UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

Modelling and numerical simulation of combustion and multi-phase flows using finite volume methods on unstructured meshes

Jordi Muela Castro

UPC CTTC

Modelling and numerical simulation of combustion and multi-phase flows using finite volume methods on unstructured meshes

Centre Tecnològic de Transferència de Calor
Departament de Màquines i Motors Tèrmics
Universitat Politècnica de Catalunya

Jordi Muela Castro

Doctoral Thesis

Modelling and numerical simulation of combustion and multi-phase flows using finite volume methods on unstructured meshes

Jordi Muela Castro

TESI DOCTORAL

presentada al

Departament de Màquines i Motors Tèrmics
E.S.E.I.A.A.T.
Universitat Politècnica de Catalunya (UPC)

per a l'obtenció del grau de

Doctor per la Universitat Politècnica de Catalunya

Terrassa, Febrer, 2018

Modelling and numerical simulation of combustion and multi-phase flows using finite volume methods on unstructured meshes

Jordi Muela Castro

Directors de la Tesi

Dr. Assensi Oliva Llena
Dr. Carlos-David Pérez-Segarra

Tribunal Qualificador

Dr. Antonio Lecuona Neumann
Universidad Carlos III de Madrid
Dr. Jesús Castro González
Universitat Politècnica de Catalunya
Dr. José Ignacio Nogueira Goriba
Universidad Carlos III de Madrid

Dedicat a totes les persones que estimo. Gràcies per voler formar part de la meva vida i fer-me feliç.

"Man, I see in fight club the strongest and smartest men who've ever lived. I see all this potential, and I see squandering. God damn it, an entire generation pumping gas, waiting tables; slaves with white collars. Advertising has us chasing cars and clothes, working jobs we hate so we can buy shit we don't need. We're the middle children of history, man. No purpose or place. We have no Great War. No Great Depression. Our Great War's a spiritual war... our Great Depression is our lives. We've all been raised on television to believe that one day we'd all be millionaires, and movie gods, and rock stars. But we won't. And we're slowly learning that fact. And we're very, very pissed off."

"You are not your job, you're not how much money you have in the bank. You are not the car you drive. You're not the contents of your wallet. You are not your fucking khakis. You are all singing, all dancing crap of the world."

Tyler Durden, *Fight Club.*

Agraïments

Es curiós com la tasca que se suposa hauria de ser la més senzilla i un mer tràmit, està resultant ser la més feixuga i pesada de tota la tesi. Plasmar en aquesta memòria tota la feina realitzada al llarg dels darrers anys no es trivial i necessita moltes hores, aprendre a superar el síndrome de la pàgina en blanc, així com prendre decisions que encara que puguin semblar irrellevants tindran el seu impacte en el resultat final, e inevitablement et generen dubtes i cavil·lacions. Crec que personalment el més complicat es trobar la motivació durant la redacció de la tesi, ja que el motiu e impuls per redactar, plasmar resultats i maquetar és el resultat final, i no el procés en si. Per contra, tota la feina prèvia consistent en aprendre coses noves, implementar nous codis, desenvolupar nous models, provar coses noves així com mirar de millorar-ne ja existents, representa per mi una motivació continua, una gasolina diària que et dona energia. Suposo que en gran part es deu a la meva forma de ser e inquietuds. Però se que acabar aquesta memòria i defensar la tesi farà que pugui mirar enrere i dir-me "tot això ha valgut la pena". Tot i així, se que la presentació d'aquesta tesi no és cap fita en si, és simplement una estació més en el trajecte, un punt més en la línia (recta o amb corbes) que és la vida.

I per sort, aquest camí no l'he fet sol, sinó que l'he fet acompanyat de molta gent molt valuosa que en major o menor grau, i de forma més directa o indirecta, ha tingut una incidència o rellevància en el procés d'elaboració d'aquesta tesi doctoral. Per tant, m'agradaria dedicar unes quantes línies d'aquesta tesis per agrair-los tota l'ajuda i suport rebut.

Crec que és just començar aquests agraïments pels companys de feina que més m'han ajudat durant el desenvolupament d'aquesta tesi doctoral: l'Oriol Lehmkuhl, que puc dir, ben orgullós, que ha sigut el meu mentor en aquest món del CFD, dels codis numèrics i de l'investigació. I el Jordi Ventosa, que ha sigut la persona amb la que probablement he tingut més xerrades sobre temes de la tesi. Va ser un bon guia per introduir-me dins del codi TermoFluids, un mestre en la temàtica de la combustió, un gran company amb el que discutir (en l'accepció positiva de la paraula) sobre models, codis, problemes, cabòries, etcètera. Però per damunt de tot ha sigut un bon company de pis i un gran amic.

També vull agrair a l'Asensi Oliva i al Carles David Pérez-Segarra la seva guia en aquest treball i la confiança mostrada en la feina feta. I als professors Quim Rigola i Carles Oliet la seva ajuda. Una petita menció especial per a l'informàtic del *lab*, l'Octavi, per solucionar-me petits problemes i entrebancs informàtics durant la feina, però sobretot per tots els cafès fets junts, les pujades a La Mola i els moments compartits fora de la feina. I òbviament a tots aquells companys, però sobretot amics del laboratori, que han format part de la meva vida durant aquests anys: el F.X. Trias, el Lluís, el Chiva, el Fede, el Kike, l'Aleix, l'Edu, el Chamo, l'Eugenio... i un llarg etcètera. Em sap greu no poder mencionar explícitament a tothom, però de forma implícita us tinc presents i us estic igual d'agraït. Gràcies.

A continuació encetaré el paràgraf més especial d'aquests agraïments: vull donar infinites gràcies a la meva família. Al meu pare Antonio i la meva mare Paquita, que tot i que no entenen en que consisteix això de *"el doctorado ese que estás haciendo"*, i molts cops sembla que vivim en realitats paral·leles, els hi dec tot i els estimo moltíssim. A me germana Yolanda, que ha sigut, és i serà un pilar bàsic en la meva vida, i a la que crec que no li demostro prou lo important que és per a mi. I als meus nebots Eudald i Emma, una dosi continua d'alegria i felicitat. Feu que al rellegir aquestes línies per comprovar si la sintaxi i la gramàtica és correcta, em vingui un somriure i una llagrimeta de felicitat. Sóc molt afortunat de que formeu part de la meva vida. Moltes gràcies.

El següent paràgraf d'agraïments els vull dedicar a l'*altre* família, a la que tries: els amics. Sou molts i a cada qual més collonut. Gràcies als *"EQPeros"*, els amics de tota la vida i amb els que he viscut mil i una aventures e histories. Gracies als *"C&G"*, el grup d'amics que vaig començar a gestar al batxillerat amb els que he compartit molts viatges i moments. Gràcies als *"Charros"*, unes persones que trobo meravelloses e increïbles, i dels quals em sento profundament afortunat que formin part de la meva vida. Gràcies als grans companys que he fet en el triatló durant aquest anys de la meva vida, per permetre'm gaudir de la meva passió amb grans persones com vosaltres. Gràcies als tres magnífics amics (sabeu qui sou) que vaig tenir la sort de fer durant la carrera. I per acabar donar les gràcies al Pablo i al Paco, aquells dos amics que potser no veus tan sovint com als altres, però que són els que han estat amb tu a la vida des de parvulari, i que es igual que els vegis dos o quatre cops a l'any, saps que sempre pots comptar amb ells.

Segur que em deixo gent, però si llegeixes aquestes línies i et trobes a faltar no t'emprenyis, creu-me que t'estic tenint present no només ara que escric aquestes línies, sinó en molts moments de la meva vida. Al cap i la fi, la vida es un camí que fas amb molta gent, alguns t'acompanyen més tros i d'altres menys, alguns des del principi fins al final i d'altres molt poqueta estona, però tothom amb el que tens la sort de compartir un trosset del camí deixa una petjada indeleble al costat de la teva. I si la teva petjada està al costat de la meva: gràcies.

En resum: moltes gràcies a tots. Gràcies per formar part de tot això. Us estimo.

Jordi Muela Castro

Abstract

The present thesis is devoted to the development and implementation of mathematical models and numerical methods in order to carry out computational simulations of complex heat and mass transfer phenomena. Several areas and topics in the field of Computational Fluid Dynamics (CFD) have been treated and covered during the development of the current thesis, specially combustion and dispersed multi-phase flows. This type of simulations requires the implementation and coupling of different physics. The numerical simulation of multiphysics phenomena is challenging due to the wide range of spatial and temporal scales which can characterize each one of the physics involved in the problem. Moreover, when solving turbulent flows, turbulence itself is a very complex physical phenomenon that can demand a huge computational effort. Hence, in order to make turbulent flow simulations computationally affordable, the turbulence should be modelled. Therefore, throughout this thesis different numerical methods and algorithms have been developed and implemented aiming to perform multiphysics simulations in turbulent flows.

The first topic addressed is turbulent combustion. Chapter 2 begins with an introduction to the mathematical formulation required to describe the flow transport and the chemical reactions involved in a combustion process. As will be shown, the system of equations describing thermochemical reactions is stiff, which makes its calculation difficult and computationally expensive. Hence, it is interesting to use special time integration methods or some modelling approaches aimed to reduce its computational cost. Moreover, when dealing with turbulent flows in LES modelling, the sub-grid chemistry-flow interaction must be modelled.

The second chapter also presents a combustion model able to notably reduce the computational cost of the simulation. The model, namely the Progress-Variable (PV) model, relies on a separation of the spatio-temporal scales between the flow and the chemistry. This assumption allows to pre-compute and store the chemical reactions as function of a reduced system of tracking scalars. During simulation time only these tracking scalars are transported, and the thermo-chemical properties of the mixture are recovered from the look-up table previously generated. Moreover, in order to account for the influence of the sub-grid species concentrations and energy fluctuations, the PV model is coupled to the Presumed Conditional Moment (PCM) model. The PV-PCM combustion model is then employed to simulate the auto-ignition process of a hydrogen flame. The model shows its capability to reproduce the results of the experiment with a relatively low computational effort.

Finally, Chapter 2 shows the development of a smart load-balancing method for the evaluation of chemical reaction rates in parallel combustion simulations. The presented algorithm employs a special implicit integration method well-suited for stiff equation systems, namely Gear's method. Gear's method is only employed to inte-

grate the chemical reactions in the reactive zones of the domain, close to the front flame, while the rest is integrated using a less-computationally demanding explicit time-integration scheme. This strategy creates a dynamic imbalance in parallel simulations, originating a bottle-neck in the calculations with some processors having to wait others with a greater computational load. In order to solve this issue, the implemented algorithm rebalances the overall computational load of the simulation between all the CPUs involved in the parallel computation. This load-balancing method is used to reproduce the previously presented hydrogen flame case, showing its capability to speed-up the parallel simulations and reduce the total simulation time.

Chapter 3 is devoted to dispersed multiphase flows. This type of flows are composed of a continuous phase and a dispersed phase in the form of unconnected particles or droplets. Several modelling approaches can be employed in order to simulate dispersed multiphase flows, each one best-suited for a certain type of flow configuration. In this thesis, the Eulergian-Lagrangian approach has been selected. This type of model is the best-suited for dispersed multiphase flows with thousands or millions of particles, and with a flow regime ranging from the very dilute up to relatively dense. Consequently, the Eulerian-Lagrangian model can be used to simulate different applications, as for example the fuel injection of combustion chambers, cyclone separators, evaporative cooling, dispersion of pollutants, deposition of inhaled medicine in the human airways, etc.

The chapter begins with a presentation of the mathematical models required to implement the Eulerian-Lagrangian approach. Several key aspects regarding the computational implementation of the models are further analysed and discussed. After that, a rigorous analysis of the influence of the sub-grid scales in the motion of the Lagrangian particles in the framework of LES modelling is presented. As part of this study, two widely employed stochastic models for sub-grid dispersion of Lagrangian particles available in the literature are analysed and assessed. Moreover, seeing the shortcomings of these models, a new model for sub-grid dispersion of heavy particles is proposed. This new model is not based on Langevin type equations as the previous ones, but in the reconstruction of statistics via Probability Density Functions (PDFs). The preliminary results obtained with this model are very promising since it is able to recover the influence of the sub-grid scales over the Lagrangian particles quite well.

The chapter finishes with the numerical simulation of two experiments employing the numerical libraries implemented throughout the chapter. The first reproduced case is a coflowing jet which creates an axisymmetric confined bluff body flow. Solid particles of different sizes are injected through the inner pipe. The case reproduces a typical industrial configuration aimed to control the mixing of fuel with the air. The second studied case is the regional deposition of inhaled medicines in an *in vitro* experiment which reproduces the upper airways of a human-based model. The results obtained from both cases show the capabilities of the implemented algorithms and

validate the numerical code.

In Chapter 4, a new method capable of performing parallel numerical simulations using non-overlapping disconnected mesh domains with adjacent boundaries is presented. This methodology is designed for applications like wind-turbines, tidal-stream turbines or compressors, characterized by having a rotational part which moves close to other static solid parts. The presented algorithm *stitches* at each iteration independent meshes and solves them as a unique domain. First, a rigorous analysis of the conservation properties of the algorithm is presented, as well as a study of its parallel scalability. Once done, the algorithm is employed in a real test case demonstrating that the method is capable of obtaining reliable numerical results, as the simulation results match the experimental ones. Closing the chapter, an enhanced version of the previous algorithm still under development is introduced. The first studies show that the new method conserves mass and kinetic energy almost as well as a case simulated in a full static mesh.

Finally, Chapter 5 addresses a transversal aspect to the previously covered topics throughout the thesis. In this chapter, a self-adaptive strategy for the maximisation of the time-step for the numerical solution of convection-diffusion equations is discussed. The method is capable of determining dynamically at each iteration which is the maximum allowable time-step which assures a stable time integration. Moreover, the method also smartly modifies the temporal integration scheme in order to maximize its stability region depending on the properties of the system matrix. The chapter begins with a detailed explanation of the mathematical basis in which the method relies, and follows detailing how the methodology applies to the temporal integration of convection-diffusion equations. The method is numerically assessed and verified for the discretization schemes and the temporal integration schemes previously detailed. The obtained results show that the method is capable of dynamically determining the optimal time-step which assures a stable temporal integration, allowing to reduce significantly the computational cost of the simulations.

Contents

1

Introduction

1.1 Prologue

Human beings are curious by nature. They try to understand and explain all the events which happen around them. Many of these events are phenomena of heat and mass transfer. These kind of physical events take place all the time throughout all the universe from its creation. Hence, there is no doubt of the interest of the human being to understand and predict this type of phenomena. Some of these events, like the motion of cigarette smoke, seem to have a totally random behaviour that can not be predicted or calculated. However, to this day it is known that all these physical phenomena are ruled by natural laws which can be described by mathematical equations, and therefore, can be calculated and predicted. Claude-Louis Navier in France and George Gabriel Stokes in England, derived independently in the early 1800's the set of equations known as the **Navier-Stokes (NS) equations**, which describe the behaviour of Newtonian viscous fluid substances [1]. These equations are a set of partial differential equations (PDEs) which are non-linear and highly coupled. Unfortunately, except for some very simple cases, this set of equations has no analytical solution. In the past, mathematicians, physicians and engineers have employed some approximations and simplifications in order to obtain some solutions. But the necessity of obtaining precise and reliable predictions of heat and mass transfer phenomena motivated the birth of a new area of study known as **Computational Fluid Dynamics** (CFD). CFD is a branch of fluid dynamics which uses numerical methods and algorithms employing computers in order to solve fluid flow problems. Different numerical methods, e.g. finite differences, finite volumes, finite elements and spectral methods, among others, have been proposed and employed, but all of them pursuing the overall same objective, the resolution of complex coupled differential equations.

Historically, it can be said that the seed of this analysis method was planted by Richardson in 1922, when he published the book *"Weather prediction by numerical process"* with a method aimed to predict the weather forecast [2].

But the real birth of CFD came together with the advent of computers. And since then, CFD has grown together with the calculus capacity of the computers. In the forties J. von Neumann and J. Charney at Princeton started to use the computer ENIAC aiming to predict weather forecasts employing a similar methodoly to the one set out by Richardson in his book [3]. In 1950, they successfully carried out the first 24-hour weather forecast. It took 24 hours to compute [4]. During the fifties and sixties, Francis H. Harlow and his Group T-3 in Los Alamos National Laboratory developed several techniques, e.g. the Particles-in-Cell (PIC) method, Marker-and-Cell (MAC) method and ICE method after the arrival of the first large computer at Los Alamos Scientific Laboratories [5]. The first study using a 3D model was published in 1967, where a Douglas Aircraft was simulated [6]. And so on up to the present time, where simulations of multi-physics phenomena are routinely carried out using thousands of CPUs [7, 8]. However, there are still lot of challenges to be faced and barriers to be overcome in CFD.

One of the main challenges is turbulence. Richard Feynman, laureate with the Nobel Prize in Physics, defined turbulence as: *"the most important unsolved problem of classical physics"*. Turbulence is an underlying phenomenon present in flows with high Reynolds numbers, which are characterized by an apparent random behaviour and three-dimensional instabilities. The source of this apparently chaotic behaviour is the non-linearity of the convection process. However, a complete theoretical understanding of the solution of Navier-Stokes equations still lacks. The existence of a smooth solution of the Navier-Stokes equations under a prescribed initial conditionls in a three-dimensional system has been not proven yet. As a fully understanding of the NS equations is a key step to comprehend turbulence, the Clay Mathematics Institute considers it as one of the seven *Millenium Problems* [9].

Turbulent flows are characterized by being transient, three-dimensional and by the presence of a wide range of length-scales, from the bigger ones, defined by the domain which contains the flow, up to the smallest ones, also called the Kolmogorov length-scales [10], where viscosity dominates and the kinetic energy is dissipated into internal thermal energy. This wide-range of scales makes unaffordable computational simulations solving all the temporal and spatial scales present in many turbulent flows. This type of simulations, where all the scales are solved, are known as **Direct Numerical Simulations** (DNS). DNS are really useful in order to understand the NS equations and the turbulence, but with the computational power available nowadays, these simulations are extremely expensive (both in time and economic resources) for the majority of industrial and commercial problems, and are mainly restricted to research applications. Therefore, modelling techniques are required to reduce the degrees of freedom of turbulent cases and allow affordable simulations of fluid-dynamic problems. The oldest approach for turbulence modelling is the **Reynolds-averaged Navier–Stokes** (RANS) equations. This mathematical model was first proposed by O. Reynolds [11], and it is based on the decomposition technique, where the instantaneous flow variables are split into two components, the mean values plus the fluctuating ones (i.e. for a generic variable ϕ, $\phi = \overline{\phi} + \phi'$). Another widely employed modelling technique is called **Large Eddy Simulation** (LES). This technique was proposed by J. Smagorinsky in 1963 [12], and is based on the Kolmogorov theory where for homogeneous turbulence small scale eddies are independent of geometry problem and are statistically isotropic. This hypothesis supports the main idea behind LES: only the large scales of the problem are calculated while the small scales are not solved, and their effect must be modelled. This separation of scales is obtained after convolution of NS equations with a filtering kernel G [13].

Another motivating challenge to be faced by CFD is how to exploit and take profit of the computational resources available nowadays. Until the beginning of this century, the processors have followed the well-known Moore law. This law, formulated by Gordon Moore in 1965, and later revised in 1975, states that the number of transistors on an integrated circuit will double approximately every two years [14]. However, as can be seen in Fig. 1.1, although in the last decade this law can be considered still true, this has been achieved by increasing the number of cores per chip, but not by improving the clock speed of the processors, which seems to be somewhat stalled [15]. Therefore, it is clear that algorithms and numerical codes for CFD should be designed, coded adapted to this reality. This means that CFD codes should be able to run as fast as possible in parallel using hundreds and thousands of CPUs without losing efficiency.

Besides, the future (but also the present) of Computational Fluid Dynamics goes through the integration of multiple physics in the same code. There is the necessity of solving fluid-dynamic problems where other physics like combustion, radiation, heat-transfer, phase-change or fluid-structure interaction are also present. These additional physics add complexity to the simulations. Therefore, new models and algorithms should be developed in order to solve advanced multi-physics simulations.

Thus, there is still a hard and long way to go in Computational Fluid Dynamics. But this path is really interesting and motivating, and this thesis wants to walk over this way and be part of the journey. Summarizing, this thesis pretends to be a small step forward in the field of CFD, aiming to increase the knowledge, know-how and understanding of heat and mass transfer phenomena.

1.2 Background at the CTTC

To fully understand and explain all the work carried out during this thesis, it is necessary to first set the reference framework where it has been developed. All this work has been done within the PhD programme **Enginyeria Tèrmica** conducted by the *Centre Tecnològic de Transferència de Calor* (CTTC) at the *Universitat Politècnica de Catalunya - BarcelonaTech* (UPC). The following historical review is by no means

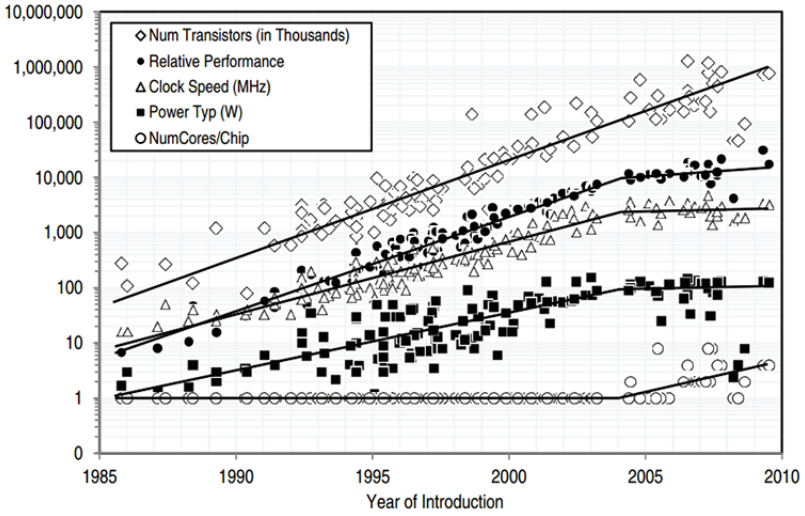


Figure 1.1: Number of transistors, performance, clock speed, power and cores per chip, graphed over time [15].

exhaustive, but intends to give a general overview of all the work and research accomplished during the last three decades by the Research Group where the present thesis has been developed.

The first efforts of the CTTC group in this area dates back to the early eighties, when A. Oliva proposed a numerical method for the resolution of heat transfer by conduction and convection in a combustion chamber [16]. Following, C.D. Pérez-Segarra contributed with a deep analysis and study about numerical aspects and criteria for numerical methods applied to CFD, work carried out during his thesis, devoted to obtaining velocity, pressure and temperature distributions in compressible flows under steady-state conditions [17].

As explained previously, the present and future of Computational Fluid Dynamics goes through parallel computing. In this context, M. Soria carried out the work of his thesis, aimed on the development of numerical methods for heat transfer and fluid flows computational simulations using low-cost parallel computers [18]. Following this research line, F.X. Trias contributed to the development of numerical algorithms for Direct Numerical Simulation (DNS) and regularization modelling of turbulent flows, employed in LES, on loosely coupled parallel computers [19]. Based on the previous work by R. Verstappen and A. Veldman [20], a kinetic-energy preserving discretization scheme based on symmetry-preserving operators was employed.

Thanks to all these mathematical and numerical backgrounds acquired by the Group in computational heat and mass transfer, the CTTC developed a general flow modelling software, known as **DPC** [21]. This numerical code was written in C and designed for RANS modelling. The DPC code was successfully employed during several years for fundamental research and investigation, as well as for industrial design and optimization. The thesis by I. Rodriguez was carried out in the framework of DPC. Her thesis was focused on the numerical simulation of unsteady laminar convection in cylindrical domains, and its application to the study of heat transfer and fluid flow which take place in stratified storage tanks [22]. DPC code was also employed by J. Jaramillo to carry out his thesis, where firstly a fundamental analysis of turbulence RANS models was carried out, and then the RANS models previously studied were employed for simulations of air curtains [23]. R. Cònsul developed the required algorithms and numerical models for the computation of combustion processes for DPC [24]. Following this research line, K. Claramunt extended the DPC combustion framework implementing the required tools to simulate both laminar and turbulent diffusion flames by means of Flamelet Modelling [25]. Furthermore, D. Carbonell extended this previous work on Flamelet models for laminar and turbulent combustion, with special emphasis on pollutant formation, i.e. soot and NO_x [26].

In spite of the successful fundamental research and industrial design carried out with the DPC code, it had some structural limitations which made its growth and extension difficult. Therefore, by the late 2000s the CTTC group started the development of a new CFD tool. As mentioned above, DPC was written in C code. Thus, it did not follow the programming paradigm of Object-oriented programming (OOP). Hence, a new numerical CFD code named **TermoFluids** (TF) [27] was created and developed from scratch in C++ language. This new code was mainly developed during the work carried out in the theses by O. Lehmkuhl [28] and R. Borrell [29]. TermoFluids is a multi-physics CFD code programmed in C++ language using the object-oriented programming paradigm designed for parallel High Performance Computing (HPC) applications. The parallelization is achieved using the Message Passing Interface (MPI) library. This guarantees a high portability of the code in all kind of HPC architectures (shared and distributed memory). Its parallel design has demonstrated an excellent performance and a very high scalability, up to peta-scale simulations [30]. The code is based on finite-volume discretization, and is devised to work in both structured and unstructured meshes. Based on the previous experience of the Group, TermoFluids employs kinetic-energy preserving discretization schemes. The aforementioned theses by O. Lehmkuhl and R. Borrell are focused on the solution of turbulent incompressible flows in complex geometries using unstructured meshes, employing both DNS and LES modelling. It is demonstrated that the conservation of kinetic energy is a key aspect to obtain accurate results in LES and DNS simulations. This kinetic energy conservation is assured when the symmetry of the discrete operators is preserved.

In recent years, the TermoFluids code has been extended to multi-physics modelling in the context of DNS and LES modelling. For example, in the field of multiphase flows, L. Jofre developed and implemented the basis for the numerical simulation of multi-phase flows of immiscible fluids using the Volume-of-Fluid (VoF) method [31]. N. Balcázar developed numerical methods based on level-set techniques suitable for the direct numerical simulation of free surface and inter-facial flows [32]. Besides, in the field of combustion, J. Ventosa developed numerical methods and algorithms for the efficient simulation of diffusion flames in the flamelet regime [33]. Fluid-Structure Interaction (FSI) phenomenology, dynamic mesh refinement [34] and dynamic meshes [35] capabilities have also been developed and included in TermoFluids.

The work of the present thesis has been carried out based on the experience and know-how of the CTTC-Group, as well as taking advantage of all the numerical infrastructure already developed, implemented and tested. Therefore, all the numerical tools and algorithms developed in this thesis have been implemented within the framework of TermoFluids code.

1.3 Objectives of the thesis

As stated previously in Section 1.1, the general purpose of this thesis is to contribute to the research in the field of heat and mass transfer phenomena.

As aforementioned, these phenomena are described by partial differential equations which do not have analytical solution for the vast majority of cases. Therefore, scientists and engineers have to relay on approximate solutions obtained by two different approaches. The first one is based on applying some simplifications and assumptions in the real problem case in order to derive a simplified version of the problem of interest, which has an analytical solution. These simplified models commonly require to employ empirical correlations and expressions which are only valid for a small range of conditions and configurations. Therefore, although these models can offer accurate results, their lack of generality restrict their application to a small range of conditions where the employed empirical inputs are valid. Out of the applicability range, these models cannot produce reliable results. On the other hand, the second approach is based on the solution of the PDEs system by means of numerical methods, which allow to transform the PDEs into a set of algebraic equations. This set of algebraic equations for the dependent variables is obtained after the application of a discretization process over the governing equations. Therefore, the unknowns of the algebraic equation system are the values of the dependent variables at a finite (discrete) number of locations in the domain of analysis. The most popular discretization techniques are the Finite Diffence Method (FDM), Finite Elements Method and Finite Volume Method (FVM) [36]. Among others, the Spectral Methods also exist, which are closely related with FEM [37], or the Meshfree and particle methods [38], which are not based on a mesh with connected nodes, but on the interaction of the nodes with their neighbours. Obviously, each one of these techniques has its own advantages and disadvantages, but its discussion is out of the scope of this brief introduction.

Turning to the issue of the two methods used to obtaining solutions for the PDEs describing the physics behind heat and mass transfer phenomena, the former can offer good results at a low computational cost. Nonetheless, these simplified models with empirical correlations do not offer generality, and each model is restricted to a very specific geometry and conditions where only a small number of parameters characterizing the case can be modified. Another drawback of these models is that although they can be very useful for designing and optimization processes, they do not offer detailed results of the physical phenomena occurring in the studied case. Therefore, they are not useful for studying and analysing the details of the basic physics taking place in the modelled systems. In order to circumvent these limitations, the research and study of heat and mass transfer phenomena must rely on numerical techniques for the solution of PDEs. As aforementioned, the branch of fluid dynamics focused on the study of heat and mass transfer problems using numerical analysis is the Computational Fluid Dynamics (CFD). This modelling approach offers generality, and the same numerical tool can be employed in a very wide range of applications. Moreover, these models can offer very accurate details of all the physical phenomena occurring at all the scales of the studied case. In fact, they can be even better than experiments to understand the basic physics of heat and mass transfer phenomena. These models offer results throughout all the studied domain, without having to place sensors or probes that sometimes are almost impossible to place at the location of interest. Furthermore, their presence can greatly modify the physics of the studied case. Thus, CFD is a very powerful and useful tool for both designing and optimizing of industrial and commercial problems, as well as for the study of the basic physic phenomena of heat and mass transfer problems.

Hence, in order to contribute to the advance and progress of knowledge in heat and mass transfer physics, the work of this thesis has been focused on the development and implementation of numerical methods and algorithms for a CFD code, specifically the in-house TermoFluids code. The work carried out can be split into two main areas, although both are intimately connected:

- The development and implementation of new models and algorithms in order to add new features to the TermoFluids code, increasing its capacity to solve new physics and problems.

- The enhancement of the performance of the computational code from two perspectives: a first one improving the algorithms increasing the computation speed using smarter parallel strategies, reducing caches misses, speeding-up the algorithms, etc., and a second one facing the problem mathematically, aiming to increase the time-step of the simulations, improving the convergence of the algorithms, etc.

Although the general objectives are clear, more specific targets should be set. As aforementioned in Section 1.2, the CTTC-Group develops and maintains the CFD code TermoFluids. In the line of continuously improving the code and adding new features to it, one of the mid-term objectives of the Group is to have the capabilities of performing complete simulations of combustion chambers and propulsion systems. These simulations are complex as they include many physical phenomena, i.e. mass transfer, chemical reactions, disperse multiphase flows, heating and evaporation of droplets, etc. Furthermore, all these different physics involve a wide range of temporal and length scales, adding a considerable difficulty to the modelling of the thermal system. Hence, several challenges have to be faced and short-term intermediate goals have to be set before achieve this ambitious mid-term goal. The steps and short-term goals set during the development of this thesis are detailed and justified.

Obviously, a basic requirement to achieve the mid-term goal of having the capability of simulating combustion chambers and propulsion systems is to simulate combustion processes. A combustion process is an exothermic chemical reaction taking place within a fluid flow. Therefore, it is required to describe the dynamics of the flow as well as the kinetics of the chemical process. Starting from the previous work in combustion carried out by the CTTC-Group, mainly presented in the thesis by J. Ventosa [33], two main goals were set in this field: the development and implementation of a combustion model well-suited for ignition events, and the implementation of a new efficient numerical method for the implicit integration of the chemical reaction rates when using finite-rates chemistry. As will be further detailed in Chapter 2, the wide span of length and temporal scales in the chemical kinetics processes is reflected in the equations describing the chemical kinetics in the high non-linear dependence on the mass fractions and temperatures of the system. Therefore, the resulting system of equations is stiff [39], which means a very small time-step must be employed to integrate numerically the system of equations in order to avoid numerical instabilities. Small variations in some terms of these stiff equations can lead to very fast changes in the solution. Hence, special integration methods must be employed in order to achieve bigger time-steps. Therefore, in the context of this thesis Gear's method [40] is used for the solution of the mass reaction rates. Moreover, the method has been implemented in an efficient numerical way using a smart load-balancing algorithm to distribute the computational load between all the processors in parallel simulations.

Once the targets in the combustion field were achieved, the following goal was the development of all the numerical infrastructure required to simulate dispersed multiphase flows. This type of flows are characterized by the presence of two (or more) phases, where a continuous phase and dispersed phases formed by a large amount of small particles or droplets coexist. Among the different methods available in the literature to model dispersed multi-phase flows, the Lagrangian-Eulerian method has been chosen and implemented, since is the methodology most well-suited for the type of multi-phase flows aimed to be simulated. In this kind of implementation the dispersed phase is modelled as particles with a size much smaller than the Eulerian computational mesh where the continuous phase is calculated. Hence, the interchanges of mass, momentum and energy between both continuous and dispersed phase requires modelling, inasmuch as this interaction occurs at sub-grid scale. Therefore, all the numerical algorithms for the tracking and interpolation of the Lagrangian particles inside the Eulerian mesh, as well as all the models required to calculate the interaction between both phases, have been developed and implemented for the CFD code TermoFluids.

In parallel to the development of all the numerical infrastructure devoted to dispersed multi-phase flows, in the CTTC-Group the necessity of performing 3D full-scale simulations of turbo-machinery and wind turbines arose. After several attempts of different methodologies which were not successful, further detailed in the thesis of D. Martinez [41], it was decided to develop a new method based on sliding meshes and the Arbitrary Lagrangian-Eulerian (ALE) method. The basic idea was to have two (or more) meshes: one static mesh containing all the fixed elements like the tower or the nacelle in wind turbines, and another moving (dynamic) mesh containing all the rotating elements, like the rotor blades. The link between this methodology and the dispersed multi-phase flows was the idea of employing the algorithms and numerical tools developed for the tracking of the particles to perform the stitching between the two meshes (the static and the dynamic ones) at each iteration. The details of the method and its development are presented in Chapter 4.

Finally, a transversal, but also very important objective set during the development of this thesis, was to enhance and maximize the time-step of the simulations. Increasing as much as possible the time-step taking into account the physics of the problem is a very efficient way to reduce the computational cost of the simulations, as well as to make more attractive and useful the CFD simulations for designing and optimization purposes. Improvements in this area are many times neglected by researchers, and the time-step determination is obtained using the classical Courant–Friedrichs–Lewy condition with conservative security factors. Since the determination of the time-step and efficiency of the numerical integration was a relevant and crucial aspect during the development of the algorithms for combustion processes, it was decided to continue and extend the previous work developed in this topic at the CTTC-Group by Trias and Lehmkuhl [42].

Certainly, after the presentation of the objectives and goals of the present thesis, it can be said that this thesis has not followed the famous saying that states *"Do not bite off more than you can chew"*, and has tackled very different areas and topics of CFD modelling. Even though this can be seen as a handicap, since addressing such different goals does not allow to deepen and broadly investigate a specific research topic, this investigation strategy has allowed to acquire a broad knowledge and transversal know-how in CFD science.

1.4 Outline of the thesis

As specified in Section 1.3, this thesis is devoted to the development of numerical tools and algorithms for a CFD code aiming to simulate complex flows including multi-physic phenomena. The developed tools are tested against reference benchmark cases and applied for the simulation of different situations of interest, demonstrating the capabilities of the new implemented features. Therefore, the present thesis encompasses the mathematical description of the developed and implemented models, the explanation of the numerical methods and algorithms coded to introduce these models in the CFD code, as well as the results obtained from the simulations carried out with the numerical code.

In the present chapter, the motivations to carry out the work of this thesis are briefly explained, and the goals to be achieved during the development of the thesis are also detailed. Furthermore, the trajectory of the Research Group where the thesis is developed is described, which helps to obtain a better understanding why these goals are set, and which is the starting point of the thesis.

In Chapter 2, the simulation of combustion processes using CFD is addressed. First, an introduction to the mathematical formulation required for the modelling and simulation of combustion using CFD is given. The chapter continuous with the presentation and implementation of a combustion model, namely the Progress-Variable model. This model allows to reduce the computational requirements needed in combustion simulations. Moreover, a closure model aiming to account for the flow-chemistry interaction occurring at sub-grid level in LES modelling, the Presumed Conditional Moment, is presented. This PV-PCM model is employed to reproduce a well-known benchmark case, the Cambridge autoignition experiment. The implementation details of an efficient load-balancing method for the computation of the reaction rates employing finite-rate chemistry are then presented. This method allows to speed-up combustion simulations, reducing considerably the computational time optimizing the computational load of parallel simulations. In order to assess the performance and scalability of the method, a detailed numerical study is carried out, and the method is employed to reproduce the Cambridge Autoignition experiment using finite-rates chemistry, previously simulated employing the Progress-Variable model.

Chapter 3 is devoted to the description and implementation of all the models and algorithms required for the simulation of dispersed multi-phase flows. This chapter first presents the mathematical formulation and models employed to describe the physics of dispersed multi-phase flows using the Lagrangian-Eulerian modelling approach. One of the critical aspects of the Lagrangian-Eulerian model is the interpolation of the continuous phase properties at the particle's position. Hence, the chapter continuous with a detailed study of interpolation methods for scattered data, as the numerical method is implemented to be employed for both structured and unstructured meshes. The next issue addressed in this chapter is the study of models for the particle sub-grid dispersion in LES modelling. First, two models available in the literature are implemented and deeply studied. With the objective of avoiding the shortcomings and deficiencies of the above mentioned two models, a new model based on the reconstruction of the sub-grid data lost in the process of LES filtering from statistical information has been proposed. The statistical information employed for the model to reconstruct the sub-grid scales is obtained from rigorous DNS studies. Closing the chapter, the numerical algorithms presented are employed to simulate two experiments with dispersed multi-phase flows. The first one is the well-known flow loop *Hercule*, which presents a similar configuration to those of the pulverized coal injection systems in the coal furnaces. The second case is an experiment focused on how inhaled medicines are transported and deposited throughout an *in-vitro* set-up reproducing the human conducting airways.

Chapter 4 covers the development and implementation of a new technique aimed to perform simulations using two or more non-overlapping meshes which present a relative motion between them. This technique is specially well-suited for cases like wind turbines or tidal-stream turbines, where static and moving parts closely coexist, and the flow interaction between both parts is of crucial importance. The chapter begins explaining the mathematical basis and parallelization strategy of the developed method. The results of an exhaustive analysis of the conservation properties of the methodology are presented, as well as the results of a study on the performance and parallel scalability of the method. Next, the method is employed for the simulation of real test cases, demonstrating the capabilities of the developed algorithm to perform this type of simulations, obtaining reliable numerical results. The chapter finishes presenting an improved version of the above mentioned method. The new methodology is currently under development. It aims to enhance the conservation properties of the previous version of the algorithm.

In Chapter 5, a self-adaptive time-integration method for convection-diffusion equations is presented. The methodology is based on the previous work by Trias and Lehmkuhl [42], who developed a self-adaptive technique for symmetry-preserving schemes using a 2^{nd} order Linear Multi-Step method. In the present work the method is generalized to other types of convective schemes, specifically upwind-like schemes, and new types of time integration schemes, like Predictor-Corrector methods and Runge-Kutta schemes. The chapter begins with an explanation regarding the mathematical basis of the method and continues with a detailed description about how the method is applied to convection-diffusion equations employed in CFD simulations. Closing the chapter, different numerical tests are carried out in order to analyse the performance and capabilities of the presented time-integration strategy.

And last but not least, in Chapter 6 the main conclusions derived from the work carried out throughout the development of this thesis, as well as the possible work and research to be carried out in a short and a mid-term future as continuation of this thesis, are presented.

References

[1] F.M. White. *Fluid Mechanics*. McGraw-Hill series in mechanical engineering. McGraw Hill, 2011.

[2] Lewis Fry Richardson. *Weather Prediction by numerical process*. Cambridge University Press, 1922.

[3] Julian C.R. Hunt. Lewis Fry Richardson and his contributions to mathematics, meteorology, and models of conflict. *Annual Review of Fluid Mechanics*, 30(1):xiii–xxxvi, 1998.

[4] Peter Lynch. The origins of computer weather prediction and climate modeling. *Journal of Computational Physics*, 227(7):3431–3444, 2008.

[5] Francis H. Harlow. Fluid dynamics in group T-3 Los Alamos National Laboratory:(la-ur-03-3852). *Journal of Computational Physics*, 195(2):414–433, 2004.

[6] J.L. Hess and A.M.O. Smith. Calculation of potential flow about arbitrary bodies. *Progress in Aerospace Sciences*, 8:1–138, 1967.

[7] Hao Zhang, F.X. Trias, Andrey Gorobets, Yuanqiang Tan, and Assensi Oliva. Direct numerical simulation of a fully developed turbulent square duct flow up to reτ= 1200. *International Journal of Heat and Fluid Flow*, 54:258–267, 2015.

[8] Haiou Wang, Evatt R. Hawkes, Bo Zhou, Jacqueline H. Chen, Zhongshan Li, and Marcus Aldén. A comparison between direct numerical simulation and experiment of the turbulent burning velocity-related statistics in a turbulent methane-air premixed jet flame at high Karlovitz number. *Proceedings of the Combustion Institute*, 36(2):2045–2053, 2017.

[9] Millennium problems: Navier-Stokes equations. http://www.claymath.org/millennium-problems/navier%E2%80%93stokes-equation. Accessed: 05/09/2017.

[10] Andrey Nikolaevich Kolmogorov. The local structure of turbulence in incompressible viscous fluid for very large Reynolds numbers. In *Dokl. Akad. Nauk SSSR*, volume 30, pages 299–303, 1941.

[11] Osborne Reynolds. On the Dynamical Theory of Incompressible Viscous Fluids and the Determination of the Criterion. *Philosophical Transactions of the Royal Society of London. A*, 186:123–164, 1895.

[12] J. Smagorinsky. General Circulation Experiments with the Primitive Equations. *Monthly Weather Review*, 91:99, 1963.

[13] S.B. Pope. *Turbulent Flows.* Cambridge University Press, 2000.

[14] Robert R. Schaller. Moore's law: past, present and future. *IEEE spectrum*, 34(6):52–59, 1997.

[15] Samuel H. Fuller and Lynette I. Millett. *The Future of Computing Performance: Game Over or Next Level?* The National Academies Press, Washington, DC, 2011.

[16] Asensio Oliva. *Resolución sistemática por métodos numéricos de la transmisión de calor por conducción y convección en condiciones subsónicas y de estabilización.* PhD thesis, Universitat Politècnica de Catalunya, 1982.

[17] Carlos David Pérez Segarra. *Criterios numéricos en la resolución de la transferencia de calor en fenómenos de convección.* PhD thesis, Universitat Politècnica de Catalunya, 1988.

[18] M. Soria. *Parallel multigrid algorithms for computational fluid dynamics and heat transfer.* PhD thesis, Universitat Politècnica de Catalunya, 2000.

[19] Francesc Xavier Trias. *Direct numerical simulation and regularization modelling of turbulent flows on loosely coupled parallel computers using symmnetry-preserving discretizations.* PhD thesis, Universitat Politècnica de Catalunya, 2006.

[20] 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.

[21] CTTC-UPC. DPC: library for the development of programs focussed on the resolution of combined heat and mass transfer problems. Universitat Politècnica de Catalunya, 2004.

[22] Ivette Rodríguez Pérez. *Unsteady laminar convection in cylindrical domains: numerical studies and application to solar water storage tanks.* PhD thesis, Universitat Politècnica de Catalunya, 2006.

[23] Julián Ernesto Jaramillo Ibarra. *Suitability of different RANS models in the description of turbulent forced convection flows: application to air curtains.* PhD thesis, Universitat Politècnica de Catalunya, 2008.

[24] Ricard Cònsul Serracanta. *Development of numerical codes for the evaluation of combustion processes. Detailed numerical simulations of laminar flames.* PhD thesis, Universitat Politècnica de Catalunya, 2002.

[25] Kilian Claramunt Altimira. *Numerical Simulation of Non-premixed Laminar and Turbulent Flames by means of Flamelet Modelling Approaches.* PhD thesis, Universitat Politècnica de Catalunya, 2005.

[26] Daniel Carbonell Sánchez. *Numerical studies of diffusion flames. Special emphasis on flamelet concept and soot formation.* PhD thesis, PhD thesis, Universitat Politecnica de Catalunya, 2008.

[27] O. Lehmkuhl, C.D. Perez-Segarra, R. Borrell, M. Soria, and A. Oliva. TERMOFLUIDS: A new Parallel unstructured CFD code for the simulation of turbulent industrial problems on low cost PC Cluster. *Parallel Computational Fluid Dynamics 2007*, pages 275–282, 2009.

[28] Oriol Lehmkuhl. *Numerical resolution of turbulent flows on complex geometries.* PhD thesis, Universitat Politècnica de Catalunya, 2012.

[29] Ricard Borrell Pol. *Parallel algorithms for computational fluid dynamics on unstructured meshes.* PhD thesis, Universitat Politècnica de Catalunya, 2012.

[30] R. Borrell, J. Chiva, O. Lehmkuhl, G. Oyarzun, I. Rodríguez, and A. Oliva. Optimising the Termofluids CFD code for petascale simulations. *International Journal of Computational Fluid Dynamics*, 30(6):425–430, 2016.

[31] Lluís Jofre Cruanyes. *Numerical simulation of multiphase immiscible flow on unstructured meshes.* PhD thesis, Universitat Politècnica de Catalunya, 2014.

[32] Néstor Vinicio Balcázar Arciniega. *Numerical simulation of multiphase flows: level-set techniques.* PhD thesis, Universitat Politècnica de Catalunya, 2014.

[33] Jordi Ventosa Molina. *Numerical simulation of turbulent diffusion flames using flamelet models on unstructured meshes.* PhD thesis, Universitat Politècnica de Catalunya, 2015.

[34] O. Antepara, O. Lehmkuhl, R. Borrell, J. Chiva, and A. Oliva. Parallel adaptive mesh refinement for large-eddy simulations of turbulent flows. *Computers & Fluids*, 110:48–61, 2015.

[35] O. Estruch, O. Lehmkuhl, R. Borrell, C.D. Pérez Segarra, and A. Oliva. A parallel radial basis function interpolation method for unstructured dynamic meshes. *Computers & Fluids*, 80:44–54, 2013.

[36] Henk Kaarle Versteeg and Weeratunge Malalasekera. *An introduction to computational fluid dynamics: the finite volume method.* Pearson Education, 2007.

[37] Joaquim Peiró and Spencer Sherwin. *Finite difference, finite element and finite volume methods for partial differential equations.* Springer, 2005.

[38] Shaofan Li and Wing Kam Liu. Meshfree and particle methods and their applications. *Applied Mechanics Reviews*, 55(1):1–34, 2002.

[39] T. R. Young and J. P. Boris. A numerical technique for solving stiff ordinary differential equations associated with the chemical kinetics of reactive-flow problems. *The Journal of Physical Chemistry*, 81(25):2424–2427, 1977.

[40] C. William Gear. *Numerical Initial Value Problems in Ordinary Differential Equations.* Prentice Hall PTR, Upper Saddle River, NJ, USA, 1971.

[41] Daniel Martínez Valdivieso. *Towards a virtual platform for aerodynamic design, performance assessment and optimization of horizontal axis wind turbines.* PhD thesis, Universitat Politècnica de Catalunya, 2017.

[42] F.X. Trias and O. Lehmkuhl. A Self-Adaptive Strategy for the Time Integration of Navier–Stokes Equations. *Numerical Heat Transfer, Part B: Fundamentals*, 60(2):1116–134, 2011.

2

Combustion

Main contents of this chapter have been presented at international conferences:

J. Muela, J. Ventosa, O. Lehmkuhl and A. Oliva. *Large eddy simulation of hydrogen autoignition in a preheated turbulent co-flow.* In Proceedings of the *8th Mediterranean Combustion Symposium 2013*, Çesme, Izmir, Turkey, 8-13 September 2013.

J. Muela, J. Ventosa, O. Lehmkuhl and A. Oliva. *Study of the autoignition of a hydrogen jet in a turbulent co-flow of heated air using LES modelling.* In Proceedings of the *6th European Combustion Meeting 2013*, Lund, Sweden, 25-28 June 2013.

J. Muela, R. Borrell, J. Ventosa, L. Jofre, O. Lehmkuhl, and A. Oliva. *An efficient parallelization method for the evaluation of chemical reaction rates.* In Proceedings of the *26th International Conference on Parallel Computational Fluid Dynamics 2014*, Trondheim, Norway, 22-24 May 2014.

And some contents of the chapter have been published at international journals:

J. Ventosa-Molina, J. Chiva, O. Lehmkuhl, J. Muela, C. D. Pérez-Segarra, and A. Oliva. *Numerical analysis of conservative unstructured discretisations for low mach flows.* International Journal for Numerical Methods in Fluids, 84(6):309–334, 2017.

Abstract. Combustion processes have been present throughout all the history of mankind. Unquestionably, combustion has played a main role and determined the evolution process of human being, since its dawn with the discovery of the fire, up to the present, where combustion is the current main source of energy of the humanity. Combustion is a complex physical problem where different chemical species reacts exothermically generating new species and heat. This physic phenomena is characterized by a wide range of spatial and temporal scales.

The conditions in which combustion takes place (flow velocity, turbulence, mixture composition, temperature, etc.) determines the amount of energy and products delivered by the combustion process. Nowadays, the society and the industry are highly dependent on fossil fuels, since they are the main source of energy to generate electricity, to power cars and air-planes, or even to generate heat. Hence, in the current geopolitical framework as well as in the upcoming years, where fossil fuels are becoming an increasingly scarce resource,

their optimal use is becoming each time more and more crucial. But not only that, pollutant emissions by combustion are the main reason of current climate change as well as the origin of many health problems of people living in polluted areas like populated cities. Hence, there is a clear necessity in the optimization of the combustion processes in order to do more with less, as well as an interest in reducing the pollutant emissions derived from combustion processes. These objectives can be achieved studying and improving the current combustion processes present in both domestic and industrial applications, like internal combustion engines or gas turbines. In order to do so, CFD simulations can be a very powerful and useful tool.

CFD simulations of combustion processes require the computation of flow transport combined together with chemical reactions. Hence, the first part of the present chapter is devoted to the introduction of the mathematical equations governing the mass, momentum and energy transport of a flow mixture and the equations describing chemical-reaction processes. Since the calculation of chemical reactions employing finite-rate methodology can be very expensive and time-consuming, a combustion model that reduces dramatically this computational cost is introduced and implemented. This combustion model, named Progress-Variable model, is assessed and validated against a reference experimental case. Instead of using simplified models, another option to reduce the computational time of combustion simulations is to enhance the computational performance of finite-rate chemistry models. With that aim, the final part of this chapter is devoted to the development of an efficient parallelization method for chemical reactions, based on the smart balancing of computational load in parallel simulations.

