterations per time step

(b) Accumulated solver iterations

FIGURE 5.7: Computational cost

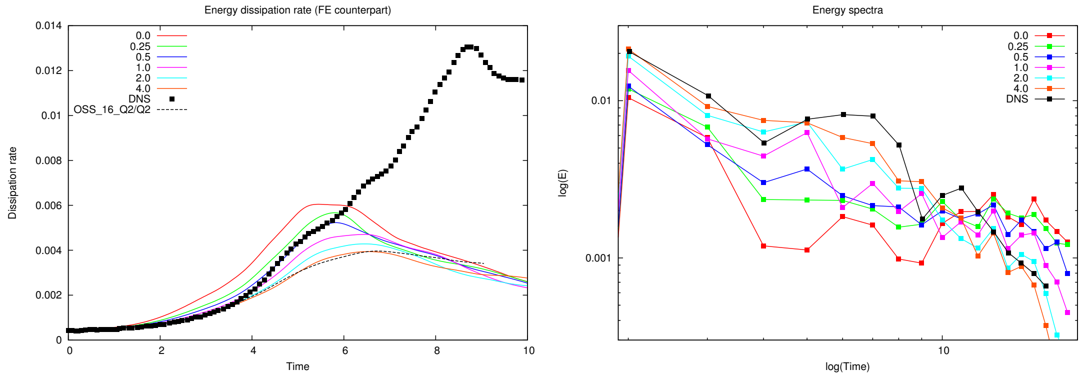
Influence of the pressure subscale term

As said above and also stated in Section 5.3.4, the pressure subscale $\tau_c(\nabla \cdot \mathbf{u}_h, \nabla \cdot \mathbf{v}_h)$ term has an important role when using ISS elements. In this subsection we are going to analyse the effect of this term on the results when simulating the TGV problem. Hence, we redefine the τ_c definition in (2.45) by

$$\tau_c = c_c \left(\nu + \frac{c_2}{c_1} h |\mathbf{u}_h| \right), \tag{5.37}$$

which is equivalent to (2.45) when $c_c = 1.0$. We keep the algorithmic parameters $c1 = 12.0$ and $c2 = 2.0$ constant, so considering different values of c_c we can evaluate the influence of the pressure subscale term on the solution. In this case we choose six different configurations $c_c = \{0.0, 0.25, 0.5, 1.0, 2.0, 4.0\}$ and solve the problem with the 16^3 Q_2/Q_1 elements mesh. We compare the solution against the one obtained with the OSS method with 16^3 Q_2/Q_2 elements discretization.

In Figure 5.8(a) the energy dissipation rate of the FE counterpart is shown. That is the viscous term $\nu\|\nabla \mathbf{u}_h\|^2$ that appears in equation (5.24). We see that when we reduce c_c the viscous dissipation introduced by the FE counterpart increase, being the case $c_c = 0.5$ the one closer to the DNS curve. But if we look at the energy spectra shown in Figure 5.8(b), we see that this is not a good choice. What is actually happening when c_c goes to zero is that the dissipation is taking place in the largest scales of the problem, while the smallest ones keep the energy, resulting in an energy pileup at the tail of the spectra. This means that the energy is not dissipating in the correct way and the small scales, which are the ones that have more influence on the viscous dissipation term, have more energy than the desired one. Therefore, a good selection is to choose $c_c = 4.0$, which results are closer to the OSS method and has a better energy spectra shape. A higher value of c_c eventually lead to unstable solutions, for instance, with $c_c = 8.0$ and $\delta t = 5.0 \cdot 10^{-2}$ the solution fails to converge at $t = 0.4$. In that case, the solution becomes unstable and the nonlinear iteratition does not reach the required tolerance.

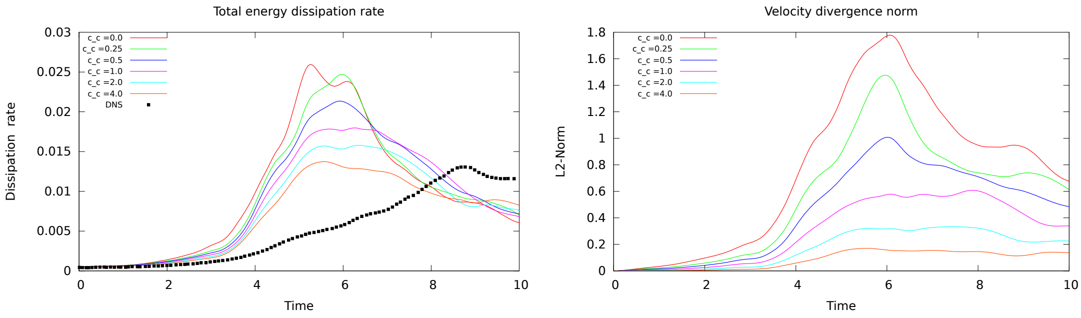


(a) Viscous energy dissipation rate

(b) Energy spectra

FIGURE 5.8: Comparison for different c_c choices for OSS-ISS with 32^3 velocity DOFs

As stated in Section 5.3.4, the pressure subscale term is essential to enforce the incompressibility constrain at the discrete level. In Figure 5.9 the total energy dissipation rate (Figure 5.9(a)) is depicted together with the velocity divergence $L2$-norm $\|\nabla \cdot \mathbf{u}_h\|$ (Figure 5.9(b)). We see that, effectively, $\|\nabla \cdot \mathbf{u}_h\|$ is reduced when τ_c is increased. The case of $c_c = 0.0$ give especially bad results in terms of mass conservation, affecting also to the energy dissipation rate. In this case, as seen in Figure 5.8(b), the over-dissipation on the large scales and the under-dissipation on the smallest ones makes that the velocity spatial derivatives become relevant, increasing the $\|\nabla \cdot \mathbf{u}_h\|$ term. The introduction of $\tau_c \|\nabla \cdot \mathbf{u}_h\|$ into the energy dissipation equation (5.24) changes the way in which the flow dissipates its energy among the different scales, decreasing the energy of the small scales and, then, reducing the importance of the spatial derivatives, which is reflected in a lower value of $\|\nabla \cdot \mathbf{u}_h\|$ but also in a lower dissipation rate, as seen in Figure 5.9(a), even if a positive term has been added to (5.24).



(a) Total energy dissipation rate (b) $\|\nabla \cdot \mathbf{u}_h\|$

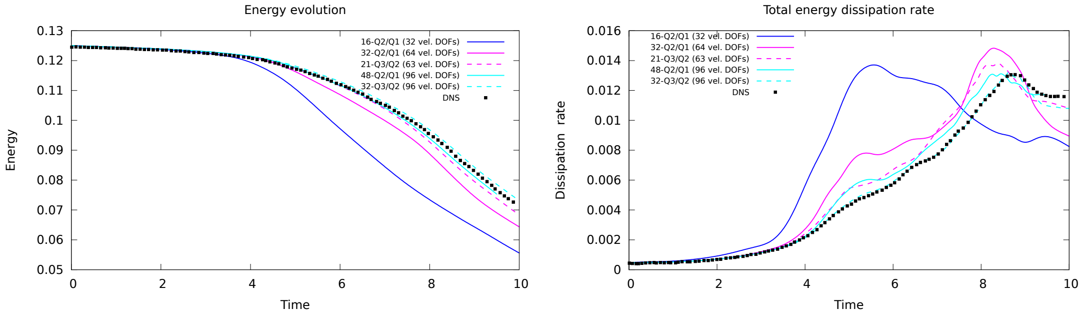
FIGURE 5.9: Total energy dissipation and velocity divergence $L2$-norm for different c_c choices for OSS-ISS with 32^3 velocity DOFs

Refinement analysis for the OSS-ISS method

In the computational cost analysis shown above we have seen that OSS-ISS has a great potential as a LES model, especially when we the computational cost is taken into account. This subsection aims to check the performance of this method when we refine the mesh both reducing the element size and increasing the interpolation order.

Once determined that the best algorithmic constant for the pressure subscale term is $c_c = 4.0$ when ISS FEs are used, we keep this value constant and change the discretization. A refinement analysis can be done to determine if the LES approach presented in this work effectively converges to the DNS solution when we refine the mesh. In section 5.5.2 the TGV problem has been solved using different discretizations with 32^3 and 64^3 velocity DOFs. Now we go further and also solve the problem with 96^3 velocity DOFs, that is a 48^3 Q_2/Q_1 elements mesh. Furthermore, here we also use Q_3/Q_2 FEs. In particular, a 21^3 and 32^3 Q_3/Q_2 elements meshes are used, corresponding to the group of 64^3 and 96^3 velocity DOFs, respectively. We also decrease the time step to $\delta t = 2.5 \cdot 10^{-2}$ for the discretizations with 64^3 velocity DOFs and $\delta t = 1.5 \cdot 10^{-2}$ for the discretizations with 96^3 velocity DOFs.

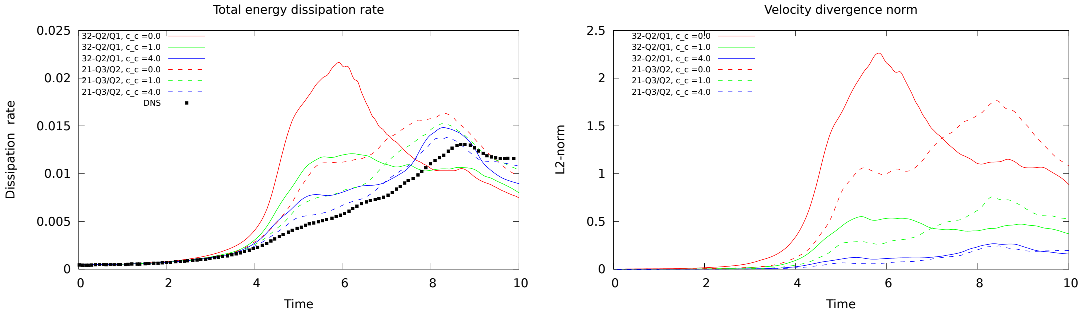
In Figure 5.10 we depict the kinetic energy and the total energy dissipation rate evolution for the different discretizations considered in this refinement analysis. Looking at the energy evolution in Figure 5.10(a) it is clearly seen that the solution converge to the DNS results, giving the 32^3 Q_3/Q_2 elements mesh a very accurate solution, which is also evident in Figure 5.10(b) where the total energy dissipation rate of this discretization is on top of the DNS solution. We also see in Figure 5.10 the relevance of the degree of interpolation, where for a given number of DOFs, the higher-order discretization results in a better solution.



(a) Energy evolution (b) Total energy dissipation rate

FIGURE 5.10: Energy and Total energy dissipation rate evolution refining the mesh with $c_c = 4.0$.

Now, we ask ourselves how the tuning of the parameter τ_c affects when finer meshes are used. In particular, we want to know how important the c_c parameter becomes when higher order FEs are used. To answer this question we solve the TGV problem for two different discretizations: 32^3 Q_2/Q_1 and 21^3 Q_3/Q_2 elements meshes, and three different values of c_c: 0.0, 1.0 and 4.0.

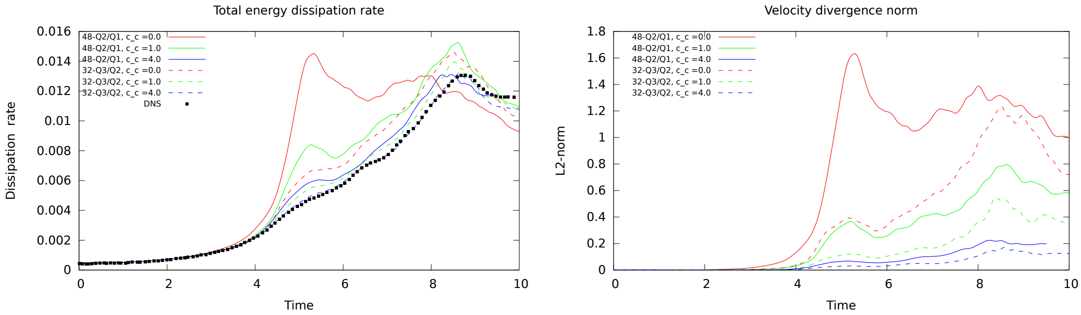


(a) Total energy dissipation rate (b) $\|\nabla \cdot \mathbf{u}_h\|$

FIGURE 5.11: Total energy dissipation and velocity divergence $L2$-norm for different c_c choices for OSS-ISS with 64^3 velocity DOFs

Figure 5.11 depicts the total energy dissipation rate (Figure 5.11(a)) and the velocity divergence $L2$-norm (Figure 5.11(b)) for the two discretizations considered and for different choices of c_c. We still see a dependence on c_c, but differences reduce when higher order of interpolation is used, as expected. It is also seen that when increasing c_c, the differences between the two discretizations are reduced (for a fixed c_c). This behaviour is particularly significant when we look at the velocity divergence norm in Figure 5.11(b), where we see that for $c_c = 0$ the two discretizations give a completely different evolution of $\|\nabla \cdot \mathbf{u}_h\|$, while for $c_c = 4.0$ the results are almost the same. In the same figure, we can see that the largest value of $\|\nabla \cdot \mathbf{u}_h\|$ for c_c is larger than the largest one depicted in Figure 5.9(b), which is a result of a 16^3 Q_2/Q_1 elements mesh.

Going further we also do the same test for the 48^3 Q_2/Q_1 and 32^3 Q_3/Q_2 elements mesh (Figure 5.12). In this figure we see that the differences between the three c_c cases are reduced. Looking at Figure 5.12(a) it is seen that for $c_c = 0$ and Q_2/Q_1 elements, although the result is far from the DNS, the maximum value of the dissipation rate is much lower than the given in Figure 5.11(a). When using Q_3/Q_2 elements, the changes on c_c produce lower differences compared against the Q_2/Q_1 approximation. The divergence norm depicted in Figure 5.12 also show improvements with respect to Figure 5.11. In this case, the maximum value of the divergence for c_c for Q_2/Q_1 elements is lower than the case of 64^3 velocity DOFs and also for the case of 32^3 velocity DOFs.



(a) Total energy dissipation rate (b) $\|\nabla \cdot \mathbf{u}_h\|$

FIGURE 5.12: Total energy dissipation and velocity divergence $L2$-norm for different c_c choices for OSS-ISS with 96^3 velocity DOFs

To summarize, we have checked that refining the mesh we converge to the DNS results and, for a given number of DOFs, the results improve when higher order interpolation is used. On the hand, we have seen that increasing c_c we reduce $\|\nabla \cdot \mathbf{u}\|$, and consequently the accuracy of the solution.

5.5.3 Turbulent channel flow at $Re_\tau = 395$

In Section 5.5.2 we have tested an homogeneous turbulent flow. Now, we want to check the behaviour of the proposed OSS-ISS stabilization method for a wall bounded turbulent test. To do so, we use the Turbulent Channel Flow test with a Reynolds number based on the wall friction (Re_τ) equal to 395. This benchmark was exhaustively tested using a VMS method with OSS in previous Chapter 4.

Setting

Recalling the problem description exposed in Section 3.3.2, the domain of the TCF problem for $Re_\tau = 395$ is given by a box of length $(2\pi\delta \times 2\delta \times 2/3\pi\delta)$. The x-direction is the flow direction, also called stream-wise direction, the y-direction is the wall-normal direction, and the z-direction is the span-wise direction. Homogeneous Dirichlet boundary conditions for the velocity DOFs are imposed on wall-normal direction boundaries ($y = -\delta$ and $y = \delta$), while periodic boundary conditions are defined on the stream-wise and span-wise directions.

The problem is solved using two different meshes, with 32^3 $Q_2 - Q_1$ and 21^3 Q_3/Q_2 elements mesh. Both meshes have refined elements near the wall in the wall-normal direction, like the one used in [56]. The algorithmic constants that appear in the OSS-ISS method will be discussed in the following subsections.

The obtained results are compared against a DNS computed in [140, 121] (MKM-DNS), then, the parameter election will be according to the ones defined in the cited paper. The bulk mean velocity and the half channel height are taken equal to one, $\bar{U} = 1$ and $\delta = 1$. Knowing the estimated Reynolds number based on the bulk mean velocity, $Re = \bar{U}2\delta/\nu \approx 13,750$ (see [150]), one can obtain the value of the viscosity, $\nu = 1.4545 \cdot 10^{-4}$. From the Reynolds number based on the friction velocity, we can determine the friction velocity magnitude: $u_\tau = Re_\tau\nu/\delta = 5.745 \cdot 10^{-2}$. Thus, the wall shear stress reads $\tau_w = u_\tau^2 = 3.3010 \cdot 10^{-3}$. A force equivalent to a pressure gradient is imposed to drive the movement of the flow in the stream-wise direction, $f_x = \tau_w/\delta$.

In order to achieve the statistically steady state solution, an initial solution is provided following [139]. This initial solution consists in a unidirectional velocity profile over which is added a fluctuation:

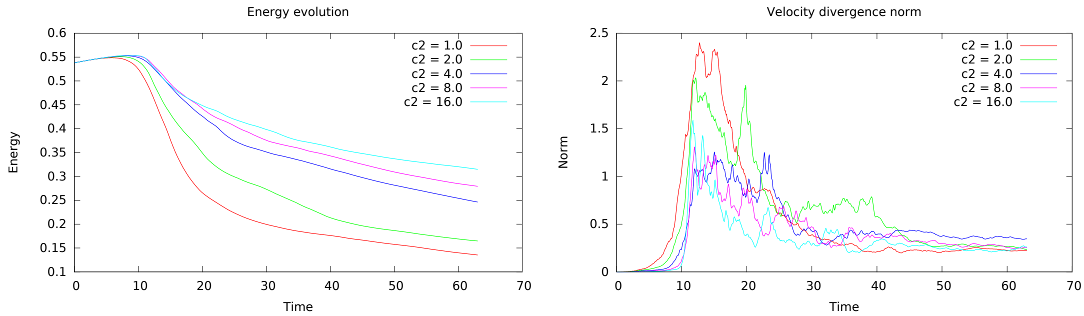
$$\begin{aligned} u_x &= C\left(1 - y^8\right) + \epsilon\frac{L_x}{2}\sin(\pi y)\cos\left(\frac{4\pi x}{L_x}\right)\sin\left(\frac{2\pi z}{L_z}\right), \\ u_y &= -\epsilon(1 + \cos(\pi y))\sin(\pi y)\sin\left(\frac{4\pi x}{L_x}\right)\sin\left(\frac{2\pi z}{L_z}\right), \\ u_z &= -\epsilon\frac{L_z}{2}\sin\left(\frac{4\pi x}{L_x}\right)\sin(\pi y)\cos\left(\frac{2\pi z}{L_z}\right). \end{aligned} \tag{5.38}$$

The constant C is chosen in such a way that the field without fluctuations would have a bulk mean velocity $\bar{U} = 1.0$. The fluctuation constant ϵ is 10% of the bulk mean velocity.

Effect of the pressure subscale term on the conservation of mass

As it has been said in Section 5.3.4, the pressure subscale term has a noticeable effect on the solution when ISS elements are used. The effect of this term has also been analyzed in a previous test, see Section 5.5.2. Here we will also assess the effect of this term on a wall-bounded flow.

First we will focus on the effect of the second term in (5.37), looking how the c_2/c_1 ratio affects the solution keeping $c_c = 1.0$. As we are in a turbulent regime, we do not expect that the viscous counterpart in (2.44) will have relevance on the solution, but we do expect it for the convective counterpart. Then, we will keep $c_1 = 12.0$ and we will increase c_2 from 1.0 to 16.0. The variations on c_2 not only have an effect on τ_c but also on τ_m. We solve the problem from $t = 0$ to $t = 20\pi$ (time needed to cross the channel 10 times, based on the initial mean bulk velocity $\bar{U}$) starting from the initial solution (5.38) using the implicit version of *(3-3)* SRK scheme. The energy and $\|\nabla \cdot \mathbf{u}_h\|$ evolution are plotted in Figure 5.13. It is clearly seen in Figure 5.13 that the modification of c_2 has an effect on the solution. We see in Figure 5.13(a) that the energy drops faster when lower values of c_2 are used. Looking at Figure 5.13(b) we see that the velocity divergence L_2-norm is higher when lower values of c_2 are taken. The evolution of $\|\nabla \cdot \mathbf{u}_h\|$ gives us information about how the flow is evolving. From an initial and structured condition, the flow starts becoming chaotic around $t = 10$, depending on the case, when the turbulent structures increase the rotation of the flow particles, growing the spatial derivatives. At this stage, the energy dissipates until the equilibrium between the internal energy and the external forces is reached. These stages can be seen in Figure 5.14, where we depict the vorticity isosurfaces for $|\omega| = 5.0$ coloured with the velocity field module at $t = 0.15$ (Figure 5.14(a)), at $t = 12.0$ (Figure 5.14(b)) and at $t = 70.0$ (Figure 5.14(d)) setting $c2 = 1.0$ and $c_c = 32.0$, and using a 32^3 Q_2/Q_1 elements mesh.

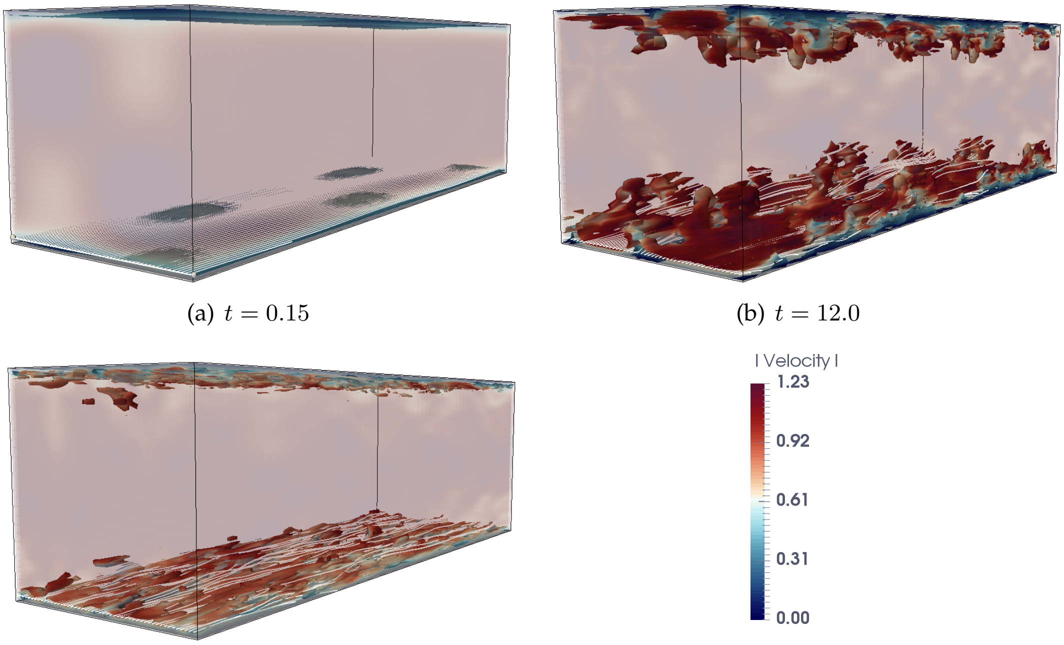


(a) Energy evolution (b) $\|\nabla \cdot \mathbf{u}_h\|$

FIGURE 5.13: Energy evolution and velocity divergence norm for different values of c_2, keeping $c_1 = 12.0$ and $c_c = 1.0$

When increasing c_2 we are decreasing the value of τ_m, but increasing τ_c. This means that in the energy dissipation equation (5.24), the dissipation through the convective term $\|\tau_m^{1/2}(\mathbf{u}_h \cdot \nabla \mathbf{u}_h - \boldsymbol{\eta}_h)\|^2$ becomes less relevant in front of the $\|\tau_c^{1/2}\nabla \cdot \mathbf{u}_h\|^2$.

Now, knowing the influence of c_2, and with the aim to distinguish the effect of modifying τ_m or τ_c, we keep this constant fixed with a value $c_2 = 8.0$ and we analyse the influence of τ_c on the solution, by changing c_c. In this situation, τ_m will remain constant for all cases and we will see the effect of the pressure subscale term. For this test, we solve the problem until $t = 300$, where the energy evolution stabilizes for all c_c choices. In Figure 5.15 we see that the energy evolution (Figure 5.15(a)) and the divergence norm evolution (Figure 5.15(b)) follow the same pattern observed in

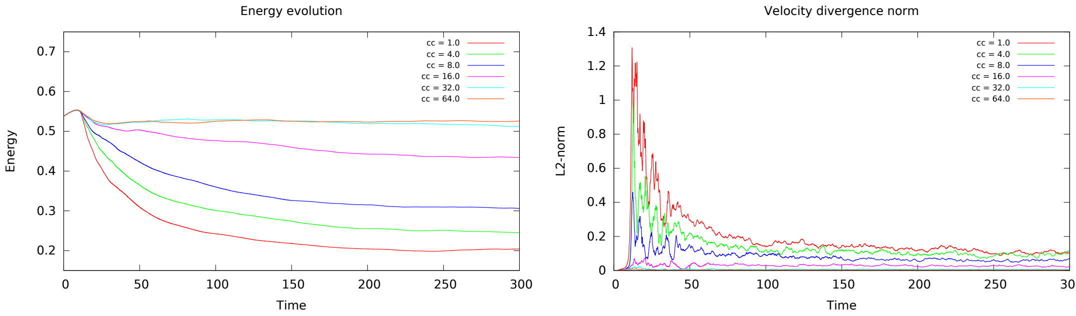


(a) $t = 0.15$

(b) $t = 12.0$

(c) $t = 70.0$

FIGURE 5.14: Velocity module at different times with $c_1 = 12.0$, $c_2 = 1.0$ and $c_c = 32.0$ using a 32^3 Q_2/Q_1 elements mesh.



(a) Energy evolution

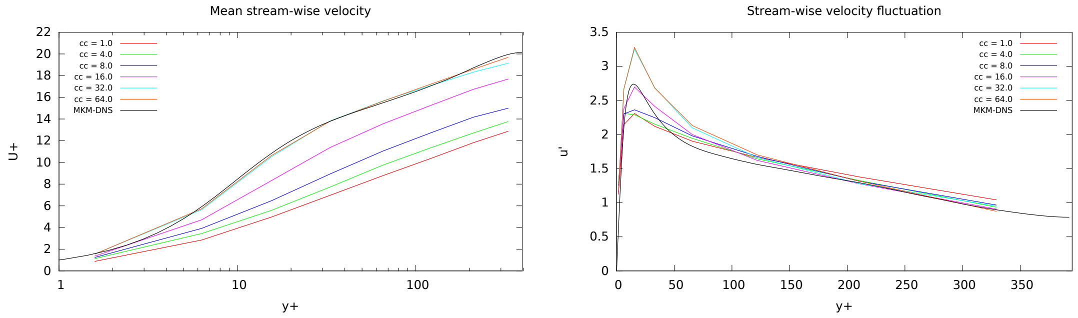
(b) $\|\nabla \cdot \mathbf{u}_h\|$

FIGURE 5.15: Energy evolution and velocity divergence norm for different values of c_c, keeping $c_1 = 12.0$ and $c_2 = 8.0$

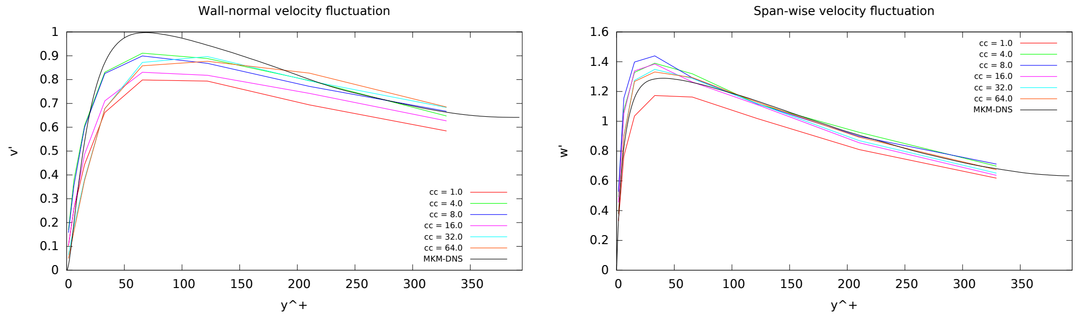
Figure 5.13. In this case we see that the improvement on the mass conservation is more pronounced when increasing c_c than when increasing c_2. We also see in Figure 5.15(a) that there is a certain threshold after which the result is improving very little. In this case, the results for $c_c = 32.0$ and $c_c = 64.0$ are almost the same.

We can compare our results with the MKM-DNS, contrasting the mean velocity and its fluctuations in Figure 5.16. To obtain these results we have solved the problem from $t = 300$ to $t = 330$ with a time step size of $\delta t = 0.03$, collecting 1000 samples to obtain the mean quantities. The mean quantities are computed integrating the desired quantity within each element and adding up all elemental results belonging to the same y-orthogonal plane. Thus, since we are using high-order elements, the number of points that will appear in the graphics will be smaller than the number of DOFs. This procedure was also followed in [78] noting that the integral averaging procedure takes into account information from the secondary nodes that the widely used point-wise averaging does not contemplate.

Figure 5.16(a) depicts the mean stream-wise velocity normalized by the prescribed wall-shear velocity, u_τ. In this picture we see the difference when changing c_c, where the dissipation of energy shown in Figure 5.15(a) becomes clear looking at the velocity magnitude. The increase of $\|\nabla \cdot \mathbf{u}_h\|$, and its consequent loss of mass conservation, results in a mean velocity profile much lower than the DNS. We see that for $c_c = 32.0$ and $c_c = 64.0$ the mean velocity profile is on top of the DNS result, even for the coarse mesh in which we are solving the problem. Looking at the stream-wise velocity fluctuation (Figure 5.16(b)) we see that it increases as we increase c_c. This behaviour is justified by the fact that for low c_c values we have less energy in the system, and then, the fluctuating magnitude is also lower. The wall-normal and span-wise velocity fluctuations also give the smallest fluctuation magnitude for the smallest c_c value, but increasing c_c the fluctuation grows until certain point and then it decreases, becoming closer to the DNS curve.



(a) Mean stream-wise velocity (b) Rms stream-wise velocity fluctuation



(c) Rms wall-normal velocity fluctuation (d) Rms span-wise velocity fluctuation

FIGURE 5.16: Mean stream-wise velocity and rms velocity fluctuations using a 16^3 Q_2/Q_1 mesh for different choices of c_c.

From Figure 5.16 we clearly see that increasing c_c the results improve, but this procedure may have some drawbacks. One of them is the ill-conditioning of the system of equations when $c_c \to \infty$. Table 5.1 summarizes the total accumulated solver iterations as well as the elapsed time needed to solve the problem for the different choices of c_c. As we see in Table 5.1, to solve the TCF problem with $c_c = 64.0$ is a 29.0% more expensive than solving it with $c_c = 1.0$. Thus, it is clear that at some point it could be preferable to refine the mesh rather than increase the c_c value if we want more accurate results.

TABLE 5.1: Solver iterations and elapsed time to solve the TCF problem from $t = 300$ to $t = 330$ for different c_c values.

c_c	Solver iterations	Elapsed time (s)	Increment in time (%)
1.0	397899	19238	0.0
4.0	402490	19674	2.27
8.0	422217	19987	3.89
16.0	466687	20803	8.13
32.0	539879	22794	18.48
64.0	626487	24817	29.00

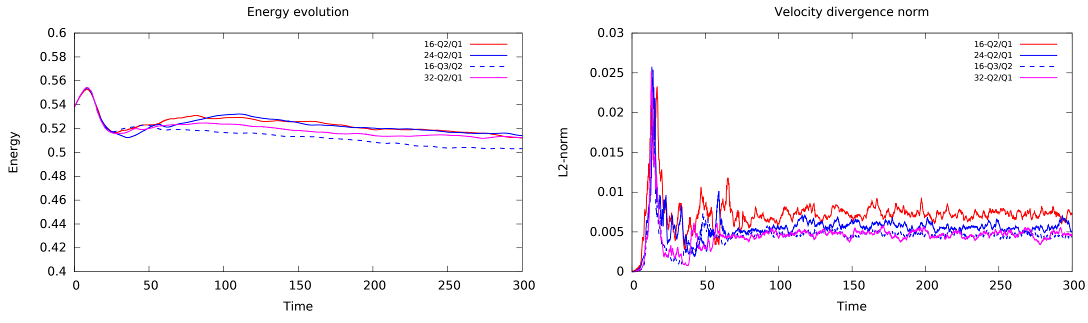
Provided that in all results shown until now, the case $c_c = 32.0$ gives almost the same solution as the $c_c = 64.0$ case, hereinafter we will only consider the case in which the parametric constants are $c_1 = 12.0$, $c_2 = 8.0$ and $c_c = 32.0$.

Refinement for a given c_c

Let us now explore what happens when the mesh is refined in the TCF test. Here we consider four different discretizations changing both the element size and the order of interpolation: 16^3 Q_2/Q_1, 24^3 Q_2/Q_1, 16^3 Q_3/Q_2 and 32^3 Q_2/Q_1. These discretizations have 32^3, 48^3, 48^3 and 64^3 velocity DOFs, respectively.

Looking at the energy evolution (Figure 5.17(a)) and the velocity divergence L_2-norm (Figure 5.17(b)) depicted in Figure 5.17 we see that there are very little differences between the cases considered in this section. This means that given the appropriate parametric constants, the energy evolves in a similar way for all discretizations, with a similar evolution of the mass conservation.

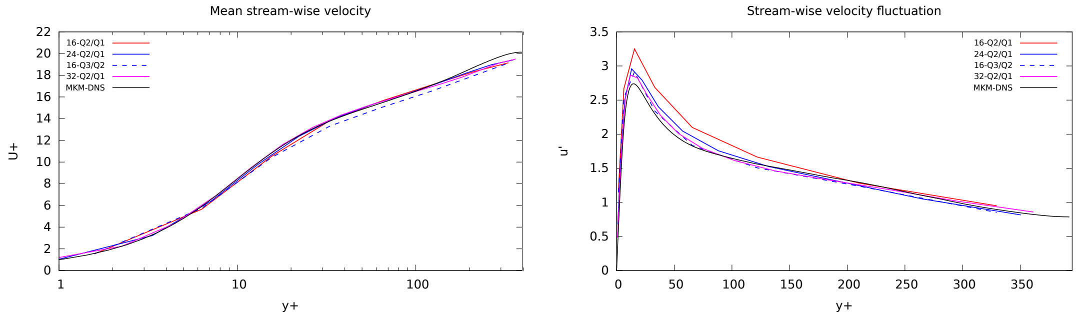
If we focus on the averaged turbulent quantities shown in Figure 5.18, we also see similar results between the different cases. In Figure 5.18(a) the mean stream-wise velocity is plotted, and it is seen that all methods are almost on top of the DNS curve. The 16^3 Q_3/Q_2 discretization gives a mean velocity profile slightly lower than the other discretizations, which is also reflected on the energy evolution shown in Figure 5.17(a). If we look at the velocity fluctuations in span-wise, wall-normal, and span-wise directions depicted in Figure 5.18(b), Figure 5.18(c) and Figure 5.18(d), respectively, the convergence to the DNS solution is clear.



(a) Energy evolution

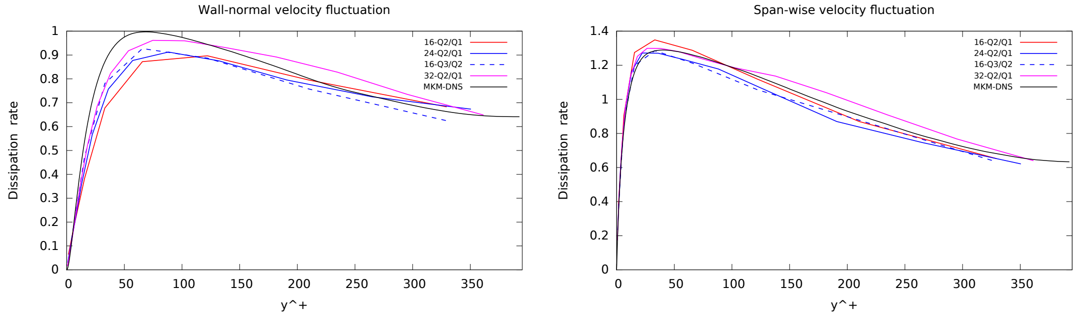
(b) $\|\nabla \cdot \mathbf{u}_h\|$

FIGURE 5.17: Energy evolution and velocity divergence norm refining the mesh, keeping $c_1 = 12.0$, $c_2 = 8.0$ and $c_c = 32.0$



(a) Mean stream-wise velocity

(b) Rms stream-wise velocity fluctuation



(c) Rms wall-normal velocity fluctuation

(d) Rms span-wise velocity fluctuation

FIGURE 5.18: Mean stream-wise velocity and rms velocity fluctuations for using different discretizations.

5.6 Conclusions

In this chapter we have considered the numerical simulation of turbulent incompressible flows using the term-by-term OSS method for convection-stabilization applied to ISS elements. We have considered an implicit treatment of the projection. For the solution of the monolithic linear systems a block preconditioning strategy that makes use of recursive block factorizations has been proposed. Among the three variants we proposed, namely diagonal, lower and upper triangular, the last two are much faster than the first one, as expected, the upper triangular being slightly faster for the problems considered.

Using this strategy, the comparison of the three methods, ASGS, term-by-term OSS, and convection-only OSS with ISS elements reveals that the accuracy is similar for the same order of interpolation of the velocity, the OSS-ISS being slightly inferior in this respect. But on the other hand, when computational cost is analyzed the OSS-ISS is clearly the cheapest one so a finer discretization can be used. Another advantage of the OSS-ISS method is that it does not change the nature in time of the problem, i.e., the discrete system is an index-2 differential-algebraic equation. Thus, it allow us, e.g., to use Segregated Runge-Kutta (SRK) time integration schemes that cannot be applied to the ASGS or OSS methods. This issue will be discussed in further works, but we can say that SRK schemes are well suited to perform high-order time integration as well as the use of an adaptive time-stepping technique. This method was firstly introduced and tested for laminar incompressible flows in [57] and will be discussed in forthcoming chapters.

Finally, we have also analyzed the influence of the grad-div stabilization on the results when ISS discretizations are considered. It has been clearly shown that this term affects the solution. Increasing the value of c_c results in a better mass conservation and better results but this saturates at some point from which little improvements are achieved while we have a high increase on the computational cost. Moreover, this effect is smaller when the mesh is refined and also when higher order interpolations are used.

Chapter 6

Segregated Runge-Kutta time integration schemes

6.1 Introduction

Many problems in science and engineering can be simulated by incompressible flow solvers, e.g., flows around aircrafts during take-off and landing, cars, bridges, wind turbines, etc. Incompressible flows are also encountered when simulating liquid metal blankets in fusion reactors or blood flow in bio-mechanics. The increasing computer power of super-computers has motivated the interest in high-fidelity massively parallel predictive tools on unstructured meshes for this type of applications. Higher levels of accuracy in space can be based on refined meshes or higher-order approximations, being the use of hp-adaptive simulations the most refined approach so far.

The transient incompressible Navier-Stokes system of partial differential equations is nonlinear (due to convection) and indefinite (due to the divergence-free constraint), which complicates its discretization and the linear solver step. A fully implicit time-integration involves nonlinear iterations at every time step, increasing computational cost. On the other hand, it is hard to define scalable parallel solvers for non-symmetric and indefinite problems. The definition of scalable preconditioners for this nonlinear system of equations is an open problem both for domain decomposition and multigrid techniques. Further, the nonlinear nature of the problem requires frequent preconditioner set-up steps, that make this approach computationally intensive.

The velocity-pressure block-segregation can be understood as a solver (introducing an additional splitting error) instead of a preconditioner, leading to the popular pressure-correction or fractional-step methods [120, 128, 53, 12]. This approach involves to solve decoupled a momentum equation for the velocity and a pressure Poisson equation. This is the most popular approach for the simulation of turbulent incompressible flows. The time integration of the momentum equation can be carried out using explicit, semi-implicit, or fully implicit methods. The fully implicit method has some of the drawbacks considered above, whereas an explicit integration of the viscous terms is not suitable for wall-bounded flows. In order to capture the viscous effects around solids, very refined anisotropic meshes are required, leading to too stringent viscous Courant-Friedrichs-Lewy (CFL) conditions. The use of hp-adaptivity makes hard to use explicit methods, since intensive local refinement in some parts, e.g., the tip of an airfoil, leads to global time steps that must go to zero as $\frac{h^2}{p^4}$ for stability purposes, being h the characteristic element size and p the degree of interpolation, and local time-stepping cannot be efficiently exploited on parallel platforms. The use of semi-implicit methods, which treat implicitly the diffusive term and explicitly the convective one, seems to be the perfect compromise for turbulent flows around objects and viscous flows. The time step restriction is given by the convective CFL condition, which is much weaker than the diffusive one and also avoids nonlinear iterations. In fact, the majority of direct and large eddy simulations of incompressible flows involve semi-implicit methods, in which only the viscous term is treated implicitly.

When high-order time integration aims to be achieved, a popular approach is to use Runge-Kutta (RK) schemes, due to the good stability properties, high-order accuracy, and the easy computation of time error estimates for adaptive time stepping (see [90, 63]). RK schemes involve several systems of equations at each time step. Further, when we use an implicit scheme all stages can be coupled, resulting in a large system of equations to be solved, which is quite impractical in terms of CPU cost. This is the case of [158], where energy-conserving implicit Runge-Kutta methods are investigated. This drawback can be bypassed using an explicit scheme, with the problems related by the time step restriction to ensure stability. An accuracy analysis of these methods is done by Sanderse et al in [159]. Diagonally Implicit RK methods (DIRK) can be used to avoid stability problems and solving implicitly each RK stage uncoupled (see, e..g, [135]). These type of method were tested by Marx in [135] and turned out to be the best compromise in accuracy, efficiency and robustness among the different time integration schemes investigated in that work.

Due to the differential-algebraic nature of the ordinary differential system that arises from the spatial discretization, the application of RK methods to the Navier-Stokes equations is not straightforward. The typical approach is to compute the velocity at the next time step by integrating the momentum equation using some RK method (freezing the pressure gradient term), and next recover the pressure using a pressure Poisson equation (see, e.g., [142]). However, it is unclear how this approach affects the convergence error of both velocities and pressures. Alternatively, other methods perform a RK time-integration in which the velocity at every stage is enforced to be divergence-free, and next a pressure segregation is applied at every stage [128, 122]. As a result, the error due to time RK discretization is spoiled by a second-order pressure splitting error. It is common in the literature not to report pressure error in time [142, 117] or to report at most second-order of accuracy [122]. An exception to this situation is the recent work [159], where the half-explicit RK (HERK) methods for index-2 algebraic-differential equation (DAE) systems (see [92]) have been applied to the incompressible Navier-Stokes equations. These methods provide error estimates for both velocities and pressures, but require an explicit treatment of both convection and diffusion terms. Other approaches include the energy-conserving implicit RK methods in [158] (coupling all stages at the linear system).

A very accurate pressure is required in many applications, especially those involving fluid-structure interaction for high Reynolds number flows, and when evaluating drag and lift coefficients on objects. With the aim to develop high-order semi-implicit methods, we propose new time-integration schemes, that we will denote as *segregated Runge-Kutta* (SRK) methods, that do not involve any additional splitting error, since the pressure-velocity decoupled computation is already obtained at the time-integration level. These methods are motivated from the projected momentum equation onto the space of divergence-free functions, which allows us to eliminate the pressure and consider general RK schemes for the time integration. This way, we can easily prove the order of the pressure time error and attain higher than two schemes for the pressure too. The benefit of this approach with respect to HERK methods is the flexibility to consider implicit and implicit-explicit (IMEX) versions of these methods.

In order to ease an effective preconditioning, we will favour SRK schemes that treat implicitly the viscous term, explicitly the convective terms, and segregated the pressure. The use of IMEX-SRK methods is very appealing for large-scale computations. At every stage, it only involves a vector-Laplacian (or elasticity-type) plus a mass matrix solver for the velocity and a discrete pressure Poisson solver. For this type of coercive and symmetric problems, we can make use of efficient and highly scalable domain decomposition or multigrid algorithms (see, e.g., [20]). Further, the set-up of these preconditioners can be kept on fixed meshes, reducing computational cost.

The statement of the incompressible Navier-Stokes equations is developed in Section 6.2. In Section 6.3 the time integration through RK schemes is introduced, giving an overview of the HERK methods and developing the proposed SRK schemes. Four different tests are exposed in Section 6.4, where the application of SRK schemes is assessed for two different manufactured analytical solutions and laminar and turbulent flow tests. Finally, some conclusions are stated in Section 6.5.

6.2 Problem statement

We start this section by briefly describing the Navier-Stokes problem, referring to Section 2.1 for a deep description of the problem statement. Let Ω be a bounded domain of $\mathbb{R}^d$, where $d = 2, 3$ is the number of space dimensions, $\Gamma = \partial\Omega$ its boundary and $(0, T]$ the time interval. The strong form of the incompressible Navier-Stokes problem consists of finding a velocity field $\mathbf{u}$ and a pressure p such that

$$\partial_t \mathbf{u} - \nabla \cdot (\nu(\nabla \mathbf{u} + \nabla \mathbf{u}^T)) + \mathbf{u} \cdot \nabla \mathbf{u} + \nabla p = \mathbf{f} \qquad \text{in } \Omega \times (0, T], \tag{6.1}$$

$$\nabla \cdot \mathbf{u} = 0 \qquad \text{in } \Omega \times (0, T], \tag{6.2}$$

with $\mathbf{f}$ the force vector and ν the kinematic viscosity. Recalling that bold characters denote vectors and tensors. Equations (6.1) and (6.2) need to be supplied with appropriate boundary and initial conditions. The boundary Γ is divided into the Dirichlet (Γ_D) and the Neumann (Γ_N) parts such that $\Gamma_D \cup \Gamma_N = \Gamma$ and $\Gamma_D \cap \Gamma_N = \emptyset$. Then, the boundary and initial conditions can be written as

$$\mathbf{u} = \mathbf{u}_g \qquad \text{on } \Gamma_D \times (0, T], \tag{6.3}$$

$$(-p \cdot \mathbf{I} + \nu(\nabla\mathbf{u} + \nabla\mathbf{u}^T)) \cdot \mathbf{n} = \mathbf{t}_N \qquad \text{on } \Gamma_N \times (0, T], \tag{6.4}$$

$$\mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0(\mathbf{x}) \qquad \text{in } \Omega \times \{0\}, \tag{6.5}$$

$\mathbf{n}$ being the unit outward vector normal to Γ.

Using the notation defined in Section 2.2, the weak form of the transient incompressible Navier-Stokes problem (6.1)-(6.5) reads as follows: find $[\mathbf{u}(t), p(t)] \in H_0^1(\Omega) \times L_0^2(\Omega)$ such that

$$(\partial_t \mathbf{u}, \mathbf{v}) + B(\mathbf{u}, (\mathbf{u}, p), (\mathbf{v}, q)) = \langle \mathbf{f}, \mathbf{v} \rangle + (g, q) \qquad \text{for any } [\mathbf{v}, q] \in H_0^1(\Omega) \times L_0^2(\Omega), \tag{6.6}$$

almost everywhere in time, satisfying the initial condition (6.5) in a weak sense, where the form $B(\mathbf{a}, (\mathbf{u}, p), (\mathbf{v}, q))$ is defined as

$$B(\mathbf{a}, (\mathbf{u}, p), (\mathbf{v}, q)) := \nu(\nabla\mathbf{u}, \nabla\mathbf{v}) + b(\mathbf{a}, \mathbf{u}, \mathbf{v}) - (p, \nabla \cdot \mathbf{v}) + (q, \nabla \cdot \mathbf{u}), \tag{6.7}$$

and $b(\mathbf{a}, \mathbf{u}, \mathbf{v})$ is the trilinear weak form of the convective term.

Let us now consider a quasi-uniform FE partition $\mathcal{T}_h$ of the domain Ω from which we can construct the finite dimensional spaces for the velocity and pressure. After the discretization in space of (6.6), we end up with an index-2 DAE system of equations:

$$\mathbb{M}\dot{\mathbf{U}} + (\mathbb{K} + \mathbb{C}(\mathbf{U}))\mathbf{U} + \mathbb{G}\mathbf{P} = \mathbf{F}, \tag{6.8}$$

$$\mathbb{D}\mathbf{U} = \mathbf{H}, \tag{6.9}$$

where $\mathbb{M}$ is the mass matrix, $\mathbb{K}$ the contribution of the diffusion term, $\mathbb{C}(\mathbf{U})$ the nonlinear convective term (related to the trilinear form b), $\mathbb{G}$ the pressure gradient operator and $\mathbb{D}$ the divergence matrix (note that $\mathbb{D} = -\mathbb{G}^T$). $\mathbf{U}$ and $\mathbf{P}$ are the nodal values of the discrete velocity and pressure, while $\mathbf{F}$ and $\mathbf{H}$ are the force terms of the momentum and incompressibility constraint equations, respectively.

Focusing on the matrix system (6.8)-(6.9), if we derive with respect to the time equation (6.9), we have that $\mathbb{D}\dot{\mathbf{U}} = \dot{\mathbf{H}}$, assuming that $\mathbb{D}$ is constant in time. Then, multiplying the first equation (6.8) by $\mathbb{D}\mathbb{M}^{-1}$ and invoking this result, we obtain an alternative equation for $\mathbf{U}$ and $\mathbf{P}$.

$$\dot{\mathbf{H}} + \mathbb{D}\mathbb{M}^{-1}(\mathbb{K} + \mathbb{C}(\mathbf{U}))\mathbf{U} + \mathbb{D}\mathbb{M}^{-1}\mathbb{G}\mathbf{P} = \mathbb{D}\mathbb{M}^{-1}\mathbf{F}. \tag{6.10}$$

Assuming that $\mathbb{D}\mathbf{U}(0) = \mathbf{H}(\mathbf{0})$, systems (6.8)-(6.9) and (6.8)-(6.10) are equivalent. Further, matrix $\mathbb{D}\mathbb{M}^{-1}G$ is invertible, due to the inf-sup condition to be satisfied by the mixed FE space (see [69]). As a result, the pressure can be expressed in terms of $\mathbf{U}$ using (6.10), getting

$$-\mathbb{D}\mathbb{M}^{-1}\mathbb{G}\mathbf{P} = \mathbb{D}\mathbb{M}^{-1}(\mathbb{K} + \mathbb{C}(\mathbf{U})\mathbf{U} - \mathbf{F}) + \dot{\mathbf{H}}. \tag{6.11}$$

Replacing this expression in (6.8), we obtain an equation for the velocity field only:

$$\mathbb{M}\dot{\mathbf{U}} + \Pi(\mathbb{K} + \mathbb{C}(\mathbf{U}))\mathbf{U} = \Pi\mathbf{F} + \mathbb{G}(\mathbb{D}\mathbb{M}^{-1}\mathbb{G})^{-1}\dot{\mathbf{H}}, \tag{6.12}$$

with

$$\Pi := (\mathbb{I} - \mathbb{G}(\mathbb{D}\mathbb{M}^{-1}\mathbb{G})^{-1}\mathbb{D}\mathbb{M}^{-1}),$$

$\mathbb{I}$ being the identity matrix. This system matrix stands for the projected Navier-Stokes system onto the discrete divergence-free space; we can easily check that $\mathbb{D}\mathbb{M}^{-1}\Pi = 0$, readily leading to $\mathbb{D}\dot{\mathbf{U}} = \dot{\mathbf{H}}$. It is a system of ordinary differential equations (ODEs) and the use of RK methods is now straightforward.

6.3 Runge-Kutta time integration

Let us consider now the Navier-Stokes semi-discrete problem given by equations (6.8)-(6.9) or (6.8) and (6.10). We consider a space discretization using mixed FE spaces satisfying the inf-sup condition. This way, we avoid the use of stabilized formulations that involve extra terms that may couple pressure and velocity fields or fill the diagonal block related to the pressure, and change the mathematical structure of the system.

Following the motivation in Section 6.1, we aim to develop RK schemes for the incompressible Navier-Stokes equations that will *segregate the velocity and pressure computation while keeping high order of accuracy*. This splitting leads to the use of optimal solvers for the velocity block and the pressure block, respectively, without the need to develop efficient and scalable algorithms for indefinite systems. Further, when considering explicitly the convective term, we can maintain the same preconditioner at all stages (while the mesh does not change) and avoid the need to deal with non-symmetric (and possibly convection-dominant) systems.

A RK scheme consists of a multistage integration in which each stage is computed as a combination of the unknowns evaluated in other stages. This combination can give an implicit scheme or an explicit scheme, depending on the definition of the Butcher tableau. Implicit and explicit schemes can be combined, leading to IMEX schemes, i.e., different Butcher tableaus are used for the implicit and explicit terms (see C). System (6.8)-(6.9) can be compactly written as

$$\mathbb{M}\dot{\mathbf{U}} = \mathcal{F}(\mathbf{U}) + \mathcal{G}(\mathbf{U}, \mathbf{P}), \qquad \mathbb{D}\mathbf{U} = \mathbf{H}, \tag{6.13}$$

where $\mathcal{F}(\mathbf{U})$ and $\mathcal{G}(\mathbf{U},\mathbf{P})$ are the terms to be treated implicitly and explicitly, respectively. For the implicit integration of $\mathcal{F}$, we will use the so called DIRK method; for a given stage i, it only involves the stages j such that $1 \leq j \leq i$. For the explicit integration of $\mathcal{G}$, in a given stage i, the method only concerns about the contribution of the stages j such that $1 \leq j \leq i-1$.

6.3.1 Half-explicit Runge-Kutta schemes

The first approach to get a RK time integration with the properties described above is the use of half-explicit RK (HERK) methods (see [33, 5, 141] for the general definition of these methods and [159] for its very recent application to the Navier-Stokes equations). So, we consider $\mathcal{F} = 0$. HERK methods combine an explicit RK method for the momentum equation with an implicit enforcement of the discrete divergence constraint. A half-explicit integration of the Navier-Stokes equations reads as:

$$\frac{1}{\delta t}\mathbb{M}\mathbf{U}_i = \frac{1}{\delta t}\mathbb{M}\mathbf{U}_n + \sum_{j=1}^{i-1} \hat{a}_{ij}\mathcal{G}(\mathbf{U}_j,\mathbf{P}_j), \qquad \mathbb{D}\mathbf{U}_i = \mathbf{H}(\mathbf{t_i}), \tag{6.14}$$

where $t_i := t_n + \hat{c}_i\delta t$. We observe that the computation of $\mathbf{U}_i$ at every stage does only depend on $\mathbf{P}_j$ with $j = 1,\ldots,i-1$. Applying $\mathbb{D}\mathbb{M}^{-1}$ over the momentum equation and recalling the discrete divergence constraint, we obtain the equivalent method

$$\frac{1}{\delta t}\mathbb{M}\mathbf{U}_i = \frac{1}{\delta t}\mathbb{M}\mathbf{U}_n + \sum_{j=1}^{i-1} \hat{a}_{ij}\mathcal{G}(\mathbf{U}_j,\mathbf{P}_j), \qquad \mathbb{D}\mathbf{U}_n + \delta t\sum_{j=1}^{i-1} \hat{a}_{ij}\mathbb{D}\mathbb{M}^{-1}\mathcal{G}(\mathbf{U}_j,\mathbf{P}_j) = \mathbf{H}(\mathbf{t_i}). \tag{6.15}$$

We can easily check that the second equation is a linear system for $\mathbf{P}_{i-1}$ with the system matrix $\mathbb{D}\mathbb{M}^{-1}\mathbb{G}$ (see [159] for different implementations). At the end of the multi-stage computation, we update the velocity field:

$$\frac{\mathbb{M}}{\delta t}\mathbf{U}_{n+1} = \frac{\mathbb{M}}{\delta t}\mathbf{U}_n + \sum_{i=1}^{s} \hat{b}_i\mathcal{G}(\mathbf{U}_i,\mathbf{P}_i), \qquad \mathbb{D}\mathbf{U}_{n+1} = \mathbf{H}(\mathbf{t_{n+1}}). \tag{6.16}$$

At the velocity update, we compute the last stage pressure $\mathbf{P}_s$ as above. $\mathbf{P}_{n+1}$ does not appear in the definition of the method, but it can easily be defined using a pressure Poisson equation (see [159] for different alternatives).

However, when considering some implicit terms, the implicit treatment of the constraint in the spirit of HERK methods is not affordable. For instance, treating the diffusive term implicitly, it would involve the system matrix $\mathbb{D}(\mathbb{M} + \delta t\mathbb{K})^{-1}\mathbb{G}$ for the pressure. So, the extension of this approach to implicit and IMEX integration schemes for the momentum equation is not feasible. It has motivated the schemes introduced below.

6.3.2 Segregated Runge-Kutta schemes

In order to get implicit or IMEX RK schemes for the incompressible Navier-Stokes equations, we consider the velocity-only projected system (6.12), which is just an ODE system that can straightforwardly be integrated using RK schemes. Let us write this problem in compact form as:

$$\mathbb{M}\dot{\mathbf{U}} = \mathcal{F}(\mathbf{U}) + \mathcal{G}(\mathbf{U}). \tag{6.17}$$

In particular, we can define a method in which the viscous term is treated implicitly and the convective and pressure-related term are treated explicitly. One choice is to define the operators $\mathcal{F}$ and $\mathcal{G}$ as

$$\mathcal{F}(\mathbf{U}) := -\mathbb{K}\mathbf{U} \qquad \text{and} \tag{6.18}$$

$$\mathcal{G}(\mathbf{U}) := \mathbf{F} - \mathbb{C}(\mathbf{U})\mathbf{U} + \mathbb{G}(\mathbb{D}\mathbb{M}^{-1}\mathbb{G})^{-1}\left(\mathbb{D}\mathbb{M}^{-1}(\mathbb{K} + \mathbb{C}(\mathbf{U})\mathbf{U} - \mathbf{F}) + \dot{\mathbf{H}}\right), \tag{6.19}$$

i.e., using an implicit treatment of the viscous term and an explicit one for the convective and forcing term. We note that the evaluation of the pressure-related term involves a discrete pressure Poisson equation $\mathbb{D}\mathbb{M}^{-1}\mathbb{G}$. However, in order to have a segregated RK method, *the term involving* $\mathbb{D}\mathbb{M}^{-1}\mathbb{G}$ *is treated always explicitly*. Alternatively, we could choose other definitions of $\mathcal{F}$ and $\mathcal{G}$, e.g., the convective term and the force term could be considered implicitly, leading to

$$\mathcal{F}(\mathbf{U}) := -\mathbb{K}\mathbf{U} + \mathbf{F} - \mathbb{C}(\mathbf{U})\mathbf{U} \qquad \text{and} \tag{6.20}$$

$$\mathcal{G}(\mathbf{U}) := \mathbb{G}(\mathbb{D}\mathbb{M}^{-1}\mathbb{G})^{-1}\left(\mathbb{D}\mathbb{M}^{-1}(\mathbb{K} + \mathbb{C}(\mathbf{U})\mathbf{U} - \mathbf{F}) + \dot{\mathbf{H}}\right). \tag{6.21}$$

IMEX-SRK methods could also be of especial interest in turbulent flows in which the time step restriction due to the convective CFL is in most situations smaller than the one needed to capture the smallest time scales in the flow (see for instance [179, 181]).

Considering a RK method with s stages, the velocity at the stage i, $\mathbf{U}_i$, for $1 \le i \le s$ is computed as

$$\frac{1}{\delta t}\mathbb{M}\mathbf{U}_i = \frac{1}{\delta t}\mathbb{M}\mathbf{U}_n + \sum_{j=1}^{i} a_{ij}\mathcal{F}(\mathbf{U}_j) + \sum_{j=1}^{i-1} \hat{a}_{ij}\mathcal{G}(\mathbf{U}_j), \tag{6.22}$$

where a_{ij} and $\hat{a}_{ij}$ are the coefficients of the implicit and explicit Butcher tableau, respectively. After some manipulation, we can rewrite (6.22) as

$$\frac{1}{\delta t}\mathbb{M}\mathbf{U}_i = \frac{1}{\delta t}\mathbb{M}\mathbf{U}_n + \sum_{j=1}^{i} a_{ij}\mathcal{F}(\mathbf{U}_j) + \sum_{j=1}^{i-1} \hat{a}_{ij}\mathcal{G}(\mathbf{U}_j, \mathbf{P}_j), \tag{6.23a}$$

$$-\mathbb{D}\mathbb{M}^{-1}\mathbb{G}(\mathbf{P}_i) = \mathbb{D}\mathbb{M}^{-1}((\mathbb{K} + \mathbb{C}(\mathbf{U}_i))\mathbf{U}_i - \mathbf{F}(t_i)) + \dot{\mathbf{H}}(t_i). \tag{6.23b}$$

For the choice of the operator in (6.18), we would define $\mathcal{G}(\mathbf{U},\mathbf{P}) := \mathbf{F} - \mathbb{C}(\mathbf{U})\mathbf{U} - \mathbb{G}\mathbf{P}$, leading to an IMEX-SRK scheme, whereas the choice in (6.20) is obtained with $\mathcal{G}(\mathbf{U},\mathbf{P}) := -\mathbb{G}\mathbf{P}$ and corresponds to the fully implicit SRK scheme.

At the end of the multi-stage computation, we update the velocity and pressure fields as it can be shown in the following equations:

$$\frac{\mathbb{M}}{\delta t}\mathbf{U}_{n+1} = \frac{\mathbb{M}}{\delta t}\mathbf{U}_n + \sum_{i=1}^{s} b_i \mathcal{F}(\mathbf{U}_i) + \sum_{i=1}^{s} \hat{b}_i \mathcal{G}(\mathbf{U}_i,\mathbf{P}_i), \tag{6.24a}$$

$$-\mathbb{D}\mathbb{M}^{-1}\mathbb{G}(\mathbf{P}_{n+1}) = \mathbb{D}\mathbb{M}^{-1}((\mathbb{K}+\mathbb{C}(\mathbf{U}_{n+1}))\mathbf{U}_{n+1} - \mathbf{F}(t_{n+1})) + \dot{\mathbf{H}}(t_{n+1}). \tag{6.24b}$$

Due to the fact that the resulting method involves a segregated computation of velocity and pressure, it is coined as *Segregated Runge-Kutta* (SRK) method. In this approach, we can naturally consider the viscous and/or convective term implicitly, while keeping a simple pressure Poisson equation. On the other hand, $\mathbf{P}_{n+1}$ is already defined, which is a difference compared to HERK methods.

Remark 6.3.1. *Note that Eq. (6.23b) (respectively, (6.24b)) is equivalent to solve a Darcy-type problem, with the following expression:*

$$\begin{bmatrix} \mathbb{M} & \mathbb{G} \\ \mathbb{D} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{U}^* \\ \mathbf{P}_k \end{bmatrix} = \begin{bmatrix} \mathbf{F}(t_k) - (\mathbb{K}+\mathbb{C}(\mathbf{U}_k))\mathbf{U}_k \\ \dot{\mathbf{H}}(t_k) \end{bmatrix}, \tag{6.25}$$

with k being i (respectively, $n+1$), and $\mathbf{U}^*$ an auxiliar velocity field, which satisfies the discrete incompressibility constraint. System (6.25) can be easily preconditioned by the spectrally equivalent matrix $\mathrm{diag}(\mathbb{M},\tilde{\mathbb{L}})$*, where $\tilde{\mathbb{L}}$ is in turn an optimal (and scalable) preconditioner of the Laplacian matrix [12, 69]. For large scale simulations, $\tilde{\mathbb{L}}$ can be, e.g., an extremely scalable balancing domain decomposition preconditioner for the Poisson problem [19, 20, 17].*

In the SRK methods the discrete divergence constraint $\mathbb{D}\mathbf{U} = \mathbf{H}$ is not explicitly enforced, a difference with respect to HERK methods. However, it is implicitly enforced by the pressure Poisson equation. Let us remind that both equations lead to equivalent systems at the continuous level. In the next proposition we analyze the equivalence between the HERK method and the fully explicit version of the scheme (6.23)-(6.24), i.e., taking $\mathcal{F} = 0$.

Proposition 6.3.1. *Let us assume that $\mathbf{H}$ is independent of time, the initial condition satisfies $\mathbb{D}\mathbf{U}_0 = \mathbf{H}$, and the SRK scheme (6.23)-(6.24) is fully explicit, i.e., $\mathcal{F} = 0$. Then, the HERK scheme (6.14)-(6.15) and the SRK scheme (6.23)-(6.24) are equivalent.*

Proof. Let us assume that $\mathbb{D}\mathbf{U}_n = \mathbf{H}$. Both methods start with $\mathbf{U}_1 = \mathbf{U}_n$ at stage 1, see C where this condition is exposed. The SRK method also computes the pressure $\mathbf{P}_1$ as the solution of $\mathbb{D}\mathbb{M}^{-1}\mathcal{G}(\mathbf{U}_1,\mathbf{P}_1) = 0$. At the second stage, the HERK method computes

$$\mathbb{M}\mathbf{U}_2 = \mathbb{M}\mathbf{U}_n + \hat{a}_{21}\delta t \mathcal{G}(\mathbf{U}_1,\mathbf{P}_1), \qquad \mathbb{D}\mathbf{U}_2 = \mathbf{H}.$$

The constraint leads to the pressure equation $\mathbb{DM}^{-1}\mathcal{G}(\mathbf{U}_1, \mathbf{P}_1) = 0$. As a result, $\mathbf{P}_1$ and $\mathbf{U}_2$ are identical for both methods, the difference being the fact that $\mathbf{P}_1$ is computed at stage 1 in SRK and at stage 2 in HERK. Next, we proceed by induction. Let us assume that both methods are equivalent till stage $i-1$, i.e., we obtain the same $\mathbf{U}_j$ for $j = 1, \ldots, i-1$ and $\mathbf{P}_j$ for j for $j = 1, \ldots, i-2$. At stage i, the HERK method computes first $\mathbf{P}_{i-1}$. Since $\mathbb{D}\mathbf{U}_n = \mathbf{H}$, and using the fact that $\mathbb{DM}^{-1}\mathcal{G}(\mathbf{U}_j, \mathbf{P}_j) = 0$ for $j = 1, \ldots, i-2$ (due to the equivalence with the SRK method and Eq. (6.23)), we finally get $\mathbb{DM}^{-1}\mathcal{G}(\mathbf{U}_{i-1}, \mathbf{P}_{i-1}) = 0$. As a result, $\mathbf{P}_{i-1}$ is the same as the one obtained with SRK. Since the velocity steps at (6.14) and (6.22) are identical for both methods, we also get the same $\mathbf{U}_i$. Within SRK, the pressure $\mathbf{P}_s$ at the last stage is computed from $\mathbb{DM}^{-1}\mathcal{G}(\mathbf{U}_s, \mathbf{P}_s) = 0$. The velocity update in both cases is also identical. Further, using the velocity update at (6.15) and proceeding as above, we can also check that the pressure at the last stage of HERK also satisfies this equation. As a result of the equivalence, we note that $\mathbb{D}\mathbf{U}_{n+1} = \mathbf{H}$ also holds for SRK. The initial assumption holds for the first time step, since $\mathbb{D}\mathbf{U}_0 = \mathbf{H}$. As a result, $\mathbb{D}\mathbf{U}_1 = \ldots = \mathbb{D}\mathbf{U}_n = \mathbf{H}$ holds, proving the proposition. □

This result has another implication. Using the SRK method, we also preserve the discrete divergence constraint exactly in many situations of interest. Now, the question that arises is whether we have this property too when $\mathcal{F} \neq 0$. We analyze the fulfillment of the discrete divergence constraint for SRK.

Proposition 6.3.2. *Let us assume that every component of* $\mathbf{H}(\mathbf{t})$ *is a* p*-th order polynomial in time, the initial condition satisfies* $\mathbb{D}\mathbf{U}_0 = \mathbf{H}(\mathbf{0})$*, the RK integrator integrates exactly polynomials of order* $p-1$*, and* $b_i = \hat{b}_i$ *for* $i = 1, \ldots, s$*,* s *being the number of stages of the scheme. Then, the SRK method preserves the exact discrete divergence constraint at all time steps.*

Proof. We assume that $\mathbb{D}\mathbf{U}_n = \mathbf{H}(\mathbf{t_n})$. The equation $\mathbb{DM}^{-1}(\mathcal{F}(\mathbf{U}_i) + \mathcal{G}(\mathbf{U}_i, \mathbf{P}_i)) = \dot{\mathbf{H}}(t_i)$ holds at every stage of the SRK method. Applying $\mathbb{DM}^{-1}$ over the velocity update (6.24a), we get $\mathbb{D}\mathbf{U}_{n+1} = \sum_{j=1}^{s} \mathbb{DM}^{-1}(b_i\mathcal{F}(\mathbf{U}_i) + \hat{b}_i\mathcal{G}(\mathbf{U}_i, \mathbf{P}_i))$. Clearly, $\mathbb{D}\mathbf{U}_{n+1} = \mathbb{D}\mathbf{U}_n + \sum_{j=1}^{s} b_i\dot{\mathbf{H}}(t_i)$ if $b_i = \hat{b}_i$. Since the components of $\dot{\mathbf{H}}$ are $p-1$ polynomials in time, their time integration is exact by assumption, i.e., $\sum_{j=1}^{s} b_i\dot{\mathbf{H}}(t_i) = \mathbf{H}(\mathbf{t_{n+1}}) - \mathbf{H}(\mathbf{t_n})$, and $\mathbb{D}\mathbf{U}_{n+1} = \mathbf{H}(\mathbf{t_{n+1}})$. Since $\mathbb{D}\mathbf{U}_0 = \mathbf{H}(\mathbf{0})$, it proves the proposition. □

This result is certainly strong. Even though we are not explicitly enforcing the discrete divergence constraint at every time step, the solution does keep this desired property in many cases. (We note that the intermediate stage corrections do not hold the discrete divergence constraint unless we consider $\mathcal{F} = 0$, see Proposition 6.3.1.)

Remark 6.3.2. *The assumption* $b_i = \hat{b}_i$ *for* $i = 1, \ldots, s$ *is satisfied by many RK time integrators; in particular, schemes (1-1), (1-2), (2-2/1), (2-3), (3-3) and (5-3) defined in C satisfy this condition. The assumption that a* p*-th order RK scheme integrates exactly* $p-1$ *polynomials is one of the standard so-called simplifying conditions of RK methods [92], stated as* $\sum_{j=1}^{s} b_j c_j^{q-1} = \frac{1}{q}$*, for* $q = 1, \ldots, p$*.*

Remark 6.3.3. *In this section, we have worked with the most general case in which* $\mathbf{H}(\mathbf{t})$ *is time-dependent. It only happens when the Dirichlet boundary conditions are enforced strongly and the velocity trace* $\mathbf{u}_g(t)$ *to be enforced on* Γ_D *is time-dependent. For exactly fulfilling the discrete divergence constraint in all cases, we strongly favour the weak imposition of the boundary conditions, i.e., using Nitsche's method, having* $\mathbb{D}\mathbf{U} = 0$ *in all cases. When* $\mathbf{H}(\mathbf{t})$ *is not a* p*-th polynomial, the method still shows the right convergence order, but the discrete divergence constraint is only computed approximately due to the time integration error.*

Remark 6.3.4. *The previous results are obtained assuming that the linear systems are being solved with exact arithmetic. Since the divergence-free condition is enforced in an incremental way, there is a potential loss of accuracy in the constraint equation due to the accumulated error of the linear system for long-term simulations (see Section 6.4). In any case, it is easy to correct it by projecting the velocity field when the divergence residual is larger than some threshold. Given a velocity field* $\mathbf{U}$*, the projected discrete divergence-free velocity field* $\tilde{\mathbf{U}}$ *is computed as follows:*

$$F\tilde{\mathbf{U}} + G\phi = F\mathbf{U}, \qquad \mathbb{D}\tilde{\mathbf{U}} = 0,$$

with $F = \mathbb{M}$ *(*L^2*-projection) or* $F = \mathbb{M}+\mathbb{K}$ *(*H^1*-projection). This kind of techniques are heavily used in MHD simulations in order to clean the induced magnetic field of its non-divergence free component.*

Finally, let us analyze the error introduced by the SRK methods, which is straightforward from the general RK schemes.

Proposition 6.3.3. *Let us consider the SRK method with a* p*-th order scheme. The error for the velocity and the pressure is reduced as* $\mathcal{O}(\delta t^p)$*.*

Proof. Let us denote the time-continuous solution as $(\mathbf{U}^{\text{ex}}(t), \mathbf{P}^{\text{ex}}(t))$. The convergence order for the velocity is $\|\mathbf{U}_n - \mathbf{U}^{ex}(t_n)\|_M \leq c\delta t^p$ (where $\|\cdot\|_{\mathbb{M}}$ denotes the norm endowed by matrix $\mathbb{M}$) since the SRK method for the velocity amounts to a standard RK scheme for the ODE system (6.17) (see [91]). The pressure error is straightforward from the velocity estimate, Eq. (6.24b), and the fact that $\mathbb{D}\mathbb{M}^{-1}\mathbb{G}$ is invertible. In order to obtain the pressure error, let us subtract Eq. (6.11) from Eq. (6.24b) at time value t_n, getting:

$$\begin{aligned}
-\mathbb{D}\mathbb{M}^{-1}\mathbb{G}(\mathbf{P}_n - \mathbf{P}^{\text{ex}}(t_n)) &= \mathbb{D}\mathbb{M}^{-1}((\mathbb{K} + \mathbb{C}(\mathbf{U}_n))\mathbf{U}_n - \mathbb{D}\mathbb{M}^{-1}(\mathbb{K} + \mathbb{C}(\mathbf{U}^{\text{ex}}(t_n))\mathbf{U}^{\text{ex}}(t_n) \\
&= \mathbb{D}\mathbb{M}^{-1}((\mathbb{K} + \mathbb{C}(\mathbf{U}_n))(\mathbf{U}_n - \mathbf{U}^{\text{ex}}(t_n))) \\
&\quad + \mathbb{D}\mathbb{M}^{-1}C(\mathbf{U}_n - \mathbf{U}^{\text{ex}}(t_n))\mathbf{U}^{\text{ex}}(t_n).
\end{aligned}$$

Since $\mathbb{D}\mathbb{M}^{-1}\mathbb{G}$ is a positive-definite matrix (due to the discrete inf-sup condition) we easily get

$$\|(\mathbf{P}_n - \mathbf{P}^{\text{ex}}(t_n))\|_{\mathbb{M}} \leq c\|\mathbf{U}_n - \mathbf{U}^{\text{ex}}(t_n)\|_{\mathbb{M}} \leq c\delta t^p,$$

where the constant c certainly depends on the spatial mesh and the modulus of the computed and exact velocity, but not on δt. It proves the proposition. ☐

6.3.3 Order reduction phenomena

It is known that RK approximation for PDEs suffer from order reduction phenomena for schemes of order greater than two. Many works have been devoted to this issue, e.g., [160, 180], where they prove that only under certain conditions the order in time of the fully discrete scheme equals the conventional order of the RK formula. However, the authors of [160, 180] say that these conditions are not natural and, in general, the order in time for schemes of conventional order greater than two will be strictly smaller. In particular, [180] stated that for DIRK schemes of third and fourth-order, in case of inhomogeneous and time dependent boundary conditions, the actual order of these methods is 2.

Boscarino in [29] derives a third-order IMEX RK method for stiff problems that *does not suffer from the order reduction phenomena*. It is stated in this work that most of IMEX RK schemes of order greater than two suffer from order reduction in the stiff regime. The important point here is that for convection-diffusion problems the semi-discrete system becomes stiffer when the spatial mesh is refined. Then, the order reduction phenomena becomes relevant when we use high-order time integrators in fine spatial meshes. This scheme is included in C, denoted by *(5-3)*, and tested in section 6.4.

6.4 Numerical experiments

In this section we aim to see the performance of the SRK methods proposed previously with different test cases. We start with an analytical manufactured solution that belongs to the FE space, so we do not have any spatial error. The next test case is the laminar flow around a cylinder, a widely used benchmark for laminar flows. Finally, we test a turbulent case, the also widely used Taylor-Green vortex flow.

All the different schemes have been implemented in the FEMPAR (Finite Element Multiphysics and massively PARallel) numerical software. FEMPAR is an open source in-house developed, parallel hybrid OpenMP/MPI, object-oriented (OO) framework which, among other features, provides the basic tools for the efficient parallel distributed-memory implementation of substructuring domain decomposition solvers [19, 20].

6.4.1 Manufactured analytical solution

With this test, we want to check that the methods proposed in this work achieve the desired order of convergence in time for the Butcher tableaus defined in C. In order to analyze the convergence order in time, we want to make sure that the solution is not polluted by the spatial error. Then, to eliminate the error of the spatial component, we define an analytical solution in the FE space. This means that we can capture exactly the solution in space but not in time. The analytical solution in a 2D domain is chosen to be:

$$\mathbf{u}(x,y,t) = \begin{pmatrix} x \\ -y \end{pmatrix} \sin\left(\frac{\pi}{10}t\right) \exp\left(\frac{t}{25}\right), \tag{6.26}$$

$$p(x,y) = x + y. \tag{6.27}$$

This solution belongs to the FE space, even for linear elements. We use inf-sup stable elements of the type Q_2/Q_1 in order to avoid the use of a stabilization which would introduce non-desired extra terms. This test is solved in the unit square $\Omega = (0,1)^2$, discretized with a very coarse mesh with 10 elements per direction. With this coarse discretization we do not expect significant impact of the order reduction phenomena exposed in Subsection 6.3.3 for small viscosity since it occurs in the stiff regime, i.e., small element sizes and high viscosity. We define three different viscosities, $\nu = \{1.0, 0.1, 0.01\}$. We run this test in the time domain $(t_0, T) = (0, 0.1)$, using several time step sizes. In particular, we start with a large time step $\delta t = 0.1$ and we reduce it by a half recursively four times until we reach $\delta t = 1.25 \cdot 10^{-2}$. The errors in the velocity and pressure fields, e_u and e_p, are computed as the ℓ^∞-norm (with respect to the nodal values) of the difference between the computed solution and the analytical one at $t = T$.

As we use either explicit or IMEX time integration schemes, we have to be careful with the CFL number. Here we will have a diffusive CFL number (CFL_ν), which will limit the method when we use a fully explicit scheme, and a convective CFL number (CFL_u), which will limit the method when we use an IMEX scheme with only the diffusive term integrated implicitly, i.e.,

$$\mathrm{CFL}_\nu = \frac{\nu \delta t}{(h/p^2)^2}, \qquad \mathrm{CFL}_u = \frac{u \delta t}{(h/p^2)},$$

being h the characteristic element size, p the degree of interpolation, and u the characteristic velocity. In this test, we have $h = 0.1, p = 2$ and $u \sim 0.0315$. Table 6.1 shows the CFL values for each time step and viscosity.

TABLE 6.1: CFL values.

δt	CFL_ν			CFL_u
	$\nu = 1.0$	$\nu = 0.1$	$\nu = 0.01$	
$1.0 \cdot 10^{-1}$	160	16	1.6	0.126
$5.0 \cdot 10^{-2}$	80	8	0.8	0.063
$2.5 \cdot 10^{-2}$	40	4	0.4	0.032
$1.25 \cdot 10^{-2}$	20	2	0.2	0.016

Note that the characteristic velocity u is very small, so the CFL_u number will also be very small (much lower than 1.0), since it only depends linearly on the mesh size.

Then we do not expect instabilities due to the explicit treatment of the convective term in this test. We cannot say the same for the CFL_ν number, which is larger than 1.0 in most of the cases because of the quadratic dependency on the mesh size.

In this test we consider three different situations: 1) Fully implicit SRK scheme; 2) SRK scheme with diffusive term integrated implicitly and convective term explicitly; 3) Fully explicit SRK scheme.

Fully implicit SRK. Here we consider a SRK scheme with the convective and diffusive terms integrated implicitly. That is to set the operators $\mathcal{F}$ and $\mathcal{G}$ that appear in Eqs. (6.23) and (6.24) equal to

$$\mathcal{F}(\mathbf{U}) := \mathbf{F} - (\mathbb{K} + \mathbb{C}(\mathbf{U}))\mathbf{U}, \qquad \text{and} \qquad \mathcal{G}(\mathbf{U},\mathbf{P}) := -G\mathbf{P}.$$

It is important to highlight here that this scheme is nonlinear, since we have the convective velocity on the left-hand side of the equation.

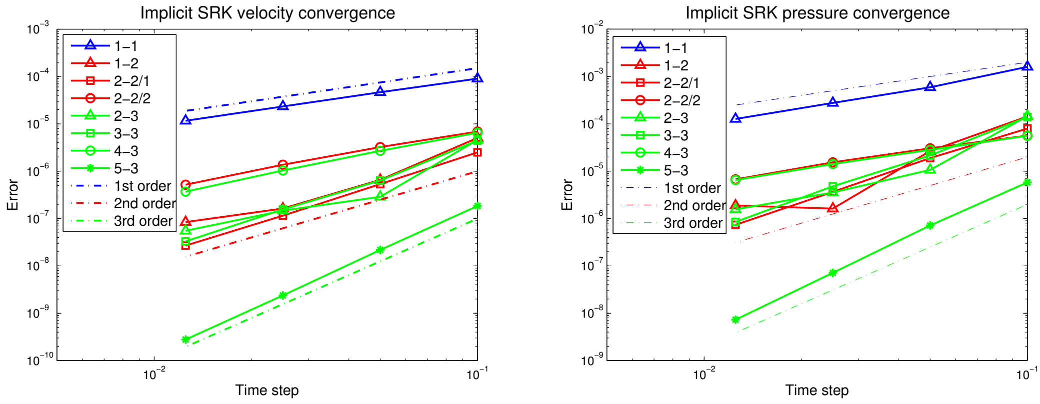
Figure 6.1 shows the convergence rate of the velocity and the pressure fields for the different viscosity choices. Looking at the stiffest case, Figure 6.1(a) and Figure 6.1(b), where $\nu = 1.0$, we see that schemes *(2-2/2)* and *(4-3)* seem to converge in a rate lower than two, which could be justified by the order reduction phenomena. Note that precisely these two methods do not satisfy the incompressibility constrain when time dependent boundary conditions are applied, see Proposition 6.3.2. Almost the same behaviour is observed for the scheme *(2-3)* which also has an order reduction in its convergence. On the other hand, scheme *(3-3) has a reduction on the order of convergence but it is still greater than two. Finally, scheme (5-3) is not affected by the order reduction phenomena.*

When we reduce the viscosity to $\nu = 0.1$, the schemes seem to start to recover their prescribed convergence rate. In Figure 6.1(c) and Figure 6.1(d) the convergence rate in time are plotted for the velocity and pressure fields, respectively. It is seen that for small time steps all schemes converge with the correct rate, but for larger time steps the *(2-2/1)* and *(4-3)* schemes have order reduction.

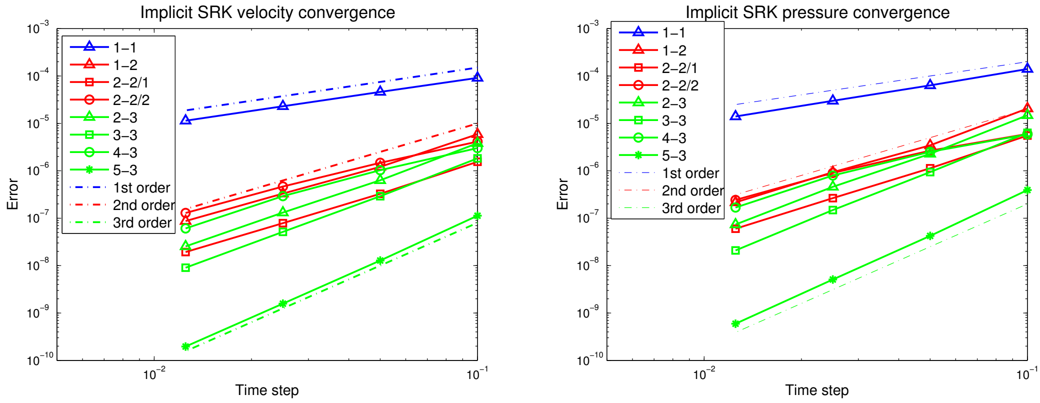
Finally, in Figure 6.1(e) and Figure 6.1(f), where we show the convergence rate for the velocity and pressure fields with $\nu = 0.01$, all schemes considered in this work perform with the prescribed convergence rate.

IMEX-SRK. In this case the SRK scheme is defined only with the diffusive term integrated implicitly, while the convective one is treated explicitly. Then, the operators $\mathcal{F}$ and $\mathcal{G}$ will be

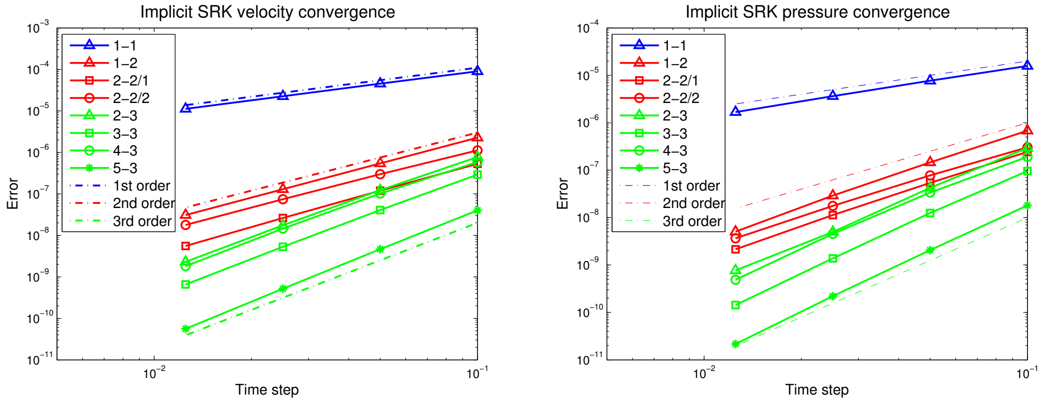
$$\mathcal{F}(\mathbf{U}) := -K\mathbf{U}, \qquad \text{and} \qquad \mathcal{G}(\mathbf{U},\mathbf{P}) := \mathbf{F} - C(\mathbf{U})\mathbf{U} - G\mathbf{P}.$$



(a) Velocity convergence, $\nu = 1.0$ (b) Pressure convergence, $\nu = 1.0$



(c) Velocity convergence, $\nu = 0.1$ (d) Pressure convergence, $\nu = 0.1$



(e) Velocity convergence, $\nu = 0.01$ (f) Pressure convergence, $\nu = 0.01$

FIGURE 6.1: Fully implicit SRK.

Note that for this case, there will be the limitation on the hyperbolic CFL_u number, which is less restrictive than the parabolic one. As it is seen in Table 6.1, it is always less than 1.0 for the chosen time step sizes.

Figure 6.2 depicts the velocity and pressure convergence rate using different viscosities for this second case. For the highest viscosity $\nu = 1.0$ (Figure 6.2(a) and Figure 6.2(b)), we note that almost all methods perform in a similar way as the fully implicit

SRK case, with the difference that scheme *(3-3)* is also showing an order reduction in its convergence rate, being of 2nd order. In this case, we can state that the *(5-3)* scheme is the only third-order scheme that does not show an order reduction in the time convergence, although for the smallest time steps, the velocity convergence rate is a little bit lower than 3.

When we reduce the viscosity to $\nu = 0.1$ or $\nu = 0.01$, all schemes show the same behaviour as the fully implicit SRK case, see Figure 6.2(c), Figure 6.2(d), Figure 6.2(e), and Figure 6.2(f). Let us note that the *(2-2/1)* scheme has much lower error in the IMEX-SRK case than in the fully implicit SRK case.

Fully explicit SRK. Finally we test the fully explicit situation, which consist on sending all terms to the right-hand side of the equation. The operators $\mathcal{F}$ and $\mathcal{G}$ in Eqs. (6.23) and (6.24) will read

$$\mathcal{F}(\mathbf{U}) := 0, \qquad \text{and} \qquad \mathcal{G}(\mathbf{U},\mathbf{P}) := \mathbf{F} - (\mathbb{K} + \mathbb{C}(\mathbf{U}))\mathbf{U} - G\mathbf{P}.$$

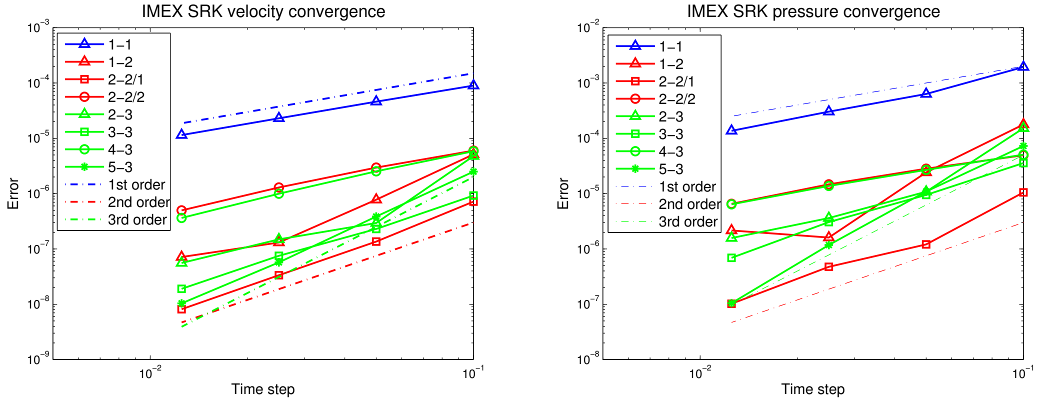
In this case the CFL number that limits the stability of the method is given by the parabolic one (CFL_ν). This is far more restrictive than CFL_u in the IMEX case, and for the setting described above we get unstable results. Only when the viscosity is $\nu = 0.01$, the CFL_ν values are in the order or smaller than the critical value 1.0.

Figure 6.3 depicts the convergence rate for this case, where we see that for the velocity (Figure 6.3(a)) and the pressure (Figure 6.3(b)) fields the order of convergence is the desired one for most of the schemes, except for the *(5-3)* scheme which converges with a higher order. Here does not appear the order reduction phenomena since it is not present for explicit schemes.

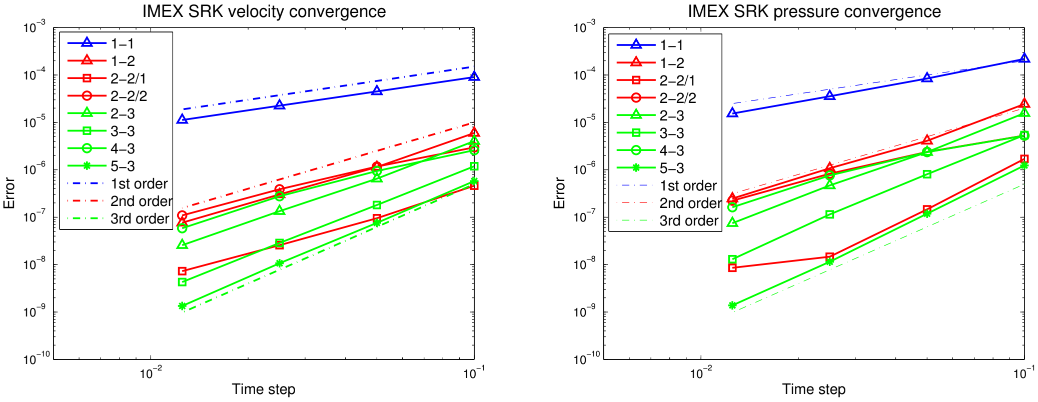
Discrete divergence constraint preservation. As stated in Proposition 6.3.2, the discrete divergence constraint is preserved when the strongly imposed Dirichlet data is a polynomial of order at most $p-1$ in time. In order to show this phenomena we solve the same problem given by (6.26)-(6.27), but considering a second order polynomial for the time dependency. The analytical solution to be solved in this case will be

$$\begin{aligned} \mathbf{u}(x,y,t) &= \begin{pmatrix} x \\ -y \end{pmatrix} t^2, \\ p(x,y) &= x + y. \end{aligned}$$

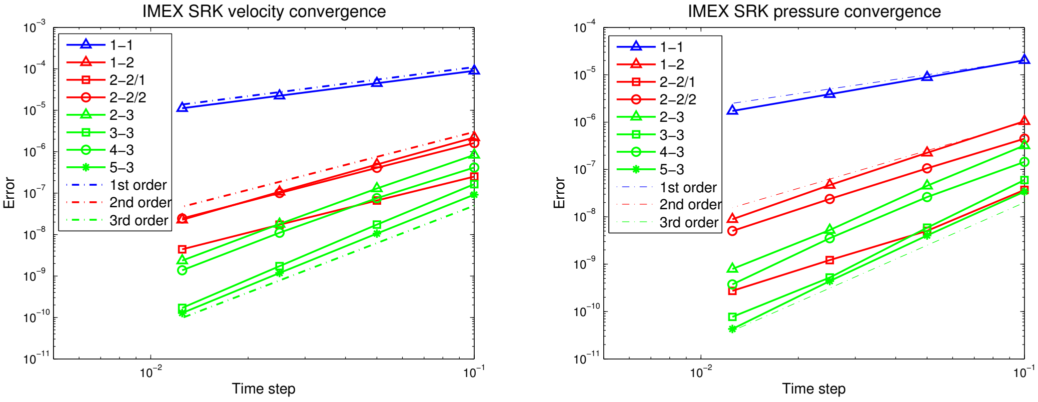
This problem is solved with the fully implicit SRK method from $t = 0$ to $t = 2.0$ using a time step size $\delta t = 1.0 \cdot 10^{-2}$ and a viscosity $\nu = 0.01$. The linear solver tolerance has been set equal to $1.0 \cdot 10^{-8}$, and for the implicit version, the nonlinear tolerance is $1.0 \cdot 10^{-6}$. In Figure 6.4 the evolution of $\|\nabla \cdot \mathbf{u}\|$ is depicted for all schemes considered in C, for both the implicit and explicit versions of the SRK method. We see that for the



(a) Velocity convergence, $\nu = 1.0$ (b) Pressure convergence, $\nu = 1.0$



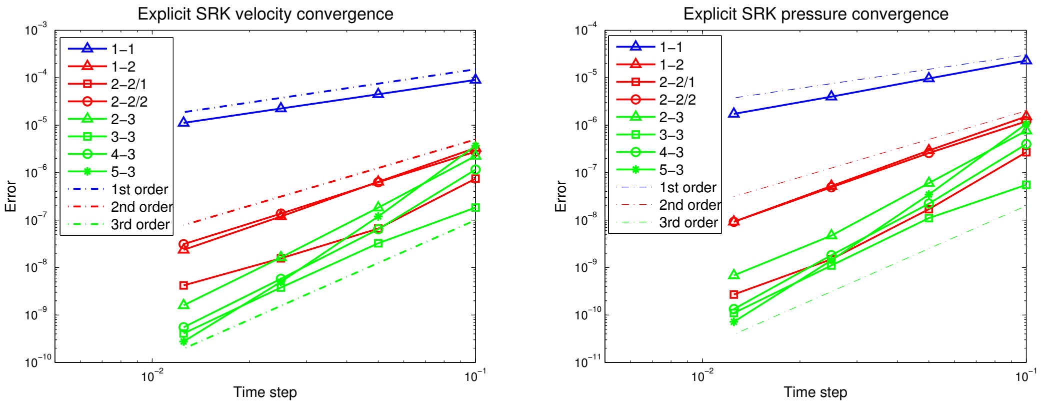
(c) Velocity convergence, $\nu = 0.1$ (d) Pressure convergence, $\nu = 0.1$



(e) Velocity convergence, $\nu = 0.01$ (f) Pressure convergence, $\nu = 0.01$

FIGURE 6.2: SRK convergence with convection integrated explicitly and diffusion integrated implicitly.