2.1 Introduction

Combustion has been, it is, and it will be present throughout all history of human being. Combustion processes surround our daily live, from heavy industry to the most simple domestic application, going through all means of transport, maritime, terrestrial, aerial and even spatial. Therefore, the study of combustion processes employing CFD simulations has received a wide attention from researchers and engineers during the last decades. This is due CFD simulations are a very efficient and interesting tool for the design and optimization of combustion applications. However, this area is characterized by a wide range of coupled problems [1], which makes combustion simulations complex and challenging. Some of the difficulties presented by CFD combustion simulations are:

- Description of the fluid-dynamic problem: the flow motion as well as the mixing of the reactants must be well-described.
- Chemical process is defined by a detailed chemical reaction schemes: it is required a well description of the chemistry involved in the combustion process in order to predict the ignition, the stabilization/extinction of the flame and the formation of pollutants.
- Combustion systems can be multi-phase: in many combustion systems the fuel is injected as liquid or solid. Hence, the transport, atomization, evaporation/sublimation, nucleation, etc. of particles or droplets must be also addressed.
- Radiative heat-transfer: due to the high temperatures present in combustion systems, radiative heat-transfer can play a major role, and must be taken into account in the design/optimization of the systems.

In this chapter, the two first issues will be addressed. In turbulent combustion problems, which represents the practical majority of combustion applications, there are involved a large number of degrees of freedom. This issue makes computationally unaffordable Direct Numerical Simulations (DNS) of real systems. Hence, it is required to draw on modelling techniques aimed to reduce the number of degrees of freedom. Historically, the first approach that allowed the resolution of turbulent flows was the Reynolds-Averaged Navier Stokes (RANS) model. This technique calculates the mean values of all quantities. The transport equations for these mean quantities are obtained by averaging the instantaneous transport equations. A more recent modelling approach, which requires greater computational resources, is the Large Eddy Simulation (LES). In LES only the large-scales of the flow are explicitly resolved, while the scales of the flow smaller than a filter-size are not computed, and should be

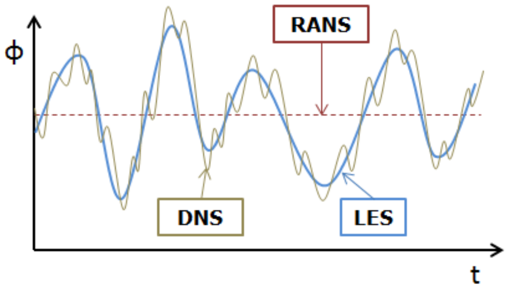


Figure 2.1: Temporal evolution.

modelled. This modelling technique is based on the application of a low-pass filter to the instantaneous transport equations.

In order to exemplify the results obtained from both models, an example of the temporal evolution of a variable ϕ in RANS, LES and DNS is shown in Fig. 2.1. As can be seen, in DNS all the frequency variations of the variable ϕ are captured, while in LES only the scales larger than the filter-size are calculated. On the other hand, in RANS only a time averaged value of ϕ is obtained. This behaviour can be translated to the spectral space, as depicted in Fig. 2.2. As stated previously, in RANS all the energy spectrum is modelled, in contrast to DNS, where all the spatial frequencies in the spectrum are resolved. In LES only the large-scales of the flow, associated to wave numbers smaller than k_c are solved, and the bigger ones have to be modelled. The cut-off wave number k_c is related to the filter-size employed in the LES filtering procedure.

In both techniques, RANS and LES, appear additional unknown terms in the transport equations that cannot be calculated explicitly and should be modelled. Since this thesis is focused on LES modelling, throughout this chapter the derivation of LES transport equations, as well as the models required to close the unresolved terms, will be presented.

The other issue faced in the current chapter is the description of the chemical reactions. Combustion processes may involve hundreds or even thousands of species and reactions, and the integration of the resulting chemistry system of ordinary differential equations (ODEs) requires a huge amount of computation resources, even for $0D$ reactors [2]. In multi-dimensional CFD simulations the computational requirements are even higher, since a transport equation for each one of the species present in the chemical reaction mechanism has to solved. And the problem is not only the size of the equations system, but also its stiffness [3]. Stiff systems of equations must use a very small and restrictive time-step to be integrated explicitly. Therefore, ro-

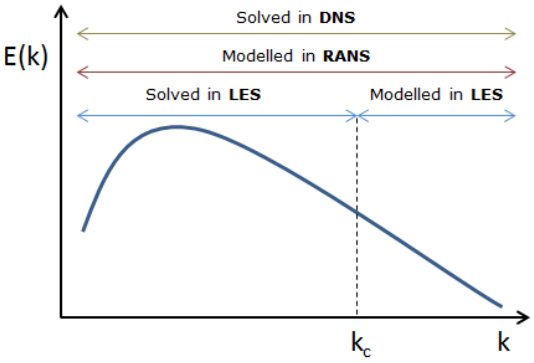


Figure 2.2: Energy spectrum.

bust implicit integration methods have to be employed [4]. Large chemical reaction mechanisms are in general less stiff than reduced ones, but involve more species and reactions.

In this thesis, these complexities are addressed from two different ways. First, a combustion model based on tabulated chemistry is developed and implemented. The basic idea is to pre-compute the chemical reactions, store them in a look-up table, and then, during the CFD simulations, the chemical reactions are obtained accessing to this look-up table instead of being calculated. Combustion models are based on some assumptions and hypothesis of the chemical reactions that allow to obtain a mathematical simplification to the chemistry, permitting to reduce the computational requirements of combustion simulations.

The second approach relies on an enhancement of the computational performance instead of simplified models of the chemical reactions. This approach is based on a robust implicit time integration technique added to the efficient distribution of the computational load in parallel computing. Basically, only the regions of the flame front, which present a stiff set of equations, are integrated using the implicit method, while the other regions are integrated explicitly. However, since the flame front is usually found in specific regions of the computational domain, which belongs only to a few CPUs, these processors have a much bigger computational demand than the other processors involved in the simulation, presenting a clear imbalance in the computational load. Therefore, an algorithm that balances this computational effort between all the processors has been developed and implemented. This algorithm allows to notably reduce the computational time of combustion simulations when complex chemical reactions mechanisms are used.

This chapter is organised as follows: at the beginning the governing equations describing both flow transport phenomena and chemical reaction processes are introduced. Moreover, since combustion processes are usually turbulent, the modelling of turbulent flows employing LES is detailed, and the physical and mathematical implications derived of the employment of this modelling technique are explained. To complete the mathematical formulation, the algorithms and integration techniques employed to solve the governing equations are also explained. Afterwards, since the calculation of the equations describing chemical reactions has a very high computational cost, a combustion model which reduces the required computational effort is introduced. The model is based on a separation of the chemical kinetics and the flow transport physics, assuming that both phenomena occurs at very different time-scales. This assumption allows to pre-compute the chemical reactions and tabulate them in a look-up table. Then, during CFD simulations, the chemical process is read from this look-up table, instead of being calculated, resulting in a significant save of the computational cost. Following, in order to assess and validate the implemented combustion model, the autoignition process of a hydrogen jet into a preheated turbulent air stream is numerically studied. This reference case consists of a fuel jet, a mixture of hydrogen and nitrogen, flowing through a nozzle placed at the centre of a co-flowing air stream. The second part of this chapter is focused on an efficient load-balancing methodology for parallel computing of combustion processes employing finite-rate chemistry. The implemented method is mainly based on an implicit integration of the mass reaction rates using the Gear's backward differentiation method for stiff sets of equations. The chapter is closed with some conclusions derived from the work carried out, as well as some ideas and challenges to be faced in order to continue this research work in the field of CFD combustion.

2.2 Mathematical formulation

The equations governing the physical phenomena required to describe and simulate combustion processes are presented in this section. First the equations describing the kinetics of chemical reaction processes are introduced, and then, the governing equations for flow mixtures in low-Mach conditions are presented.

2.2.1 Chemical kinetics

Before introducing the governing equations of chemical reactions, it is required to define some values and magnitudes that describe the composition of a mixture composed by N species. The basic magnitude required to define a mixture is the mass fraction of each species k present on it:

$$Y_k = \frac{m_k}{m} \tag{2.1}$$

where m_k is the mass of species k and m the total mass of the mixture in a volume

V. Obviously, the density of the mixture is $\rho = \frac{m}{V}$. For a mixture of N perfect gases, the density can also be calculated through the ideal gas law:

$$\rho = \frac{pW}{TR} \tag{2.2}$$

where p and T are the pressure and temperature of the system, $R = 8.314$ J/(mol K) is the perfect gas constant, and W the molecular mass of the mixture. The latter is defined from the individual molecular masses W_k and the species mass fractions Y_k as:

$$\frac{1}{W} = \sum_{k=1}^{N} \frac{Y_k}{W_k} \tag{2.3}$$

In a mixture of perfect gases, the total pressure is the sum of partial pressures, which can be calculated from the ideal gas law:

$$p = \sum_{k=1}^{N} p_k \quad \text{where} \quad p_k = \rho_k \frac{R}{W_k} T \tag{2.4}$$

The molar fraction X_k (moles of species k / total moles) and the molar concentration $[X_k]$ (moles of species k / volume) can be obtained from the previously defined magnitudes:

$$X_k = \frac{W}{W_k} Y_k \tag{2.5}$$

$$[X_k] = \rho \frac{Y_k}{W_k} = \rho \frac{X_k}{W} \tag{2.6}$$

A chemical reaction is a thermodynamic process where different substances, named reactants, are transformed into others, named products, modifying their molecular structures and bonds. The kinetic of the reaction is determined by the conditions of the system where reaction takes place. The equations describing the chemical reactions of N species through M reactions can be written as:

$$\sum_{k=1}^{N} \nu^{'}_{k,j} \mathcal{M}_k \rightleftarrows \sum_{k=1}^{N} \nu^{''}_{k,j} \mathcal{M}_k \quad ; \quad j = 1, 2, ..., M. \tag{2.7}$$

where $\mathcal{M}_k$ stands for the species involved in the different chemical reactions, and values $\nu^{'}_{k,j}$ and $\nu^{''}_{k,j}$ are the reactant and product stoichiometric coefficients of species k, respectively, in chemical reaction j. Obviously, due to mass conservation, the following constraint is fulfilled:

$$\sum_{k=1}^{N} \nu^{'}_{k,j} W_k = \sum_{k=1}^{N} \nu^{''}_{k,j} W_k \quad or \quad \sum_{k=1}^{N} \nu_{k,j} W_k = 0 \quad ; \quad j = 1, 2, ..., M \tag{2.8}$$

where $\nu_{k,j} = \nu^{'}_{k,j} - \nu^{''}_{k,j}$. The mass reaction rate per unit volume of each species k, $\dot{w}_k$, is obtained from the sum of the rates $\dot{w}_{k,j}$ for all M reactions:

$$\dot{w}_k = \sum_{j=1}^{M} \dot{w}_{k,j} = W_k \sum_{j=1}^{M} \nu_{k,j} \mathcal{R}_j \tag{2.9}$$

where $\mathcal{R}_j$ is the progress rate of reaction j and is calculated as:

$$\mathcal{R}_j = k^f_j \prod_{k=1}^{N} [X_k]^{\nu^{'}_{k,j}} - k^r_j \prod_{k=1}^{N} [X_k]^{\nu^{''}_{kj}} \tag{2.10}$$

where k^f_j and k^r_j are the forward and reverse kinetic rate constant of reaction j. Spite of their name, these values are not strictly constant and must be calculated. The forward rate constant is evaluated using the empirical Arrhenius law:

$$k^f_j = A^f_j T^{\beta_j} \exp\left(-\frac{E_{a,j}}{RT}\right) \tag{2.11}$$

where A^f_j is the pre-exponential factor, β_j is the temperature exponential and $E_{a,j}$ is the activation energy. This activation energy is related with the activation temperature as $E_{a,j} = RT_{a,j}$. The value of these constants are obtained from chemical schemes that are constructed and generated by researchers experimentally. Deciding which species or how many reactions should be taken into account to properly describe a chemical reaction is not a trivial issue. Lot of different chemical mechanisms can be found in the literature, ranging from very detailed mechanisms as the GRI-Mech 3.0 [5], designed to model natural gas combustion, including 53 species and 325 reactions, to single-step mechanism like the one defined by Lange et al. for methane/air flames [6], going through reduced mechanisms, such as the chemical scheme obtained by Mueller et al. for hydrogen [7], which includes 9 species and 21 reactions. Obviously, as more species and reactions are included in the mechanism, more detailed results are obtained, although the calculation costs increase exponentially. Combustion simulations can be very sensitive to the chemical mechanism employed. Therefore, special care should be taken when selecting it, trying to find the best compromise between accuracy and computational cost. The reverse rate constant k^r_j is usually not provided in these reaction mechanisms and is calculated from the forward rate constant k^f_j through the reaction equilibrium constants:

$$k_j^r = \frac{k_j^f}{\left(\frac{p_a}{RT}\right)^{\sum_{k=1}^{N} \nu_{kj},} \exp\left(\frac{\Delta S_j^0}{R} - \frac{\Delta H_j^0}{RT}\right)} \tag{2.12}$$

where $p_a = 0.1$ MPa, and S_j^0 and H_j^0 are the standard entropy and enthalpy changes in chemical reaction j respectively. The term inside the exponential part is intimately related with the Gibbs free energy (G) of the reaction. If the mass reaction rate of all species are summed, and using Eq. (2.8), it is demonstrated that the total mass of the system is conserved:

$$\sum_{k=1}^{N} \dot{w}_k = \sum_{k=1}^{N} W_k \left(\sum_{j=1}^{M} \nu_{k,j} \mathcal{R}_j \right) = \sum_{j=1}^{M} \mathcal{R}_j \left(\sum_{k=1}^{N} W_k \nu_{k,j} \right) = 0 \tag{2.13}$$

2.2.2 Transport equations

The equations governing the thermodynamic and fluid-dynamic behaviour of a fluid flow in a combustion process are the Navier-Stokes equations. Depending on the conditions of the flow, basically its characteristic Mach number, these equations can be formulated using a compressible or a low-Mach number formulation. In the present thesis, the focus is placed on the low-Mach number regime. Following the procedure of Lessani and Papalexandris [8], the continuity and momentum equations are:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \tag{2.14}$$

$$\frac{\partial (\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u}\mathbf{u}) = -\nabla p + \nabla \cdot \boldsymbol{\tau} + \rho \mathbf{g} \tag{2.15}$$

where t represents the time, $\mathbf{u}$ is the velocity vector, $\mathbf{g}$ the gravity and $\boldsymbol{\tau}$ is the viscous stress tensor. Assuming that the fluid is Newtonian, $\boldsymbol{\tau}$ is defined as:

$$\boldsymbol{\tau} = \mu \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T \right) + \mathbf{I} \mu_2 \nabla \cdot \mathbf{u} \tag{2.16}$$

where $\mathbf{I}$ is the unit tensor, μ the viscosity and μ_2 the second coefficient of viscosity. The latter is usually modelled as: $\mu_2 = -\frac{2}{3}\mu$, so Eq. (2.16) can be rearranged as:

$$\boldsymbol{\tau} = \mu \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T - \frac{2}{3} \mathbf{I} \nabla \cdot \mathbf{u} \right) \tag{2.17}$$

The energy conservation equation reads as:

$$\frac{\partial (\rho h)}{\partial t} + \nabla \cdot (\rho \mathbf{u} h) = \frac{Dp}{Dt} - \nabla \cdot \dot{\mathbf{q}} + \boldsymbol{\tau} \nabla \cdot \mathbf{u} + \dot{\mathcal{Q}} \tag{2.18}$$

where $\dot{\mathbf{q}}$ is the conduction heat flux evaluated from the Fourier's law, $\dot{\mathbf{q}} = -\lambda \nabla T$, and $\dot{\mathcal{Q}}$ is a heat source term, including radiation. The energy equation is solved for the total enthalpy h, which includes both sensible and chemical (formation) enthalpy, it is:

$$h = \sum_{k=1}^{N} h_k Y_k = \sum_{k=1}^{N} \left(\underbrace{\int_{T_0}^{T} C_{p,k}}_{\text{sensible}} + \underbrace{\Delta h_{f,k}^{0}}_{\text{formation}} \right) Y_k \tag{2.19}$$

Moreover, it is required to solve a transport equation for each one of the N species of the mixture. The mass conservation equation for species k is:

$$\frac{\partial(\rho Y_k)}{\partial t} + \nabla \cdot (\rho(\mathbf{u} + \mathbf{V_k})Y_k) = \dot{w}_k \tag{2.20}$$

where $\mathbf{V_k}$ is the diffusion velocity of the species k. The sum of the diffusion fluxes of all the species must be null. It can be proved summing Eq. (2.20) for the N species:

$$\sum_{k=1}^{N} \left(\frac{\partial \rho Y_k}{\partial t} + \nabla \cdot (\rho \mathbf{u} Y_k) \right) = -\sum_{k=1}^{N} \nabla \cdot (\rho \mathbf{V_k} Y_k) + \sum_{k=1}^{N} \dot{w}_k \tag{2.21}$$

The summation in the left hand side is the continuity equation, hence its value is null, and as demonstrated in Eq. (2.13) the sum of all species mass reaction rate is zero. Therefore, the sum of all diffusion velocities must cancel out:

$$-\sum_{k=1}^{N} \nabla \cdot (\rho \mathbf{V_k} Y_k) = 0 \tag{2.22}$$

In order to obtain the diffusion velocities $\mathbf{V_k}$ for all the N species of a mixture, the model of Stefan-Maxwell for multi-component systems is [9]:

$$\nabla X_p = \sum_{k=1}^{N} \frac{X_p X_k}{\mathcal{D}_{pk}} (\mathbf{V_k} - \mathbf{V_p}) \quad \text{for } p = 1..N \tag{2.23}$$

where $\mathcal{D}_{pk}$ is the binary mass diffusion coefficient of species k into species p. Note that $\mathcal{D}_{pk} = \mathcal{D}_{kp}$. This model assumes that system pressure is constant, the collisions are purely elastic (kinetic energy is conserved) and there is not shear stress (negligible velocity gradients). The Soret effect (diffusion due to temperature gradients) is also neglected. The solution of this linear system for a multi-component mixture is computationally expensive[10]. Therefore, usually in CFD simulations simplified approaches are employed [11]. The two most common models and that will be employed from now on in the present thesis are the Fick's law [12] and the Hirschfelder and Curtiss approximation [13]. If the mass diffusivity coefficient is assumed equal for all the species, or in case of binary mixtures, the solution to Eq. (2.23) is exact, and the Fick's law is obtained, which states:

$$\mathbf{V_k} = -D\frac{\nabla Y_k}{Y_k} \tag{2.24}$$

where $D = D_{pk}$ is the mass diffusivity for all species. If this assumption of equal diffusivity is not made, complex models for multi-species mixtures should be employed. The best first-order approximation to Eq. (2.23) is the Hirschfelder and Curtiss model, which reads as:

$$\mathbf{V_k} = -D_k\frac{\nabla X_k}{X_k} \quad \text{with} \quad D_k = \frac{1 - Y_k}{\sum_{j\neq k}\frac{X_j}{\mathcal{D}_{jk}}} \tag{2.25}$$

where D_k is the mass diffusion coefficient of species k into the mixture. For binary diffusions, the same exact solution as in Stefan-Maxwell and Fick's law models is recovered. However, when the mixture is composed by three or more species this is not longer true. Another issue is the mass conservation when this model is employed. If the expression in Eq. (2.25) is replaced in Eq. (2.22), the following expression is obtained:

$$\nabla\cdot\left(\rho\sum_{k=1}^{N}D_k\frac{\nabla X_k}{X_k}Y_k\right) = \nabla\cdot\left(\rho\sum_{k=1}^{N}D_k\frac{W_k}{W}\nabla X_k\right) \neq 0 \tag{2.26}$$

which is no longer null, and therefore, the mass conservation constraint is not fulfilled. In order to solve this issue there are two possibilities:

1. Only the mass conservation equation Eq. (2.14) and $N-1$ species transport equations Eq. (2.20) are solved. The mass fraction of the last species Y_N, usually the diluent of the mixture, is obtained as $Y_N = 1 - \sum_{k=1}^{N-1} Y_k$, absorbing all the discrepancies due model simplifications and numerical issues, and forcing global mass conservation.

2. Adding a correction velocity $\mathbf{V_c}$ to the species transport equation. This correction velocity is derived in such way that the global mass conservation equation is forced, and is calculated using the following expression:

$$\mathbf{V_c} = \sum_{k=1}^{N}D_k\frac{W_k}{W}\nabla X_k \tag{2.27}$$

so the final expression for the transport equation of species k reads as:

$$\frac{\partial \rho Y_k}{\partial t} + \nabla \cdot (\rho(\mathbf{u} + \mathbf{V_c})Y_k) = \nabla \cdot \left(\rho D_k \frac{W_k}{W} \nabla X_k\right) + \dot{w}_k \tag{2.28}$$

Recapitulating, the required set of transport equations to simulate a mixture flow of N species in a combustion case are continuity equation, the momentum equations, energy equation and N species mass-fraction equations, together with their constitutive equations: Newton's law, Fourier's law and diffusion velocity of species.

2.2.3 Thermodynamic pressure

In low Mach flows, the temperature, the density and a spatially uniform thermodynamic pressure P_0 are coupled through the perfect gas state equation:

$$P_0 = \rho \frac{R}{W} T = \rho R_g T \tag{2.29}$$

which results in a constraint of the energy equation on the flow. Employing this equation together with mass conservation equation (Eq. (2.14)), energy conservation equation (Eq. (2.18)) and the species mass fraction conservation equation(Eq. (2.20)), the velocity divergence constraint for perfect gases becomes [14]:

$$\begin{aligned}\nabla \cdot \mathbf{u} = \frac{R_g}{c_p P_0} \Bigg[&\nabla \cdot (\lambda \nabla T) - \left(\frac{c_p}{R_g} - 1\right) \frac{dP_0}{dt} - \sum_{k=1}^{N} h_k \dot{w}_k \\ &+ \sum_{k=1}^{N} \rho c_{p,k} \left(\frac{D_k}{M_w} \nabla \left(M_w Y_k\right) - \mathbf{V_c} Y_k\right) \nabla T \Bigg]\end{aligned} \tag{2.30}$$

If the previous expression is integrated over a domain Ω, it is obtained an equation for the variation of the thermodynamic pressure P_0 is obtained:

$$\begin{aligned}\frac{dP_0}{dt} = &\frac{1}{\int_\Omega \left(\frac{c_p}{R_g} - 1\right) d\Omega} \left[\int_\Omega \nabla \cdot (\lambda \nabla T)\, d\Omega - \frac{P_0}{R_g} \int_\Omega c_p \nabla \cdot \mathbf{u} d\Omega \right. \\ &\left. - \int_\Omega \sum_{k=1}^{N} h_k \dot{w}_k d\Omega + \int_\Omega \sum_{k=1}^{N} \rho c_{p,k} \left(\frac{D_k}{M_w} \nabla \left(M_w Y_k\right) - \mathbf{V_c} Y_k\right) \nabla T d\Omega \right]\end{aligned} \tag{2.31}$$

If the domain Ω is closed, the mass conservation constraint must be fulfilled and the thermodynamic pressure P_0 can be obtained at a given time t knowing the total mass of the system M_0 and the temperature field:

$$M_0 = \frac{P_0}{R_g} \int_\Omega \frac{1}{T_0} d\Omega \tag{2.32}$$

$$P_0(t) = \frac{M_0 R_g}{\int_\Omega \frac{1}{T(t)} d\Omega} \tag{2.33}$$

On the other hand, if the system is open, this thermodynamic pressure P_0 is considered constant and equal to the reference pressure of the specific case $P_0 = P_r$, usually the ambient pressure.

2.2.4 LES Modelling

The vast majority of combustion processes, excepting candles, lighters and some domestic furnaces, are turbulent. As aforementioned in the introductory chapter of the thesis, turbulent flows are transient, three-dimensional and present a wide-range of length and temporal scales. This intricate flow structure adds complexity to flow simulations. Direct Numerical Simulations resolve all the scales and flow structures, from the bigger ones up to the Kolmogorov scales. Although this technique is becoming more and more affordable thanks to recent advancements in High Performance Computing, it is still restricted to academic and research cases. Therefore, some modelling should be applied in order to make turbulent simulations affordable. As previously detailed, the two most common techniques are Reynolds-Averaged Navier Stokes (RANS) modelling and Large Eddy Simulation (LES) modelling. In RANS, all the scales of the energy spectrum are modelled, while in LES the large scales of the flow are solved, and only the sub-grid scales are modelled. According to Kolmogorov theory, this small scales are independent of the large length-scales and statistically isotropic at high Reynolds numbers. This assumption allows the generation of more general models and approaches for LES than for RANS, since RANS models also the large-scales of the flow, that are case-dependant. Therefore, even though LES simulations are more expensive than RANS simulations, the former ones allow more generality and give more accurate results. They allow to intrinsically capture the transient behaviour of turbulent flows, as well as many flow structures that are not resolved in RANS modelling. Hence, this thesis is developed in the framework of LES modelling.

As said, in LES modelling the large scales of the flow are solved, while the sub-grid scales are modelled. This scale separation is obtained applying a low-pass filter to the transport equations. This filtering operation reads as:

$$\overline{\phi}(\mathbf{x}) = \int G(x-r)\phi(r)dr \tag{2.34}$$

where G is the LES filter which satisfies de normalization condition $\int G(r)dr = 1$. There are different types of filters [15], but the most common one is the box filter in physical space with size $\Delta = \sqrt[3]{V}$, where V is the volume of the mesh cell ($V = \Delta_x \Delta_y \Delta_z$ in a hexahedral structured mesh). If this filter is applied to the continuity equation Eq. (2.14), the result is:

$$\frac{\partial \overline{\rho}}{\partial t} + \nabla \cdot \left(\overline{\rho \mathbf{u}}\right) = 0 \tag{2.35}$$

which gives a new filtered variable $\overline{\rho \mathbf{u}}$. In order to avoid introducing a model for the sub-grid scales on the continuity equation, Favre [16] proposed a density-weighted variable defined as:

$$\widetilde{\phi} = \frac{\overline{\rho \phi}}{\overline{\rho}} \tag{2.36}$$

allowing to express the continuity equation in terms of a density-weighted velocity:

$$\frac{\partial \overline{\rho}}{\partial t} + \nabla \cdot \left(\overline{\rho} \widetilde{\mathbf{u}}\right) = 0 \tag{2.37}$$

The concept of Favre filtering is commonly employed in LES simulations for variable density flows, such as low-Mach number and compressible flows. In the limit of incompressible flow the normal filtering is recovered since $\widetilde{\phi} = \overline{\phi}$. If Favre filtering is applied to momentum equation (Eq. (2.15)), the obtained result is:

$$\frac{\partial \overline{\rho} \widetilde{\mathbf{u}}}{\partial t} + \nabla \cdot \left(\overline{\rho} \widetilde{\mathbf{u}\mathbf{u}}\right) = -\nabla \overline{p} + \nabla \cdot \left(\overline{\boldsymbol{\tau}}\right) + \overline{\rho} \mathbf{g} \tag{2.38}$$

Notice that the filtered product of the velocities $\widetilde{\mathbf{u}\mathbf{u}}$ is different than the product of the filtered velocities $\widetilde{\mathbf{u}}\widetilde{\mathbf{u}}$. This difference $\widetilde{\mathbf{u}\mathbf{u}} - \widetilde{\mathbf{u}}\widetilde{\mathbf{u}}$ is known as the *Unresolved Reynolds stresses*, which should be modelled somehow. As in momentum, this unclosed term will arise in the convective term of any transported scalar in LES modelling. Hence, in general, the convective term of transport equations in LES is split into a resolved part and an unresolved part:

$$\nabla \cdot \left(\overline{\rho} \widetilde{\mathbf{u}\phi}\right) = \nabla \cdot \left(\overline{\rho} \widetilde{\mathbf{u}} \widetilde{\phi}\right) + \nabla \cdot \left(\overline{\rho} \left(\widetilde{\mathbf{u}\phi} - \widetilde{\mathbf{u}} \widetilde{\phi}\right)\right) \tag{2.39}$$

where $\left(\widetilde{\mathbf{u}\phi} - \widetilde{\mathbf{u}} \widetilde{\phi}\right)$ is the unresolved flux of scalar ϕ. Hence, the momentum, energy and species transport equation in LES modelling are expressed as:

$$\frac{\partial \overline{\rho} \widetilde{\mathbf{u}}}{\partial t} + \nabla \cdot \left(\overline{\rho} \widetilde{\mathbf{u}} \widetilde{\mathbf{u}}\right) = -\nabla \overline{p} + \nabla \cdot \left(\overline{\boldsymbol{\tau}} - \overline{\rho} \left(\widetilde{\mathbf{u}\mathbf{u}} - \widetilde{\mathbf{u}} \widetilde{\mathbf{u}}\right)\right) + \overline{\rho} \mathbf{g} \tag{2.40}$$

$$\frac{\partial \overline{\rho} \widetilde{h}}{\partial t} + \nabla \cdot \left(\overline{\rho} \widetilde{\mathbf{u}} \widetilde{h}\right) = \overline{\frac{Dp}{Dt}} - \nabla \cdot \left(\overline{\dot{\mathbf{q}}} + \overline{\rho} \left(\widetilde{\mathbf{u}h} - \widetilde{\mathbf{u}} \widetilde{h}\right)\right) + \overline{\boldsymbol{\tau} \nabla \cdot \mathbf{u}} + \overline{\dot{Q}} \tag{2.41}$$

$$\frac{\partial \overline{\rho} \widetilde{Y_k}}{\partial t} + \nabla \cdot \left(\overline{\rho} \widetilde{\mathbf{u}} \widetilde{Y_k}\right) = -\nabla \cdot \left(\overline{\rho \mathbf{V_k} Y_k} + \overline{\rho} \left(\widetilde{\mathbf{u} Y_k} - \widetilde{\mathbf{u}} \widetilde{Y_k}\right)\right) + \overline{\dot{w}_k} \tag{2.42}$$

In order to model the *unresolved Reynolds stresses* the Boussinesq assumption for turbulence viscosity is usually invoked [17]:

$$\overline{\rho}\left(\widetilde{\mathbf{u}\mathbf{u}} - \widetilde{\mathbf{u}}\widetilde{\mathbf{u}}\right) = -\mu_t\left(\nabla\widetilde{\mathbf{u}} + (\nabla\widetilde{\mathbf{u}})^T - \frac{2}{3}\mathbf{I}\nabla\cdot\widetilde{\mathbf{u}}\right) \tag{2.43}$$

where μ_t is the *turbulent viscosity* or *sub-grid scale viscosity.* This term have to be modelled and calculated from values ready-available in LES simulations. There are several models available in the literature in order to calculate μ_t. One popular model is the one proposed by Smagorinsky [18], based on the Prandtl mixing length applied to sub-grid scale modelling. In this model the turbulent viscosity is proportional to the strain, with a proportionality constant C_s, which value lies between 0.1 and 0.24. Another model is the one proposed by Germano et al. [19], known as *Dynamic Eddy-viscosity model*, since in this model the empirical constant C_s is dynamically adjusted. There are also available several other models like the QR eddy-viscosity model proposed by Verstappen [20], the Wall-adapting eddy viscosity model (WALE) developed by Nicoud and Ducros [21] or the Regularization models proposed by Geurts and Holm [22]. Unfortunately, there is no one general model best-suited for all the cases, and each one of them has its owns pros and cons, offering more or less accurate results depending on flow and simulation conditions. An extensive review of available turbulent models for LES modelling can be found in the thesis of O. Lehmkuhl [23]. Regarding the unresolved part of the energy and species transport equations, the unresolved scalar fluxes, are usually modelled employing a gradient assumption [10]:

$$\widetilde{\mathbf{u}h} - \widetilde{\mathbf{u}}\widetilde{h} = -\frac{\overline{\lambda_t}}{\overline{c_p}}\nabla\widetilde{T} \tag{2.44}$$

$$\widetilde{\mathbf{u}Y_k} - \widetilde{\mathbf{u}}\widetilde{Y_k} = -\overline{D_{k,t}}\nabla\widetilde{Y_k} \tag{2.45}$$

where λ_t and $D_{k,t}$ are a turbulent thermal conductivity and a turbulent diffusivity for species k respectively. These quantities are analogous to the turbulent viscosity μ_t and are usually calculated assuming a turbulent Prandtl number Pr_t and a turbulent Schmidt number Sc_t as:

$$\lambda_t = \frac{\mu_t c_p}{Pr_t} \qquad D_{k,t} = \frac{\mu_t}{\rho Sc_t} \tag{2.46}$$

Apart from the unresolved transport terms, there are other terms that also must be modelled. The laminar diffusion fluxes are usually modelled assuming a simple linearisation like:

$$\overline{\rho\mathbf{V_k}Y_k} = -\overline{\rho D_k\nabla Y_k} \approx -\overline{\rho}\,\overline{D_k}\nabla\widetilde{Y_k} \tag{2.47}$$

$$\overline{\dot{\mathbf{q}}} = \overline{\lambda \nabla T} \approx \overline{\lambda} \nabla \widetilde{T} \tag{2.48}$$

Another filtered term that requires special attention is the filtered reaction rate $\overline{\dot{w}_k}$ that appears in the species transport equation (see Eq. (2.42)). The simplest approximation is to consider that there is a perfect mixing at sub-grid scale level, assuming that the turbulent sub-grid time-scale is shorter than the time-scales of the chemical reactions ($\tau_t \ll \tau_c$). If the simplest case with an irreversible one-step chemical reaction mechanism and only two species (fuel and oxidizer) is considered, the resulting filtered reaction rate will be:

$$\overline{\dot{w}_k} = \overline{A\rho^2 T^\beta Y_f Y_o \exp(-\frac{E_a}{RT})} \approx A\overline{\rho}^2 \widetilde{T}^\beta \widetilde{Y_f} \widetilde{Y_o} \exp(-\frac{E_a}{R\widetilde{T}}) \tag{2.49}$$

However, the consideration of perfect mixing at sub-grid level is only reasonable accurate for cases of combustion in the atmospheric boundary layer [24]. For other combustion applications the approximation in Eq. (2.49) is no longer valid. A first attempt to model the filtered reaction rate is apply a Taylor series expansion to this term. Nonetheless, even for the simplest case considered here of an irreversible reaction between a fuel and an oxidizer, new quantities of high order requiring closure appear after the expansion. Hence, this method is not well-suited to derive a model for $\overline{\dot{w}_k}$. Consequently, given the inherent complexity of modelling the filtered reaction rate, added to the high computational cost of computing the chemical reactions even for the simplest reaction mechanisms, has lead to the development of different combustion models based on physical analysis [10], aiming to simulate the chemical reactions as well as recover the sub-grid scale effects lost during LES filtering. Later on this chapter, in Section 2.3, different combustion models are briefly introduced, and one of them is further explained and implemented.

2.2.5 Pressure-velocity coupling

The pressure-velocity coupling is solved by means of the Fractional Step projection method [25]. The idea behind this technique is to split the momentum in two steps, with a first explicit step where an intermediate velocity $\hat{\mathbf{u}}$ is obtained, followed by a second step where the pressure is solved implicitly and the pseudo-velocity is corrected obtaining the physical velocity. Hence, if a linear multi-step method is employed to integrate the momentum equation to a time t^{n+k} from the previous time t^{n+k-1}, where $t^{n+k} = t^{n+k-1} + \Delta t$, Eq. (2.15) can be expressed as:

$$\alpha_k(\rho^{n+k}\hat{\mathbf{u}}^{n+k}) + \sum_{j=0}^{k-1} \alpha_j(\rho^{n+j}\mathbf{u}^{n+j}) = \Delta t \sum_{j=0}^{k-1} \beta_j \left[-\mathbf{C}(\mathbf{u}) + \mathbf{D}(\mathbf{u})\right]^{n+j} \tag{2.50}$$

$$\alpha_k(\rho^{n+k}\mathbf{u}^{n+k} - \rho^{n+k}\hat{\mathbf{u}}^{n+k}) = -\Delta t \nabla p^{n+k} \tag{2.51}$$

where $\mathbf{C}(\mathbf{u})$ is the convective operator, $\mathbf{D}(\mathbf{u})$ the diffusive operator and the gravity has been dropped for the sake of clarity and compactness. The value k is the order of the linear k-step method, and the coefficients α_j and β_j are function of the linear multistep method. As will be detailed in Section 2.2.6, the integration algorithm employed for low-Mach flows calculates the scalars before the momentum equation. This allows to evaluate the density ρ^{n+k} from Eq. (2.29) before momentum calculation. Hence, once $\hat{\mathbf{u}}^{n+k}$ has been calculated from Eq. (2.50), the next step is to obtain the pressure field p^{n+k}. In order to do so, the divergence operator is applied to Eq. (2.51), where two possibilities arise:

$$\nabla \cdot (\rho^{n+k}\mathbf{u}^{n+k}) - \nabla \cdot (\rho^{n+k}\hat{\mathbf{u}}^{n+k}) = -\frac{\Delta t}{\alpha_k}\nabla^2 p^{n+k} \tag{2.52}$$

$$\nabla \cdot (\mathbf{u}^{n+k}) - \nabla \cdot (\hat{\mathbf{u}}^{n+k}) = -\frac{\Delta t}{\alpha_k}\nabla \cdot \left(\frac{1}{\rho^{n+k}}\nabla p^{n+k}\right) \tag{2.53}$$

Using both approaches a Poisson equation is obtained. The approach using Eq. (2.52) presents a matrix with constant coefficients while the second approach, employing Eq. (2.53), has a matrix with variable coefficients due to the presence of the density ρ^{n+k}, which means that at each iteration the matrix should be recomputed. This second approach requires the evaluation of the velocity divergence at time t^{n+k}, i.e. $\nabla \cdot (\mathbf{u}^{n+k})$, which can be approximated using Eq. (2.30). On the other hand, the first approach requires the estimation of the mass divergence $\nabla \cdot (\rho^{n+k}\mathbf{u}^{n+k})$ at time t^{n+k}, which is not known, but can be approximated using the continuity equation (Eq. (2.14)). In the present work the first approach is preferred due to the lower computational cost. Hence, in order to compute the mass divergence at time t^{n+k}, the mass conservation equation is employed:

$$\nabla \cdot (\rho^{n+k}\mathbf{u}^{n+k}) = -\left.\frac{\partial \rho}{\partial t}\right|^{n+k} \tag{2.54}$$

As reported by several authors [8, 26, 27], this term introduces numerical instabilities and has not been found any closed form well suited to approximate the time derivative. In this work the second-order backward approximation proposed by Nicoud [26] has been employed:

$$\left.\frac{\partial \rho}{\partial t}\right|^{k} = \frac{((\Delta t^k + \Delta t^{k-1})^2 - (\Delta t^k)^2)\rho^k - (\Delta t^k + \Delta t^{k-1})^2\rho^{k-1} + (\Delta t^k)^2\rho^{k-2}}{\Delta t^k \Delta t^{k-1}(\Delta t^k + \Delta t^{k-1})} \tag{2.55}$$

where in all the superscripts the value n has been neglected for compactness, so $\phi^k \equiv \phi^{n+k}$.

2.2.6 Temporal integration

The temporal integration scheme employed to solve the Low-Mach number transport equations presented in Section 2.2.2 is based on a Predictor-corrector scheme [28]. For the transported scalar, the predictor step is integrated using a second-order Adams-Bashforth integration scheme, while a second-order Crank-Nicolson is employed in the corrector step. On the other hand, both predictor and corrector steps of momentum equation are integrated using a second-order Adams-Bashforth scheme. As was mentioned in the previous section, the pressure-velocity coupling is solved by means of the Fractional Step projection method described by Nicoud [26], where a Poisson matrix with constant coefficients is employed.

The presented temporal integration method is a variant of the predictor-corrector scheme introduced by Najm et al. [29]. The main difference between both integration strategies is that in the present work the density is calculated through the gas state law and the scalars, i.e. the enthalpy h and the species mass fractions Y_k, are transported. This integration strategy is similar to the one followed by Lessani and Papalexandris [8], where the energy equation is solved using the temperature instead of the enthalpy, or the scheme named *algorithm A1* presented by Knikker [27], where a semi-implicit third-order Runge-Kutta/Crank-Nicolson scheme is employed and the energy equation is solved in its temperature form.

In the present work, the scalar transport equations are solved in non-conservative form:

$$\frac{\partial \rho\phi}{\partial t} + \nabla \cdot (\rho \mathbf{u}\phi) = \rho \left(\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla\phi \right) + \phi \left(\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) \right) \tag{2.56}$$

where the second term in the right-hand side is null because it coincides with the mass conservation equation. Hence, using Eq. (2.56), Eqs. (2.18) and (2.20) are rewritten as:

$$\rho \frac{\partial h}{\partial t} + \rho \mathbf{u} \cdot \nabla h = \frac{Dp}{Dt} - \nabla \cdot \dot{\mathbf{q}} + \boldsymbol{\tau} \nabla \cdot \mathbf{u} + \dot{\mathcal{Q}} \tag{2.57}$$

$$\rho \frac{\partial Y_k}{\partial t} + \rho \mathbf{u} \cdot \nabla Y_k = \nabla \cdot (\rho \mathbf{V_k} Y_k) + \dot{w}_k \tag{2.58}$$

This is because the scalars must be transported in non-conservative form when explicit temporal algorithms are employed, since the density is calculated a posteriori. As demonstrated by Knikker [27], the mass conservation equation, the energy conservation equation and the gas state law only can be fulfilled simultaneously if an implicit temporal scheme is employed. In the algorithm employed in the present work, the mass conservation constraint is fulfilled, but the energy conservation is not satisfied.

The steps to be performed at each iteration to advance from a time t^n to t^{n+1}, where $t^{n+1} = t^n + \Delta t$ are detailed below. The intermediate pseudo-implicit time of predictor-corrector schemes is noted as t^*.

Predictor step

1. Advance the transported scalars using a second-order Adams-Bashforth scheme:

$$\rho^n \frac{\phi^* - \phi^n}{\Delta t} = \frac{3}{2}\left(\rho^n \frac{\partial \phi}{\partial t}\bigg|^n \right) - \frac{1}{2}\left(\rho^{n-1} \frac{\partial \phi}{\partial t}\bigg|^{n-1} \right) \tag{2.59}$$

   where in this case $\phi = Y_k, h$.

2. If the thermodynamic pressure P_0 is not constant (closed domain) evaluate it from Eq. (2.33).

3. Evaluate the temperature T^* from the enthalpy h^* and mass fraction concentrations Y_k^*.

4. Evaluate the density ρ^* using the state Eq. (2.29) from the temperature T^*.

5. Following, momentum equation is solved be means of the fractional step method:

   (a) First, calculate the pseudo-velocity $\hat{\mathbf{u}}^*$ from Eq. (2.50) applying the second-order Adams-Bashforth scheme using the last computed density value ρ^*:

$$\frac{\rho^* \hat{\mathbf{u}}^* - \rho^n \hat{\mathbf{u}}^n}{\Delta t} = \frac{3}{2}\left(\frac{\partial(\rho \mathbf{u})}{\partial t}\bigg|^n \right) - \frac{1}{2}\left(\frac{\partial(\rho \mathbf{u})}{\partial t}\bigg|^{n-1} \right) \tag{2.60}$$

   (b) Then the Poisson equation (Eq. (2.52)) is solved, obtaining the pressure field p^*:

$$\nabla \cdot (\rho^* \mathbf{u}^*) - \nabla \cdot (\rho^* \hat{\mathbf{u}}^*) = -\Delta t \nabla^2 p^* \tag{2.61}$$

   where the mass divergence at time t^* is obtained from the mass conservation Eq. (2.54), which is approximated using Eq. (2.55):

$$\nabla \cdot (\rho^* \mathbf{u}^*) = - \left.\frac{\partial \rho}{\partial t}\right|^* \tag{2.62}$$

   (c) Once the pressure field is obtained, the velocity field is calculated using Eq. (2.51):

$$\rho^* \mathbf{u}^* = \rho^* \hat{\mathbf{u}}^* - \Delta t \nabla p^* \tag{2.63}$$

Corrector step

1. In the corrector step, the scalars are computed by means of a second-order Crank-Nicolson scheme:

$$\rho^* \frac{\phi^{n+1} - \phi^n}{\Delta t} = \frac{1}{2}\left(\rho^* \left.\frac{\partial \phi}{\partial t}\right|^*\right) + \frac{1}{2}\left(\rho^n \left.\frac{\partial \phi}{\partial t}\right|^n\right) \tag{2.64}$$

where, as in the predictor step, ϕ stands for the mass fractions species Y_k and the enthalpy h.

2. If the thermodynamic pressure P_0 is not constant (closed domain) evaluate it from Eq. (2.33).

3. Evaluate the temperature T^{n+1} from the enthalpy h^{n+1} and mass fraction concentrations Y_k^{n+1}.

4. Evaluate the density ρ^{n+1} using the state Eq. (2.29) from the temperature T^{n+1}.

5. The momentum equation is solved be means of the fractional step method:

   (a) The pseudo-velocity $\hat{\mathbf{u}}^{n+1}$ is obtained from Eq. (2.50) applying the second-order Adams-Bashforth scheme. The difference with the predictor step is that now the density ρ^{n+1} is available:

$$\frac{\rho^{n+1}\hat{\mathbf{u}}^{n+1} - \rho^n \hat{\mathbf{u}}^n}{\Delta t} = \frac{3}{2}\left(\left.\frac{\partial(\rho \mathbf{u})}{\partial t}\right|^n\right) - \frac{1}{2}\left(\left.\frac{\partial(\rho \mathbf{u})}{\partial t}\right|^{n-1}\right) \tag{2.65}$$

   Note that the right-hand size of this equations is exactly the same as in the predictor step. Therefore, it is not required any new function evaluation, since these terms have already been calculated.

   (b) The pressure equation field p^{n+1} is obtained solving the Poisson Eq. (2.52):

$$\nabla \cdot (\rho^{n+1}\mathbf{u}^{n+1}) - \nabla \cdot (\rho^{n+1}\hat{\mathbf{u}}^{n+1}) = -\Delta t \nabla^2 p^{n+1} \tag{2.66}$$

   As previously, the mass divergence at time t^{n+1} is obtained using the mass conservation Eq. (2.54), which is approximated by means of Eq. (2.55):

$$\nabla \cdot (\rho^{n+1}\mathbf{u}^{n+1}) = -\left.\frac{\partial \rho}{\partial t}\right|^{n+1} \tag{2.67}$$

   (c) Finally, the velocity field at the new time-step $\mathbf{u}^{n+1}$ is obtained employing Eq. (2.51):

$$\rho^{n+1}\mathbf{u}^{n+1} = \rho^{n+1}\hat{\mathbf{u}}^{n+1} - \Delta t \nabla p^{n+1} \tag{2.68}$$

2.3 Combustion modelling

Combustion processes are the result of chemical reactions involving different substances coexisting in a gas mixture under specific conditions. As detailed in Section 2.2.1, the kinetics of combustion chemical reactions are complex processes described by the empirical Arrhenius law. Moreover, the species and the reactions that describe a specific chemical reaction must be obtained experimentally. Therefore, the decision of which species and reactions must be accounted to properly describe a chemical process is an issue that must be decided by the experimentalists that creates the specific chemical scheme.

Some chemical mechanisms are very detailed and involve tens of species and hundreds of chemical reactions. However, this implies that in CFD simulations tens of conservation equations must be solved (one for each specie), and the calculation of mass reaction rates requires the evaluation of hundreds of chemical reactions, becoming many times computationally unaffordable. On the other hand, using simpler chemical mechanisms may reduce the number of equations to solve, but usually increase the complexity of the expressions describing the reaction rates, resulting in a stiffer set of equations. Moreover, some reduced mechanisms also involves negative exponents that could generate numerical problems. Moreover, in simulations of combustion processes using RANS or LES, modelling of the sub-grid phenomena occurring between flow and chemistry is required, increasing even more the complexity of the computations.

Therefore, it is not surprising that these difficulties have motivated the development of combustion models. The aim of these combustion models is to reduce the computational cost and the numerical complexity associated with the calculation of detailed chemical reactions, in order to make affordable the simulation of combustion processes.

Mainly, there are two types of combustion models: the ones based on the tabulation of the chemistry, and the models based on automatic chemistry reduction [10]. The former are based on the pre-computation and storage of the chemical reactions in look-up tables as function of a reduced set of variables. Then, during CFD simulations only these variables are solved and employed in order to access to the look-up table and recover all the information about the state of the chemical reaction. Examples of this type of models are the Intrinsic Low Dimensional Manifold (ILDM) [30, 31], the Flame Prolongation of ILDM (FPI) [32], and the Flamelet Generated Manifold (FGM) [33, 34].

Regarding the automatic chemistry reduction models, they arose aiming to overcome some of the drawbacks of tabulated-chemistry models. When look-up tables of many dimensions are used, a large number of multi-linear interpolations are needed, and even memory allocation can be a problem. At the end, only a small portion of the look-up table is really accessed during the computational time, since the thermophysical conditions of combustion tend to remain similar. This fact is exploited by these methods, that generate the chemical table on the fly during the computational time. At the beginning of the simulation the look-up table is empty. As the simulation runs, chemical reactions taking place in the computational domain are stored as function of the mass fractions and enthalpy (or temperature). If the conditions are the first time that occur at the simulation, the chemical reaction is computed and added to the look-up table. On the other hand, if that conditions have existed before, the chemical reaction is read from the dynamic look-up table instead of being computed. Hence, the chemical table grows dynamically, and as the simulation advance more chemical reactions are just obtained from the tabulated data and less are calculated. This second group of models were proposed by Pope [35] and Yang and Pope [36], and are known as In Situ Adaptive Tabulation (ISAT).

In Section 2.3 a combustion model based on tabulated chemistry is presented. A closure to model sub-grid interaction between flow and chemistry in LES modelling is also presented. These models have been implemented in the in-house CFD code TermoFluids.

2.3.1 Progress-Variable Model

The combustion model presented and developed in the present section is the **Progress-Variable** (PV) model. This combustion model falls in the spirit of the method proposed by Pierce and Moin [37], where the main idea is as follows: instead of solving a transport equation for each species k, and compute the chemical reaction rates using the finite-rates chemistry presented in Section 2.2.1, the model adopts an indirect mapping approach, where all the detailed chemistry is computed previously and tabulated as function of a reduced system of tracking scalars. In laminar cases, or turbulent flames solved using DNS, only two tracking scalars are considered: the mixture fraction (Z), which tracks the mixing between the fuel and the oxidizer, and the term denoted as progress variable (Y_c), which follows the global reaction state for each mixture fraction. This progress-variable is defined as a weighted linear combination of the species mass fraction:

$$Y_c = \sum_{k}^{N} \xi_k Y_k \tag{2.69}$$

where ξ_k is the weight of the mass fraction of species k. Different definitions of this value have been proposed in the literature, even using the temperature. Nonetheless, there exists no a general definition for this progress-variable, and it must be defined for each case in such way that the progress of the chemical reaction should be captured in the best possible way, i.e., the mass fractions of species that clearly track the evolution of the combustion process must be selected. The final results obtained with this model are dependent on the definition of this progress-variable [38]. Hence, a proper choice is a crucial aspect for obtaining accurate results.

Therefore, apart from continuity and momentum equations, this model requires to solve two additional scalar transport equations (mixture fraction and progress-variable), instead of the $N + 1$ equations (energy and species mass fraction) needed when finite-rate chemistry is considered. The two required transport equations are:

$$\frac{d\rho Z}{dt} + \nabla \cdot (\rho \mathbf{u} Z) = \nabla \cdot (\rho D_Z \nabla Z) \tag{2.70}$$

$$\frac{d\rho Y_c}{dt} + \nabla \cdot (\rho \mathbf{u} Y_c) = \nabla \cdot (\rho D_{Y_c} \nabla Y_c) + \dot{w}_{Y_c} \tag{2.71}$$

where D_Z and D_{Y_c} are the mixture fraction and the progress-variable diffusivities respectively. The mixture fraction diffusivity is calculated assuming a unit Lewis number, i.e., $Le_z = {}^{\lambda}\!/\!_{(\rho D_Z c_p)} = 1$, and the diffusivity of the progress-variable is considered equal to mixture fraction diffusivity, $D_{Y_c} = D_Z$. The progress-variable reaction rate $\dot{w}_{Y_c}$ is defined, analogously to the progress-variable Y_c, as the weighted linear combination of the reaction rates of the species forming the progress-variable $\dot{w}_{Y_c} = \sum_k^N \xi_k \dot{w}_k$.

The main difference between the model presented in this thesis and the one proposed by Pierce and Moin, is that while in the latter the detailed chemistry is calculated solving a one-dimensional reactive-diffusion equation, in the present model the mapping functions, which describe all the detailed chemical process, are obtained solving a homogeneous gas mixture in a closed system. This approximation is particularly well adapted for auto-ignition modelling [39, 40]. In this case, the $0D$ spatially homogeneous reactor is considered an adiabatic system with constant pressure. Therefore, the required governing equations are:

$$\rho \frac{dY_k}{dt} = \dot{w}_k \qquad\qquad \frac{dh}{dt} = 0 \tag{2.72}$$

where the enthalpy, $dh = de + vdp + pdv$, is constant for isobaric ($dp = 0$) and adiabatic systems ($de + pdv = 0$).

In order to generate the solutions table, the homogeneous reactor is solved as function of the mixture fraction Z using as initial conditions a linear interpolation between the conditions of the fuel and the oxidizer of the case that is going to be simulated using CFD.

$$\phi^{t=0} = \phi_{oxid.} + Z(\phi_{fuel} - \phi_{oxid.}) \tag{2.73}$$

After computing the solutions for all the discrete mixture fraction values Z, a look-up table with the solution map is obtained. As said, this table will be function of the two tracking scalars, Z and Y_c, and consequently, can be considered an *Equation of State* (EoS) of type:

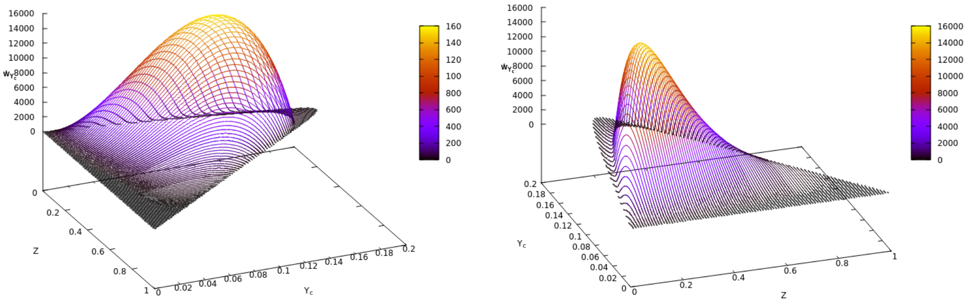


Figure 2.3: Reaction rate of the progress variable $\dot{w}_{Y_c}$ as function of the mixture fraction Z and the progress variable Y_c for a flame of hydrogen H_2.

$$\phi = f(Z, Y_c) \tag{2.74}$$

where ϕ represents any thermo-chemical variable $(\rho, T, Y_k, \dot{w}_k, ...)$, since Z and Y_c define the state of the reaction. For example, in Fig. 2.3 the solution map for the reaction rate of the progress variable $\dot{w}_{Y_c}$ as function of the mixture fraction Z and the progress-variable Y_c is given, so clearly $\dot{w}_{Y_c} = f(Z, Y_c)$. This case corresponds to a combustion of a hydrogen jet H_2 surrounded by a hot air coflow. Each line of the $3D$ plot corresponds to a solution of the homogeneous reactor for a specific mixture fraction Z. In this example, the dimension Z has been discretized in 100 points. As can be seen, for each solved mixture fraction value there is a different final value of the progress variable. This final value is the equilibrium value of the progress variable Y_c^{eq}, which is reached in fully burnt gases, and is defined following the same linear interpolation of the progress-variable, it is, $Y_c^{eq} = \sum_k^N \xi_k Y_k^{eq}$. This number is useful in order to obtain the normalized progress-variable c:

$$c = \frac{Y_c}{Y_c^{eq}} \tag{2.75}$$

2.3.2 Presumed Conditional Moment (PCM)

When solving combustion laminar cases, or in DNS modelling, the PV model presented in the previous section can be applied straightforwardly. However, in LES modelling, analogously to the filtered mass fraction equation (Eq. (2.42)), the transport equation for the filtered progress variable reads as:

$$\frac{\partial \overline{\rho}\widetilde{Y_c}}{\partial t} + \nabla \cdot (\overline{\rho}\widetilde{\mathbf{u}}\widetilde{Y_c}) = \nabla \cdot \left(\overline{\rho}(\overline{D_{Y_c}} + \overline{D_{Y_c,t}})\widetilde{Y_c}\right) + \overline{\rho}\widetilde{\dot{w}_{\rho,Y_c}} \tag{2.76}$$

where the filtered reaction rate of the progress-variable, $\widetilde{\dot{w}_{\rho,Y_c}}$, appears. This term is not closed and have to be modelled. Note that the progress variable reaction rate in Eq. (2.76) has been density-weighted, which allows to apply naturally to this term the Favre-filtering operation detailed above. In the same way, it is required to solve the transport equation for the filtered mixture fraction $\widetilde{Z}$:

$$\frac{\partial \overline{\rho}\widetilde{Z}}{\partial t} + \nabla \cdot (\overline{\rho}\widetilde{\mathbf{u}}\widetilde{Z}) = \nabla \cdot \left(\overline{\rho}(\overline{D_Z} + \overline{D_{Z,t}})\widetilde{Z}\right) \tag{2.77}$$

Consequently, apart from the transported scalars, all the thermo-chemical properties tabulated in the solution tables for non-turbulent cases must be redefined in its filtered form, i.e. $(\overline{\rho}, \widetilde{T}, \widetilde{Y_k}, \widetilde{\dot{w}_k}, ...)$. In order to do so, in this section the model named Presumed Conditional Moment (PCM) is introduced. This model can be considered as a simplified version of the Conditional Moment Closure (CMC) model, since in PCM the conditional moments are presumed instead of being calculated. The CMC model was originally developed by Klimenko and Bilger [41–43]. In CMC the conditional filtered reaction scalars, conditioned on the mixture fraction space, are solved. On the other hand, when the PCM model is applied to the PV model explained in the previous section, these conditional moments are presumed for the progress-variable Y_c and the mixture fraction Z, which are the transported scalars. In the PCM model, the distribution of the transported scalars (i.e. mixture fraction Z and progress variable Y_c) is modelled using a probability density function (pdf). Therefore, any filtered thermo-chemical quantity $\widetilde{\phi}$ is obtained integrating the product of the tabulated quantity ϕ obtained from Eq. (2.74) and the joint pdf of Y_c and Z, it is:

$$\widetilde{\phi} = \int_0^1 \int_0^{Y_c^{eq}} \phi(Y_c, Z)\widetilde{P}(Y_c, Z) dY_c dZ \tag{2.78}$$

where $\widetilde{P}(Y_c, Z)$ is the Favre-averaged joint pdf of Y_c and Z. This density-weighted pdf is related to the unweighted one using Eq. (2.36):

$$\overline{\rho}\widetilde{P}(Y_c, Z) = \overline{\rho P(Y_c, Z)} \tag{2.79}$$

It should be stressed that this joint pdf $P(Y_c, Z)$ is also a function of space and time, i.e. $P(Y_c, Z; \mathbf{x}, t)$, but for the sake of simplicity this additional notation is neglected. Employing the Baye's theorem [44] any joint pdf of two independent variables can be expressed as the product of the conditional pdf of one variable over the other and the marginal pdf of this second one:

$$\widetilde{P}(Y_c, Z) = P(Y_c|Z)\widetilde{P}(Z) \tag{2.80}$$

where $P(Y_c|Z)$ is the conditional probability of Y_c for a given mixture fraction Z. Moreover, if a weak dependence of Y_c from Z is assumed, meaning that the two variables are considered statistically independent, the previous conditional pdf can be expressed as the marginal pdf $P(Y_c|Z) \approx P(Y_c)$. Therefore, under these assumptions, Eq. (2.78) can be rearranged as:

$$\widetilde{\phi} = \int_0^1 \int_0^{Y_c^{eq}} \phi(Y_c, Z)P(Y_c)\widetilde{P}(Z)dY_c dZ \tag{2.81}$$

obtaining an expression for the EoS of the filtered variables that is function of two marginal pdfs and not a joint one. All the thermo-chemical quantities can be recovered employing Eq. (2.81), except the Favre-averaged density, which must be obtained as:

$$\frac{1}{\overline{\rho}} = \int_0^1 \int_0^{Y_c^{eq}} \frac{1}{\rho(Y_c, Z)} P(Y_c)\widetilde{P}(Z)dY_c dZ \tag{2.82}$$

Thus, the next requirement is to obtain the pdfs defining the distributions for the mixture fraction and the progress-variable. As it has been demonstrated by several authors, the $\beta - pdf$ is a good representation of the statistical distribution for conserved scalars [10, 45, 46]. Therefore, the marginal pdf of the mixture fraction $\widetilde{P}(Z)$ is modelled as a $\beta - pdf$, which is calculated as [47]:

$$\widetilde{P}(Z) = \widetilde{\beta}(Z) = \frac{1}{B(a,b)} Z^{a-1}(1-Z)^{b-1} = \frac{\Gamma(a+b)}{\Gamma(a)+\Gamma(b)} Z^{a-1}(1-Z)^{b-1} \tag{2.83}$$

where $B(a, b)$ is the beta function and acts as a normalization factor ensuring that the total probability is equal to unity, and $\Gamma(x)$ is the gamma function. Both functions are defined by the following integrals:

$$B(a,b) = B(b,a) = \int_0^1 t^{a-1}(1-t)^{b-1}dt \tag{2.84}$$

$$\Gamma(x) = \int_0^\infty t^{x-1}e^{-t}dt \tag{2.85}$$

The parameters a and b are obtained from the mean and the variance of the mixture fraction, $\widetilde{Z}$ and $\widetilde{Z''^2}$ respectively:

$$a = \widetilde{Z}\gamma \tag{2.86}$$

$$b = (1-\widetilde{Z})\gamma \tag{2.87}$$

$$\gamma = \frac{\widetilde{Z}(1-\widetilde{Z})}{\widetilde{Z''^2}} - 1 \geq 0 \tag{2.88}$$

On the other hand, the distribution of the progress-variable $P(Y_c)$ is modelled employing a $\delta - pdf$, i.e. $P(Y_c) = \delta(Y_c - Y_c^*)$, which means that the sub-grid fluctuations of the progress variable are neglected. A model where the marginal pdf $P(Y_c)$ is modelled using a $\beta - pdf$ is presented by Michel et al. [40]. In that work, it is shown that this approximation tends to generate unphysical situations, because at low values of the progress variable, when autoignition begins, the simulations show large values for the variance of the progress variable, which means that part of the igniting gases are burned instantly for all mixture fractions. In addition, as stated by Raman et al. [48], a $\beta - pdf$ is not a good option for non conserved scalars. Hence, under the assumptions presented above, the filtered thermo-chemical quantities are finally calculated as:

$$\widetilde{\phi}(\widetilde{Z}, \widetilde{Z''^2}, Y_c^*) = \int_0^1 \int_0^{Y_c^{eq}} \phi(Y_c, Z)\delta(Y_c - Y_c^*)\widetilde{\beta}(Z; \widetilde{Z}, \widetilde{Z''^2})dY_c dZ \tag{2.89}$$

In order to define the value of the input progress-variable Y_c^*, and relate it to the filtered progress-variable $\widetilde{Y}_c$, which is the value available in LES, the relation proposed by Vervisch et al. [49] is employed. This relation approximates the unfiltered normalized progress-variable to the filtered one as:

$$\frac{Y_c^*}{Y_c^{Eq}(Z)} \approx \frac{\widetilde{Y}_c}{\widetilde{Y_c^{Eq}}} \quad ; \quad Y_c^* \approx \frac{\widetilde{Y}_c}{\widetilde{Y_c^{Eq}}} Y_c^{Eq}(Z) \tag{2.90}$$

where the filtered equilibrium value of the progress-variable is calculated following the expression:

$$\widetilde{Y_c^{Eq}} = \int_0^1 Y_c^{Eq}(Z)\widetilde{P}(Z)dZ = \int_0^1 Y_c^{Eq}(Z)\widetilde{\beta}(Z; \widetilde{Z}, \widetilde{Z''^2})dZ \tag{2.91}$$

and consequently, it is defined a $\widetilde{Y_c^{Eq}}$ value for each one of the discrete values of both mean mixture fraction $\widetilde{Z}$ and variance mixture fraction $\widetilde{Z''^2}$. This approximation is also used successfully by Michel et al. [40] and Kulkarni et al. [50]. From Eq. (2.90), it is obvious that although for the pdfs distribution a weak dependence of Y_c from

Z has been previously assumed, the input progress-variable Y_c^* in Eq. (2.89) is still conditioned by the mixture fraction. Thus, in order to emphasize this fact, this progress-variable conditioned on the mixture fraction is rewritten as:

$$Y_c|Z(Z;\widetilde{Z},\widetilde{Z''^2},\widetilde{Y_c}) \approx \frac{\widetilde{Y_c}}{\widetilde{Y_c^{Eq}}(\widetilde{Z},\widetilde{Z''^2})} Y_c^{Eq}(Z) \qquad (2.92)$$

Therefore, using this approximation, Eq. (2.89) can be finally rewritten as:

$$\widetilde{\phi}(\widetilde{Z},\widetilde{Z''^2},\widetilde{Y_c}) = \int_0^1 \phi(Y_c|Z,Z)\widetilde{\beta}(Z;\widetilde{Z},\widetilde{Z''^2})dZ \qquad (2.93)$$

meaning that for turbulent cases, the EoS of the solution map containing all the thermo-chemical properties of the modelled combustion process reads as:

$$\widetilde{\phi} = f(\widetilde{Z},\widetilde{Z''^2},\widetilde{Y_c}) \qquad (2.94)$$

Finally, the expression employed to calculate the filtered reaction rate of Eq. (2.76) is:

$$\widetilde{\dot{w}}_{Y_c} = \bar{\rho}\widetilde{\dot{w}_{\rho,Y_c}} = \bar{\rho}\int_0^1 \frac{\dot{w}_{Y_c}(Y_c|Z,Z)}{\rho}\widetilde{\beta}(Z;\widetilde{Z},\widetilde{Z''^2})dZ \qquad (2.95)$$

Furthermore, as can be derived from the previous presentation, this model requires de evaluation of the filtered mixture fraction sub-grid variance $\widetilde{Z''^2}$, since is an input parameter of the $\beta-pdf$ employed to model the statistical distribution of the mixture fraction. It can be obtained from two different approaches: the first one consists on computing a transport equation for $\widetilde{Z''^2}$, and the second possibility is to model the filtered sub-grid variance algebraically. For the work carried out in the present thesis, this second option has been chosen. Specifically, the following dynamic model has been employed:

$$\bar{\rho}\widetilde{Z''^2} = C_v\bar{\rho}\Delta^2|\nabla\widetilde{Z}|^2 \qquad (2.96)$$

where value C_v is calculated using the *Leonard term expansion dynamic model* (LED) proposed by Balarac [51], and Δ represents the LES filter width. The test filter $\widehat{\Delta}$ used in the present work is an unstructured top-hat filter constructed from a stencil of cells connected by faces.

2.4 The Cambridge autoignition experiment

The autoignition process of a hydrogen jet into a preheated turbulent air stream is numerically studied. The test case consists of a fuel jet, a mixture of hydrogen H_2 and nitrogen N_2 , flowing through a nozzle placed at the centre of a co-flowing air stream (see Fig. 2.4). Two different ways of computing the chemical reactions are studied: a first one employing the Progress-Variable (PV) model, where the turbulence-chemistry interactions are modelled using the Presumed Conditional Moment (PCM), and a second one using finite-rates (FR) chemistry where perfect mixing at sub-grid scale level is assumed, so no closure model is required for the mass reaction rate source term. The main objective of this section is to assess the capacity of the presented models to predict accurately the autoignition behaviour of the proposed experimental configuration.

The obtained results are in good agreement with the experimental data. The PV-PCM model reproduces satisfactorily the physical behaviour found in the experiments, although the model tends to under-predict the autoignition length. The results obtained employing the finite-rate model also capture pretty well the autoignition phenomenology observed experimentally and the autoignition lengths are closer to those obtained experimentally.

Introduction

Nowadays, accurate prediction of the autoignition behaviour of non-premixed or partially premixed flows is a field of great importance in many industrial applications, such as diesel engines, gas turbines or ramjets. Therefore, having CFD codes capable of simulating autoignition phenomena offer a very interesting way to improve and enhance these devices. Moreover, numerical simulations can offer data and information that is difficult or impossible to obtain via experiments, allowing a better understanding of the physical phenomena behind the autoignition process.

The autoignition of a non-premixed or partially premixed flow is a phenomena clearly characterised by two different stages. First, when the fuel and the oxidizer are injected in the combustion chamber, both start to mix and the chemical species begin to react generating a *pool* of reactants, but without a significant heat release. Afterwards, when the mixture reaches the critical temperature (due the heat release in the first stage) a second step starts, where the reactants are quickly consumed and the products of the reaction are generated, with a remarkable heat release, increasing the temperature very fast. This second step, where the autoignition begins, takes place inside *spots* or *kernels*, which are small regions having the proper conditions (temperature and mass fractions), and could act as a bridge to auto-ignite the near regions, or could be quenched, depending on the flow and the species concentrations. Therefore, mixing between fuel and oxidizer is a crucial point in order to determine correctly the place where autoignition takes place. In contrast to Unsteady Reynolds Averaged Navier-Stokes (U-RANS) simulations, where all the energy spectra is modelled, in Large Eddy Simulation (LES) the large scale structures of the flow are well resolved, and only the sub-grid scales (sgs) are modelled, increasing notably the accuracy, but with lower computational requirements compared to Direct Numerical Simulations

(DNS). Therefore, since the solution of the transported variables plays a key role in the autoignition behaviour, special interest is turned on the accurate solution of transport equations. In order to avoid numerical diffusion, conservative discretizations are used for the differential operators [52]. These discretizations conserve kinetic energy in the momentum equations and offer accurate approximations.

In chemically reacting turbulent flows, turbulence has a strong influence in the autoignition delay time, since turbulence affects the transport of both fuel and oxidizer, i.e. the mixture fraction. An increase in the turbulence intensity tends to enhance mixing and the autoignition length is shortened. Nevertheless, if there is an excess of turbulence, it could dilute too much the reactants and quench the kernel before this could propagate. Hence, in order to predict accurately the auto-ignition phenomena in a turbulent flow, a model which deals properly the interactions at sub-grid level between the flow and the chemistry is required.

This section is aimed to assess the capability of the Progress-Variable (PV) model presented above [37]. In order to reproduce the autoignition behaviour of hydrogen, the turbulent chemical-flow interaction is closed by means of the Presumed Conditional Moment (PCM) approach. The experiment performed by Markides and Mastorakos [53] has been numerically studied and the experimental results are compared against those obtained by simulation. Furthermore, a second objective is to assess the influence of the sub-grid interaction between flow and chemistry in the autoignition of the above mentioned H_2/air jet. Therefore, for this second target, a new set of numerical simulations are carried out, where the finite-rate chemistry model assuming a perfect mixing at sub-grid scale is employed. This assumption means that the sub-grid chemistry-flow interaction is not modelled.

Finally, the underlying objective is to assess the capability of the in-house CFD code TermoFluids to reproduce accurately autoignition events using unstructured meshing, aiming to be able to simulate ignition phenomena in complex geometries. Unstructured meshing allows a better and smarter discretising strategy, allowing an easy refinement in areas or regions of particular interest or crucial importance. Furthermore, most of the industrial combustion chambers show complex geometries and the use of unstructured meshes is unavoidable.

Description of the Cambridge autoignition experiment

The experimental configuration presented in [53] consists in a fuel jet, a mixture of hydrogen H_2 and nitrogen N_2, flowing through a nozzle placed at the centre of a co-flowing air stream. The air is preheated at different temperatures in order to study its impact in the autoignition behaviour of hydrogen under this geometrical configuration. The co-flowing air is forced to pass through a perforated plate to promote turbulence. The perforated plate (3.0 mm holes and 44 % blockage) is located 63 mm upstream of the fuel nozzle in order to allow turbulence to develop. The fuel nozzle has a diameter of 2.25 mm and is thin-walled (0.32 mm). The main test section consists of a 500 mm

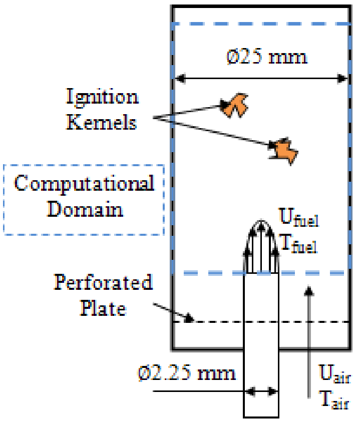


Figure 2.4: Schematic representation of the experiment presented by Markides and Mastorakos [53].

Table 2.1: Cases studied.

Case	U_{jet} [m/s]	U_{air} [m/s]	T_{jet} [K]	T_{air} [K]
0	26.0	26.0	750	950
1	26.0	26.0	750	955
2	26.0	26.0	750	960
3	26.0	26.0	750	980

long and 25 mm inner diameter vacuum insulated quartz tube.

The experiment was performed over a wide range of operating conditions and four regimes were identified, namely *No ignition*, *Random spots*, *Flashback* and *Lifted flame*. In the present work, four different operating conditions will be analysed (see Table 2.1). For all the studied cases, the fuel is a mixture of H_2 and N_2 ($Y_{H_2} = 0.13$, $Y_{N_2} = 0.87$) and the co-flow oxidizer is air ($Y_{O_2} = 0.233$, $Y_{N_2} = 0.767$). Cases 0, 1 and 2 fall inside the *Random spots* regime, where auto-ignition kernels appear but are quenched before they could act as a flame anchoring point or a flashback, and are convected out of the domain. The last case belongs to the *Flashback* regime, where a first autoignition spot appears, and then a flashback occurs, finally obtaining a normal jet diffusion flame.

Simulation set-up

The following study is carried out under the framework of LES modelling. Hence, the *unresolved Reynolds stresses* of Eq. (2.40) must be closed using a turbulence model. In the present work, the wall-adapting eddy viscosity sub-grid scale model (WALE) [21] is employed. Previous LES studies about this autoignition case were mainly carried out using the dynamic Smogarinsky-Lilly model [50, 54]. The WALE model is based on the square of the velocity gradient tensor. The sub-grid scale viscosity obtained with this model takes into account the strain and the rotation rate of the smallest resolved turbulent fluctuations. A relevant feature of this model is its capability of switching off in two-dimensional flows, laminar flows and when the length-scale is in the range of $Re^{-3/4}$. Additionally, it also has a cubic behaviour near walls with respect to the normal direction of the wall.

As mentioned before, the chemical reaction mechanism is of crucial importance and plays a key role in the autoignition behaviour. In the present work the chemical reactions are modelled using a detailed reaction mechanism of 9 species and 21 reactions developed by Mueller et al. [7].

The PV combustion model requires the definition of the progress variable Y_c, which must be picked up as a linear combination of one or more thermo-chemical variables solved in the $0D$ spatially homogeneous reactor. Since in the present work a hydrogen-oxygen (H_2/O_2) reaction is being considered, an obvious choice is to select the main product of the chemical reaction, water H_2O, plus an intermediate species which could track the first stage of the autoignition process. In this case, the hydroperoxyl radical HO_2 is selected. Therefore, the progress variable used in this work is defined as:

$$Y_c = H_2O + HO_2 \tag{2.97}$$

The simulations are performed in a computational domain which starts at the exit of the nozzle and extends 135 mm in the downstream direction (see Fig. 2.4). The cylindrical domain is discretized using an unstructured mesh of approximately 400000 control volumes. More specifically, the domain is generated revolving in 16 planes a 2D plane with 25000 control volumes in the azimuthal direction. The mesh is refined in the air-fuel shear layer. All the simulations have been carried out using the same mesh and 64 CPUs. The Poisson equation is solved employing the FFT-based Poisson Solver presented by Borrell et al. [55]. The computations have been carried out on the UPC JFF cluster (Technical specifications: 40 cluster nodes, each node has two AMD Opteron with 16 cores for each CPU linked with 64 gigabytes of RAM memory and an infiniband QDR 4X network interconnection between nodes with latencies of 1.07 microseconds with a 40 Gbits/s bandwidth) [56].

In order to reproduce the turbulence generated by the perforated plate, an auxiliary non-reactive simulation is performed in an annular mesh, recreating the physical

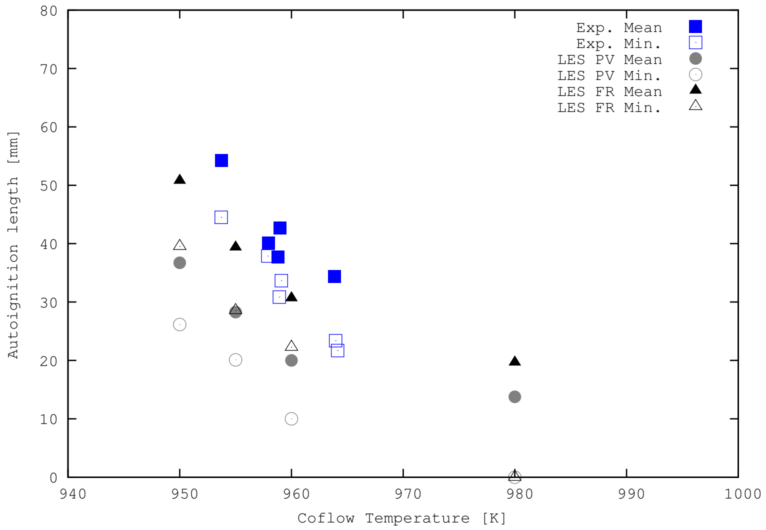


Figure 2.5: Mean and minimum ignition lengths.

domain upstream to the injector lips, where the plate is placed inside the domain using the immerse boundary technique [57]. The velocity field generated 63 millimetres downstream of the perforated plate is saved, and later injected in the simulated domain through the co-flowing section. This method allows a significant saving of computational resources during simulation time, and develops a realistic divergence-free velocity field, which is not possible using some synthetic turbulence generators based on digital filters. For the fuel inflow a laminar parabolic velocity profile is employed with $U_{mean} = U_{fuel}$. For the walls, a free-slip boundary condition is used, so the flow is not well resolved near the wall, but since all the phenomena of interest takes places far from the wall, a well resolved shear-layer near the wall is not of interest [54].

Results and discussion

The autoignition length obtained employing both models (PV-PCM and finite-rate chemistry) for all the cases under study (see Table 2.1) are depicted in Fig. 2.6. The ignition length is determined using as ignition criterion a rise of 1 % over the initial co-flow temperature. For both models, in cases 0, 1 and 2, the ignition length oscillates around a mean. This is caused by the appearance of random auto-ignition kernels, which are quenched and convected out of the domain before they can act as an anchoring point or derive in a flashback. This process where an ignition kernel appear, briefly grows, and then is quenched is repeated periodically along the simulation

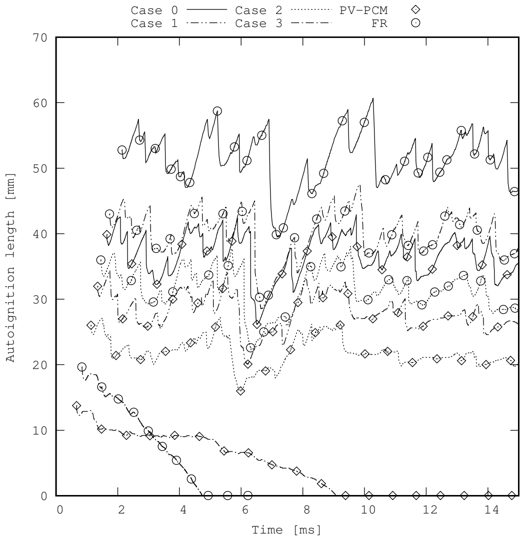
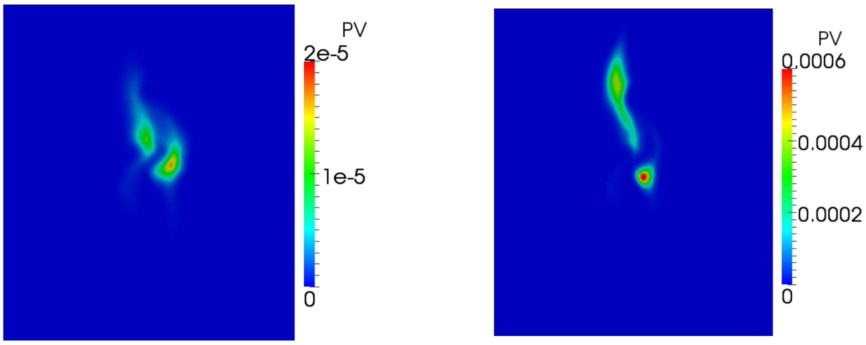


Figure 2.6: Evolution of ignition lenghts. *(The combination of lines and points relates the four cases and the two models.)*

time, resulting in the oscillatory auto-ignition length. Thus, these three cases clearly belong to the *random spots* regime previously described. In contrast, in case 3, once ignited, the ignition length decreases progressively as the flame propagates upstream. Therefore, this case falls inside the behaviour described for the *flashback* regime.

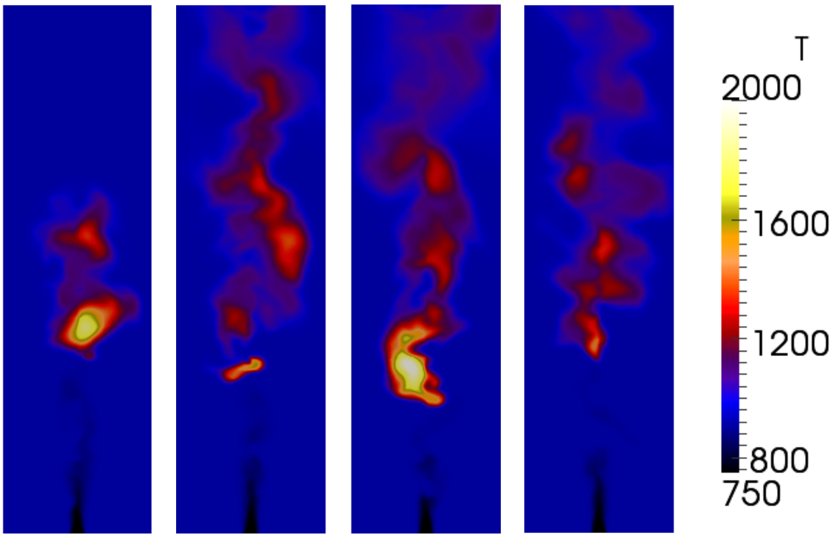
These results are in agreement with the ones obtained in the experiment carried out by Markides and Mastorakos where, for the conditions reproduced in the present work, the boundary between both regimes was found around a co-flow temperature of 965 K. Hence, both models are able to reproduce the physics behind the autoignition process of hydrogen in this configuration.

Although the PV-PCM model and the FR model present the same behaviour, the calculated ignition length is different. In Fig. 2.5, the mean and minimum autoignition lengths determined by both models are summarized and compared against the results obtained in the experiment (since case 3 is a *flashback* flame, the mean autoignition length plotted is the position where the first autoignition event takes place). At first glance, it can be seen that the FR model presents results closer to the experiment than the PV-PCM model, although no closure for the mass reaction rate has been employed in the former. As stated by Michel et al. [40], the PV-PCM model, where a $\beta - pdf$ for the distribution of mixture fraction and a $\delta - pdf$ for the progress variable distribution have been employed, tends to over-predict the reactivity of the progress



(a) Finite-rates (b) PV-PCM

Figure 2.7: Progress variable at same instant for case 3.



(a) 3 ms (b) 5 ms (c) 7 ms (d) 9 ms

Figure 2.8: Temperature evolution for case 0 at different time instants (PV-PCM model).

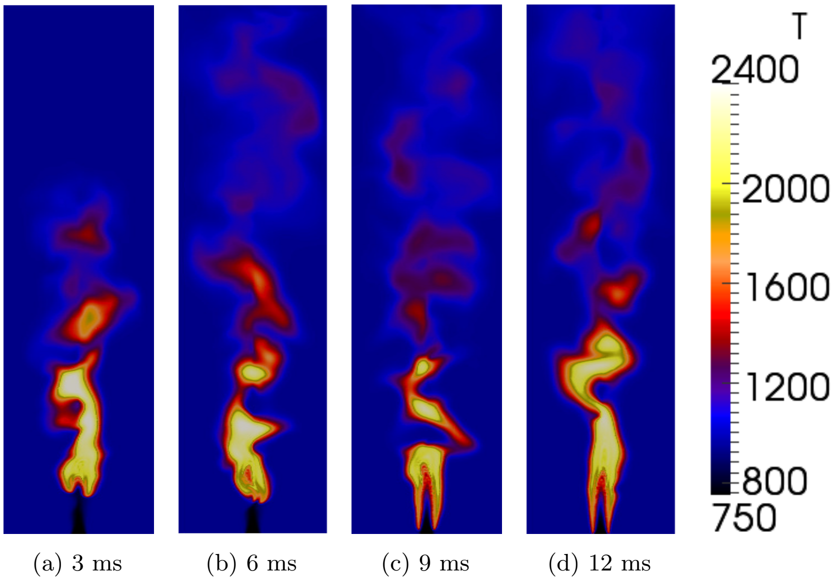


Figure 2.9: Temperature evolution for case 3 at different time instants (PV-PCM model).

variable, and therefore, predicts short ignition lengths. In the work presented by Ihme et al. [58], where a Flamelet Progress Variable (FPV) approach was employed, they arrived at a similar conclusion. Since the $\delta - pdf$ could only describe the first moment, it is impossible to capture the non-linear behaviour of the chemical reaction rate, resulting in larger $\widetilde{\dot{w}}_{Y_c}$ for many situations. In order to see an example of this fact, the value of the progress variable at the same instant is shown for both models in Fig. 2.7. Although the velocity and mixture fraction fields are almost equal, it can be seen that the progress variable differs in more than one order of magnitude.

The temporal evolution of case 0 is shown in Fig. 2.8. In the image sequence can be seen how random kernels appear but are quenched and carried out of the domain by the flow. Moreover, the temporal evolution for case 3 is shown in Fig. 2.9. As stated previously, in this case, once the hydrogen is ignited, the ignition kernel overcomes the convection of the flow, and the flame propagates upstream until it becomes an attached flame, in a clear *flashback* process. The lateral wave motion of the flame is related to the large turbulent scale of the co-flow. This behaviour has been obtained using both models, which is consistent with the experiment of Markides and Mastorakos [53], as well as the numerical results obtained in previous LES simulations [50, 54].

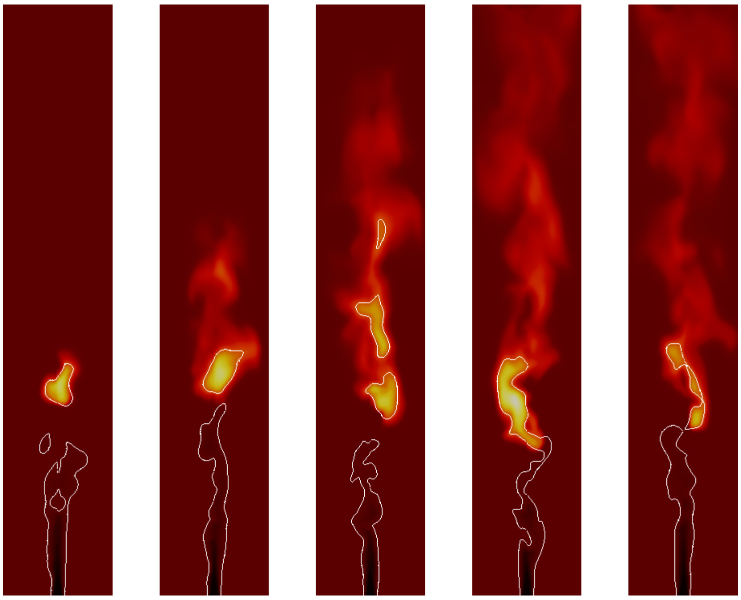


Figure 2.10: Instantaneous temperature field of case 0 at different times. The white line is the isoline indicating the most reactive mixture fraction, $Z_{mr} = 0.035$ in this case.

As reported in the experiment by Markides and Mastorakos [53], it is found that the autoignition kernels always appear inside regions enclosed by the most reactive mixture fraction Z_{mr}. Instantaneous temperature fields for case 0 are shown in Fig. 2.10. For this case, the most reactive mixture fraction has a value of $Z_{mr} = 0.035$, and all the ignition kernels emerge in regions enclosed by the isoline indicating the most reactive mixture fraction (see Fig. 2.10).

Apart from the simulation results, another important aspect to analyse is the numerical performance of both combustion models. As aforementioned, the finite-rate combustion model is very demanding computationally speaking due to two reasons: a transport equation for each species have to be solved and the resulting system of ODEs devoted to compute the mass reaction rates is stiff. For the current example, when using finite-rate chemistry, nine transport equations have to be solved and a very small time-step is required because the explicit time-integration scheme. On the other hand, the PV model is less computationally demanding and allow higher time-steps. For the current simulations, carried out in the JFF cluster on 64 CPUs, each one of the four cases using the PV model required 68 min of clock time to advance 1 ms the simulation time, i.e., 68 $\min_{cpu}/\mathrm{ms}_{sim}$. Regarding the simulations using the finite-rate model, the case 0 (coflow temperature of 950 K) required 525 $\min_{cpu}/\mathrm{ms}_{sim}$, and the case 3 (coflow temperature of 980 K) demanded 3634 $\min_{cpu}/\mathrm{ms}_{sim}$. Therefore, these cases have, respectively, a computational cost 7.7 and 53.66 times higher than the simulations carried out using the PV model. The differences in the computational time between the two cases using the FR model are due to the active presence of chemical reactions. For the case 0, which belongs to the *random spots* regime, there are no active chemical reactions during all the simulation time, since the ignition kernels are quenched after a short live-time. On the other hand, in the case 3, where an attached flame is developed, there are chemical reactions during all the simulation time, increasing the computational demand.

2.5 Load-balancing method for the evaluation of chemical reaction rates

The development and assessment of an efficient parallelization method for the evaluation of the reaction rates in combustion simulations is presented. Combustion simulations where the finite-rate chemistry model is employed have a heavy computational load for two reasons: in these simulations a transport equation for each species present in the chemical reaction mechanism have to be solved, and the resulting system of equations is stiff. Stiff sets of equations are a kind of ordinary differential equations (ODEs) characterized for presenting numerical instabilities, unless a very small time-step is employed. This limited time-step is inherent for both explicit numerical schemes as well as classical implicit integration methods employing fixed point iteration. Therefore, advanced implicit methods must be applied to obtain accurate solutions using reasonable time-steps. Nonetheless, as expected, these well-suited implicit methods for stiff equations demand higher computational resources than explicit or classical implicit methods. Therefore, in the present work a new algorithm aimed to enhance the numerical performance of the time integration of stiff equation systems in parallel combustion simulations is presented. The developed algorithm identifies the controls volumes where the system of equations is stiff. Then, they are integrated using Gear's method, while non-reactive control volumes, that do not present a stiff behaviour are integrated explicitly. Since reactive zones are close to the front flame, in parallel simulations only a few processors where the flame is live present a large amount of cells with stiff equations, while the other processors far from the flame can be integrated fully explicitly, creating a heavy imbalance in the computations. In order to solve this issue, an efficient dynamic load-balancing method of the computational load is implemented, increasing noteworthy the computational performance of the simulations, and consequently, reducing the computer time required to perform the numerical combustion studies.

2.5.1 Introduction

As previously detailed in Section 2.2.1, in the finite-rates chemistry model for transient numerical combustion simulations, the rates of production and destruction of all the species present in the domain are calculated by means of the empirical Arrhenius expression [10]. The resulting set of ordinary differential equations (ODEs) is stiff [59]. A stiff equation is characterized for having a smooth solution with a slow variation which numerical integration presents instabilities due to the presence of nearby solutions with very fast variations. One of the first studies addressing the numerical integration of stiff equations is the one presented by Curtiss and Hirschfelder [3]. In their paper, the following one-dimensional equation is presented as an example of stiff equation:

$$y' = \frac{y - G(x)}{a(x,y)} \tag{2.98}$$

where G(x) is a smooth well-behaved function. If the desired or reasonable integration step to integrate numerically this equation is Δx, the above equation will be stiff when:

$$\left| \frac{a(x,y)}{\Delta x} \right| \ll 1 \tag{2.99}$$

In order to clearly exemplify stiffness let define $G(x) = cos(x)$ and $a(x,y) = (-1/50)$. Employing these values, the solution curves for different initial conditions as well as the solution for $cos(x)$ are given in Fig. 2.11. As can be seen, there is a general solution close to $y \approx cos(x)$ where the map solution converges after a rapid transient phase. However, the nearby solutions present a very fast change in the slope of the derivative close to this general solution. For a certain value of x, small values of y have large positive slopes, while large values of y present big negative slopes. Close to the general solution $cos(x)$ there is a sudden and fast change in the slope of the map solution. These *fast*-changing solutions close to the general *slow*-varying solution are the reason of the stiffness. When using explicit numerical integration methods, the employed integration step Δx must be small enough to capture all these small-scales presenting fast variations, although the general solution have a much larger temporal scale which makes desirable employ a larger integration step, enough to capture accurately the solution, and allowing a reduction of the integration steps, and consequently, the computational cost. Otherwise, explicit integration methods will produce numerical instabilities leading to heavy oscillatory solutions or even causing the solution to blow up. For example, the numerical integration of the previous stiff equation using the first-order explicit Euler for different integration steps Δx is shown in Fig. 2.12. As can be seen, integration steps of $\Delta x = 1.95/50$ and $\Delta x = 1.85/50$ that should be enough to capture the solution after $x \approx 0.1$ present a heavy oscillatory solution. Very small integration steps should be employed to avoid this unstable

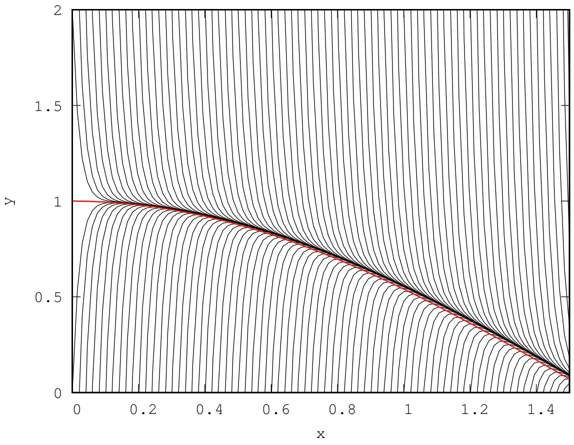


Figure 2.11: Solution map for $y' = -50(y - cos(x))$ (black lines) and solution for $cos(x)$ (red line).

behaviour. Hence, this is the main issue of stiff problems, the integration step is forced to be extremely small, much smaller than the one required to obtain accurate results for the general solution. Therefore, a large amount of integration steps, increasing the calculation effort, is needed.

The previous example illustrates the stiffness for one-equation. However, stiffness is a characteristic that could arise also when dealing with sets of more than one ODE. Stiff sets of equations are characterized by the presence of a wide range of time-scales in their variables during the integration time interval, and where the general solution of interest have a *slow* variation. At same time, this general solution is strongly influenced by the solutions of the variables with smallest time-scales that change very fast. Basically, this means that although the fastest (shortest) scales have a negligible effect in the final solution, and their accurate resolution is not necessary for a final accurate result, they must be solved in order to guarantee the numerical stability of the integration method. An illustrative example of stiff sets of equations is the one proposed by Robertson (1966) [60].

As demonstrated, explicit integration methods are not well-suited for stiff equations, due to the required tiny integration steps. The employment of classical implicit integration methods using fixed point iteration does not solve this issue either, and they also require the use of very limiting integration steps to avoid numerical instabilities and converge stiff problems. From the works of Curtiss and Hirschfelder [3] and Gear [4] most of the methods employed to solve stiff equations are based on a family of implicit linear multi-step methods known as Backward Differentiation Formulas

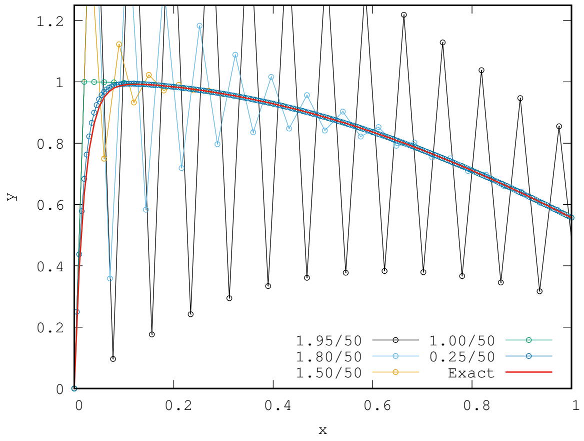


Figure 2.12: Numerical integration of $y' = -50(y' - cos(x))$ using first-order explicit Euler for different integration steps Δx.

(BDFs). BDFs have the following form [61]:

$$\mathbf{y}^{n+k} = \Delta x \beta_k \mathbf{f}(x^{n+k}, \mathbf{y}^{n+k}) + \mathbf{g}^{n+k} \tag{2.100}$$

where vector $\mathbf{g}^{n+k}$ is known since its value is obtained from the previous calculated steps:

$$\mathbf{g}^{n+k} = -\sum_{j=0}^{k-1} \alpha_j \mathbf{y}^{n+j} \tag{2.101}$$

Note that in this class of linear multi-step methods only the function value evaluated in the last time-step t^{n+k} is employed. They are clearly implicit since $\beta_k \neq 0$ and $\beta_j = 0$ for $j = 0, .., k-1$. The value α_k does not appear since is always equal to unity. If a fixed point iteration method is used to obtain the value $\mathbf{y}^{n+k}$, the iterative method will converge to a fixed point x_i if the Lipschitz condition is satisfied:

$$|g(x_1) - g(x_2)| \leq M\,|x_1 - x_2| \quad \forall x_i \in G \tag{2.102}$$

with M such that $0 \leq M \leq 1$, where $g(x_i) = x_i$ and G is a closed subset of $\mathbb{R}$ or $\mathbb{C}$ [62]. This condition is known as the Contraction Mapping Theorem, which can be applied to Eq. (2.100), leading to:

$$|\Delta x \beta_k \mathbf{f}(x_1, \mathbf{y}) - \Delta x \beta_k \mathbf{f}(x_2, \mathbf{y})| \leq |x_1 - x_2| \tag{2.103}$$

that can be rearranged as:

$$|\Delta x \beta_k L| \leq 1 \tag{2.104}$$

where L is the Lipschitz constant of the continuously differentiable function $\mathbf{f}(x, \mathbf{y})$, defined as:

$$L = \max_{(x,\mathbf{y})\in\mathbb{R}} \left| \frac{\partial \mathbf{f}(x, \mathbf{y})}{\partial x} \right| \tag{2.105}$$

As demonstrated above, stiff equations are characterized by having large derivatives values, and hence, this Lipschitz constant L will be a large value. Therefore, in order to fulfil the condition expressed in Eq. (2.104), the integration step Δx must be very small. This analysis shows that implicit integration methods also lead to very restrictive integration steps if fixed point iteration methods are employed [63]. Therefore, non-classical integration methods able to overcome this constraint must be used in order to integrate stiff equations using reasonable integration steps, aiming to obtain accurate solutions in an affordable computational time. Some of these methods are the Rosenbrock method, the Semi-Implicit Extrapolation Method or the Gear's-like methods [47].

2.5.2 Numerical integration of finite-rate combustion equations

Combustion processes are exothermic chemical reactions which take place in a fluid flow. Therefore, combustion simulations require to solve the dynamics of the flow where combustion occurs as well as the kinetics of the chemical process. As consequence, in finite-rate combustion the Navier-Stokes equations, together with the energy conservation equation and a mass fraction conservation equation per each species present in the flow mixture, have to be solved. Since the current work is focused on the efficient parallelization of the integration of the reaction rates, the focus is placed on the species mass fraction conservation equation. Hence, the mass fraction conservation equation for species k in non-conservative form reads as:

$$\rho \frac{\partial Y_k}{\partial t} + \rho \mathbf{u} \cdot \nabla Y_k = -\nabla \cdot (\rho \mathbf{V_k} Y_k) + \dot{w}_k \tag{2.106}$$

where $\mathbf{V_k}$ is the diffusion velocity and $\dot{w}_k$ the mass reaction rate of the species k. This last term is responsible for possible stiffness when integrating Eq. (2.106). The calculation of this term relies on the empirical Arrehnius law (see Section 2.2.1). The presence of exponential and non-linear terms in these calculations are the reason of the stiffness of the mass reaction rate. Since this source term is the one requiring a special temporal integration strategy, for the sake of clarity, lets redefine the previous expression as:

$$\rho\frac{\partial Y_k}{\partial t} = \mathbf{F}(Y_k) + \dot{w}_k \tag{2.107}$$

where $\mathbf{F}(Y_k)$ includes the convective and diffusive operators of mass fraction conservation equation for species k, i.e.:

$$\mathbf{F}(Y_k) = -\rho\mathbf{u}\cdot\nabla Y_k - \nabla\cdot(\rho\mathbf{V_k}Y_k) \tag{2.108}$$

The integration strategy of the transport equations is based on a predictor-corrector scheme for Low-Mach number flows further detailed in the work of Ventosa et al. [28]. At each step of the Predictor-corrector scheme, first the transported scalars are integrated in non-conservative form in order to compute the density at the new time step, which is afterwards employed in the integration of momentum equation.