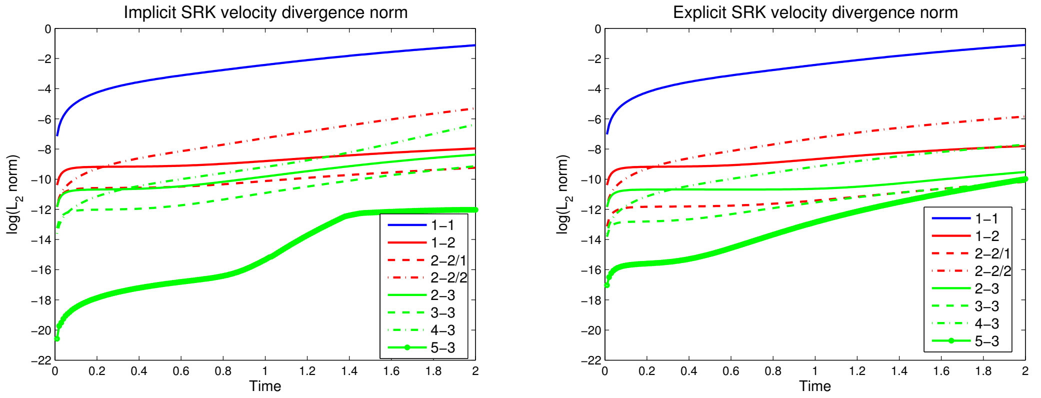
first-order *(1-1)* scheme the discrete divergence constraint is not preserved as it was expected. Moreover, the second-order schemes seem to give really accurate results when evaluating the discrete divergence, even when the time dependence of the solution is of order 2. The *(2-2/2)* and *(4-3)* schemes, which do not satisfy the condition $b_i = \hat{b}_i$ for $i = 1, ..., s$, have the worst performance compared to the other methods of the same



(a) Velocity convergence, $\nu = 0.01$ (b) Pressure convergence, $\nu = 0.01$

FIGURE 6.3: Fully explicit SRK convergence.

order. Although the errors are very small, we can observe the effect of the accumulation of solver error commented in Remark 6.3.4, which leads to an increasing value of $\|\nabla \cdot \mathbf{u}\|$. In any case, the third order schemes that have been proved to preserve the discrete divergence constraint (see Proposition 6.3.2) keep $\|\nabla \cdot \mathbf{u}\|$ below 10^{-9}.



(a) Implicit SRK (b) Explicit SRK

FIGURE 6.4: $\|\nabla \cdot \mathbf{u}\|$ for the implicit and explicit SRK schemes.

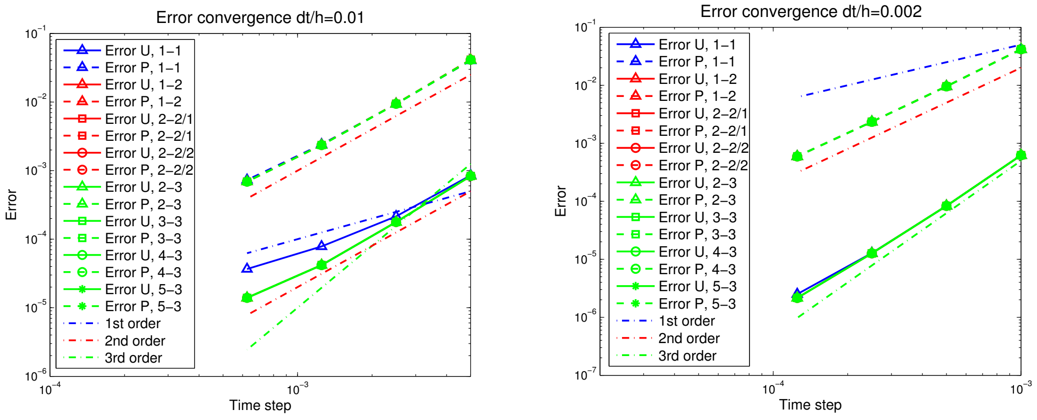
6.4.2 Beltrami flow

In the manufactured analytical solution stated in (6.26)-(6.27) the spatial error is not present since the solution belongs to the FE space. In order to check the behaviour of SRK methods when solving problems with spatial error originated by the discretization in space, we consider an analytical solution that does not belong to the FE space. A 3D Beltrami flow like the one defined in [70] is used in this subsection, but in this case a pressure with no dependence in time is defined. The flow is solved in a cube centered on $(x, y, z) = (0, 0, 0)$ and with a edge size $L = 2$, the viscosity of the problem in this case is set as $\nu = 0.01$.

$$\mathbf{u}(x,y,z,t) = \begin{pmatrix} -a\left[e^{ax}\sin(ay+dz)+e^{az}\cos(ax+dy)\right] \\ -a\left[e^{ay}\sin(az+dx)+e^{ax}\cos(ay+dz)\right] \\ -a\left[e^{az}\sin(ax+dy)+e^{ay}\cos(az+dx)\right] \end{pmatrix} e^{-d^2 t}, \tag{6.28}$$

$$p(x,y,z) = -\frac{a^2}{2}\Big[e^{2ax}+e^{2ay}+e^{2ay}+2\sin(ax+dy)\cos(az+dx)e^{a(y+z)} \tag{6.29}$$
$$+2\sin(ay+dz)\cos(ax+dy)e^{a(z+x)}+2\sin(az+dx)\cos(ay+dz)e^{a(x+y)}\Big].$$

To analyze the effect of the spatial error we solve the problem with the analytical solution (6.28)-(6.29) refining both in time and space keeping the ratio $\delta t/h$ constant. The problem is solved using Q_2/Q_1 elements with the IMEX SRK method, for all schemes stated in C. A first result is obtained setting the $\delta t/h = 0.01$, with the time step sizes $\delta t = \{5.0\cdot 10^{-3}, 2.5\cdot 10^{-3}, 1.25\cdot 10^{-3}, 6.25\cdot 10^{-4}\}$ and solving from $t = 0$ till $T = 5.0\cdot 10^{-3}$. A second test is done with a smaller ratio, $\delta t/h = 0.002$, being the time steps sizes $\delta t = \{1.0\cdot 10^{-3}, 5.0\cdot 10^{-4}, 2.5\cdot 10^{-4}, 1.25\cdot 10^{-4}\}$ from $t = 0$ till $T = 1.0\cdot 10^{-3}$. In Figure 6.5 the error convergence for both velocity and pressure fields is depicted. The results for the case with $\frac{\delta t}{h} = 0.01$ are shown in Figure 6.5(a), where we clearly see that the convergence rate for the pressure field is two, the one prescribed by the spatial discretization error, so here all schemes give the same result since the spatial error prevails over the temporal error. But looking at the velocity field results at the same figure, it is seen that the third-order of convergence given by the theoretical spatial error convergence is reduced. Here we see how the first order scheme *(1-1)* starts loosing the convergence rate given by the spatial error and exhibits the convergence rate prescribed by the temporal discretization. All the other schemes converge with a second order slope, even the third-order schemes. This order reduction phenomena is also observed in [160] when both δt and h are refined simultaneously. When we select a smaller ratio



(a) Error convergence, $\frac{\delta t}{h} = 0.01$ (b) Error convergence, $\frac{\delta t}{h} = 0.002$

FIGURE 6.5: Beltrami flow error convergence (red and blue lines below green line).

of $\delta t/h$, see Figure 6.5(b), the temporal error is masked by the spatial error, giving a third-order convergence for the velocity field and a second-order convergence for the pressure field. A little reduction of the order is observed at the smallest time step sizes for the velocity field in Figure 6.5(b).

6.4.3 2D Laminar flow around a cylinder

Once studied the behaviour of the different methods proposed in Section 6.3 for a manufactured analytical solution, we study a widely used laminar flow benchmark which is the flow around a cylinder for a low Reynolds number $Re = 100$. A detailed overview of benchmark computations of laminar flow around a cylinder are given in [161]. The test performed in the current work is called *2D-2* in that paper and is defined as shown in Figure 6.6. It basically consists in a rectangular channel with a cylinder located near the inflow boundary. A non-slip condition is imposed in the cylinder wall and the channel walls that are perpendicular to the flow direction (x). The inflow condition is

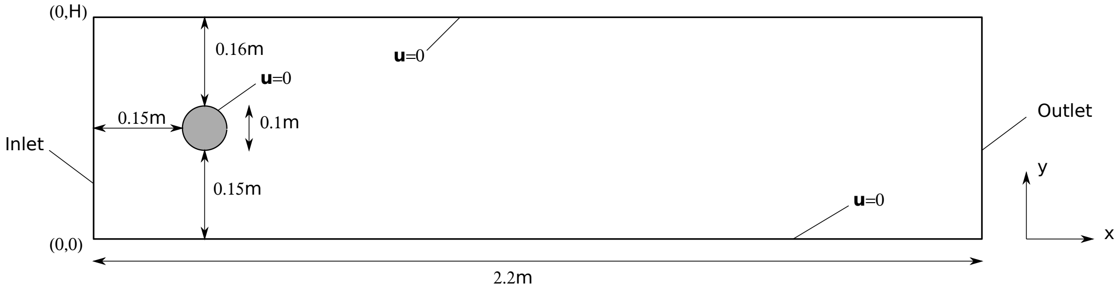


FIGURE 6.6: Flow around a cylinder test geometry.

$$\mathbf{u}(0,y,t) = \begin{pmatrix} u_x \\ u_y \end{pmatrix} = \begin{pmatrix} 4U_m \frac{y(H-y)}{H^2} \\ 0 \end{pmatrix},$$

with the maximum velocity $U_m = 1.5$ m/s and $H = 0.41$ the channel height.

Our aim is not only to see the order of convergence of the time integration schemes proposed, but also to compare the results with a detailed benchmark that has been used to test different algorithmic approaches. Then, we compute the parameters needed for this comparison, namely the drag coefficient c_D, the lift coefficient c_L, and the pressure difference ΔP as functions of time for one period $[t_0, t_0 + 1/f]$, f being the frequency of separation. The values that we will use in the comparison are the maximum drag coefficient $c_{D_{\max}}$, the maximum lift coefficient $c_{L_{\max}}$, the Strouhal number St and the pressure difference $\Delta P(t)$ at $t = t_0 + 0.5/f$. The initial time t_0 should correspond to the flow state with $c_{L_{\max}}$. The drag coefficient c_D and the lift coefficient c_L are given by

$$c_D = \frac{2F_D}{\rho \bar{U}^2 D}, \qquad c_L = \frac{2F_L}{\rho \bar{U}^2 D},$$

being $\rho = 1.0$ the fluid density, $D = 0.1$ the cylinder diameter, and $\bar{U} = 1.0$ the mean velocity. The drag F_D and lift F_L forces are defined as

$$F_D = \int_S \left(\rho\nu \frac{\partial v_t}{\partial n} n_y - p n_x \right) \mathrm{d}S, \qquad F_L = -\int_S \left(\rho\nu \frac{\partial v_t}{\partial n} n_x - p n_y \right) \mathrm{d}S,$$

with S the cylinder surface and n the normal vector on S, with n_x and n_y the x-component and y-component, respectively. v_t is the tangential velocity on S for the tangent vector $t = (n_y, -n_x)$. These surface integrals are computed as the residual of the weak form on boundary nodes, as advocated in [37] for accuracy reasons.

In order to reduce computational cost, we first compute the flow from $t = 0$ to $t = 8.0$ with a monolithic fully implicit Crank-Nicolson scheme with a time step equal to $\delta t = 0.05$. In Figure 6.7 we show the vorticity field at $t = 8.0$. It can be seen that at that time the flow is fully developed. So, at $t = 8.0$ we can start the computations with the different schemes studied in this work. The computations are performed from $t = 8.0$ to $t = 8.4$, ensuring that we have two maximums of the lift coefficient, so we have a complete period of data after the first lift coefficient maximum.

FIGURE 6.7: Vorticity field at $t = 8.0$.

As exposed in the discussion of results in [161], the use of explicit schemes for the time integration of laminar flows is not an efficient approach. The restriction on the time step size to ensure stability of the method is critical since the physical time scale may be much larger. As a result, we will only focus on the fully implicit and IMEX approaches. In this test we consider two different situations: 1) Fully implicit SRK scheme; 2) SRK scheme with diffusive term integrated implicitly and convective term explicitly.

This test is solved using a mesh with 13886 Q_2/Q_1 elements, for all the schemes defined by the Butcher tableaus exposed in C and for several time step sizes. In particular, the problem is solved with $\delta t = \{2.0 \cdot 10^{-2}, 1.0 \cdot 10^{-2}, 5.0 \cdot 10^{-3}, 2.5 \cdot 10^{-3}, 1.25 \cdot 10^{-3}\}$ for each scheme. Furthermore, an extra computation for the *(3-3)* scheme is done with $\delta t = 3.125 \cdot 10^{-4}$ in order to have a more accurate result from which we can compare to do the convergence analysis.

In [161] there is not a prescribed correct value for the benchmark quantities, but there is a range within which most of the reported values are located. Then, we expect that our computation results will fit into these bounds.

Fully implicit SRK. Here, as the flow is laminar, we do not expect the nonlinearity of the convective term to play a decisive role in the computational cost. So, it seems natural to consider an implicit treatment of this term, especially taking into account that the explicit treatment of this term involves time stepping restrictions. Thus, the schemes used in this first case are given by defining the operators $\mathcal{F}$ and $\mathcal{G}$ in Eqs. (6.23) and (6.24) as

$$\mathcal{F}(\mathbf{U}) := \mathbf{F} - (\mathbb{K} + \mathbb{C}(\mathbf{U}))\mathbf{U}, \qquad \text{and} \qquad \mathcal{G}(\mathbf{U}, \mathbf{P}) := -G\mathbf{P}.$$

Let us point out that for this test the *(1-2)* and *(2-3)* schemes are unstable for the largest time step sizes, i.e. $\delta t \geq 1.25 \cdot 10^{-3}$ for the *(1-2)* scheme and $\delta t \geq 2.5 \cdot 10^{-3}$ for the *(2-3)* scheme. Therefore, these two schemes will not be taken into account in the results shown in this first case. The benchmark quantities of the different computations are given in Table 6.2, together with the lower and upper bounds of the results in [161]. It is seen that most of the computed quantities fit into the benchmark bounds; the Strouhal number seems to be a little bit greater than the upper bound in some cases. Note that aside the Strouhal number, the other quantities converge to a value that is inside the benchmark range. As it is expected, the schemes with higher order have more results in the correct range and give very accurate results earlier when we refine the time step.

To check the convergence in time of the different time integration schemes, we perform a convergence analysis computing the ℓ^{∞}-norm of the velocity and the pressure errors, e_u and e_p. In this case, the solution is not compared against an analytical one but computed with a finer time step ($\delta t = 3.125 \cdot 10^{-4}$) with the *(3-3)* scheme. The reason of this choice is related to the efficiency of this scheme, discussed below.

In Figure 6.8 we show the convergence rate for both velocity and pressure fields, Figure 6.8(a) and Figure 6.8(b), respectively. A first conclusion that we can make is that for the largest time steps the order of convergence is not the prescribed one, especially for the first order scheme. It is explained by the fact that the largest time step sizes are greater than the one required for stability purposes when a fully-explicit scheme is used. For the time steps smaller than $2.0 \cdot 10^{-2}$ the convergence rates are the expected ones, with the exception of *(1-1)* and *(4-3)* schemes, with a convergence rate lower than the expected one.

Figure 6.8 shows that the most accurate schemes are *(3-3)* and *(5-3)*. Now, let us analyze the computational cost of these methods based on the CPU time for a given target error. In Figure 6.9 it is clearly seen that the most efficient scheme is *(3-3)*, even for relatively high error ($\sim 4 \cdot 10^{-2}$). For larger target errors, the *(2-2/1)* schemes is competitive. We note that the first order *(1-1)* scheme is not competitive at all, since the error is too large and there is no significant gain in CPU time.

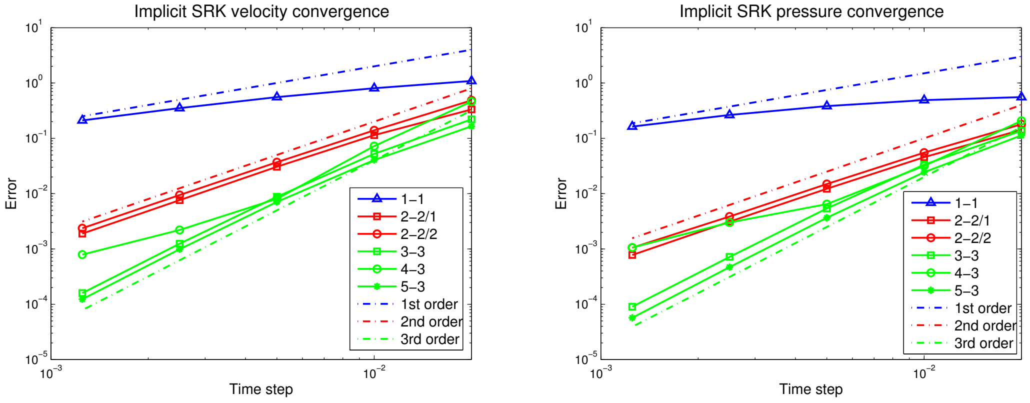
TABLE 6.2: Benchmark 2D-2 results with implicit convection.

Scheme	δt	$c_{D_{max}}$	$c_{L_{max}}$	St	ΔP
$(1-1)$	2.000e-02	3.1984	0.8261	0.2941	2.4291
$(1-1)$	1.000e-02	3.2024	0.8704	0.2941	2.4398
$(1-1)$	5.000e-03	3.2108	0.9158	0.3030	2.4568
$(1-1)$	2.500e-03	3.2182	0.9441	0.3030	2.4700
$(1-1)$	1.250e-03	3.2230	0.9598	0.3042	2.4780
$(2-2/1)$	2.000e-02	3.2423	1.0121	0.3125	2.5082
$(2-2/1)$	1.000e-02	3.2329	1.0039	0.3030	2.4867
$(2-2/1)$	5.000e-03	3.2304	1.0008	0.3077	2.4852
$(2-2/1)$	2.500e-03	3.2298	1.0009	0.3053	2.4884
$(2-2/1)$	1.250e-03	3.2296	1.0007	0.3065	2.4883
$(2-2/2)$	2.000e-02	3.2418	1.0819	0.3125	2.4954
$(2-2/2)$	1.000e-02	3.2323	1.0063	0.3030	2.4835
$(2-2/2)$	5.000e-03	3.2301	0.9973	0.3077	2.4845
$(2-2/2)$	2.500e-03	3.2297	0.9997	0.3053	2.4883
$(2-2/2)$	1.250e-03	3.2296	1.0003	0.3065	2.4883
$(3-3)$	2.000e-02	3.2361	1.0138	0.3125	2.5027
$(3-3)$	1.000e-02	3.2304	1.0013	0.3030	2.4849
$(3-3)$	5.000e-03	3.2298	0.9999	0.3077	2.4848
$(3-3)$	2.500e-03	3.2296	1.0007	0.3053	2.4883
$(3-3)$	1.250e-03	3.2296	1.0006	0.3065	2.4883
$(4-3)$	2.000e-02	3.2321	1.1003	0.3125	2.4788
$(4-3)$	1.000e-02	3.2285	0.9948	0.3030	2.4802
$(4-3)$	5.000e-03	3.2285	0.9954	0.3077	2.4839
$(4-3)$	2.500e-03	3.2294	1.0000	0.3053	2.4881
$(4-3)$	1.250e-03	3.2296	1.0005	0.3065	2.4882
$(5-3)$	2.000e-02	3.2249	0.9770	0.3125	2.4951
$(5-3)$	1.000e-02	3.2290	0.9977	0.3030	2.4843
$(5-3)$	5.000e-03	3.2295	0.9995	0.3077	2.4847
$(5-3)$	2.500e-03	3.2296	1.0006	0.3053	2.4883
$(5-3)$	1.250e-03	3.2296	1.0006	0.3065	2.4883
lower bound		3.2200	0.9900	0.2950	2.4600
upper bound		3.2400	1.0100	0.3050	2.5000

IMEX-SRK. In this case, we consider an explicit time integration approach of the convective term. That is to define the operators $\mathcal{F}$ and $\mathcal{G}$ in Eqs. (6.23) and (6.24) as

$$\mathcal{F}(\mathbf{U}) := \mathbf{F} - K\mathbf{U}, \qquad \text{and} \qquad \mathcal{G}(\mathbf{U}, \mathbf{P}) := C(\mathbf{U}) - G\mathbf{P}.$$

As it has been exposed above, the explicit treatment of the convective term implies time stepping restrictions that are given by the condition on the hyperbolic CFL_u number. As the mesh is not homogeneous and the velocity is not the same over all the domain, we can obtain a bound for the maximum CFL_u number taking the maximum



(a) Velocity convergence

(b) Pressure convergence

FIGURE 6.8: Fully implicit SRK convergence.

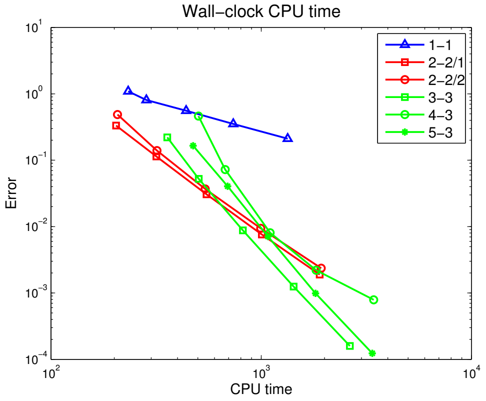


FIGURE 6.9: Fully implicit SRK CPU time efficiency.

value over all the elements. With these parameters, the maximum CFL_u depending on the time step size will be of the order shown in Table 6.3.

Looking at Table 6.3 we see that in this case the time step needs to be much smaller than in the previous case if we want to guarantee a stable solution. For the implicit treatment of the convective term, for time step sizes below or equal to $\delta t \leq 1.0 \cdot 10^{-2}$, the method is stable and gives good convergence rates. Here, for this time step we see that we have a $\mathrm{CFL}_u \sim 3.4$, which is grater than the critical ~ 1.0.

In effect, the largest time step size for which all the schemes give stable results for this case is $\delta t = 3.125 \cdot 10^{-4}$. Some schemes also are stable for $\delta t = 6.25 \cdot 10^{-4}$ or even for $\delta t = 1.25 \cdot 10^{-3}$, but no one is stable for $\delta t = 2.5 \cdot 10^{-3}$, which is of the order of the critical time step size. Thus, it is clear that for this type of problems, an explicit time integration of the convective term implies the use of much smaller time steps than for an implicit time integration of this term.

In Table 6.4 we show the benchmark quantities of the cases that attained convergence, till $\delta t = 1.5625 \cdot 10^{-4}$. It is clearly seen that for such small time step sizes all the results are very similar, showing that the IMEX-SRK scheme also gives good results

TABLE 6.3: CFL_u values.

δt	CFL_u
$2.0 \cdot 10^{-2}$	6.81
$1.0 \cdot 10^{-2}$	3.48
$5.0 \cdot 10^{-3}$	1.79
$2.5 \cdot 10^{-3}$	0.91
$1.25 \cdot 10^{-3}$	0.46
$6.25 \cdot 10^{-4}$	0.21
$3.125 \cdot 10^{-4}$	0.11
$1.5625 \cdot 10^{-4}$	0.05
$7.8125 \cdot 10^{-5}$	0.03

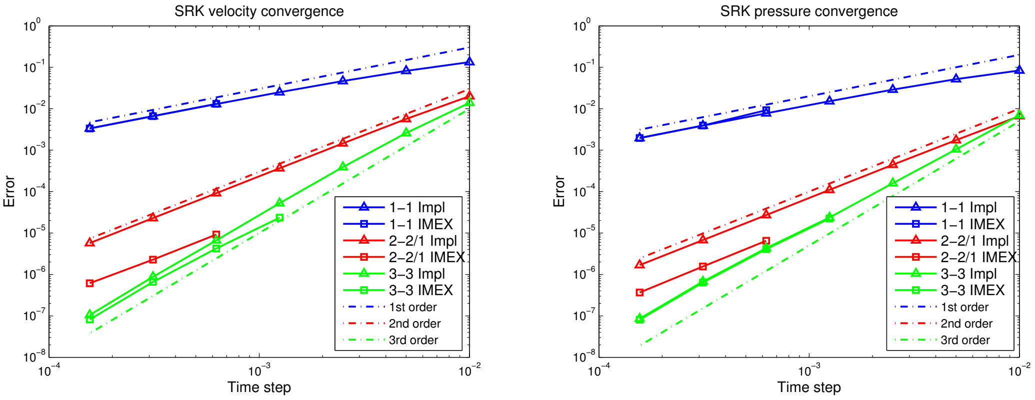
considering the explicit integration of the convective term, whenever the time step is sufficiently small to give a stable result. The results that are out of the range are in italics.

TABLE 6.4: Benchmark 2D-2 results with explicit convection.

Scheme	δt	$c_{D_{max}}$	$c_{L_{max}}$	St	ΔP
$(1-1)$	3.125e-04	3.2272	*0.9882*	*0.3059*	2.4862
$(1-1)$	1.563e-04	3.2280	0.9944	*0.3062*	2.4879
$(1-2)$	3.125e-04	3.2296	1.0006	*0.3068*	2.4891
$(1-2)$	1.563e-04	3.2296	1.0006	*0.3065*	2.4893
$(2-2/1)$	6.250e-04	3.2296	1.0006	*0.3065*	2.4891
$(2-2/1)$	3.125e-04	3.2296	1.0006	*0.3065*	2.4895
$(2-2/1)$	1.563e-04	3.2296	1.0006	*0.3065*	2.4893
$(2-2/2)$	6.250e-04	3.2296	1.0006	*0.3065*	2.4892
$(2-2/2)$	3.125e-04	3.2296	1.0006	*0.3065*	2.4896
$(2-2/2)$	1.563e-04	3.2296	1.0006	*0.3065*	2.4893
$(2-3)$	3.125e-04	3.2296	1.0006	*0.3065*	2.4895
$(2-3)$	1.563e-04	3.2296	1.0006	*0.3065*	2.4893
$(3-3)$	6.250e-04	3.2296	1.0006	*0.3065*	2.4891
$(3-3)$	3.125e-04	3.2296	1.0006	*0.3065*	2.4895
$(3-3)$	1.563e-04	3.2296	1.0006	*0.3065*	2.4893
$(4-3)$	6.250e-04	3.2296	1.0006	*0.3065*	2.4891
$(4-3)$	3.125e-04	3.2296	1.0006	*0.3065*	2.4895
$(4-3)$	1.563e-04	3.2296	1.0006	*0.3065*	2.4893
$(5-3)$	6.250e-04	3.2296	1.0006	*0.3065*	2.4891
$(5-3)$	3.125e-04	3.2296	1.0006	*0.3065*	2.4895
$(5-3)$	1.563e-04	3.2296	1.0006	*0.3065*	2.4893
lower bound		3.2200	0.9900	0.2950	2.4600
upper bound		3.2400	1.0100	0.3050	2.5000

In order to compare the efficiency between the implicit and explicit approaches for the time integration of the convective term for this particular test, we perform the convergence analysis for both methods at $t = 8.01$ s. We restrict this test only to the most efficient schemes for each order of approximation, which are *(1-1)*, *(2-2/1)*, and *(3-3)* schemes. The results of the convergence in time for this comparison are shown in Figure 6.10. Unstable results for the explicit convective term integration are not plotted in this figure.

The fully implicit versions of the SRK schemes considered here show a good agreement with the prescribed convergence rate. Furthermore, when the time step is sufficiently small, the explicit version seems to give more accurate results. This is especially remarkable for the *(2-2/1)* scheme, where the differences are bigger. Same conclusions can be pointed out for both velocity and pressure fields, since there are not significant differences between their convergence rates, see Figure 6.10(a) and Figure 6.10(b). Figure 6.10 shows that the *(3-3)* scheme is the most accurate one. Apart from the *(2-2/1)* scheme, implicit and explicit versions give similar results when the time step is sufficiently small.



(a) Velocity convergence

(b) Pressure convergence

FIGURE 6.10: Fully implicit and IMEX-SRK convergence rate comparison.

Next, we look at the overall computational time needed for each approach. This is shown in Figure 6.11, where the error e_u is plotted against the CPU time for each scheme and for each approach. We see that, as expected, the approach with explicit time integration of the convective term is more efficient than the implicit one when the time step is sufficiently small. For large time steps, we cannot use the explicit versions since the CFL$_u$ condition is limiting the stability of the method, and therefore its accuracy. Note that there is not a big difference on the efficiency for *(1-1)* and *(3-3)* schemes, since the gain of treating the convective term explicitly is not too much relevant in laminar problem types. As exposed at the beginning of this subsection, explicit approaches involve time step size restrictions, which may be much smaller than the physical time scale. Furthermore, as it has also been exposed before, the nonlinearity

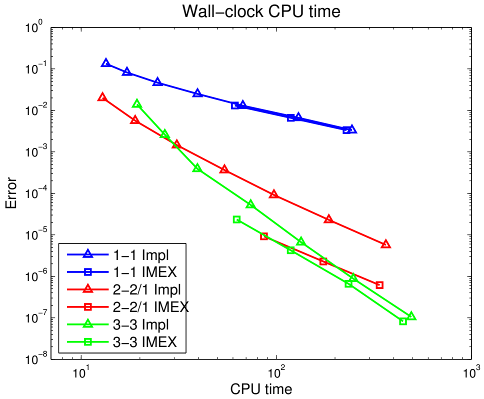


FIGURE 6.11: Fully implicit and IMEX-SRK CPU time efficiency comparison.

of the convective term when it is integrated implicitly does not increase the cost too much. Despite of that, for errors smaller than $\sim 10^{-4}$, the most efficient scheme is the *(3-3)* using explicit time integration of the convective term, at least for this test.

Adaptive time stepping technique

When solving generic transient flow problems, it is very useful to use an adaptive time stepping technique that will automatically provide a (dynamic) value of the time step size for a target accuracy. Time stepping techniques allow us to adapt the time step size to the flow conditions, that is to change the time step when the physical scales of the flow change, e.g., transition to turbulence. Adaptive time-stepping techniques have been implemented to satisfy accuracy requirements for the incompressible Navier-Stokes equations and several successful tests can be found in the literature, as it can be shown in [72, 115, 118, 178].

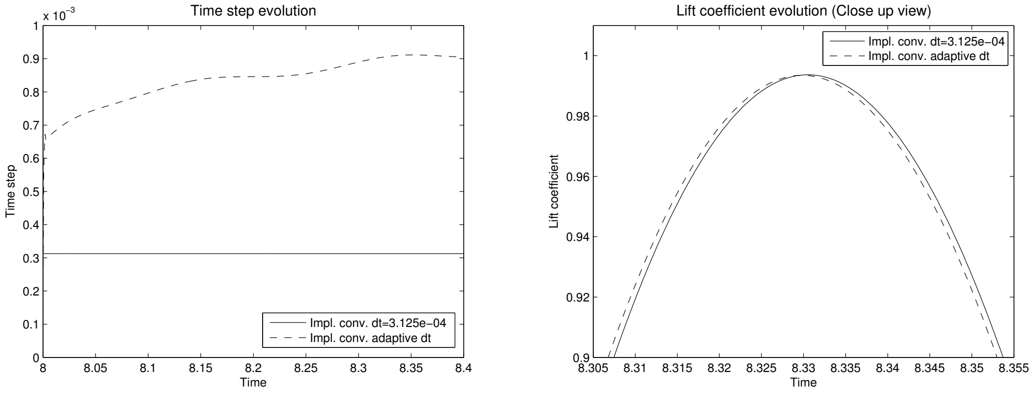
These adaptive time stepping techniques can straightforwardly be applied to SRK methods. In fact, for all multi-step or multi-stage methods like SRK schemes, the implementation of an adaptive time step technique is widely used, since we only need a different evaluation of the final unknown at each step that can be done using a different Butcher tableau, see for instance [88, 89]. Here we use the so called *PI11 controller* by [173] and suggested in [89], which computes the time step size as follows

$$\delta t_{n+1} = \left(\frac{\epsilon}{r_{n+1}}\right)^{1/k} \left(\frac{r_n}{r_{n+1}}\right)^{1/k} \frac{\delta t_n}{\delta t_{n+1}} \delta t_n,$$

with $\epsilon = 0.8 \cdot TOL$, where TOL is a given tolerance that we take as $1 \cdot 10^{-6}$ and 0.8 is a safety factor. The local error is $r_{n+1} = \|\mathbf{U} - \hat{\mathbf{U}}\|$ if the error per step (EPS) is controlled or $r_{n+1} = \|(\mathbf{U} - \hat{\mathbf{U}})/\delta t_n\|$ if the error per unit step (EPUS) is controlled. In the former case $k = p + 1$ (EPS) and for the second one (EPUS) $k = p$, p being the order of the time integration scheme which has been used to compute the estimated velocity $\hat{\mathbf{U}}$.

We solve the problem from $t = 8.0$ to $t = 8.4$ using the SRK *(3-3)* scheme, considering the implicit time integration of the convective term. We compare the solutions against the one obtained with the same scheme, but using a fixed time step size $\delta t = 3.125 \cdot 10^{-4}$, the smallest time step considered in previous analysis. Let us note that an adaptive time stepping technique for the explicit time integration of the convective term does not make sense in this case. We see in Table 6.3 that the CFL_u number of the SRK with explicit versions of the convective term with a time step of $\delta t = 3.125 \cdot 10^{-4}$ is already above the critical value of 1.0. This means that if we increase the time step size, we will have stability problems due to the hyperbolic CFL condition.

To compute $\hat{\mathbf{U}}$ we have used the 2nd order Butcher tableau referred as *Embedded Formula of Order 2* for the *Third Order Strongly S-Stable Formula* defined in [45], which is 3rd order and corresponds to the *(3-3)* scheme defined in C. The initial time step size is set to be $\delta t_0 = 1.0 \cdot 10^{-5}$, small enough to get an accurate first solution. In Figure 6.12(a) we show the time step evolution for the two different cases considered in this subsection. We see that the time step size for the scheme with adaptive time stepping is increasing with a variable rate and seems to converge to an optimal one. Figure 6.12(b) depicts a close up view of the lift coefficient evolution. We see that for the implicit adaptive time step case, the results are really close to those obtained with the same scheme with a fixed time step of $\delta t = 3.125 \cdot 10^{-4}$. Note that the adaptive time step size are twice or even three times larger than the fixed one. The total elapsed CPU time for the implicit scheme with adaptive time step is 3615 s, while the total time for the implicit scheme with fixed time step is 8453 s, which supposes a reduction of a 42.8% of time consumption with a very little difference in the result.



(a) Time step evolution

(b) Lift coefficient (close up view)

FIGURE 6.12: Adaptive time stepping.

6.4.4 Taylor-Green vortex flow

The next step of this work is to check the performance of the SRK methods for turbulent incompressible flows. The use of IMEX-SRK methods is usually favoured for turbulent flows, since the time step size required to satisfy the CFL_u condition is of the same order as the physical one needed for accuracy purposes. The Taylor-Green vortex (TGV) problem is a typical and widely used problem in turbulence numerical simulations, in which we can see the basic turbulence decay mechanisms in a relatively simple flow. Here, the computational domain is the cube $(0, 2\pi)^3$ with periodical boundary conditions. The initial analytical condition for this problem is given by (see, e.g., [31])

$$\mathbf{u}(x,y,z,0) = \begin{pmatrix} u_x \\ u_y \\ u_z \end{pmatrix} = \begin{pmatrix} u_0 \cos(x)\sin(y)\sin(z) \\ -u_0 \sin(x)\cos(y)\sin(z) \\ 0 \end{pmatrix} \tag{6.30}$$

$$p(x,y,z,0) = p_0 + \frac{1}{16}\left(\cos(2x)+\cos(2y)\right)\left(\cos(2z)+2\right),$$

with

$$u_0 = \frac{2}{\sqrt{3}}\sin\left(\gamma + \frac{2\pi}{3}\right).$$

We choose $\gamma = 0$, which gives the mean initial velocity $u_0 = 1$. We solve the TGV problem using a Reynolds number $\mathrm{Re} = 1600$, but in the literature the same test using different Reynolds numbers (e.g., $\mathrm{Re} = 800$ and $\mathrm{Re} = 3000$) can be found (see, e.g., [80, 96]).

The problem is solved in a 64^3 Q_2/Q_1 elements mesh, and no additional sub-grid modelling is being used. Our concern is about the time integration, therefore the spatial accuracy does not take a crucial role in this work. Anyway, we analyze some physical quantities, like the global kinetic energy or the kinetic energy dissipation rate, that are typically used for this test to calibrate different methods.

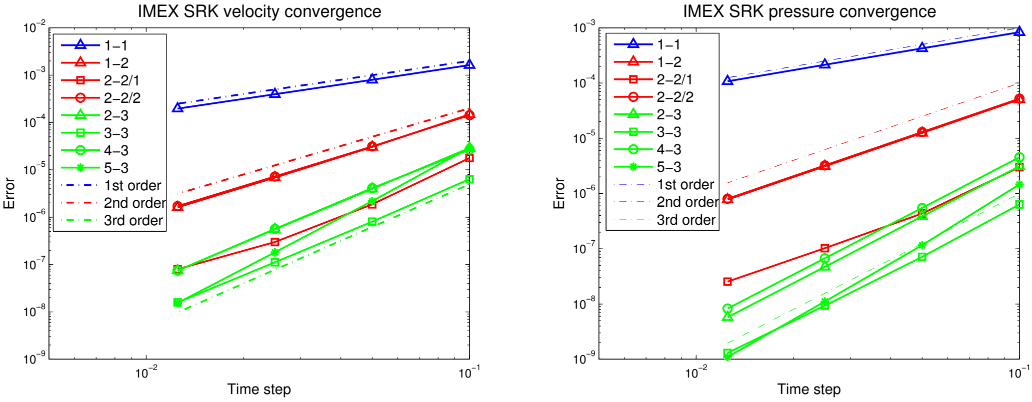
We only use the IMEX-SRK method with the convection treated explicitly, avoiding the need of nonlinear iterations. This situation is given by defining the operators $\mathcal{F}$ and $\mathcal{G}$ in Eqs. (6.23) and (6.24) as

$$\mathcal{F}(\mathbf{U}) := -K\mathbf{U}, \qquad \text{and} \qquad \mathcal{G}(\mathbf{U},\mathbf{P}) := \mathbf{F} - C(\mathbf{U})\mathbf{U} - G\mathbf{P}.$$

First of all, we perform a time step convergence analysis at the beginning of the simulation, where the flow is still laminar. This convergence analysis consists in solving the problem from $t = 0.0$ to $t = 0.1$ for all the schemes proposed in C for several time step sizes. In particular, we solve the problem with four different time step sizes ($\delta t = \{0.1, 0.05, 0.025, 0.0125\}$) and we compare the solution against the one obtained with the *(5-3)* scheme with a time step equal to $\delta t = 6.25 \cdot 10^{-3}$. The L^2-norm of the kinetic energy error compared against the reference kinetic energy solution given by a DNS computation can be found in [27]. Our approach is to compute the L^∞-norm of the solution, comparing against a solution computed with a finer time step, but with the same spatial discretization. With this approach we are eliminating the spatial error and we are using a more restrictive error norm.

We show in Figure 6.13 the order of convergence in time for both velocity (Figure

6.13(a)) and pressure (Figure 6.13(b)) fields. We can see that the order of convergence in this regime follows the predicted rate for almost all schemes; the *(5-3)* schemes seems to converge with a higher rate than the expected one, being this performance especially remarkable for the velocity field convergence.



(a) Velocity

(b) Pressure

FIGURE 6.13: IMEX-SRK convergence for the laminar regime.

Another point to be highlighted in Figure 6.13 is that the *(2-2/1)* scheme shows a much lower error compared to the other second-order schemes. This performance is also observed on the previous tests when an IMEX-SRK (with explicit treatment of the convection) is used.

As it has been done for the 2D laminar flow around a cylinder test (see Subsection 6.4.3) we analyze the efficiency of the methods comparing error against CPU time. This comparison is in Figure 6.14, where the error is plotted in terms of the averaged elapsed CPU time per processor needed to complete the simulation from $t = 0$ to $t = 0.1$.

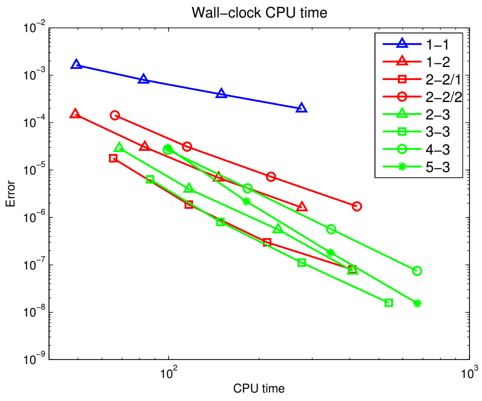
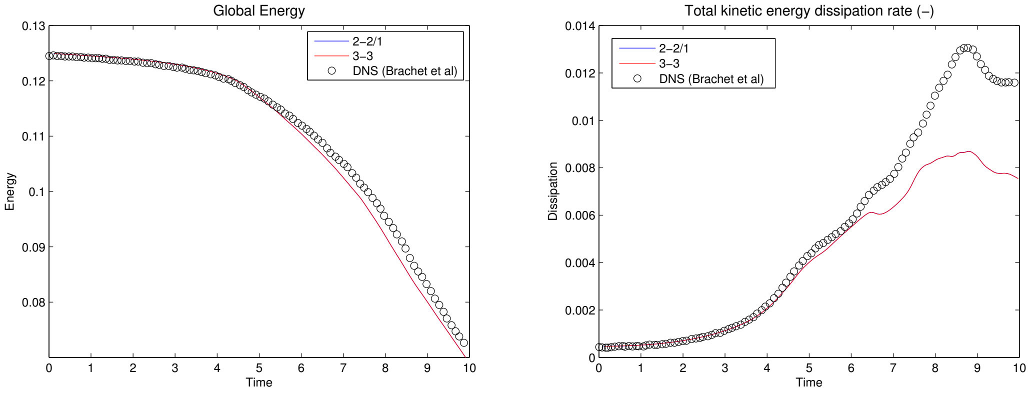


FIGURE 6.14: IMEX-SRK efficiency for the laminar regime.

It is seen in Figure 6.14 that the most efficient schemes are the *(2-2/1)* and *(3-3)* ones, which for errors greater than $\sim 10^{-6}$ give the same error-CPU time ratio. As it is natural, below this threshold, the second-order scheme *(2-2/1)* starts loosing efficiency and the *(3-3)* scheme becomes the most efficient one for errors smaller than $\sim 10^{-6}$.

Once we have determined that the most efficient schemes for this test are the *(2-2/1)* and *(3-3)* ones, we now want to see their performance for a larger time interval. The aim is to solve the problem until the turbulence becomes fully developed, which for this Reynolds number case takes place around $t \sim 9$. Following [31], the time interval will go from $t = 0.0$ to $T = 10.0$, so we will be able to compare the results with their DNS. For this computation, we select the time step size that has an error at $t = 0.1$ in Figure 6.13(a) of the order $\sim 1 \cdot 10^{-6}$, i.e., $\delta t = 5.0 \cdot 10^{-2}$ for the *(3-3)* scheme and $\delta t = 3.6 \cdot 10^{-2}$ for the *(2-2/1)* scheme.

Figure 6.15 depicts the total energy evolution (Figure 6.15(a)) and the kinetic energy dissipation rate of the resolved scales (Figure 6.15(b)) compared against the DNS provided by [31]. The result is exactly the same for both schemes, despite the fact that the third-order scheme uses a larger time step. Looking at Figure 6.15(a), we see that the energy evolution of the solution is not far from the DNS, but there is a gap after $t = 6$, when the turbulence is developed. This is caused by the lack of any turbulent model which would capture the small scales proper dissipation. For the same reason we see big differences between our solutions and the DNS results in Figure 6.15(b) after $t = 6$.