When the term $\dot{w}_k$ is non-stiff, both terms on the right-hand side of Eq. (2.107) are integrated explicitly. On the other, when this mass reaction rate is stiff, a splitting technique is applied, and then the term $\mathbf{F}(Y_k)$ is integrated explicitly, while the mass reaction rate, $\dot{w}_k$, is integrated implicitly using an integration technique well-suited for stiff equations. Among the several types of operator-splitting techniques reported in the literature [64–68], the one employed in the present work is based on a pseudo-time splitting procedure similar to the one employed by Consul [69] and Vos [66]. The splitting technique used in the present work has been developed for the predictor-corrector scheme detailed in Section 2.2.6 and can be summarized as:

Predictor step

The mass fraction conservation equations of the N species are fully explicitly integrated in order to obtain the predicted values Y_k^*:

$$\rho^n\frac{{Y_k}^* - {Y_k}^n}{\Delta t} = \frac{3}{2}\left(\mathbf{F}(Y_k^n) + \dot{w}_k^n\right) - \frac{1}{2}\left(\mathbf{F}(Y_k^{n-1}) + \dot{w}_k^{n-1}\right) \tag{2.109}$$

where the value $\mathbf{F}(Y_k^n)$ is stored in memory.

Corrector step

In the corrector step, a first value Y_k^{n+1} is calculated fully explicitly for all the nodes of the mesh, according to:

$$\rho^*\frac{Y_k^{n+1} - {Y_k}^n}{\Delta t} = \frac{1}{2}\left(\mathbf{F}(Y_k^*) + \dot{w}_k^*\right) + \frac{1}{2}\left(\mathbf{F}(Y_k^n) + \dot{w}_k^n\right) \tag{2.110}$$

During this integration loop, it is estimated which nodes present a *stiff* equation system and which not. The employed criterion is discussed later. If the node is considered non-stiff, species mass fraction equations are integrated explicitly. Then, the mass fraction value at time t^{n+1} is the one obtained from Eq. (2.110). On the other hand, if the node is detected as stiff, the mass reaction rate $\dot{w}_k$ is treated implicitly using a splitting technique. Then, Eq. (2.110) is rearranged as:

$$\rho^* \frac{Y_k^{n+1} - Y_k{}^n}{\Delta t} = \frac{1}{2}\Bigl(\mathbf{F}(Y_k^*)\Bigr) + \frac{1}{2}\Bigl(\mathbf{F}(Y_k^n)\Bigr) + \dot{w}_k^{n+1} \tag{2.111}$$

which is split in two parts:

$$\rho^* \frac{Y_k^p - Y_k{}^n}{\Delta t} = \frac{1}{2}\Bigl(\mathbf{F}(Y_k^*)\Bigr) + \frac{1}{2}\Bigl(\mathbf{F}(Y_k^n)\Bigr) \tag{2.112}$$

$$\rho^* \frac{Y_k^{n+1} - Y_k{}^p}{\Delta t} = \dot{w}_k^{n+1} \tag{2.113}$$

where Eq. (2.112) is integrated explicitly, since the values at the right-hand side are already known, and Eq. (2.113) is integrated implicitly using an integration method well-suited for stiff equation systems. The value Y_k^{n+1} obtained in Eq. (2.110) is employed as initial *seed* of the implicit integration method.

This pseudo-time splitting technique allows the employment of any integration method for stiff equations. The one employed in the present work is the Gear's method [4], which is the basis of the vast majority of integration methods developed and used for stiff computations [59]. The Gear's method is based on the Backward Differentiation Formulas (BDFs) coupled together the Newton's method [47]. Hence, this integration technique possess the high stability of the BDF methods and overcomes the fixed point iteration issue of implicit schemes making use of the Newton iteration strategy. Applying the Newton's method to Eq. (2.100), results in:

$$\mathbf{y}_{[s+1]}^{n+k} = \mathbf{y}_{[s]}^{n+k} - \frac{\mathbf{y}_{[s]}^{n+k} - \Delta x \beta_k \mathbf{f}\left(\mathbf{y}_{[s]}^{n+k}\right) - \mathbf{g}^{n+k}}{\mathbf{I} - \Delta x \beta_k \frac{\partial}{\partial \mathbf{y}} \mathbf{f}\left(\mathbf{y}_{[s]}^{n+k}\right)} \tag{2.114}$$

where $\mathbf{I}$ is the identity matrix. Notice that in the denominator appears the Jacobian matrix $\mathbf{J} = \frac{\partial}{\partial \mathbf{y}} \mathbf{f}(\mathbf{y})$. New Jacobian matrix and its inverse are needed at each iteration of the Newton's method. Depending on the system size (i.e., the number of species N), the Jacobian evaluation plus the matrix inversion can be computationally very expensive. The Gear's method try to reduce this calculation effort performing the Jacobian evaluation and the matrix inversion only occasionally. This calculations are performed at the first iteration ($s = 0$) and when the convergence rate of the Newton's method is slow or even stalled.

A key aspect of the presented method is the criterion to decide which control volumes are integrated explicitly and which ones are integrated employing the implicit strategy by means of the Gear's method presented above. This criterion can be based on a mathematical analysis of the equation system, as the methods presented by Hairer and Wanner [59] for the automatic stiffness detection. Another possibility is to define a threshold based on a physical criterion. In this work, the latter option is preferred, since its computational cost is almost negligible and the methodology does not depend on the selected stiff integration method. The idea is to define a *chemical time step* Δt_{chem} sufficient to integrate accurately the mass reaction rates using an explicit time integration method, in some sense a CFL-like condition for the chemical reactions. This *chemical time step* acts as a *detector* of the regions where the chemical process is most reactive, and it is defined as:

$$\Delta t_{chem} = f_r \frac{\rho}{c_p} \frac{\left(h - \sum_{k=1}^{N} Y_k \Delta h_{f,k}^0\right)}{\sum_{k=1}^{N} \dot{w}_k \Delta h_{f,k}^0} \tag{2.115}$$

where f_r is a security factor, which in this work is set to $f_r = 0.05$. This criterion is designed trying to limit the increment of energy per unit mass, avoiding rapid increases of enthalpy and temperature, and aiming to follow all the scales of the chemical reaction process. The integration time step Δt of the simulation is limited by the requirements of accuracy and stability for both convective and diffusive terms of the governing equations integrated explicitly. This time step Δt can be obtained from the classical CFL-condition. However, a self-adaptive time strategy based on the estimation of the eigenvalues of the system of equations is employed in this work. This method is further detailed in Chapter 5. Hence, in the corrector step of the algorithm, for each control volume is estimated the *chemical time step* Δt_{chem}. If it is bigger than the simulation time step, $\Delta t_{chem} > \Delta t$, the mass reaction rates of the control volume are integrated implicitly using the Gear's method. Otherwise, if the *chemical time step* is equal or smaller than the simulation time step, $\Delta t_{chem} \leq \Delta t$, all the terms are integrated explicitly.

This integration strategy allows to notably reduce the total amount of computational requirements of the simulation. Only the control volumes with *active* chemical reactions are integrated implicitly using Gear's method. The vast majority of the computational domain is integrated explicitly, avoiding the huge computational effort demanded by the Gear's method, since it implies an iterative process involving evaluations of Jacobians and matrix inversions.

2.5.3 Dynamic balancing algorithm

The presented integration methodology for combustion simulations using finite-rate chemistry presented in the previous section obviously helps to reduce the computational effort of the simulations, although this reduction strongly depend on the implemented parallel technique. In parallel simulations the computational mesh is divided in several parts, and each partition is solved by one of the CPUs involved in

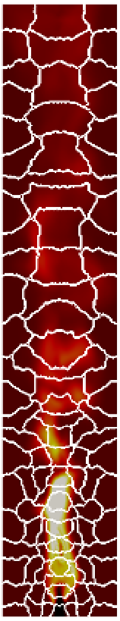


Figure 2.13: Slice showing a flame and the partitions of the computational mesh.

the computation. Usually, the partition of the mesh is carried out assuming that all the elements in a mesh will have the same computational load. The number of control volumes in the computational boundaries should be minimized to reduce the amount of data to be communicated between the CPUs. In the present case, the partition algorithm is based on the widely employed software METIS [70]. A flame and the partitions of the computational mesh employed in a parallel simulation is shown in Fig. 2.13. As can be seen, the flame is mainly located in a few CPUs. This means that in these processors the majority of the control volumes will have very active chemical reactions process presenting stiff equation systems, while the other CPUs far from the flame will be free of stiff sets of equations.

Therefore, this situation will create an undesired imbalance between the different processors involved in the simulation. This load imbalance can be faced statically if the computational load of each node is known a priori, making the mesh partitioning taking into account the relative computational load of each node. However, this strategy is not possible if the computational load distribution on the domain is not previously known or if the computational load changes dynamically during simulation time. In order to face this kind of situations, an algorithm that dynamically balance tasks that are imbalanced when employing the initial domain decomposition has been developed and assessed. This dynamic balancing algorithm has been implemented in the unstructured parallel CFD code TermoFluids [71], and is specially well-suited for situations where the load imbalance changes along the simulation. In these situations, a fixed domain decomposition based in the load of the cells is not possible. Below are detailed the steps followed by the dynamic balancing algorithm at each iteration of the simulation in order to balance the computational load:

1. Firstly, the cells with a heavier computational load are identified.

2. Then, the algorithm determines an efficient distribution of the computational load and defines the optimal communication scheme. In this step, each processor runs a sequential algorithm in order to determine the new tasks distribution as well as the required communication scheme. Once finished this process, each CPU knows the data to be sent and received. This sequential algorithm is replicated in each processor involved in the parallel computation. Its cost is almost negligible.

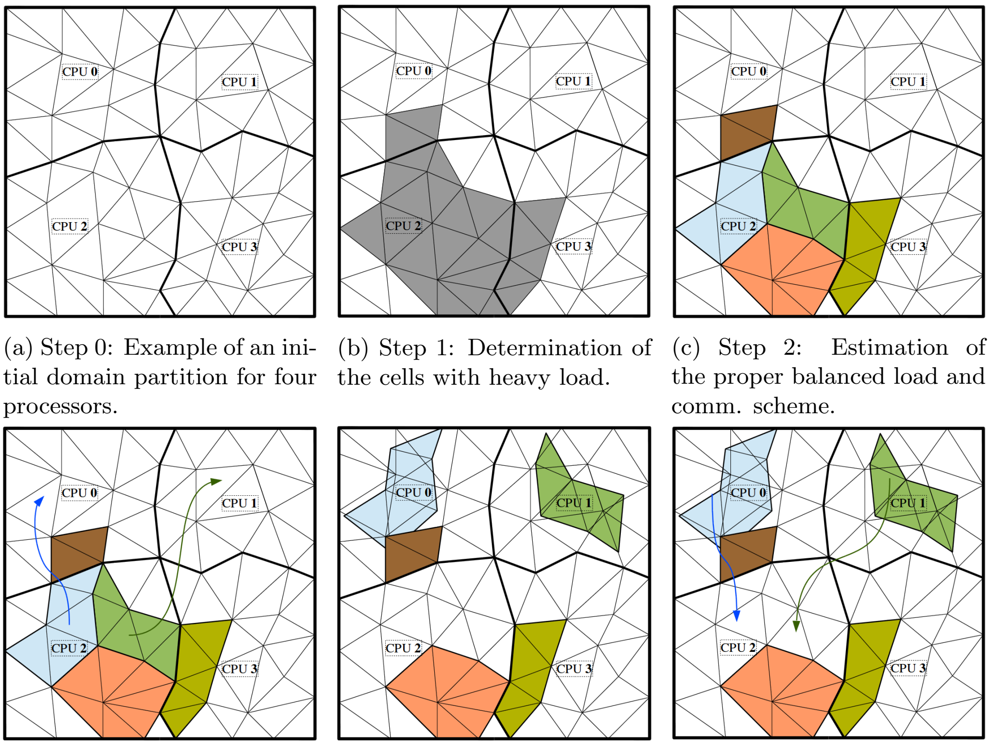
3. The next step is the distribution of the computational load. All the data required to perform the calculations is packed and distributed according to the communication scheme defined in the previous step. This distribution is done by means of a point-to-point communication between the different processors.

4. Once all the data have been communicated, each processor solves its assigned tasks. The external tasks, i.e., the ones initially owned by other processors, are solved in first place, aiming to return the outgoing results as soon as possible to the origin processor.

5. Finally, the processors that have sent part of their tasks to be solved in other CPUs receive the solution back, which is unpacked and stored in the proper memory location.

These steps have been summarized in Fig. 2.14. This dynamic balancing algorithm allows to overcome the load imbalance generated by the Gear's method employed for the integration of stiff sets of equations found in the zones with active chemical reactions. The algorithm allows to achieve load balanced simulations using finite-rate chemistry for cases of auto-ignition or with *dynamic* flames, without the necessity of knowing in advance the position and distribution of the flame/s.

Although in the present work the dynamic balancing algorithm has been implemented for combustion simulations, this load-balancing tool can be easily adapted to other applications presenting imbalanced loads. It is only required to select the proper data to be packed and sent, since the *engine* of the balancing algorithm is independent of the application.



(a) Step 0: Example of an initial domain partition for four processors.

(b) Step 1: Determination of the cells with heavy load.

(c) Step 2: Estimation of the proper balanced load and comm. scheme.

(d) Step 3: Distribution of the computational load between the processors.

(e) Step 4: Resolution of the new assigned tasks as well as the own tasks.

(f) Step 5: Communication between the processors of the resolved *outsourced* tasks.

Figure 2.14: Steps followed by the dynamic balancing algorithm in order to balance the computational load of the simulation.

2.5.4 Performance analysis

Once presented the dynamic balancing algorithm, its performance and scalability have been analysed in detail. Several parametric studies have been carried out, including the main parameters affecting the efficiency of the balancing algorithm. Specifically, the parameters considered in these studies have been:

- The size of the message that is sent per each heavy node.
- The computational cost of the heavy calculation.
- The number of nodes with heavy calculation per processor.
- The number of processors with heavy nodes.

Note that *heavy node* denotes the nodes presenting a higher computational load than the average in the mesh. *Heavy calculation* refers to the set of mathematical operations resulting in this additional load. For each parameter, both weak and strong speed-up tests have been defined. The strong speed-up tests are done fixing the size of the problem, in this case the number of control volumes, and varying the number of CPUs used to do the calculations. On the other hand, weak speed-up tests are done keeping constant the size of the problem assigned to each CPU. In both cases, the speed-up (S) is defined as:

$$S(N) = \frac{t_1}{t_N} \tag{2.116}$$

where t_1 is the amount of time required by one CPU, and t_N the computational time spent by N CPUs.

The designed *base case* for the weak speed-up tests is formed by 250 nodes per processor, where one of the N CPUs involved in the parallel computation has 125 *heavy nodes*. The size of the message communicated between processors for each *heavy node* present in a processor is assigned to be of 100 *doubles* and the *heavy calculation* is composed by a matrix inversion plus a numerical Jacobian evaluation repeated 10 times for a system of 5 equations, mimicking the Newton's method. A schematic representation of this *base case* is depicted in Fig. 2.15. Regarding the strong speed-up tests, the *base case* is similar to the one employed in the weak speed-up tests, but increasing up to 5000 the total number nodes, being *heavy nodes* 250 of them. This is done in order to have a relevant number of nodes per CPU when dividing the computational domain in several parts.

All the studies have been carried out in the UPC JFF cluster (technical specifications: 40 cluster nodes, each node has two AMD Opteron with 16 cores for each CPU linked with 64 gigabytes of RAM memory and an infiniband QDR 4X network interconnection between nodes with latencies of 1.07 microseconds with a 40 Gbits/s

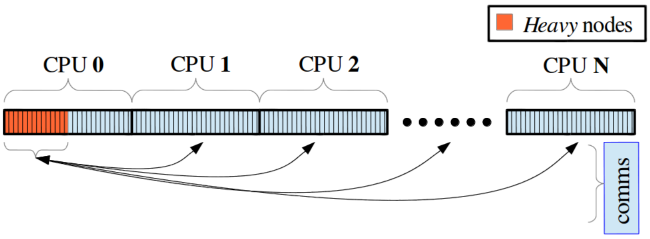
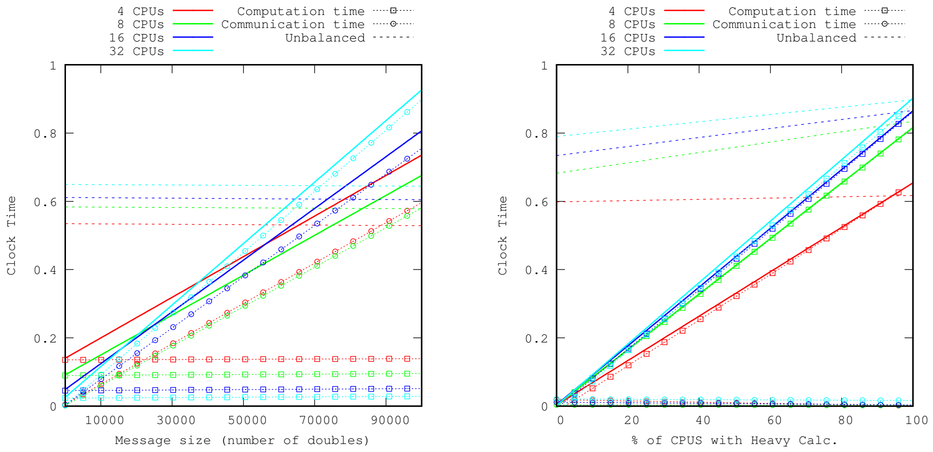


Figure 2.15: Schematic representation of the *base case*.

bandwidth) [56]. In order to avoid spurious and artificial oscillations in the results, all the tests have been repeated several times, and the shown results are linear regressions obtained employing all the values. For each case, the clock time spent in the communications between the processors has been calculated, together with the clock time spent doing the calculations. The results show the total clock time of each case, differentiating the part spent in computation and the one in communications. Moreover, in order to assess the benefits of the dynamic balancing algorithm, the clock time employed for the same cases without using the developed balancing algorithm, i.e. the unbalanced cases, is also shown.

The first studied parameter is the size of the message that is sent per each *heavy* node and how it affects the performance of the implemented balancing algorithm. In order to do so, a parametric study modifying the size of the message sent and received by each *heavy* node, ranging from 50 to 100000 *doubles*, has been performed. The results for the weak speed-up test are shown in Fig. 2.16a. As expected, the higher the data size sent per each *heavy* node, the higher the clock time is, since the time spent in communications increase almost linearly with the size of the message. On the other hand, the unbalanced cases present a uniform clock-time for all the message sizes, since no communications between the processors are taking place. For very large message sizes the case is *degenerated* and the time spent in communications is higher than the time saved with the balanced computations. Nonetheless, these situations are very unlikely to happen, since that means that the required data to be communicated for each *heavy* node is very large compared to the size of the data employed in the *heavy* computations. In the current example, for systems with a size of 5, the number of doubles to produce this degenerated situation should be higher than 60000, meaning that for each variable of the system there are communicated 12000 *double* values. Regarding the strong speed-up test, the clock-time versus the message size is depicted in Fig. 2.20a. As can be seen, the results are very similar to those of the weak speed-up test. In this case strikes that the unbalanced case using 32 CPUs present a smaller clock-time than the other cases, oppositely to the trend found



(a) Varying the message size.

(b) Varying % of CPUs with *heavy* nodes.

Figure 2.16: Weak speed-up results.

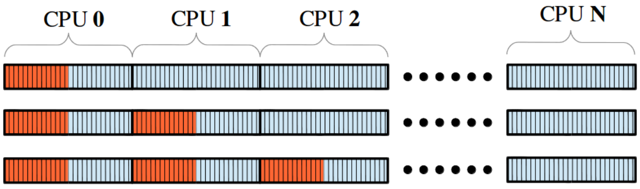
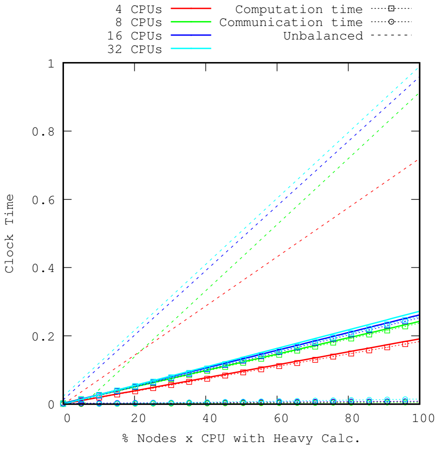


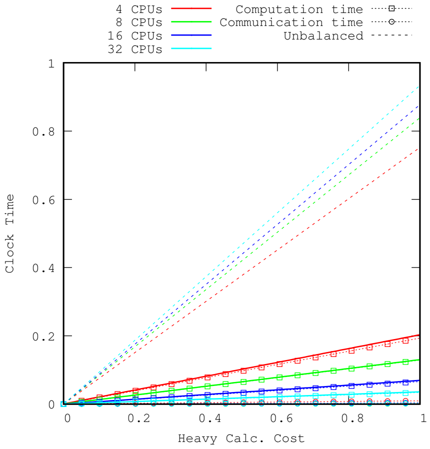
Figure 2.17: Schematic representation of the parametric study varying the number of processors with *heavy* nodes.

in the weak speed-up test. This is because when 32 CPUs are used, the number of nodes per processor are smaller than the total number of *heavy* nodes in the domain, and these must be distributed in two processors. More interesting is the graph shown in Fig. 2.20b, where the results are rearranged to plot the speed-up. The study shows that the speed-up slope decrease when increasing the message size, since the weight of the communication time respect to the total time increases with the size of the message.

The next parameter that has been studied is the number of processors with *heavy* nodes. Starting from the base case where only one processor has *heavy* nodes, it has been increased one by one the number of processors having the half of their nodes with *heavy* calculations, as depicted schematically in Fig. 2.17. As can be seen in Fig. 2.16b, the dynamically balanced computations always outperform the unbalanced



(a) Varying number of *heavy* nodes.



(b) Varying comp. cost of *heavy* calcs.

Figure 2.18: Weak speed-up results.

cases. When the 100 % of the CPUs have *heavy* nodes, both balanced and unbalanced cases collapse at the same clock time, since in these cases the computations are not imbalanced, because all the processors have to do the same computational effort. Notice that the time spent in communications is almost negligible compared to the one spent performing the *heavy* calculations. For that case, only the weak speed-up tests have been carried out.

It has been also studied how the number of *heavy* nodes affects the performance of the dynamic balancing algorithm. The weak speed-up study has been carried out increasing the number of *heavy* nodes in one processor, from a situation where there are no *heavy* nodes up to a case where all the nodes of one processor must perform *heavy* calculations. This parametric study is schematically represented in Fig. 2.19 and the obtained results are shown in Fig. 2.18a. The study shows that the higher the number of *heavy* nodes is, the greater the gain obtained from the dynamic balancing algorithm is. This is the expected behaviour, since increasing the number of *heavy* nodes in one processor means increasing the imbalance of the overall computational load. In order to carry out the strong speed-up test for this parametric study, the strategy changes a little bit: now is modified the number of *heavy* nodes respect the total number of nodes. Specifically, five cases ranging from 10 % up to 90 % of *heavy* nodes in the domain have been analysed, and the obtained results are plot in Fig. 2.22. In this specific case, instead of plotting the linear regression for the unbalanced case, the different values obtained at each test are shown in Fig. 2.22a.

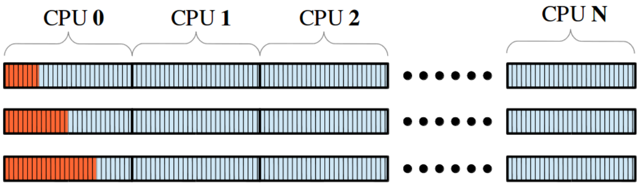


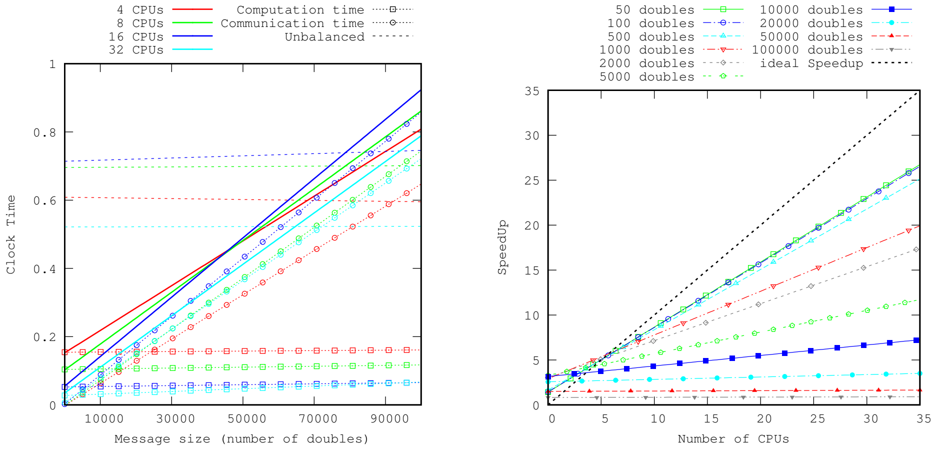
Figure 2.19: Schematic representation of the parametric study varying the number of *heavy* nodes in a CPU.

This is done because in this study the unbalanced cases do not have a linear behaviour and, as can be seen, cases run in parallel using 16 and 32 CPUs are collapsed from the very beginning, meaning that one of the CPUs is already full of *heavy* nodes. On the other hand, unbalanced cases employing 4 and 8 CPUs only collapse when increasing the total number of *heavy* nodes. Similarly to the weak speed-up test, the dynamically balanced cases outperforms the unbalanced cases for all the situations, except when all the nodes have to do *heavy* calculations, since this is, per se, a balanced situation. As can be seen in Fig. 2.22b, the scalability of the implemented algorithm is independent of the number of nodes with *heavy* calculations.

The last parameter analysed is the computational cost of the *heavy* calculation. This parametric study has been carried out modifying the number of times that the matrix inversion and the Jacobian are evaluated, ranging from 1 up to 20 times. In the results, the computational cost is normalized by the case with a bigger cost (in this case, the one of 20 iterations). The results for the weak speed-up test are depicted in Fig. 2.18b and, as can be seen, the results shown that the more expensive the computational task, the greater the benefits obtained using the balancing algorithm. This is the expected behaviour since when the computational cost of the *heavy* task is increased, the overall imbalance of the parallel simulation is also incremented. Regarding the strong speed-up analysis for this parametric study, the results are shown in Fig. 2.21. As can be appreciated in Fig. 2.21a, it is interesting to note that when the computational cost of the *heavy* calculation is incremented, the scalability of the algorithm is enhanced, since the weight of the time spent in the calculations increase, and the weight of the time spent in the communications is reduced.

2.5.5 Real case test: the Cambridge autoignition experiment

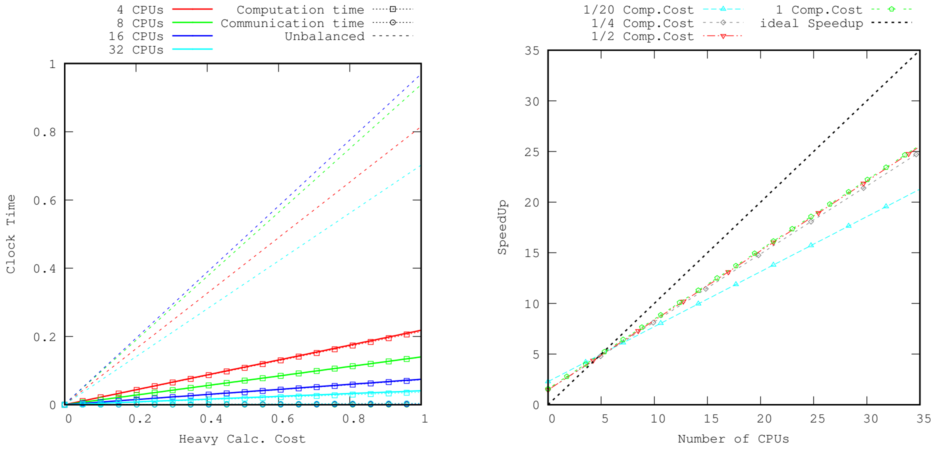
After the parametric study carried out in order to analyse the performance and capabilities of the implemented dynamic balancing algorithm, this is tested in a combustion simulation. The chosen reference case is the well-known Cambridge autoignition experiment previously presented in Section 2.4. Since the experiment set-up has been detailed previously, the description of the experiment is not repeated here. For the



(a) Strong speed-up.

(b) Strong speed-up CPUs.

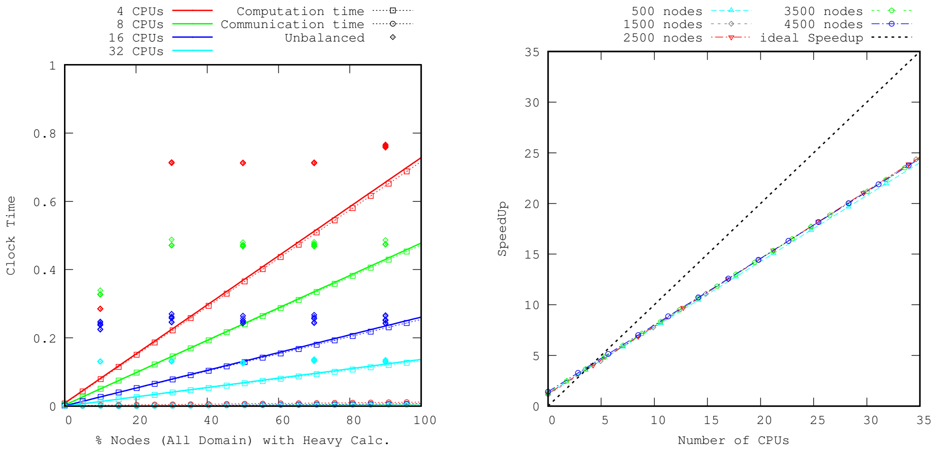
Figure 2.20: Strong speed-up results varying the message size sent per each *heavy* node.



(a) Strong speed-up.

(b) Strong speed-up CPUs.

Figure 2.21: Strong speed-up results varying the computational cost of the *heavy* calculation.



(a) Strong speed-up.

(b) Strong speed-up CPUs.

Figure 2.22: Strong speed-up results varying the total number of *heavy* nodes in the domain.

present study, both weak speed-up and strong speed-up tests have been carried out. The case with a coflowing air temperature of 950 K has been selected (see Table 2.1), which belongs to the *random spots* regime.

Simulation set-up

The simulations have been carried out employing the finite-rate chemistry model under the framework of LES modelling. Hence, the Favre-filtered mass (Eq. (2.37)), momentum (Eq. (2.40)), energy (Eq. (2.41)) and a mass fraction conservation equation (Eq. (2.42)) for each one of the chemical species present in the mixture must be solved.

In the present study the Wall-adapting eddy viscosity model (WALE) sub-grid scale model [21] is used. The WALE sub-grid scale model is based on the square of the velocity gradient tensor. The sub-grid scale viscosity obtained with this model takes into account the strain and the rotation rate of the smallest resolved turbulent fluctuations. Some features of this model are its capability of switching off in two-dimensional flows, in laminar flows, and when the length-scale is in the range of $Re^{-3/4}$. It also has a cubic behaviour near walls with respect to the normal direction of the wall. The unresolved scalar fluxes of both energy and species transport equations are modelled using a gradient assumption [10], where a turbulent Prandtl number Pr_t and a turbulent Schmidt number Sc_t are assumed for the energy and species conservation equations respectively.

For the current case a perfect mixing at sub-grid scale is assumed, meaning that the sub-grid chemistry-flow interaction is not modelled, and therefore $\overline{\dot{w}_k} \approx \dot{w}_k$. This assumption is supported in the relatively low-Reynolds number of the case and in the results obtained in the previous study presented in Section 2.4. The chemical reactions are modelled employing the detailed reaction mechanism for hydrogen of 9 species and 21 reactions developed by Mueller et al. [7].

The pressure-velocity coupling is solved by means of the Fractional step method detailed at Section 2.2.5, and the Poisson equation is solved employing the FFT-based Poisson Solver developed by Borrell et al. [55].

All the meshes employed in the simulations are unstructured and have been refined in the air-fuel shear layer. They have been constructed from a 2D plane revolved in the azimuthal direction. The meshes employed for the weak speed-up studies are:

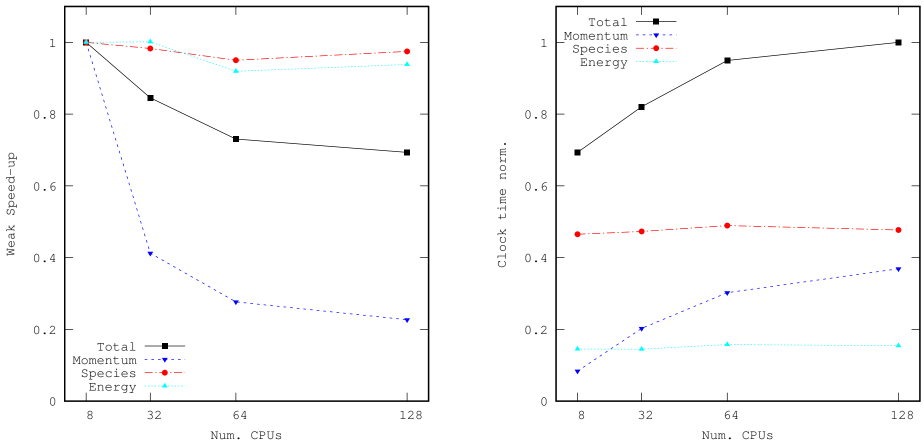
- 8 CPUs: $\sim$ 50000 cells. (6250 cells/plane x 8 planes).
- 32 CPUs: $\sim$ 200000 cells. (12500 cells/plane x 16 planes).
- 64 CPUs: $\sim$ 400000 cells. (25000 cells/plane x 16 planes).
- 128 CPUs: $\sim$ 800000 cells. (25000 cells/plane x 32 planes).

Regarding the strong speed-up test, the study has been done in 8, 16, 32, 64 and 128 CPUs using an unstructured mesh with 445888 cells (27868 cells/plane x 16 planes), and with a mean control volume size of $\bar{x}_i \simeq 0.15$ mm in the air-fuel shear layer region.

One relevant aspect for the dynamic balancing algorithm is the size of the message that is sent between the processors when distributing the tasks, as well as when recollecting the *outsourced* solutions. Each node that delegates the implicit integration to an external processor sends a message of $4 + 2N$ *doubles*, where N is the number of species. This message contains the size of the buffer sent per each node, an ID identifying the node sending the info, the density ρ, the temperature T, and values Y_k^{n+1} plus $Y_k{}^p$, where the former is employed as first guess in the Newton's method. Regarding the recollection step, the size of the message is of $3 + N$ *doubles*, including the ID of the node that request the solution, the integrated values Y_k^{n+1}, and two auxiliary values indicating the number of iterations and Jacobian evaluations performed by Gear's method to converge the solution. Although in this case some of the values can be treated as *integers* instead of *doubles*, helping to reduce the message size, the *heart* of the balancing algorithm has been developed seeking generality and in this first version all the communicated values are considered as *doubles.* However, it is an aspect that can be improved in upcoming versions of the algorithm.

Results and discussion

This section shows the results obtained for both weak and strong speed-up tests for the dynamic balancing algorithm in the selected benchmark combustion case,



(a) Weak speed-up.

(b) Normalized clock time.

Figure 2.23: Results for the transport equations in the weak speed-up test.

concretely the Cambridge autoignition experiment. The weak speed-up values for the momentum, species and energy solvers, as well as the global weak speed-up of the simulation are depicted in Fig. 2.23a. As can be seen, the scalability of both energy and species solvers is almost ideal. On the other hand, the global scalability is affected by the momentum solver, which presents a poor weak speed-up factor. This poor scalability of the momentum solver is due to the pressure-velocity coupling and the Poisson equation. For the current weak speed-up test the number of control volumes per CPU is of only 6250, which is a very small number. Consequently, the clock time spent in communications between the processors becomes more important than the time spent in the solution of the Poisson equation. In order to have a good scalability for momentum more control volumes per CPU are required, as shown by Borrell et al. [55], where a case with 50000 nodes per CPU presents a good weak speed-up factor. Nevertheless, the focus of the current work is placed on the species solver, which exhibits a very good weak speed-up scalability. Aiming to see the relevance of each solver regarding the total time, the normalized clock time spent solving each transport equation, as well as the total time of the simulation, are depicted in Fig. 2.23b. As can be seen, the increase in the simulation time is mainly due to the increment of the time spent solving momentum.

The species equation is analysed in more detail, and more specifically, the attention is placed on the computational step involving the implicit integration of the mass reaction rates. The normalized clock time spent in the implicit integration is shown

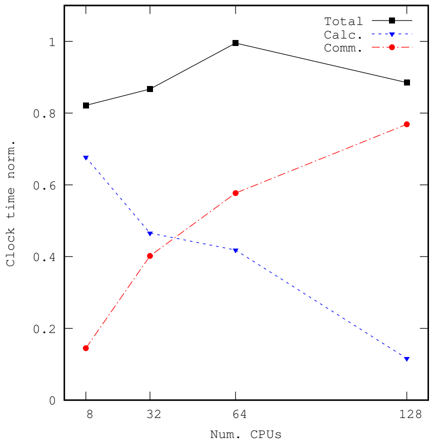
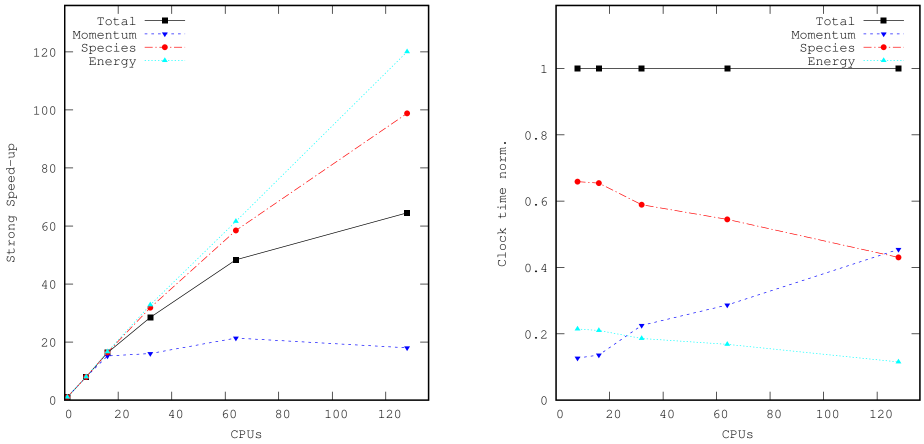


Figure 2.24: Normalized clock time in the weak speed-up test of the implicit integration stage.

in Fig. 2.24. The total time has been split in two parts: one involving the calculations (i.e., the time spent doing the implicit integration by means of the Gear's method), and a second part accounting for the communication of the data and its buffering (although this last part is very small compared to the time spent in communications). The results show that, although the total time dedicated to the implicit integration is quite similar in all the cases, the time spent in calculations and communications changes depending on the number of processors. For the case with 8 CPUs almost all the computational time is dedicated to the calculations, and the communication time represents a small portion of the total time. On the other hand, the case with 128 CPUs shows a completely opposite behaviour, where most of the computational time is spent to communications and only a small part is dedicated to calculations. This is because when the number of processors increases, there are more and more processors available to perform the calculation tasks, and thanks to the dynamic balancing algorithm the computational load can be distributed equitably between all the processors, and the required tasks can be done faster. However, since more processors are involved, the cost in the communications is also increased.

The results for the strong speed-up test are depicted in Fig. 2.25a. As can be seen, the obtained results show a similar trend to that of the weak speed-up test, where the scalability of both species and energy solvers look good. However, the overall scalability of the simulation is severely affected by the momentum equation. Nonetheless, notice that the strong speed-up scalability of the momentum solver for the cases up to 16 processors is good. In this test, the mesh employed for 16 CPUs



(a) Strong speed-up.

(b) Normalized clock time.

Figure 2.25: Results for the transport equations in the strong speed-up test.

has 27868 nodes per CPU. This behaviour supports the argument given previously for the poor scalability found in the momentum solver, which is basically due to the small amount of nodes per CPUs used in the present cases. Figure 2.25b shows the clock time of each simulation normalized by the total time of each one, aiming to see the computational effort spent by each solver. As can be appreciated, when increasing the number of CPUs, the relative effort dedicated to the momentum solver increases, while the relative computational effort to solve species and energy equations is reduced.

As done for the weak speed-up test, Fig. 2.26 gives the normalized time spent in the calculations and in the communications for the implicit integration of the mean reaction rates. The strong speed-up test also reproduces the results found in the weak speed-up test, i.e., when more CPUs are employed in the simulations, the computational load can be distributed into more processors, reducing the time spent in calculations, but increasing the time spent in communications.

Finally, Fig. 2.27 compares the evolution of the normalized clock time versus the number of iterations for an imbalanced simulations and a balanced simulation that uses the dynamic balancing algorithm presented in this work. The case chosen to perform the computation is the one corresponding to the mesh employed for the strong speed-up test and using 64 CPUs. It can be seen that the balanced simulation clearly outperforms the imbalanced one, demonstrating the great advantage that comes when the dynamic balancing algorithm is used. But this figure also gives additional inter-

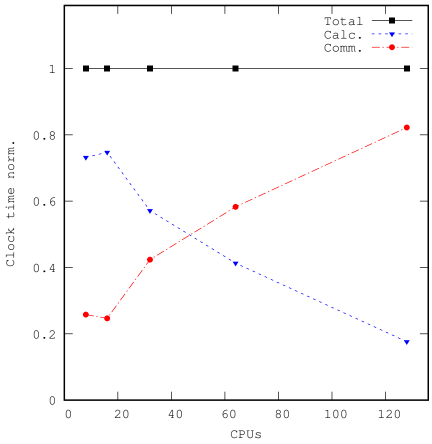


Figure 2.26: Normalized clock time in the strong speed-up test of the implicit integration stage.

esting information. As explained previously in Section 2.4, the solved case belongs to the *Random spots* regime, where auto-ignition kernels appear but are quenched before they could act as a flame anchoring point or a flashback, and are convected out of the domain. Up to iteration $\sim$ 35000 both balanced and imbalanced simulations need the same computational time. This is because before this time, there is no combustion, and only when auto-ignition occurs, and the first ignition kernel appears, the simulations begins to have imbalanced CPUs due to the presence of *active* chemical reactions. Notice that the slope of the curve belonging to the balanced simulations is almost equal before and after the combustion begins, meaning that the dynamic balancing algorithm is able to distribute this additional computational load very well, and thanks to the algorithm, the additional computational load has a very small impact on the computational time. On the other hand, when the auto-ignition kernels start to appear, imbalanced simulations become clearly slower. The curves belonging to the imbalanced case present some changes in their slope, that are due to the behaviour of the *Random spots* regime, where ignition kernels appear but are quenched after a short living time. More pronounced slopes are related to the presence of ignition kernels, and reductions in the slopes occur after the kernels are quenched and before the ignition of a new combustion kernel.

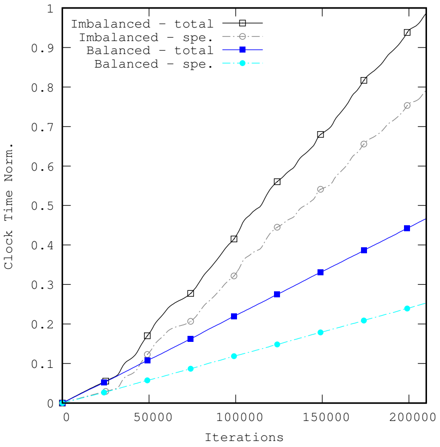


Figure 2.27: Evolution of the computational time versus the number of iterations comparing an imbalanced and a balanced simulation.

2.6 Conclusions

In this chapter some aspects and issues in the field of numerical simulations of combustion processes in CFD have been addressed. Combustion is a physical process involving a large amount of species in a mixture. Their interaction is strongly affected by the flow characteristics as well as the thermo-physical state of the mixture. In other words, the chemical reactions of combustion processes depend on the mixing, the temperature, the velocity, the turbulence characteristics, etc. Therefore, combustion is a very complex phenomena characterized by the presence of a wide range of temporal and spatial scales. This complexity causes that numerical simulations of combustion processes are computationally very expensive. As a matter of fact, turbulent combustion simulations solving all the scales for detailed chemical reaction schemes are practically unaffordable with the computational resources available nowadays. Hence, two possible strategies to make viable the numerical simulation of complex combustion processes have been presented and assessed in this thesis. The first one reduces the dimensionality of the problem using a combustion model. The second one optimizes the computational performance of the simulations improving the load-balance in parallel combustion simulations.

First, a Progress-variable (PV) approach closed by means of the Presumed Conditional Moment (PCM) for LES combustion has been implemented. This combustion model has been successfully applied in order to reproduce numerically the hydrogen auto-ignition experiment carried out by Markides and Mastorakos. The results obtained show that this model is able to reproduce correctly the physical behaviour observed in the experiment, although it over-predicts the reaction rate of the progress variable, and as a result, this causes an under-prediction in the auto-ignition length. The major advantage of this simplified model is the large reduction in the computational costs compared to the Finite-Rate (FR) model (either without explicit closure for the reaction rate, or even higher if a closure model like CMC is used). Therefore, further work is necessary in order to obtain a model with a similar low computational cost but able to give more accurate results. In order to improve the obtained results with the PV-PCM model, two possibilities can be explored: the first one is to extend the present work using an Unsteady Flamelet approach, to take into account the effect of the scalar dissipation rate (this effect is not considered in the present model). Another option is to develop a more accurate approximation to model the distribution of the progress variable, because the obtained results show that modelling the distribution of the progress variable with a $\delta - pdf$ is not a good enough approximation.

On the other hand, the results obtained for the Finite-Rate model without closure show that although the sub-grid scale chemistry-flow interactions are neglected, the results are accurate enough compared with those obtained in the experiment. Nonetheless, further studies should be conducted to obtain a better knowledge about the influence of sub-grid chemistry-flow interaction in hydrogen autoignition events.

Another important aspect that must be taken into account is the mechanism. As was shown by Stankovic et al. [72], or Lee and Mastorakos [73], the autoignition behaviour is strongly influenced by the detailed mechanism employed. For example, Lee et al. [73] shows that for the same case, the autoignition length could vary up to 12 times the nozzle diameter depending on the mechanism. Therefore, further investigation in this field is also necessary.

Additionally, the capability of the presented methodologies to reproduce ignition events using unstructured meshing has been demonstrated and assessed.

Then, it has been presented and assessed a dynamic balancing algorithm well suited for parallel numerical simulations where the computational load of the processors is imbalanced. Usually, the partitioning of the meshes for parallel simulations is done assuming a uniform distribution of the load assigned to each computational node. Nevertheless, in some cases this assumption is not valid. If the computational load distribution is known a priori, the partition of the mesh can be done using this information. However, many times this information is not known and other times the computational load of the nodes changes throughout the simulation, making impossible a proper partitioning of the mesh. The implemented dynamic balancing algorithm aims to help in these cases.

In the present work, the dynamic balancing algorithm has been adapted and employed for combustion simulations. Specifically, it has been used to properly distribute the additional computational load that appears due to the implicit integration of the mass reaction rates of the species transport equations. This mass reaction rate is obtained from a stiff set of equations that requires a special implicit integration method, which creates an imbalance in the simulation.

A deep analysis of the performance of the algorithm has been presented, demonstrating its capacity to properly distribute the computational load of imbalanced simulations for different situations. Moreover, the dynamic balancing algorithm has been tested in a reference combustion case, the well-known Cambridge autoignition experiment. The presented study shows that balanced simulations clearly outperform the imbalanced ones, and that the balancing algorithm presents a good parallel scalability.

This dynamic balancing algorithm has been designed seeking generality and not case-specificity. Therefore, it can be employed for other parallel simulations presenting imbalanced computational loads. Some future works can involve the adaptation of the algorithm to other physics. Another point to treat in future revisions is try to enhance and optimize some parts of the dynamic balancing algorithm.

References

[1] Denis Veynante and Luc Vervisch. Turbulent combustion modeling. *Progress in Energy and Combustion Science*, 28(3):193 – 266, 2002.

[2] Federico Perini, Emanuele Galligani, and Rolf D Reitz. An analytical jacobian approach to sparse reaction kinetics for computationally efficient combustion modeling with large reaction mechanisms. *Energy & Fuels*, 26(8):4804–4822, 2012.

[3] C.F. Curtiss and Joseph O. Hirschfelder. Integration of stiff equations. *Proceedings of the National Academy of Sciences*, 38(3):235–243, 1952.

[4] C. William Gear. *Numerical Initial Value Problems in Ordinary Differential Equations*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1971.

[5] Gregory P. Smith, David M. Golden, Michael Frenklach, Nigel W. Moriarty, Boris Eiteneer, Mikhail Goldenberg, C. Thomas Bowman, Ronald K. Hanson, Soonho Song, William C. Gardiner, Vitali V. Lissianski, and Zhiwei Qin. GRI-Mech 3.0. http://www.me.berkeley.edu/gri-mech/. Accessed: 13/09/2017.

[6] H.C. de Lange and L.P.H. de Goey. Two-dimensional methane/air flame. *Combustion science and technology*, 92(4-6):423–427, 1993.

[7] M.A. Mueller, T.J. Kim, R.A. Yetter, and F.L. Dryer. Flow reactor studies and kinetic modeling of the H2/O2 reaction. *International Journal of Chemical Kinetics*, 31(2):113–125, 1999.

[8] Bamdad Lessani and Miltiadis V. Papalexandris. Time-accurate calculation of variable density flows with strong temperature gradients and combustion. *Journal of Computational Physics*, 212(1):218–246, 2006.

[9] R.B. Bird, W.E. Stewart, and E.N. Lightfoot. *Transport Phenomena.* Wiley International edition. Wiley, 2007.

[10] Thierry Poinsot and Denis Veynante. *Theoretical and numerical combustion.* RT Edwards, Inc., 2005.

[11] Vincent Giovangigli. Mass conservation and singular multicomponent diffusion algorithms. *IMPACT of Computing in Science and Engineering*, 2(1):73–97, 1990.

[12] K.K. Kuo. *Principles of combustion.* John Wiley, 2005.

[13] J.O. Hirschfelder and C.F. Curtiss. The theory of flame propagation. *The Journal of Chemical Physics*, 17(11):1076–1081, 1949.

[14] Jordi Ventosa Molina. *Numerical simulation of turbulent diffusion flames using flamelet models on unstructured meshes.* PhD thesis, Universitat Politècnica de Catalunya, 2015.

[15] S.B. Pope. *Turbulent Flows.* Cambridge University Press, 2000.

[16] A. Favre. Turbulence - Space-time statistical properties and behavior in supersonic flows. *Physics of Fluids*, 26:2851–2863, October 1983.

[17] P. Sagaut. *Large Eddy Simulation for Incompressible Flows: An Introduction.* Scientific Computation. Springer, 2006.

[18] Joseph Smagorinsky. General circulation experiments with the primitive equations: I. The basic experiment. *Monthly weather review*, 91(3):99–164, 1963.