(a) Global energy. (b) Kinetic energy dissipation rate.

FIGURE 6.15: Global energy and Kinetic energy dissipation rate evolution (blue line below red line).

Looking at Figure 6.15(b) we clearly see that the error given by the spatial discretization when the flow becomes turbulent is too large to appreciate significant differences between time integration schemes. We want to highlight here that an adaptive time step technique could be used in order to efficiently solve the transient problem. However, the use of an adaptive time stepping technique without an accurate solution in space does not make sense. Future work in this direction involves the introduction of LES models within the IMEX-SRK approach proposed in this work.

6.5 Conclusions

The segregated Runge-Kutta methods proposed in this work enjoy two nice features, namely the velocity and pressure segregation at the time integration level (without the need to perform additional fractional step techniques that spoil high orders of accuracy) and the provable same order of accuracy for both velocities and pressures. These methods have been motivated as an implicit-explicit Runge-Kutta time integration of the projected Navier-Stokes system onto the discrete divergence-free space. The terms in this system that involve the inverse of a discrete Laplacian $\mathbb{D}\mathbb{M}^{-1}G$ are treated explicitly in all cases, in order to make the resulting method numerically feasible. Viscous and convection terms can be treated using implicit, combined implicit/explicit, or fully explicit schemes, leading to implicit, IMEX, or explicit SRK schemes, respectively. The pressure can be recovered by using a discrete pressure Poisson equation, and the SRK scheme can be finally recasted in a velocity-pressure formulation.

Explicit SRK methods have been proved to be equivalent to existing half-explicit RK methods. Further, these methods exactly satisfy the divergence-constraint equation in most situations of interest; in all cases when weakly enforcing Dirichlet boundary conditions, and for fixed or at most a point-wise p-th order polynomial variation in time (for a p-th order method) of the Dirichlet data for a strong enforcement. Further, it is easy to check that the error for the pressure is of the same order as the one for the velocity.

We have performed a wide set of numerical experiments to evaluate the segregated Runge-Kutta algorithms; first to third order schemes have been implemented and analyzed. They include convergence tests for problems with manufactured solutions for time-dependent Dirichlet boundary data. This way, we can also evaluate the well-known order reduction effect of RK methods. We have also performed numerical tests for the laminar flow around a cylinder (evaluating drag and lift coefficients) and the turbulent Taylor-Green vortex flow. The different methods have been compared in terms of CPU cost for a target error. Fully implicit, implicit viscous/explicit convective, and fully explicit methods have been evaluated, considering their respective CFL conditions. Fully implicit SRK schemes have shown a remarkably strong stability and high accuracy till about 100 times the explicit CFL condition. Further, segregated Runge-Kutta schemes with adaptive time stepping have been proposed and analyzed numerically.

The use of SRK schemes is very appealing for large scale computations of incompressible flows, since the monolithic indefinite system is replaced by segregated positive-definite velocity and pressure blocks. The pressure block involves a Poisson solver, whereas the velocity block is a vector-Laplacian or elasticity matrix when the convective term is treated explicitly. Massively parallel solvers for these problems can be found in the literature (see, e.g., [20]) and are at the user's disposal, e.g., within the FEMPAR scientific computing software.

Chapter 7

Segregated Variational Multiscale Finite Element methods

7.1 Introduction

The numerical simulation of turbulent flows is widely used for scientific purposes and highly demanded in the industry to solve a large amount of engineering problems. The algorithms employed in the computational fluid dynamics field are constantly evolving, adapting to the new trends and tailoring to the continually changing computational requirements. The increasing computational power acquired with the new improvements on super-computer also involves additional advances in the software able to be executed in such machines.

As exposed in Section 4.1, simulating incompressible turbulent flows involve the resolution of multiple scales, both in space and time, becoming a really challenging numerical problem. DNS of turbulent flows are used to capture the physical phenomena at all scales, even the smallest ones. This approach has the inconvenience that consumes a large amount of computational resources. A technique that saves a lot of computational cost is the LES, which basically consist on separating the flow in a coarse scale and a fine one, simulating the coarser and modelling the finer [156].

In order to model the fine scales in a LES method, we can consider a physically based approach, which is defined taking into account the physical phenomena that takes place on the smallest scales, or a purely numerical approach, that does not introduce any modification to the governing equations at the continuous level. This last numerical approach is commonly denoted as ILES, which stands for Implicit LES, see for instance [28].

The VMS method (extensively described in Section 2.4) introduced by Hughes in [100, 101] is a framework to develop stable and accurate numerical approximations of partial differential equations, preventing numerical instabilities that arise when the standard Galerkin FE method is used. The use of the VMS method as an ILES method was firstly suggested in [104, 105, 51] and, since then, several variants have been developed and used as ILES. We can distinguish between those that introduce a three scale decomposition into resolved large and small scales and unresolved scales [125, 113, 114, 41], with a Smagorinsky type model for the influence of unresolved scales onto the small resolved ones, and those that introduce a two scale decomposition into resolved and unresolved ones. The later is the case of [23, 56], where a residual based or projection based model of the unresolved scales is used to account for their influence into the resolved ones. We recall that [56] has motivated Chapter 4.

In this chapter we will consider a two scale VMS approach based on an orthogonal definition of the subscales space, firstly proposed in [50], and named OSS method. Moreover, an alternative definition of the OSS method was proposed in [52], making use of a term-by-term stabilization that does not involve the full residual. Furthermore, we will follow the approach considered in [58], where a symmetric projection stabilization of the convective term using a OSS decomposition is used for inf-sup stable elements. It worth pointing out that this approach is the one described in Chapter 5, so we refer to that chapter for a deeper understanding of such methods.

It is a common approach in the CFD field to consider a strong imposition of the Dirichlet boundary conditions. That is to impose the solution at the nodes of the Dirichlet boundary. This approach may lead to inaccurate solutions in some situations, specially when LES methods to simulate wall-bounded turbulent flows are considered. LES methods are used to simulate turbulent flows in quite coarse meshes, which often are insufficiently fine to capture the boundary layers that appear in wall-bounded flows. In this case, the effect of not capturing properly the solution at the boundary layer can affect to the mean flow, resulting in imprecise simulations. In order to overcome this issue, weak imposition of the Dirichlet boundary conditions can be contemplated. This technique was considered in [24] and later improved with a wall-law based approach for turbulent flows in [25]. In [112] a weak treatment of the boundary conditions is also contemplated for the simulation of flows considering slip with friction and penetration with resistance boundary conditions. Other examples of using weak Dirichlet boundary conditions can be found in [60].

Furthermore, in [24, 25] the wall-normal component of the velocity on the Dirichlet boundary is imposed strongly. This is an approach that we want to avoid due to its complex implementation on curved boundaries. In complicated geometries, the normal vector is not well defined on the boundary nodes because each element surrounding a given node has a different wall-normal vector. Thus, there is not a clear way to distinguish the velocity normal component on the boundary nodes. In this chapter we also consider the weak imposition of the wall-normal component.

As noticed before, the turbulent phenomena is characterized by having not only many spatial scales, but also a multiscale description in time. Then, the time discretization becomes an important issue when simulating this kind of flows. Many authors favor implicit time integration schemes to avoid the time step restriction given by the Courant-Friedrichs-Lewy (CFL) number. However, at high Reynolds number, the hyperbolic CFL number (given by the convective term) has to be kept of the order of the unity, see [47]. Then, it is a common practice to consider a semi-implicit time integration scheme where only the convective term is treated explicit, which prevents on using too small time step sizes given by the restriction of the parabolic CFL number. See for instance [128], where a fractional-step method with Runge-Kutta schemes are used. Another example of this type of time integration schemes can be found in [74], where a semi-implicit BDF time integration scheme is considered together with a VMS-LES spatial discretization approach.

The SRK time integration schemes for the incompressible Navier-Stokes equations where firstly proposed in [57], that work has motivated Chapter 6. These schemes are based on two main goals. First, the segregation of the velocity and pressure computations at the time integration level, without the need to perform additional fractional step techniques that spoil high orders of accuracy. Second, the preservation of the same order of accuracy for both velocities and pressures. In this work we will consider the IMEX version of the SRK scheme that consists on treating implicitly the diffusive term and explicitly the convective term.

When trying to solve increasingly larger problems on super-computers, one has to ensure efficiency on the computational cost. That is to build scalable solvers that guarantee that the computational cost will not increase when more resources are used. In this work we consider the use of BDDC preconditioners, firstly introduced in [66], following the implementations described in [20, 134].

The velocity-pressure segregation introduced by the SRK schemes lead to elasticiy-type and Darcy-type problems that can be preconditioned using a block-preconditioning technique, see [18]. A recursive block-preconditioning technique was used in [58] to solve the monolithic Navier-Stokes problem in serial, giving scalable results in terms of the number of solver iterations. The performance of the block-preconditioning technique together with the BDDC methods proposed in [134], are going to be tested in this chapter.

This chapter is organized as follows. In Section 7.2 the Navier-Stokes problem is stated and the weak Dirichlet boundary conditions imposition is described. In Section 7.3 the VMS formulation is introduced. The SRK method, and its peculiarities when weak boundary conditions are considered, is defined in Section 7.4. In Section 7.5 the block-preconditioning technique and the parallel solver are defined. Two different tests have been considered in the numerical results section (Section 7.6), the Taylor-Green Vortex flow and the Turbulent Channel Flow test. Finally, some conclusions are pointed out in Section 7.7.

7.2 Problem statement

7.2.1 Navier-Stokes equations

We begin with a brief revision of the Navier-Stokes equations statement, referring to Section 2.1 for a deeper description. Let Ω be a bounded domain of $\mathbb{R}^d$, where $d = 2, 3$ is the number of space dimensions, $\Gamma = \partial\Omega$ its boundary and $(0, T]$ the time interval. The strong form of the steady Navier-Stokes problem consists of finding the velocity field $\mathbf{u}$ and the pressure field p such that

$$\partial_t \mathbf{u} - \nu\Delta\mathbf{u} + \mathbf{u}\cdot\nabla\mathbf{u} + \nabla p = \mathbf{f} \qquad \text{in } \Omega \times (0, T], \tag{7.1}$$

$$\nabla\cdot\mathbf{u} = 0 \qquad \text{in } \Omega \times (0, T], \tag{7.2}$$

with $\mathbf{f}$ the force vector and ν the kinematic viscosity. Like in previous chapters, bold characters will denote vectors and tensors.

Equations (7.1) and (7.2) need to be supplied with appropriate boundary and initial conditions. The boundary Γ is divided into the Dirichlet (Γ_D) and the Neumann (Γ_N) parts such that $\Gamma_D \cup \Gamma_N = \Gamma$ and $\Gamma_D \cap \Gamma_N = \emptyset$. Then, the boundary and initial conditions can be written as

$$\mathbf{u} = \mathbf{u}_g \qquad \text{on } \Gamma_D \times (0, T], \tag{7.3}$$

$$(-p\mathbf{I} + \nu(\nabla\mathbf{u} + \nabla\mathbf{u}^T))\cdot\mathbf{n} = \mathbf{t}_N \qquad \text{on } \Gamma_N \times (0, T], \tag{7.4}$$

$$\mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0(\mathbf{x}) \qquad \text{in } \Omega \times \{0\}, \tag{7.5}$$

$\mathbf{n}$ being the unit outward vector normal to Γ.

From equations (7.1)-(7.5), and making use of the notation defined in Section 2.2, one can derive the weak form of the problem, which consists in finding $[\mathbf{u}, p] \in \mathbf{L}^2(0, T; \mathcal{V}_0) \times L^1(0, T; \mathcal{Q}_0)$ such that

$$(\partial_t \mathbf{u}, \mathbf{v}) + B(\mathbf{u}, (\mathbf{u}, p), (\mathbf{v}, q)) = \langle \mathbf{f}, \mathbf{v} \rangle \qquad \forall \mathbf{v} \in \mathcal{V}_0, \quad \forall q \in \mathcal{Q}_0, \tag{7.6}$$

satisfying the initial condition (7.5) in a weak sense. Here $\mathcal{V}_0 := \mathbf{H}_0^1(\Omega)$ and $\mathcal{Q}_0 := L^2(\Omega)/\mathbb{R}$ and the form $B(\mathbf{u}, (\mathbf{u}, p), (\mathbf{v}, q))$ is defined as

$$B(\mathbf{u}, (\mathbf{u}, p), (\mathbf{v}, q)) := \nu(\nabla\mathbf{u}, \nabla\mathbf{v}) + b(\mathbf{u}, \mathbf{u}, \mathbf{v}) - (p, \nabla\cdot\mathbf{v}) + (q, \nabla\cdot\mathbf{u}) \tag{7.7}$$

with the triliniar form of the convective term $b(\mathbf{u}, \mathbf{v}, \mathbf{w})$ defined in its skewsymmetric version

$$b(\mathbf{u}, \mathbf{v}, \mathbf{w}) = \frac{1}{2}(\mathbf{u}\cdot\nabla\mathbf{v}, \mathbf{w}) - \frac{1}{2}(\mathbf{v}, \mathbf{u}\cdot\nabla\mathbf{w}) + \frac{1}{2}(\mathbf{v}, (\mathbf{u}\cdot\mathbf{n})\mathbf{w})_{\Gamma_N}. \tag{7.8}$$

We refer to Section 2.3 for a more complete description of the variational formulation of the Navier-Stokes equations.

7.2.2 Weak Dirichlet boundary conditions

There are some situations in which the strong imposition of the boundary conditions may have negative effects on the simulation performance. That is the case, for instance, of LES methods for wall-bounded flow problems. It is well known that LES methods have several difficulties when trying to solve wall-bounded problems, see for instance [169, 168]. The issue of solving this kind of problems using a LES method is that the boundary layer that appear next to the wall can not be captured properly using relatively coarse meshes. Then, there is the need of using very thin elements next to the wall in order to have at least one point in the viscous sublayer layer ($y^+ \sim 1$), see [150]. Usually, this requirement is satisfied by the use of stretched meshes, with very thin elements on the boundary and coarser ones in the middle of the channel.

Despite of that, in many engineering problems it is more important to capture properly the large-scale flow properties than the fine-scales, focusing on the effect of the boundary layers on the mean flow instead of trying to solve properly the boundary layer itself. In this direction, an approach is to make use of the fact that the velocity profile at the boundary layer has been shown to have a relation to the wall-normal distance, [150]. Knowing this relation, one can impose weak boundary conditions on the Dirichlet boundary enforcing that the traction generated at that walls is the one given by an analytical expression. This approach was followed by Bazilevs and co-workers in [25], showing that weakly-imposed boundary conditions provide the same results than strongly-imposed ones when using stretched meshes, but the former improves significantly the accuracy when using uniform meshes.

When weak Dirichlet boundary conditions are used, the functional space of the test functions loose the property of having zero trace on Γ, then $\mathbf{v} \in \mathcal{V}$ and $q \in \mathcal{Q}$. Therefore, we have to consider the terms that arise when we use integration by parts to get the weak form of the Navier-Stokes equation (7.6). That is to include the following terms to the bilinear form.

$$\begin{aligned} B_\Gamma(\mathbf{u},(\mathbf{u},p),(\mathbf{v},q)) = {} & B(\mathbf{u},(\mathbf{u},p),(\mathbf{v},q)) \\ & - ((-p\mathbf{I} + 2\nu\nabla^s\mathbf{u}) \cdot \mathbf{n}, \mathbf{v})_\Gamma + \frac{1}{2}((\mathbf{u}\cdot\mathbf{n})\mathbf{u},\mathbf{v})_\Gamma. \end{aligned} \tag{7.9}$$

Assuming that $\mathbf{u} \cdot \nabla = 0$, the term that is in charge of enforcing a given traction on the wall can be written as

$$(\tau_w \mathbf{t}, \mathbf{v})_{\Gamma_D} = \left(\tau_w \frac{(\mathbf{u} - \mathbf{u}_g)}{\|\mathbf{u} - \mathbf{u}_g\|}, \mathbf{v} \right)_{\Gamma_D} = (\alpha_b(\mathbf{u} - \mathbf{u}_g), \mathbf{v})_{\Gamma_D}, \tag{7.10}$$

where $\mathbf{t}$ denotes the normalized vector acting on the direction of the traction and τ_w the wall shear stress magnitude. The term (7.10) is nothing else than a penalty term with a parameter

$$\alpha_b := \frac{\tau_w}{\|\mathbf{u} - \mathbf{u}_g\|}. \tag{7.11}$$

In addition to this term, we also consider other boundary terms that arise when the so called Nitche's method is used, see [143, 170]. Then, the resulting bilinear form (equivalent to (7.7)) will read

$$\begin{aligned} B_{weak}(\mathbf{u},(\mathbf{u},p),(\mathbf{v},q)) = {} & B_\Gamma(\mathbf{u},(\mathbf{u},p),(\mathbf{v},q)) \\ & + (\alpha_b(\mathbf{u}-\mathbf{u}_g),\mathbf{v})_{\Gamma_D} - ((q\mathbf{I}+2\nu\nabla^s\mathbf{v})\cdot\mathbf{n},(\mathbf{u}-\mathbf{u}_g))_{\Gamma_D}. \end{aligned} \tag{7.12}$$

Note that the sign of the pressure test function in the second term of (7.12) is changed to keep the skewsymmetry of the velocity-pressure blocks.

The definition of the traction parameter τ_w comes from the minimization of the residual of the Spalding equation

$$y^+ = f(u^+) = u^+ + e^{-\chi B}\left(e^{\chi u^+} - 1 - \chi u^+ - \frac{(\chi u^+)^2}{2} - \frac{(\chi u^+)^3}{6}\right), \tag{7.13}$$

being $\chi = 0.4$, $B = 5.5$, $y^+ := \frac{y u_\tau}{\nu}$ the wall distance and $u^+ := \frac{\|\mathbf{u}_h\|}{u_\tau}$ the mean flow velocity in non-dimensional wall units. In this case, the distance to the wall is approximated to be proportional to the wall-normal mesh size, $y = \frac{h_b}{C_b}$, and $\tau_w = u_\tau^2$. Here, h_b denotes the wall-normal mesh size and C_b a positive algorithmic constant.

A deeper description of this method as well as the algorithm for computing the parameter α_b can be found in [25]. However, in that work, the authors impose strongly the normal component of the velocity ($\mathbf{u}\cdot\mathbf{n} = 0$ on Γ_D), a thing we want to avoid in the current work. As noticed in the introduction of the present chapter, the normal vector is not well defined on the nodes that belong to curved boundaries. Thus, we will split the penalty term distinguishing between the normal and tangential counterparts.

Let us consider a function $\mathbf{v} \in \mathcal{V}$. We can split its normal and tangential components as follows

$$\mathbf{v} = (\mathbf{v}\cdot\mathbf{n})\mathbf{n} + (\mathbf{v} - (\mathbf{v}\cdot\mathbf{n})\mathbf{n}) = (\mathbf{n}\otimes\mathbf{n})\mathbf{v} + (\mathbf{I} - \mathbf{n}\otimes\mathbf{n})\mathbf{v}. \tag{7.14}$$

Introducing the decomposition (7.14) into the penalty term $(\alpha_b(\mathbf{u}-\mathbf{u}_g),\mathbf{v})_{\Gamma_D}$, and considering two different penalty parameters for each component, $\alpha_{b,n}$ and $\alpha_{b,t}$, we obtain the equivalent penalty terms

$$(\alpha_{b,n}(\mathbf{u}-\mathbf{u}_g),(\mathbf{n}\otimes\mathbf{n})\mathbf{v})_{\Gamma_D} + (\alpha_{b,t}(\mathbf{u}-\mathbf{u}_g),(\mathbf{I}-\mathbf{n}\otimes\mathbf{n})\mathbf{v})_{\Gamma_D}. \tag{7.15}$$

In this case, the normalized tangential vector appearing in (7.10) reads $\mathbf{t} = \frac{(\mathbf{I}-\mathbf{n}\otimes\mathbf{n})(\mathbf{u}-\mathbf{u}_g)}{\|(\mathbf{I}-\mathbf{n}\otimes\mathbf{n})(\mathbf{u}-\mathbf{u}_g)\|}$.

We define the wall-normal component parameter to be proportional to the original definition of the penalty term in [24], leading to the following parameter definitions

$$\alpha_{b,n} := \beta \frac{C_b \nu}{h_b}, \tag{7.16}$$

$$\alpha_{b,t} := \frac{\tau_w}{\|(\mathbf{I} - \mathbf{n} \otimes \mathbf{n})(\mathbf{u} - \mathbf{u}_g)\|}. \tag{7.17}$$

In (7.16), β is a constant that we can tune to adjust the model. An assessment of the parameter selection is done in Section 7.6.2.

Replacing the penalty terms from (7.15) into (7.12), we get a new expression for the bilinear form

$$\begin{aligned} B_{weak}(\mathbf{u}, (\mathbf{u}, p), (\mathbf{v}, q)) = {} & B_\Gamma(\mathbf{u}, (\mathbf{u}, p), (\mathbf{v}, q)) \\ & + (\alpha_{b,n}(\mathbf{u} - \mathbf{u}_g), (\mathbf{n} \otimes \mathbf{n})\mathbf{v})_{\Gamma_D} + (\alpha_{b,t}(\mathbf{u} - \mathbf{u}_g), (\mathbf{I} - \mathbf{n} \otimes \mathbf{n})\mathbf{v})_{\Gamma_D} \\ & - ((\mathbf{u} - \mathbf{u}_g), (q\mathbf{I} + 2\nu \nabla^s \mathbf{v}) \cdot \mathbf{n})_{\Gamma_D}. \end{aligned} \tag{7.18}$$

7.3 The VMS method as an LES model

In this section we will give some insights about the suitability of a particular definition of a VMS method as an LES model. Here we consider the application of the term by term OSS method developed in Chapter 5 with inf-sup stable elements, together with weak boundary conditions imposition. For the sake of completeness of the present chapter, a brief review of the VMS method is firstly given (see Section 2.4 for a deeper understanding). Later, the particular definition of the mixed FE method with convection stabilization through OSS is stated, refreshing the formulation given in Section 5.3.3. Finally, the equivalence of such formulation with an LES model is discussed.

7.3.1 VMS framework

In order to define the semi-discrete problem, we consider a FE partition $\mathcal{T}_h$ of the domain Ω, from which we construct the conforming FE spaces for the velocity field, $\mathcal{V}_h \subset \mathcal{V}$, and for the pressure field, $\mathcal{Q}_h \subset \mathcal{Q}$. The Galerkin Finite Element problem equivalent to (7.6) consists in finding $[\mathbf{u}_h, p_h] \in \mathbf{L}^2(0, T; \mathcal{V}_{g,h}) \times L^1(0, T; \mathcal{Q}_{0,h})$ such that

$$(\partial_t \mathbf{u}_h, \mathbf{v}_h) + B(\mathbf{u}_h, (\mathbf{u}_h, p_h), (\mathbf{v}_h, q_h)) = \langle \mathbf{f}, \mathbf{v}_h \rangle \qquad \forall \mathbf{v}_h \in \mathcal{V}_{0,h}, \forall q_h \in \mathcal{Q}_{0,h}. \tag{7.19}$$

With the boundary and initial conditions (7.3)-(7.5) satisfied in a weak sense. In (7.19) the subsets $\mathcal{V}_{0,h}$ and $\mathcal{Q}_{0,h}$ denote the set of functions belonging to $\mathcal{V}_h$ and $\mathcal{Q}_h$, respectively, with zero trace on Γ_D. Moreover, the velocity field function space is defined as $\mathcal{V}_{g,h} := \{\mathbf{v}_h \in \mathcal{V}_h : \mathbf{v}_h|_{\Gamma_D} = \mathbf{u}_g\}$.

When weak boundary conditions are considered, the problem results in finding $[\mathbf{u}_h, p_h] \in \mathbf{L}^2(0,T;\mathcal{V}_h) \times L^1(0,T;\mathcal{Q}_h)$ such that

$$(\partial_t \mathbf{u}_h, \mathbf{v}_h) + B_{weak}(\mathbf{u}_h, (\mathbf{u}_h, p_h), (\mathbf{v}_h, q_h)) = \langle \mathbf{f}, \mathbf{v}_h \rangle \qquad \forall \mathbf{v}_h \in \mathcal{V}_h, \forall q_h \in \mathcal{Q}_h. \tag{7.20}$$

Note that in this case $\mathbf{u}_h, \mathbf{v}_h \in \mathcal{V}_h$ and $p_h, q_h \in \mathcal{Q}_h$.

Henceforward, we will assume that in the case when Dirichlet boundary conditions are considered, $\Gamma_D \neq \emptyset$, they will be imposed weakly, making use of the bilinear form defined in (7.18).

Following the VMS approach, [101], we consider a two-scale decomposition of the continuous spaces as $\mathcal{V} = \mathcal{V}_h \oplus \widetilde{\mathcal{V}}$ and $\mathcal{Q} = \mathcal{Q}_h \oplus \widetilde{\mathcal{Q}}$. Where $\widetilde{\mathcal{V}}$ and $\widetilde{\mathcal{Q}}$ are infinite-dimesnional subscale spaces that complement the FE spaces $\mathcal{V}_h$ and $\mathcal{Q}_h$, respectively. Then, introducing the VMS decomposition into (7.6) with the bilinear definition (7.18), we get the following semi-discrete problem: find $[\mathbf{u}_h, p_h] \in \mathbf{L}^2(0,T;\mathcal{V}_h) \times L^1(0,T;\mathcal{Q}_h)$ such that

$$\begin{aligned}(\partial_t \mathbf{u}_h, \mathbf{v}_h) + (\partial_t \tilde{\mathbf{u}}, \mathbf{v}_h) + B_{weak}(\mathbf{a}, (\mathbf{u}_h, p_h), (\mathbf{v}_h, q_h)) \\ + (\tilde{\mathbf{u}}, \mathcal{L}^*_{\mathbf{a}}(\mathbf{v}_h, q_h))_h - (\tilde{p}, \nabla \cdot \mathbf{v}_h)_h = \langle \mathbf{f}, \mathbf{v}_h \rangle \quad \forall \mathbf{v}_h \in \mathcal{V}_h, \forall q_h \in \mathcal{Q}_h,\end{aligned} \tag{7.21}$$

where $\mathcal{L}^*_{\mathbf{a}}(\mathbf{v}_h, q_h)$ is the formal adjoint operator of the momentum equation, defined as

$$\mathcal{L}^*_{\mathbf{a}}(\mathbf{v}_h, q_h) := -\nabla \cdot (2\nu \nabla^s \mathbf{v}_h) - \mathbf{a} \cdot \nabla \mathbf{v}_h - \nabla q_h. \tag{7.22}$$

The fourth and fifth terms in (7.21) involving the velocity and pressure subscales, respectively, come from an element-wise integration by parts, in which the boundary terms $(\mathbf{v}_h, \nu \mathbf{n} \cdot \nabla \tilde{\mathbf{u}})_{\partial h}$ and $(q_h, \mathbf{n} \cdot \tilde{\mathbf{u}})_{\partial h}$ have been neglected (the subscript ∂h is used to denote the sum over all elements of the integral on the boundary of each element). It also involves the approximation $b(\mathbf{a}, \tilde{\mathbf{u}}, \mathbf{u}_h) \approx -(\tilde{\mathbf{u}}, \mathbf{a} \cdot \nabla \mathbf{v}_h)$ which implies neglecting $(\mathbf{v}_h, \mathbf{n} \cdot \mathbf{a}\tilde{\mathbf{u}})_{\partial h}$ and $(\tilde{\mathbf{u}}, \nabla \cdot \mathbf{a}\,\mathbf{v}_h)$. These approximations are discussed in [55] together with the choice of $\mathbf{a}$ which defines the type of scale splitting (linear or nonlinear), see also [56].

The resulting semi-discrete problem (7.21) cannot be solved directly since the subscales belong to infinite-dimensional spaces, $\tilde{\mathbf{u}} \in \widetilde{\mathcal{V}}$ and $\tilde{p} \in \widetilde{\mathcal{Q}}$. Therefore, the modelling of the small scales is needed in order to obtain a feasible method. Considering the general case where the subscales are treated as time-dependent, and approximating the Navier-Stokes operator by two stabilization parameters τ_m^{-1} and τ_c^{-1} (see for example [55]), the fine scale problem can be written as

$$\partial_t \tilde{\mathbf{u}} + \tau_m^{-1} \tilde{\mathbf{u}} = \mathcal{P}(\mathbf{R}_u), \tag{7.23}$$

$$\tau_c^{-1} \tilde{p} = \mathcal{P}(R_p). \tag{7.24}$$

In (7.23)-(7.24) $\mathcal{P}$ denotes the projection onto the space of subscales. The vector $\mathbf{R} = [\mathbf{R}_u, R_p]^T$ is the residual of the Navier-Stokes equations (7.1)-(7.2), with

$$\mathbf{R}_u = \mathbf{f} - \partial_t \mathbf{u}_h - \nabla \cdot (2\nu \nabla^s \mathbf{u}_h) + \mathbf{a} \cdot \nabla \mathbf{u}_h + \nabla p, \tag{7.25}$$

$$R_p = -\nabla \cdot \mathbf{u}_h, \tag{7.26}$$

and the expressions of the stabilization parameters τ_m and τ_c is

$$\tau_m = \left(\frac{c_1 \nu}{h^2} + \frac{c_2 |\mathbf{a}|}{h} \right)^{-1}, \tag{7.27}$$

$$\tau_c = \frac{h^2}{c_1 \tau_m}, \tag{7.28}$$

where h is the mesh size and c_1 and c_2 are algorithmic constants.

Alternatively to (7.28) one can consider the following definition of τ_c

$$\tau_c = c_c \left(\nu + \frac{c_1}{c_2} h |\mathbf{u}_h| \right), \tag{7.29}$$

which results from introducing (7.27) into (7.28) and considering a new constant c_c.

Equations (7.21)-(7.28) define a generic VMS method that can be tuned depending on the selection of three different ingredients. The first one is the definition of the projection, $\mathcal{P}$, used in (7.23)-(7.24). One possible choice is to select the projection as the identity, $\mathcal{P} := \mathbf{I}$, which would lead to the ASGS method. Another option is to choose the subscale space to be orthogonal to the FE space, then $\mathcal{P} := \Pi_h^{\perp} = \mathbf{I} - \Pi_h$, which lead to the OSS method introduced in [50]. The second ingredient that can vary the definition of the VMS method is the nonlinearity of the advective velocity $\mathbf{a}$. One can choose between a linear approach taking only the FE counterpart, $\mathbf{a} = \mathbf{u}_h$, or a nonlinear advection term with $\mathbf{a} = \mathbf{u}_h + \tilde{\mathbf{u}}$. Finally the velocity subscale time-dependency is the third ingredient that specializes the VMS method. We can consider the definition (7.23), resulting in a dynamic approach for the velocity subscale, or we could assume quasi-static velocity subscales. In the last case, (7.23) would read $\tau_m^{-1} \tilde{\mathbf{u}} = \mathcal{P}(\mathbf{R}_u)$, and the second term in (7.21) would be removed.

All combinations resulting from changing the different definitions of the VMS method have been studied for LES of incompressible turbulent flows in [56] (see Chapter 4). The final definition of the VMS method used in this chapter has been developed in Section 5.3.3 and is restated in the following subsection.

7.3.2 OSS with implicit projection for mixed FE methods

Let us consider a quasi-static definition of the velocity and pressure subscales, neglecting the temporal derivative $\partial_t \tilde{\mathbf{u}}$ in (7.23). Regarding the definition of the advection velocity, we set $\mathbf{a} = \mathbf{u}_h$, resulting in a linear approach for the velocity subscale. Furthermore, we select an orthogonal definition of the subscale space, $\mathcal{P} := \Pi_h^{\perp} = \mathbf{I} - \Pi_h$.

This setting for equal-order FEs would result in the so called *static linear OSS (Sta-Lin-OSS)* method described in Chapter 4. In the cited chapter, an assessment of the different choices that can conform the VMS method have been assessed, showing that all methods give similar results. Despite of that, the dynamic and nonlinear version with orthogonal subscales was shown to be the most efficient choice in terms of computational cost.

An alternative to the residual-based OSS method is to consider a term-by-term projection approach, proposed in [52]. Instead of having all residuals coupled, result of considering the residual projection in (7.23) and the full adjoint operator in (7.21), only the essential terms to guarantee stability are taken into account. That is to replace the fourth term in (7.21) by the orthogonal projection of the convective term plus the orthogonal projection of the pressure gradient term. Note that we have to keep the orthogonal projection of the pressure gradient when equal-order elements are used to satisfy the inf-sup condition (2.36). Using static and linear subscales, this approach reads: find $[\mathbf{u}_h, p_h] \in \mathbf{L}^2(0,T;\mathcal{V}_h) \times L^1(0,T;\mathcal{Q}_h)$ such that

$$\begin{aligned}(\partial_t \mathbf{u}_h, \mathbf{v}_h) + B_{weak}(\mathbf{u}_h, (\mathbf{u}_h, p_h), (\mathbf{v}_h, q_h)) + \left(\tau_m \Pi_h^{\perp}(\mathbf{u}_h \cdot \nabla \mathbf{u}_h), \mathbf{u}_h \cdot \nabla \mathbf{v}_h\right)_h \quad (7.30)\\ + \left(\tau_m \Pi_h^{\perp}(\nabla p_h), \nabla q_h\right)_h - (\tilde{p}, \nabla \cdot \mathbf{v}_h)_h = \langle \mathbf{f}, \mathbf{v}_h \rangle \quad \forall \mathbf{v}_h \in \mathcal{V}_h, \forall q_h \in \mathcal{Q}_h.\end{aligned}$$

This approach has been tested in Chapter 5 where an implicit treatment of the projections was considered. Together with the static and linear version of the ASGS method, the performance of the mentioned methods was tested for the LES of incompressible turbulent flows, and compared against an inf-sup stable mixed FE method with convection stabilization through orthogonal subscales.

We will refer to the inf-sup stable mixed FE method with convection stabilization through orthogonal subscales as ISS-OSS method. It is defined with an inf-sup stable velocity-pressure pair composed by Q_2/Q_1 or Q_3/Q_2 elements with a term-by-term OSS method that only considers the convection stabilization term.

After all this considerations, the final semi-discrete problem in which the current work is based consists in finding $[\mathbf{u}_h, p_h] \in \mathbf{L}^2(0,T;\mathcal{V}_h) \times L^1(0,T;\mathcal{Q}_h)$ such that

$$\begin{aligned}(\partial_t \mathbf{u}_h, \mathbf{v}_h) + B_{weak}(\mathbf{u}_h, (\mathbf{u}_h, p_h), (\mathbf{v}_h, q_h)) + (\tau_m \mathbf{u}_h \cdot \nabla \mathbf{u}_h, \mathbf{u}_h \cdot \nabla \mathbf{v}_h)_h \quad (7.31)\\ - (\tau_m \boldsymbol{\eta}_h, \mathbf{u}_h \cdot \nabla \mathbf{v}_h)_h + (\tau_c \nabla \cdot \mathbf{u}_h, \nabla \cdot \mathbf{v}_h)_h = \langle \mathbf{f}, \mathbf{v}_h \rangle \quad \forall \mathbf{v}_h \in \mathcal{V}_h, \forall q_h \in \mathcal{Q}_h,\end{aligned}$$

being $\boldsymbol{\eta}_h := \Pi_h(\mathbf{u}_h \cdot \nabla \mathbf{u}_h)$ the projection of the convective term into the FE space, satisfying

$$(\tau_m \boldsymbol{\eta}_h, \mathbf{v}_h) = (\tau_m \mathbf{u}_h \cdot \nabla \mathbf{u}_h, \mathbf{v}_h) \quad \forall \mathbf{v}_h \in \mathcal{V}_h \quad (7.32)$$

In (7.31), the last term of the left-hand side, $(\tau_c \nabla \cdot \mathbf{u}_h, \nabla \cdot \mathbf{v}_h)_h$, is usually called grad-div stabilization term, and in general it is not required to ensure stability of the problem. Nevertheless, when an inf-sup stable discretization is used, the influence of the grad-div stabilization on the accuracy of the method is important. It has been shown that this term has an influence on the mass conservation [130], but it also influences the computational cost of the linear solvers [146, 95]. In Chapter 5, the effect of this term is assessed for the Taylor-Green Vortex problem and the Turbulent Channel Flow test using an ISS-OSS method.

7.3.3 Equivalence with LES models

In order to justify that the method that we use in this work acts as a LES model without the need of adding any kind of physical based extra dissipation, we focus on the energy balance equations. To do so, we replace $\mathbf{v}_h$ by $\mathbf{u}_h$ and q_h by p_h in (7.31). The resulting equation is

$$\begin{aligned}(\partial_t \mathbf{u}_h, \mathbf{u}_h) &+ (2\nu\nabla^s \mathbf{u}_h, \nabla \mathbf{u}_h) + b(\mathbf{u}_h, \mathbf{u}_h, \mathbf{u}_h) \\ &- ((-p_h\mathbf{I} + 2\nu\nabla^s\mathbf{u}_h)\cdot\mathbf{n}, \mathbf{u}_h)_\Gamma \\ &+ (\alpha_{b,n}(\mathbf{u}_h - \mathbf{u}_g), (\mathbf{n}\otimes\mathbf{n})\mathbf{u}_h)_\Gamma + (\alpha_{b,t}(\mathbf{u}_h - \mathbf{u}_g), (\mathbf{I} - \mathbf{n}\otimes\mathbf{n})\mathbf{u}_h)_\Gamma \\ &- ((\mathbf{u}_h - \mathbf{u}_g), (p_h\mathbf{I} + 2\nu\nabla^s\mathbf{u}_h)\cdot\mathbf{n})_\Gamma \\ &+ \Big(\tau_m \Pi_h^\perp(\mathbf{u}_h\cdot\nabla\mathbf{u}_h), \mathbf{u}_h\cdot\nabla\mathbf{u}_h\Big)_h + (\tau_c\nabla\cdot\mathbf{u}_h, \nabla\cdot\mathbf{u}_h)_h = \langle \mathbf{f}, \mathbf{u}_h\rangle .\end{aligned} \tag{7.33}$$

Using the skewsymmetric convetive term defined in (7.8), the third term in (7.33) reads

$$\begin{aligned}b(\mathbf{u}_h, \mathbf{u}_h, \mathbf{u}_h) &= \frac{1}{2}(\mathbf{u}_h\cdot\nabla\mathbf{u}_h, \mathbf{u}_h) - \frac{1}{2}(\mathbf{u}_h, \mathbf{u}_h\cdot\nabla\mathbf{u}_h) + \frac{1}{2}\left((\mathbf{u}_h\cdot\mathbf{n})\mathbf{u}_h, \mathbf{u}_h\right)_\Gamma \\ &= \frac{1}{2}\left((\mathbf{u}_h\cdot\mathbf{n})\mathbf{u}_h, \mathbf{u}_h\right)_\Gamma = \left\|(\mathbf{u}_h\cdot\mathbf{n})^{1/2}\mathbf{u}_h\right\|_\Gamma^2 .\end{aligned}$$

Making use of the orthogonality of the projection $\Pi_h^\perp$, splitting the boundary terms into its normal and tangential components, and operating, equation (7.33) can be rewritten as

$$\begin{aligned}&\frac{d}{dt}\left(\frac{1}{2}\|\mathbf{u}_h\|^2\right) + 2\nu\|\nabla^s\mathbf{u}_h\|^2 + \left\|\tau_m^{1/2}\Pi_h^\perp(\mathbf{u}_h\cdot\nabla\mathbf{u}_h)\right\|_h^2 + \left\|\tau_c^{1/2}\nabla\cdot\mathbf{u}_h\right\|_h^2 \\ &+ (\alpha_{b,n}(\mathbf{u}_h - \mathbf{u}_g) - (-p_h\mathbf{I} + 2\nu\nabla^s\mathbf{u}_h)\cdot\mathbf{n}, (\mathbf{n}\otimes\mathbf{n})\mathbf{u}_h)_\Gamma \\ &+ (\alpha_{b,t}(\mathbf{u}_h - \mathbf{u}_g) - (-p_h\mathbf{I} + 2\nu\nabla^s\mathbf{u}_h)\cdot\mathbf{n}, (\mathbf{I} - \mathbf{n}\otimes\mathbf{n})\mathbf{u}_h)_\Gamma \\ &+ \left\|(\mathbf{u}_h\cdot\mathbf{n})^{1/2}\mathbf{u}_h\right\|_\Gamma^2 \\ &- ((\mathbf{u}_h - \mathbf{u}_g), (p_h\mathbf{I} + 2\nu\nabla^s\mathbf{u}_h)\cdot\mathbf{n})_\Gamma \\ &= \langle \mathbf{f}, \mathbf{u}_h\rangle .\end{aligned} \tag{7.34}$$

Let us analyze in more detail the implication of each term of (7.34) into the energy evolution of the problem. To simplify the notation, we rewrite (7.34) as

$$\begin{aligned}\frac{dk_h}{dt} = &-\int_\Omega \varepsilon_{mol,h}\, d\Omega - \sum_e \int_{\Omega_e} \widetilde{\varepsilon}_m\, d\Omega - \sum_e \int_{\Omega_e} \widetilde{\varepsilon}_c\, d\Omega \\ &- \int_\Gamma \varepsilon_{\mathbf{a},h}\, d\Gamma - \int_\Gamma \varepsilon_{N,h}\, d\Gamma \\ &+ \int_\Omega W_{f,h}\, d\Omega + \int_\Gamma W_{t,h}\, d\Gamma,\end{aligned} \tag{7.35}$$

being $k_h := \frac{1}{2}\left\|\mathbf{u}_h\right\|^2$ the kinetic energy of the resolved scales, $\varepsilon_{mol,h} := 2\nu\left(\nabla^s\mathbf{u}_h\right)^2$ the molecular dissipation introduced by the resolved scales, $\widetilde{\varepsilon}_m := \tau_m\left(\Pi_h^\perp(\mathbf{u}_h\cdot\nabla\mathbf{u}_h)\right)^2$ the energy transfer through the momentum subscale, $\widetilde{\varepsilon}_c := \tau_c\left(\nabla\cdot\mathbf{u}_h\right)^2$ the energy transfer through the continuity subscale, $\varepsilon_{\mathbf{a},h} := (\mathbf{u}_h\cdot\mathbf{n})\mathbf{u}_h\cdot\mathbf{u}_h$ the dissipation introduced by the convective term on the boundary, $\varepsilon_{N,h} := (\mathbf{u}_h-\mathbf{u}_g)\cdot(p_h\mathbf{I}+2\nu\nabla^s\mathbf{u}_h)\cdot\mathbf{n}$ the dissipation introduced by the Nitsche's terms, $W_{f,h} := \mathbf{f}\cdot\mathbf{u}_h$ the work of the volumetric external forces, and $W_{t,h} := [\alpha_{b,n}(\mathbf{u}_h-\mathbf{u}_g)-(-p_h\mathbf{I}+2\nu\nabla^s\mathbf{u}_h)\cdot\mathbf{n}]\cdot[(\mathbf{n}\otimes\mathbf{n})\mathbf{u}_h] + [\alpha_{b,t}(\mathbf{u}_h-\mathbf{u}_g)-(-p_h\mathbf{I}+2\nu\nabla^s\mathbf{u}_h)\cdot\mathbf{n}]\cdot[(\mathbf{I}-\mathbf{n}\otimes\mathbf{n})\mathbf{u}_h]$ the work of the boundary forces.

Going further, assuming that strong boundary conditions are applied, we have that $\varepsilon_{\mathbf{a},h}=\varepsilon_{N,h}=W_{t,h}=0$. For a fully developed turbulent flow, if the resolved scales are fine enough to capture the inertial subrange, the total energy of the problem accounts for almost all the energy of the problem, $k_h \approx k$. Furthermore, when external forces mainly acts on the large scales of the flow we have that $W_{f,h}\approx W_f$. Hence, knowing that the energy dissipation rate at the continuous level is

$$\frac{dk}{dt} = -\int_\Omega \varepsilon_{mol}\, d\Omega + \int_\Omega W_f\, d\Omega,$$

from the energy balance equation (7.35), we can say that our method acts properly as a LES method if

$$\sum_e \int_{\Omega_e} \widetilde{\varepsilon}_m\, d\Omega + \sum_e \int_{\Omega_e} \widetilde{\varepsilon}_c\, d\Omega \approx \int_\Omega \varepsilon_{mol,h}\, d\Omega.$$

According to [85], this condition is satisfied since

$$\frac{1}{V_e}\left\langle \int_{\Omega_e} \widetilde{\varepsilon}_m\, d\Omega \right\rangle + \frac{1}{V_e}\left\langle \int_{\Omega_e} \widetilde{\varepsilon}_c\, d\Omega \right\rangle \sim \varepsilon_{mol,h}\left[1+\mathcal{O}\left(\frac{h}{L}\right)^{2/3}\right],$$

with L the characteristic large eddies length.

7.4 Segregated Runge-Kutta time integration

7.4.1 Matricial form

Let us suppose that we solve the problem (7.1)-(7.5) using the implicit *term by term* OSS method with Inf-Sup stable elements, which definition is given by equations (7.30)-(7.31). Assuming that $\mathbf{u}_h$, p_h and $\boldsymbol{\eta}_h$ are defined by a FE interpolation from the nodal values $\{\mathbf{U}^a\}_{a=1,...,N_u}$, $\{P^b\}_{b=1,...,N_p}$ and $\{\boldsymbol{\Upsilon}^c\}_{c=1,...,N_\eta}$, the FE approximation of the velocity, pressure and projection fields can be written as

$$\mathbf{u}_h(\mathbf{x}) = \sum_{a=1}^{N_u} \boldsymbol{\phi}_a(\mathbf{x})\mathbf{U}^a, \quad p_h(\mathbf{x}) = \sum_{b=1}^{N_p} \psi_b(\mathbf{x})P^b, \quad \boldsymbol{\eta}_h(\mathbf{x}) = \sum_{c=1}^{N_\eta} \boldsymbol{\phi}_\eta(\mathbf{x})\boldsymbol{\Upsilon}^c,$$

where $\{\boldsymbol{\phi}_{a,i}\}_{a=1,...,N_u;i=1,...,d}$, $\{\psi_b\}_{b=1,...,N_p}$ and $\{\boldsymbol{\phi}_{c,i}\}_{c=1,...,N_\eta;i=1,...,d}$ are the Langrangian bases associated to $\mathcal{V}_h$ and $\mathcal{Q}_h$. N_u, N_p and N_η are the total amount of nodes for the velocity, pressure and projection fields.