[19] Massimo Germano, Ugo Piomelli, Parviz Moin, and William H Cabot. A dynamic subgrid-scale eddy viscosity model. *Physics of Fluids A: Fluid Dynamics*, 3(7):1760–1765, 1991.

[20] Roel Verstappen. When does eddy viscosity damp subfilter scales sufficiently? In *Quality and Reliability of Large-Eddy Simulations II*, pages 421–430. Springer, 2011.

[21] Franck Nicoud and Frédéric Ducros. Subgrid-scale stress modelling based on the square of the velocity gradient tensor. *Flow, turbulence and Combustion*, 62(3):183–200, 1999.

[22] Bernard J Geurts and Darryl D Holm. Regularization modeling for large-eddy simulation. *Physics of fluids*, 15(1):L13–L16, 2003.

[23] Oriol Lehmkuhl. *Numerical resolution of turbulent flows on complex geometries*. PhD thesis, Universitat Politècnica de Catalunya, 2012.

[24] FTM Nieuwstadt and JP Meeder. Large-eddy simulation of air pollution dispersion: a review. In *New tools in turbulence modelling*, pages 265–280. Springer, 1997.

[25] 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.

[26] Franck Nicoud. Conservative high-order finite-difference schemes for low-mach number flows. *Journal of Computational Physics*, 158(1):71–97, 2000.

[27] R. Knikker. A comparative study of high-order variable-property segregated algorithms for unsteady low Mach number flows. *International Journal for Numerical Methods in Fluids*, 66:403–427, 2011.

[28] J. Ventosa-Molina, J. Chiva, O. Lehmkuhl, J. Muela, C. D. Pérez-Segarra, and A. Oliva. Numerical analysis of conservative unstructured discretisations for low mach flows. *International Journal for Numerical Methods in Fluids*, 84(6):309–334, 2017. fld.4350.

[29] Habib N. Najm, Peter S. Wyckoff, and Omar M. Knio. A Semi-Implicit Numerical Scheme for Reacting Flow. *Journal of Computational Physics*, 143:381–402, 1998.

[30] Ulrich Maas and Stephen B. Pope. Implementation of simplified chemical kinetics based on intrinsic low-dimensional manifolds. In *Symposium (International) on Combustion*, volume 24, pages 103–112. Elsevier, 1992.

[31] Ulrich Maas and Stephen B. Pope. Simplifying chemical kinetics: intrinsic low-dimensional manifolds in composition space. *Combustion and flame*, 88(3):239–264, 1992.

[32] Olivier Gicquel, Nasser Darabiha, and Dominique Thévenin. Liminar premixed hydrogen/air counterflow flame simulations using flame prolongation of ILDM with differential diffusion. *Proceedings of the Combustion Institute*, 28(2):1901–1908, 2000.

[33] J.A. van Oijen and L.P.H. de Goey. Modelling of premixed laminar flames using flamelet-generated manifolds. *Combustion Science and Technology*, 161(1):113–137, 2000.

[34] J.A. van Oijen, F.A. Lammers, and L.P.H. de Goey. Modeling of complex premixed burner systems by using flamelet-generated manifolds. *Combustion and Flame*, 127(3):2124–2134, 2001.

[35] S. B. Pope. Computationally efficient implementation of combustion chemistry using in situ adaptive tabulation. *Combustion Theory and Modelling*, 1:41–63, 1997.

[36] B. Yang and S.B. Pope. Treating chemistry in combustion with detailed mechanisms—in situ adaptive tabulation in principal directions—premixed combustion. *Combustion and Flame*, 112(1):85–112, 1998.

[37] C.D Pierce and P. Moin. Progress-variable approach for large-eddy simulation of non-premixed turbulent combustion. *Journal of Fluid Mechanics*, 504:73–97, 2004.

[38] Matthias Ihme, Lee Shunn, and Jian Zhang. Regularization of reaction progress variable for application to flamelet-based combustion models. *Journal of Computational Physics*, 231(23):7715–7721, 2012.

[39] Jean-Baptiste Michel, Olivier Colin, and Denis Veynante. Modeling ignition and chemical structure of partially premixed turbulent flames using tabulated chemistry. *Combustion and Flame*, 152(1):80 – 99, 2008.

[40] J.B. Michel, O. Colin, and D. Veynante. Comparison of Differing Formulations of the PCM model by their Application to the Simulation of an Auto-igniting H2/air jet. *Flow, Turbulence and combustion*, 83:33–60, 2009.

[41] A.Y. Klimenko. Multicomponent diffusion of various admixtures in turbulent flow. *Fluid dynamics*, 25(3):327–334, 1990.

[42] R.W. Bilger. Conditional moment closure for turbulent reacting flow. *Physics of Fluids A: Fluid Dynamics*, 5(2):436–444, 1993.

[43] Alex Y Klimenko and Robert William Bilger. Conditional moment closure for turbulent combustion. *Progress in energy and combustion science*, 25(6):595–687, 1999.

[44] Matthias Ihme and Heinz Pitsch. Prediction of extinction and reignition in nonpremixed turbulent flames using a flamelet/progress variable model: 1. A priori study and presumed PDF closure. *Combustion and Flame*, 155(1):70 – 89, 2008.

[45] Norbert Peters. *Turbulent Combustion*. Cambridge Monographs on Mechanics. Cambridge University Press, 2000.

[46] Fengshan Liu, Hongsheng Guo, Gregory J Smallwood, ÖL Gülder, and MD Matovic. A robust and accurate algorithm of the β-pdf integration and its application to turbulent methane–air diffusion combustion in a gas turbine combustor simulator. *International Journal of Thermal Sciences*, 41(8):763–772, 2002.

[47] William H. Press. *Numerical recipes 3rd edition: The art of scientific computing*. Cambridge University Press, 2007.

[48] V. Raman, H. Pitsch, and R.O. Fox. Eulerian transported probability density function sub-filter model for large eddy simulations of turbulent combustion. *Combustion Theory And Modelling*, 10:439–458, 2006.

[49] L. Vervisch, R. Hauguel, P. Domingo, and M. Rullaud. Three facets of turbulent combustion modelling: DNS of premixed V-flame, LES of lifted nonpremixed flame and RANS of jet-flame. *Journal of Turbulence*, 5(4):1–36, 2004.

[50] R. Kulkarni and W. Polifke. Large Eddy Simulation of Autoignition in a Turbulent Hydrogen Jet Flame Using a Progress Variable Approach. *Journal of Combustion*, 2012(1):1–11, 2012.

[51] G. Balarac, H. Pitsch, and V. Raman. Development of a dynamic model for the subfilter scalar variance using the concept of optimal estimators. *Physics of Fluids*, 20:035114, 2008.

[52] R.W.C.P. Verstappen and A.E.P. Veldman. Symmetry-preserving discretization of turbulent flow. *Journal of Computational Physics*, 187:343–368, 2003.

[53] C.N. Markides and E. Mastorakos. An experimental study of hydrogen autoignition in a turbulent co-flow of heated air. *Proceedings of the Combustion Institute*, 30(1):883–891, 2005.

[54] W.P. Jones and S. Navarro-Martinez. Study of hydrogen auto-ignition in a turbulent air co-flow using a Large Eddy Simulation approach. *Computers and Fluids*, 37(7):802–808, 2008.

[55] R. Borrell, O. Lehmkuhl, F.X. Trias, and A. Oliva. Parallel direct poisson solver for discretisations with one fourier diagonalisable direction. *Journal of Computational Physics*, 12:4723–4741, 2011.

[56] Heat and Mass Transfer Technological Center (CTTC): Computer facilities. http://www.cttc.upc.edu/node/29. Accessed: 20/10/2017.

[57] Rajat Mittal and Gianluca Iaccarino. Immersed boundary methods. *Annu. Rev. Fluid Mech.*, 37:239–261, 2005.

[58] M. Ihme, C.M. Cha, and H. Pitsch. Prediction of local extinction and re-ignition effects in non-premixed turbulent combustion using a flamelet/progress variable approach. *Proceedings of the Combustion Institute*, 30:793–800, 2005.

[59] E. Hairer, S.P. Nørsett, and G. Wanner. *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems.* Lecture Notes in Economic and Mathematical Systems. Springer, 1993.

[60] R.A. Willoughby. International Symposium on Stiff Differential Systems. In *Stiff Differential Systems*, pages 1–19. Springer, 1974.

[61] U.M. Ascher and L.R. Petzold. *Computer methods for ordinary differential equations and differential-algebraic equations*, volume 61. Siam, 1998.

[62] A.S. Ackleh, E.J. Allen, R.B. Kearfott, and P. Seshaiyer. *Classical and Modern Numerical Analysis: Theory, Methods and Practice.* Chapman & Hall/CRC Numerical Analysis and Scientific Computing Series. CRC Press, 2009.

[63] Endre Süli. *Numerical Solution of Ordinary Differential Equations.* Mathematical Institute, University of Oxford, 2014.

[64] J.A. Miller and R.J. Kee. Chemical nonequilibrium effects in hydrogen-air laminar jet diffusion flames. *The Journal of Physical Chemistry*, 81(25):2534–2542, 1977.

[65] R.J. Kee and J.A. Miller. A split-operator, finite-difference solution for axisymmetric laminar-jet biffusion flames. *AIAA Journal*, 16(2):169–176, 1978.

[66] J.B. Vos. Calculating turbulent reacting flows using finite chemical kinetics. *AIAA journal*, 25(10):1365–1372, 1987.

[67] P.J. Coelho and J.C.F. Pereira. Calculation of a confined axisymmetric laminar diffusion flame using a local grid refinement technique. *Combustion science and technology*, 92(4-6):243–264, 1993.

[68] O. Holm-Christensen, I.P. Jones, N.S. Wilkes, B.A. Splawski, and P.J. Stopford. The solution of coupled flow and chemistry problems. *Progress in Computational Fluid Dynamics, an International Journal*, 1(1-3):43–49, 2001.

[69] Ricard Cònsul Serracanta. *Development of numerical codes for the evaluation of combustion processes. Detailed numerical simulations of laminar flames.* PhD thesis, Universitat Politècnica de Catalunya, 2002.

[70] George Karypis and Vipin Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on scientific Computing*, 20(1):359–392, 1998.

[71] O. Lehmkuhl, C.D. Pérez Segarra, R. Borrell, M. Soria, and A. Oliva. Termofluids: A new Parallel unstructured CFD code for the simulation of turbulent industrial problems on low cost PC cluster. *Proceedings of the Parallel CFD Conference*, pages 1–8, 2007.

[72] I. Stankovic and B. Merci. Analysis of auto-ignition of heated hydrogen-air mixtures with different detailed reaction mechanisms. *Combustion Theory And Modelling*, 15(3):409–436, 2011.

[73] C.W. Lee and E. Mastorakos. Transported scalar PDF calculations of autoignition of a hydrogen jet in a heated turbulent co-flow. *Combustion Theory And Modelling*, 12(6):1153–1178, 2008.

3

Dispersed multi-phase flows

Some contents of this chapter have been published at international journals:

P. Koullapis, S.C. Kassinos, J. Muela, C. Perez-Segarra, J. Rigola, O. Lehmkuhl, Y. Cui, M. Sommerfeld, J. Elcner, M. Jicha, I. Saveljic, N. Filipovic, F. Lizal, and L. Nicolaou. *Regional aerosol deposition in the human airways: The siminhale benchmark case and a critical assessment of in silico methods.* European Journal of Pharmaceutical Sciences, pages -, 2017.

Another parts of this chapter have been presented at international conferences:

J. Muela, O. Lehmkuhl, C.D Pérez-Segarra and A. Oliva. *A new statistical model for subgrid dispersion in large eddy simulations of particle-laden flows.* Journal of Physics: Conference Series. Vol. 745. No. 3. IOP Publishing, 2016.

J. Muela, O. Lehmkuhl and A. Oliva. *Study of stochastic models for subgrid dispersion in lagrangian-eulerian formulation.* In Proceedings of *10th International ERCOFTAC Workshop on Direct and Large-Eddy Simulation 2015*, Limassol, Cyprus, 27-29 May 2015.

Abstract. Emission and transport of pollutants in air, injection of fuel in combustion chambers or drug administration via aerosol are flows characterized by the presence of two or more phases, where one phase is continuous and the other ones are dispersed, in form of unconnected small particles or droplets. This type of flows are known as dispersed multiphase flows. These flows present two different characteristic length-scales: the domain where the continuous phase moves or it is contained, and the size of the particles of the dispersed phase. These two scales tend to be several orders of magnitude different. Moreover, there is an interaction between all the present phases in the flow. These aspects generate additional complexities that are not present in simulations of single-phase flows, and hence, specific techniques and additional models are required in order to simulate dispersed multi-phase flows. In the present chapter these techniques are introduced and the models required to take into account all the physical phenomena present in this kind of flows are explained. Special emphasis is placed in numerical aspects like interpolation techniques for unstructured meshes and efficient time-integration strategies. Two well-known stochastic models

for particle sub-grid dispersion modelling in the framework of LES are introduced, implemented and carefully analysed. Due to the shortcomings and deficiencies presented by these stochastic models, a new sub-grid dispersion model is proposed, developed and studied. This new model is based on the reconstruction of statistical data using probability density functions. Closing the chapter some benchmark cases are introduced and simulated in order to demonstrate the capabilities of the implemented algorithms to simulate dispersed multiphase flows. All the numerical libraries have been implemented and tested in the in-house CFD code TermoFluids.

3.1 Introduction

Dispersed multi-phase flows are a kind of flows characterized by the presence of one (or more) dispersed phases in the form of particles, droplets or bubbles within a continuous carrier phase. This kind of flows is present in many domestic, commercial and industrial applications, as for example, internal combustion engines, inhaled drug delivery, dispersion of contaminants, steam turbines, evaporative cooling, fluidized bed reactors, fire suppression systems, etc. Hence, there is a clear interest from both scientific and industrials fields to have a better knowledge of the fundamental physics involved in the complex physic phenomena present in this kind of flows, as well as to have the technology and tools able to design and optimize facilities involving dispersed multi-phase flows. One possibility to achieve these goals is via experiments. Nonetheless, experiments are usually laborious, expensive, high time consuming and can be really difficult observe and analyse all the information of interest. Moreover, sometimes the probes and sensors can alter and modify the real physics of observed phenomena. Another possibility is to perform numerical simulations, which are usually cheaper, flexible and allow to obtain detailed information of all the studied domain without altering the physics. Therefore, there is a high motivation to develop efficient numerical tools able to perform reliable numerical simulations of dispersed multi-phase flows.

Aiming towards the development of a numerical software to simulate dispersed multi-phase flows, there are available several computational methods. Each method has its owns advantages and disadvantages, and each one of them is most well suited for a certain application depending on flow characteristics and/or the scope and requirements of the simulation. Following, a brief introduction to the numerical methods currently available for the simulation of dispersed multi-phase flows is presented, where the advantages and disadvantages of each numerical approximation are detailed.

- **Fully resolved methods.** There are various methods falling inside this category, like Front-tracking methods, Level Set methods (LS) or Volume of Fluids (VoF) methods. The common point of all these methods is that they resolve all the relevant flow scales around the particles, allowing to compute the interaction between both phases at the interface. Therefore, the acceleration and/or deformation of the particles, droplets or bubbles can be directly calculated. Obviously, these methods require that the computational grid should be several times smaller than the resolved particles, which makes that the computational resources required to solve a large amount of particles or droplets is huge. The last advancements in computational power, as well as in numerical methods, have allowed 3D simulations of multi-phase dispersed flows employing fully resolved methods up to $\sim \mathcal{O}(10^3)$ particles. Therefore, these methods cannot be employed for commercial and industrial applications, where more than $\mathcal{O}(10^5)$ particles/droplets are present. Nonetheless, the results of the simulations obtained with these methods allow a deep knowledge and understanding of the complex physical phenomena encountered in dispersed multiphase flows, becoming really useful, since their results can be employed to derive closures and models for high-level numerical methods like the ones described below.

- **Eulergian-Lagrangian method.** The Eulerian-Lagrangian method is the natural and standard way to deal with dispersed multi-phase flows where millions of small particles are present. It is based on a point-particle approach, where particles or groups of identical particles, known as parcels, are tracked individually throughout all the computational domain. Hence, in this method the dispersed phase is represented employing a Lagrangian reference framework, while the continuous phase is solved using the classical Eulerian frame [1]. Due to its nature, in this modelling technique the particles or droplets have a much smaller size than the computational cells of the Eulerian grid. Consequently, all the interactions between the continuous and the dispersed phases are not explicitly resolved, since these interactions occurs at sub-grid level. Therefore, all the interactions and exchanges between both phases are solved employing mathematical models derived from analytical analysis, experiments or fully resolved simulations. Nevertheless, this modelling technique is the only one that allows simulations where hundreds of millions of poly-dispersed particles are present.

- **Euler-Euler (Two-fluid) method.** Unlike the previous presented method, in this one the particles of the dispersed phase are not resolved individually, but as an average using an Eulerian reference approach. In this method all the phases of the multi-phase flow are considered as inter-penetrating fluids. Hence, in the Euler-Euler model the phases coexist at the same place at same time, and it is the volume fraction quantity which determines the relative amount of each phase in every control volume of the Eulerian mesh. The solved equations are constructed via averaging technique, resulting in a set of equations for each phase, it is, for each phase present in the simulation the continuity, the momentum and the energy conservation equations are solved. Therefore, the computational load will not be function of the number of particles or droplets. Although this represents a clear advantage of the method computationally speaking, it is also a clear downside, since only the average size and the average velocity of the particles (or droplets) are represented. There is also a *closing* problem, similar to those in single-phase flow RANS simulations, and the averaged form of all the interactions between the phases must be modelled. The details of the flow around individual particles are lost in the averaging procedure. For very dilute regimes, the assumptions made in order to derive the Euler-Euler governing equations are violated, since the mean free path of the particles tends

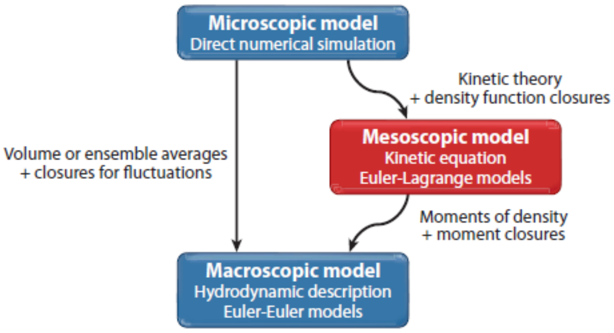


Figure 3.1: Scheme of multi-phase flow modelling approaches [4].

to infinite. In the same way, these assumptions are also violated for very dense flows. Moreover, in this approach the mesh dependency issues are important. Generally, this method is the preferred for large systems and is well suited for applications like fluidized-bed or bubble columns [2].

- **Eulerian Moment Equation.** Although this method employs an Eulerian reference framework for both dispersed and continuous phases, it should not be confused with the previous Euler-Euler Two-fluid method, since the transported fields are different. This model is based on the *method of moments*, so the transported equations are the velocity-mass moments [3]. The scheme from the Annual Review of Fox [4] depicted in Fig. 3.1 shows very clearly the difference between these models. The Euler-Euler Two-fluid method is derived by averaging from the Microscopic model (previously referred as fully resolved methods, since all the phases are fully resolved), while the current model is obtained deriving the moments of the kinetic equation for particles in which is based the Eulerian-Lagrangian method. One intrinsic difficulty of this method is that when deriving the moment transport equations, it does no matter how many moments are solved, the higher solved moment of order n always will have the presence of moment $n{+}1$. Hence, a moment closure is required. The *standard* closures present problems for inertial particles in dilute flows, so this method can not offer accurate simulations of dilute dispersed multi-phase flows [4, 5]. The method also present difficulties dealing with non-equilibrium situations like crossing jets or jet-wall interactions. In the last years, some efforts have been devoted to these issues, and methods like the Quadrature-Based moment closure have shown promising results [5].

Among all the numerical models previously described, the work presented in this chapter is devoted to the Lagrangian-Eulerian model. The reason to choose this model between the different models previously presented is basically determined by the final purpose of the numerical tool: since the motivation was to simulate dispersed multi-phase flows for applications like spray jets, inhaled medicines, evaporative cooling or cyclone separators, where up to $\sim \mathcal{O}(10^6) - \mathcal{O}(10^9)$ particles or droplets are present, the Eulerian-Lagrangian model was the most well-suited option. As detailed before, fully resolved methods are limited to simulations up to $\sim \mathcal{O}(10^3)$ particles/droplets due to computational resources, so they are not an option. Regarding both Eulerian methods, they are discarded because these techniques present some undesirable lacks in accuracy and not reliability in the obtained results for certain simulations due to issues related with their formulation and derivation, although they are very interesting from a computational point of view, since their computational cost is independent of the number of particles. The Lagrangian-Eulerian method allows to easily capture and account for non-linearities, multi-scale interactions and non-equilibrium effects in poly-dispersed multi-phase flows [1]. For example, this method naturally takes into account the non-linear dependence of particle acceleration or particle evaporation, is able to deal without difficulties with poly-dispersed flows, and it is the only method which simulates accurately dispersed flows with inertial particles [5]. Moreover, the Lagrangian-Eulerian approach minimizes the artificial diffusion that appears in the transported fields when Eulerian grids are employed. This method presents only two major drawbacks: its computational cost depends on the number of particles present in the simulation, and it is not well-suited to calculate statical data in Eulerian reference due to statistical noise. The only way to reduce this statistical noise is increase the number of particles, but this will increment the computational cost of the simulation.

Hence, along the present chapter of this thesis, the mathematical and numerical implementation of the physical models required for the development of a Lagrangian-Eulerian solver in the in-house CFD code TermoFluids are detailed and explained. The present chapter is structured as follows: first, the detailed mathematical formulation of the Lagrangian-Eulerian method for dispersed multi-phase flows is presented. Then, since the correct evaluation of the continuous phase properties at the particle's location is a key aspect of the method, interpolation methods for unstructured meshes (i.e. interpolation methods for scattered data) are presented and analysed. The following section is devoted to the modelling of sub-grid dispersion of Lagrangian particles in the framework of LES modelling. Finally, some benchmark cases demonstrating all the features and models described throughout the chapter are presented and discussed. The chapter is closed with the final conclusions derived from the work carried out.

3.2 Mathematical formulation

This section is devoted to the mathematical formulation required for the implementation of a Lagrangian-Eulerian method for the numerical simulation of dispersed multi-phase flows. First, the Lagrangian equations describing the thermal and fluid dynamic behaviour of the discrete phase are presented. Following, the governing equations of the continuous (carrier phase) are detailed. Finally, the coupling between both phases is presented and analysed.

3.2.1 Particle motion

The motion of particles and droplets in a fluid using a Lagrangian framework can be described by classical equations of motion, i.e., Newton's law. Hence, a detailed knowledge of all the forces acting over the particles is required. Pioneering work in that field was carried out by Basset [6], Boussinesq [7] and Oseen [8]. Therefore, the equation of motion for particles derived from their work is known as the **BBO-equation**. A detailed analysis and derivation of the BBO-equation in non-uniform flow for small rigid particles was done by Maxey and Riley [9]. The equation proposed by Maxey and Riley reads as:

$$m_p \frac{d\mathbf{v_p}}{dt} = \mathbf{F_d} + \mathbf{F_b} + \mathbf{F_{am}} + \mathbf{F_{pg}} + \mathbf{F_{hist}} \tag{3.1}$$

where m_p is the particle mass and $\mathbf{v_p}$ the particle velocity. The forces appearing in the right-hand side of Eq. (3.1) are respectively the drag force, the buoyancy force, the added mass force, the pressure gradient force and the Basset history force. This equation was derived under the assumption of low Reynolds number and a particle size smaller than the Kolmogorov length-scale. Apart from these forces, often other forces like lift or electrostatic force are also taken into account. Moreover, corrections of the presented forces to extend its validity to higher Reynolds number are also introduced. Hence, in general, the ordinary differential equations required to describe the particle motion are:

$$\frac{d\mathbf{x_p}}{dt} = \mathbf{v_p} \tag{3.2}$$

$$m_p \frac{d\mathbf{v_p}}{dt} = \sum_i \mathbf{F_i} \tag{3.3}$$

where $\mathbf{x_p}$ is the particle's position and $\sum_i \mathbf{F_i}$ is the sum of all the relevant forces acting over the particle. Following, the most common and important forces are introduced and detailed.

Drag force

Usually, the most relevant and important force acting over a particle and driving its motion is the drag force, which is expressed as:

$$\mathbf{F_d} = \frac{1}{2}\rho_c C_D A \left|\mathbf{u} - \mathbf{v}\right| (\mathbf{u} - \mathbf{v}) \tag{3.4}$$

where ρ_c is the continuous phase density, A the cross-section of the particle in the direction of the relative velocity, $\mathbf{u}$ the continuous phase velocity at particle's position and C_D the drag coefficient. If the particle is assumed spherical, particle mass is $m_p = \frac{\rho_p \pi d_p^3}{6}$ and the cross-section area $A = \frac{\pi d_p^2}{4}$. Then:

$$\mathbf{f_d} = \frac{F_d}{m_p} = \frac{3}{4}\frac{\rho_c C_D}{\rho_p d_p} \left|\mathbf{u} - \mathbf{v}\right| (\mathbf{u} - \mathbf{v}) \tag{3.5}$$

where ρ_p is the particle density and d_p the particle diameter. The drag coefficient value C_D is a function of the particle Reynolds number Re_p, defined as:

$$Re_p = \frac{\rho_c \left|\mathbf{u} - \mathbf{v}\right| d_p}{\mu_c} \tag{3.6}$$

where μ_c is the dynamic viscosity of the continuous phase. Replacing Eq. (3.6) in the previous Eq. (3.5), the following expression for the drag force is obtained:

$$\mathbf{f_d} = \frac{3}{4}\frac{\mu_c C_D Re_p}{\rho_p d_p^2} (\mathbf{u} - \mathbf{v}) \tag{3.7}$$

For small particle Reynolds numbers ($Re_p < 1$) the viscous effects dominates and an analytical solution for the drag force acting over a spherical particle was derived by Stokes [10]:

$$\mathbf{F_d} = 3\pi \mu_c d_p (\mathbf{u} - \mathbf{v}) \tag{3.8}$$

$$\mathbf{f_d} = 18\frac{\mu_c}{\rho_c d_p} (\mathbf{u} - \mathbf{v}) \tag{3.9}$$

In this regime, known as Stokes flow, the drag coefficient can be obtained from Eq. (3.7) and Eq. (3.9):

$$C_D = \frac{24}{Re_p} \tag{3.10}$$

For Re_p larger than unity inertial effects begin to be relevant, appearing flow separation around the particle and the analytical expression derived by Stokes does not hold any more. First, around $Re_p \approx 24$ appears the first vortices behind the spherical particle, which are symmetric. As increasing the Reynolds number, around $Re_p \approx 130$, these vortices start to be unstable and periodic. Until $Re_p \approx 1000$, the pressure in the wake progressively reduces, increasing the pressure drag. This flow regime is known as the transition region. From $Re_p \approx 1000$ almost all the drag force is due to the pressure gradient (form drag), and the relevance of skin friction (shear drag) in the total drag force is almost negligible, becoming constant the drag coefficient around $C_D \approx 0.44$. This drag coefficient value is valid until the critical Reynolds number $Re_p \approx 3e6$, since the size and form of vortices in the wake for this Reynolds range remains practically constant. This regime is known as the Newton regime. Above $Re_p \approx 3e6$, the separation point is displaced downstream due to the boundary layer becomes turbulent, and the drag coefficient is sharply reduced. Many correlations that fits the experimental curves more or less accurately have been derived. A classical correlation for the drag coefficient of spherical particles is the one proposed by Schiller and Naumman [11]:

$$C_D = \begin{cases} \frac{24}{Re_p}\left(1 + 0.15 Re_p^{0.687}\right), & \text{if } Re_p < 1000 \\ 0.44, & \text{if } Re_p \geq 1000 \end{cases} \tag{3.11}$$

Another common correlation for the drag coefficient of spherical particles is the one derived by Yuen and Chen in their analysis of evaporating liquid droplets [12]:

$$C_D = \begin{cases} \frac{24}{Re_p}\left(1 + \frac{1}{6} Re_p^{\frac{2}{3}}\right), & \text{if } Re_p < 1000 \\ 0.44, & \text{if } Re_p \geq 1000 \end{cases} \tag{3.12}$$

The two drag coefficient correlations presented in Eq. (3.11) and Eq. (3.12), as well as the analytical solution proposed by Stokes, are compared against the experimental data from Brown et al. [13] in Fig. 3.2:

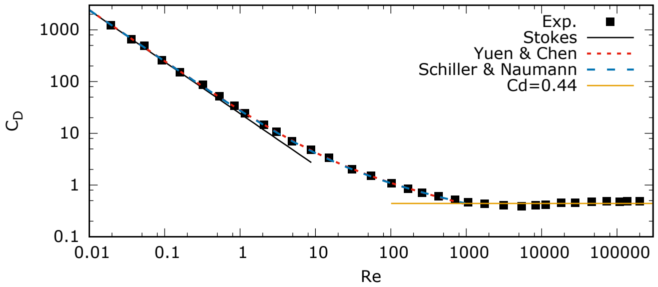


Figure 3.2: Drag coefficient correlations as function of particle Reynolds numbers.

These correlations are valid for smooth spherical particles in normal conditions. Nonetheless, the drag coefficient values obtained from these correlations may be modified by multiple reasons such as: roughness, particle shape, compressibility, rarefaction effects, near-wall effects or particle concentration effects. All these effects over the drag coefficient can only be taken into account using correlations derived from detailed experiments, or more recently, from Direct Numerical Simulations (DNS) like the work for non-spherical particles of Zastawny et al. [14].

Pressure gradient and buoyancy forces

The force exerted over the particle due to the pressure gradient and the shear stress of the carrier phase can be expressed as:

$$\mathbf{F_{pgb}} = m_c \left(-\nabla p + \nabla \cdot (\boldsymbol{\tau}) \right) \tag{3.13}$$

where m_c is the mass of the continuous phase displaced by the particle. Employing the relation $m_c = \frac{\rho_c}{\rho_p} m_p$ and substituting the Navier-Stokes momentum equation (see Eq. (2.15)) in Eq. (3.13):

$$\mathbf{F_{pgb}} = \frac{\rho_c}{\rho_p} m_p \left(\frac{D\mathbf{u}}{Dt} - g \right) \tag{3.14}$$

Expression Eq. (3.14) can be splitted in two forces, one known as the Buoyancy force:

$$\mathbf{F_b} = -\frac{\rho_c}{\rho_p} m_p g \tag{3.15}$$

and a second one named the Pressure gradient force, since historically in its derivation the shear stress term of the continuous phase was neglected:

$$\mathbf{F_{pg}} = \frac{\rho_c}{\rho_p} m_p \left(\frac{D\mathbf{u}}{Dt} \right) \tag{3.16}$$

Notice that the relevance of these forces is proportional to the density ratio of both phases ${}^{\rho_c}/_{\rho_p}$. Hence, if the ratio of material densities is small ${}^{\rho_c}/_{\rho_p} \ll 1$, both forces can be safely neglected. This is the typical situation in gas-liquid and gas-solid multiphase flows. On the other hand, for bubbles or slurry flows these forces are important and should be taken into account.

Added mass force

When a body is accelerated or decelerated there is also a mass of the surrounding phase that is accelerated or decelerated. The force required to change the velocity of this additional mass is known as the Added mass force. The expression accounting for this force is:

$$\mathbf{F_{am}} = 0.5 \rho_c V_p \left(\frac{D\mathbf{u}}{Dt} - \frac{d\mathbf{v}}{dt} \right) \tag{3.17}$$

This expression is derived analytically, and is only valid for small Reynolds number. In the work of Odar and Hamilton [15] is proposed a correction factor C_{am} for Eq. (3.17):

$$C_{am} = 2.1 - \frac{0.132}{0.12 + A_c^2} \tag{3.18}$$

where A_c is the *acceleration parameter* and is defined as:

$$A_c = \frac{|\mathbf{u} - \mathbf{v}|^2}{d_p \frac{d|\mathbf{u}-\mathbf{v}|}{dt}} \tag{3.19}$$

Numerous previous works analyzed the relevance of the Added mass force [15, 16], and it is found that as for the Pressure Gradient force, for small and heavy particles ($\rho_c/\rho_p \ll 1$), the influence of the Added mass force compared to the drag force is negligible.

Basset force

The Basset force, as the Added mass force, is related with the acceleration and deceleration of the relative velocity and appears due to the delay in the development of the boundary layer as a result of the variation of the relative velocity between the particle and the surrounding phase. The Basset force is expressed as:

$$\mathbf{F_{hist}} = \frac{3}{2}\sqrt{\pi \rho_c \mu_c} d_p^2 \int_0^t \frac{\frac{d(\mathbf{u}-\mathbf{v})}{dt'}}{\sqrt{t-t'}} dt' + \frac{(\mathbf{u}-\mathbf{v})_0}{\sqrt{t}} \tag{3.20}$$

As for the Added mass force, Odar and Hamilton [15] proposed a correction coefficient for the Basset force:

$$C_{hist} = 0.48 - \frac{0.52}{(1 + A_c)^3} \tag{3.21}$$

As can be noticed from Eq. (3.20), the value of the Basset force is function of the *historic* of the relative velocity, being also known as History force. Since its calculation requires an integration along all the trajectory of the particle at each time-step, or at least a relevant part of it, its calculation is quite time-consuming, and could be computationally prohibitive for a large amount of particles. Nonetheless, as shown in previous works [15, 16], for large density ratios ($\rho_c/\rho_p \ll 1$) the Basset force can be neglected. However, it should be accounted in cases like bubbles or neutrally buoyant particles [17].

Body forces

There are various relevant body forces acting on particles that could be important depending on the application. Following, some of them are briefly detailed:

Gravity force The most usual body force is the gravity force, which is expressed as:

$$\mathbf{F_g} = m_p \mathbf{g} \tag{3.22}$$

being $\mathbf{g}$ the gravitational acceleration vector.

Coloumb force The Coloumb force appears when the particle has an electric charge and it moves inside an electric field. Its value is the product of the charge q and the intensity of the electric field $\mathbf{E}$:

$$\mathbf{F_c} = q\mathbf{E} \tag{3.23}$$

Coloumb force can also arises due to the so called *dielectric effect*. For example, in pharmaceutical aerosols, the deposition of the droplets is enhanced by this effect. When a particle with an electric charge moves along the respiratory airways, its charge induces an orientation in airways tissues, which in turns causes a net force over the droplets. If the dielectric constant ϵ_0 is very large, the Coloum force acting over the particle is as it was in free space with a mirror particle at the same distance of the closest airway tissue and the same electrical charge but with opposite polarity [18]. Hence, this force is known as Electrostatic image force, and its value can be calculated as:

$$\mathbf{F_{image}} = \frac{q^2}{16\pi\epsilon_0 \left|\mathbf{r_{p,w}}\right|^2} \frac{\mathbf{r_{p,w}}}{\left|\mathbf{r_{p,w}}\right|} \tag{3.24}$$

where $\mathbf{r_{p,w}}$ is the vector pointing from the particle to the closest surface.

Thermophoretic force The Thermophoretic force acts when there are important temperature gradients in the carrier phase. The particle tends to be moved from the hot region to the cold one due to density differences and rarefaction effects. For example, in exhaust pipes this force enhances the deposition of particles in walls. This force is only relevant for small particles.

Compressibility and rarefaction effects

The compressibility effects on the particle drag force will be function of the relative Mach number, defined as:

$$Ma_p = \frac{\mathbf{u} - \mathbf{v}}{c_c} \tag{3.25}$$

where c_c is the speed of sound in the carrier phase and is given by $c_c = \sqrt{\gamma RT}$, being γ the ratio of the specific heats. Compressibility effects will be relevant for $Ma_p > 0.6$ [19]. Such conditions can be expected in rocket engines, and under these circumstances, compressibility effects modify particle dynamics and should be taken into account. On the other hand, the relevance of the rarefaction effects over particles is related to the Knudsen number Kn, that is the ratio between the molecular mean free path λ of surrounding phase and particle diameter d_p. For an ideal gas, Kn can be also expressed as function of particle Mach Ma_p and particle Reynolds Re_p numbers:

$$Kn = \frac{\lambda}{d_p} = \sqrt{\frac{\pi\gamma}{2}}\left(\frac{Ma_p}{Re_p}\right) \tag{3.26}$$

Schaaf et al. [20] identified four different flow regimes as function of the Knudsen number. For values $Kn < 10^{-3}$ the flow can be considered as *continuum*, and the no-slip boundary condition can be applied at particle surface, since a large amount of molecules of the surrounding phase collide with it. For larger Knudsen numbers the continuum hypothesis cannot be considered any more. When the Knudsen number is in the range $10^{-3} < Kn < 0.25$, the regime is named *slip flow*, since there is a slip velocity between the particle and the fluid adjacent to its surface. If the Knudsen number is $Kn > 0.25$, the flow is treated as free molecules that impact and rebound at particle surface, but the molecular impacts near the particle are neglected. This flow regime is named *free molecular flow*. For a Knudsen number between $0.25 < Kn < 10$ the *Transitional flow* regime is found. In this flow regime the collisions of molecules close to the surface begin to be relevant. In general, rarefaction effects will be relevant for very small particles or for low-pressure environments. Unfortunately, there is no a universal correlation valid for all the flow regimes. In the *Stokes* regime, usually valid for very small particles, the *Cunningham* correlation factor (C_c) is commonly employed to correct the standard drag coefficient, obtaining a reduced one accounting for the rarefaction effects [21]:

$$C_c = 1 + 2Kn\left(1.257 + 0.4\exp\left(-\frac{0.55}{Kn}\right)\right) \tag{3.27}$$

where the final drag coefficient $C_{D,cor}$ is obtained as:

$$C_{D,cor} = \frac{C_D}{C_c} \tag{3.28}$$

As stated by Lothe et al. [22], compressibility effects usually are relevant for very high particle Reynolds number $Re_p \gg 1$, while rarefaction effects dominates at low Reynolds numbers.

Brownian motion

The Brownian motion effect is relevant for submicron particles. These very small particles are sensible to the *discrete* behaviour of the continuous phase since they notice the interaction with the molecules of the surrounding phase. Hence, the particles exhibit a *random* motion due to the collisions that they experience with the molecules of gas or liquid where they are immersed. The Brownian motion can be modeled as a Gaussian random white noise [23]. The i component of the force acting over a particle due to Brownian effect is:

$$F_{brown,i} = \zeta_i \sqrt{\frac{2k_b^2 T^2}{\widetilde{D}\Delta t}} \tag{3.29}$$

where ζ_i is a Gaussian distribution with zero-mean and unit-variance, T the absolute temperature of the fluid, k_b = 1.38e-23 J/K the Boltzmann constant and $\widetilde{D}$ the Brownian diffusion coefficient defined as:

$$\widetilde{D} = \frac{k_b T C_c}{3\pi\mu_c d_p} \tag{3.30}$$

Particle response time

A relevant parameter for the particle motion is the *particle response time* τ_p. This parameter characterizes the behaviour of a particle inside a flow, and it allows to better understand the evolution of its trajectory. In order to derive the *particle response time*, the particle acceleration equation (Eq. (3.3)) assuming only Drag force is employed:

$$m_p \frac{d\mathbf{v_p}}{dt} = \mathbf{F_d} \tag{3.31}$$

Dividing Eq. (3.31) at both sides by particle mass m_p:

$$\frac{d\mathbf{v_p}}{dt} = \frac{3}{4}\frac{\mu_c C_D Re_p}{\rho_p d_p^2}(\mathbf{u} - \mathbf{v}) \tag{3.32}$$

and simplifying Eq. (3.32):

$$\frac{d\mathbf{v_p}}{dt} = \frac{(\mathbf{u} - \mathbf{v})}{\tau_p} \tag{3.33}$$

where τ_p is the *particle response time* and it is defined as:

$$\tau_p = \frac{4}{3}\frac{\rho_p d_p^2}{\mu_c C_D Re_p} \tag{3.34}$$

If a particle initially at rest ($\mathbf{v_0} = 0$) is placed inside a flow with a velocity $\mathbf{u_c}$, then Eq. (3.33) has the following solution:

$$\mathbf{v} = \mathbf{u_c}\left(1 - \exp\left(-\frac{t}{\tau_p}\right)\right) \tag{3.35}$$

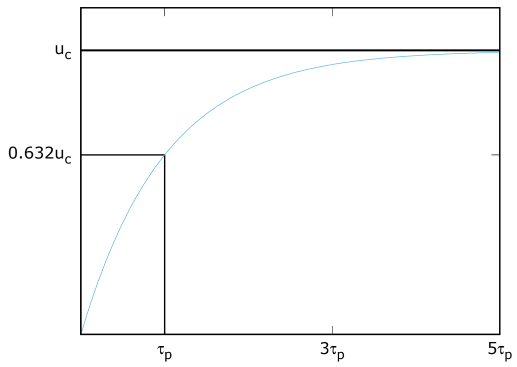


Figure 3.3: Particle response time (τ_p).

As can be seen in Fig. 3.3, τ_p is the time required for a particle with initial null velocity to achieve the 63.2% of the flow velocity $\mathbf{u_c}$. Hence, the *particle response time* will give an idea of how fast a particle will follow the variations of carrier phase velocity. In the Stokes regime, where the drag coefficient is $C_D = 24/Re_p$, the *particle response time* becomes:

$$\tau_p = \frac{\rho_p d_p^2}{18\mu_c} \tag{3.36}$$

This equation clearly shows that heavier particles have a larger τ_p and will require longer time to adapt their velocity to the changes in the velocity of the surrounding phase. On the other hand, light particles will easily follow the carrier phase. If a time-scale τ_c characterizing the surrounding flow is introduced, the non-dimensional number named *Stokes number* (St) is defined as the ratio of both time-scales:

$$St = \frac{\tau_p}{\tau_c} \tag{3.37}$$

3.2.2 Droplet heating and evaporation

When dealing with liquid droplets, there is not only an interchange of momentum between the particle and the surrounding phase, but also an interchange of mass and energy between both phases. Hence, two additional equations governing the evaporation/condensation of the droplet as well as its temperature evolution are required. The heat and mass transfer of the droplets are driven by complex physical phenomena that take place at droplet's surface and are highly coupled. The *Classical vaporization model* was derived by Godsave [24] and Spalding [25] assuming a constant droplet temperature and unitary Lewis number, among other simplifications.

This model is also known as the D^2*-law model*, since it states that the square of the diameter varies linearly with time. Although its 'simplicity', this model has been extensively employed to simulate evaporating droplets. Later, Abramzon and Sirignano [26] reviewed the *Classical vaporization model* in order to include the effects of Stefan flow and non-unitary Lewis number in the gas film. Nonetheless, in these two models is presumed an infinite liquid conductivity, and therefore assuming an uniform droplet temperature. Aiming to take into account possible temperature gradients inside the droplet, Bellan and Harstad [27] developed a model considering finite liquid thermal conductivity as well as non-equilibrium effects for evaporation based on the Langmuir-Knudsen law. More advanced models including the effect of possible recirculation inside the droplet [28] or for bi-component droplets [29] have been also presented. Nevertheless, for complex simulations with millions of evaporating particles there should be a compromise between the computational cost and the accuracy of the model, otherwise, realistic simulations can become unaffordable [28]. Several reviews of droplet heating and evaporation models have been done [28, 30] in order to compare and assess the behaviour of multiple models under different evaporating conditions.

In this thesis, the model assuming non-equilibrium conditions and infinite liquid conductivity of Bellan and Harstad [27, 30] is presented. The equations governing the energy and mass transfer of a single droplet are:

$$\frac{dT_p}{dt} = G\frac{6Nu\lambda_g}{c_{p,l}\rho_p d_p^2}\left(T_g - T_p\right) + \frac{Lv}{c_{p,l}}\frac{\dot{m}_p}{m_p} \tag{3.38}$$

$$\dot{m}_p = \frac{dm_p}{dt} = -Sh\pi d_p(\rho\Gamma)_g H_m \tag{3.39}$$

The subscript (p) refers to the droplet (*particle*), (g) to a property evaluated at the carrier (gas) phase, (l) to a liquid property of the droplet and (v) to the vapour phase of the evaporating droplet. Lv is the latent heat of evaporation and Γ_g is the binary diffusion coefficient. The non-dimensional Nusselt Nu and Sherwood Sh numbers are defined following the Ranz-Marshall correlations [31]:

$$Nu = 2 + 0.552 Re_p^{\frac{1}{2}} Pr_g^{\frac{1}{3}} \tag{3.40}$$

$$Sh = 2 + 0.552 Re_p^{\frac{1}{2}} Sc_g^{\frac{1}{3}} \tag{3.41}$$

From Eqs. (3.40) and (3.41) can be readily inferred that higher slip velocities between both phases ($Re\uparrow$) enhance the convective heat and mass transfer. Pr_g and Sc_g are, respectively, the *Prandtl* and *Schmidt* non-dimensional numbers, which are defined as:

$$Pr_g = \frac{\mu_g c_{p,g}}{\lambda_g} \qquad (3.42) \qquad\qquad Sc_g = \frac{\mu_g}{\rho_g \Gamma_g} \qquad (3.43)$$

The value H_m is defined as $H_m = \ln(1 + B_m)$ and represents the specific driven potential of a droplet to evaporate, similarly to $(T_g - T_p)$ in the droplet heating equation (Eq. (3.38)) and $(\mathbf{u} - \mathbf{v})$ in the particle velocity equation (Eq. (3.4)). B_m is the *Spalding mass transfer number* and is obtained from:

$$B_m = \frac{Y_{f,s} - Y_{f,g}}{1 - Y_{f,s}} \qquad (3.44)$$

where $Y_{f,g}$ is the vapour mass fraction of the fuel at free steam flow and $Y_{f,s}$ at droplet surface, calculated as:

$$Y_{f,s} = \frac{\chi_{f,s} W_f}{\sum_i \chi_{i,s} W_i} \qquad (3.45)$$

being χ_i the molar fraction and W_i the molecular weight of specie i. In the present model the vapour mole fraction at droplet surface is defined including non-equilibrium effects via the non-equilibrium Langmuir-Knudsen evaporation law:

$$\chi_s = \chi_{s,eq} - \left(\frac{L_k}{0.5 d_p}\right) \beta \qquad (3.46)$$

where L_k is the *Knudsen layer thickness* and β a non-dimensional evaporation parameter defined as:

$$L_k = \frac{\mu_g \sqrt{2\pi T_p R / W_v}}{\alpha_\epsilon Sc_g P_g} \qquad (3.47)$$

$$\beta = -1.5 Pr_g \tau_p \frac{\dot{m}_p}{m_p} \qquad (3.48)$$

In Eq. (3.47) α_ϵ is the molecular accommodation coefficient, assumed equal to unity (α_ϵ=1), R is the universal gas constant and P_g the absolute pressure at droplet position. The equilibrium mole fraction of vapour at droplet surface is obtained from $\chi_{s,eq} = {}^{P_{sat}}/_{P_g}$ and the vapour saturated pressure is calculated employing the $Clasius - Clapeyron$ correlation:

$$P_{sat} = P_{atm} \exp\left(\frac{LvW_v}{R}\left(\frac{1}{T_b} - \frac{1}{T_p}\right)\right) \qquad (3.49)$$

where P_{atm} is the atmospheric reference pressure and T_b the liquid phase normal boiling temperature. In Eq. (3.38) the factor G accounts for a reduction in the convective heat transfer due to the evaporation rate of the particle, and is calculated as:

$$G = \frac{\beta}{e^{\beta} - 1} \tag{3.50}$$

Hence, in case of non-evaporative particles $\beta = 0 \rightarrow G = 1$.

Reference conditions for properties evaluation

A crucial aspect on heating and evaporation models are the reference conditions under which the gas properties are evaluated. As reported by Miller et al. [30], several studies demonstrated that evaporation and heat transfer rates are sensitive to the employed reference conditions. The following gas properties are function of the temperature (T_r) and mass fraction (Y_r) reference:

$$\rho_g(Y_r, T_r) \qquad \mu_g(Y_r, T_r) \qquad c_{p,g}(Y_r, T_r) \qquad \Gamma_g(Y_r, T_r) \tag{3.51}$$

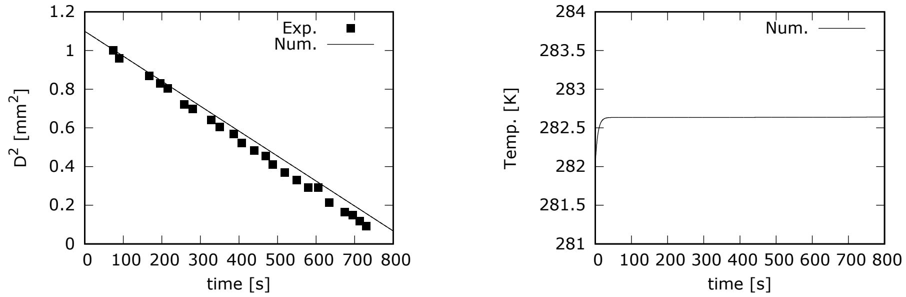
These reference conditions (Y_r, T_r) should be selected at some *point* between the conditions at droplet surface $(Y_{p,s}, T_{p,s})$ and the free steam flow (Y_∞, T_∞), it is:

$$Y_r = Y_{p,s} + A\left(Y_\infty - Y_{p,s}\right) \qquad T_r = T_{p,s} + A\left(T_\infty - T_{p,s}\right) \tag{3.52}$$

where A is a weighting coefficient ranging from 0 to 1. Firstly, Law and Williams [32] proposed the value $A = 1/2$. Lately, Yuen and Chen [12] recommended a value of $A = 1/3$, known as the *'1/3 rule'*, which is the most widely adopted criteria in the heating and evaporation models, e.g. the one of Abramzon and Sirignano [26]. On the other hand, the evaporation model presented above is formulated considering the wet bulb temperature T_{wb} as reference temperature and the mass fraction of the free stream flow (Y_∞) [30]. The wet bulb temperature T_{wb} is calculated using the following correlation:

$$T_{wb} = 137\left(\frac{T_b}{373.15}\right)^{0.68} log_{10}\left(T_g\right) - 45 \tag{3.53}$$

As can be seen, the selection of proper reference conditions for the evaluation of gas phase physical properties is still an open issue. Many authors have studied and analyzed the sensitivity of the evaporation models to the employed reference conditions [30, 33, 34]. In fact, as stated in the *Best Practice Guide for CFD of dispersed multiphase flows* by Sommerfeld et al. [2], for each simulation it is recommended to test the evaporation model case against single droplet evaporative experiments with similar conditions in order to check and assess the reference conditions employed to evaluate the gas properties.



(a) D^2 evolution.

(b) Temperature evolution.

Figure 3.4: Water droplet evaporation.

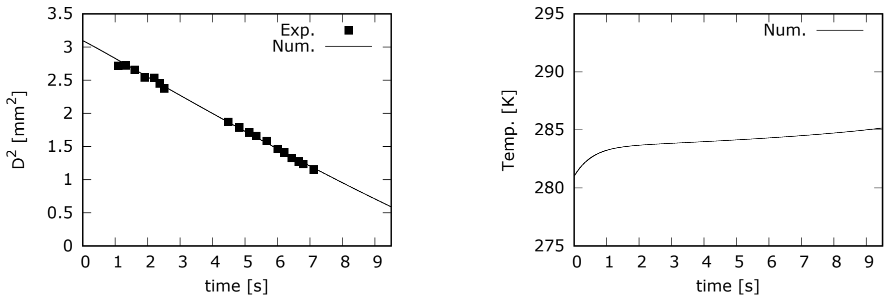
Validation

Several single droplet simulations have been carried out in order to validate the implemented evaporation model. Different conditions (from low to high evaporation rates) and several substances for the droplets have been tested. Additionally, for some cases a sensitivity analysis of the reference conditions used to evaluate the gas properties has been done.

The first test case is a single water droplet with initial diameter $d_{p,0} = 1.1\ mm$ and temperature $T_{p,0} = 282\ K$ [31]. It is immersed in quiescent air at $T_{air} = 298\ K$. This case presents a low-evaporation rate. The results are shown in Fig. 3.4. As can be seen, for this low-evaporation rate the $D^2 - law$ is fulfilled, since the squared diameter evolves linearly. The numerical results present a good agreement with the experimental data provided by Ranz et al. [31].

The second test case stands for a medium-evaporation rate and consists in a hexane droplet with an initial diameter $d_{p,0} = 1.76\ mm$ and initial temperature $T_{p,0} = 281\ K$. The droplet is inside a convective air flow at $T_{air} = 437\ K$ with $Re_{p,0} = 110$. As can be seen in Fig. 3.5, the numerical results also agree with the experimental results of Downing [35]. Although the evaporation rate is higher than in the previous case, the $D^2 - law$ still holds, since the wet boiling temperature is quickly reached.

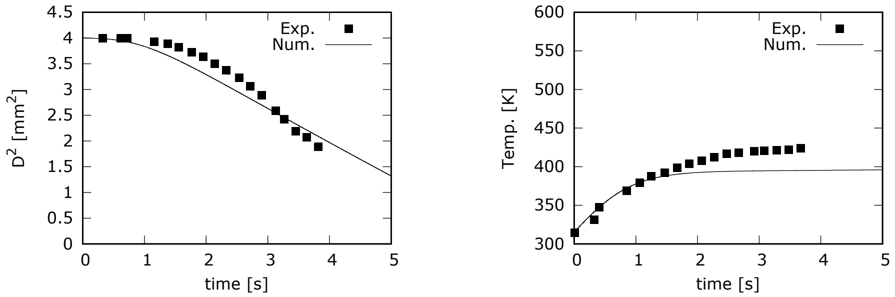
A third test is based on the experiment done by Wong and Lin [36]. It consists in a stand alone decane droplet with initial diameter $d_{p,0} = 2\ mm$ and initial temperature $T_{p,0} = 315\ K$, immersed in an air stream at temperature $T_{air} = 1000\ K$ with initial particle Reynolds $Re_{p,0} = 17$. The comparison between the numerical and the experimental results are shown in Fig. 3.6. In this case, with high-evaporation rate, two evaporation stages can be distinguished. Initially, there is a strong heat up of the droplet due to the large initial temperature difference $(T_{air} - T_{p,0})$ that not fulfils the $D^2 - law$. This first stage is followed by a second evaporative phase where droplet



(a) D^2 evolution.

(b) Temperature evolution.

Figure 3.5: Hexane droplet evaporation.



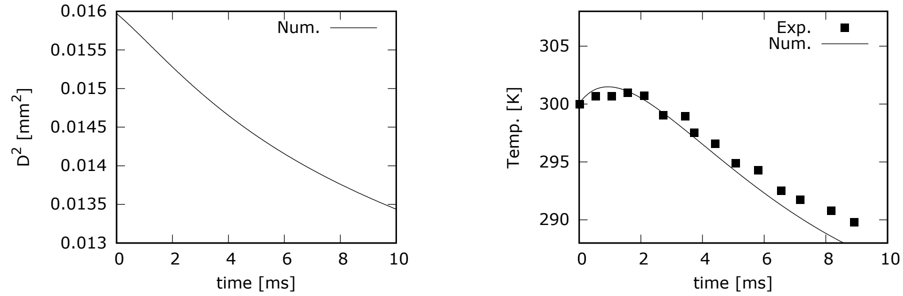
(a) D^2 evolution.

(b) Temperature evolution.

Figure 3.6: Decane droplet evaporation.

temperature is almost constant and the $D^2 - law$ is satisfied.

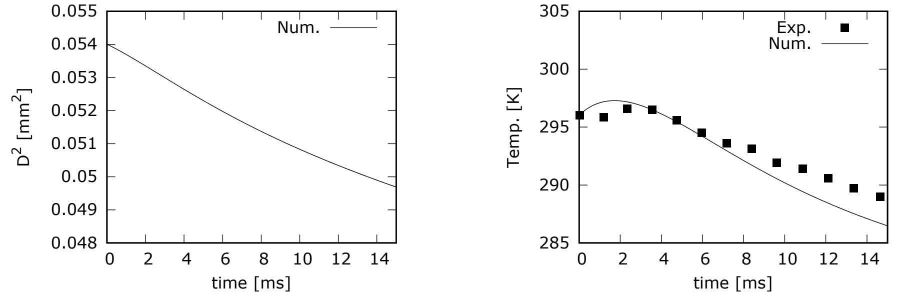
Two additional tests have been carried out based on the experiments done by Maqua et al. [37]. These experiment consisted in a mono-dispersed stream of droplets generated by Rayleigh disintegration of a liquid jet where acetone droplets are heated and evaporated in a hot air flow. In the first case, named *Third experiment* in the reference paper, a droplet with initial conditions $d_{p,0} = 126.4\ \mu m$ and $T_{p,0} = 294\ K$ is analysed. The second case studied stands for the one named *Fourth experiment*, where an acetone droplet with initial diameter $d_{p,0} = 232.4\ \mu m$ and initial temperature $T_{p,0} = 300\ K$ is studied. Functions for the relative velocity and ambient temperature evolution can be found in Maqua et al. [37]. Notice that in that cases the initial droplet diameter is an order of magnitude smaller than in the previous experiments, and non-equilibrium effects are more relevant [30]. The comparison between numerical



(a) D^2 evolution.

(b) Temperature evolution.

Figure 3.7: Acetone droplet evaporation *(Third Experiment).*



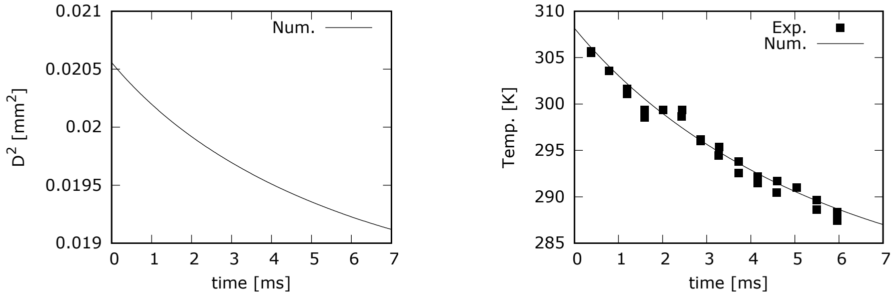
(a) D^2 evolution.

(b) Temperature evolution.

Figure 3.8: Acetone droplet evaporation *(Fourth Experiment).*

and experimental results are shown in Figs. 3.7 and 3.8. As can be seen, the droplet temperature evolution is well captured by the heating and evaporation model in both cases.

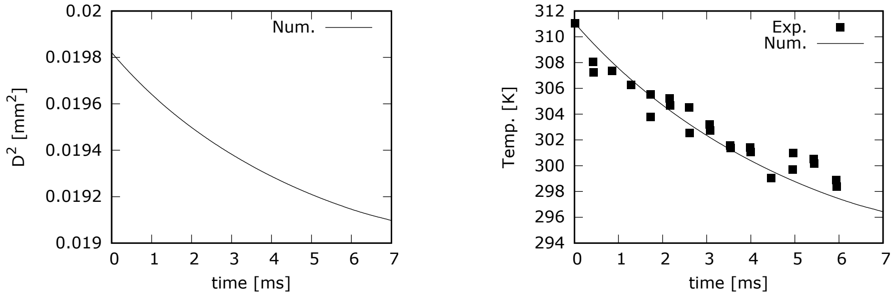
The heating and evaporation model is also tested against two experiments carried out by Sazhin et al. [29]. The experiments were done at the same experimental facility than [37]. The first test consists in an acetone droplet with initial temperature $T_{p,0} = 308.25\ K$ and diameter $d_{p,0} = 143.4\ \mu m$ surrounded by air at $T_{air} = 294.65\ K$. The second experiment simulated is an ethanol droplet with initial diameter $d_{p,0} = 140.8\ \mu m$ and temperature $T_{p,0} = 311.15\ K$, evaporating in air at $T_{air} = 295.15\ K$. The main difference with the experiments of [37] is the air temperature. Results of the simulations are depicted at Figs. 3.9 and 3.10, and as can be seen, the numerical model reproduce fairly well the experimental results.



(a) D^2 evolution.

(b) Temperature evolution.

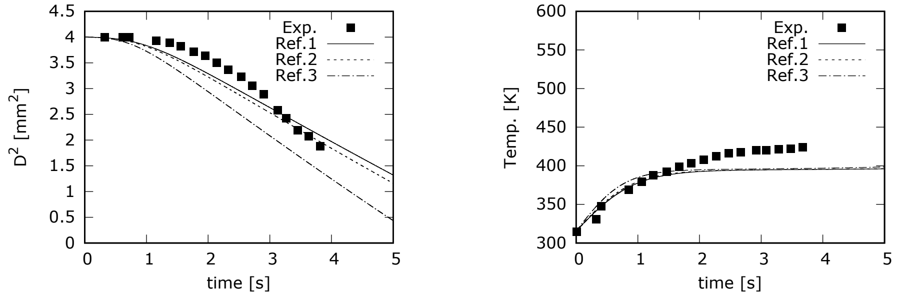
Figure 3.9: Acetone droplet evaporation.



(a) D^2 evolution.

(b) Temperature evolution.

Figure 3.10: Ethanol droplet evaporation.



(a) D^2 evolution.

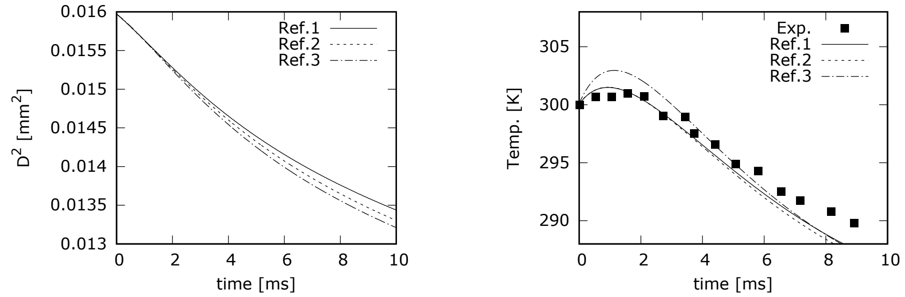
(b) Temperature evolution.

Figure 3.11: Decane droplet evaporation. *(Ref. conditions comparison).*

Finally, in order to illustrate the sensitivity of the heating and evaporation model to the reference conditions employed to calculate the continuous phase properties, as stated in Section 3.2.2, three of the experiments presented before are simulated using different reference conditions:

- *Ref.1*: employing the wet-bulb temperature as suggested by Miller et al. [30], it is, $T_r = T_{wb}$ and $Y_r = Y_\infty$
- *Ref.2*: using the boiling temperature of the droplet. This criteria could be employed if does not exist any reliable correlation to calculate the wet-bulb temperature. Hence, $T_r = T_b$ and $Y_r = Y_\infty$
- *Ref.3*: where *'1/3 rule'* is employed. Therefore, $T_r = T_{p,s} + 1/3\,(T_\infty - T_{p,s})$ and $Y_r = Y_{p,s} + 1/3\,(Y_\infty - Y_{p,s})$

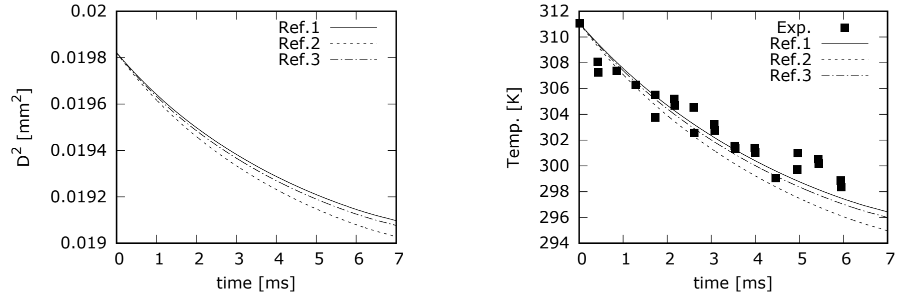
The results employing these three different reference conditions for the acetone droplet of Wong and Lin [36], the *third experiment* of Maqua et al. [37] and the ethanol droplet of Sazhin et al. [29] are shown in Figs. 3.11 to 3.13 respectively. As can be seen, the best results are obtained employing the reference conditions *Ref.1*. While in the first and second case, the reference conditions *Ref.2* seems to perform better than *Ref.3*, in the third case the results obtained using *Ref.3* are closer to those obtained with *Ref.1*. This could be explained because while in the two first cases the droplet is evaporating in hotted air, in the last case the ambient is only at $T_{air} = 295.15\ K$ and if the boiling temperature is employed to evaluate the continuous phase properties, this temperature is overestimated, leading a higher evaporation rate. This could be also the reason why in the first case the *Ref.3* over-predicts the evaporation rate. Since the air is at $T_{air} = 1000\ K$, the *'1/3 rule'* employs a temperature that is larger than the wet boiling temperature and the boiling temperature, which seem closer to



(a) D^2 evolution.

(b) Temperature evolution.

Figure 3.12: Acetone droplet evaporation *(Third Experiment) (Ref. conditions comparison).*



(a) D^2 evolution.

(b) Temperature evolution.

Figure 3.13: Ethanol droplet evaporation *(Ref. conditions comparison).*

the real conditions. Summarizing, the heating and evaporation models of droplets are fairly sensible to the reference conditions, and special attention to this point should be paid for each case.

3.2.3 Numerical integration methods for particles

The ODEs describing the particle motion (see Eqs. (3.2) and (3.3)) as well as the equations governing the heating and evaporation of droplets (Eqs. (3.38) and (3.39)) require numerical integration, due to its complexity to be analytically integrated. As detailed in Section 3.2.1, the *particle response time* τ_p determines how the particle will react to the variations in the continuous phase. In the same way, response times (or time-scales) for particle heating τ_c and evaporation τ_m can be defined. If Eqs. (3.38) and (3.39) are rearranged as:

$$\frac{dT_p}{dt} = \frac{(T_g - T_p)}{\tau_c} - \frac{Lv}{c_{p,l}} \frac{1}{\tau_m} \tag{3.54}$$

$$\frac{dm_p}{dt} = -\frac{m_p}{\tau_m} \tag{3.55}$$

it follows that definition for heating and evaporation particle time-scales are:

$$\tau_c = \frac{c_{p,l}\rho_p d_p^2}{6GNu\lambda_g} \tag{3.56}$$

$$\tau_m = \frac{3Sc_g\rho_p d_p^2}{6\mu_g ShH_m} \tag{3.57}$$

The time-scales of the particles can be several orders of magnitude smaller than the required time-step to integrate de continuous flow phase (Δt_f), especially in evaporating cases, where particles can become very small. Hence, if both continuous and dispersed phases are integrated employing the most restrictive time-scale of both phases, the simulations can become very expensive and computationally unaffordable. In order to avoid this issue, a smart and efficient time-integration strategy needs to be adopted: in a first step, the integration time-step for each Lagrangian particle τ_{min} is selected as its most restrictive time-scale multiplied by a security factor:

$$\tau_{min} = \min(sf_p\tau_p, sf_c\tau_c, sf_m\tau_m) \tag{3.58}$$

where sf_i is a security factor for each time-scale. After that, if this value is larger than the time-step of the continuous phase ($\tau_{min} \geq \Delta t_f$), the particle is integrated in one-step employing the time-step of the flow solver $\Delta t_p = \Delta t_f$. On the other hand, if the most restrictive particle time-scale is smaller than the continuous phase time-step ($\tau_{min} < \Delta t_f$), the particle is integrated multiple times during one continuous phase iteration. Then, the particle time step is selected between τ_{min} and the remaining time to be integrated Δt until the time-step of continuous phase is reached (Δt_f):

$$\Delta t_p = \min(\tau_{min}, \Delta t) \tag{3.59}$$

At first sub-step the time to be integrated Δt is set equal to flow time-step $\Delta t = \Delta t_f$ and at the end of each sub-step it is updated as:

$$\Delta t = \Delta t - \Delta t_p \tag{3.60}$$

In Fig. 3.14 this time-integration strategy is schematically depicted. At time $(n-1)$ the most restrictive time-scale of the particle (τ_{min}) is larger than the flow time-step (Δt_f) and the particle is integrated in a single step using $\Delta t_p = \Delta t_f$. On the other

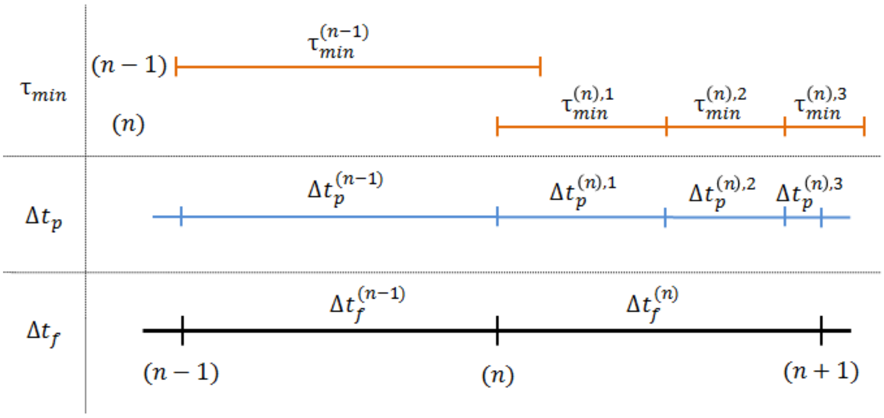


Figure 3.14: test

hand, at time (n) the most restrictive time-scale of the particle (τ_{min}) is smaller than the flow time-step (Δt_f) and the particle is integrated in three sub-steps. During these sub-steps the continuous phase *seen* by the particle is *frozen*. That idea is similar to the one followed by Apte et al. [38] and Chrigui et al. [39].

Hence, since the equations describing the thermal and fluid dynamic behaviour of the dispersed phase have to be numerically integrated, it is of interest to study and analyse different numerical integration methods. With that aim, the benchmark case presented by Mazumdar et al. [40] has been selected. The case consists in a spherical solid object submerged in a fluid that is projected downwards vertically with an initial velocity u_{ini}. The spherical object has a smaller density than the surrounding fluid and therefore it will tend to float. Hence, the relevant forces acting over the solid sphere are: added mass, drag, gravity, and buoyancy. A schematical representation of the configuration is depicted in Fig. 3.15. Since the problem can be considered 1D, the Newton's law describing its motion can be expressed as:

$$\frac{du}{dt} = \frac{\rho_p - \rho_g}{\rho_p + \rho_g C_{am}} g - \frac{3 C_D \rho_g}{4 d_p} \frac{u\,|u|}{\rho_p + \rho_g C_{am}} \tag{3.61}$$

If both drag and added mass coefficients are considered constants, the Eq. (3.61) can be rewritten as:

$$\frac{du}{dt} = A - B u\,|u| \tag{3.62}$$

where coefficients A and B are constant and can be defined as:

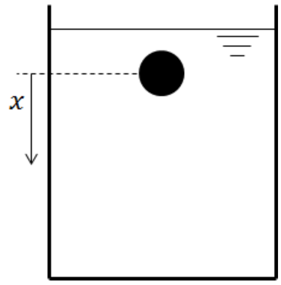


Figure 3.15: Scheme of the case studied for the numerical integration schemes.

$$A = \frac{\gamma-1}{\gamma+C_{am}} g \tag{3.63}$$

$$B = \frac{3C_D}{4d_p(\gamma+C_{am})} \tag{3.64}$$

where $\gamma = \rho_p/\rho_g$ is the density ratios. If Eq. (3.62) is split in two different equations: one for when the sphere is moving downwards (Eq. (3.65)), and another for when it is moving upwards (Eq. (3.66)), these expressions can be integrated analytically, allowing to compare the analytical solution with the numerical results.

$$\frac{du}{dt} = A - Bu^2 \tag{3.65}$$

$$\frac{du}{dt} = A + Bu^2 \tag{3.66}$$

The present study is focused on explicit temporal integration schemes. However, when solving the particle trajectory equations, one question arises: if first it is solved the velocity equation (Eq. (3.3)), obtaining $\mathbf{v_p^{n+1}}$, then the particle position equation (Eq. (3.2)) can be integrated *explictly* employing $\mathbf{v_p^n}$ or *implicitly* using the just calculated new time-step velocity $\mathbf{v_p^{n+1}}$. Hence, these two possibilities are included in the study presented below. The analysed combinations of temporal integration schemes are summarized in Table 3.1, where **fE1** stands for *Forward Euler* 1^{st} *order*, **bE1** for *Backward Euler* 1^{st} *order*, **AB2** means *Adams-Bashforth* 2^{nd} *order*, **AM2** is *Adams-Moulton* 2^{nd} *order* and **RK2** refers to *Runge-kutta* 2^{nd} *order* (specifically, the Heun's method). Hence, in the second and fourth cases the position equation is integrated implicitly, while in all other cases both equations are integrated explicitly. For the present study the following parameters have been considered: $\rho_p = 600\ kg/m^3$, $\rho_g = 1000\ kg/m^3$, $d_p = 0.01\ m$, $u_{ini} = 2.73\ m/s$, $C_D = 0.4$ and $C_{am} = 0.4$. A fixed time-step of $\Delta t = 0.001$ has been chosen. The trajectories of the solid sphere obtained

Table 3.1: Combination of temporal integration schemes.

Comb.	Velocity eq.	Position eq.
1	fE1	fE1
2	fE1	bE1
3	AB2	AB2
4	AB2	AM2
5	RK2	RK2

Table 3.2: Numerical and analytical solutions for constant time-step $\Delta t = 0.001\ s$.

Comb.	Maximum depth $[mm]$	$\mathrm{Error}_{\mathrm{md}}$ [%]	Max depth time $[s]$	Immersion time $[s]$
Exact	67.6681	0	0.132636	0.383224
1	66.7868	1.30245	0.13	0.379
2	64.0568	5.33684	0.133	0.371
3	67.6416	0.03922	0.132	0.384
4	66.2762	2.05691	0.133	0.379
5	67.6808	0.01868	0.133	0.384

for the different numerical integration schemes as well as the analytical solution are depicted in Fig. 3.16. At first glance, can be seen that combinations 3 and 5 are the ones with better accuracy. In order to quantify the accuracy of the temporal integration schemes, the error in the maximum depth obtained from each combination has been calculated. This error has been defined as:

$$error_{md} = \frac{|\text{max. depth analytical} - \text{max. depth numerical}|}{\text{max. depth analytical}} 100 \tag{3.67}$$

Table 3.2 summarizes the maximum depth, the time when the maximum depth is achieved and the total time that the solid sphere remains submerged obtained for the different numerical integration schemes and the exact solution. The error in the maximum depth obtained using Eq. (3.67) is also presented. Analysing these results the previous presented question about if it is better to integrate the position equation *explictly* or *implictly* is clearly answered. Comparing the results obtained from combination 1 versus combination 2, as well as confronting combinations 3 and 4, it is clear that if the velocity equation is integrated *explicitly*, the position equation should be also integrated *explicitly*. As expected, the combinations employing 2^{nd} order schemes perform better than the 1^{st} order schemes.

From Table 3.2 can be stated that combinations 3 and 5 are the best choices among the presented options. Although both temporal integration schemes are 2^{nd} order,

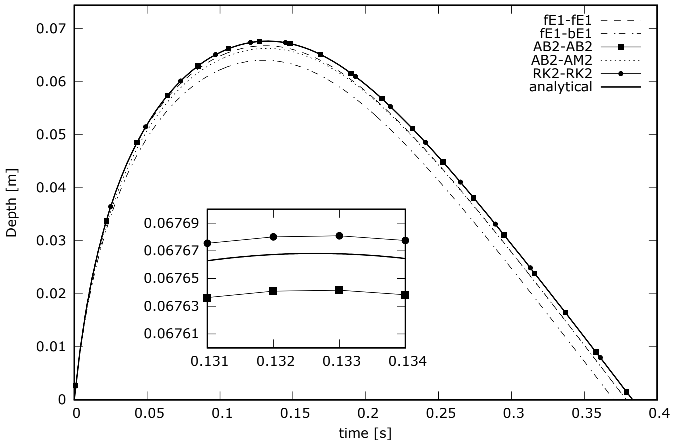


Figure 3.16: Particle trajectory for constant $\Delta t = 0.001\ s$

the computational effort required for the Adams-Bashforth (AB) scheme is smaller, since Runge-Kutta (RK) schemes of 2^{nd} order requires two function evaluations per iteration. However, AB schemes present a downside for the current application that RK schemes do not have. As explained previously, the time-step employed for the particle integration is estimated dynamically, meaning that it would change at each iteration. For solid particles this time-step variation will be small, since it is only subjected to variations in the slip velocity or possible changes in the continuous phase time-integration step, but the time-step variation between two consecutive iterations can be relevant for evaporative droplets. AB schemes are a class of explicit Linear Multi-Step methods characterized by employing not only the last time, but also previous steps in order to gain accuracy. In its derivation the *temporal mesh* is considered uniform, it is, the time-step is constant. Hence, when two or more consecutive iterations have a different time-step Δt and a high order linear multi-step scheme is employed, there is an intrinsic error. In order to analyse this behaviour, a new numerical experiment has been designed. Now, the time-step is obtained from a bounded random distribution $\Delta t = \left[5 \times 10^{-4}, 1 \times 10^{-3}\right]$. The time-step variation is limited by a factor fr which limits the increase and decrease of the time-step. If this factor is set to unity the time-step would remain constant. Different values for the factor fr have been tested. For each case, several simulations have been done in order to obtain a statistical converged value for the mean value in the maximum depth error. The obtained results are depicted in Fig. 3.17, where the explicit-implicit combinations have been neglected. As can be seen in the figure, when a variable time-step is employed, the scheme AB2 suffers a decrease in its accuracy. As factor fr grows,

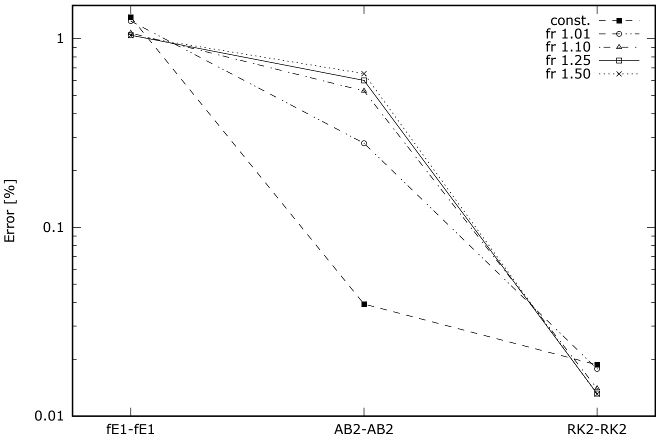


Figure 3.17: Mean maximum depth error [%] for bounded random $\Delta t = \left[5 \times 10^{-4}, 1 \times 10^{-3}\right]\ s$

Table 3.3: Mean time-step $\overline{\Delta t}$ for the variable random time-step analysis.

fr	$\overline{\Delta t}\ [s]$
const	1×10^{-3}
1.01	7.64×10^{-4}
1.10	7.50×10^{-4}
1.25	7.48×10^{-4}
1.50	7.49×10^{-4}

the error also increase. This is due to the variation of the time-step Δt between to consecutive iterations is allowed to be bigger. On the other hand, the fE1 method and the RK2 scheme seem to present an opposite behaviour, improving its accuracy when factor fr grows. Nonetheless, this reduction is due to a smaller mean time-step $\overline{\Delta t}$ of the simulations. In Table 3.3 the mean time-step $\overline{\Delta t}$ obtained for each case is shown. Since fE1 and RK2 methods only employ the information of the last iteration, they are not affected by the dynamic time-step variation. Nevertheless, it is worth to note that although there is a reduction in the accuracy of the linear multi-step methods when a variable time-step is employed, if the factor fr is small enough, the accuracy of method AB2 continues outperforming the accuracy of the fE1 scheme.

3.2.4 Continuous phase

The fluid-dynamic behaviour of the continuous phase, solved in the Eulerian mesh, is described by the Navier-Stokes transport equations. As previously detailed in Section 2.2.2, the fluid-dynamic behaviour of the continuous phase in low-Mach flows is governed by the continuity and momentum equations, which are:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = S_m \tag{3.68}$$

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u}\mathbf{u}) = -\nabla p + \nabla \cdot (\boldsymbol{\tau}) + \rho \mathbf{g} + S_{\mathbf{u}} \tag{3.69}$$

Notice that these equations are identical to the ones defined in Section 2.2.2 except for the two source terms S_m and $S_{\mathbf{u}}$ appearing at the right-hand side of the equations. These source terms are included in order to take into account the influence of the dispersed phase over the continuous phase when two-way coupling is considered. Further details about this aspect will be given in Section 3.2.5. Moreover, when working with mixtures including multiple substances and flows with variable thermophysical properties it is required to solve a transport equation for the energy as well as a transport equation for each one of the species present in the mixture. Before presenting the conservation equations governing for energy and species, let's define a general transport equation for a scalar ϕ including a source term S_ϕ in conservative form:

$$\frac{\partial \rho \phi}{\partial t} + \nabla \cdot (\rho \mathbf{u} \phi) = \nabla \cdot (\rho \Gamma (\nabla \phi)) + S_\phi \tag{3.70}$$

As detailed in Section 2.2.6, the algorithm employed to integrate the low-Mach number transport equations requires that the scalars must be solved in its non-conservative form. Hence, if Eq. (3.70) is expressed in non-conservative form:

$$\rho \frac{\partial \phi}{\partial t} + \rho \mathbf{u} \cdot \nabla (\phi) + \phi \left(\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) \right) = \nabla \cdot (\rho \Gamma (\nabla \phi)) + S_\phi \tag{3.71}$$

where the expression between parentheses in the third term of the left-hand side is the continuity equation. Usually, this term is null since the mass is conserved, but in the present case there may exist an addition or subtraction of mass due to evaporation or condensation of the dispersed phase. As can be seen, the parenthesized expression is equal to the mass source term of Eq. (3.68). Hence, rearranging the previous expression:

$$\rho \frac{\partial \phi}{\partial t} + \rho \mathbf{u} \cdot \nabla (\phi) = \nabla \cdot (\rho \Gamma (\nabla \phi)) + S_\phi - \phi S_m \tag{3.72}$$

Since in the numerical code TermoFluids [41] the convective term is calculated in conservative form, it is useful to redefine the second term in the left-hand side of the previous equation, leading to:

$$\rho\frac{\partial\phi}{\partial t} + \nabla\cdot(\rho\mathbf{u}\phi) - \phi\nabla\cdot(\rho\mathbf{u}) = \nabla\cdot(\rho\Gamma(\nabla\phi)) + S_\phi - \phi S_m \tag{3.73}$$

Once a general expression for the transport equation of a scalar ϕ has been provided, transport equations for the energy and the mass fraction of a specie k can be written as follows:

$$\rho\frac{\partial h}{\partial t} + \nabla\cdot(\rho\mathbf{u}h) - h\nabla\cdot(\rho\mathbf{u}) = \frac{Dp}{Dt} - \nabla\cdot\dot{\mathbf{q}} + \boldsymbol{\tau}\nabla\cdot\mathbf{u} + \dot{\mathcal{Q}} + S_e - hS_m \tag{3.74}$$

$$\rho\frac{\partial Y_k}{\partial t} + \nabla\cdot(\rho\mathbf{u}Y_k) - Y_k\nabla\cdot(\rho\mathbf{u}) = \nabla\cdot(\rho\mathbf{V_k}Y_k) + \dot{w}_k + S_{Y_k} - Y_kS_m \tag{3.75}$$

where the total enthalpy ($h = h_s + \Delta h_f^0$) in the energy equation is being solved (see Eq. (2.19)).

Depending on the flow conditions, two different time-integration strategies are followed:

i) when there exists an exchange of mass and energy between the dispersed phase and the continuous phase, i.e., when droplet heating and evaporation or condensation are being considered, or when dealing with variable density cases, the low-Mach flow approximation is employed, and the time-integration strategy detailed in Section 2.2.6 is followed, using a Predictor-Corrector scheme.

ii) when there is no exchange of mass and energy between both phases, and the incompressibility constraint can be considered, the Navier-Stokes equations are simplified to:

$$\nabla\cdot\mathbf{u} = 0 \tag{3.76}$$

$$\rho\frac{\partial\mathbf{u}}{\partial t} + \nabla\cdot(\rho\mathbf{u}\mathbf{u}) = -\nabla p + \nabla\cdot(\boldsymbol{\tau}) + \rho\mathbf{g} + S_{\mathbf{u}} \tag{3.77}$$

and the momentum equation is integrated employing a one-step explicit second-order linear multi-step method. Further details of this time-integration strategy will be given in Section 5.2.2 and can be found in the work of Trias and Lehmkuhl [42].

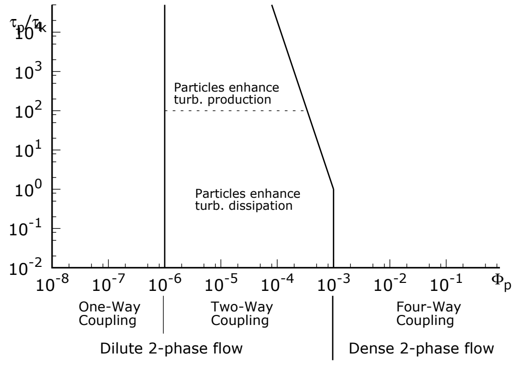


Figure 3.18: Particle coupling scheme by [44].

3.2.5 Coupling

One key aspect for multi-phase flow simulations is the coupling between phases. Based on the available data from experiments and Direct Numerical Simulations, Elghobashi presented a classification map for the coupling between the dispersed and the continuous phases in particle-laden turbulent flows [43, 44]. As depicted in Fig. 3.18, the coupling between the phases depends on the volumetric fraction (Φ_p) and the ratio between the particle response time and the Kolmogorov time scale (τ_p/τ_k). The volumetric fraction is defined as the ratio between the volume occupied by the dispersed phase V_p in a certain volume V and this volume V:

$$\Phi_p = \frac{V_p}{V} \tag{3.78}$$

Analogously, a volume fraction for the continuous phase, sometimes referred as the *void fraction*, can be defined as:

$$\Phi_c = \frac{V_c}{V} \tag{3.79}$$

where V_c is the volume occupied by the continuous phase. In Fig. 3.19 a schematic representation of a certain volume V including dispersed phase elements is depicted. Obviously, the sum of both fractions must be equal to unity:

$$V_p + V_c = 1 \tag{3.80}$$

If the volumetric fraction is smaller than $\Phi_p \leq 10^{-3}$ the two-phase flow is considered **dilute**, otherwise the flow is considered **dense**. In the dilute regime, for sufficiently small volumetric fractions, below $\Phi_p \leq 10^{-6}$, the dispersed phase have

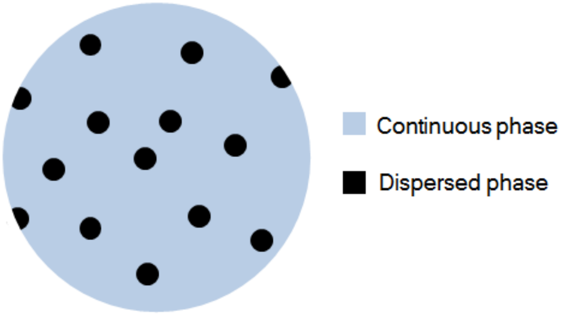


Figure 3.19: Schematic representation of a volume V containing dispersed phase elements.

a negligible influence over the continuous phase, and only the effect of the flow over the particles/droplets is considered. This regime is named **one-way coupling**. For bigger volumetric fractions, the effect of the dispersed phase over the carrier phase cannot be neglected and should be taken into account. Hence, for volumetric fractions ranging between $10^{-6} < \Phi_p \leq 10^{-3}$ the regime is known as the **two-way coupling**. When dealing with dense two-phase flow, with volumetric fractions of $\Phi_p > 10^{-3}$, the interaction between the particles becomes also relevant and should be considered. This regime is denominated **four-way coupling**.

In the two-way coupling regime, Elghobashi found that the type of influence of the dispersed phase over the flow can be of two types, depending on the ratio between the temporal scales of both phases. For a fixed volumetric fraction Φ_p and the same particle material and flow viscosity, if the ratio (τ_p/τ_k) decreases, this means that particles are smaller $d_p \downarrow$, increasing the surface area of contact between the phases and therefore enhancing the dissipation rate of turbulent energy. On the other hand, when increasing the ratio (τ_p/τ_k) Elghobashi found that for $Re_p \geq 400$ vortex shedding takes place, increasing the production of turbulent energy. The inclination of the line dividing the two-way and four-way coupling accounts for the fact that collision rate between particles is enhanced when increasing the ratio $(\tau_p/\tau_k) \uparrow$, converting the two-way coupling in four-way coupling regime.

Source terms

In the one-way coupling regime the influence of the carrier phase over the dispersed phase is accounted using the equations of mass, momentum and energy presented in Sections 3.2.1 and 3.2.2. For the two-way coupling regime, the influence of the dispersed phase over the continuous phase should be taken into account. This is done by means of the source terms presented in Section 3.2.4 and further detailed below in Eqs. (3.82) to (3.85). Particles influence the carrier phase adding its effect to source terms in the control volume where the particle is located at each iteration. As explained in Section 3.2.3, the time-step employed for the particles can be smaller than the time-step used to integrate the continuous phase, therefore, a particle can be integrated several times along the same time-step of the continuous phase Δt_f. Therefore, the influence of the particle over the continuous phase at each particle time-step Δt_p is weighted by a time-step ratio $\Phi_{\Delta t}$, defined as:

$$\Phi_{\Delta t} = \frac{\Delta t_p}{\Delta t_f} \tag{3.81}$$

The source term in species mass fraction equation will be the sum of all the evaporated mass of substance i in the control volume v:

$$S_{Y_i} = -\sum_{v=1}^{N_v} \frac{dm_{p,i}}{dt} \frac{\Phi_{\Delta t}}{V_v} \tag{3.82}$$

where V_v is the volume of the cell v where particle is located. The source term for the continuity equation will be the sum of source terms for all the species of the mixture:

$$S_m = \sum_{i=1}^{N_{spe}} S_{Y_i} \tag{3.83}$$

Regarding the source term of the momentum equation, the influence of the forces acting over the particle should be accounted, except the body forces like gravity, buoyancy or Coulomb forces. Hence, when dealing with heavy particles or droplets ($^{\rho_c}/_{\rho_p} \ll 1$), basically the drag and lift forces are considered. Besides, for evaporating droplets it is required to include the effect of the evaporated mass on momentum equation:

$$S_{\mathbf{u}} = -\sum_{v=1}^{N_v} \left(m_p \sum_i \mathbf{f_i} + \dot{m}_p \mathbf{u} \right) \frac{\Phi_{\Delta t}}{V_v} \tag{3.84}$$

The energy source term can also be split in two parts: one due to the convective heat transfer and a second one due to the mass evaporation. Since in Eq. (3.74) the total enthalpy ($h = h_s + \Delta h_f^0$) is being solved, this second term should account for sensible and formation enthalpy of the evaporated species:

$$S_e = -\sum_{v=1}^{N_v} \left(m_p c_{p,l} conv_{ht} + \dot{m}_p h \right) \frac{\Phi_{\Delta t}}{V_v} \tag{3.85}$$

where the term $conv_{ht}$ accounts for the convective heat transfer and is the first term at right-hand side of Eq. (3.38), it is:

$$conv_{ht} = G\frac{6Nu\lambda_g}{c_{p,l}\rho_p d_p^2}\left(T_g - T_p\right) \tag{3.86}$$

With the aim of better understand the previous expression, if the second term in the right-hand side of Eq. (3.38) is dropped, the following expression is obtained:

$$\frac{dT_p}{dt} = G\frac{6Nu\lambda_g}{c_{p,l}\rho_p d_p^2}\left(T_g - T_p\right) = conv_{ht} \tag{3.87}$$

Rearranging the previous expression, using the classical Nusselt definition $Nu = \alpha_c d_p/\lambda_g$ and the surface of a spherical particle $A_p = \pi d_p^2$:

$$\begin{aligned} m_p c_{p,l}\frac{dT_p}{dt} &= \rho_p \pi \frac{d_p^3}{6} c_{p,l} G\frac{6Nu\lambda_g}{c_{p,l}\rho_p d_p^2}\left(T_g - T_p\right) & (3.88)\\ m_p c_{p,l}\frac{dT_p}{dt} &= \pi d_p^2 G\alpha_c\left(T_g - T_p\right) & (3.89)\\ m_p c_{p,l}\frac{dT_p}{dt} &= \left(G\alpha_c\right)\left(T_g - T_p\right)A_p = m_p c_{p,l} conv_{ht} & (3.90)\end{aligned}$$

where the relation between the employed expression and the classical Newton's law for convective heat transfer is readily seen.

3.3 Interpolation

When the Lagrangian-Eulerian approach is adopted to model dispersed multi-phase flows, some fields of the continuous phase must be known at particles' location. Since the fields of the carrier phase are stored at the nodes of the Eulerian mesh, these scalar and vectorial fields must be interpolated in the position of the particles/droplets. Therefore, one crucial aspect in the Lagrangian-Eulerian model for multi-phase dispersed flows is the interpolation method. When structured meshes are employed, several interpolation methods are available: linear interpolation, lagrange polynomials, cubic splines, etc. In general, these methods allow to achieve high-order accuracy if it is desired. However, the interpolation methods that exploit the advantages of regular grids can not be used when dealing with unstructured meshes, and is difficult to achieve high-order accuracy. As stated by Bianco et al. [45], the accuracy of the interpolation method is a key aspect, since interpolation errors can alter the particle trajectory as well as modify the droplet heating and evaporation behaviourr. In the same way, Michalek et al. [46] reports that is known that second-order accuracy interpolation methods can act as additional filtering in LES simulations, losing small-scales information. Moreover, as shown by Miller and Bellan [47], the interpolation can represent an important part of the computational effort devoted to the

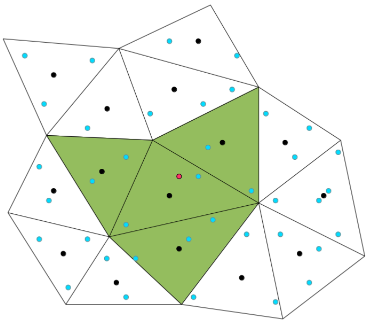


Figure 3.20: Face stencil.

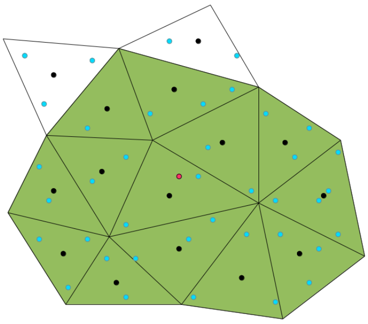


Figure 3.21: Vertex stencil.

dispersed phase. Hence, clearly there are two relevant aspects to consider in the interpolation method employed for dispersed multi-phase flows simulations: accuracy and computational cost. Since the CFD code TermoFluids where the current Lagrangian-Eulerian model has been implemented is aimed to unstructured meshes, it is necessary to implement interpolation methods for scattered data in multi-dimensions. Following, a detailed study of various interpolation methods for scattered data, well-suited for unstructured meshes is presented. The focus of the study is placed in the accuracy and the computational cost of the studied methods. The interpolation methods considered in this study are:

- 1^{st} order Taylor series.
- The Shepard method [48].
- The Radial Basis Function (RBF) method using the compact support radial function Wendland C2 [48, 49].
- The pseudo-Laplacian method introduced by Holmes and Connell [50].

The interpolation method based on Taylor series requires the gradient of the interpolated scalar field. Hence, different methods to calculate the gradient are tested. The analysed methods are based on the Green-Gauss theorem and the least-squares method [51]. The other interpolation methods (Shepard, RBF and Pseudo-Laplacian), as well as the least-squares method, require a cloud of data points surrounding the position where the scalar field must be interpolated. In order to obtain this data points, the strategy adopted is to select the node of the cell where the particle is living plus the nodes of some neighbour cells surrounding it. Two different stencils have been considered: a first one constructed with the cells connected by faces and a second one with cells connected by vertices (see Figs. 3.20 and 3.21 respectively).

Four methods for gradient evaluation have been con considered. One method is the Green-Gauss theorem (gg), which evaluates the gradient of scalar field ϕ in control volume v as:

$$\nabla \phi_v \approx \frac{1}{V_v} \int_{V_v} \nabla \phi_v dV_v = \frac{1}{V_v} \int_{S_v} \phi_v dS_v \approx \frac{1}{V_v} \sum_f \phi_f S_f \tag{3.91}$$

The second considered method is also based on the Green-Gauss theorem although slightly modified (ggs), and it is calculated as:

$$\nabla \phi_v \approx \sum_f \left[\phi_f S_f \frac{1}{0.5\,(V_v + V_{nb})} \right] \tag{3.92}$$

The last two interpolation methods are based on the least-squares method: one employing face connectivity (lsf) and the other one using a bigger stencil formed by cells connected by vertices (lsv).

The study has been conducted for structured (hexahedral) and unstructured (tetrahedral) meshes. The domain of study is a cube of unitary length. Four meshes with different control volume sizes have been generated for each type of mesh. The accuracy analysis has been performed injecting 1×10^5 particles at random locations in the interval $x_i = [0.2, 0.8]$, trying to avoid boundary effects. A scalar field $\mathbf{\Phi}$ is filled at each mesh node with the following function:

$$f(x,y,z) = x^2(x-0.1)^{0.2}y^2(y-0.1)^{0.2}z^2(z-0.1)^{0.2} \tag{3.93}$$

Then, the value of this scalar field is interpolated at the random particles injected in the computational domain, and the root-mean-square error (RMSE) is calculated comparing the numerical value interpolated at particles and the analytical value obtained from Eq. (3.93) as:

$$error_{rms}(x,y,z) = \sqrt{\frac{1}{N} \sum_p^N \left|(f_{num}(x,y,z) - f_{analytical}(x,y,z)\right|^2} \tag{3.94}$$

Moreover, the obtained results are fitted in a function with the form $f(x) = ax^e$, where the exponent e will be the accuracy order of the interpolation method. The obtained results are depicted in Figs. 3.22 to 3.25. The mesh size in the abscissa axis is defined for both type of meshes as $\Delta x = \sqrt[3]{\overline{Vol_{CV}}}$, where $\overline{Vol_{CV}}$ is the mean volume of all the control volumes of the computational mesh. The RMSE of the analysed interpolation methods are shown in Figs. 3.22 and 3.23 for structured and unstructured meshes, respectively. In these figures, the Taylor method shown is the

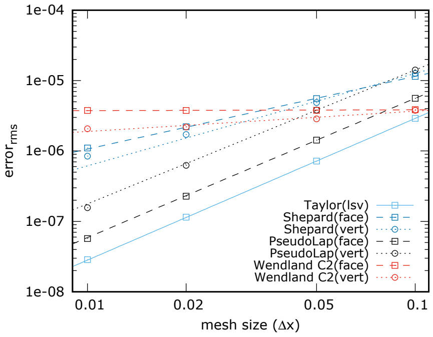


Figure 3.22: RMSE of analysed interpolation methods in structured (hexahedral) meshes.

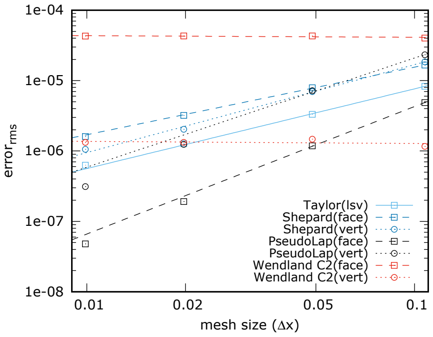


Figure 3.23: RMSE of analysed interpolation methods in unstructured (tetrahedral) meshes.

one with the gradient calculated employing the least-squares method with vertex connectivity. In Figs. 3.24 and 3.25 the different gradient calculation methods for the 1^{st} order Taylor interpolation method are compared. The points stand for the RMSE calculated using Eq. (3.94) while the lines show the fitted functions obtained by regression (note that the lines are straight due to the axis are in logarithmic scale). The coefficients obtained after the fitting procedure for function $f(x) = ax^e$ are shown in Table 3.4 for structured meshes and in Table 3.5 for the unstructured case. As expected, the accuracy order of the methods when working in structured meshes is bigger than under unstructured meshes. For the Taylor method, in structured meshes all the gradient calculation methods allow second order accuracy, while in unstructured meshes the accuracy order is more dependant on gradient evaluation method. For both cases, the least-squares method with vertex connectivity is the one with better accuracy. Regarding the Shepard method, this study shows that it is first order accurate. As one might expect, the accuracy of the method increases when more data points are employed (vertex stencil). On the other hand, the pseudo-Laplacian method shows second order accuracy in both structured and unstructured meshes, although in the structured case is better. Surprisingly, in this method the stencil of cells with face connectivity shows a better accuracy than the one formed by cells connected by vertices. Another major conclusion that could be derived from this study, is that RBF methods are not well suited for this application, at least using this local stencils and employing the compact support radial function Wendland C2, since this study shows a zero order accuracy for the method under this framework.

Moreover, as mentioned above, there is another key aspect of the interpolation method besides accuracy, its computational cost. Hence, a second study analysing

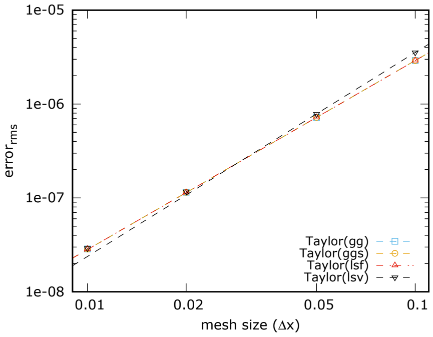


Figure 3.24: RMSE of 1^{st} order Taylor method using different gradiend evaluation methods in structured (hexahedral) meshes.