The matricial form of the stabilized problem (7.30) introducing the definition (7.31) and considering strong Dirichlet boundary conditions can be written as follows.

$$\begin{bmatrix} \mathbb{M}\dot{\mathbf{U}} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbb{K}+\mathbb{C}+\mathbb{A}_\tau & \mathbb{G} & \mathbb{B}_\tau \\ \mathbb{D} & 0 & 0 \\ -\mathbb{B}_\tau^T & 0 & \mathbb{M}_\tau \end{bmatrix} \begin{bmatrix} \mathbf{U} \\ \mathbf{P} \\ \boldsymbol{\Upsilon} \end{bmatrix} = \begin{bmatrix} \mathbf{F}_u \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \tag{7.36}$$

which definition are given by

$$\begin{aligned}
&\mathbb{M}^{ab} := (\boldsymbol{\phi}_a, \boldsymbol{\phi}_b), && a,b = 1,...,N_u,\\
&\mathbb{K}^{ab} := 2\nu(\nabla\boldsymbol{\phi}_a, \nabla^s\boldsymbol{\phi}_b), && a,b = 1,...,N_u,\\
&\mathbb{C}^{ab} := (\boldsymbol{\phi}_a, \mathbf{u}\cdot\nabla\boldsymbol{\phi}_b), && a,b = 1,...,N_u,\\
&\mathbb{A}_\tau^{ab} := \tau_m(\mathbf{u}\cdot\nabla\boldsymbol{\phi}_a, \mathbf{u}\cdot\nabla\boldsymbol{\phi}_b) + \tau_c(\nabla\cdot\boldsymbol{\phi}_a, \nabla\cdot\boldsymbol{\phi}_b), && a,b = 1,...,N_u,\\
&\mathbb{G}^{ab} := -(\nabla\cdot\boldsymbol{\phi}_a, \psi_b), && a = 1,...,N_u, b = 1,...,N_p,\\
&\mathbb{D}^{ab} := (\psi_a, \nabla\cdot\boldsymbol{\phi}_b), && a = 1,...,N_p, b = 1,...,N_u,\\
&\mathbb{B}_\tau^{ab} := -\tau_m(\mathbf{u}\cdot\nabla\boldsymbol{\phi}_a, \boldsymbol{\phi}_b), && a = 1,...,N_u, b = 1,...,N_\eta,\\
&\mathbb{M}_\tau^{ab} := -\tau_m(\boldsymbol{\phi}_a, \boldsymbol{\phi}_b), && a,b = 1,...,N_\eta,
\end{aligned}$$

being a and b the node identification. Note that $\mathbb{D} = -\mathbb{G}^T$, when boundary conditions are considered.

7.4.2 Time integration through Segregated Runge-Kutta schemes

Focusing in the time integration of the problem (7.36) we consider now the application of a Segregated Runge-Kutta scheme. SRK methods have been proposed in [57] and successfully tested for laminar flows, see Chapter 6. The main goal of this kind of time integrators is the segregation of velocity and pressure computations, giving systems much easier to be solved, taking advantage of an IMEX Runge-Kutta scheme. The use of a Rung-Kutta scheme also allows the use a high order integration in time with no extra implementation efforts. Furthermore, with this kind of time integrators, an adaptive time stepping technique can be easily used.

Let us now consider the matricial problem (7.36), which can be written in a compact form in terms of the velocity and projection fields as follows

$$\mathbb{M}\dot{\mathbf{U}} = \mathcal{F}(\mathbf{U}) + \mathcal{G}(\mathbf{U}, \boldsymbol{\Upsilon}), \tag{7.37a}$$

$$-\mathbb{B}_{\tau}^{T}\mathbf{U} + \mathbb{M}_{\tau}\boldsymbol{\Upsilon} = \mathbf{0}. \tag{7.37b}$$

Where $\mathcal{F}$ and $\mathcal{G}$ are certain operators that contain the terms that will be integrated implicitly or explicitly, respectively. In particular we can define a SRK method in which the linear viscous term is integrated implicitly and the nonlinear terms, such as the convective and projection terms, and the pressure term are integrated explicitly. In this case, the operators $\mathcal{F}$ and $\mathcal{G}$ will read

$$\mathcal{F}(\mathbf{U}) := -\mathbb{K}\mathbf{U}, \tag{7.38}$$

$$\begin{aligned}\mathcal{G}(\mathbf{U}, \boldsymbol{\Upsilon}) := {} & \mathbf{F}_u - (\mathbb{C} + \mathbb{A}_\tau)\mathbf{U} - \mathbb{B}_\tau\boldsymbol{\Upsilon} \\ & + \mathbb{G}\left(\mathbb{D}\mathbb{M}^{-1}\mathbb{G}\right)^{-1}\left(\mathbb{D}\mathbb{M}^{-1}\left((\mathbb{K} + \mathbb{C} + \mathbb{A}_\tau)\mathbf{U} + \mathbb{B}_\tau\boldsymbol{\Upsilon} - \mathbf{F}_u\right)\right).\end{aligned} \tag{7.39}$$

Note that this choice will lead to a linear system of equations since the only term integrated implicitly is the viscous term. Another fact to be highlighted here is that the projection and the velocity fields are not coupled with this definition. Other choices could be used to define the operators $\mathcal{F}$ and $\mathcal{G}$, for instance, the convective and the projection terms could be integrated implicitly. In this later case, the resulting system would be nonlinear and would couple velocity and projection fields. We choose definitions (7.38)-(7.39) because we are interested on solving turbulent flows, where the time step restriction due to the hyperbolic CFL condition is often smaller than the restriction on the time step size required to capture the small time scales of the flow.

We now proceed with the time integration scheme for equation (7.37a). Let us consider an IMEX RK method with s stages, the velocity and projection at the stage i, $\mathbf{U}_i$ and $\boldsymbol{\Upsilon}_i$, for $1 \leq i \leq s$ are computed as

$$\frac{1}{\delta t}\mathbb{M}\mathbf{U}_i = \frac{1}{\delta t}\mathbb{M}\mathbf{U}_n + \sum_{j=1}^{i} a_{ij}\mathcal{F}(\mathbf{U}_j) + \sum_{j=1}^{i-1} \hat{a}_{ij}\mathcal{G}(\mathbf{U}_j, \boldsymbol{\Upsilon}_j), \tag{7.40a}$$

$$-\mathbb{B}_{\tau}^{T}\mathbf{U}_i + \mathbb{M}_{\tau}\boldsymbol{\Upsilon}_i = \mathbf{0}. \tag{7.40b}$$

where a_{ij} and $\hat{a}_{ij}$ are the coefficients of the implicit and explicit Butcher tableau, respectively. After some manipulation, we can rewrite (7.40) as

$$\frac{1}{\delta t}\mathbb{M}\mathbf{U}_i = \frac{1}{\delta t}\mathbb{M}\mathbf{U}_n + \sum_{j=1}^{i} a_{ij}\mathcal{F}(\mathbf{U}_j) + \sum_{j=1}^{i-1} \hat{a}_{ij}\mathcal{G}(\mathbf{U}_j, \mathbf{P}_j, \boldsymbol{\Upsilon}_j), \tag{7.41a}$$

$$-\mathbb{B}_\tau^T\mathbf{U}_i + \mathbb{M}_\tau\boldsymbol{\Upsilon}_i = \mathbf{0}, \tag{7.41b}$$

$$-\mathbb{D}\mathbb{M}^{-1}\mathbb{G}(\mathbf{P}_i) = \mathbb{D}\mathbb{M}^{-1}\left((\mathbb{K} + \mathbb{C} + \mathbb{A}_\tau)\mathbf{U}_i + \mathbb{B}_\tau\boldsymbol{\Upsilon}_i - \mathbf{F}_u(t_i)\right). \tag{7.41c}$$

The final update after the s stages will read as follows

$$\frac{1}{\delta t}\mathbb{M}\mathbf{U}_{n+1} = \frac{1}{\delta t}\mathbb{M}\mathbf{U}_n + \sum_{i=1}^{s} b_i\mathcal{F}(\mathbf{U}_i) + \hat{b}_i\mathcal{G}(\mathbf{U}_i, \mathbf{P}_i, \boldsymbol{\Upsilon}_i), \tag{7.42a}$$

$$-\mathbb{B}_\tau^T\mathbf{U}_{n+1} + \mathbb{M}_\tau\boldsymbol{\Upsilon}_{n+1} = \mathbf{0}, \tag{7.42b}$$

$$-\mathbb{D}\mathbb{M}^{-1}\mathbb{G}(\mathbf{P}_{n+1}) = \mathbb{D}\mathbb{M}^{-1}\left((\mathbb{K} + \mathbb{C} + \mathbb{A}_\tau)\mathbf{U}_{n+1} + \mathbb{B}_\tau\boldsymbol{\Upsilon}_{n+1} - \mathbf{F}_u(t_{n+1})\right). \tag{7.42c}$$

Note that equations (7.41c) and (7.42c) are equivalent to solve a Darcy type problem, with the following expression

$$\begin{bmatrix} \mathbb{M} & \mathbb{G} \\ \mathbb{D} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{U}^* \\ \mathbf{P}_k \end{bmatrix} = \begin{bmatrix} \mathbf{F}_u(t_k) - (\mathbb{K} + \mathbb{C} + \mathbb{A}_\tau)\mathbf{U}_k - \mathbb{B}_\tau\boldsymbol{\Upsilon}_k \\ 0 \end{bmatrix}, \tag{7.43}$$

with k being i or $n+1$, depending of the equation that we are solving, and $\mathbf{U}^*$ an auxiliar velocity field, which satisfies the discrete incompressibility constraint. Then, in practice, equations (7.41c) and (7.42c) are implemented as a Darcy problem (7.43), which can be efficiently solved through block preconditioning technique like what it is explained in subsection 7.5.1.

The system of equations (7.41a)-(7.42c) that arise when using a SRK time integration scheme over the particular definition of a VMS method exposed in Section 7.3.2, will be denoted as *Segregated Variational MultiScale* (SVMS) method.

7.4.3 Weak Dirichlet boundary conditions for SRK schemes

In Section 7.2.2 we have stated that in some situations weak Dirichlet boundary conditions may be used. In this case, the SRK time integration schemes have to take into account the boundary terms described in (7.9) and (7.12), which implementation is discussed in this subsection.

The use of weak Dirichlet boundary conditions give a matricial equation equivalent to (7.36) that can be written as follows

$$\begin{bmatrix} \mathbb{M}\dot{\mathbf{U}} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbb{K}+\mathbb{K}_\Gamma+\mathbb{C}+\mathbb{C}_\Gamma+\mathbb{A}_\tau+\mathbb{M}_\Gamma & \mathbb{G}+\mathbb{G}_\Gamma & \mathbb{B}_\tau \\ \mathbb{D}+\mathbb{D}_\Gamma & 0 & 0 \\ -\mathbb{B}_\tau^T & 0 & \mathbb{M}_\tau \end{bmatrix} \begin{bmatrix} \mathbf{U} \\ \mathbf{P} \\ \boldsymbol{\Upsilon} \end{bmatrix} = \begin{bmatrix} \mathbf{F}_u+\mathbf{F}_{u,\Gamma} \\ \mathbf{F}_{p,\Gamma} \\ \mathbf{0} \end{bmatrix}, \tag{7.44}$$

with

$$\mathbb{M}_\Gamma^{ab} := (\alpha_{b,n}(\mathbf{n}\otimes\mathbf{n})\boldsymbol{\phi}_a, \boldsymbol{\phi}_b)_{\Gamma_D} + (\alpha_{b,t}(\mathbf{I}-\mathbf{n}\otimes\mathbf{n})\boldsymbol{\phi}_a, \boldsymbol{\phi}_b)_{\Gamma_D}, \qquad a,b=1,...,N_u,$$

$$\mathbb{K}_\Gamma^{ab} := -2\nu(\boldsymbol{\phi}_a, \nabla^s\boldsymbol{\phi}_b\cdot\mathbf{n})_\Gamma - 2\nu(\nabla^s\boldsymbol{\phi}_a\cdot\mathbf{n}, \boldsymbol{\phi}_b)_{\Gamma_D}, \qquad a,b=1,...,N_u,$$

$$\mathbb{C}_\Gamma^{ab} := \frac{1}{2}(\boldsymbol{\phi}_a, (\mathbf{u}\cdot\mathbf{n})\boldsymbol{\phi}_b)_\Gamma, \qquad a,b=1,...,N_u,$$

$$\mathbb{G}_\Gamma^{ab} := (\boldsymbol{\phi}_a, \psi_b\mathbb{I}\cdot\mathbf{n})_\Gamma, \qquad a=1,...,N_u, b=1,...,N_p,$$

$$\mathbb{D}_\Gamma^{ab} := -(\psi_a\mathbb{I}\cdot\mathbf{n}, \boldsymbol{\phi}_b)_{\Gamma_D}, \qquad a=1,...,N_p, b=1,...,N_u,$$

$$\mathbf{F}_{u,\Gamma}^a := \alpha_b(\boldsymbol{\phi}_a, \mathbf{u}_g)_\Gamma - 2\nu(\nabla^s\boldsymbol{\phi}_a\cdot\mathbf{n}, \mathbf{u}_g)_{\Gamma_D}, \qquad a=1,...,N_u,$$

$$\mathbf{F}_{p,\Gamma}^a := -(\psi_a\mathbb{I}\cdot\mathbf{n}, \mathbf{u}_g)_{\Gamma_D}, \qquad a=1,...,N_p,$$

being a and b the node identification. The velocity-pressure block keeps its skewsymmetry since $\mathbb{D}_\Gamma = -\mathbb{G}_\Gamma^T$.

Note that the incompressibility constrain equation in this case reads

$$\widetilde{\mathbb{D}}\mathbf{U} = \mathbf{F}_{p,\Gamma}, \tag{7.45}$$

being $\widetilde{\mathbb{D}} := (\mathbb{D}+\mathbb{D}_\Gamma)$. If we derive (7.45) in time and we operate with first equation of (7.44), we have that

$$\overbrace{\widetilde{\mathbb{D}}\mathbb{M}^{-1}\mathbb{M}\dot{\mathbf{U}}}^{\dot{\mathbf{F}}_{p,\Gamma}} + \widetilde{\mathbb{D}}\mathbb{M}^{-1}\left[(\widetilde{\mathbb{K}}+\widetilde{\mathbb{C}}+\mathbb{A}_\tau+\mathbb{M}_\Gamma)\mathbf{U}+\widetilde{\mathbb{G}}\mathbf{P}+\mathbb{B}_\tau\boldsymbol{\Upsilon}\right] = \widetilde{\mathbb{D}}\mathbb{M}^{-1}\widetilde{\mathbf{F}}_u, \tag{7.46}$$

with $\widetilde{\mathbb{K}} := (\mathbb{K}+\mathbb{K}_\Gamma)$, $\widetilde{\mathbb{C}} := (\mathbb{C}+\mathbb{C}_\Gamma)$, $\widetilde{\mathbb{G}} := (\mathbb{G}+\mathbb{G}_\Gamma)$ and $\widetilde{\mathbf{F}}_u := (\mathbf{F}_u+\mathbf{F}_{u,\Gamma})$. From (7.46) we get

$$\widetilde{\mathbb{D}}\mathbb{M}^{-1}\widetilde{\mathbb{G}}\mathbf{P} = \widetilde{\mathbb{D}}\mathbb{M}^{-1}\left[\widetilde{\mathbf{F}}_u - (\widetilde{\mathbb{K}}+\widetilde{\mathbb{C}}+\mathbb{A}_\tau+\mathbb{M}_\Gamma)\mathbf{U} - \mathbb{B}_\tau\boldsymbol{\Upsilon}\right] - \dot{\mathbf{F}}_{p,\Gamma}, \tag{7.47}$$

where $\dot{\mathbf{F}}_{p,\Gamma}^a = -(\psi_a\mathbb{I}\cdot\mathbf{n}, \dot{\mathbf{u}}_g)_\Gamma$, for $a=1,...,N_p$. Then, when using weak Dirichlet boundary conditions in SRK time integration schemes, the Darcy type problem equivalent to (7.43) to be solved is

$$\begin{bmatrix} \mathbb{M} & \widetilde{\mathbb{G}} \\ \widetilde{\mathbb{D}} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{U}^* \\ \mathbf{P}_k \end{bmatrix} = \begin{bmatrix} \widetilde{\mathbf{F}}_u(t_k) - (\widetilde{\mathbb{K}}+\widetilde{\mathbb{C}}+\mathbb{A}_\tau+\mathbb{M}_\Gamma)\mathbf{U}_k - \mathbb{B}_\tau\boldsymbol{\Upsilon}_k \\ \dot{\mathbf{F}}_{p,\Gamma}(t_k) \end{bmatrix}, \tag{7.48}$$

with k corresponding to the stage i or the update $n+1$.

Remark 7.4.1. *Note that the auxiliar velocity* $\mathbf{U}^*$ *in (7.48) is nothing else than the velocity derivative. Then, as the mass matrix* $\mathbb{M}$ *is the same that is considered for the momentum equation, when weak Dirichlet boundary conditions are considered for the velocity field, any condition is imposed to its time derivative. In order to ensure stability of the time discretization scheme, we also impose weak boundary conditions to the velocity derivative, leading to an equivalent Darcy problem to (7.48) like*

$$\begin{bmatrix} \widetilde{\mathbb{M}} & \widetilde{\mathbb{G}} \\ \widetilde{\mathbb{D}} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{U}^* \\ \mathbf{P}_k \end{bmatrix} = \begin{bmatrix} \widetilde{\mathbf{F}}_u(t_k) - (\widetilde{\mathbb{K}} + \widetilde{\mathbb{C}} + \mathbb{A}_\tau + \mathbb{M}_\Gamma)\mathbf{U}_k - \mathbb{B}_\tau \boldsymbol{\Upsilon}_k + \dot{\mathbf{F}}_{u,\Gamma}(t_k) \\ \dot{\mathbf{F}}_{p,\Gamma}(t_k) \end{bmatrix}. \tag{7.49}$$

Here $\widetilde{\mathbb{M}} := \mathbb{M} + \dot{\mathbb{M}}_\Gamma$, *with*

$$\dot{\mathbb{M}}_\Gamma^{ab} := (\alpha_{b,n}\delta t(\mathbf{n}\otimes\mathbf{n})\boldsymbol{\phi}_a, \boldsymbol{\phi}_b)_{\Gamma_D} + (\alpha_{b,t}\delta t(\mathbf{I} - \mathbf{n}\otimes\mathbf{n})\boldsymbol{\phi}_a, \boldsymbol{\phi}_b)_{\Gamma_D} \tag{7.50}$$

and

$$\dot{\mathbf{F}}_{u,\Gamma}^a(t_k) := (\alpha_{b,n}\delta t(\mathbf{n}\otimes\mathbf{n})\boldsymbol{\phi}_a, \dot{\mathbf{u}}_g(t_k))_{\Gamma_D} + (\alpha_{b,t}\delta t(\mathbf{I} - \mathbf{n}\otimes\mathbf{n})\boldsymbol{\phi}_a, \dot{\mathbf{u}}_g(t_k))_{\Gamma_D},$$

for each $a, b = 1, ..., N_u$.

Remark 7.4.2. *In order to give consistency to the time integration scheme, when considering also weak Dirichlet boundary conditions for the velocity derivatives, as exposed in Remark 7.4.1, the modified mass matrix* $\widetilde{\mathbb{M}}$ *will also be used at all the equations of the SRK scheme. That is to consider* $\widetilde{\mathbb{M}}$ *instead of* $\mathbb{M}$ *at the equivalent equations to (7.41a)-(7.42c).*

It is important to highlight that in the momentum equations (7.41a) and (7.42a), the penalty terms introduced by $\mathbb{M}_\Gamma$ *and* $\dot{\mathbb{M}}_\Gamma$ *are of the same order since, as it is seen in (7.50),* $\dot{\mathbb{M}}_\Gamma$ *is scaled by* δt.

7.4.4 Adaptive time-stepping technique

One of the advantages when using a multi-stage time integration scheme, like a SRK method, is that an adaptive time stepping technique can be used straightforwardly. In fact, for a SRK scheme, the usage of an adaptive time stepping technique only requires an extra evaluation of the final velocity update, which involve a mass matrix solver, see [88, 89].

This kind of strategy allow us to adapt the time step size to the flow conditions dynamicaly, that is to change the time step when the physical scales of the flow change, e.g., transition to turbulence. Many authors have been used adaptive time stepping to accurately solve the incompressible Navier-Stokes equations, see for instance [72, 115, 118, 178].

Our approach to define an adaptive time step is to follow the *PI11 controller* method described in [173] and suggested in [89], which computes the time step size as

$$\delta t_{n+1} = \left(\frac{\epsilon}{r_{n+1}}\right)^{1/k} \left(\frac{r_n}{r_{n+1}}\right)^{1/k} \frac{\delta t_n}{\delta t_{n+1}} \delta t_n,$$

with $\epsilon = 0.8 \cdot TOL$, where TOL is a given tolerance that we take as $1 \cdot 10^{-5}$ and 0.8 is a safety factor. The local error is $r_{n+1} = \|\mathbf{U} - \hat{\mathbf{U}}\|$ if the error per step (EPS) is controlled or $r_{n+1} = \|(\mathbf{U} - \hat{\mathbf{U}})/\delta t_n\|$ if the error per unit step (EPUS) is controlled. In the former case $k = p + 1$ (EPS) and for the second one (EPUS) $k = p$, p being the order of the time integration scheme which has been used to compute the estimated velocity $\hat{\mathbf{U}}$. In this work we will favour the EPS definition.

Moreover, when treating with explicit versions of the SRK scheme, we have to limit the time step size to guarantee that the CFL condition is satisfied. When either the parabolic CFL condition, $\mathrm{CFL}_\nu = \frac{\nu \delta t}{(h/p^2)^2}$, or the hyperbolic CFL condition, $\mathrm{CFL}_u = \frac{u \delta t}{(h/p^2)}$, exceed the bound 1.0, the time step is rejected and reduced according to the safety factor ϵ, $\delta t_{n+1,\mathrm{new}} = \epsilon \delta t_{n+1,\mathrm{old}}$.

7.5 Scalable solvers for incompressible Navier-Stokes equations

In this section we will discuss the construction of efficient and scalable solvers for the stabilized Navier-Stokes equations in turbulent flows through BDDC methods in parallel contexts. First of all we will have a look at the block preconditioning technique used in the different systems involved in the SRK scheme. After that, the main characteristics of the BDDC solvers used in this work will be described.

7.5.1 Block-preconditioning technique for the segregated problem

Let us assume that the matricial problem (7.37) is solved with an IMEX version of the SRK scheme given by the operators $\mathcal{F}$ and $\mathcal{G}$ defined in (7.38)-(7.39). In this case, at the stage i, for $1 \leq i \leq s$, the resulting matrix from the system (7.40) can be factorized into an exact LU matrix product as follows

$$\mathbb{A}_u := \begin{bmatrix} \mathbb{M}_\tau & -\mathbb{B}_\tau^T \\ \mathbb{B}_\tau & \mathbb{K}_\tau \end{bmatrix} = \begin{bmatrix} \mathbb{I} & 0 \\ \mathbb{B}_\tau \mathbb{M}_\tau^{-1} & \mathbb{I} \end{bmatrix} \begin{bmatrix} \mathbb{M}_\tau & -\mathbb{B}_\tau^T \\ 0 & \mathbb{S}_u \end{bmatrix}, \tag{7.51}$$

where $\mathbb{S}_u := \mathbb{K}_\tau + \mathbb{B}_\tau^T \mathbb{M}_\tau^{-1} \mathbb{B}_\tau$ is the Schur complement with respect to $\mathbf{U}$. We define a block-preconditioner, $P(\mathbb{A}_u)$, based on inexact factorization of $\mathbb{A}_u$ given by the upper triangular matrix in (7.51), reading

$$P(\mathbb{A}_u) := \begin{bmatrix} \mathbb{M}_\tau & -\mathbb{B}_\tau^T \\ 0 & \mathbb{S}_u \end{bmatrix}^{-1} = \begin{bmatrix} \mathbb{M}_\tau^{-1} & \mathbb{M}_\tau^{-1} \mathbb{B}_\tau^T \mathbb{S}_u^{-1} \\ 0 & \mathbb{S}_u^{-1} \end{bmatrix}. \tag{7.52}$$

To solve the system resulting from (7.37) we use a preconditioned GMRES method with the preconditioner defined in (7.52). The application of the preconditioner $P(\mathbb{A}_u)$ simplifies to the computation of the inverse of the diagonal blocks, $\mathbb{M}_\tau^{-1}$ and $\mathbb{S}_u^{-1}$. The mass matrix inverse is approximated by one application of a diagonal DD preconditioner constructed from the diagonal values of $\mathbb{M}_\tau$. The Schur complement is approximated to the velocity-velocity block matrix, $\mathbb{S}_u \approx \mathbb{K}_\tau$, neglecting the contribution of the term coming from the projections $\mathbb{B}_\tau^T\mathbb{M}_\tau^{-1}\mathbb{B}_\tau$. This approximation is equivalent to consider a solution corresponding to a non-consistent formulation in which streamline diffusion is added, ensuring stability of the preconditioner. At his turn, the inverse $\mathbb{K}_\tau^{-1}$ is approximated by one application of a BDDC preconditioner over this matrix.

Focusing on the Darcy-type equation (7.43), we also consider an LU factorization of the form

$$\mathbb{A}_p := \begin{bmatrix} \mathbb{M} & \mathbb{G} \\ \mathbb{D} & 0 \end{bmatrix} = \begin{bmatrix} \mathbb{I} & 0 \\ \mathbb{D}\mathbb{M}^{-1} & \mathbb{I} \end{bmatrix} \begin{bmatrix} \mathbb{M} & \mathbb{G} \\ 0 & \mathbb{S}_p \end{bmatrix}, \tag{7.53}$$

being $\mathbb{S}_p := -\mathbb{D}\mathbb{M}^{-1}\mathbb{G}$ the Schur complement with respect to $\mathbf{P}$. In this case we also consider a block-preconditioner based on an inexact LU factorization of $\mathbb{A}_p$, determined by the upper triangular matrix,

$$P(\mathbb{A}_p) := \begin{bmatrix} \mathbb{M} & \mathbb{G} \\ 0 & \mathbb{S}_p \end{bmatrix}^{-1} = \begin{bmatrix} \mathbb{M}^{-1} & -\mathbb{M}^{-1}\mathbb{G}\mathbb{S}_p^{-1} \\ 0 & \mathbb{S}_p^{-1} \end{bmatrix}. \tag{7.54}$$

The resolution of system (7.43) with a preconditioned GMRES method involve the computation of the inverse of matrices $\mathbb{M}$ and $\mathbb{S}_p$. The approach followed to approximate $\mathbb{M}^{-1}$ is the same we have defined for $\mathbb{M}_\tau^{-1}$, with one application of a diagonal DD preconditioner. Concerning about the Schur complement, we approximate it by a pressure Laplacian $\mathbb{S}_p \approx \mathbb{L}_p$. Another time, its inverse is approximated by one application of a BDDC preconditioner over such matrix.

The matricial systems arising from equations (7.42a) and (7.42b) are solved using a preconditioned GMRES with a diagonal DD preconditioner to approximate $\mathbb{M}^{-1}$ and $\mathbb{M}_\tau^{-1}$, respectively.

7.5.2 BDDC solver

As stated above, the solution of the resulting discrete equations (7.41a)-(7.42c) is done through a block-preconditioning technique that, for the diagonal blocks, makes use of BDDC preconditioners. The BDDC method, firstly introduced by Dohrmann [66], basically consists on decomposing the FE mesh into a set of subdomains that share an interface between each other. The nodes in this interface are classified by corners, edges and faces (this last class only appears in the 3D case), and conform a coarse problem that is used to precondition the finer. The continuity among subdomains is only weakly enforced, e.g., the value on subdomain corners and mean values on subdomain edges and subdomain faces, so each interface entity is equivalent to a node at the coarse level.

The main properties of BDDC preconditioner that make this kind of preconditioners very suitable to reach extreme scalability are:

- The mathematically supported extremely aggressive coarsening, keeping the sparsity pattern of the coarse matrix similar to the original system matrix.
- The possibility to compute in parallel the coarse and the fine components due to the orthogonal definition of the coarse space with respect to the finer.
- The possibility to extend the algorithm with a multilevel approach, based on the similarity of the coarse matrix pattern.

Regarding the last point, a fully-distributed, communicator-aware, recursive, and interlevel-overlapped message-passing implementation of the multilevel balancing domain decomposition by constraints (MLBDDC) preconditioner has been developed in [134]. In that work, excellent weak scalability results have been obtained for the resolution of a 3D Laplacian and elasticity problems. Hence, as a result of the block-preconditioning technique described in Section 7.5.1, we end up with a Laplacian-type and an elasticity-type problem, which can use the implementation of the MLBDDC proposed in [134].

In some situations the mesh partition may lead to singular coarse problems, especially for the Laplacian-type problem. This problem can be overcome by the use of a perturbed BDDC with the addition of a properly scaled mass matrix at the preconditioner, see [21]. This approach will be used in Chapter 8, where a non-structured mesh partition is used.

7.6 Numerical experiments

In this section we show the numerical results for two different incompressible turbulent flow tests. The suitability of the OSS-ISS method, with weak boundary conditions and SRK time integration schemes, as a LES model for large scale problems is assessed in these tests. First, we analyze the advantages of using SRK methods in the TGV problem, where a weak scalability analysis is performed. Note that in this test we do not have boundary conditions since periodic boundary conditions are imposed in all directions. Then, the suitability of weak boundary conditions in wall-bounded flows is analyzed in the Turbulent Channel Flow test.

In [58] has been shown that OSS-ISS has a great potential as a LES model, especially when we the computational cost is taken into account. The experiments of this chapter aim to check the performance of this method when it is used together with a Segregated Runge-Kutta time integration scheme.

The main goal of a SRK scheme is the velocity and pressure decoupling, allowing the use of more efficient solvers. This only happen when there is not any other term apart from the divergence in the pressure row, see the matricial form (7.36), which is a feature of the OSS-ISS method. When we solve turbulent flows, some stabilization method is needed, not only to control the convection instabilities, but also to model the effect of the small scales. The ASGS and OSS stabilization methods proposed in the cited work introduce extra terms in the pressure row, which disallow the use of SRK schemes for the time integration. This is not the case of OSS-ISS method, which only introduces an extra term in the velocity-velocity block. Then, the applicability of SRK schemes to this method is straight forward as it has been exposed in subsection 7.4.2.

7.6.1 Taylor Green Vortex flow

The TGV is a widely used benchmark for turbulent numerical simulations. It consists in a free-evolving flow from an analytical solution, with a computational domain which is the cube $(0, 2\pi)^3$ and with periodical boundary conditions (see, e.g., [31]). We refer to Section 3.2.3 to a deeper description of this benchmark test. The initial solution is given by the following expression

$$\mathbf{u}(x,y,z,0) = \begin{pmatrix} u_x \\ u_y \\ u_z \end{pmatrix} = \begin{pmatrix} u_0 \cos(x)\sin(y)\sin(z) \\ -u_0 \sin(x)\cos(y)\sin(z) \\ 0 \end{pmatrix} \tag{7.55}$$

$$p(x,y,z,0) = p_0 + \frac{1}{16}\left(\cos(2x) + \cos(2y)\right)\left(\cos(2z) + 2\right),$$

with

$$u_0 = \frac{2}{\sqrt{3}} \sin\left(\gamma + \frac{2\pi}{3}\right).$$

We choose $\gamma = 0$, which gives the mean initial velocity $u_0 = 1$. We solve the TGV problem using a Reynolds number $\mathrm{Re} = 1600$.

Setting

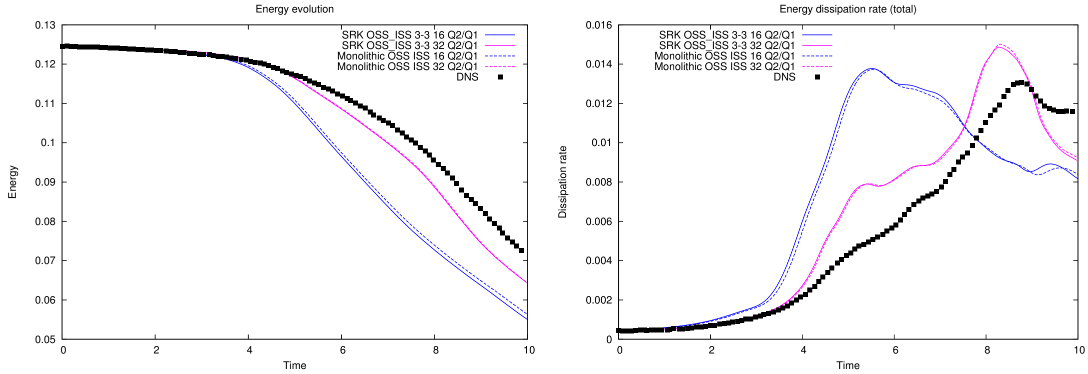
The problem is solved from $t = 0.0$ to $T = 10.0$ and the results are compared against a DNS by Brachet et al [31]. We discretize the domain using different choices of the number of elements and the order of approximation, with either Q_2/Q_1 or Q_3/Q_2 velocity-pressure pairs.

As it was shown in [58] (see also Chapter 5), the most appropriate algorithmic parameters for the TGV test when using inf-sup stable discretizations are $c_1 = 12.0$, $c_2 = 4.0$ and $c_c = 4.0$.

Monolithic vs Segregated Runge-Kutta

A first point that we want to analyze is the comparison between the results obtained with a monolithic approach, using a Crank-Nicolson time integration scheme, against the ones computed using the IMEX version of the SRK method. In the later case we use the *(3-3)* scheme, see Appendix C.

The results shown in Figure 7.1 have been computed using two different discretizations, one with 16^3 Q_2/Q_1 elements and the other one with 32^3 Q_2/Q_1. In Figure 7.1(a) the energy evolution is depicted and in Figure 7.1(b) we show the energy dissipation rate. It is seen that very little differences appear using the SRK time integration method instead of a monolithic Crank-Nicolson scheme. It is also clear that the differences become smaller when the mesh is refined.



(a) Energy evolution

(b) Total energy dissipation rate

FIGURE 7.1: Energy and Total energy dissipation rate evolution with 33^3 velocity DOFs

h-p refinement and adaptive time-stepping technique

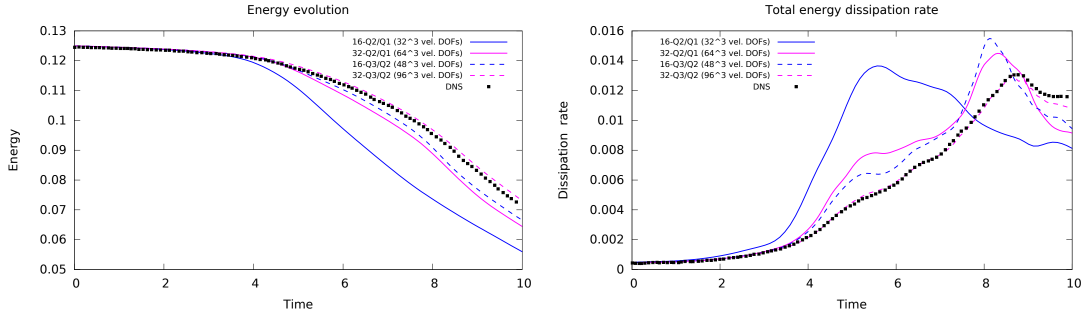
In Fig 7.1 we clearly see that refining the mesh, the results computed using the OSS-ISS method converge to the DNS. Moreover, it has been shown in Chapter 5 that a 32^3 Q_3/Q_2 elements mesh is fine enough to get very accurate results with the OSS-ISS method.

One of the main problems when using explicit or IMEX time integration schemes is the restriction on the CFL number. In the case of the IMEX version of the SRK scheme, the diffusive term is treated implicitly and no restriction has to be satisfied for the parabolic CFL, but the convective term is treated explicitly. Then, we have to restrict the time step size in order to satisfy the parabolic CFL condition, $\text{CFL}_u = \delta t u/h < 1$. This restriction becomes more important when the mesh is refined, i. e. h is reduced.

An important goal of using SRK schemes is the possibility to easily implement an adaptive time stepping technique, see section 7.4.4. The important point of using this kind of techniques is that one can dynamicaly (and automaticaly) adapt the time step in order to satisfy both, the physical and numerical, requirements. The physical requirement on the time step size will be given by the change in the solution, while the numerical requirement will be given by the CFL condition.

In order to check the performance of the SRK method using adaptive time stepping, we solve the TGV problem with different mesh sizes and interpolations. Particularly, we use 16^3 and 32^3 elements mesh with Q_2/Q_1 and Q_3/Q_2 discretizations, depicted in

Fig 7.2. Looking at the kinetic energy evolution (Figure 7.2(a)) and at the total energy



(a) Energy evolution

(b) Total energy dissipation rate

FIGURE 7.2: Energy and total energy dissipation rate evolution refining the mesh

dissipation rate (Figure 7.2(b)), we can clearly see that when the mesh is refined the solution tends to the DNS result. It is important to highlight here the improvement of the results when higher order of approximation is used. We see that using Q_3/Q_2 elements we get better results than using Q_2/Q_1, even with less DOFs. This behaviour is observed comparing the 16^3 Q_3/Q_2 elements mesh, with 48^3 velocity DOFs, against the 32^3 Q_2/Q_1 elements discretization, which have 64^3 velocity DOFs. It is seen that the former mesh have better agreement with the DNS solution.

As stated above, the mesh refinement has a direct implication in the time step size when explicit or IMEX schemes are used. To determine the influence of this refinement using an adaptive time stepping technique, we show the time step evolution for the four cases considered in this subsection, see Figure 7.3. The initial time step is $\delta t_0 = 5 \cdot 10^{-2}$ for all cases. It is seen that the coarser mesh (16^3 Q_2/Q_1 elements) allow a higher time step, which is constantly increased until the difference on the solution between to time steps is too high. It occurs around $t = 4.0$, moment from which the time step size is adapted, remaining between 0.1 and 0.12. The 16^3 Q_3/Q_2 and the 32^3 Q_2/Q_1 show moreorless the same behaviour, with a first stage where the time step is increased until the CFL restriction is violated, point at which the time step is reduced. This pattern is repeated until the solution between two time steps is different enough to require a smaller time step to give more accurate solution. In what concerns the finest mesh, 32^3 Q_3/Q_2, the CFL restriction prevails over the physical phenomena restriction. Consequently, the time step increase-decrease pattern is followed during the whole simulation.

Scalability analysis

All the solutions have been computed with FEMPAR (Finite Element Multiphysics and massively PARallel) numerical software. FEMPAR is an open source in-house developed, parallel hybrid OpenMP/MPI, object-oriented (OO) framework which, among

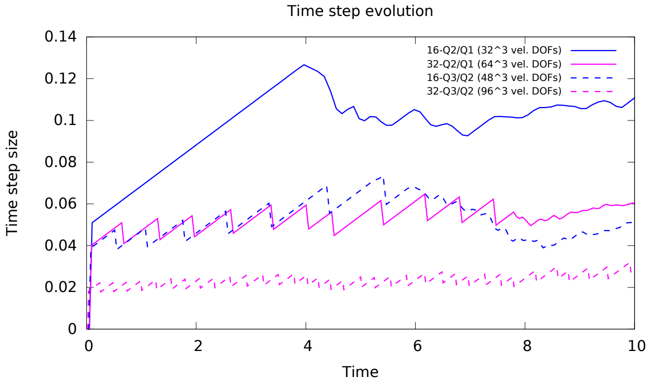


FIGURE 7.3: Time step size evolution refining the mesh

other features, provides the basic tools for the efficient parallel distributed-memory implementation of substructuring domain decomposition solvers [19, 20]. Under this parallel framework, an important issue is the scalability of the algorithms when large scale problems are solved. This means that when refining the mesh, there is the need of having scalable solvers that do not increase the elapsed CPU time when the number of processors are increased. To check the suitability of the explicit SRK algorithm for large scale problems we perform a weak scalability test solving the TGV problem.

Let us denote H the unidirectional size of the domain partition and L the unidirectional domain length. The TGV problem is solved in a domain of size L^3, with structured subdomain partitions of size H^3, and with a uniformly structured mesh composed of elements of size h^3. In a weak scalability test we keep the local meshes constant, that is to keep the number of elements per processor constant, which means that $H/h = C$. In order to preserve the numerical properties, i. e. keep h constant, instead of reducing h we increase L. Hence, as the reference problem is computed in a domain of size $L_0 = 2\pi$, the scalabitilty analysis will be done in a domain of size $L = \beta L_0$, being $\beta = \{1, 2, 3, 4, 5\}$. The reference partition size is $H_0 = L_0/4$ and will be kept constant.

We compute the solution after one time step of size $\delta t = 5 \cdot 10^{-2}$ with the IMEX version of the *(1-2)* scheme (see Appendix C), using the Q_2/Q_1 element type, and reporting the number of solver iterations and the CPU time consumed by each system resolution. Different local mesh sizes are considered to see the effect of the local mesh size on the scalability of the solver. In particular, we will consider three cases: $H/h = \{4, 8, 12\}$. The total amount of processors used to solve the problem will be $(L/H)^3 = (4\beta)^3$. The smallest mesh used to solve the problem in this analysis is a 16^3 Q_2/Q_1 elements mesh, while the biggest one is a 240^3 Q_2/Q_1 elements mesh, which give more than 680 million DOFs.

The following results are computed with a two-level BDDC solver, considering the corners, edges and faces of the subdomain partitions as part of the coarse system. We recall that the first step of a SRK *(1-2)* scheme consist in six system resolutions that are specified in Algorithm 1. Note that the first system solver is only needed at the first step, since after that $p_1 = p_n$.

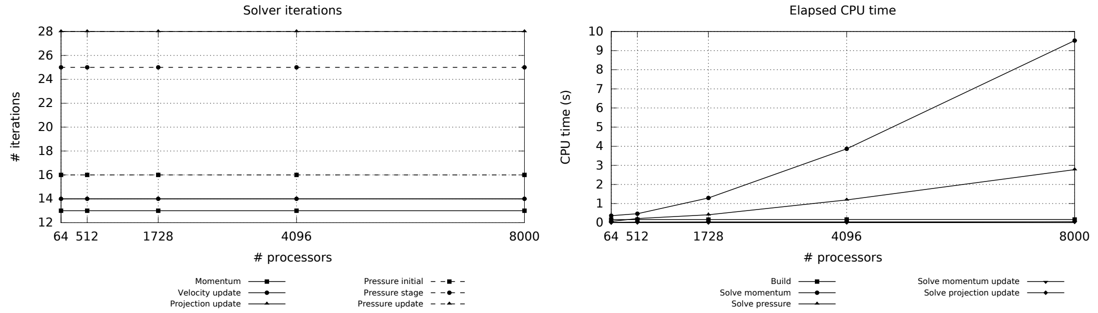
Algorithm 1: SRK system resolutions for one time step using the *(1-2)* scheme

```
1. Solve initial pressure (equation (7.41c)).
2. Solve second-stage momentum system (equations (7.41a)-(7.41b)).
3. Solve second-stage pressure (equation (7.41c)).
4. Solve velocity update (equation (7.42a)).
5. Solve projection update (equation (7.42b)).
6. Solve pressure update (equation (7.42c)).
```

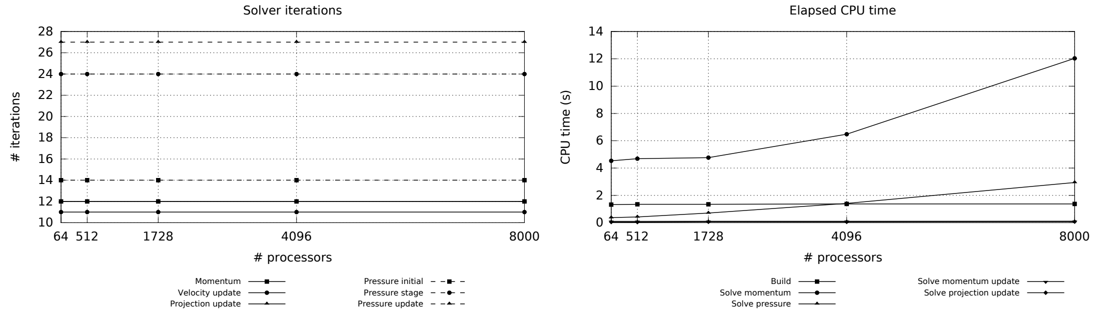
Figure 7.4 depicts the solver iterations and the elapsed CPU time for the weak scalability analysis with the three different local mesh sizes. All the six system solvers are depicted in the solver iteration plots (Figure 7.4(a), Figure 7.4(c) and Figure 7.4(e)), while for the CPU time graphics (Figure 7.4(b), Figure 7.4(d) and Figure 7.4(f)) the pressure system resolutions have been grouped and the building time has been added. The building time includes the time consumed to integrate all matrices and vectors and the time consumed computing the preconditioners.

Looking at the solver iterations, we see that the number of iterations for each system resolution is kept constant when the number of processors is increased, independently of the local mesh size. We notice that when the local mesh sizes increase, the number of iterations decrease a little for all systems. Another thing that we must point out looking at the solver iterations column in Figure 7.4 is that, as expected, the velocity update and the projection update give the same number of solver iterations. This is caused by the fact that both systems solve a mass matrix, with the only difference that the velocity update mass matrix is scaled by $\frac{1}{\delta t}$. We also see that, in terms of solver iterations, the hardest system is the pressure computation (equations (7.41c) and (7.42c)).