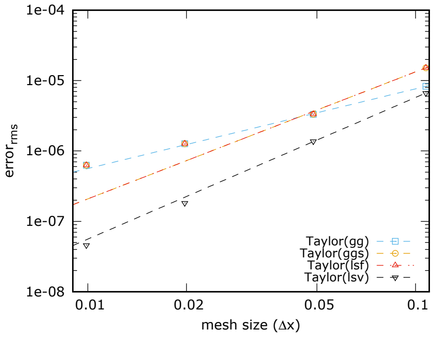


Figure 3.25: RMSE of 1^{st} order Taylor method using different gradiend evaluation methods in unstructured (tetrahedral) meshes.

Table 3.4: Fitted values for expression $f(x) = ax^e$ in structured meshes.

Interp. Method	a	e
Taylor (gg)	$2.991\,87 \times 10^{-4}$	$2.012\,65$
Taylor (ggs)	$2.991\,87 \times 10^{-4}$	$2.012\,65$
Taylor (lsf)	$2.991\,87 \times 10^{-4}$	$2.012\,65$
Taylor (lsv)	$5.258\,79 \times 10^{-4}$	$2.171\,81$
Shepard(face)	$1.235\,54 \times 10^{-4}$	$1.032\,33$
Shepard(vertex)	$2.597\,06 \times 10^{-4}$	$1.312\,75$
PseudoLap(face)	$5.321\,75 \times 10^{-4}$	$1.977\,56$
PseudoLap(vertex)	$1.128\,37 \times 10^{-3}$	$1.901\,40$
Wendland C2(face)	$3.938\,59 \times 10^{-6}$	$1.076\,60 \times 10^{-2}$
Wendland C2(vertex)	$7.152\,73 \times 10^{-6}$	$2.886\,79 \times 10^{-1}$

Table 3.5: Fitted values for expression $f(x) = ax^e$ in unstructured meshes.

Interp. Method	a	e
Taylor (gg)	$1.033\,65 \times 10^{-4}$	$1.131\,63$
Taylor (ggs)	$8.474\,10 \times 10^{-4}$	$1.804\,09$
Taylor (lsf)	$8.474\,10 \times 10^{-4}$	$1.804\,09$
Taylor (lsv)	$5.907\,52 \times 10^{-4}$	$2.011\,24$
Shepard(face)	$1.417\,31 \times 10^{-4}$	$9.602\,97 \times 10^{-1}$
Shepard(vertex)	$2.989\,52 \times 10^{-4}$	$1.248\,83$
PseudoLap(face)	$2.769\,93 \times 10^{-4}$	$1.810\,60$
PseudoLap(vertex)	$7.416\,25 \times 10^{-4}$	$1.549\,91$
Wendland C2(face)	$3.904\,54 \times 10^{-5}$	$-2.307\,94 \times 10^{-2}$
Wendland C2(vertex)	$1.192\,58 \times 10^{-6}$	$-3.007\,15 \times 10^{-2}$

the computational cost of each presented method has been carried out. For that purpose, several simulations varying the number of Lagrangian particles present in the computational domain are performed. Since the 1^{st} order Taylor interpolation method employs the gradient of the interpolated scalar field, the analysis is done for the different gradient calculation methods employed in the accuracy analysis. Moreover, the cost of Taylor method is *split* in two parts: the computational cost of the gradient calculation and the CPU time spent in the interpolation process itself for all the Lagrangian particles. This is done because the gradient of the interpolated scalar field can be calculated previously in the simulation for other purposes (for example, the gradient of the velocity when some turbulent models are employed), and then the computational cost of this part can be considered *free*, since the gradient is already computed. The study has been done for structured and unstructured meshes. Both meshes have a similar control volume characteristic size ($\Delta x = 0.05$), with 8000 CV for the structured mesh and 8630 CV for the unstructured one. The obtained results for both structured and unstructured meshes respectively are depicted in Figs. 3.26 and 3.27. On that figures, the computational cost of the gradient for the Taylor method has not been taken into account. The interpolation method with the smallest cost is the Taylor method, followed by the Shepard method, then the pseudo-Laplacian method, and the RBF method is the one with the highest computational cost. Obviously, for the same method, when a larger stencil (vertex connectivity) is employed, the method increases its computation time. As can be seen, all the methods present a linear behaviour, increasing its computational cost as the number of particles increases. In order to highlight this linear behaviour, the CPU time and the number of particles have been normalized in Figs. 3.28 and 3.29. Although all the methods increase linearly for both type of meshes, can be appreciated a slight (and different) deviation from slope one for the studied methods. These small deviation

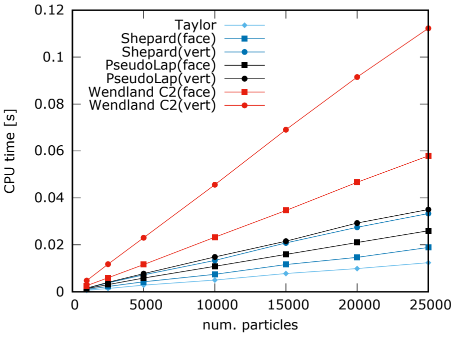


Figure 3.26: Computational time for the different interpolation methods in structured mesh.

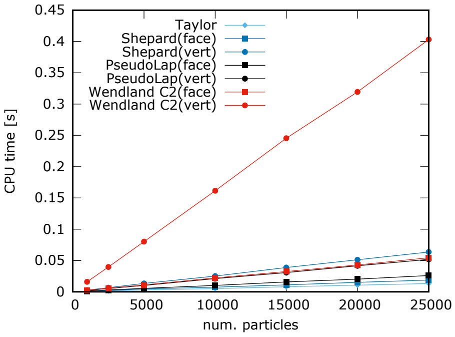


Figure 3.27: Computational time for the different interpolation methods in unstructured mesh.

in the slope, which tends to be smaller than one, can be associated to an increase in the efficiency of cache memory access. Regarding the Taylor interpolation method, in Figs. 3.30 and 3.31 it is shown the computational cost of the gradient evaluation and the cost of the interpolation process. As can be seen, the cost of gradient evaluation is independent from the number of particles, and only depends on the method employed to calculate it and the number of cells of the computational mesh. The results presented are only an illustrative example to show the behaviour of the computational cost associated to this method, since the weight in the computational effort of each part will depend on the size of the mesh, the number of particles and the number of scalar fields to interpolate. In general, leaving aside accuracy considerations, since the cost of the interpolation process for the Taylor method is the smallest one of all presented methods, as depicted in Figs. 3.26 and 3.27, the Taylor method can be the computationally most efficient in two cases: if the gradient evaluation can be considered *free*, or when the mesh is small enough and there is a large amount of particles in the computational domain.

3.4 Particle sub-grid dispersion modeling

Under the framework of LES modelling, the large scales of the flow are well-resolved, while the sub-grid scales (sgs) are modelled. The velocity of the carrier phase $\mathbf{u}$ appearing in different forces like: drag force (Eq. (3.4)), added mass force (Eq. (3.17)), basset force (Eq. (3.20)), etc., is the velocity of the carrier phase at particle's position. This velocity can be decomposed as $\mathbf{u} = \widetilde{\mathbf{u}} + \mathbf{u}_{sgs}$, where $\widetilde{\mathbf{u}}$ is the resolved velocity field and $\mathbf{u}_{sgs}$ represents the sgs velocity contribution *lost* due to

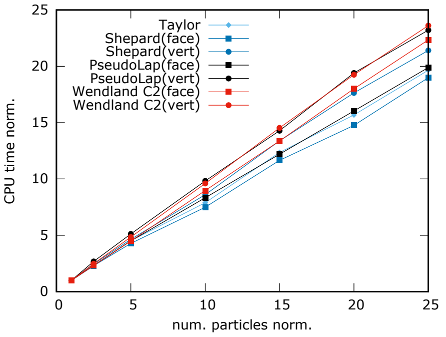


Figure 3.28: Normalized computational time of interpolation methods in structured mesh.

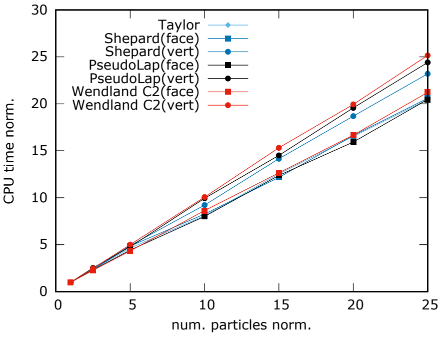


Figure 3.29: Normalized computational time of interpolation methods in unstructured mesh.

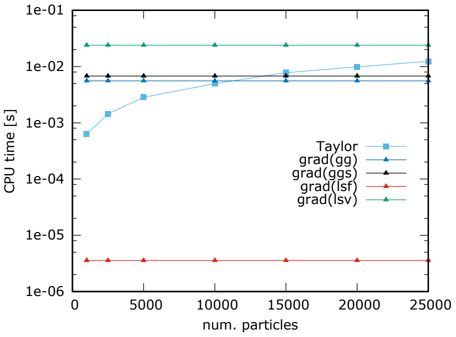


Figure 3.30: Computational time of interpolation and gradient calculation for Taylor method in structured mesh.

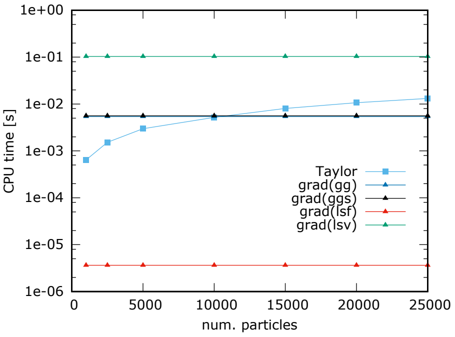


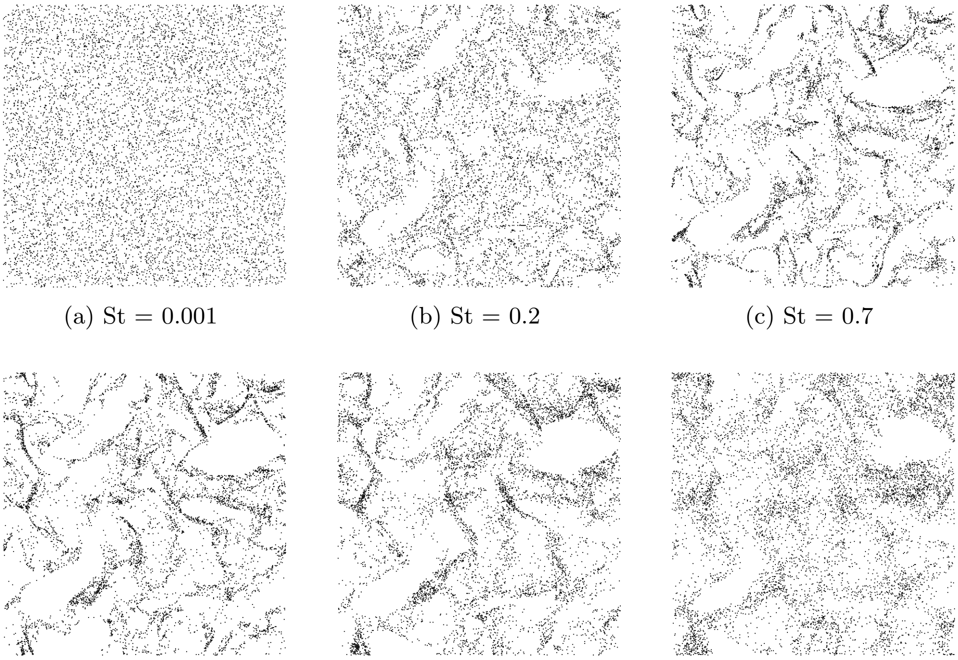
Figure 3.31: Computational time of interpolation and gradient calculation for Taylor method in unstructured mesh.

the filtering procedure applied in LES modelling.

A key aspect in the development of numerical methods for dispersed multiphase flows is to asses and study the influence and importance of the contribution of the sub-grid scales on the dispersed phase. So far, in many studies and simulations the effect of the sub-grid velocity over the particles has been directly neglected (i.e., $\mathbf{u} = \widetilde{\mathbf{u}}$) [38, 52]. This option is reasonable if there is a low residual energy content in the key regions of the computational domain. Otherwise, the contribution of the sub-grid scale effects over the Lagrangian particles should be taken into account [53]. In order to include the influence of the sub-grid scales over the particle's behaviour there are two main approaches: deterministic models using Approximate Deconvolution Models (ADM) [54, 55] and Stochastic Models [56, 57]. The ADM has proven to be favourable correcting the resolved eddies near cut-off scale, but cannot be used to recover the ones below the cut-off scale [58, 59]. For example, Marchioli et al. [60] performed a study in a channel flow at $Re_\tau = 150$ in LES using fine and coarse meshes. The obtained results show that for fine LES, the ADM helps to improve the results compared to a LES simulation without model. On the other hand, when ADM is used in a coarse mesh, there is an overestimation of particle velocity fluctuations, obtaining results that are worst than without sub-grid dispersion model. An analytical comparison between an ADM model and two stochastic models was performed by Gobert [61]. According to the analysis, stochastic models (using optimal parameters) can predict well first and second moments for particles with small Stokes number, while ADM is good predicting first moments for high Stokes number, but can not retrieve second moments for any Stokes number value, and conclude that the stochastic models seem a better choice than ADM, since stochastic models can predict first and second moments correctly.

3.4.1 Stochastic models analysis

In the present subsection, two stochastic models for sub-grid dispersion of Lagrangian particles are analysed and studied. As a first step, DNS simulations have been performed in order to obtain reference results. These DNS results are then compared against the ones obtained employing the studied stochastic models, allowing to assess their performance. Hence, aiming to study the behaviour of the dispersed phase in turbulent cases, DNS simulations of particle-laden flows with solid particles and gaseous carrier phase have been carried out. The simulations are forced isotropic turbulence tests at $Re_\lambda = 40$ with periodic boundary conditions. The turbulent flow field has been generated employing the linear forcing technique proposed by Lundgren [62]. Particles with different weight (different Stokes number) and initial velocity equal to the gas velocity at injection position are randomly injected inside the domain. The computational domain consists in a cube of length 2π, periodic in the three directions and discretized in 96^3 cells. The simulations have been computed in parallel using 96 CPUs, and for each simulation the calculated particle field was formed by 64^3 dis-



(a) St = 0.001

(b) St = 0.2

(c) St = 0.7

(d) St = 1

(e) St = 2

(f) St = 4

Figure 3.32: Snapshots of particle positions in DNS simulations.

crete particles. The particle fields are assumed diluted, therefore, one-way coupling is considered and particle collisions are neglected. The pressure-velocity coupling is solved by means of the Fractional Step Method (FSM). The Poisson equation is solved employing the FFT-based Poisson Solver presented by Borrell et al. [63].

Some results obtained from the DNS simulations are shown in Fig. 3.32, where snapshots of particle locations in a 2D slice located at $x = \pi$ for different Stokes numbers are depicted. It is well-known that the instantaneous structures of a turbulent flow influence the motion of the particles depending on their inertia [64]. Particle inertia is characterized using the Stokes number, which in the present study is defined as the particle relaxation time normalized by the Kolmogorov time scale $St = \frac{\tau_p}{\tau_K}$. Some particles tend to correlate with certain eddy structures leading to the effect of preferential concentration. This effect can be clearly appreciated in Fig. 3.32, where is shown how particles with a Stokes number close to unity tend to follow the eddy structures of the continuous phase. Specifically, particles tend to accumulate in flow regions of low vorticity and high rate of strain. These results are in agreement with previous observations found in the literature [65].

The stochastic models implemented in the current study are the model proposed by Bini and Jones (BJ) [66, 67] and the model developed by Pozorski and Apte (PA) [65]. Both stochastic models are similar in 'flavour', but differ in the mathematical derivation and working-way. Following, the studied models are briefly presented. As previously explained, in LES modelling only the resolved velocity field is obtained $\widetilde{\mathbf{u}}$, but the fully resolved velocity is required to properly simulate the Lagrangian particles displacement. Hence, substituting $\mathbf{u} = \widetilde{\mathbf{u}} + \mathbf{u}'$ in Eq. (3.33):

$$\frac{d\mathbf{v}}{dt} = \frac{\widetilde{\mathbf{u}} - \mathbf{v}}{\tau_p} + \frac{\mathbf{u}'}{\tau_p} \tag{3.95}$$

where the sub-grid velocity $\mathbf{u}'$ is not available in LES simulations and must be modelled. The model of Pozorski and Apte assumes that the sub-grid velocity seen by the particle is governed by the following Langevin type equation:

$$du_i' = -\frac{u_i'}{\tau_l} dt + b dW_i \tag{3.96}$$

where τ_l is the time scale of residual motions seen by the particle and dW_i is an increment of a Wiener process [68]. The diffusion coefficient b of this model is defined as:

$$b = \sqrt{\frac{4}{3} \frac{k_{sgs}}{\tau_l}} \tag{3.97}$$

The time scale of the sub-grid scales, τ_l, accounts for the crossing trajectory effect using the Csanady expressions [69], which define different time scales for the parallel and perpendicular directions of the relative velocity between the particle and the carrier phase $\mathbf{u_{rel}} = \widetilde{\mathbf{u}} - \mathbf{v}$. In the model of Pozorski and Apte these time scales are calculated as:

$$\tau_{l,\parallel} = \frac{\tau_{sgs}}{\sqrt{1+\beta^2\xi^2}} \tag{3.98}$$

$$\tau_{l,\perp} = \frac{\tau_{sgs}}{\sqrt{1+4\beta^2\xi^2}} \tag{3.99}$$

where β is assumed equal to unity $\beta = 1$ and ξ is the normalized drift velocity calculated as:

$$\xi = \frac{|\widetilde{\mathbf{u}} - \mathbf{v}|}{\sigma_{sgs}} \tag{3.100}$$

The term σ_{sgs} stands for the characteristic sub-grid velocity seen by the particle, determined from $\sigma_{sgs} = \sqrt{\frac{2}{3}k_{sgs}}$, and the value τ_{sgs} is defined as:

$$\tau_{sgs} = C_0 \frac{\Delta_f}{\sqrt{\frac{2}{3}k_{sgs}}} \tag{3.101}$$

On the other hand, the model of Bini and Jones, instead of tracking the sub-grid velocity along particle's path, models Eq. (3.95) as:

$$dv_i = \frac{\widetilde{u_i} - v_i}{\tau_p} dt + b dW_i$$

where the Langevin type equation is applied to the particle velocity itself. In this model the diffusion coefficient b follows the expression:

$$b = \sqrt{C_0 \frac{k_{sgs}}{\tau_t}} \tag{3.102}$$

The associated time scale is obtained from:

$$\tau_t = \tau_p \left(\frac{\tau_p \sqrt{k_{sgs}}}{\Delta_f} \right)^{2\alpha - 1} \tag{3.103}$$

and the constant α is set to a value $\alpha = 0.8$ after the study carried out by Bini and Jones [66]. Moreover, both models employ a model constant C_0 in their formulations. The value of this model constant is case-dependant and its correct adjustment is still an open issue. For the current analysis, its value has been set to unity.

Aiming to better understand the presented stochastic models, and before the detailed analysis, some simple 1D tests have been carried out trying to discern the mathematical and numerical behaviour of both models. The first test consists in a particle with a relaxation time equal to one $\tau_p = 1$ and null initial velocity $v = 0$ m/s. The carrier phase velocity seen by the particle is the sinusoidal function defined in Eq. (3.104). The sub-grid kinetic energy is set to $k_{sgs} = 1 \times 10^{-3}$ and the filter width has a value $\Delta_f = 1$. The obtained results are plot in Fig. 3.33. As can be seen, the BJ model is more noisy than the PA model, since the first model does not have *memory*, and just adds a new Wiener process to the particle velocity at each new iteration, uncorrelated with the previous one. Furthermore, it can be noticed that for the same sub-grid values, the model of Bini and Jones tends to estimate that the filtered scales has a higher effect on the velocity particle than the model of Pozorski and Apte.

$$\begin{aligned} &u(t) = u_0 + A \sin(\omega t + \varphi) \ \text{m/s} \\ &u_0 = 1 \ \text{m/s} \\ &A = 0.25 \quad \omega = \pi \quad \varphi = 0. \end{aligned} \tag{3.104}$$

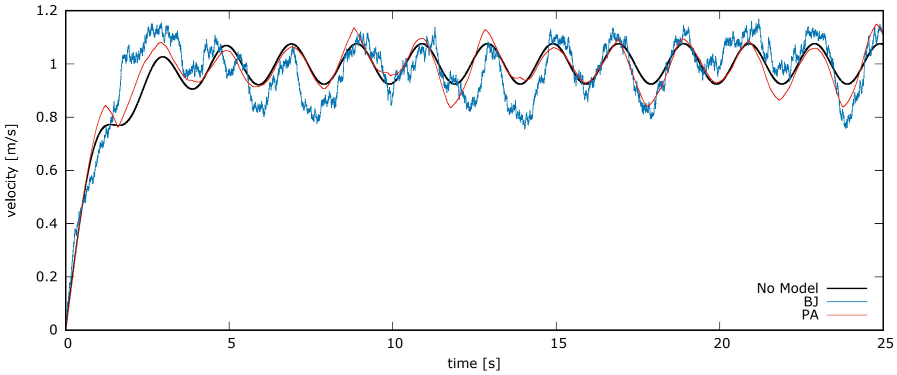


Figure 3.33: Particle velocity in the simple 1D test with sinusoidal velocity for carrier phase.

A second test has been performed considering that the gas velocity seen by the particle is set to a constant value equal to one $u(t) = 1$ m/s. The results obtained under this new conditions are depicted in Fig. 3.34. As can be seen, now the particle velocity for the PA model becomes numerically stalled during large periods of time. This numerical issue happens when the relative velocity between the particle and the carrier phase, $\mathbf{u_{rel}}$, is smaller than the modelled sub-grid velocity $\mathbf{u_{rel}} < \mathbf{u}'$. This numerical issue comes up due to in the model of Pozorski and Apte the sub-grid velocity is tracked along a path following the relative velocity $\mathbf{u_{rel}}$, and if this relative velocity is smaller than the sub-grid velocity $\mathbf{u}'$, the relative velocity changes its sign at each iteration, arising this stalling effect. Since this numerical artifact is completely undesirable, four alternatives paths to track the sub-grid velocity have been implemented and tested:

- The Cartesian axis.
- Along $\mathbf{u}'$ itself.
- Throughout particle velocity $\mathbf{v}$.
- Smart swapping between $\mathbf{u_{rel}}$ and $\mathbf{u}'$.

The results obtained employing these alternative paths are depicted in Fig. 3.35, concluding that all the proposed alternatives are satisfactory avoiding the numerical problem. However, this numerical effect is not the only aspect to consider in order to choose a suitable path to track the sub-grid velocity. Further ahead in this section, within the context of the study and analysis of both presented stochastic models,

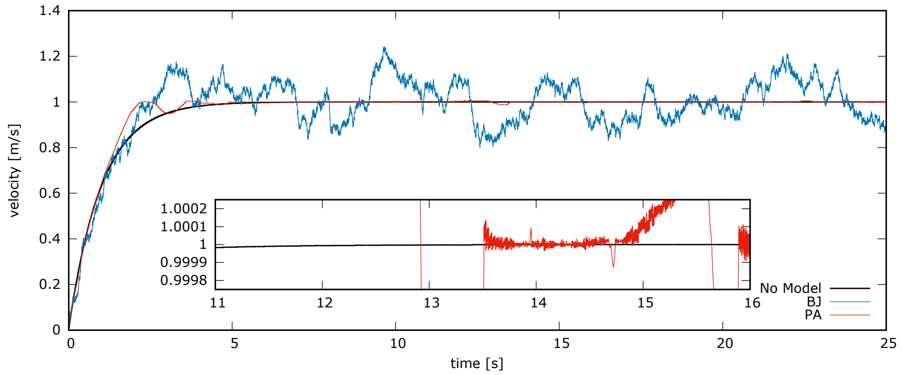


Figure 3.34: Particle velocity in the simple 1D test using lineal velocity for carrier phase.

a study of the four introduced alternatives is presented and the best suited path is selected.

In order to study and analyse the presented sub-grid stochastic models, several simulations where the DNS velocity field has been spatially filtered (FDNS simulations) have been computed. For each simulation, a particle field is solved using the DNS velocity field, while other particle fields are computed using the LES-like velocity field (FDNS) for three different filter sizes: FDNS2 ($\Delta_f = 2^1\Delta_x$), FDNS3 ($\Delta_f = 2^2\Delta_x$) and FDNS4 ($\Delta_f = 2^3\Delta_x$), where Δ_x is the control volume size. The filtered cases are solved for both BJ and PA models, as well as without any sub-grid model (*No Model*). In the current study, two main quantities have been selected to compare the LES results against the reference DNS results: the Preferential Concentration Effect (PCE) and the Kinetic energy. In order to measure the PCE, in the present work it is calculated using the criterion defined by Wang and Maxey [70]:

$$D = \sum_{n=0}^{\infty} [f_d(n) - f_p(n)]^2 \tag{3.105}$$

where $f_d(n)$ is the discrete pdf of the simulated particles distribution and $f_p(n)$ the discrete Poisson (random) distribution. The preferential concentration effect has been measured for different bin sizes Δ_{bin}, where the bin size is defined as $\Delta_{bin} = V_{bin}^{1/3}$ and V_{bin} is the volume employed to calculate the particles number density (see Fig. 3.36, where at the left is represented a discrete Poisson distribution and at the right a particle distribution conditioned by turbulent structures).

Once the methodology and criteria to perform the detailed analysis of the stochastic models have been presented, it is possible to define a specific experiment devoted

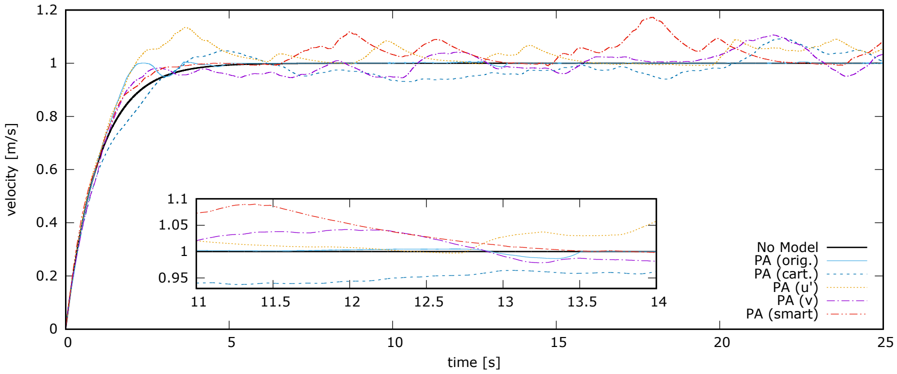


Figure 3.35: Particle velocity using PA model for the different proposed paths tested to track the sub-grid velocity.

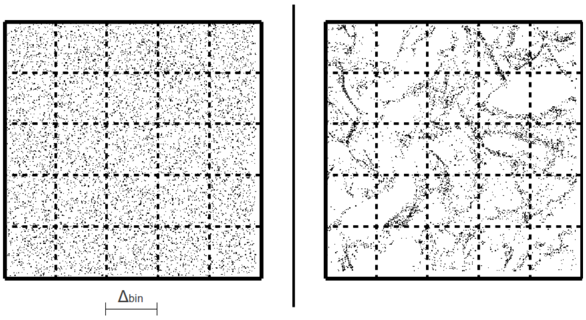
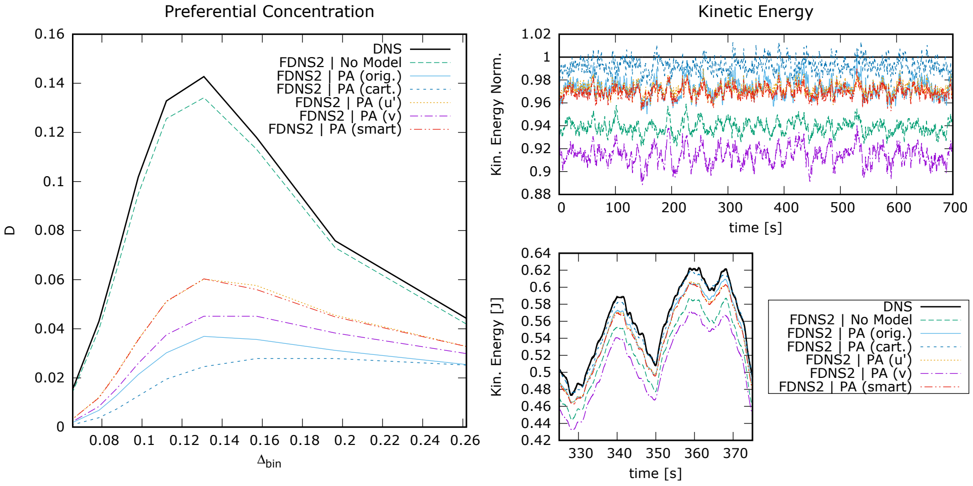


Figure 3.36: Left: Discrete Poisson distribution. Right: Particle distribution conditioned by turbulence.



(a) Preferential Concentration.

(b) Kinetic Energy.

Figure 3.37: Results for the different proposed tracking paths for the sub-grid velocity in the Pozorski and Apte model.

to select the best suited path to track the sub-grid velocity of the Pozorski and Apte model among the alternatives previously presented. In this experiment particles with a Stokes number of $St = 0.7$ are simulated, and the DNS velocity field has been filtered one time (FDNS2). The simulations are carried out for various particle fields employing the sub-grid model of Pozorski and Apte, where each particle field uses one of the proposed paths to track sub-grid velocity. In Fig. 3.37 are depicted the preferential concentration and the kinetic energy of the particles obtained from these simulations.

The results show that the path employing the cartesian axis to track the sub-grid velocity seems to retrieve the kinetic-energy quite well, but it clearly destroys the preferential concentration effect. On the other hand, the method using the particle velocity as path does not work properly, since the kinetic-energy is smaller than in the case without sub-grid model. Therefore, these two alternatives paths are discarded.

The other options, i.e., the original tracking path employing the relative velocity, the one using the sub-grid velocity, and the smart swapping between the relative velocity and the sub-grid velocity, are able to recover similar levels of kinetic energy and, in fact, the smart swapping as well as the sub-grid velocity perform better than the original tracking path regarding the preferential concentration effect. Therefore, the proposed alternative path using smart swapping to track the sub-grid velocity in

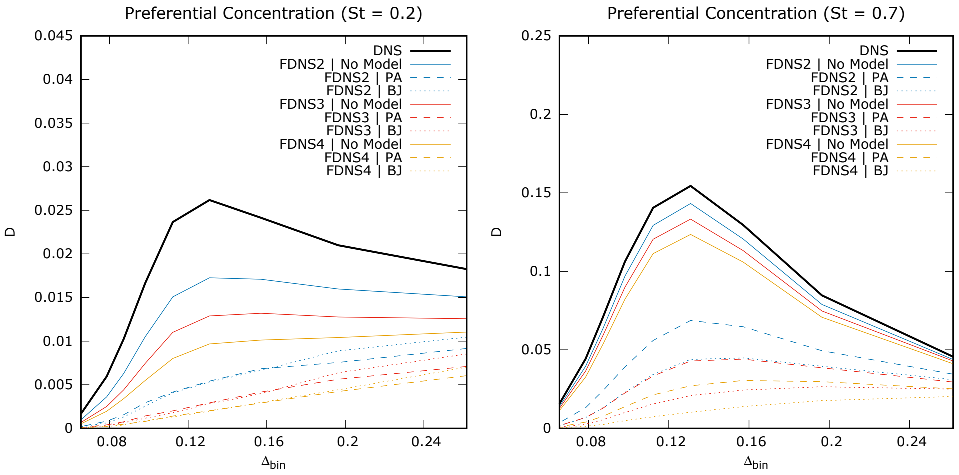


Figure 3.38: Preferential concentration for cases $St < 1$.

the Pozorski and Apate method is selected as the best suited option. Hence, from now on, the presented results of the Pozorski and Apte model are obtained employing this option.

Following, the results of the stochastic models study are shown and analyzed. In Fig. 3.38 the preferential concentration effect for small inertia particles is depicted. As can be seen, for Stokes number lower than unity, the preferential concentration effect is dissipated with filtering. The reason is that small-inertia particles tend to follow all the scales of the flow, including small eddies that are removed in the LES filtering procedure. The stochastic sub-grid models not only do not fix this issue, but also worsen it, since the models tend to introduce a scattering effect. On the other hand, as is depicted in Fig. 3.39, for particles with Stokes number larger than one, the preferential concentration effect seems to be enhanced by the filtering. In this type of particles inertia dominates, and small scales eddies only have a stirring effect on them, randomizing their distribution. Therefore, for this kind of particles the stochastic models help to restore the randomizing effect lost with filtering. However, in the current simulations, the randomizing effect introduced by the models is higher than the stirring effect lost by filtering.

Regarding the kinetic energy of the particles, results in Fig. 3.40 clearly show how the filtering procedure reduce the kinetic energy of the particles. Both sub-grid stochastic models help to recover the kinetic energy level, although the models are not able to retrieve the exact kinetic energy level of the DNS simulation. Better values of kinetic energy can be obtained with a proper fitting of the constant C_0 appearing

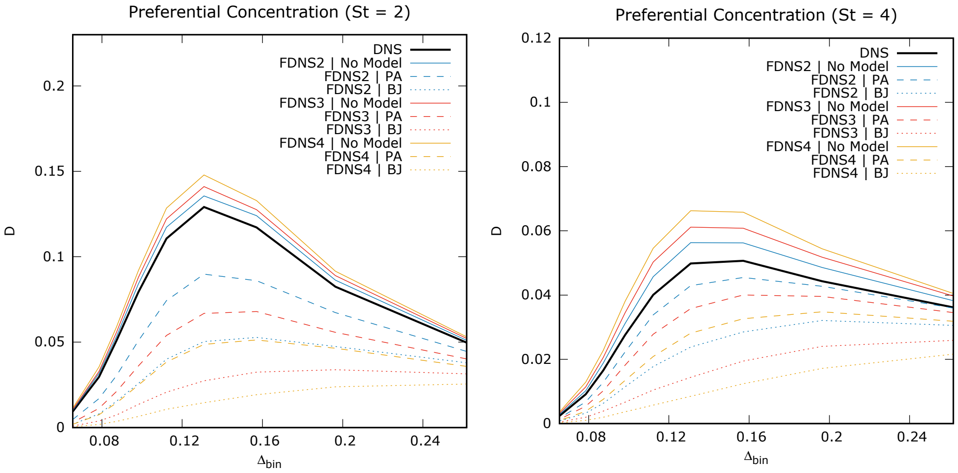


Figure 3.39: Preferential concentration for cases $St > 1$.

in both models. For the present cases, where the value $C_0 = 1$ is employed, the PA model slightly under-predicts the kinetic-energy, while the BJ model over-predicts it. In addition, it is worth noting that the BJ model seems to be quite more dependent on filter size Δ_f than the PA model.

3.4.2 Development of a new statistical model for sub-grid dispersion

Given the shortcomings and deficiencies of the models studied in the Section 3.4.1, the development of a new model for sub-grid dispersion of heavy particles without these downsides is investigated. The objective of the model is the reconstruction of the sub-grid velocity seen by the particles, and obviously, this sub-grid velocity should be recovered and modelled employing values and magnitudes ready-available in LES. In order to do so, the idea is to perform an exhaustive statistical analysis with the aim to obtain statistical information about how the sub-grid velocity seen by the particles is related with values available in LES, like sub-grid kinetic energy k_{sgs}, vorticity $\boldsymbol{\omega}$, strain $\mathcal{S}$, etc., as a function of different parameters like the Stokes number of the particles St or the LES filtering size Δ_f. This statistical analysis is done through DNS simulations of isotropic turbulence where the DNS velocity field is spatially filtered obtaining a LES-like velocity field, where $\mathbf{u}'$ is ready available. The particles present in the simulation save at each time step information of the magnitude and direction of the sub-grid velocity $\mathbf{u}'$ that they see as function of different parameters, like the sub-grid kinetic energy k_{sgs} or the vorticity $\boldsymbol{\omega}$. As example, in Fig. 3.41 the mean and standard deviation of the sub-grid velocity magnitude ($|\mathbf{u}'|$) as a function of the sub-grid kinetic

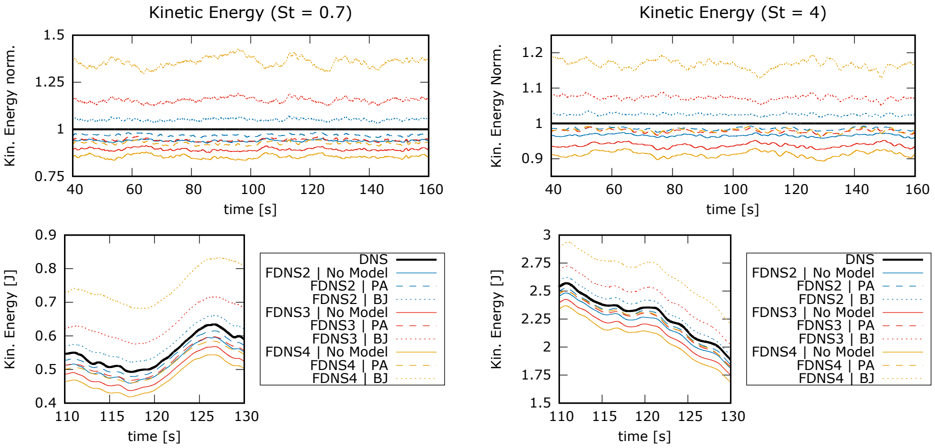


Figure 3.40: Kinetic Energy.

energy k_{sgs} seen for particles with different Stokes number and various filter size Δ_f are depicted. These results are for a case of isotropic turbulence $Re_\lambda = 40$. Moreover, for each saved value of sub-grid kinetic energy k_{sgs}, the discrete pdf of the sub-grid velocity can be recovered, as is shown in Fig. 3.42 for $k_{sgs} = 6.625 \times 10^{-3}$ and $St = 0.5$. This discrete pdf can be represented as a continuous distribution, in this case a $\beta - pdf$. From this analysis arises the methodological concept in which is based the new sub-grid dispersion model developed in this section: the idea is to generate functions fitting the mean and the variance of the sub-grid velocity $(\mathbf{u}')$ obtained from DNS simulations as function of parameters ready-available in LES. Then, these functions will be employed in LES simulations in order to dynamically reconstruct the probability density functions characterizing the statistical information obtained from DNS and recovering the sub-grid velocity seen by the particle.

Since the sub-grid velocity seen by the particle is a vector $\mathbf{u}'$, three values are required to reconstruct it: the sub-grid velocity magnitude $|\mathbf{u}'|$ and two angles defining the direction in the space θ_1, θ_2. These two angles should be referenced to one or two vectors which directions are known in LES. Therefore, a rigorous analysis trying to determine the directions best-suited to characterize statistically the direction of the sub-grid velocity has been performed. In order to do so, the angle between the sub-grid velocity and the followings magnitudes has been studied:

- The particle velocity $\mathbf{v}$.
- The LES resolved velocity $\widetilde{\mathbf{u}}$.

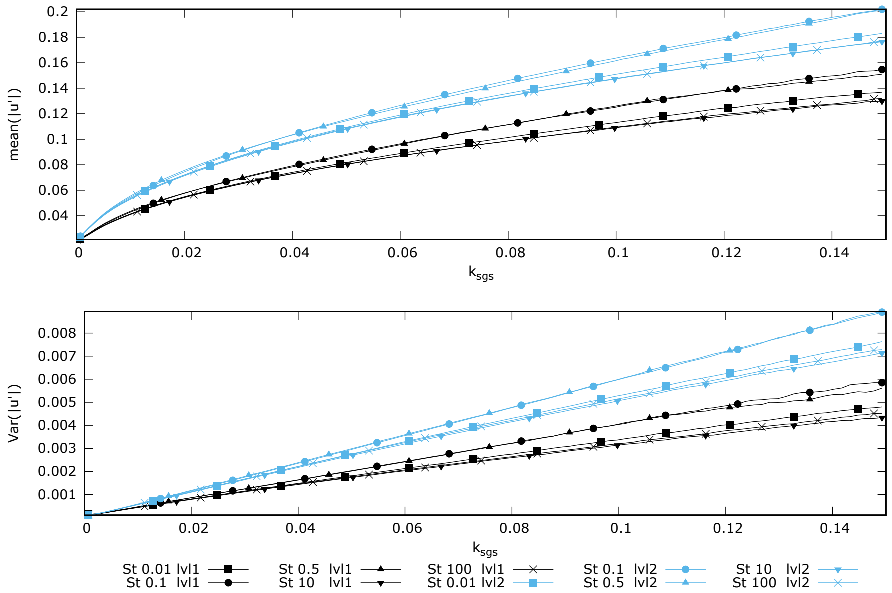


Figure 3.41: Statistical relation between the sub-grid velocity magnitude and the sub-grid kinetic energy seen by the particles in a case of isotropic turbulence with $Re_\lambda = 40$.

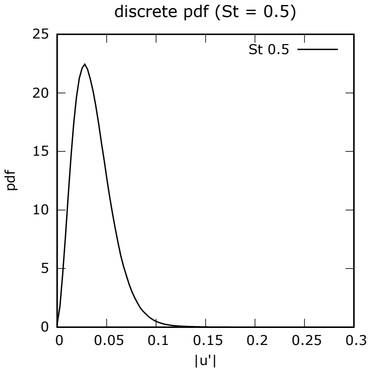
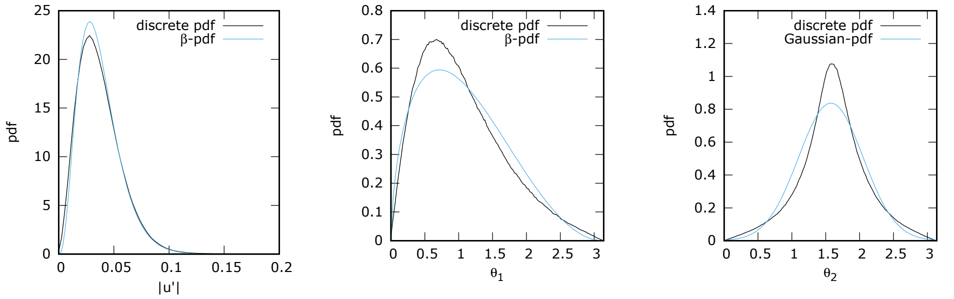


Figure 3.42: Discrete pdf of the sub-grid velocity magnitude seen by particles.



(a) Pdfs for $|\mathbf{u}'|$ (b) Pdfs for θ_1 (c) Pdfs for θ_2

Figure 3.43: Discrete and continuous pdfs for $k_{sgs} = 6.625 \times 10^{-3}$ and $St = 0.5$.

- The gradient of the vorticity $\nabla |\overline{\omega}|$.
- The gradient of the strain $\nabla \left|\overline{\mathcal{S}}\right|$.

and has been estimated that the most interesting directions to define the angles θ_1 and θ_2 are the LES resolved velocity $\widetilde{\mathbf{u}}$ and the gradient of the vorticity $\nabla |\overline{\omega}|$ respectively. It is required to define the continuous pdfs best fitting the statistical distribution of the sub-grid velocity magnitude and the two angles. As shown in Fig. 3.43, these pdfs are a $\beta - pdf$ for the sub-grid velocity magnitude and the angle between the sub-grid velocity and the LES resolved velocity, and a $Gaussian - pdf$ for the angle between the sub-grid velocity and the gradient of the vorticity. Hence, the sub-grid velocity seen by the particle is reconstructed as:

$$\mathbf{u}' = f(|u'|, \theta_1, \theta_2) \quad where \begin{cases} |u'| \rightarrow \beta_{pdf}(a_{|u'|}, b_{|u'|}, c_{|u'|}, d_{|u'|}) \\ \theta_1 \rightarrow \beta_{pdf}(a_{\theta_1}, b_{\theta_1}, c_{\theta_1}, d_{\theta_1}) \\ \theta_2 \rightarrow \mathcal{N}_{pdf}(\mu_{\theta_2}, \sigma_{\theta_2}) \end{cases} \tag{3.106}$$

The parameters a_x, b_x, c_x, d_x of the $\beta{-}pdf$ and the parameters μ_x, σ_x for the $Gaussian{-}pdf$ are obtained from functions. These function are piecewise functions based on third grade polynomials $f(x) = ax^3 + bx^2 + cx + d$ generated fitting the mean and the standard deviation values obtained from DNS simulations. Another required value is the sub-grid time scale of residual motions seen by the particle τ_{sgs}. Currently, this value is estimated using the sub-grid kinetic energy k_{sgs} and the filter size Δ_f as:

$$\tau_{sgs} = \frac{\Delta_f}{\sqrt{\frac{2}{3}k_{sgs}}} \tag{3.107}$$

However, as detailed by Jin et al. [71], the particle inertia affects the sub-grid time scale seen by particles with mass. Therefore, the developed model should be improved including a better correlation to estimate τ_{sgs}, employing also the Stokes number of the particles.

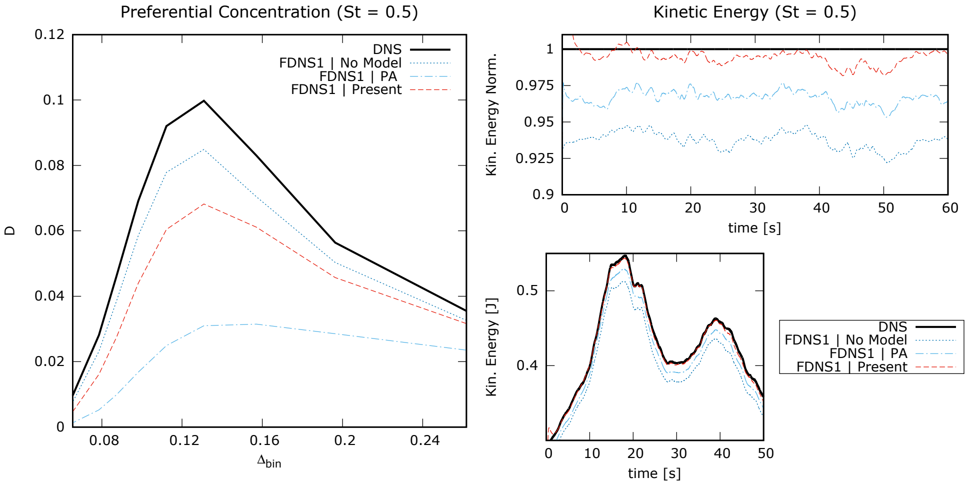
Some preliminary simulations have been done in order to test the capabilities and performance of this new methodology. The statistical information for particles with $St = 0.5$ has been analysed and fitted in an isotropic turbulence simulation of $Re_\lambda = 40$. Afterwards, this statistical information in form of piecewise functions has been used as source to feed the continuous pdfs employed in the presented model. The results obtained using the developed model has been compared against DNS, a LES without sub-grid model, and another LES employing the stochastic model of Pozorski and Apte [65] studied before.

The preferential concentration obtained from the simulations is shown in Fig. 3.44a. As can be seen, the developed model performs better than the stochastic method of Pozorski and Apte. This improvement in the results can also be observed in Fig. 3.45, where the instantaneous spatial distribution of particles in a 2D slice is depicted. As stated in Section 3.4.1, filtering dissipates the preferential concentration effect for particle with $St < 1$, as can be seen comparing DNS results (Fig. 3.45a) and LES without sub-grid model (Fig. 3.45b). Besides, the scattering effect introduced by the model of Pozorski and Apte can be clearly observed in Fig. 3.45c, since fewer flow structures are followed by the particles. The present model is able to improve these results, since more large scales structures can be observed when using this new model (Fig. 3.45d) than when using the stochastic model of Pozorski and Apte (Fig. 3.45c). Therefore, the new model better preserves the preferential concentration effect than the stochastic model. Nonetheless, it is expected to improve this result, since there is still further work to do in order to find the directions and values best suited to estimate the sub-grid velocity direction. Regarding the kinetic energy, shown in Fig. 3.44b, the new proposed model is able to restore fairly well the kinetic energy level of the particles, since the value obtained using the model is almost matching the DNS value.

As stated previously, it is known that the sub-grid time scale of residual motions is affected by particle inertia. Hence, a new analysis adding a constant C to Eq. (3.107) and testing different values for it has been carried out. Obviously, this is not an ideal option for the final model, but this analysis allows to see if a better estimation of this parameter can improve the results of the model. Four different values ranging from $C = 0.01$ to $C = 0.5$ has been tested in Eq. (3.108).

$$\tau_{sgs} = C\frac{\Delta_f}{\sqrt{\frac{2}{3}k_{sgs}}} \tag{3.108}$$

The obtained results are shown in Fig. 3.46. As can be seen in Fig. 3.46a, for the value $C = 0.01$ the preferential concentration effect is even better than without any



(a) Preferential concentration.

(b) Kinetic Energy.

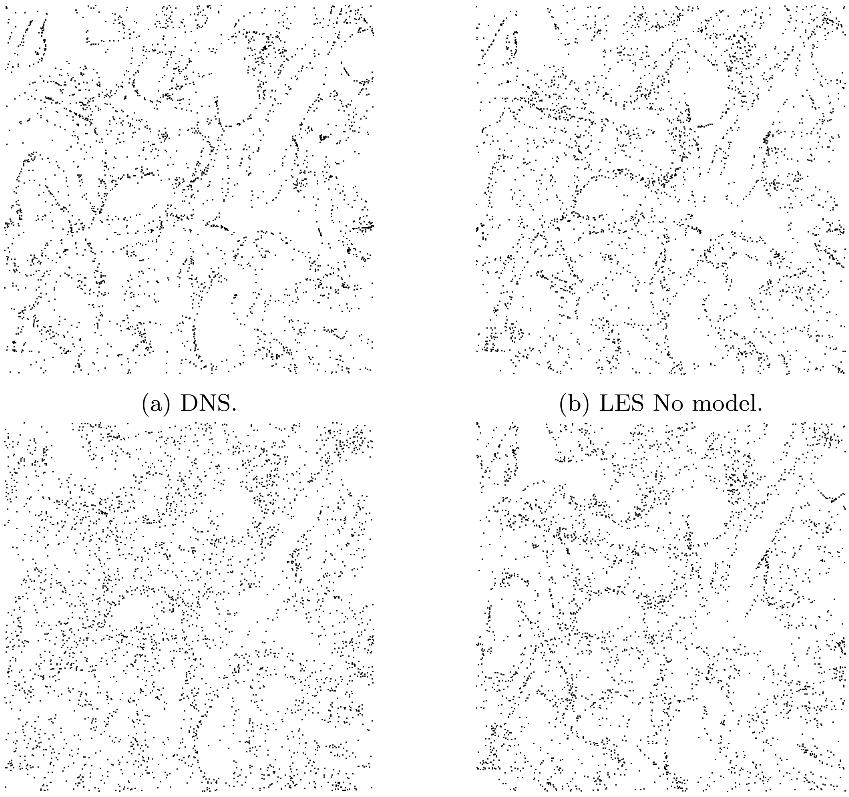
Figure 3.44: Results obtained for $St = 0.5$ in a simulation of force isotropic turbulence with $Re_\lambda = 40$.

sub-grid model, avoiding the scattering effect, while the recovery of the kinetic energy level is minimally affected. Therefore, this analysis demonstrates that the new model can obtain accurate results if a proper dynamic estimation of the sub-grid time scale τ_{sgs} is implemented.