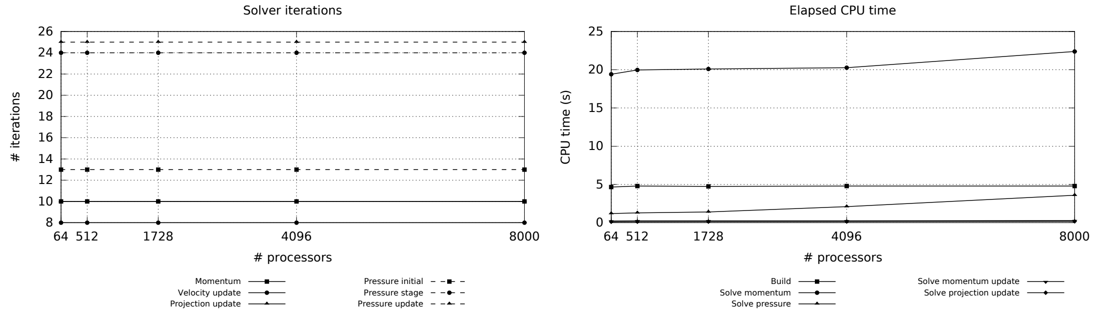
In what refers to the consumed CPU time, a first conclusion we can take looking at Figure 7.4(b), Figure 7.4(d) and Figure 7.4(f) is that the most expensive resolution is the momentum system (equations (7.41a)-(7.41b)). Although the number of solver iterations is smaller than the pressure computations, in terms of time consumed, the momentum equations resolution is much expensive, even counting together the three pressure solvers. This difference in the computational time required for the momentum equations versus the pressure ones, is due to the number of DOFs that are being solved. We have to take into account that both velocity and projection fields are computed with a second order polynomial, Q_2, while the pressure is first order, Q_1. Then, we not only have two vectorial fields versus a scalar field, but we have a second order interpolation versus a first order. Leaving aside the mass matrix blocks, the size of the system of equations to be solved in the velocity block of the momentum computation is ~ 24



(a) Iterations, $H/h = 4$ (b) CPU time, $H/h = 4$



(c) Iterations, $H/h = 8$ (d) CPU time, $H/h = 8$



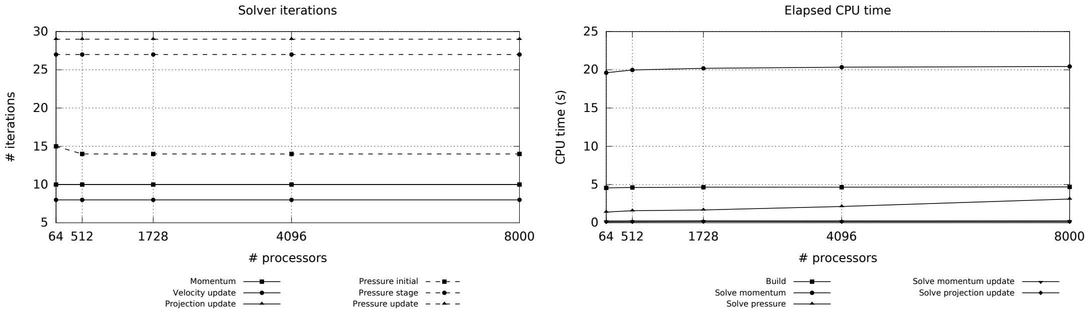
(e) Iterations, $H/h = 12$ (f) CPU time, $H/h = 12$

FIGURE 7.4: Weak scalability test. Number of iterations and elapsed CPU time for solving one time step of the TGV problem with different local mesh sizes.

times larger than the one that arise for the pressure block of the Darcy computations, taking into account that the pressure system is solved using (7.48).

keeping with the discussion about the CPU time consumed shown in Figure 7.4, we see that the computations of the momentum and pressure systems do not scale in terms of time. This performance has an easy explanation, which is given by the ratio between the fine and coarse system sizes. The coarse system size only depends on the number of processors (subdomains) used by the BDDC method. When corners, edges and faces of the subdomain are considered, the total amount of degrees of freedom of the coarse system is equivalent to a local mesh of Q_2 elements (without the interior node) with as many elements as subdomains. Consequently, for a $H/h = 4$ local mesh, we can only expect to have weak scalability until 64 (4^3) processors, for $H/h = 8$ until 512 (8^3) processors, and for $H/h = 12$ until 1728 (12^3) processors. In fact this is what we are seeing in Figure 7.4, where, taking into account that the local meshes have the interior DOF inside each element, the results show the expected scalability performance.

In order to have even better results, one can reduce the size of the coarse system by considering only the corners and the edges of the subdomain. In Figure 7.5 we depict the results obtained in this case, for a $H/h = 12$ local mesh size. It is seen that when the subdomain faces are ignored, the number of solver iterations increase, see Figure 7.5(a). Despite that, in the trade-off of having more solver iterations with smaller system of equations, ignoring the subdomain faces in the coarse system worth. As we can see in Figure 7.5(b), the momentum system is scalable up to 8.000 processors since the local problem is more expensive than the coarse one. We can not scale in time for the pressure solver because the local problem is much smaller than the coarse as we use Q_1 elements for the pressure field.



(a) Iterations

(b) CPU time

FIGURE 7.5: Weak scalability test. Number of iterations and elapsed CPU time for solving one time step of the TGV problem with $H/h = 12$ and only corners and edges at the coarse level.

Although that the weak scalability of the proposed method have been demonstrated up to 8000 processors, we could go further making use of a three-level method as it is defined in the multilevel implementation MLBDDC proposed in [134].

7.6.2 Turbulent channel flow at $Re_\tau = 395$

In the previous section we have tested the suitability of the proposed VMS method using an IMEX SRK time integration scheme for the LES simulation of homogeneous turbulence. After that, we aim to show the performance of such methods for wall-bounded turbulent flows. In order to check the behaviour of the proposed approach, we solve the TCF test with a Reynolds number based on the wall velocity $Re_\tau = 395$.

One of the main purposed of this test is to assess the accuracy of the solution when using the weak boundary conditions formulation given by the bilinear form described in (7.18). Then, the TCF test is suitable for this assessment.

Setting

The TCF problem is solved in a boxed domain of size $(2\pi\delta \times 2\delta \times 2/3\pi\delta)$. Where the x-direction is the flow direction, also called stream-wise direction, the y-direction is the wall-normal direction, and the z-direction is the span-wise direction. Homogeneous Dirichlet boundary conditions for the velocity DOFs are weakly imposed on wall-normal direction boundaries ($y = -\delta$ and $y = \delta$), while periodic boundary conditions are defined on the stream-wise and span-wise directions. We refer to Section 3.3.2 for a deeper description of this test.

We select the problem parameters according to the DNS computation performed by Moser et al in [140, 121] (MKM-DNS), so we can compare our results with the cited work. The bulk mean velocity and the half channel height are taken equal to one, $\bar{U} = 1$ and $\delta = 1$. The estimated Reynolds number based on the bulk mean velocity is known, $Re = \bar{U}2\delta/\nu \approx 13,750$ (see [150]). Therefore, one can obtain the value of the viscosity, $\nu = 1.4545 \cdot 10^{-4}$, and from the Reynolds number based on the friction velocity, we can determine the friction velocity magnitude: $u_\tau = Re_\tau \nu/\delta = 5.745 \cdot 10^{-2}$. Thus, the wall shear stress reads $\tau_w = u_\tau^2 = 3.3010 \cdot 10^{-3}$. A force equivalent to a pressure gradient is imposed to drive the movement of the flow in the stream-wise direction, $f_x = \tau_w/\delta$.

As we have done in Section 5.5.3, an initial solution proposed in [139] is used to reach the statistically steady state faster, which consists in a unidirectional velocity profile over which is added a fluctuation:

$$\begin{aligned}
u_x &= C\left(1 - y^8\right) + \epsilon\frac{L_x}{2}\sin(\pi y)\cos\left(\frac{4\pi x}{L_x}\right)\sin\left(\frac{2\pi z}{L_z}\right), \\
u_y &= -\epsilon(1 + \cos(\pi y))\sin(\pi y)\sin\left(\frac{4\pi x}{L_x}\right)\sin\left(\frac{2\pi z}{L_z}\right), \\
u_z &= -\epsilon\frac{L_z}{2}\sin\left(\frac{4\pi x}{L_x}\right)\sin(\pi y)\cos\left(\frac{2\pi z}{L_z}\right).
\end{aligned} \tag{7.56}$$

The constant C is chosen in a such way that the field without fluctuations would have a bulk mean velocity $\bar{U} = 1.0$. The fluctuation constant ϵ is 10% of the bulk mean velocity.

To solve the problem we use two different meshes, composed by 16^3 Q_2/Q_1 and Q_3/Q_2 elements. As a weak imposition of the boundary conditions is considered, we use uniformly distributed elements along the wall-normal direction. The effect of considering uniform or stretched meshes in the wall-normal direction is assessed in [25], concluding that similar results are achieved when weak Dirichlet boundary conditions are used.

According to the results obtained in Chapter 5, the algorithmic constants that appear in (7.27) and (7.28) are chosen as $c_1 = 12.0$ and $c_2 = 8.0$. The constant that appear in the weak boundary conditions formulation is set to $C_b = 32.0$. We will analyze the effect of the constant c_c for the current problem setting as it plays an important role when using inf-sup stable elements.

Effect of c_c in uniform meshes

In Section 5.5.3 we have analyzed the effect of the constant c_c that appear in equation (7.29) for the case in which we have strong imposition of the boundary conditions together with the use of stretched meshes that capture the boundary layer. In this section we aim to check the performance of the method when changing the c_c algorithmic parameter in uniform meshes and weak boundary conditions imposition. In this case, in order to avoid the influence of the weak imposition of the wall normal component, we restrict strongly the wall-normal velocity, e.g. $\mathbf{u} \cdot \mathbf{n} = 0$ on Γ_D.

In Figure 7.6 we show the energy evolution of the solution for different values of c_c from $t = 0.0$ to $t = 100.0$, and computed on a 16^3 Q_2/Q_1 mesh. It can be shown that for uniform meshes, high values of c_c makes the solution more energy conserving, so the energy is not dissipating properly. The value $c_c = 1.0$ is the first case that reaches the statistically stable state, at $t \sim 280.0$, with the lowest global energy. Then, in following sections we will use $c_c = 1.0$.

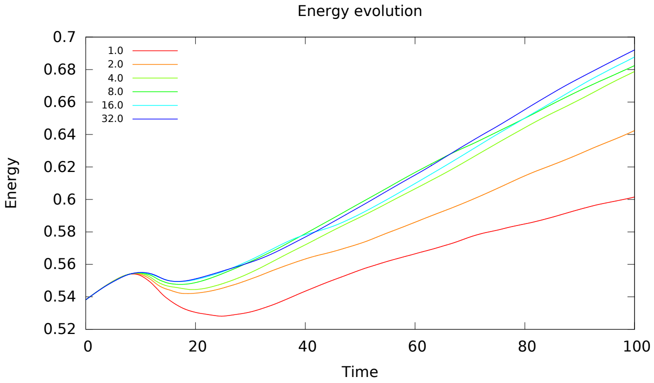


FIGURE 7.6: Energy evolution for the TCF test for different values of c_c.

Influence of the wall-normal component

A significant difference on the approach followed for the weak enforcement of the Dirichlet boundary conditions considered in this work, with respect to the originally proposed in [25], is the treatment of the normal component. In [25] a no-penetration condition is considered, which means that the normal component of the velocity is imposed strongly, $\mathbf{u} \cdot \mathbf{n} = 0$ on Γ_D. We propose to keep the weak boundary imposition even in the wall-normal direction. The main advantage of keeping the same approach for all components is revealed in curved boundaries. In that case, the different treatment of each component could lead to tedious code implementation.

As stated in section 7.2.2, we define two difference penalty parameters. One for the tangential component, based on the law of the wall, and another one for the wall-normal component, which is proportional to the original penalty parameter proposed in [24], $\alpha_{b,n} := \beta \frac{C_b \nu}{h_b}$. As it is obvious, when $\beta \rightarrow \infty$ the solution at the wall becomes closer to the one obtained imposing strongly the normal component, but the penalty term may disturb the proper solution.

In order to determine the best value for β, we solve the TCF test from $t = 0.0$ to $t = 100.0$, with different values $\beta = \{1.0, 10, 100, 1000\}$ in a 16^3 Q_2/Q_1 elements mesh. The problem is solved using an IMEX SRK scheme with adaptive time adaptive technique.

In Figure 7.7 we show the energy evolution for the different settings of β. We clearly see that the choice of β has not a large implication on the results unless too low values of $beta$ are used. For too low values of β, the case of $\beta = 1.0$, the solution starts dissipating too much energy at the early stages of the simulation. When increasing β, the results tend to the ones obtained considering a strong imposition of the normal component, denoted by *strong*. However, as it has been said before, the proposed method on uniform meshes is too little dissipative, so we will choose the value of $\beta = 10.0$, which is a little bit more dissipative than greater values.

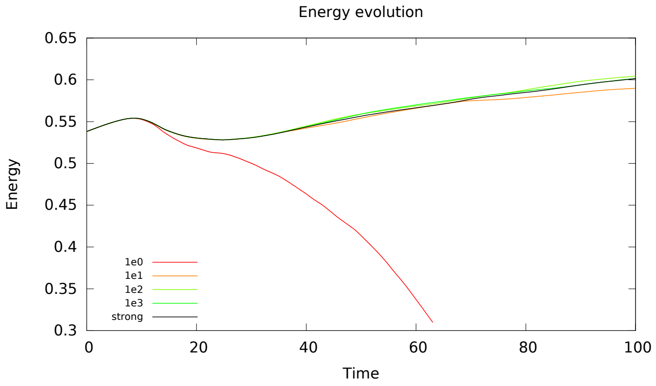


FIGURE 7.7: Energy evolution for the TCF test for different values of β.

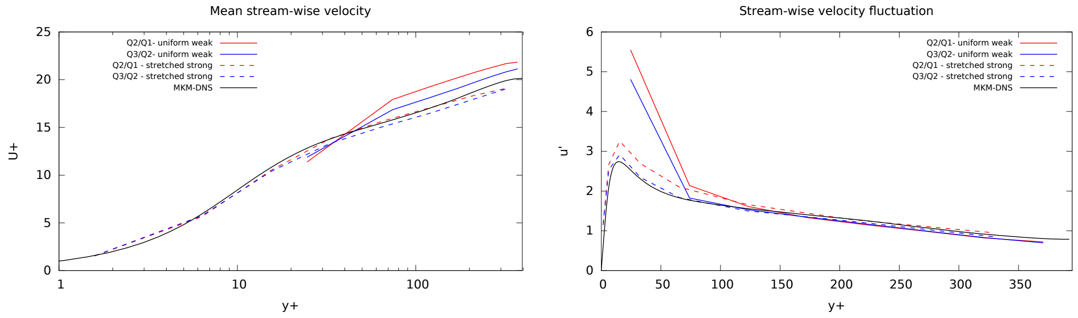
Comparison between strong and week boundary conditions imposition

Once analyzed the effect of the β parameter when Dirichlet boundary conditions are imposed weakly for the wall-normal component, let us check the performance of the approach followed in this work compared against a strong Dirichlet boundary conditions definition. We will compare with the results presented in [58], where the TCF test was solved using strong Dirichlet boundary conditions in a stretched mesh. In Table 7.1 we state the distance to the wall of the nearest velocity DOF for an stretched and a uniform mesh, with two different discretizations.

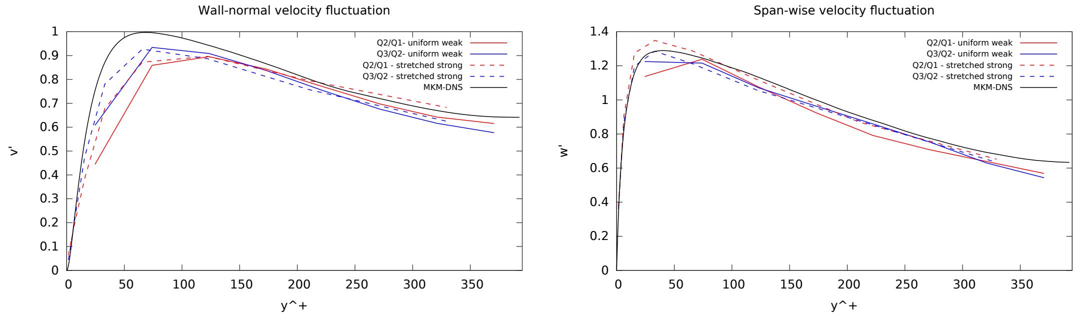
TABLE 7.1: Distance to the wall of the nearest velocity DOF in wall units.

Mesh type	Discretization	y^+
Stretched	16^3 Q_2/Q_1	1.6
	16^3 Q_3/Q_2	1.0
Uniform	16^3 Q_2/Q_1	24.7
	16^3 Q_3/Q_2	16.5

The TCF test is solved from $t = 0.0$ until the flow reach a statistically stable solution. In this case it is until $t = 300.0$ for the 16^3 Q_2/Q_1 mesh and to $t = 240.0$ for the 16^3 Q_3/Q_2 mesh. We store the statistics from the last 10.0 seconds, accumulating more than 1000 samples. The results are shown in Figure 7.8, where the mean stream-wise velocity is plotted (Figure 7.8(a)) together with the velocity fluctuations in the stream-wise, wall-normal and span-wise directions (Figs. 7.8(b), 7.8(c) and 7.8(d), respectively).



(a) Mean stream-wise velocity (b) Rms stream-wise velocity fluctuation



(c) Rms wall-normal velocity fluctuation (d) Rms span-wise velocity fluctuation

FIGURE 7.8: Mean stream-wise velocity and rms velocity fluctuations for using different discretizations.

It is seen in Figure 7.8(a) that the mean stream-wise velocity profile is a little bit far from the DNS results and also from the results obtained with strong boundary conditions in a stretched mesh. This behaviour is related to the energy evolution that we have shown in Figure 7.6, where it is seen that the method does not dissipate the energy as it should. An explanation of such behaviour is the fact that the uniform mesh has the first node located too far from the wall, as seen in Table 7.1.

Similar results are observed for the velocity fluctuations. Looking at Figure 7.8(b), where the stream-wise velocity fluctuation is depicted, we see that the nearest to the wall point is far to the DNS curve for the weak and uniform case, but we see that the remaining points fit to the desired results. As it is expected, an improvement in the solution is observed when higher order discretization is used. Same behaviour is observed for the wall-normal and span-wise velocity fluctuations, Figures 7.8(c) and 7.8(d).

7.7 Conclusions

In this chapter we have addressed the suitability of using SRK time integrations schemes for the LES of turbulent incompressible flows. A mixed FE formulation with convection stabilization through OSS method have been used for the spatial discretization of the Navier-Stokes equations, which, together with the SRK methods, conform a promising combination of spatial and temporal discretizations that are suitable for the simulation of turbulent incompressible flows. This particular combination have been denoted as *Segregated Variational MultiScale*, SVMS, method.

The application of the SVMS method to the simulation of turbulent incompressible flows has been tested with the TGV test, showing that same results are achieved when considering a SRK time integration scheme instead of the traditional Crank-Nicolson scheme. The convergence to the DNS solution has also been demonstrated when a refinement both in h and p is considered, giving more accurate results the p-refinement.

One of the advantages of using SRK schemes is the easy implementation of time adaptive techniques that allow the automatic time step adaptation to the numerical and physical requirements. This issue has also been addressed for the TGV test, where we have seen that the method is able to adapt to the numerical restriction given by the CFL conditions and to the physical requirements given by the change in the solution.

Another advantage of the SVMS methods is the possibility to use block preconditioning techniques that lead to the approximation of the inverse of Laplacian-type and elasticity-type matrices, which at his turn are suitable to be preconditioned with BDDC algorithms. The weak scalability of this approach for the resolution of one time step of the TGV test have been demonstrated up to 8000 cores. Furthermore, a three-level MLBDDC algorithm could be used to achieve weak scalability in a higher number of processors.

Moreover, aside from the TGV test, we have also checked the performance of SVMS method for the wall-bounded TCF test. In this case, we have also proposed the use of weak Dirichlet boundary conditions with the particularity of considering a wall-law based tangential traction and also a weak imposition of the wall-normal component.

This last test shows that the approach of imposing weakly the tangential and the normal velocity components on the Dirichlet boundary lead to acceptable results, even when uniform meshes with very coarse elements near the wall are considered. However, comparing the results against the ones obtained with a stretched mesh and strong boundary conditions imposition, we see that the near wall fluctuations are not properly captured. Then, in order to obtain more accurate results, a mesh with thinner elements near the wall should be considered.

We have seen that the weak imposition of the Dirichlet boundary conditions is suitable when SRK methods are considered. First, because the continuity equation is not polluted by the strong imposition of the temporal derivative of the velocity divergence on the boundary. This occurs when time-dependent boundary conditions are imposed on the Dirichlet boundary. Furthermore, the weak imposition of the velocity wall-normal component considered in this chapter, avoids the complexity of imposing strongly such component on curved boundaries.

Chapter 8

Flow around an airfoil at low Reynolds number

8.1 Introduction

As comented in Chapter 1, a common application of CFD software is the simulation of flows around airfoils. These simulations are needed by the aerospace industry to optimize their products and reduce the costs of experimentation.

The aim of this chapter is to apply all the numerical improvements that this thesis has contributed with to the resolution of a realistic problem. We consider the simulation of turbulent incompressible flow around an airfoil, particularly a NACA 0012 airfoil [106].

The NACA airfoils are widely used as a benchmark of a real life problem, note that many airplane and wind turbines wings are based on these profiles. Both numerical simulations and experiments in wind tunnels have been done with several configurations of this type of airfoil. The NACA 0012 is one of the most used configurations, see for instance [166, 137, 152], but others like the NACA 4412 are also widely studied, see [182, 94]. Another type of airfoil that are also used as a benchmark in the literature is the Aerospatiale-A airfoil, used for example in the LESFOIL project [61].

Although most of the works that can be found in the literature deal with challenging configurations, i.e. high Reynolds numbers and high angles of attack (see [107, 116, 164, 138]), we will restrict ourselves to a more modest scenario. That is to consider low Reynolds number with moderate angles of attack. The main reason for considering this case is the possibility to compare the results obtained with a weak treatment of the Dirichlet boundary conditions against the obtained with a strong imposition of the Dirichlet boundaries, without the need of spending too many computational resources.

Nevertheless, the simulation of turbulent flow around an airfoil at low Reynolds number is gaining interest in the recent years. This increase on the demand of the simulation of such flows is caused by the fact that aircraft-like devices have been proposed as possible Mars explorers, see [123, 167, 3]. These vehicles are being designed to take photographs of Mars while traveling at low-speed in a thin atmosphere, resulting in a low Reynolds number ($10^3 \leq Re \leq 10^5$) environment.

In this type of problems the mesh design plays an important role on the solution, in particular, a proper spacing has to be used at the boundary layer. If a strong definition of the Dirichlet boundary conditions is considered, we say that the simulation is a *Wall-Resolved LES* (WRLES). According to [149], the coherent structures that appear in the turbulent boundary layer can be captured with a WRLES method if the near-wall node of the mesh is located at a wall distance $y^+ < 2$ and the stream-wise cell size is within the range $50 < \Delta x^+ < 150$. Otherwise, when a wall-law model is used considering weak imposition of the Dirichlet boundary conditions, we say that the simulation is a *Wall-Modeled LES* (WMLES).

In this chapter we will contemplate the SVMS approach, described in Chapter 7, considering the weak imposition of Dirichlet boundary conditions. Our concern is to check the feasibility of the proposed method as a WMLES method for the simulation of turbulent flows around an aircraft. In such simulations many flow characteristics are stressed, dealing not simply with turbulent flows, but also with other difficulties like the presence of very thin laminar boundary layers, transition from laminar to turbulent regimes, wall-bounded flow or flow that separates from curved surfaces. The fact is that LES methods have been shown to be very useful for flows where the turbulent structure is dominated by the large-scale structures, like the TGV test presented in Section 3.2.3 or homogeneous turbulent problems (see [56]). Despite that, the simulation of near-wall flows using LES methods has been demonstrated to be a challenging task, manly because of the complex physical phenomena like the reduction of the large scales structures near the wall, the flow anisotropy or the mesh sensitivity to high aspect ratios.

Many works have been done discussing the suitability of LES methods for the simulation of flow around an airfoil. This is the case of the European project LESFOIL [61], where nine different academic and industrial groups worked with the aim to identify the potential of LES as a prediction method for separation in high-Reynolds-number airfoil flows. In this case, the simulated geometry was the Aerospatiale-A airfoil operating at a chord Reynolds number equal to $2.1 \cdot 10^6$ with an angle of attack equal to 13.3^o and a Mach number of 0.15. Other authors have been worked with NACA type airfoils. This is the case of [107, 116, 164], where the use of LES method for the simulation of the flow around a NACA4412 airfoil is studied using unstructured, structured and semi-structured grids, respectively.

This chapter is structured as follows. First we define the test setting in Section 8.2, followed by the numerical results exposed in Section 8.3. Finally, some conclusions are pointed out in Section 8.4.

8.2 Test setting

The formulation used for the NACA simulation is the one defined in Chapter 7. We consider the use of a mixed FE formulation with convection stabilization through an

OSS method. Here we also use weak Dirichlet boundary conditions imposition for both the tangential and normal velocity components, detailed in Section 7.2.2. A SRK scheme is used to integrate the equations in time, as described in Section 7.4.

In this test we study the flow around a NACA 0012 airfoil with a span-wise length of $0.16c$, being c the chord length. In Figure 8.1 we depict the geometry of the studied airfoil. The computational domain is defined with an inlet and outlet boundaries around 10 chord lengths away from the airfoil surface. At the inlet boundary a free-stream velocity $U_\infty = 10$ has been settled, while the outlet boundary has left free. We consider an angle of attack of $\alpha = 6$ deg and a Reynolds number based on the chord length $Re_c = 2.3 \cdot 10^4$, the same used in [123, 119].

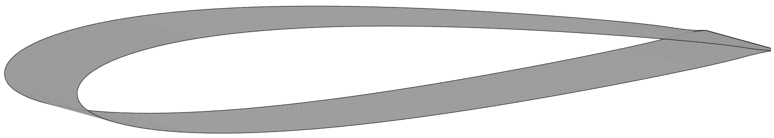
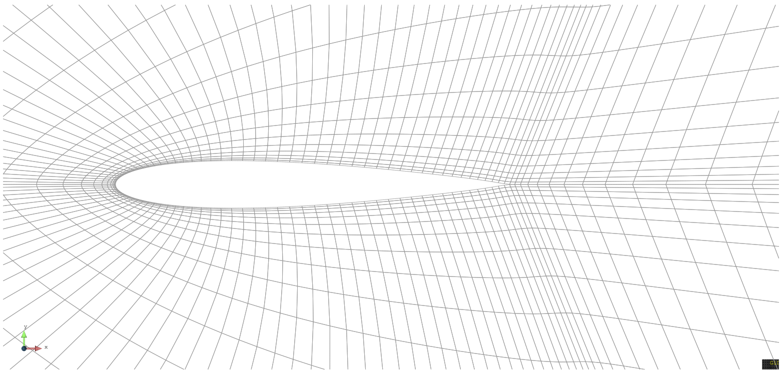


FIGURE 8.1: NACA 0012 airfoil geometry.

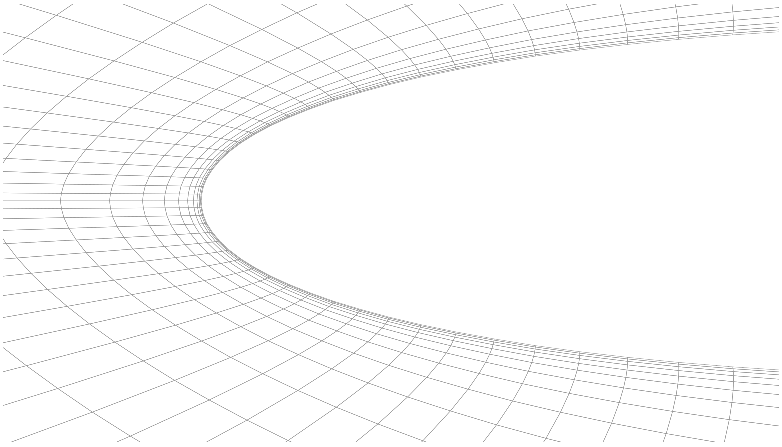
The simulation of this test is done on a structured C-type mesh around the airfoil profile. We consider two different 2-dimensional meshes of quadrilaterals, one much finer than the other, which are extruded along the span-wise direction. The goal of using two different meshes is to assess the performance of the weak imposition of the Dirichlet boundary conditions at the airfoil surface. We define the meshes in a such way that we have fine enough elements around the airfoil surface, the turbulent boundary layer and the wake region, while coarse elements are used in the far field region. In Figure 8.2 we depict the two meshes used in this test. Focusing on the finer mesh, shown in Figure 8.2(b), at the leading edge, the near wall-node is located at $y \sim 2.0 \cdot 10^{-4}c$ which leads to a wall distance of $y^+ < 1$. This distance is kept almost constant at the laminar region and it is increased constantly until it reaches the maximum of $y \sim 2.0 \cdot 10^{-3}$ at the trailing edge, where the wall distance is less restrictive. The maximum stream-wise elemental length is $\Delta x \sim 0.028c$ located at the suction side of the airfoil, giving a normalized distance of $\Delta x^+ \sim 40$. We see that the mesh sizes satisfy the conditions needed to capture the boundary layer phenomena.

Regarding the coarse mesh, see Figure 8.2(a) and Figure 8.2(c), the near wall-node is located at $y \sim 1.2 \cdot 10^{-3}c$ leading to a wall distance of the order of $y^+ \sim 2$. This distance is constantly increased until it reaches a value of $y \sim 6.4 \cdot 10^{-3}c$ at the trailing edge, much grater than the finer mesh. In this mesh, the maximum stream-wise elemental length is $\Delta x \sim 0.032c$ that is equivalent to a normalized distance of $\Delta x^+ \sim 50$. We only compute the 3D test on the coarse mesh, with a constant span-wise elemental length equal to $\Delta z \sim 0.02c$.

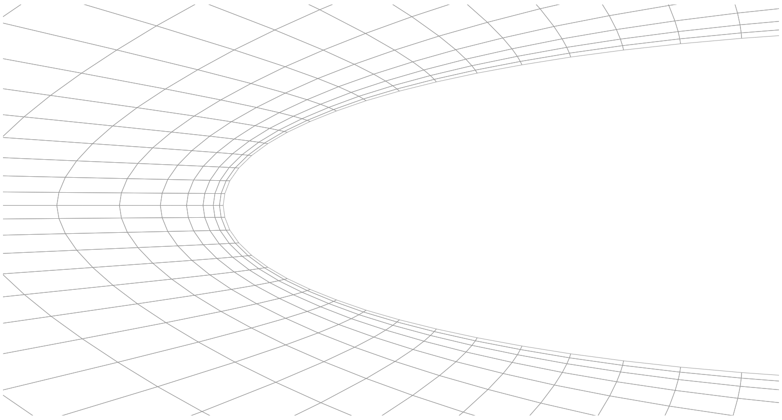
The spatial discretization is done using inf-sup stable elements Q_2/Q_1. These type of elements will allow us to use the SVMS approach described in the previous Chapter



(a) Coarse mesh.



(b) Close up view of the fine mesh.



(c) Close up view of the coarse mesh.

FIGURE 8.2: NACA 0012 meshes.

7. Here we use the same algorithmic parameters, $c_1 = 12.0$ and $c_2 = 8.0$, as used in Section 5.5.3. The problem is solved using the IMEX version of the SRK method introduced in Chapter 6 and also used in Chapter 7 with a *(3-3)* scheme, see Appendix C, and an adaptive time stepping technique as exposed in Section 7.4.4. The time step size evolves tending to $\delta t = 5.0 \cdot 10^{-4} c/U_\infty$ giving a maximum hyperbolic CFL around 0.3.

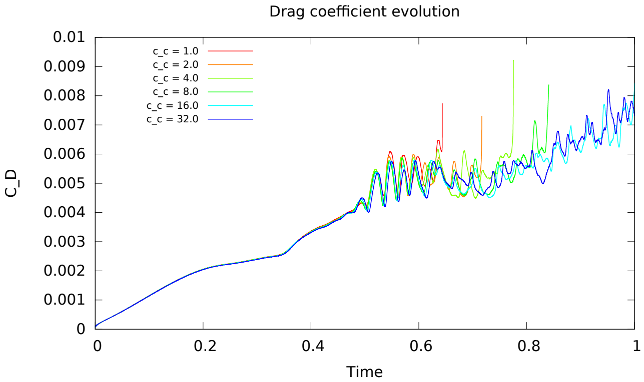
8.3 Simulation results and discussion

In order to check the suitability of the proposed methods we start with a 2-dimensional simulation comparing the results obtained with the fine and coarse meshes. For such computations, we begin with an initial solution that has been computed solving the Stokes problem. We solve the problem from $t = 0.0$ to $t = 1.0$ which is equivalent to 10 time units (1 time unit $= c/U_\infty$).

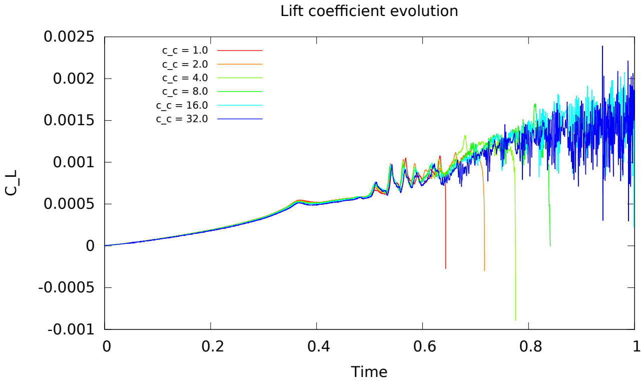
8.3.1 Effect of c_c in a 2D mesh with strong boundary conditions

We first analyze the effect of the c_c constant that appear in equation (7.29). We know by the experience acquired on the turbulent tests performed in Chapter 5 and Chapter 7 that this parameter plays a relevant role when inf-sup stable elements are used. Here we choose the set of values $c_c = \{1.0, 2.0, 4.0, 8.0, 16.0, 32.0\}$ and solve the problem from $t = 0.0$ to $t = 1.0$ using the finer mesh with strong boundary conditions.

We depict the Drag and Lift coefficients in Figure 8.3, computed as stated in Section 6.4.3. It is seen in both the drag (Figure 8.3(a)) and lift (Figure 8.3(b)) coefficients that at some point, the computations for $c_c < 32.0$ blows up. Let us now focus on the



(a) Drag coefficient.



(b) Lift coefficient.

FIGURE 8.3: Drag and Lift coefficients for the 2D computation for different c_c values.

evolution of the velocity divergence norm, which is shown in Figure 8.4 scaled by the kinetic energy of the problem. It is seen on this figure that, for $c_c < 32.0$, the velocity divergence norm is increased with a constant rate until it reaches an unstable point from which it explodes. This behavior does not occur for $c_c = 32.0$ until $t = 1.0$. We also see that the rate of increase of the velocity divergence tends to the one followed by the $c_c = 32.0$ case. This constant increase shown in Figure 8.4 is normal and may be caused by the flow development from a laminar initial solution to a fully developed turbulent flow. In the forthcoming computations of this test we will use $c_c = 32.0$.

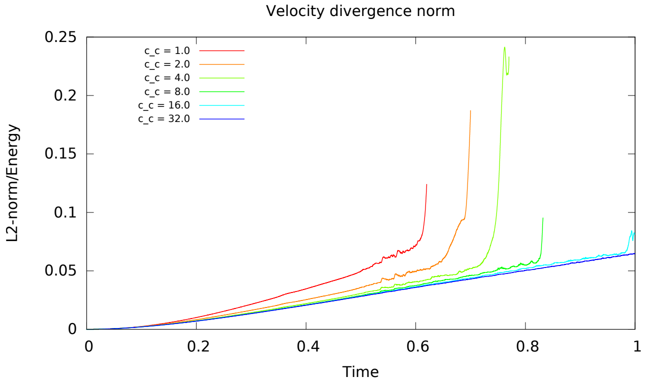
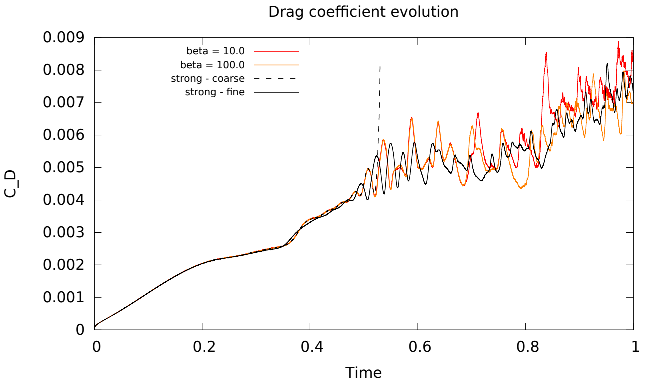


FIGURE 8.4: $\|\nabla \cdot \mathbf{u}\|/K$ for the 2D computation for different c_c values.

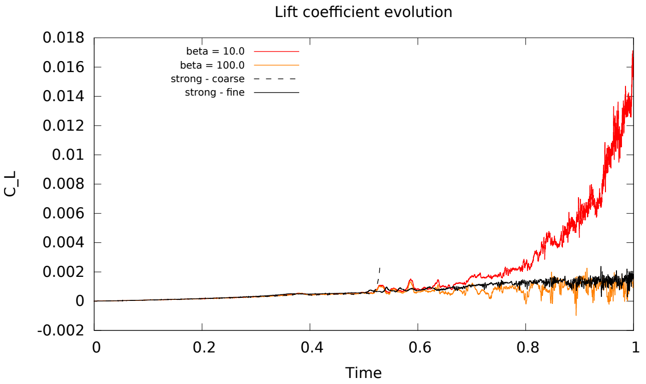
8.3.2 Effect of β in a 2D mesh with weak boundary conditions

Once analyzed the effect of the constant c_c on the solution, we proceed to check the effect of the β constant that appear on the definition of the weak imposition of the wall-normal component on the Dirichlet boundary. In this case we select four different values, $\beta = 10, 100$ and we solve the problem on the coarse mesh from $t = 0.0$ to $t = 1.0$. The case in which $\beta = 1000$ have been discarded since the computational cost is much greater than the other two.

In this analysis, as it is seen in Figure 8.5, very little differences can be distinguished during the laminar transition to turbulent flow for the different values of β. Furthermore, if we compare against the solution obtained with strong imposition of the boundary conditions using the fine mesh, we see that the results are very similar for the laminar transition. Things change when the flow starts evolving to turbulent behaviour, around $t = 0.5$, where we see that the differences between the different parameters selection begin to increase. The solution with strong boundary conditions in a coarse mesh explodes when the turbulence starts developing. The cases in which we consider a weak boundary imposition keep being stable, with a drag coefficient similar the one obtained with strong boundary conditions in a finner mesh, see Figure 8.5(a). However, if we look at Figure 8.5(b), we see that for $\beta = 10$ the lift coefficient starts increasing more than the reference solution, while for $\beta = 100$ this coefficient remains of the same order than the strong boundary conditions case. Given these results, in



(a) Drag coefficient.



(b) Lift coefficient.

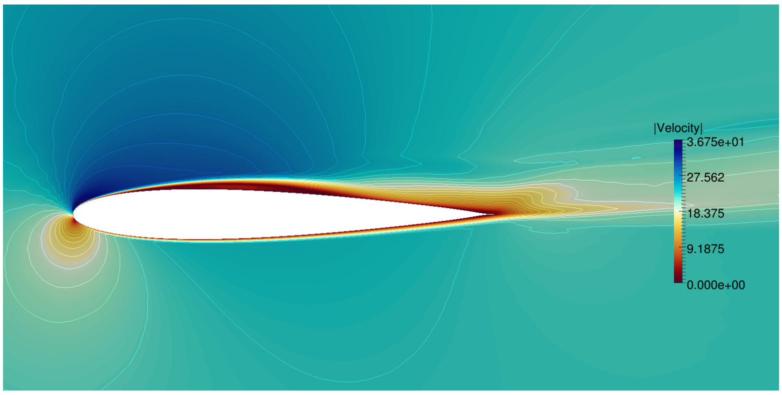
FIGURE 8.5: Drag and Lift coefficients for the 2D computation for different β values.

coming computations we will favour $\beta = 100$.

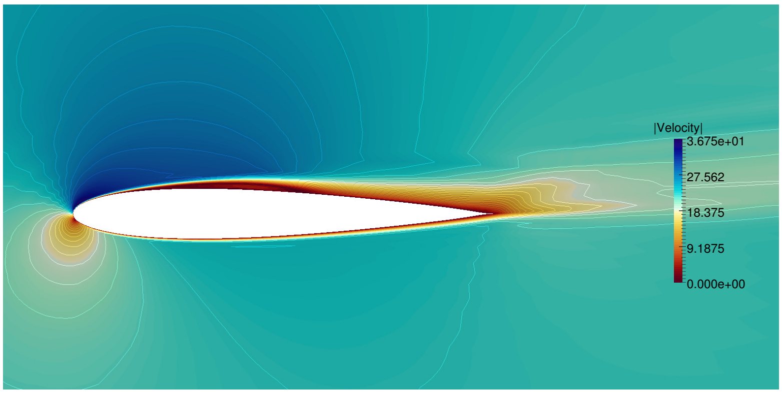
8.3.3 Strong versus weak boundary conditions

In order to see the difference on the computed solution when using strong or weak Dirichlet boundary conditions, we show the mean flow for both, fine and coarse meshes, respectively, averaging the results obtained between $t = 0.8$ and $t = 1.0$.

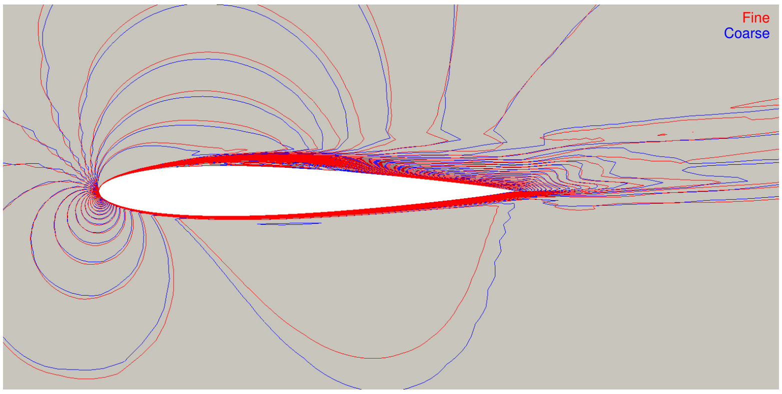
In Figure 8.6 we depict the mean velocity magnitude for the case in which we impose strongly the boundary conditions in a fine mesh (Figure 8.6(a)) and the case in which we impose weakly the boundary conditions in a coarse mesh (Figure 8.6(b)). We see the same velocity distribution in both cases, with slight changes between them. In order to better see the difference between both solutions, in Figure 8.6(c) we depict the mean velocity isolines for the two cases. We see that the solution is very similar for the two computations, conserving the same structures. Focusing on the velocity fluctuations, in Figure 8.7 we depict the standard deviation of the velocity magnitude computed from $t = 0.8$ to $t = 1.0$. In this case, we also see very little differences, noting that the fluctuating regions are located at the same positions in both cases.



(a) Fine mesh, strong boundary conditions.



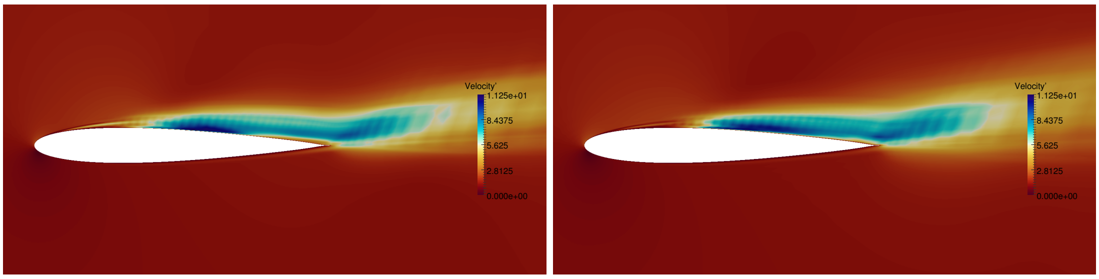
(b) Coarse mesh, weak boundary conditions.



(c) Velocity magnitude isolines.

FIGURE 8.6: Mean velocity magnitude for the fine and coarse meshes.

Looking now at the mean pressure plots depicted in Figure 8.8 we can observe the same similarities between the two computations. In Figure 8.8(a) the fine mesh with strong boundary conditions solution is plotted, while the solution in the coarse mesh and weak boundary conditions is shown in Figure 8.8(b). The little differences between the two results can be distinguished in Figure 8.8(c), where the pressure isolines are depicted.



(a) Fine mesh, strong boundary conditions. (b) Coarse mesh, weak boundary conditions.

FIGURE 8.7: Standard deviation of the velocity magnitude for the fine and coarse meshes.

8.3.4 Instantaneous flow fields for the 3D case

Once analyzed the effect of using weak Dirichlet boundary conditions on a two-dimensional mesh, we check the performance for the 3D case. Here we use as an initial solution the extruded solution of the 2D case at $t = 1.0$. Then, we let the flow evolve until it reach a 3D description, which occurs after about one time unit.

In this section we will analyze the instantaneous flow fields, trying to identify the turbulent structures that are generated along the wing.

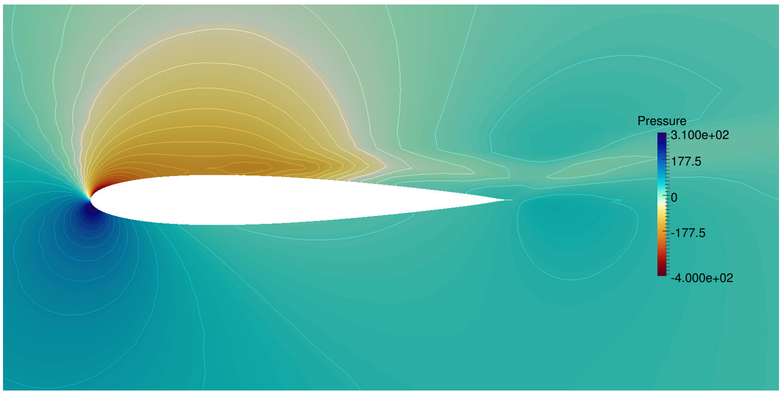
In Figure 8.9 we can see the velocity isosurfaces for $\|\mathbf{u}\| = 25.0$ colored by the pressure value. In this figure we can clearly see the flow transition from a laminar regime at the leading edge to a turbulent flow, starting the 3D flow development near the maximum wing thickness point.

To study the structures that arise in the turbulent flow we use the Q-criterion defined as $Q = \frac{1}{2}(|\omega|^2 - |\varepsilon(\mathbf{u})|^2)$, being ω the vorticity and ε the strain rate tensor. The isosurface of $Q = 5 \cdot 10^5$ is depicted in Figure 8.10, where we can see the generation of 2D coherent vortices that break up near the maximum thickness point, turning into hairpin vortices. Similar results are obtained in [123], where a much finer mesh is used to solve the same problem. In Figure 8.11 we present a top view of the $Q = 5 \cdot 10^5$ isosurface. There, we can see the position of the first 2D coherent vortex, located at $x/c = 0.37$. This result is similar to the obtained in [123], where the first 2D vortex is located at $x/c = 0.41$.

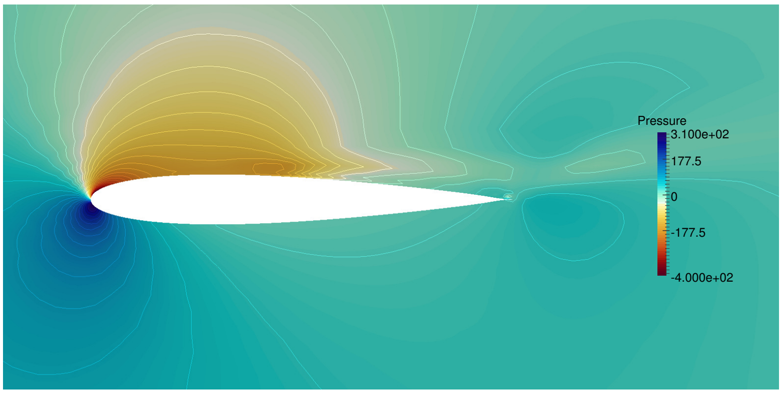
8.4 Conclusions

In this chapter we have tested the behavior of the SVMS method with weak boundary conditions based on a wall-law model for the simulation of turbulent flows around an airfoil.

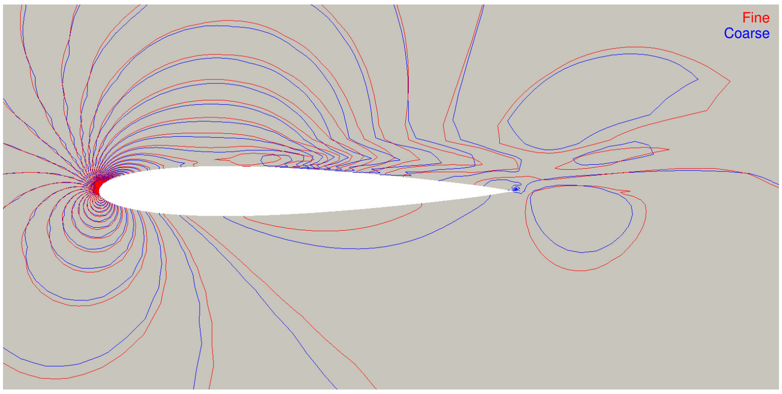
Considering weak boundary conditions when coarse meshes are used have been demonstrated to be necessary when the mesh size at the wall does not allow to capture properly the boundary layer. Furthermore, the weak imposition of the wall-normal



(a) Fine mesh, strong boundary conditions.



(b) Coarse mesh, weak boundary conditions.



(c) Velocity magnitude isolines.

FIGURE 8.8: Mean pressure for the fine and coarse meshes.

component have been successfully used in this test, avoiding the complex techniques that are required for the strong imposition of only the normal component on curvilinean geometries.

The effect of the algorithmic constants c_c and β have been assessed for this test, concluding that, as noted in previous turbulent tests, the c_c constant has a clear effect on the solution. This phenomena have been studied in previous chapters, where the

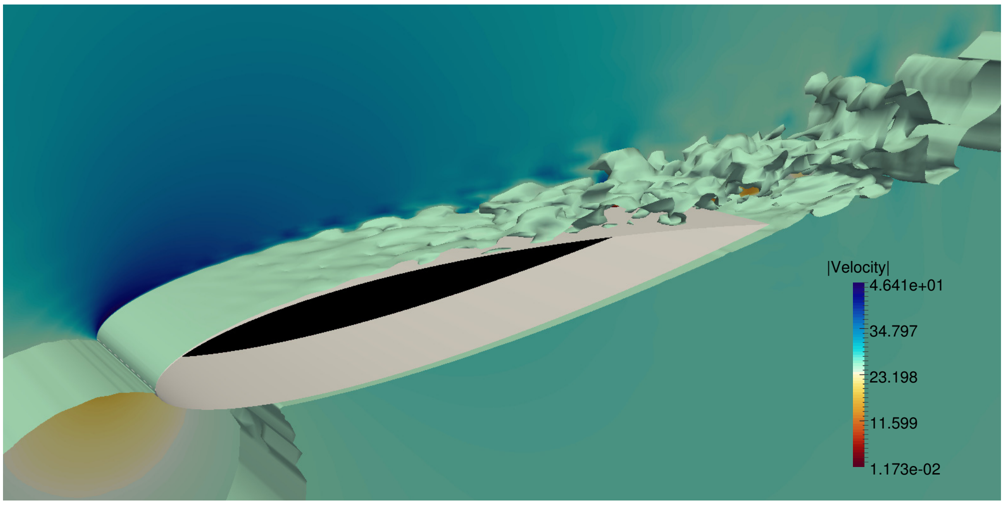


FIGURE 8.9: Velocity isosurface for $\|\mathbf{u}\| = 25.0$.

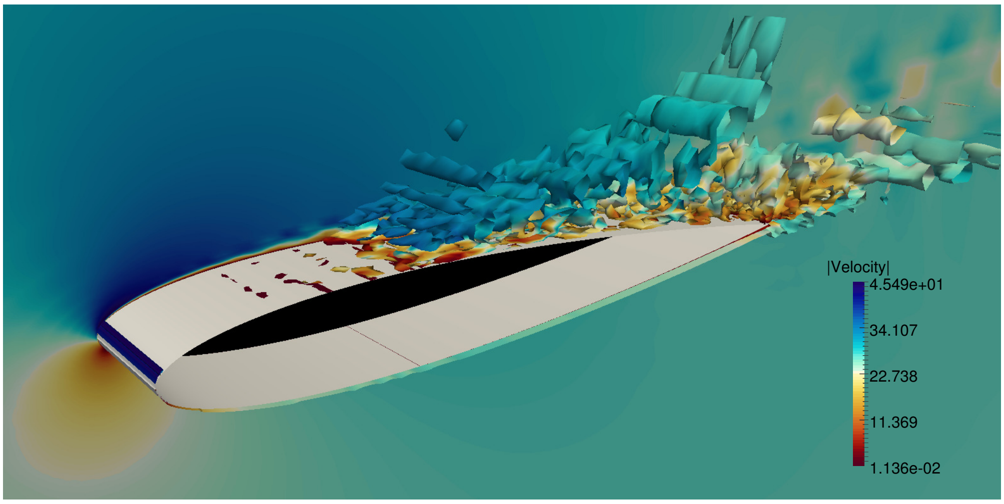


FIGURE 8.10: Q-criterion isosurface for $Q = 5 \cdot 10^5$ and colored by velocity magnitude.

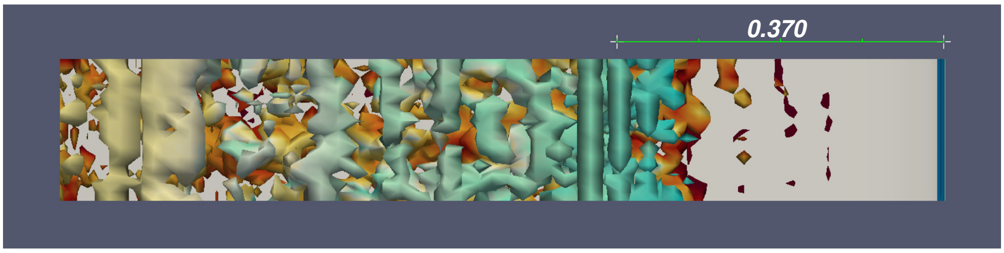


FIGURE 8.11: Top view of Q-criterion isosurface for $Q = 5 \cdot 10^5$.

dependency of the solution on the c_c value have been related to the fact that the grad-div stabilization term is highly important when using inf-sup stable elements.

Further computations need to be carried out to study with a higher degree of detail the problem solved in this chapter. Particularly for the 3D case, where the aerodynamic coefficients like the pressure coefficient, drag coefficient and lift coefficient should be addresses and compared with the existing results in the literature. Moreover, it also has to be studied the effect of the mesh resolution in such computations, since a very coarse mesh has been used here.

Chapter 9

Conclusions and future work

9.1 Conclusions

The development of new algorithms based on the FE method for the simulation of large scale turbulent incompressible flows has been studied in this thesis.

In order to take advantage of the continuously increasing computational power acquired with the new improvements on super-computers, an advance in software design is imperative. New algorithms need to be designed to be used in the exascale computing environment. In this thesis we have proposed a framework for the High Performance Computing of turbulent incompressible flows that relies on the extreme scalability of Balancing Domain Decomposition by Constraints preconditioners and the use of Variational Multiscale stabilization methods as Large Eddy Simulation models. Furthermore, a novel time integration scheme that segregates velocity and pressure computations through Runge-Kutta schemes has been developed.

The thesis has sought to address the main objectives that we have defined in the introduction chapter, whose fulfillment is discussed in what follows.

- **RB-VMS methods as LES models for turbulent incompressible flows**
  The applicability of RB-VMS methods as LES models for the simulation of turbulent incompressible flows has been demonstrated in Chapter 4. In that chapter, some theoretical aspects have been discussed, such as the dissipative structure of the methods and the way energy is conserved, which we have numerically verified.

  The most important conclusion that we can point out from the results obtained in Chapter 4 is that OSS and ASGS methods yield similar results, all displaying the features of turbulent flows, reproducing appropriately global outputs such as energy spectra. Moreover, the methods are stable and converge to reference solutions, both when the mesh is refined and when the polynomial order is increased.

- **Mixed FE formulations for LES of turbulent incompressible flows**
  With respect to this objective, in Chapter 5 the comparison of three methods, ASGS, term-by-term OSS, and convection-only OSS with ISS elements have been tested. It has been shown that the accuracy is similar for the same order of interpolation of the velocity for all methods, the OSS-ISS being slightly inferior in this

respect. But on the other hand, when computational cost is analyzed the OSS-ISS is clearly the cheapest one so a finer discretization can be used for a given computational cost.

- **High-order FE methods**
  The use of high-order FE methods, up to third order, have been considered in the core chapters of the thesis. Besides, all the proposed algorithms can be used with arbitrary order of interpolation, making them extensible for orders greater than three.

- **High-order time integration methods**
  One of the main concerns of this work has been the use of high-order schemes for the time integration. The use of such methods has motivated the Segregated Runge-Kutta algorithm proposed in Chapter 6 and later also used in Chapter 7 and Chapter 8. The proposed SRK method has been used with orders of convergence in time up to three, for both velocity and pressure fields. Nevertheless, the SRK scheme is not restricted to third order since the order only depends on the definition of the Butcher tableaus.

- **Adaptive time integration schemes**
  Related to the previous point, the Segregated Runge-Kutta methods also allows the easy implementation of adaptive time-stepping techniques. These techniques has been successfully analyzed in Chapter 6 and effectively used for the simulation of turbulent flows in the tests done in Chapter 7 and Chapter 8.

- **Segregation of velocity and pressure fields**
  The segregated Runge-Kutta methods proposed in this work allows the velocity and pressure segregation at the time integration level (without the need to perform additional fractional step techniques that spoil high orders of accuracy). The use of such schemes is very appealing for large scale computations of incompressible flows, since the monolithic indefinite system is replaced by segregated positive-definite velocity and pressure blocks. The pressure block involves a Poisson solver, whereas the velocity block is a vector-Laplacian or elasticity matrix when the convective term is treated explicitly.

- **Large scale and scalable FE solvers**
  The use of massively parallel and scalable solvers for the problems that arise after the segregation introduced by the SRK scheme, have been considered in Chapter 7. This approach also enables the use of block preconditioning techniques that lead to the approximation of Laplacian-type and elasticity-type problems, suitable to be preconditioned with BDDC algorithms. The weak scalability of this approach for the resolution of one time step of the TGV test have been demonstrated up to 8000 cores.

- **Application**
  In Chapter 8 the methods proposed in Chapter 7 have been used to assess their performance when simulating turbulent flows around an airfoil, in particular a NACA 0012 profile. This is a widely used benchmark in the aerospace industry since many airplane and wind turbines wings are based on these kind of profiles.

Although all the objectives stated at the introduction have been addressed in this work, there are some limitations that worth pointing out.

One of the weakness of the proposed VMS methods is the high dependency on the algorithmic constants that appear on the formulation, especially when coarse meshes are used. Albeit the constant election is not critical when laminar flows are simulated, their influence for turbulent incompressible flows have been revealed in all tests performed in the thesis.

Furthermore, it has been shown that the best configuration of the algorithmic constants is problem dependent. Meaning that in order to obtain the best results for a given turbulent problem, e.g. isotropic turbulence, wall-bounded flow, etc., the setting of the algorithmic constants change. Even for different mesh configurations, i.e. stretched meshes versus uniform meshes, the influence of the constant parameters is notorious.

However, it is important to stress that this high algorithmic parameter influence is important when very coarse meshes are used, being less important for fine meshes.

9.2 Open lines of research

Some lines of research could be considered to further investigate the applicability of the proposed methods, as well as to deal with the limitations inherent to them. These open lines of research are described below:

- **Embedded methods and mesh refinement**
  One of the main problems we have faced during the dissertation is the need of defining good meshes a priory. Then, the development of adaptive meshes able to refine where the turbulent structures are being developed is one of the lines of research that could resolve this issue.

  The approach that could be followed is to consider the use of Cartesian meshes with embedded boundaries. This approach basically consist on having most of the domain covered by completely regular Cartesian cells and considering embedded boundary methods to impose the boundary conditions. This methodology could also be related to the work done in this thesis when weak boundary conditions are imposed considering a wall-law model.

- **Parallel in time**
  The turbulent phenomena is characterized by having not only many spatial scales,

but also a multiscale description in time. Then, the time discretization of Navier-Stokes equations at high Reynolds number becomes an important issue. Many efforts are being dedicated to the parallelization of the time discretization schemes. It could be interesting to seek the applicability of parallel time integration techniques when using SRK schemes for the time integration of the Navier-Stokes equations, with a particular interest on the simulation of turbulent flows.

- **Divergence-free Finite Elements**
  It has been shown in this work that when using inf-sup stable elements, the grad-div stabilization term plays an important role to ensure the satisfaction of the incompressibility constraint. In order to address this weakness, we could consider to make use of FE types that are divergence-free, and inf-sup stable, by construction.

Appendix A

Energy spectrum computation

In this appendix we will briefly resume the code that has to be implemented to obtain the energy spectra of a turbulent fluid sample.

We assume that we are working with a two-dimensional fluid flow with periodic boundary conditions on both directions. We also assume that we compute the energy spectrum from the velocity field, coming from the Navier-Stoke equations. There also are many works on the turbulence world that use the vorticity field to calculate the energy spectra.

As we are using a Finite Element Method to calculate the velocity field, our data will be a discrete data with the values of the velocity on each node of the FEM mesh. The first thing we need to do is to store the velocity field into an array with the same components as problem dimensions. Usually, the velocity field that arise after the solution of the FE system of equations is stored in a vector. Then, we must transform the velocity field into two-dimensional array, having a matrix with the same rows as nodes in the vertical direction and the same columns as nodes in the horizontal direction. The matrix values must be stored in the equivalent position than the node position in the FEM mesh.

It also has to be said that for a problem with periodic boundary conditions we only have to transform those values that are not repeated. For a two-dimensional grid with $N_x \times N_y$ divisions ($(N_x + 1) \times (N_y + 1)$ nodes), the values to be transformed are those corresponding to the nodes 1 to N_x, in the x-direction, and 1 to N_y, in the y-direction. The $(N_x + 1)$-th and $(N_y + 1)$-th nodes have the same value as the first ones.

Once we have obtained the velocity field and we have transformed into a two dimensional array, we follow the following steps.

1. *FFT transform of the velocity field*

   At this point we perform the Discrete Fast Fourier Transform (DFFT) of each component of the velocity field. To do that we use the *fft99* and *cfft99* packages that we can find in [148]. The first code transforms a real data vector to a half-complex vector with the transformed field. The second one transforms a complex vector to another complex vector with the transformed field.

Both packages are based on the following Discrete Fast Fourier Transform definitions:

$$f_k = \sum_{n=0}^{N-1} f_n e^{-i\frac{2\pi kn}{N}}, \qquad \text{(Forward transform)}$$

$$f_n = \sum_{k=0}^{N-1} f_k e^{i\frac{2\pi kn}{N}}, \qquad \text{(Backward transform)}$$

with k the wave number, N the number of values to transform, f_k the k-th value of the transformed vector and f_n the n-th value of the original vector.

We also must have into account that to use these packages the lengths of the transforms must be an even number greater than 4 that has no other prime factors than 2, 3 and 5.

Since we have a two-dimensional real data field, and using the Fourier Transforms theory, we compute the two-dimensional transform by first transforming a set of vectors using *fft99* along one direction and then transforming the results using *cfft99* along the orthogonal direction.

By this way, if we first transform the horizontal direction, the real to complex transformation give us a half-complex matrix with dimension $\left(N_x \times \left(\frac{N_y}{2}+1\right)\right)$. It means that the matrix is built by N_x vectors each containing the Fourier coefficients $a(k_y)$ and $b(k_y)$ for $0 \leq k_y \leq \frac{N_y}{2}$, defining the Fourier transform as

$$c(k_y) = a(k_y) + ib(k_y).$$

Note that with this definitions we should have a matrix of dimension $\left(N_x \times \left(\frac{N_y}{2}+2\right)\right)$. But the fact that the input values are real implies that $b(0) = b(N_y/2) = 0$, then we can get rid of one value for each row.

After that we perform the transform along the vertical direction. As we already have a complex matrix, that is to compute the transforms of $\left(\frac{N_y}{2}+1\right)$ complex vectors with length N_x.

The result is a $\left(N_x \times \left(\frac{N_y}{2}+1\right)\right)$ complex matrix with the transformed values.

2. *Wave number definition*

The next step is to set the wave numbers. The amount of wave numbers that we have in each direction is equal to the corresponding size of the original matrix, then, the Nyquist wave number will be $\frac{N_x}{2}$ and $\frac{N_y}{2}$, respectively. It means that wave numbers higher than the Nyquist one will encounter a "symmetric" (with respect to the Nyquist wave number), back into lower wave numbers. Hence, we define the wave number vectors as follows.

$$\vec{k}_x = \left[0, 1, ..., \frac{N_x}{2}, -\frac{N_x}{2}+1, -\frac{N_x}{2}+2, ..., -1\right],$$

$$\vec{k}_y = \left[0, 1, ..., \frac{N_y}{2}\right].$$

3. *Energy spectrum calculation*

   Finally, the third step is to compute the energy spectrum. Here, the point is to represent the energy evolution from a two-dimensional field into a one-dimensional. So we need to group the two-dimensional grid values into some strips.

   To do that we first define the one-dimensional wave number vector and the maximum one-dimensional wave number as follows.

$$k(l) = integer\left(\sqrt{\vec{k}_x(i)^2 + \vec{k}_y(j)^2} + 0.5\right), \qquad \begin{cases} i = 1, ..., N_x, \\ j = 1, ..., \frac{N_y}{2} + 1, \\ l = 0, ..., k_{max}, \end{cases}$$

$$k_{max} = \max\left\{integer\left(\sqrt{k_x^2 + k_y^2} + 0.5\right)\right\}.$$

   It is equivalent to say that each one-dimensional wave number $k(l)$ will be composed for all k_x and k_y such that the value of $\sqrt{k_x^2 + k_y^2}$ is in the interval $[k - 0.5, k + 0.5]$.

   To compute the energy spectra we only use those wave numbers between 0 and $\frac{k_{max}}{\sqrt{2}}$. Suppose that we have a squared domain with the same number of nodes in both direction, that is $N_x = N_y = N$. Then the maximum wave number will be:

$$\begin{aligned} k_{max} &= int\left(\sqrt{\left(\frac{N_x}{2}\right)^2 + \left(\frac{N_y}{2}\right)^2} + 0.5\right) = int\left(\sqrt{2\left(\frac{N}{2}\right)^2} + 0.5\right) \\ &= int\left(\sqrt{2}\left(\frac{N}{2}\right) + 0.5\right). \end{aligned}$$

   We can see that the contributions from k_x and k_y will increase until the wave number k reaches $\frac{N}{2}$. After that the number of two-dimensional wave numbers that contributes to the one-dimensional wave number decreases. For this reason, the realistic maximum wave number must be $integer\left(\frac{k_{max}}{\sqrt{2}}\right)$.

   If we define the interval $I_k = [k - 0.5, k + 0.5]$ for each k between 0 and $\frac{k_{max}}{\sqrt{2}}$, we can construct the energy spectra, $E(l)$, as follows.

   For each $i = 1, ..., N_x$ and for each $j = 1, ..., \frac{N_y}{2} + 1$, $|k| = \sqrt{\vec{k}_x(i)^2 + \vec{k}_y(j)^2}$ and

$$\begin{aligned} E(k) = \sum_{i,j \backslash |k| \in I_k} \frac{1}{2} &\left[\hat{u}_x(k_x(i), k_y(j)) \cdot \hat{u}_x^*(k_x(i), k_y(j))\right. \\ &\left. + \hat{u}_y(k_x(i), k_y(j)) \cdot \hat{u}_y^*(k_x(i), k_y(j))\right], \end{aligned}$$

   where $\hat{u}_i$ is the transformed i-direction velocity field and $\hat{u}_i^*$ its complex conjugate.

Algorithm

In the following table the algorithm described above to compute the energy spectrum of a turbulent sample in a two-dimensional structured mesh is resumed.

Algorithm 2: Algorithm to compute the energy spectrum.

```
1. Transform the velocity vector into a matrix corresponding to the mesh points:
       $u_x = u_x(1:N_x, 1:N_y)$, $u_y = u_y(1:N_x, 1:N_y)$.
2. Compute the FFT of the velocity fields: $\hat{u}_x = fft(u_x)$, $\hat{u}_y = fft(u_y)$.
3. Define the wave number vectors:
       $\vec{k}_x = \left[0, 1, ..., \frac{N_x}{2}, -\frac{N_x}{2}+1, -\frac{N_x}{2}+2, ..., -1\right]$, $\vec{k}_y = \left[0, 1, ..., \frac{N_y}{2}\right]$.
4. Calculate the maximum one-dimensional wave number $k_{max}$.
5. Compute the one-dimensional wave number vector $k$ and the energy spectra
   $E(k)$.
```

Appendix B

VMS methods implementation

In this appendix we give some tips and some comments on how the VMS methods defined in Chapter 4 are implemented in a FE software.

We will describe the algorithms used to implement the dynamic and nonlinear versions for both the ASGS and the OSS methods. The quasi-static alternatives and the linear subscales definition can be though as a particular case of the former version.

Algebraic Subgrid Scale (ASGS)

The Algebraic Subgrid Scale (ASGS) method is characterized by the projection $\mathcal{P}$ appearing in the right-hand side of (2.39) and (2.40) defined as

$$\mathcal{P} := \mathbf{I} \tag{B.1}$$

Then, the equivalent Navier-Stokes problem expressions are given by

$$\begin{aligned}(\partial_t \mathbf{u}_h, \mathbf{v}_h) + (\partial_t \tilde{\mathbf{u}}, \mathbf{v}_h) + B(\mathbf{a}; [\mathbf{u}_h, p_h], [\mathbf{v}_h, q_h]) \\ + (\tilde{\mathbf{u}}, \mathcal{L}^*_{\mathbf{a}}(\mathbf{v}_h, q_h))_h - (\tilde{p}, \nabla \cdot \mathbf{v}_h) = \langle \mathbf{f}, \mathbf{v}_h \rangle \,,\end{aligned} \tag{B.2}$$

together with the subscales equations

$$\partial_t \tilde{\mathbf{u}} + \tau_m^{-1} \tilde{\mathbf{u}} = \mathbf{R}_u, \tag{B.3}$$

$$\tau_c^{-1} \tilde{p} = R_p. \tag{B.4}$$

The previous expressions are solved using the Picard method for nonlinear equations. Once discretized in time using a Backward Euler discretization, we have that at the step $n+1$ and each nonlinear iteration i, the values of $\mathbf{u}_h^{n+1,i}$ and $p_h^{n+1,i}$ are calculated solving the following decomposed equations

$$\begin{aligned}
&\frac{1}{\delta t}(\mathbf{u}_h^{n+1,i},\mathbf{v}_h)+(\mathbf{a}^i\cdot\nabla\mathbf{u}_h^{n+1,i},\mathbf{v}_h)+\nu(\nabla\mathbf{u}_h^{n+1,i},\nabla\mathbf{v}_h)-(p_h^{n+1,i},\nabla\cdot\mathbf{v}_h)\\
&+(q_h,\nabla\cdot\mathbf{u}_h^{n+1,i})-\tau_{m,t}(\mathbf{a}^i\cdot\nabla\mathbf{u}_h^{n+1,i}-\nu\Delta\mathbf{u}_h^{n+1,i}+\nabla p_h^{n+1,i},-\mathbf{a}^i\cdot\nabla\mathbf{v}_h-\nu\Delta\mathbf{v}_h-\nabla q_h)\\
&+\tau_c(\nabla\cdot\mathbf{u}_h^{n+1,i},\nabla\cdot\mathbf{v}_h)-\tau_{m,t}\frac{1}{\delta t}(\mathbf{a}^i\cdot\nabla\mathbf{u}_h^{n+1,i}-\nu\Delta\mathbf{u}_h^{n+1,i}+\nabla p_h^{n+1,i},\mathbf{v}_h)\\
&-\tau_{m,t}\frac{1}{\delta t}(\mathbf{u}_h^{n+1,i},-\mathbf{a}^i\cdot\nabla\mathbf{v}_h-\nu\Delta\mathbf{v}_h-\nabla q_h)-\tau_{m,t}\frac{1}{\delta t^2}(\mathbf{u}_h^{n+1,i},\mathbf{v}_h)\\
&=\langle\mathbf{f},\mathbf{v}_h\rangle+\frac{1}{\delta t}(\mathbf{u}_h^n,\mathbf{v}_h)-\tau_{m,t}(\mathbf{f},-\mathbf{a}^i\cdot\nabla\mathbf{v}_h-\nu\Delta\mathbf{v}_h-\nabla q_h)-\tau_{m,t}\frac{1}{\delta t}(\mathbf{f},\mathbf{v}_h)\\
&-\tau_{m,t}\frac{1}{\delta t}(\mathbf{u}_h^n,-\mathbf{a}^i\cdot\nabla\mathbf{v}_h-\nu\Delta\mathbf{v}_h-\nabla q_h)-\tau_{m,t}\frac{1}{\delta t^2}(\mathbf{u}_h^n,\mathbf{v}_h)\\
&-\tau_{m,t}\frac{1}{\delta t}(\tilde{\mathbf{u}}^n,-\mathbf{a}^i\cdot\nabla\mathbf{v}_h-\nu\Delta\mathbf{v}_h-\nabla q_h)+\tau_{m,t}\frac{1}{\delta t\tau_m}(\tilde{\mathbf{u}}^n,\mathbf{v}_h),
\end{aligned}\tag{B.5}$$

where

$$\tilde{\mathbf{u}}^{n+1,i}=\tau_{m,t}\frac{1}{\delta t}\tilde{\mathbf{u}}^n+\tau_{m,t}\left[\mathbf{f}-(\mathbf{a}^i\cdot\nabla\mathbf{u}_h^{n+1,i}-\nu\Delta\mathbf{u}_h^{n+1,i}+\nabla p_h^{n+1,i})\right].\tag{B.6}$$

With $\mathbf{a}^i=\mathbf{u}_h^{n+1,i-1}+\tilde{\mathbf{u}}^{n+1,i-1}$ and $\tau_{m,t}=\left[\frac{1}{\delta t}+\frac{1}{\tau_m}\right]^{-1}$

The implementation of the dynamic subscales has been done only for a linear approximation. It means that the terms involving the laplacian $\Delta(\cdot)$ are equal to zero. Given this assumption, the simplified expressions of (B.5) and (B.6) are:

$$\begin{aligned}
&\frac{1}{\delta t}(\mathbf{u}_h^{n+1,i},\mathbf{v}_h)+(\mathbf{a}^i\cdot\nabla\mathbf{u}_h^{n+1,i},\mathbf{v}_h)+\nu(\nabla\mathbf{u}_h^{n+1,i},\nabla\mathbf{v}_h)-(p_h^{n+1,i},\nabla\cdot\mathbf{v}_h)\\
&+(q_h,\nabla\cdot\mathbf{u}_h^{n+1,i})s-\tau_{m,t}(\mathbf{a}^i\cdot\nabla\mathbf{u}_h^{n+1,i}+\nabla p_h^{n+1,i},-\mathbf{a}^i\cdot\nabla\mathbf{v}_h-\nabla q_h)\\
&+\tau_c(\nabla\cdot\mathbf{u}_h^{n+1,i},\nabla\cdot\mathbf{v}_h)-\tau_{m,t}\frac{1}{\delta t}(\mathbf{a}^i\cdot\nabla\mathbf{u}_h^{n+1,i}+\nabla p_h^{n+1,i},\mathbf{v}_h)\\
&-\tau_{m,t}\frac{1}{\delta t}(\mathbf{u}_h^{n+1,i},-\mathbf{a}^i\cdot\nabla\mathbf{v}_h-\nabla q_h)-\tau_{m,t}\frac{1}{\delta t^2}(\mathbf{u}_h^{n+1,i},\mathbf{v}_h)\\
&=\langle\mathbf{f},\mathbf{v}_h\rangle+\frac{1}{\delta t}(\mathbf{u}_h^n,\mathbf{v}_h)-\tau_{m,t}(\mathbf{f},-\mathbf{a}^i\cdot\nabla\mathbf{v}_h-\nabla q_h)-\tau_{m,t}\frac{1}{\delta t}(\mathbf{f},\mathbf{v}_h)\\
&-\tau_{m,t}\frac{1}{\delta t}(\mathbf{u}_h^n,-\mathbf{a}^i\cdot\nabla\mathbf{v}_h-\nabla q_h)-\tau_{m,t}\frac{1}{\delta t^2}(\mathbf{u}_h^n,\mathbf{v}_h)-\tau_{m,t}\frac{1}{\delta t}(\tilde{\mathbf{u}}^n,-\mathbf{a}^i\cdot\nabla\mathbf{v}_h-\nabla q_h)\\
&+\tau_{m,t}\frac{1}{\delta t\tau_m}(\tilde{\mathbf{u}}^n,\mathbf{v}_h),
\end{aligned}\tag{B.7}$$

with

$$\tilde{\mathbf{u}}^{n+1,i}=\tau_{m,t}\frac{1}{\delta t}\tilde{\mathbf{u}}^n+\tau_{m,t}\left[\mathbf{f}-(\mathbf{a}^i\cdot\nabla\mathbf{u}_h^{n+1,i}+\nabla p_h^{n+1,i})\right].\tag{B.8}$$

An alternative equation to (B.8) can be the equivalent to (4.32).

Orthogonal Subscales (OSS)

The Orthogonal Subscales (OSS) method is characterized by the following projection definition.

$$\mathcal{P} := \Pi_\tau^\perp = \mathbf{I} - \Pi_\tau \tag{B.9}$$

Then, the equivalent subscales equations to (B.3)-(B.4) are given by

$$\partial_t \tilde{\mathbf{u}} + \tau_m^{-1}\tilde{\mathbf{u}} = \Pi_\tau^\perp(\mathbf{R}_u), \tag{B.10}$$

$$\tau_c^{-1}\tilde{p} = \Pi_\tau^\perp(R_p). \tag{B.11}$$

As we have done for the ASGS method, using a Backward Euler time discretization and the Picard method for solving the nonlinearity, at the time step $n+1$ and at each iteration i, $\mathbf{u}_h^{n+1,i}$ and $p_h^{n+1,i}$ are calculated solving the following decomposed equations

$$\begin{aligned}
&\frac{1}{\delta t}(\mathbf{u}_h^{n+1,i}, \mathbf{v}_h) + (\mathbf{a}^i \cdot \nabla \mathbf{u}_h^{n+1,i}, \mathbf{v}_h) + \nu(\nabla \mathbf{u}_h^{n+1,i}, \nabla \mathbf{v}_h) - (p_h^{n+1,i}, \nabla\cdot\mathbf{v}_h) \qquad \text{(B.12)}\\
&+ (q_h, \nabla\cdot\mathbf{u}_h^{n+1,i}) - \tau_{m,t}(\mathbf{a}^i\cdot\nabla\mathbf{u}_h^{n+1,i} - \nu\Delta\mathbf{u}_h^{n+1,i} + \nabla p_h^{n+1,i}, -\mathbf{a}^i\cdot\nabla\mathbf{v}_h - \nu\Delta\mathbf{v}_h - \nabla q_h)\\
&+ \tau_c(\nabla\cdot\mathbf{u}_h^{n+1,i}, \nabla\cdot\mathbf{v}_h) - \tau_{m,t}\frac{1}{\delta t}(\mathbf{a}^i\cdot\nabla\mathbf{u}_h^{n+1,i} - \nu\Delta\mathbf{u}_h^{n+1,i} + \nabla p_h^{n+1,i}, \mathbf{v}_h)\\
&- \tau_{m,t}\frac{1}{\delta t}(\mathbf{u}_h^{n+1,i}, -\mathbf{a}^i\cdot\nabla\mathbf{v}_h - \nu\Delta\mathbf{v}_h - \nabla q_h) - \tau_{m,t}\frac{1}{\delta t^2}(\mathbf{u}_h^{n+1,i}, \mathbf{v}_h)\\
&= \langle\mathbf{f}, \mathbf{v}_h\rangle + \frac{1}{\delta t}(\mathbf{u}_h^n, \mathbf{v}_h) - \tau_{m,t}(\mathbf{f}, -\mathbf{a}^i\cdot\nabla\mathbf{v}_h - \nu\Delta\mathbf{v}_h - \nabla q_h) - \tau_{m,t}\frac{1}{\delta t}(\mathbf{f}, \mathbf{v}_h)\\
&- \tau_{m,t}\frac{1}{\delta t}(\mathbf{u}_h^n, -\mathbf{a}^i\cdot\nabla\mathbf{v}_h - \nu\Delta\mathbf{v}_h - \nabla q_h) - \tau_{m,t}\frac{1}{\delta t^2}(\mathbf{u}_h^n, \mathbf{v}_h)\\
&+ \tau_{m,t}(\Pi_\tau(\mathbf{R}_u^{i-1}), -\mathbf{a}^i\cdot\nabla\mathbf{v}_h - \nu\Delta\mathbf{v}_h - \nabla q_h) + \tau_c(\Pi_\tau(\nabla\cdot\mathbf{u}^{n+1,i-1}), \nabla\cdot\mathbf{v}_h)\\
&+ \tau_{m,t}\frac{1}{\delta t}(\Pi_\tau(\mathbf{R}_u^{i-1}), \mathbf{v}_h) - \tau_{m,t}\frac{1}{\delta t}(\tilde{\mathbf{u}}^n, -\mathbf{a}^i\cdot\nabla\mathbf{v}_h - \nu\Delta\mathbf{v}_h - \nabla q_h) + \tau_{m,t}\frac{1}{\delta t\tau_m}(\tilde{\mathbf{u}}^n, \mathbf{v}_h),
\end{aligned}$$

where

$$\tilde{\mathbf{u}}^{n+1,i} = \tau_{m,t}\frac{1}{\delta t}\tilde{\mathbf{u}}^n + \tau_{m,t}\left[\mathbf{f} - (\mathbf{a}^i\cdot\nabla\mathbf{u}_h^{n+1,i} - \nu\Delta\mathbf{u}_h^{n+1,i} + \nabla p_h^{n+1,i})\right] - \tau_{m,t}\Pi_\tau(\mathbf{R}_u^{i-1}). \tag{B.13}$$

With $\mathbf{R}_u^i = \mathbf{f} - (\mathbf{a}^i\cdot\nabla\mathbf{u}_h^{n+1,i} - \nu\Delta\mathbf{u}_h^{n+1,i} + \nabla p_h^{n+1,i})$.

As the ASGS method, the implementation of the dynamic subscales has been done only for a linear approximation. Moreover, some of the terms that appear in (B.13) can be neglected, see Chapter 4. So the resulting expressions are the following.

$$\begin{aligned}
&\frac{1}{\delta t}(\mathbf{u}_h^{n+1,i}, \mathbf{v}_h) + (\mathbf{a}^i \cdot \nabla \mathbf{u}_h^{n+1,i}, \mathbf{v}_h) + \nu(\nabla \mathbf{u}_h^{n+1,i}, \nabla \mathbf{v}_h) - (p_h^{n+1,i}, \nabla \cdot \mathbf{v}_h) \\
&+ (q_h, \nabla \cdot \mathbf{u}_h^{n+1,i}) - \tau_{m,t}(\mathbf{a}^i \cdot \nabla \mathbf{u}_h^{n+1,i} + \nabla p_h^{n+1,i}, -\mathbf{a}^i \cdot \nabla \mathbf{v}_h - \nabla q_h) \\
&- \tau_{m,t}\frac{1}{\delta t}(\mathbf{u}_h^{n+1,i}, -\mathbf{a}^i \cdot \nabla \mathbf{v}_h - \nabla q_h) + \tau_c(\nabla \cdot \mathbf{u}_h^{n+1,i}, \nabla \cdot \mathbf{v}_h) \\
&= \langle \mathbf{f}, \mathbf{v}_h \rangle + \frac{1}{\delta t}(\mathbf{u}_h^n, \mathbf{v}_h) - \tau_{m,t}(\mathbf{f}, -\mathbf{a}^i \cdot \nabla \mathbf{v}_h - \nabla q_h) \\
&+ \tau_{m,t}(\Pi_\tau(\mathbf{R}_u^{i-1}), -\mathbf{a}^i \cdot \nabla \mathbf{v}_h - \nabla q_h) \\
&- \tau_{m,t}\frac{1}{\delta t}(\tilde{\mathbf{u}}^n, -\mathbf{a}^i \cdot \nabla \mathbf{v}_h - \nabla q_h) + \tau_c(\Pi_\tau(\nabla \cdot \mathbf{u}^{n+1,i-1}), \nabla \cdot \mathbf{v}_h),
\end{aligned} \tag{B.14}$$

with

$$\tilde{\mathbf{u}}^{n+1,i} = \tau_{m,t}\frac{1}{\delta t}\tilde{\mathbf{u}}^n + \tau_{m,t}\left[\mathbf{f} - (\mathbf{a}^i \cdot \nabla \mathbf{u}_h^{n+1,i} + \nabla p_h^{n+1,i})\right] - \tau_{m,t}\Pi_\tau(\mathbf{R}_u^{i-1}). \tag{B.15}$$

Terms to be implemented

In the following table (Table B.1) we list the terms needed to be implemented for each version of the VMS methods used in Chapter 4.

Note that the quasi-static and the dynamic versions of the OSS method, have to compute almost the same terms. The difference between these two versions is the need to compute explicitly $\tilde{\mathbf{u}}^{n+1,i}$ even for the linear definition of the subscales, and the definition of the parameter $\tau_{m,t}$, which for the quasi-static version is $\tau_{m,t} = \tau_m$.

Algorithm

In this section, the generic algorithm followed to implement the VMS methods used in Chapter 4 is described. The nonlinear description of the subscales is considered in Algorithm 3.

In order to compute the velocity subscale $\tilde{\mathbf{u}}^{n+1,i}$, we must have an explicit implementation of the residual projection. For the global nonlinear iterations, the residual projection is treated implicitly. In Algorithm 4, the steps followed to compute the velocity subscale are described.

TABLE B.1: List of VMS terms

Term	Subgrid method			
	ASGS		OSS	
	QS	DYN	QS	DYN
$\frac{1}{\delta t}(\mathbf{u}_h^{n+1,i},\mathbf{v}_h)$	✓	✓	✓	✓
$(\mathbf{a}^i\cdot\nabla\mathbf{u}_h^{n+1,i},\mathbf{v}_h)$	✓	✓	✓	✓
$\nu(\nabla\mathbf{u}_h^{n+1,i},\nabla\mathbf{v}_h)$	✓	✓	✓	✓
$(p_h^{n+1,i},\nabla\cdot\mathbf{v}_h)$	✓	✓	✓	✓
$(q_h,\nabla\cdot\mathbf{u}_h^{n+1,i})$	✓	✓	✓	✓
$\tau_{m,t}(\mathbf{a}^i\cdot\nabla\mathbf{u}_h^{n+1,i},\mathbf{a}^i\cdot\nabla\mathbf{v}_h)$	✓	✓	✓	✓
$\tau_{m,t}(\mathbf{a}^i\cdot\nabla\mathbf{u}_h^{n+1,i},\nabla q_h)$	✓	✓	✓	✓
$\tau_{m,t}(\nabla p_h^{n+1,i},\mathbf{a}^i\cdot\nabla\mathbf{v}_h)$	✓	✓	✓	✓
$\tau_{m,t}(\nabla p_h^{n+1,i},\nabla q_h)$	✓	✓	✓	✓
$\tau_c(\nabla\cdot\mathbf{u}_h^{n+1,i},\nabla\cdot\mathbf{v}_h)$	✓	✓	✓	✓
$\tau_{m,t}\frac{1}{\delta t}(\mathbf{a}^i\cdot\nabla\mathbf{u}_h^{n+1,i},\mathbf{v}_h)$	✗	✓	✗	✗
$\tau_{m,t}\frac{1}{\delta t}(\nabla p_h^{n+1,i},\mathbf{v}_h)$	✗	✓	✗	✗
$\tau_{m,t}\frac{1}{\delta t}(\mathbf{u}_h^{n+1,i},-\mathbf{a}^i\cdot\nabla\mathbf{v}_h)$	✓	✓	✓	✓
$\tau_{m,t}\frac{1}{\delta t}(\mathbf{u}_h^{n+1,i},-\nabla q_h)$	✓	✓	✓	✓
$\tau_{m,t}\frac{1}{\delta t^2}(\mathbf{u}_h^{n+1,i},\mathbf{v}_h)$	✗	✓	✗	✗
$\langle\mathbf{f},\mathbf{v}_h\rangle$	✓	✓	✓	✓
$\frac{1}{\delta t}(\mathbf{u}_h^n,\mathbf{v}_h)$	✓	✓	✓	✓
$\tau_{m,t}(\mathbf{f},-\mathbf{a}^i\cdot\nabla\mathbf{v}_h)$	✓	✓	✓	✓
$\tau_{m,t}(\mathbf{f},-\nabla q_h)$	✓	✓	✓	✓
$\tau_{m,t}\frac{1}{\delta t}(\mathbf{f},\mathbf{v}_h)$	✗	✓	✗	✗
$\tau_{m,t}\frac{1}{\delta t}(\mathbf{u}_h^n,-\mathbf{a}^i\cdot\nabla\mathbf{v}_h)$	✓	✓	✓	✓
$\tau_{m,t}\frac{1}{\delta t}(\mathbf{u}_h^n,-\nabla q_h)$	✓	✓	✓	✓
$\tau_{m,t}\frac{1}{\delta t^2}(\mathbf{u}_h^n,\mathbf{v}_h)$	✗	✓	✗	✗
$\tau_{m,t}(\Pi_\tau(\mathbf{R}_u^{i-1}),-\mathbf{a}^i\cdot\nabla\mathbf{v}_h)$	✗	✗	✓	✓
$\tau_{m,t}(\Pi_\tau(\mathbf{R}_u^{i-1}),-\nabla q_h)$	✗	✗	✓	✓
$\tau_c(\Pi_\tau(\nabla\cdot\mathbf{u}^{n+1,i-1}),\nabla\cdot\mathbf{v}_h)$	✗	✗	✓	✓
$\tau_{m,t}\frac{1}{\delta t}(\tilde{\mathbf{u}}^n,-\mathbf{a}^i\cdot\nabla\mathbf{v}_h)$	✗	✓	✗	✗
$\tau_{m,t}\frac{1}{\delta t}(\tilde{\mathbf{u}}^n,-\nabla q_h)$	✗	✓	✗	✗
$\tau_{m,t}\frac{1}{\delta t\tau_m}(\tilde{\mathbf{u}}^n,\mathbf{v}_h)$	✗	✓	✗	✗
$\tilde{\mathbf{u}}^{n+1,i}$	✗	✓	✗	✓

Algorithm 3: VMS method algorithm

```
Read (or compute) $\mathbf{u}_h^0$ and set $p_h^0 = 0$, $\tilde{\mathbf{u}}^0 = \mathbf{0}$.
FOR $n = 0, ..., N-1$ DO:
    Set $i = 0$
    Set $\mathbf{u}_h^{n+1,0} = \mathbf{u}_h^n$, $p_h^{n+1,0} = p_h^n$, $\tilde{\mathbf{u}}^{n+1,0} = \tilde{\mathbf{u}}^n$
    WHILE (not converged) DO:
        $i \leftarrow i + 1$
        Set $\mathbf{a}^i = \mathbf{u}_h^{n+1,i-1} + \tilde{\mathbf{u}}^{n+1,i-1}$
        Compute $\tau_m$, $\tau_{m,t}$ and $\tau_c$
        Compute $\mathbf{u}_h^{n+1,i}$ and $p_h^{n+1,i}$ by solving (B.7) or (B.14)
        Update the subscales $\tilde{\mathbf{u}}^{n+1,i}$ from (B.8) or (B.15), see Algorithm 4
        For OSS, compute the projections $\Pi_\tau(\mathbf{R}_u^i)$ and $\Pi_\tau(\nabla \cdot \mathbf{u}^{n+1,i})$
        Check convergence
    END
    Set up the converged values $\mathbf{u}_h^{n+1} = \mathbf{u}_h^{n+1,i}$, $p_h^{n+1} = p_h^{n+1,i}$ and $\tilde{\mathbf{u}}^{n+1} = \tilde{\mathbf{u}}^{n+1,i}$
END
```

Algorithm 4: Velocity subscale computation algorithm

```
Compute the residual without the contribution of the subscale on the convective
term. That is:
        $\mathbf{R}_u^{n+1,i} = \mathbf{f} - \left( \frac{1}{\delta t}(\mathbf{u}_h^{n+1,i} - \mathbf{u}_h^n) + \mathbf{u}_h^{n+1,i} \cdot \nabla \mathbf{u}_h^{n+1,i} + \nabla p_h^{n+1} \right)$
If OSS, substract the projection. Here we have the projection $\Pi(\mathbf{R}_u^{n+1,i-1})$
        $\mathbf{R}_u^{n+1,i} - \Pi(\mathbf{R}_u^{n+1,i-1}) \rightarrow \mathbf{R}_u^{n+1,i}$
WHILE (not converged) DO:
    Compute, and add to the residual, the contribution of the subscale on the
convective term:
        $\mathbf{R}_u^{n+1,i} - (\tilde{\mathbf{u}}^{n+1,i,k-1} \cdot \nabla \mathbf{u}_h^{n+1,i}) \rightarrow \hat{\mathbf{R}}_u^{n+1,i,k-1}$
    Compute $\tau_m$, $\tau_{m,t}$ and $\tau_c$
    Update the subscale:
        $\tilde{\mathbf{u}}^{n+1,i,k} = \tau_{m,t} \left( \frac{1}{\delta t} \tilde{\mathbf{u}}^n + \hat{\mathbf{R}}_u^{n+1,i,k-1} \right)$
    Check convergence
END
```

Appendix C

Butcher tableaus for the Segregated Runge-Kutta schemes

Butcher tableaus

We define different settings of the Butcher tableaus, depending on the stages and the order of accuracy of the time integration scheme. For the DIRK tableaus we use the following notation:

$$\begin{array}{c|c} \mathbf{c} & A \\ \hline & \mathbf{b} \end{array} = \begin{array}{c|cccc} c_1 & a_{11} & 0 & ... & 0 \\ c_2 & a_{21} & a_{22} & ... & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_s & a_{s1} & a_{s2} & ... & a_{ss} \\ \hline & b_1 & b_2 & ... & b_s \end{array}$$

with the convention $c_i = \sum_{j=1}^{s} a_{ij}$.

In the explicit case the tables are similar, but with zeros on the diagonal. A r-stage explicit scheme can be written as follows.

$$\begin{array}{c|c} \hat{\mathbf{c}} & \hat{A} \\ \hline & \hat{\mathbf{b}} \end{array} = \begin{array}{c|ccccc} 0 & 0 & 0 & 0 & ... & 0 \\ \hat{c}_2 & \hat{a}_{21} & 0 & 0 & ... & 0 \\ \hat{c}_3 & \hat{a}_{31} & \hat{a}_{32} & 0 & ... & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \hat{c}_r & \hat{a}_{r1} & \hat{a}_{r2} & \hat{a}_{r3} & ... & 0 \\ \hline & \hat{b}_1 & \hat{b}_2 & \hat{b}_3 & ... & \hat{b}_r \end{array}$$

Note that in general the number of implicit stages (s) are not the same than the explicit ones (r). In fact, usually we have one additional stage for the explicit Butcher Tableau ($r = s + 1$). Then, in order to facilitate the IMEX schemes implementation, we pad the implicit s-stage scheme with zeroes, obtaining a new tableau with the same size than the explicit one. We can easily see that with this procedure we will always have $\mathbf{U}_1 = \mathbf{U}_n$ since the first rows for the explicit and the implicit tableaus are filled with zeros. Given that, no computation is needed at the first stage, so we will say that the IMEX scheme has s stages when the associated Butcher tableaus are of the size $(s + 1)$.

We select from the literature some Butcher table settings for IMEX RK schemes that have been already tested for convection-diffusion type problems, mostly from [6], but also from [29]. It is important to keep in mind that the properties described below are given for the ODE system in time that arises from the convection-diffusion equation and may not be preserved for an index 2 DAE system like the semi-discrete incompressible Navier-Stokes problem.

- **1 stage, 1st order (1-1):** This is the so called *Forward-Backward Euler* and was proposed by Ascher *et al.* in [6]. There are two different versions of this scheme, one that satisfies $\hat{\mathbf{b}} = \mathbf{b}$ and another that does not satisfy this condition. We use the first version, just to be consistent with the following schemes, although it requires an extra evaluation per step of the explicit counterpart. Note that this scheme has stability problems when the implicit operator is zero. Furthermore, the choice $\hat{\mathbf{b}} = \mathbf{b}$ preserves the incompressibility constraint under the conditions stated in Proposition 6.3.2.

Implicit:

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 0 & 1 \\ \hline & 0 & 1 \end{array}$$

Explicit:

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 1 & 0 \\ \hline & 0 & 1 \end{array}$$

- **1 stage, 2nd order (1-2):** This scheme was also proposed in [6], where they called it as *Implicit-explicit midpoint*. It is second-order accurate and as stated in [6] performs comparably to the popular Crank-Nicolson Adams-Bashforth method, with the addition that it has better symmetry properties. It is important to remark that it does not have any attenuation at the stiffness limit.

Implicit:

$$\begin{array}{c|cc} 0 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} \\ \hline & 0 & 1 \end{array}$$

Explicit:

$$\begin{array}{c|cc} 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 \\ \hline & 0 & 1 \end{array}$$

- **2 stages, 2nd order, case 1 (2-2/1):** Another second-order IMEX RK scheme was proposed in [6], but in this case it has two stages. This scheme turns out to be *L*-stable and it is based on a stiffly accurate two-stage, second-order DIRK scheme.

Implicit:

$$\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ \gamma & 0 & \gamma & 0 \\ 1 & 0 & 1-\gamma & \gamma \\ \hline & 0 & 1-\gamma & \gamma \end{array}$$

Explicit:

$$\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ \gamma & \gamma & 0 & 0 \\ 1 & \delta & 1-\delta & 0 \\ \hline & 0 & 1-\gamma & \gamma \end{array}$$

with $\gamma = (2 - \sqrt{2})/2$ and $\delta = -2\sqrt{2}/3$.

- **2 stages, 2nd order, case 2 (2-2/2):** An alternative definition of the previous scheme is also proposed in [6], where the explicit table is defined satisfying the condition $\hat{b}_j = \hat{a}_{sj}$ for all $j = 1, ..., s$ instead of satisfying that $\hat{\mathbf{b}} = \mathbf{b}$. The resulting scheme is also *L*-stable and second-order accurate in time. Note that this scheme will not satisfy the incompressibility constrain under the assumptions exposed in Proposition 6.3.2.

Implicit:

$$\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ \gamma & 0 & \gamma & 0 \\ 1 & 0 & 1-\gamma & \gamma \\ \hline & 0 & 1-\gamma & \gamma \end{array}$$

Explicit:

$$\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ \gamma & \gamma & 0 & 0 \\ 1 & \delta & 1-\delta & 0 \\ \hline & \delta & 1-\delta & 0 \end{array}$$

with $\gamma = (2 - \sqrt{2})/2$ and $\delta = 1 - 1/(2\gamma)$.

- **2 stages, 3rd order (2-3):** A two-stage combination, third-order accurate in time IMEX scheme is defined in [6]. In this case, the implicit table is based on a two-stage, third-order DIRK scheme.

Implicit:

$$\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ \gamma & 0 & \gamma & 0 \\ 1-\gamma & 0 & 1-2\gamma & \gamma \\ \hline & 0 & \frac{1}{2} & \frac{1}{2} \end{array}$$

Explicit:

$$\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ \gamma & \gamma & 0 & 0 \\ 1-\gamma & \gamma-1 & 2(1-\gamma) & 0 \\ \hline & 0 & \frac{1}{2} & \frac{1}{2} \end{array}$$

with $\gamma = (3 + \sqrt{3})/6$.

- **3 stages, 3rd order (3-3):** A stiffly accurate third-order, three-stage DIRK scheme combined with a four-stage third-order explicit RK scheme is proposed in [6]. The resulting scheme is third-order accurate in time and *L*-stable.

Implicit:

$$\begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ 0.4358665215 & 0 & 0.4358665215 & 0 & 0 \\ 0.7179332608 & 0 & 0.2820667392 & 0.4358665215 & 0 \\ 1 & 0 & 1.208496649 & -0.644363171 & 0.4358665215 \\ \hline & 0 & 1.208496649 & -0.644363171 & 0.4358665215 \end{array}$$

Explicit:

$$\begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ 0.4358665215 & 0.4358665215 & 0 & 0 & 0 \\ 0.7179332608 & 0.3212788860 & 0.3966543747 & 0 & 0 \\ 1 & -0.105858296 & 0.5529291479 & 0.5529291479 & 0 \\ \hline & 0 & 1.208496649 & -0.644363171 & 0.4358665215 \end{array}$$

- **4 stages, 3rd order (4-3):** Finally, it is proposed in [6] a four-stage third-order accurate in time IMEX scheme.

Implicit:

$$\begin{array}{c|ccccc}
0 & 0 & 0 & 0 & 0 & 0 \\
1/2 & 0 & 1/2 & 0 & 0 & 0 \\
2/3 & 0 & 1/6 & 1/2 & 0 & 0 \\
1/2 & 0 & -1/2 & 1/2 & 1/2 & 0 \\
1 & 0 & 3/2 & -3/2 & 1/2 & 1/2 \\
\hline
 & 0 & 3/2 & -3/2 & 1/2 & 1/2
\end{array}$$

Explicit:

$$\begin{array}{c|ccccc}
0 & 0 & 0 & 0 & 0 & 0 \\
1/2 & 1/2 & 0 & 0 & 0 & 0 \\
2/3 & 11/18 & 1/18 & 0 & 0 & 0 \\
1/2 & 5/6 & -5/6 & 1/2 & 0 & 0 \\
1 & 1/4 & 7/4 & 3/4 & -7/4 & 0 \\
\hline
 & 1/4 & 7/4 & 3/4 & -7/4 & 0
\end{array}$$

- **5 stages, 3rd order (5-3):** Boscarino developed in [29] a third-order five-stage IMEX RK scheme that does not suffer from order reduction phenomena when stiff problems containing a non-stiff term and a stiff term are solved. Note that in this case we say that the method has five stages because the first column of the implicit table is not zero, although the first row is still full of zeros so the first stage will give $\mathbf{U}_1 = \mathbf{U}_n$.

Implicit:

$$\begin{array}{c|ccccc}
0 & 0 & 0 & 0 & 0 & 0 \\
c_2 & \gamma & \gamma & 0 & 0 & 0 \\
c_3 & a_{31} & a_{32} & \gamma & 0 & 0 \\
c_4 & a_{41} & a_{42} & a_{43} & \gamma & 0 \\
1 & b_1 & b_2 & b_3 & b_4 & \gamma \\
\hline
 & b_1 & b_2 & b_3 & b_4 & \gamma
\end{array}$$

Explicit:

$$\begin{array}{c|ccccc}
0 & 0 & 0 & 0 & 0 & 0 \\
c_2 & \gamma & 0 & 0 & 0 & 0 \\
c_3 & \hat{a}_{31} & \hat{a}_{32} & 0 & 0 & 0 \\
c_4 & \hat{a}_{41} & \hat{a}_{42} & \hat{a}_{43} & 0 & 0 \\
1 & \hat{a}_{51} & \hat{a}_{52} & \hat{a}_{53} & \hat{a}_{54} & 0 \\
\hline
 & b_1 & b_2 & b_3 & b_4 & \gamma
\end{array}$$

The parameter values can be found in the appendix of [29].

Bibliography

[1] R. Alexander. Diagonally Implicit Runge–Kutta Methods for Stiff O.D.E.'s. *SIAM Journal on Numerical Analysis*, 14(6):1006–1021, 1977.

[2] Andrea D. Beck and Gregor J. Gassner. Numerical Simulation of the Taylor-Green Vortex at Re=1600 with the Discontinuous Galerkin Spectral Element Method for well-resolved and underresolved scenarios, 2012.

[3] M. Anyoji, K. Nose, S. Ida, D. Numata, H. Nagai, and K. Asai. Low reynolds number airfoil testing in a mars wind tunnel. *AIAA Paper*, 4627, 2010.

[4] A. Arakawa. Computational design for long-term numerical integration of the equations of fluid motion: Two-dimensional incompressible flow. part i. *Journal of Computational Physics*, 1(1):119–143, Aug. 1966.

[5] M. Arnold. Half-explicit Runge-Kutta methods with explicit stages for differential-algebraic systems of index 2. *BIT Numerical Mathematics*, 38(3):415–438, 1998.

[6] U. M. Ascher, S. J. Ruuth, and R. J. Spiteri. Implicit-explicit Runge-Kutta methods for time-dependent partial differential equations. *Applied Numerical Mathematics*, 25(2–3):151–167, 1997.

[7] U. M. Ascher, S. J. Ruuth, and B. T. R. Wetton. Implicit-Explicit Methods for Time-Dependent Partial Differential Equations. *SIAM Journal on Numerical Analysis*, 32(3):797–823, 1995.

[8] M. Avila, R. Codina, and J. Principe. Large eddy simulation of low Mach number flows using dynamic and orthogonal subgrid scales. *Computers & Fluids*, 99:44–66, 2014.

[9] M. Avila, J. Principe, and R. Codina. A finite element dynamical nonlinear subscale approximation for the low Mach number flow equations. *Journal of Computational Physics*, 230(10–11):7988–8009, 2011.

[10] P. I. Babuska. Error-bounds for finite element method. *Numerische Mathematik*, 16(4):322–333, 1971.

[11] S. Badia. On stabilized finite element methods based on the Scott-Zhang projector. Circumventing the inf-sup condition for the Stokes problem. *Computer Methods in Applied Mechanics and Engineering*, 247–248(0):65–72, 2012.

[12] S. Badia and R. Codina. Algebraic pressure segregation methods for the incompressible Navier-Stokes equations. *Archives of Computational Methods in Engineering*, 15(3):343–369, 2008.

[13] S. Badia and R. Codina. On a multiscale approach to the transient Stokes problem: Dynamic subscales and anisotropic space–time discretization. *Applied Mathematics and Computation*, 207(2):415–433, 2009.

[14] S. Badia, R. Codina, and J. V. Gutiérrez-Santacreu. Long-term stability estimates and existence of a global attractor in a finite element approximation of the Navier-Stokes equations with numerical subgrid scale modeling. *SIAM J. Numer. Anal.*, 48(3):1013–1037, 2010.

[15] S. Badia and J. V. Gutiérrez-Santacreu. Convergence towards weak solutions of the Navier–Stokes equations for a finite element approximation with numerical subgrid-scale modelling. *IMA Journal of Numerical Analysis*, 34(3):1193–1221, 2014.

[16] S. Badia, A. Martín, and J.Principe. Enhanced balancing Neumann-Neumann preconditioning in computational fluid and solid mechanics. *International Journal for Numerical Methods in Engineering*, 96(4):203–230, 2013.

[17] S. Badia, A. F. Martín, and J. Principe. On an overlapped coarse/fine implementation of balancing domain decomposition with inexact solvers. *Submitted*, 2014.

[18] S. Badia, A. F. Martín, and R. Planas. Block recursive LU preconditioners for the thermally coupled incompressible inductionless MHD problem. *Journal of Computational Physics*, 274:562–591, 2014.

[19] S. Badia, A. F. Martín, and J. Principe. Implementation and Scalability Analysis of Balancing Domain Decomposition Methods. *Archives of Computational Methods in Engineering*, 20(3):239–262, 2013.

[20] S. Badia, A. F. Martín, and J. Principe. A Highly Scalable Parallel Implementation of Balancing Domain Decomposition by Constraints. *SIAM Journal on Scientific Computing*, 36(2):C190–C218, 2014.

[21] S. Badia and H. Nguyen. Balancing Domain Decomposition by perturbation. *Submitted*, 2015.

[22] K.-J. Bathe. *Finite Element Procedures*. Klaus-Jurgen Bathe, 2006.

[23] Y. Bazilevs, V. M. Calo, J. A. Cottrell, T. J. R. Hughes, A. Reali, and G. Scovazzi. Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows. *Computer Methods in Applied Mechanics and Engineering*, 197(1–4):173–201, 2007.

[24] Y. Bazilevs and T. J. R. Hughes. Weak imposition of Dirichlet boundary conditions in fluid mechanics. *Computers & Fluids*, 36(1):12–26, 2007.

[25] Y. Bazilevs, C. Michler, V. M. Calo, and T. J. R. Hughes. Weak Dirichlet boundary conditions for wall-bounded turbulent flows. *Computer Methods in Applied Mechanics and Engineering*, 196(49–52):4853–4862, 2007.

[26] T. Belytschko and T. J. R. Hughes. *Computational Methods for Transient Analysis*. North-Holland, 1983.

[27] P. Boom and D. Zingg. Time-accurate flow simulations using an efficient newton-krylov-schur approach with high-order temporal and spatial discretization. In *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*. American Institute of Aeronautics and Astronautics, 2013.

[28] J. P. Boris, F. F. Grinstein, E. S. Oran, and R. L. Kolbe. New insights into large eddy simulation. *Fluid Dynamics Research*, 10(4-6):199, 1992.

[29] S. Boscarino. On an accurate third order implicit-explicit Runge-Kutta method for stiff problems. *Applied Numerical Mathematics*, 59(7):1515–1528, 2009.

[30] M. Braack and E. Burman. Local Projection Stabilization for the Oseen Problem and Its Interpretation as a Variational Multiscale Method. *SIAM Journal on Numerical Analysis*, 43(6):2544–2566, 2006.

[31] M. Brachet. Direct simulation of three-dimensional turbulence in the Taylor—Green vortex. *Fluid Dynamics Research*, 8(1–4):1–8, 1991.

[32] M. E. Brachet, D. I. Meiron, S. A. Orszag, B. G. Nickel, R. H. Morf, and U. Frisch. Small-scale structure of the Taylor–Green vortex. *Journal of Fluid Mechanics*, 130:411–452, 1983.

[33] V. Brasey and E. Hairer. Half-explicit Runge-Kutta methods for differential-algebraic systems of index 2. *SIAM J. Numer. Anal.*, 30(2):538–552, 1993.

[34] R. Brayton, F. Gustavson, and G. Hachtel. A new efficient algorithm for solving differential-algebraic systems using implicit backward differentiation formulas. *Proceedings of the IEEE*, 60(1):98–108, 1972.

[35] S. Brenner and R. Scott. *The Mathematical Theory of Finite Element Methods*. Springer Science & Business Media, 2007.

[36] F. Brezzi. On the existence, uniqueness and approximation of saddle-point problems arising from lagrangian multipliers. *Revue française d'automatique, informatique, recherche opérationnelle. Analyse numérique*, 8(2):129–151, 1974.

[37] F. Brezzi, T. J. Hughes, and E. Süli. Variational approximation of flux in conforming finite element methods for elliptic partial differential equations: a model problem. *Atti della Accademia Nazionale dei Lincei. Classe di Scienze Fisiche, Matematiche e Naturali. Rendiconti Lincei. Matematica e Applicazioni*, 12(3):159–166, 2001.

[38] A. N. Brooks and T. J. Hughes. Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 32(1–3):199–259, 1982.

[39] E. Burman and M. A. Fernández. Galerkin Finite Element Methods with Symmetric Pressure Stabilization for the Transient Stokes Equations: Stability and Convergence Analysis. *SIAM Journal on Numerical Analysis*, 47(1):409–439, 2009.

[40] R. Calderer and A. Masud. Residual-based variational multiscale turbulence models for unstructured tetrahedral meshes. *Computer Methods in Applied Mechanics and Engineering*, 254:238–253, 2013.

[41] R. Calderer and A. Masud. Residual-based variational multiscale turbulence models for unstructured tetrahedral meshes. *Computer Methods in Applied Mechanics and Engineering*, 254:238–253, 2013.

[42] V. Calo. *Residual based mulstiscale turbulence modeling: finite volume simulations of bypass transition*. PhD thesis, Department of Civil and Environmental Engineering, Stanford University, 2004.

[43] C. Canuto. *Spectral methods in fluid dynamics*. Springer-Verlag, 1988.

[44] M. Case, V. Ervin, A. Linke, and L. Rebholz. A connection between Scott-Vogelius and grad-div stabilized Taylor-Hood FE approximations of the Navier-Stokes equations. *SIAM Journal on Numerical Analysis*, 49(4):1461–1481, 2011.

[45] J. R. Cash. Diagonally implicit Runge-Kutta formulae with error estimates. *IMA Journal of Applied Mathematics*, 24(3):293–301, 1979.

[46] F. Chalot, B. Marquez, M. Ravachol, F. Ducros, F. Nicoud, and T. Poinsot. A consistent finite element approach to large eddy simulation. *AIAA paper*, 2652:1998, 1998.

[47] H. Choi and P. Moin. Effects of the computational time step on numerical solutions of turbulent flow. *Journal of Computational Physics*, 113(1):1–4, 1994.

[48] P. G. Ciarlet. *The Finite Element Method for Elliptic Problems*. Elsevier, 1978.

[49] P. G. Ciarlet and P. A. Raviart. General lagrange and hermite interpolation in Rn with applications to finite element methods. *Archive for Rational Mechanics and Analysis*, 46(3):177–199, 1972.

[50] R. Codina. Stabilization of incompressibility and convection through orthogonal sub-scales in finite element methods. *Computer Methods in Applied Mechanics and Engineering*, 190(13–14):1579–1599, 2000.

[51] R. Codina. Stabilized finite element approximation of transient incompressible flows using orthogonal subscales. *Computer Methods in Applied Mechanics and Engineering*, 191(39–40):4295–4321, 2002.

[52] R. Codina. Analysis of a stabilized finite element approximation of the Oseen equations using orthogonal subscales. *Applied Numerical Mathematics*, 58(3):264–283, 2008.

[53] R. Codina and S. Badia. On some pressure segregation methods of fractional-step type for the finite element approximation of incompressible flow problems. *Computer Methods in Applied Mechanics and Engineering*, 195(23-24):2900–2918, 2006.

[54] R. Codina, J. Principe, and S. Badia. Dissipative structure and long term behavior of a finite element approximation of incompressible flows with numerical subgrid scale modeling. In R. Borst and E. Ramm, editors, *Multiscale Methods in Computational Mechanics*, volume 55 of *Lecture Notes in Applied and Computational Mechanics*, pages 75–93. Springer Netherlands, 2011.

[55] R. Codina, J. Principe, O. Guasch, and S. Badia. Time dependent subscales in the stabilized finite element approximation of incompressible flow problems. *Computer Methods in Applied Mechanics and Engineering*, 196(21–24):2413–2430, 2007.

[56] O. Colomés, S. Badia, R. Codina, and J. 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.

[57] O. Colomés and S. Badia. Segregated Runge–Kutta methods for the incompressible Navier–Stokes equations. *International Journal for Numerical Methods in Engineering*, 105(5):372–400, 2016. nme.4987.

[58] O. Colomés, S. Badia, and J. Principe. Mixed finite element methods with convection stabilization for the large eddy simulation of incompressible turbulent flows. *Submitted*, 2015.

[59] G. Comte-Bellot and S. Corrsin. Simple eulerian time correlation of full-and narrow-band velocity signals in grid-generated, turbulence. *Journal of Fluid Mechanics*, 48(02):273–337, 1971.

[60] L. Davidson. LESFOIL: A European Project on Large Eddy Simulations Around a High-lift Airfoil at High Reynolds Number. In *Chalmers Publication Library (CPL)*, 2000.

[61] L. Davidson, D. Cokljat, J. Fröhlich, M. A. Leschziner, C. Mellen, and W. Rodi. *LESFOIL: Large Eddy Simulation of Flow Around a High Lift Airfoil: Results of the Project LESFOIL Supported by the European Union 1998 – 2001*. Springer Science & Business Media, 2012.

[62] P. A. Davidson. *Turbulence: An Introduction for Scientists and Engineers*. OUP Oxford, 2004.

[63] J. J. de Swart and G. Söderlind. On the construction of error estimators for implicit Runge-Kutta methods. *Journal of Computational and Applied Mathematics*, 86(2):347–358, 1997.

[64] D. C. Del Rey Fernández, J. E. Hicken, and D. W. Zingg. Review of summation-by-parts operators with simultaneous approximation terms for the numerical solution of partial differential equations. *Computers & Fluids*, 95:171–196, May 2014.

[65] W. Dettmer and D. Perić. An analysis of the time integration algorithms for the finite element solutions of incompressible Navier–Stokes equations based on a stabilised formulation. *Computer Methods in Applied Mechanics and Engineering*, 192(9–10):1177–1226, 2003.

[66] C. R. Dohrmann. A preconditioner for substructuring based on constrained energy minimization. *SIAM Journal on Scientific Computing*, 25(1):246–258, 2003.

[67] J. Donea, S. Giuliani, H. Laval, and L. Quartapelle. Finite element solution of the unsteady Navier-Stokes equations by a fractional step method. *Computer Methods in Applied Mechanics and Engineering*, 30(1):53–73, 1982.

[68] J. Douglas and J. P. Wang. An absolutely stabilized finite element method for the Stokes problem. *Mathematics of Computation*, 52(186):495–508, 1989.

[69] H. C. Elman, D. J. Silvester, and A. J. Wathen. *Finite Elements and Fast Iterative Solvers : with Applications in Incompressible Fluid Dynamics: with Applications in Incompressible Fluid Dynamics*. Oxford University Press, 2005.

[70] C. R. Ethier and D. A. Steinman. Exact fully 3d Navier-Stokes solutions for benchmarking. *International Journal for Numerical Methods in Fluids*, 19(5):369–375, 1994.

[71] D. Fauconnier, C. De Langhe, and E. Dick. Construction of explicit and implicit dynamic finite difference schemes and application to the large-eddy simulation of the Taylor–Green vortex. *Journal of Computational Physics*, 228(21):8053–8084, 2009.

[72] Y. Feng and D. Perić. A time-adaptive space-time finite element method for incompressible lagrangian flows with free surfaces: computational issues. *Computer Methods in Applied Mechanics and Engineering*, 190(5-7):499–518, 2000.

[73] C. Foias, O. Manley, R. Rosa, and R. Temam. *Navier-Stokes Equations and Turbulence*. Cambridge University Press, 2001.

[74] D. Forti and L. Dedè. Semi-implicit BDF time discretization of the Navier–Stokes equations with VMS-LES modeling in a High Performance Computing framework. *Computers & Fluids*, 117:168–182, 2015.

[75] L. P. Franca and T. J. R. Hughes. Two classes of mixed finite element methods. *Computer Methods in Applied Mechanics and Engineering*, 69(1):89–129, 1988.

[76] C. Fureby and F. F. Grinstein. Large eddy simulation of high-Reynolds-number free and wall-bounded flows. *Journal of Computational Physics*, 181(1):68–97, 2002.

[77] G Lube, G Rapin, and J Löwe. Local projection stabilization for incompressible flows: Equal-order vs. inf-sup stable interpolation. *Electronic Transactions on Numerical Analysis*, 32:106–122, 2008.

[78] P. Gamnitzer, V. Gravemeier, and W. A. Wall. Time-dependent subgrid scales in residual-based large eddy simulation of turbulent channel flow. *Computer Methods in Applied Mechanics and Engineering*, 199(13-16):819–827, 2010.

[79] G. Gassner. A skew-symmetric discontinuous galerkin spectral element discretization and its relation to SBP-SAT finite difference methods. *SIAM Journal on Scientific Computing*, 35(3):A1233–A1253, Jan. 2013.

[80] G. J. Gassner and A. D. Beck. On the accuracy of high-order discretizations for underresolved turbulence simulations. *Theoretical and Computational Fluid Dynamics*, 27(3-4):221–237, 2013.

[81] T. Gelhard, G. Lube, M. A. Olshanskii, and J.-H. Starcke. Stabilized finite element schemes with LBB-stable elements for incompressible flows. *Journal of Computational and Applied Mathematics*, 177(2):243–267, 2005.

[82] V. Gravemeier, M. W. Gee, M. Kronbichler, and W. A. Wall. An algebraic variational multiscale–multigrid method for large eddy simulation of turbulent flow. *Computer Methods in Applied Mechanics and Engineering*, 199(13–16):853–864, 2010.

[83] P. M. Gresho, D. F. Griffiths, and D. J. Silvester. Adaptive Time-Stepping for Incompressible Flow Part I: Scalar Advection-Diffusion. *SIAM Journal on Scientific Computing*, 30(4):2018–2054, 2008.

[84] F. F. Grinstein, L. G. Margolin, and W. J. Rider. *Implicit large eddy simulation: computing turbulent fluid dynamics*. Cambridge university press, 2007.

[85] O. Guasch and R. Codina. Statistical behavior of the orthogonal subgrid scale stabilization terms in the finite element large eddy simulation of turbulent flows. *Computer Methods in Applied Mechanics and Engineering*, 261–262:154–166, 2013.

[86] J.-L. Guermond. Faedo–Galerkin weak solutions of the Navier–Stokes equations with Dirichlet boundary conditions are suitable. *Journal de Mathématiques Pures et Appliquées*, 88(1):87–106, 2007.

[87] J. Gullbrand and F. K. Chow. The effect of numerical errors and turbulence models in large-eddy simulations of channel flow, with and without explicit filtering. *Journal of Fluid Mechanics*, 495:323–341, 2003.

[88] K. Gustafsson. Control theoretic techniques for stepsize selection in explicit Runge-Kutta methods. *ACM Trans. Math. Softw.*, 17(4):533–554, 1991.

[89] K. Gustafsson. Control-theoretic Techniques for Stepsize Selection in Implicit Runge-Kutta Methods. *ACM Trans. Math. Softw.*, 20(4):496–517, 1994.

[90] E. Hairer, C. Lubich, and M. Roche. *The Numerical Solution of Differential-Algebraic Systems by Runge-Kutta Methods*, volume 1409. Springer Verlag, 1989.

[91] E. Hairer, S. P. Nørsett, and G. Wanner. *Solving Ordinary Differential Equations I: Nonstiff Problems*. Springer Science & Business Media, 2008.

[92] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*. Springer, 1993.

[93] F. Ham, F. Lien, and A. Strong. A fully conservative second-order finite difference scheme for incompressible flow on nonuniform grids. *Journal of Computational Physics*, 177(1):117–133, 2002.

[94] R. C. Hastings and B. R. Williams. Studies of the Flow Field near a NACA 4412 Aerofoil at Nearly Maximum Lift. Technical report, 1984.

[95] T. Heister and G. Rapin. Efficient augmented Lagrangian-type preconditioning for the Oseen problem using Grad-Div stabilization. *International Journal for Numerical Methods in Fluids*, 71(1):118–134, 2013.

[96] S. Hickel, N. A. Adams, and J. A. Domaradzki. An adaptive local deconvolution method for implicit LES. *Journal of Computational Physics*, 213(1):413–436, Mar. 2006.

[97] J. O. Hinze. *Turbulence*. McGraw-Hill, 1975.

[98] M.-C. Hsu, Y. Bazilevs, V. Calo, T. Tezduyar, and T. Hughes. Improving stability of stabilized and multiscale formulations in flow simulations at small time steps. *Computer Methods in Applied Mechanics and Engineering*, 199(13–16):828–840, 2010.

[99] T. Hughes, J. Cottrell, and Y. Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 194(39–41):4135 – 4195, 2005.

[100] T. J. Hughes. Multiscale phenomena: Green's functions, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles and the origins of stabilized methods. *Computer Methods in Applied Mechanics and Engineering*, 127(1–4):387–401, 1995.

[101] T. J. Hughes, G. R. Feijóo, L. Mazzei, and J.-B. Quincy. The variational multiscale method—a paradigm for computational mechanics. *Computer Methods in Applied Mechanics and Engineering*, 166(1–2):3–24, 1998.

[102] T. J. Hughes, L. P. Franca, and M. Balestra. A new finite element formulation for computational fluid dynamics: V. Circumventing the Babuska-Brezzi condition: a stable Petrov-Galerkin formulation of the stokes problem accommodating equal-order interpolations. *Computer Methods in Applied Mechanics and Engineering*, 59(1):85–99, 1986.

[103] T. J. R. Hughes, L. Mazzei, and K. E. Jansen. Large Eddy Simulation and the variational multiscale method. *Computing and Visualization in Science*, 3(1-2):47–59, 2000.

[104] T. J. R. Hughes, L. Mazzei, and A. A. Oberai. The multiscale formulation of large eddy simulation: Decay of homogeneous isotropic turbulence. *Physics of fluids*, 13(2):505–512, Feb. 2001.

[105] T. J. R. Hughes, A. A. Oberai, and L. Mazzei. Large eddy simulation of turbulent channel flows by the variational multiscale method. *Physics of Fluids (1994-present)*, 13(6):1784–1799, 2001.

[106] Jacobs, Eastman N Ward, Kenneth E Pinkerton, Robert M. The characteristics of 78 related airfoil sections from tests in the variable-density wind tunnel. *naca-report-460*, 1933.

[107] K. E. Jansen. A stabilized finite element method for computing turbulence. *Computer Methods in Applied Mechanics and Engineering*, 174(3-4):299–317, 1999.

[108] K. E. Jansen, C. H. Whiting, and G. M. Hulbert. A generalized-α method for integrating the filtered Navier–Stokes equations with a stabilized finite element method. *Computer Methods in Applied Mechanics and Engineering*, 190(3):305–319, 2000.

[109] JB Chapelier, M. De La Llave Plata, and F. Renac, E. Martin. Final abstract for ONERA Taylor-Green DG participation, 2012.

[110] E. W. Jenkins, V. John, A. Linke, and L. G. Rebholz. On the parameter choice in grad-div stabilization for the Stokes equations. *Advances in Computational Mathematics*, 40(2):491–516, 2013.

[111] J. Jimenez. A selection of test cases for the validation of large-eddy simulations of turbulent flows. Technical report, 1998.

[112] V. John. Slip with friction and penetration with resistance boundary conditions for the Navier–Stokes equations—numerical tests and aspects of the implementation. *Journal of Computational and Applied Mathematics*, 147(2):287–300, 2002.

[113] V. John and A. Kindl. Variants of projection-based finite element variational multiscale methods for the simulation of turbulent flows. *International Journal for Numerical Methods in Fluids*, 56(8):1321–1328, 2008.

[114] V. John and A. Kindl. Numerical studies of finite element variational multiscale methods for turbulent flow simulations. *Computer Methods in Applied Mechanics and Engineering*, 199(13-16):841–852, 2010.

[115] V. John and J. Rang. Adaptive time step control for the incompressible Navier–Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 199(9–12):514–524, 2010.

[116] H.-J. Kaltenbach and H. Choi. Large-eddy simulation of flow around an airfoil on a structured mesh. Technical report, 1995.

[117] N. A. Kampanis and J. A. Ekaterinaris. A staggered grid, high-order accurate method for the incompressible Navier-Stokes equations. *Journal of Computational Physics*, 215(2):589–613, 2006.

[118] D. A. Kay, P. M. Gresho, D. F. Griffiths, and D. J. Silvester. Adaptive Time-Stepping for Incompressible Flow Part II: Navier–Stokes Equations. *SIAM Journal on Scientific Computing*, 32(1):111–128, 2010.

[119] D.-H. Kim and J.-H. Yang. Boundary Layer and Near-Wake Measurements of NACA 0012 Airfoil at Low Reynolds Numbers (AIAA), 2009.

[120] J. Kim and P. Moin. Application of a fractional-step method to incompressible Navier-Stokes equations. *Journal of Computational Physics*, 59(2):308–323, 1985.

[121] J. Kim, P. Moin, and R. Moser. Turbulence statistics in fully developed channel flow at low Reynolds number. *Journal of Fluid Mechanics*, 177:133–166, 1987.

[122] R. Knikker. Study of a staggered fourth-order compact scheme for unsteady incompressible viscous flows. *International Journal for Numerical Methods in Fluids*, 59(10):1063–1092, 2009.

[123] R. Kojima, T. Nonomura, A. Oyama, and K. Fujii. Large-Eddy Simulation of Low-Reynolds-Number Flow Over Thick and Thin NACA Airfoils. *Journal of Aircraft*, 50(1):187–196, 2013.

[124] A. Kolmogorov. The Local Structure of Turbulence in Incompressible Viscous Fluid for Very Large Reynolds' Numbers. *Akademiia Nauk SSSR Doklady*, 30:301–305, 1941.

[125] B. Koobus and C. Farhat. A variational multiscale method for the large eddy simulation of compressible turbulent flows on unstructured meshes–application to vortex shedding. *Computer Methods in Applied Mechanics and Engineering*, 193(15-16):1367–1383, 2004.

[126] O. A. Ladyzhenskaya and R. A. Silverman. *The mathematical theory of viscous incompressible flow*, volume 76. Gordon and Breach New York, 1969.

[127] J. D. Lambert. *Numerical Methods for Ordinary Differential Systems: The Initial Value Problem*. Wiley, 1991.

[128] H. Le and P. Moin. An improvement of fractional step methods for the incompressible Navier-Stokes equations. *Journal of Computational Physics*, 92(2):369–379, 1991.

[129] M. Lesieur. *Turbulence in Fluids: Stochastic and Numerical Modelling*. Springer Science & Business Media, 2012.

[130] A. Linke. Collision in a cross-shaped domain - A steady 2d Navier-Stokes example demonstrating the importance of mass conservation in CFD. *Computer Methods in Applied Mechanics and Engineering*, 198(41-44):3278–3286, 2009.

[131] A. Linke, L. G. Rebholz, and N. E. Wilson. On the convergence rate of grad-div stabilized Taylor-Hood to Scott-Vogelius solutions for incompressible flow problems. *Journal of Mathematical Analysis and Applications*, 381(2):612–626, 2011.

[132] J. Mandel. Balancing domain decomposition. *Communications in Numerical Methods in Engineering*, 9(3):233–241, 1993.

[133] N. N. Mansour and A. A. Wray. Decay of isotropic turbulence at low Reynolds number. *Physics of Fluids*, 6(2):808–814, 1994.

[134] A. Martín, S. Badia, and J. Principe. Multilevel balancing domain decomposition at extreme scales. *Submitted*, 2015.

[135] Y. P. Marx. Time integration schemes for the unsteady incompressible Navier-Stokes equations. *Journal of Computational Physics*, 112(1):182–209, 1994.

[136] A. Masud and R. Calderer. A variational multiscale method for incompressible turbulent flows: Bubble functions and fine scale fields. *Computer Methods in Applied Mechanics and Engineering*, 200(33–36):2577–2593, 2011.

[137] W. McCroskey. A critical assessment of wind tunnel results for the NACA 0012 airfoil. Technical report, DTIC Document, 1987.

[138] C. P. Mellen, J. Fröhlich, and W. Rodi. Lessons from LESFOIL project on large-eddy simulation of flow around an airfoil. *AIAA journal*, 41(4):573–581, 2003.

[139] P. Moin and J. Kim. On the numerical solution of time-dependent viscous incompressible fluid flows involving solid boundaries. *Journal of Computational Physics*, 35(3):381–392, 1980.

[140] R. D. Moser, J. Kim, and N. N. Mansour. Direct numerical simulation of turbulent channel flow up to Re[sub tau] = 590. *Physics of Fluids*, 11(4):943–945, 1999.

[141] A. Murua. Partitioned half-explicit Runge-Kutta methods for differential-algebraic systems of index 2. *Computing*, 59(1):43–61, 1997.

[142] N. Nikitin. Third-order-accurate semi-implicit Runge–Kutta scheme for incompressible Navier–Stokes equations. *International Journal for Numerical Methods in Fluids*, 51(2):221–233, 2006.

[143] J. Nitsche. Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, 36(1):9–15, 1971.

[144] X. Nogueira, L. Cueto-Felgueroso, I. Colominas, and H. Gómez. Implicit large eddy simulation of non-wall-bounded turbulent flows based on the multiscale properties of a high-order finite volume method. *Computer Methods in Applied Mechanics and Engineering*, 199(9–12):615–624, Jan. 2010.

[145] M. Olshanskii, G. Lube, T. Heister, and J. Löwe. Grad-div stabilization and subgrid pressure models for the incompressible Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 198(49-52):3975–3988, 2009.

[146] M. Olshanskii and A. Reusken. Grad-div stablilization for Stokes equations. *Mathematics of Computation*, 73(248):1699–1718, 2004.

[147] M. A. Olshanskii. A low order Galerkin finite element method for the Navier–Stokes equations of steady incompressible flow: a stabilization issue and iterative methods. *Computer Methods in Applied Mechanics and Engineering*, 191(47–48):5515–5536, 2002.

[148] P. Orlandi. *Fluid Flow Phenomena: A Numerical Toolkit*. Springer Science & Business Media, 2012.

[149] U. Piomelli and J. R. Chasnov. Large-Eddy Simulations: Theory and Applications. In M. Hallbäck, D. S. Henningson, A. V. Johansson, and P. H. Alfredsson, editors, *Turbulence and Transition Modelling*, number 2 in ERCOFTAC Series, pages 269–336. Springer Netherlands, 1996.

[150] S. B. Pope. *Turbulent Flows*. Cambridge University Press, 2000.

[151] J. Principe, R. Codina, and F. Henke. The dissipative structure of variational multiscale methods for incompressible flows. *Computer Methods in Applied Mechanics and Engineering*, 199(13–16):791–801, 2010.

[152] J. A. Rivera, B. E. Dansberry, R. M. Bennett, M. H. Durham, and W. A. Silva. *NACA0012 benchmark model experimental flutter results with unsteady pressure distributions*. National Aeronautics and Space Administration, Langley Research Center, 1992.

[153] R. S. Rogallo. *Numerical experiments in homogeneous turbulence*, volume 81315. National Aeronautics and Space Administration, 1981.

[154] L. Röhe and G. Lube. Analysis of a variational multiscale method for Large–Eddy simulation and its application to homogeneous isotropic turbulence. *Computer Methods in Applied Mechanics and Engineering*, 199(37–40):2331–2342, 2010.

[155] P. G. Saffman. The large-scale structure of homogeneous turbulence. *Journal of Fluid Mechanics*, 27(03):581–593, 1967.

[156] P. Sagaut. *Large Eddy Simulations for Incompressible Flows*, volume 3. Springer Berlin, 2000.

[157] P. Sagaut. *Large eddy simulation for incompressible flows: an introduction*. Springer Science & Business Media, 2006.

[158] B. Sanderse. Energy-conserving Runge–Kutta methods for the incompressible Navier–Stokes equations. *Journal of Computational Physics*, 233:100–131, 2013.

[159] B. Sanderse and B. Koren. Accuracy analysis of explicit Runge–Kutta methods applied to the incompressible Navier–Stokes equations. *Journal of Computational Physics*, 231(8):3041–3063, 2012.

[160] J. M. Sanz-Serna, J. G. Verwer, and W. H. Hundsdorfer. Convergence and order reduction of Runge-Kutta schemes applied to evolutionary problems in partial differential equations. *Numerische Mathematik*, 50(4):405–418, 1986.

[161] M. Schäfer, S. Turek, F. Durst, E. Krause, and R. Rannacher. Benchmark computations of laminar flow around a cylinder. In P. D. E. H. Hirschel, editor, *Flow Simulation with High-Performance Computers II*, number 48 in Notes on Numerical Fluid Mechanics (NNFM), pages 547–566. Vieweg+Teubner Verlag, 1996.

[162] O. Schenk and K. Gärtner. Solving unsymmetric sparse systems of linear equations with PARDISO. *Future Generation Computer Systems*, 20(3):475 – 487, 2004.

[163] O. Schenk and K. Gärtner. On fast factorization pivoting methods for sparse symmetric indefinite systems. *Electronic Transactions on Numerical Analysis*, 23:158–179, 2006.

[164] S. Schmidt, M. Franke, and F. Thiele. Assessment of SGS models in LES applied to a NACA 4412 airfoil, 2001.

[165] L. R. Scott and M. Vogelius. Conforming finite element methods for incompressible and nearly incompressible continua. *Lectures in Applied Mathematics*, 22:221–244, 1985.

[166] R. E. Sheldahl and P. C. Klimas. Aerodynamic Characteristics of Seven Symmetrical Airfoil Sections Through 180-Degree Angle of Attack for Use in Aerodynamic Analysis of Vertical Axis Wind Turbines. Technical Report SAND-80-2114, Sandia National Labs., Albuquerque, NM (USA), 1981.

[167] S. Smith, M. Guynn, C. Streett, and G. Beeler. Mars Airplane Airfoil Design with Application to ARES. In *2nd AIAA "Unmanned Unlimited" Conf. and Workshop & Exhibit*. American Institute of Aeronautics and Astronautics, 2003.

[168] P. Spalart, W. Jou, M. Strelets, and S. Allmaras. Comments on the feasibility of les for wings, and on a hybrid rans/les approach. *Advances in DNS/LES*, 1:4–8, 1997.

[169] P. R. Spalart. Strategies for turbulence modelling and simulations. *International Journal of Heat and Fluid Flow*, 21(3):252–263, 2000.

[170] R. Stenberg. On some techniques for approximating boundary conditions in the finite element method. *Journal of Computational and Applied Mathematics*, 63(1–3):139–148, 1995.

[171] E. Sterner. Semi-implicit Runge-Kutta schemes for the Navier-Stokes equations. *BIT Numerical Mathematics*, 37(1):164–178, 1997.

[172] M. Svärd and J. Nordström. Review of summation–by–parts schemes for initial–boundary-value problems. *Journal of Computational Physics*, 268:17–38, July 2014.

[173] G. Söderlind. Automatic Control and Adaptive Time-Stepping. *Numerical Algorithms*, 31(1-4):281–310, 2002.

[174] G. I. Taylor and A. E. Green. Mechanism of the Production of Small Eddies from Large Ones. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 158(895):499–521, 1937.

[175] R. Temam. *Navier Stokes equations*, volume 2 of *Studies in mathematics and its applications*. Elsevier Science, The Netherlands, 3th edition, 1984.

[176] H. Tennekes and J. L. Lumley. *A First Course in Turbulence*. MIT Press, 1972.

[177] F. X. Trias, O. Lehmkuhl, A. Oliva, C. D. Pérez-Segarra, and R. W. C. P. Verstappen. Symmetry-preserving discretization of Navier–Stokes equations on collocated unstructured grids. *Journal of Computational Physics*, 258:246–267, Feb. 2014.

[178] A. Veneziani and U. Villa. ALADINS: An ALgebraic splitting time ADaptive solver for the incompressible Navier-Stokes equations. *Journal of Computational Physics*, 238:359–375, 2013.

[179] R. W. C. P. Verstappen and A. E. P. Veldman. Symmetry-preserving discretization of turbulent flow. *Journal of Computational Physics*, 187(1):343–368, 2003.

[180] J. G. Verwer. Convergence and order reduction of diagonally implicit Runge-Kutta schemes in the method of lines. In *Numerical Analysis: Proceedings of the Dundee Conference on Numerical Analysis, 1975*, volume 140 of *Pitman Research Notes in Mathematics*, pages 220–237, 1986.

[181] A. W. Vreman and J. G. M. Kuerten. Comparison of direct numerical simulation databases of turbulent channel flow at $Re_\tau = 180$. *Physics of Fluids (1994-present)*, 26(1):015102, 2014.

[182] A. J. Wadcock. Investigation of low-speed turbulent separated flow around airfoils. Technical report, 1987.

[183] M. Yano and D. L. Darmofal. BDDC preconditioning for high-order Galerkin Least-Squares methods using inexact solvers. *Computer Methods in Applied Mechanics and Engineering*, 199(45–48):2958–2969, 2010.

[184] S. Zhang. A family of $Q_{k+1,k} \times Q_{k,k+1}$ divergence-free finite elements on rectangular grids. *SIAM Journal on Numerical Analysis*, 47(3):2090–2107, 2009.