Encouraged by the previous results, and based on the work of [72], a new direction for the sub-grid velocity seen by the particle has been studied. Instead of calculating the direction of the sub-grid velocity $\mathbf{u}'$ from two known directions and two angles θ_1 and θ_2, the employed direction is the laplacian of the resolved velocity $\nabla^2\widetilde{\mathbf{u}}$. The assumption of this new direction for the sub-grid velocity is based on the work of Germano [73, 74], where the filtered velocity $\widetilde{\mathbf{u}}$ is defined as the solution of an elliptic PDE [73, 74]. Specifically, the filtered velocity resolved in LES is expressed using the elliptic differential filter (the Helmholtz filter) defined as:

$$\mathbf{u} = \widetilde{\mathbf{u}} - \epsilon^2\nabla^2\widetilde{\mathbf{u}} \tag{3.109}$$

Hence, now the sub-grid velocity magnitude is calculated as before, but the direction is the laplacian of the resolved velocity. As done before, different values for the constant C are tested in order to calculate the sub-grid time scale τ_{sgs} employed in the model. The obtained results are depicted in Fig. 3.47. As can be seen, with this new sub-grid velocity direction, the developed model is able to reproduce almost



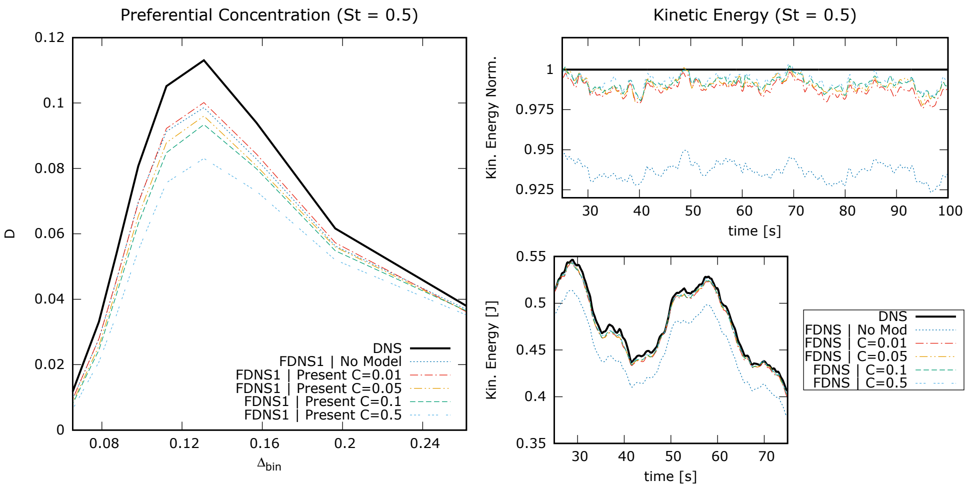
(a) DNS.

(b) LES No model.

(c) LES PA model.

(d) LES present model.

Figure 3.45: Snapshots of particle positions

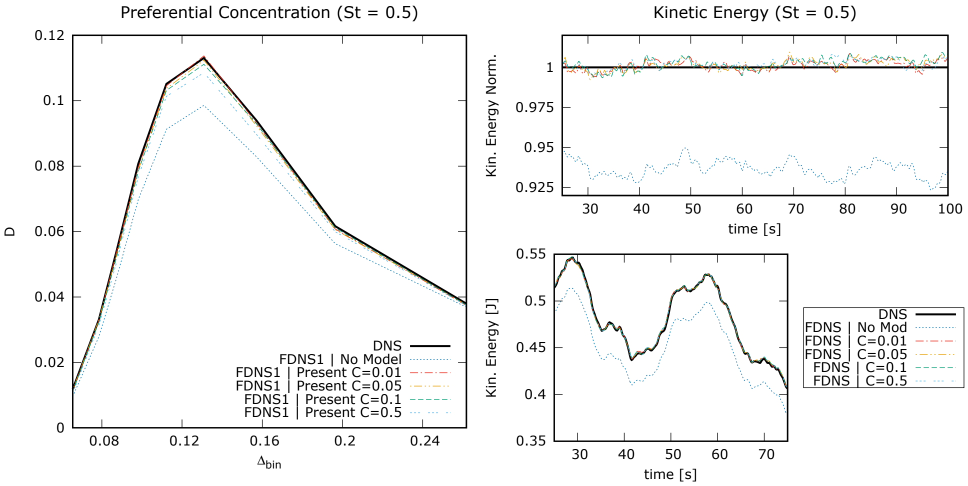


(a) Preferential concentration.

(b) Kinetic Energy.

Figure 3.46: Results obtained for $St = 0.5$ in a simulation of force isotropic turbulence with $Re_\lambda = 40$.

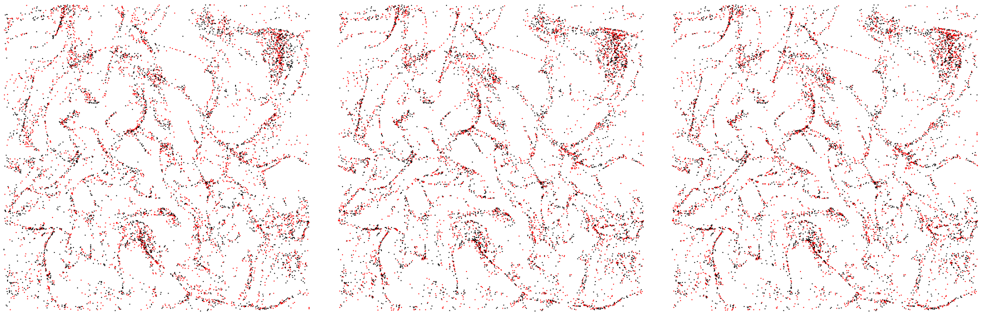
exactly the kinetic energy level of the particles obtained in DNS. Moreover, for a value of C between 0.01 and 0.05 the preferential concentration effect of the DNS simulation is exactly recovered. Aiming to illustrate this improvement in the preferential concentration effect, Fig. 3.48 shows the particle distribution comparing DNS results versus LES results without sub-grid model and with the sub-grid model developed in this section. As can be appreciated, some structures that are lost in LES are recovered when employing the new model. Hence, it is clearly shown that employing the laplacian of the resolved velocity $\nabla^2\widetilde{\mathbf{u}}$ as direction for the sub-grid velocity seen by the particles and with the correct determination of the sub-grid time scale of residual motion τ_{sgs}, the model developed in the present section works really well and without the shortcomings and downsides of the stochastic models previously analysed. Obviously, it is required to test the model for a wider range of Stokes numbers and turbulent conditions, but the results obtained up to now are promising and encourage to continue working, developing and refining this model for sub-grid dispersion of Lagrangian particles.



(a) Preferential concentration.

(b) Kinetic Energy.

Figure 3.47: Results obtained for $St = 0.5$ in a simulation of forced isotropic turbulence with $Re_\lambda = 40$.



(a) No Model

(b) Present Model $C = 0.01$

(c) Present Model $C = 0.05$

Figure 3.48: Slices comparing the particle locations when using the laplacian of the resolved velocity $\nabla^2 \widetilde{\mathbf{u}}$ as direction for the sub-grid velocity seen by the particle (Black points: DNS, red points: LES).

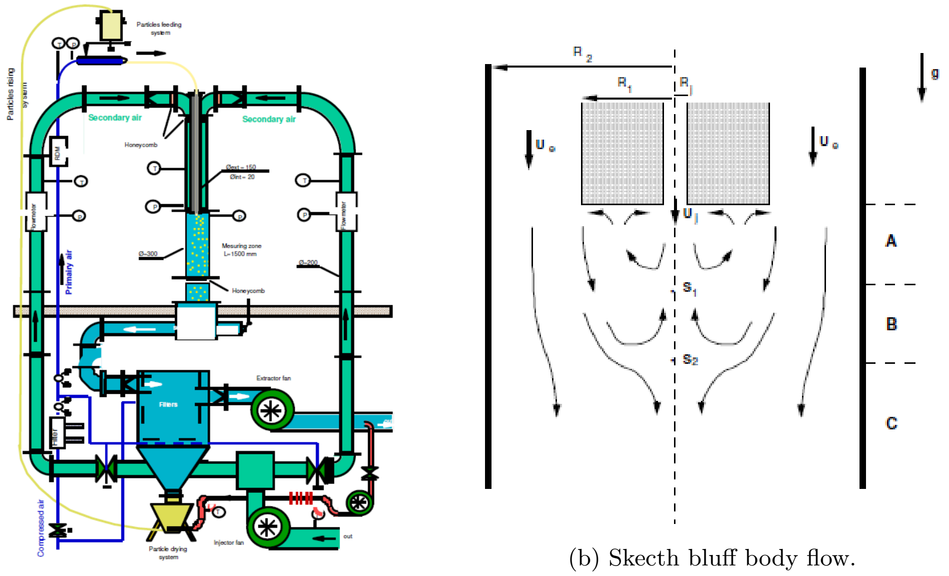
3.5 Hercule Experiment

In this section, a real test case of a particle-laden flow is simulated. The chosen case is the flow loop Hercule of Borée et al. [75], which creates an axisymmetric confined bluff body flow. The velocity of the inner jet is low in order to obtain two stagnation points on the axis of the flow. This configuration with two stagnation points is interesting since inertia properties of the particles and fluid-particle coupling in the inner jet are expected to play a dominant role. This configuration is typical of an industrial application where the objective is to control the mixing of fuel with the air. As stated by Minier [76], this experiment is a challenging case very well suited for the validation of numerical codes of multiphase dispersed flows. In the present work the experiment with a mass loading ratio in the inner jet of $M_i = 22\%$ has been reproduced, which allows to assume one-way coupling regime. The configuration of the Hercule experiment is depicted in Fig. 3.49a. A schematic representation of the inlet zone is shown in Fig. 3.49b, where the inner jet, the coflow, and the two expected stagnation points are represented. The inner jet has a radius of $R_i = 10$ mm, and the inner and outer radius of the coflow are $R_1 = 75$ mm and $R_2 = 150$ mm respectively. In the experiment, poly-dispersed glass particles with density $\rho_p = 2470$ m^3/kg are injected in the inner pipe. The particle distribution of the injected particles is shown in Fig. 3.50.

3.5.1 Numerical set-up

In the present study the *Unresolved Reynolds stresses* of momentum equation are closed by means of the Dynamic One-Equation sub-grid Model (DOEM) developed by Davidson [77] and successfully tested for different flow configurations as channel flows [78] or various bluff bodies [79, 80]. This one-equation model solves an equation for the sub-grid kinetic energy k_{sgs}, which value is required to use the stochastic models previously presented in Section 3.4.1. Three different simulations are carried out changing the model employed for the sub-grid velocity seen by the particles $\mathbf{u}'$. One simulation is computed neglecting the influence of the sub-grid scales over the Lagrangian particles ($\mathbf{u} = \widetilde{\mathbf{u}}$), and one simulation is carried out for each one of the stochstic models previously analysed, the one presented by Pozorski & Apte [65] and the one developed by Bini & Jones [66, 67].

The case has been simulated in an unstructured tetrahedral mesh with almost $2M$ cells. Regarding the boundary conditions, for the inner pipe a synthetic turbulence generator [81] has been used to reproduce the turbulent inlet, while for the coflow the mean velocity profile obtained from the experiment has been imposed. For all the solid walls in the domain a no-slip boundary condition has been employed. Particles are injected in the inner pipe using the particle size distribution and the mean velocity measured in the experiment. The pressure-velocity coupling is resolved by means of the Fractional Step method and the Poisson equation solved using the iterative



(a) Hercule experiment configuration.

(b) Skecth bluff body flow.

Figure 3.49: Scheme of the set-up of the Hercule experiment.

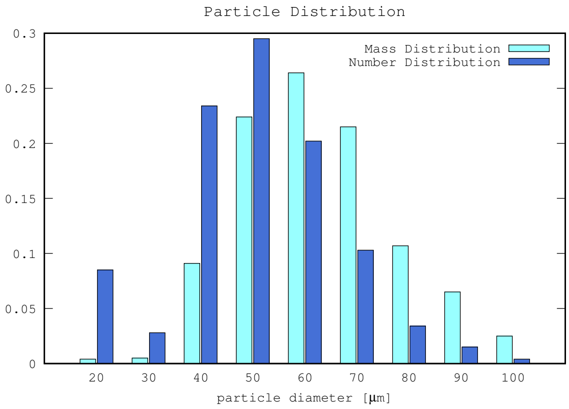


Figure 3.50: Particle distribution of injected poly-dispersed particles.

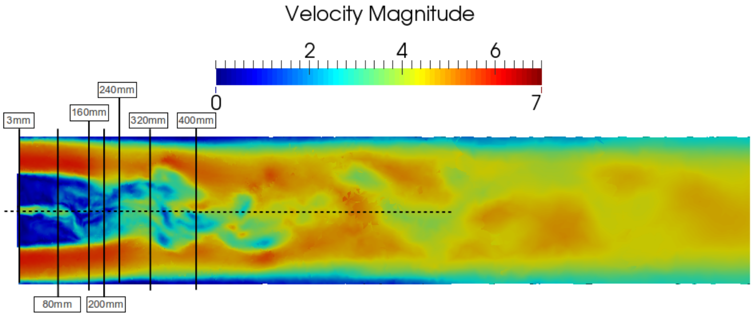


Figure 3.51: Slice with the instantaneous velocity field showing the location of the radial stations.

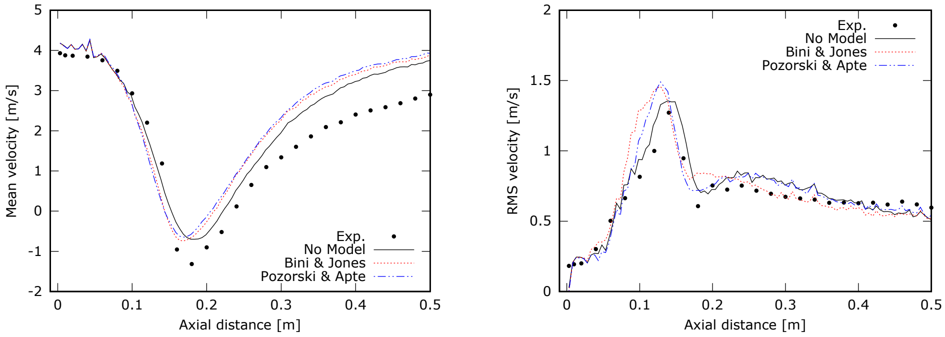
Conjugate-Gradient (CG) method with Jacobi diagonal scaling.

3.5.2 Results

Flow and particles measurements were obtained in the axis and in radial stations located at 3, 80, 160, 200, 240, 320 and 400 mm downstream the edge of the bluff body (see Fig. 3.51). Figures 3.52a and 3.52b show the mean and RMS value of the carrier phase velocity in the axis of the jet respectively, and as can be seen, the experimental results are quite well captured by the current simulations (the minor differences between the cases are due to time-averaging reasons). The mean-axial velocity of the carrier phase is slightly over-predicted and the two stagnation points are not perfectly captured. This is due to, as stated by Riber et al. [82], the case configuration is extremely sensitive to inlet boundary conditions, and it is difficult to exactly reproduce the experimental set-up. Hence, in order to better match the experimental results, a more precise setting of boundary conditions and/or a refinement of the computational mesh would be required.

As consequence of this small over-prediction in the carrier phase velocity, the mean axial velocity of the particles is also overestimated, as can be seen in Figs. 3.53 and 3.54. In any case, the results obtained for particles velocities is consistent with the experimental results. It can be clearly observed how the lightest particles ($d_p =$ 20 µm) tend to follow the flow, while the inertial effects dominate in the heaviest particles ($d_p =$ 80 µm). In this section only two different representative class of particles are depicted, while the results for all the others particles' sizes are depicted in Appendix B.1.

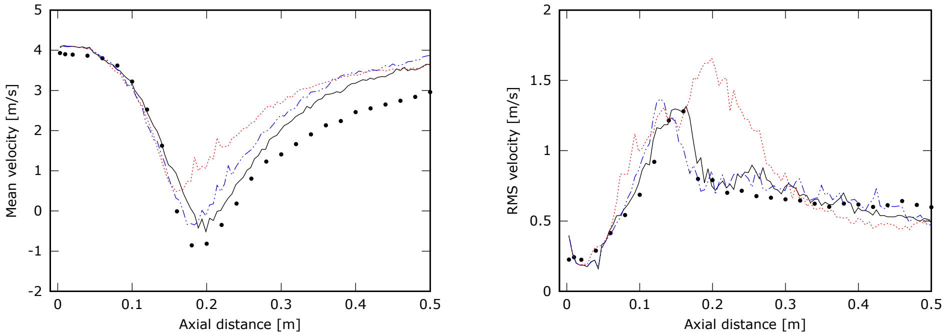
The radial velocity profiles obtained from simulations have also been compared against experimental results for the continuous phase (Figs. 3.55a and 3.55b) and the dispersed phase (Figs. 3.56a, 3.56b, 3.57a and 3.57b). As can be observed, the first



(a) Mean velocity.

(b) RMS velocity.

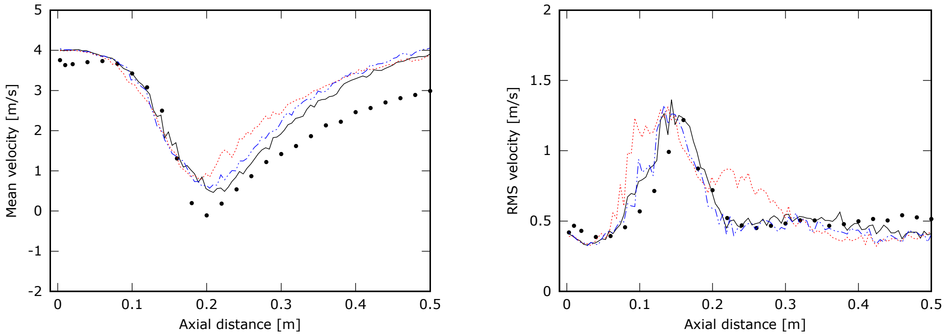
Figure 3.52: Axial results of carrier phase.



(a) Mean velocity.

(b) RMS velocity.

Figure 3.53: Axial results of dispersed phase (particles $d_p = 20$ µm).



(a) Mean velocity.

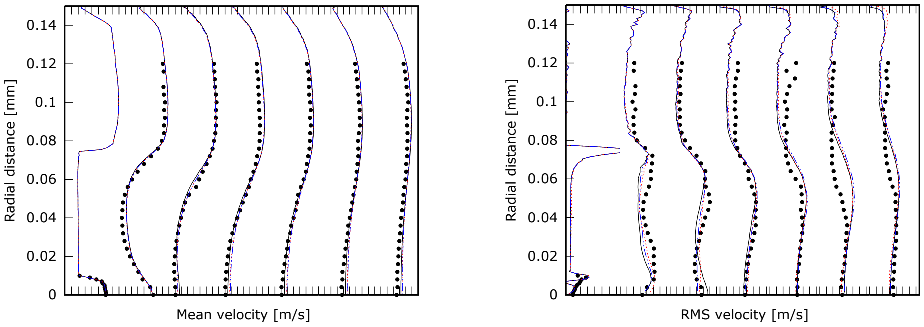
(b) RMS velocity.

Figure 3.54: Axial results of dispersed phase (particles $d_p = 80$ μm).

order statistics are almost perfectly captured. Close to the axis, the over-prediction in the velocity previously mentioned can be recognized. In the radial stations the RMS values are worse captured than in the axial station, probably due to the mesh is not fine enough and also more averaging time will be required. Nonetheless, the trending obtained in the experiment is well captured in the simulations. The overestimation of RMS velocity value in the first radial station is due to there is no available data from the experiments for the velocity fluctuations at the inlet. Hence, an estimated turbulence level intensity using a white-noise model have been used. On the other hand, the discrepancies and non-physical values obtained for the velocity fluctuation far from the axis in the radial profiles are due to in that locations there are very few particles, and their number is not enough to obtain a proper averaged value.

Regarding the effect of the turbulent models over the particles, as could be expected from the analysis done in the isotropic turbulence test and presented in Section 3.4.1, both models increase the kinetic energy of the particles, specially the model of Bini and Jones, which also overestimates the RMS value between the stagnation points. This effect is more remarkable for small inertia particles.

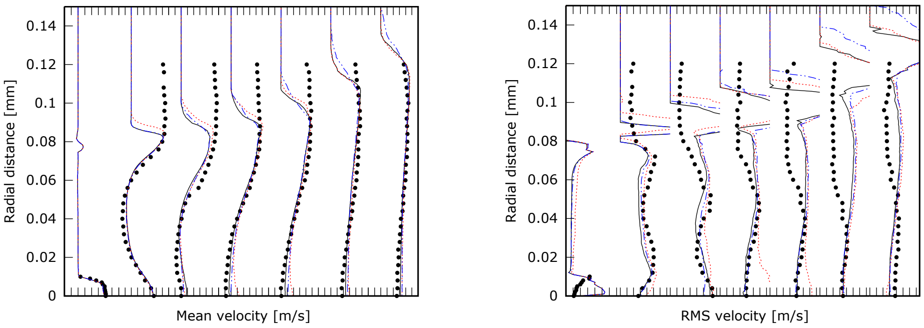
The instantaneous velocity field of the flow-loop Hercule for both continuous and dispersed phases of a simulation carried out in this work is depicted Fig. 3.58a. In Fig. 3.58b can be clearly seen the effect of the Stokes number in the particles' motion, and how the particles tend to be clustered in certain regions of the domain depending on their weight/size. Regarding the particle concentration effect, as previously mentioned in this chapter, the Fig. 3.59 clearly shows how particles tend to be clustered in regions of low vorticity and high rate of strain.



(a) Mean velocity.

(b) RMS velocity.

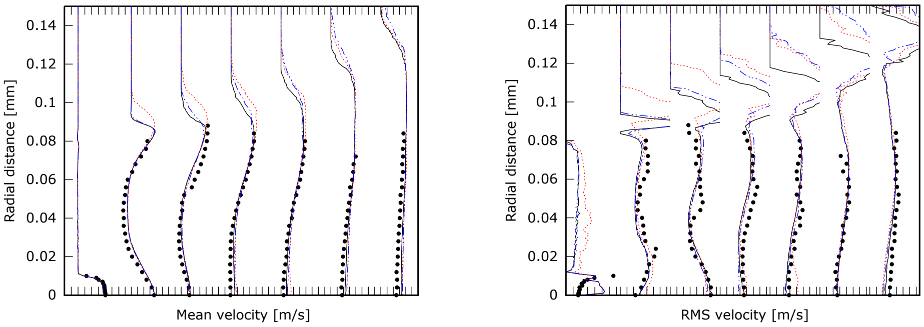
Figure 3.55: Radial results of carrier phase.



(a) Mean velocity.

(b) RMS velocity.

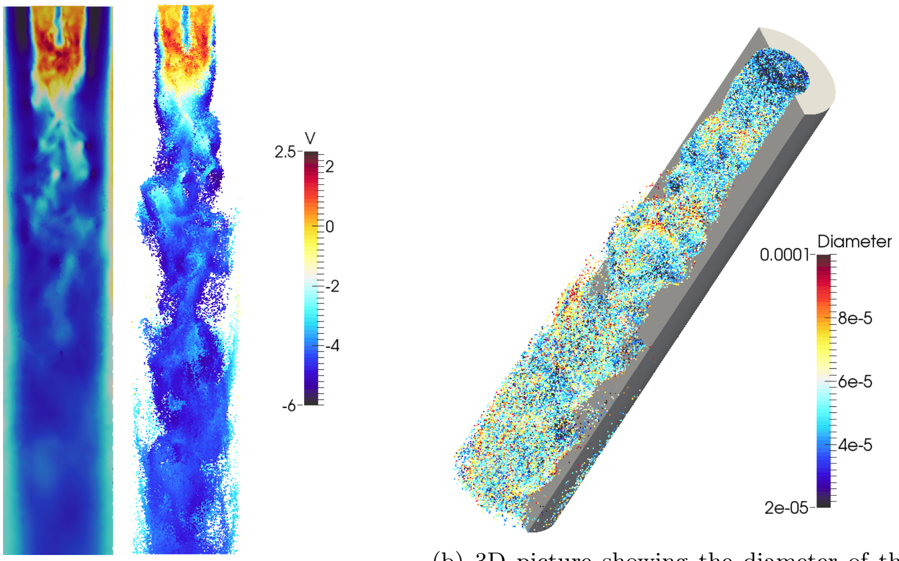
Figure 3.56: Radial results of dispersed phase (particles $d_p = 20$ µm).



(a) Mean velocity.

(b) RMS velocity.

Figure 3.57: Radial results of dispersed phase (particles $d_p = 80$ µm).



(a) Instantaneous velocity field of both continuous and dispersed phases.

(b) 3D picture showing the diameter of the particles.

Figure 3.58: Instantaneous results of the flow-loop Hercule simulation.

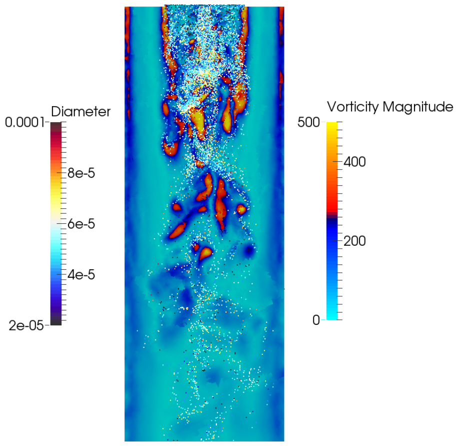


Figure 3.59: Slice showing the vorticity of the continuous phase and the position of the particles.

3.6 SimInhale benchmark case

Inhaled drugs via aerosol are used to treat different respiratory diseases like asthma, chronic obstructive pulmonary disease (COPD), cystic fibrosis and pulmonary infections, or to deliver the chemotherapy agents in cases of tumours in the respiratory system like lung tumours. During last years there has been a huge progress in the most common technologies applied to pulmonary imaging, like Computed Tomography (CT) and Magnetic Resonance Imaging (MRI), and there have emerged some promising technologies like Phase-contrast Imaging, Grating Interferometry (GI) or Laboratory Propagation-base Phased-contrast Imaging (PBI) [83]. Nonetheless, neither of these technologies can offer nowadays enough resolution to quantify regional deposition of inhaled drugs. Within this framework Computational Fluid Dynamics simulations can be a powerful instrument in order to fill this gap and provide accurate information over regional deposition in respiratory airways. However, after a survey of the recent literature it has been found that there is a wide range and heterogeneity in the methods and assumptions applied into this field, revealing the necessity of a good-practice guide as well as a benchmark test case intended for comparison and assessment of the numerical codes developed to study regional deposition of inhaled drugs. With the aim of provide to the scientific community a reference case in order to assess and validate CFD studies of regional deposition in respiratory airways, *in vitro* deposition measurements have been conducted in a human-based model of the upper airways, and several research groups have carried out numerical simulations reproducing the experiment using different modelling and discretization approaches.

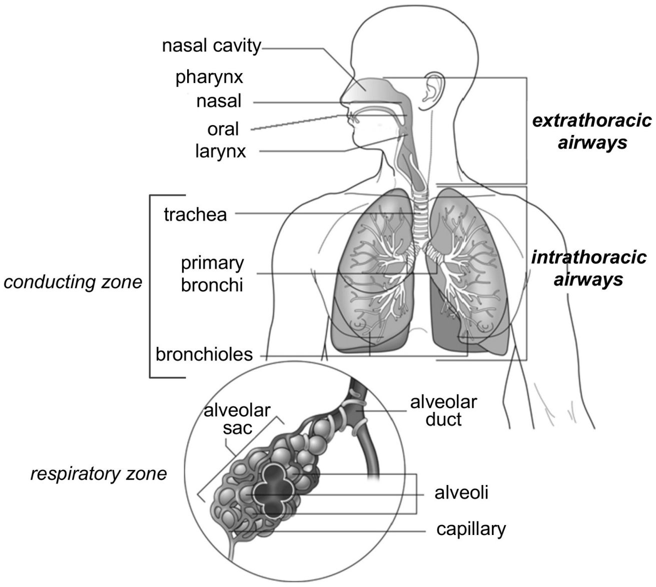


Figure 3.60: Schematic of the respiratory system.

The work presented in this section is one of the numerical simulations included within the framework of the SimInhale COST Action MP1404 [84]. The results of this collaborative effort have been publish in [85], where the numerical results obtained using the different approaches are compared and assessed against the experimental results. Therefore, below is presented the numerical simulation of a case reproducing the regional aerosol deposition in an *in vitro* experiment imitating realistic human airways.

3.6.1 Benchmark case

The realistic airway geometry used to construct the benchmark case model is shown in Fig. 3.61(a). It comprises of the oral cavity, larynx and tracheobronchial airways down to the 12^{th} generation of branching. The tracheobronchial tree was acquired from a human lung of an adult male, excised at autopsy and fixed with a liquid rubber solution at nearly end-inspiratory volume. The lung tissue was removed and the rubber cast of the bronchial tree was scanned using high-resolution computed tomography (HRCT) (Schmidt et al. [86]). The extrathoracic airways were obtained from the Lovelace Respiratory Research Institute (LRRI) upper airway model. The oral cavity was moulded from an *in vivo* dental impression of a Caucasian male at approximately 50% of the full opening, and the remaining of the model was acquired from a cadaver (Cheng et al. [87]). The LRRI geometry was obtained as a wax cast,

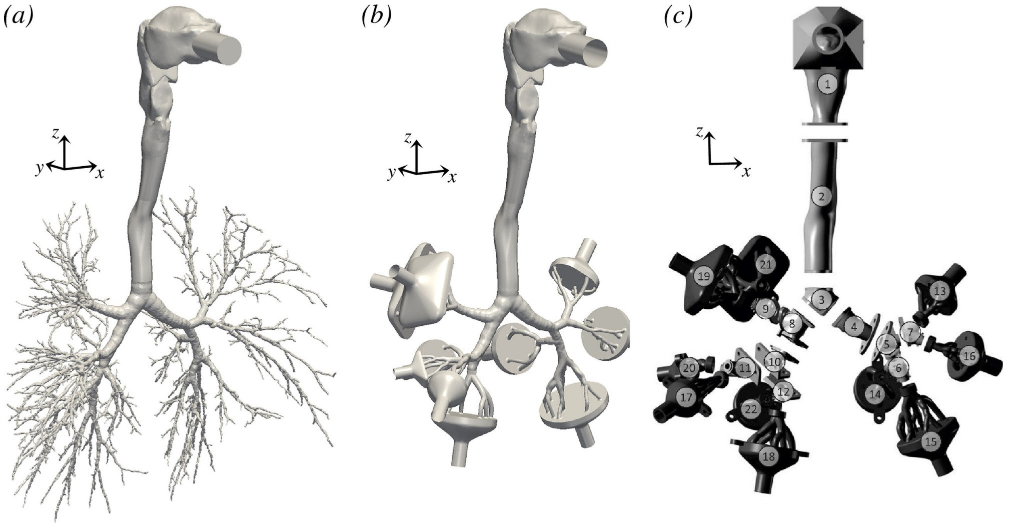


Figure 3.61: Geometry of the respiratory airways: (a) original realistic airway geometry; (b) geometry adopted for the benchmark case; (c) physical segmented model for deposition measurements.

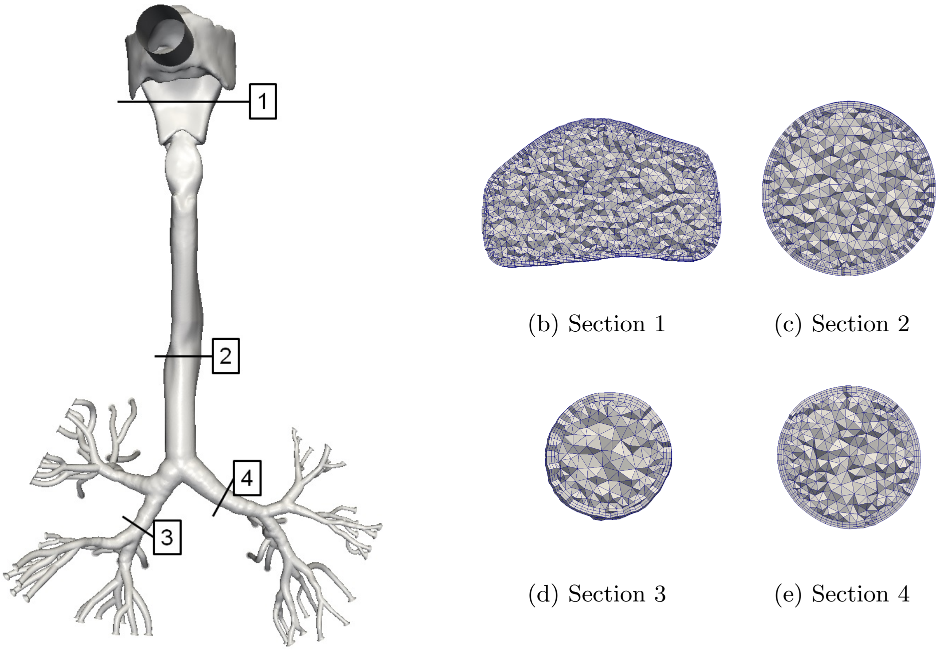
which was scanned by an Atos (GOM, Braunschweig, Germany) device, converted to STL format, and concatenated with the bronchial tree model at the trachea. Further details on the construction of the airway model can be found in Lizal et al. [88].

In Fig. 3.61(b) is shown the geometry employed in the *in vitro* experiments and numerical simulations. Only branches with diameter above 3 mm were used, and the terminal bronchi segments were connected to 10 outlets. The physical model for deposition measurements was produced by stereolithography (Lizal et al. [89]). A 3 mm thick envelope was created around the original geometry to obtain a negative cast of the airways. The model was divided into sections to facilitate the measurement of regional deposition by various methods, such as optical microscopy and gravimetry Fig. 3.61(c).

The experiments were performed at steady-state inhalation for three different flow rates: 15 l/min, 30 l/min and 60 l/min.

3.6.2 Numerical setup

The numerical simulation of the experiment for a flow rate of 60 l/min has been reproduced employing the numerical code TermoFluids. Since the case is turbulent, a LES approach is employed, where the *Unresolved Reynolds stresses* of momentum equation have been modelled using the Variational Multi-Scale (VMS) model of Hughes et al. [90]. As inlet boundary condition, a mapped inlet obtained from an auxiliary simulation in a turbulent pipe has been employed. At the outlets, an uniform flow-rate has been imposed according to the mean values recorded in the experiment. The employed computational mesh is an unstructured tetrahedral mesh of around 10M cells with a prismatic wall boundary-layer composed by three layers. Slides of different cross-sections of the mesh are depicted in Fig. 3.62.



(b) Section 1

(c) Section 2

(d) Section 3

(e) Section 4

(a) General overview.

Figure 3.62: Mesh details.

The forces accounted for in the dispersed phase are the drag force, the gravity force and the Brownian motion. The drag-force correlation employed in the present study is the one proposed by Schiller and Naumann [11]. Moreover, since some of the injected particles are very small, the Cunningham correction factor proposed by Davies [21] is used in order to account for the rarefaction effects. In the present study the influence of sub-grid velocity over the particles has been neglected, it is, the carrier-phase velocity seen by particles is the filtered velocity resolved in LES ($\mathbf{u} = \widetilde{\mathbf{u}}$). The criteria employed to consider that a particle is deposited is by contact, i.e., as soon as the particle comes into contact with an airway wall the particle is considered deposited. Reflection and re-suspension are not contemplated, since in the *in vitro* experiments liquid particles are used, which deposit when they hit the surface of the cast. The same assumption can be adopted for *in vivo* cases, where the existence of a mucus layer on the inner walls of the airways ensures that particles colliding with the surface deposit. Ten different particle's sizes with a diameter d_p ranging from 0.5 µm to 20 µm have been considered. For each particle size 100000 particles have been injected in the domain during the initial 0.025 seconds of the simulation. The particles are injected assuming a random distribution along all the inlet diameter. The particle density is set to $\rho_p = 914$ kg/m^3, which is the density of the diethylhexyl sebacate (DEHS) used in the experiment in air at room temperature.

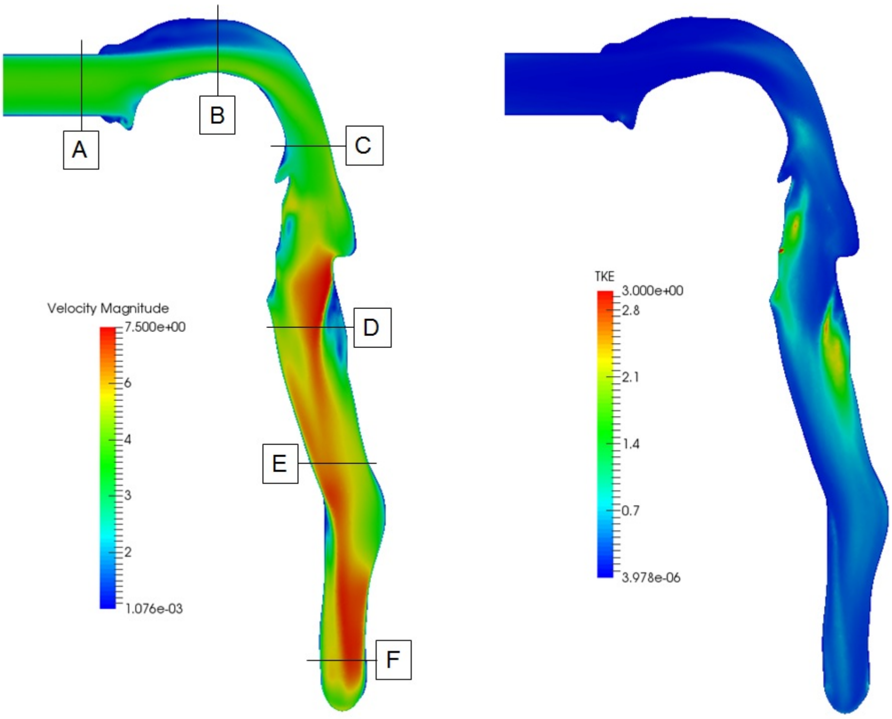
3.6.3 Results

Air flow

Figure 3.63 depicts the mean velocity magnitude and the mean turbulent kinetic energy (tke) in the central sagittal plane of the extrathoracic airways and the trachea. Moreover, in Figs. 3.64 to 3.69 the profiles at various cross-sections are shown, allowing to see a more detailed information of the velocity and the kinetic energy. The velocities in the mouth (B) and the pharynx (C) are low, observing a large separated region along the upper wall of the oral cavity. As a result of the glottal constriction, the flow accelerates in the larynx and a separated shear-layer develops at the level of the vocal cords due to the airway curvature (D). A recirculation region is observed near the posterior wall of the trachea, behind the separated shear-layer (D). Further downstream, a bend in the trachea (E) causes the high-speed velocity to shift from the anterior to the posterior wall (F), and leads to the formation of a small region of separation at the anterior wall.

Regarding the turbulent kinetic energy, it has small values at the inlet, and start to increase towards the back of the mouth and the upper pharynx. The maximum kinetic energy occurs in the larynx, near the anterior wall, downstream cross-section C, and in the trachea, near the edge of the separated shear layer (D).

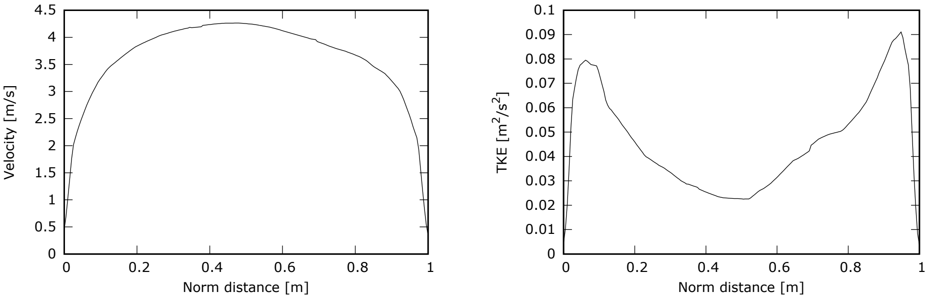
Downstream of the trachea, the flow enters the bronchial tree. Contours of the mean velocity and mean turbulent kinetic energy in the carina and main bronchi are shown in Figs. 3.70 to 3.72. The outlet conditions prescribed in the simulation, based on the flow distribution measured in the *in vitro* experiment, result in high asymmetry in the ventilation of the two lungs, with the left lung receiving 29% of the inhaled air, and the right lung receiving 71%. This asymmetry is evident in the flow entering the tracheal bifurcation. A high-speed jet is observed along the right wall (G), and the stagnation point at the carina is offset towards the left. A larger recirculation region develops along the top wall of the left main bronchus (H), as it is at a more acute angle than the right. The asymmetry in the incoming flow and in the geometry also results in higher velocities inside the right main bronchus.



(a) Mean velocity.

(b) Mean TKE.

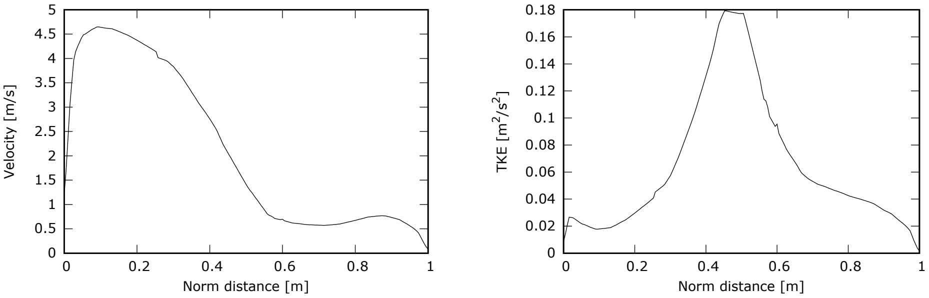
Figure 3.63: Central sagittal plane of the extrathoracic airways and trachea (segments 1 and 2).



(a) Mean velocity.

(b) Mean TKE.

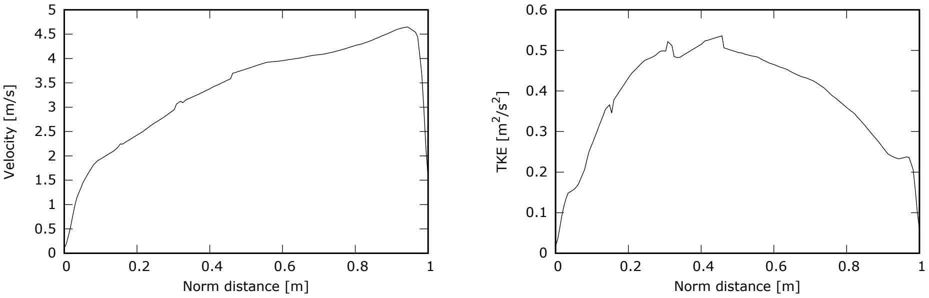
Figure 3.64: Cross-section A.



(a) Mean velocity.

(b) Mean TKE.

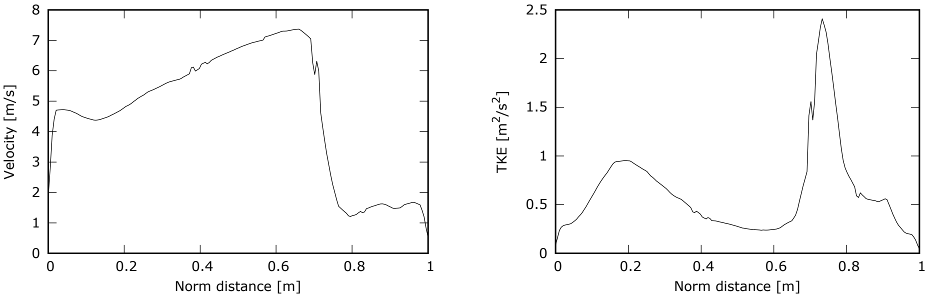
Figure 3.65: Cross-section B.



(a) Mean velocity.

(b) Mean TKE.

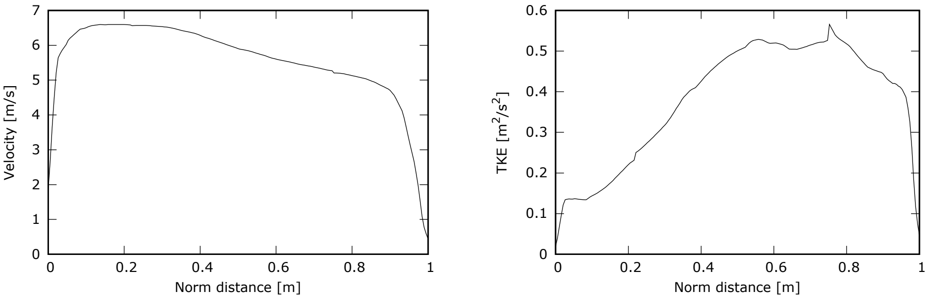
Figure 3.66: Cross-section C.



(a) Mean velocity.

(b) Mean TKE.

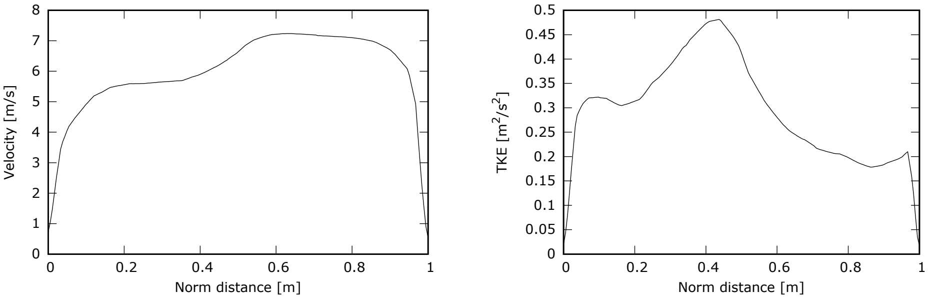
Figure 3.67: Cross-section D.



(a) Mean velocity.

(b) Mean TKE.

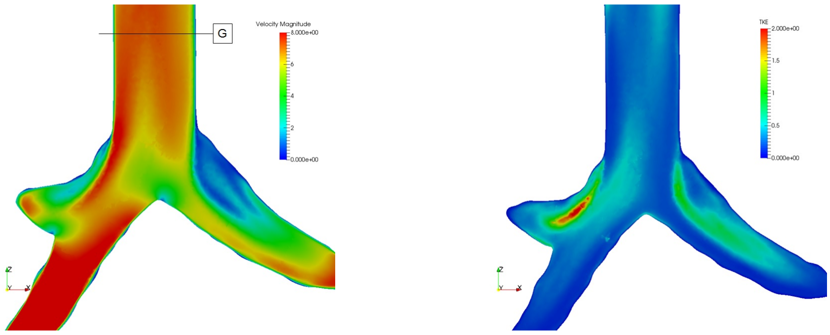
Figure 3.68: Cross-section E.



(a) Mean velocity.

(b) Mean TKE.

Figure 3.69: Cross-section F.



(a) Mean velocity.

(b) Mean TKE.

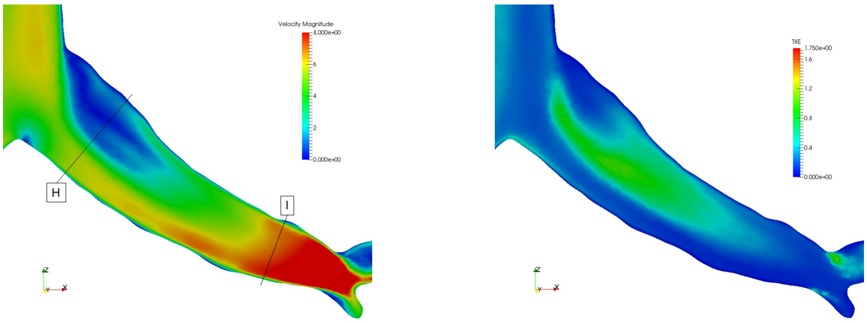
Figure 3.70: Carina (segment 3).

Although the levels of mean turbulent kinetic energy appeared to decay in the trachea, high levels of tke are observed in this region, in zones of high mean shear. The maximum kinetic energy occurs in the right main bronchus, between the separated shear layer and the recirculation region. New flow instabilities are induced in the bifurcation regions due to the severe geometric transition from the parent to the daughter branches, and result in high turbulence in the first few generation of the tracheobronchial tree [91].

Profiles of the mean velocity and the mean turbulent kinetic energy at the displayed cross-sections are provided in Figs. 3.73 to 3.75.

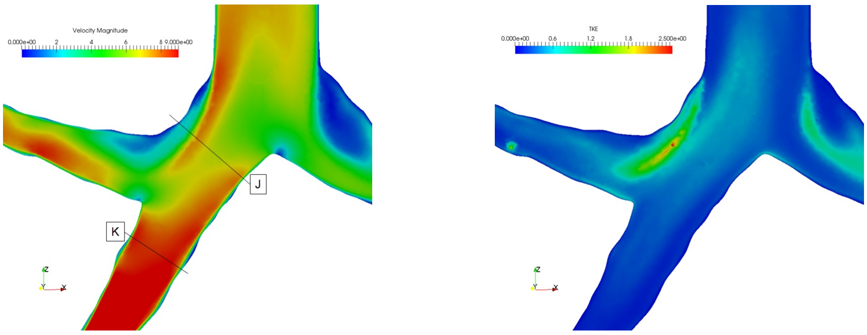
Following, in Figs. 3.76 to 3.79 the contours of mean velocity and mean kinetic energy in smaller airways of the bronchial tree are displayed As the left main bronchus narrows downstream, higher velocities are observed. The asymmetric branching results in a larger region of separation on the outer wall of the upper daughter branch. The airways are short, so the flow cannot develop fully. Due to upstream effects and different branching geometries, the flow varies significantly across the various bifurcations. For example, in segment 7 (Fig. 3.78), the flow enters the bifurcation with a large separation zone along the upper wall and a high-speed jet along the bottom, which leads to significant asymmetry in the flow distribution between the daughter branches. Flow separation along the outer walls of the the daughter branches is observed in most bifurcations, due to the relatively high local Reynolds number ($Re > 1000$) and abrupt increases in cross-sectional area. The turbulent kinetic energy remains high in these airways which belong to generation 2 and 3. Turbulence has been observed to propagate to a few generations even at low local Reynolds number, due to the enhancement of flow instabilities at the bifurcations [91].

Contours and profiles for other segments and stations have been included in Ap-



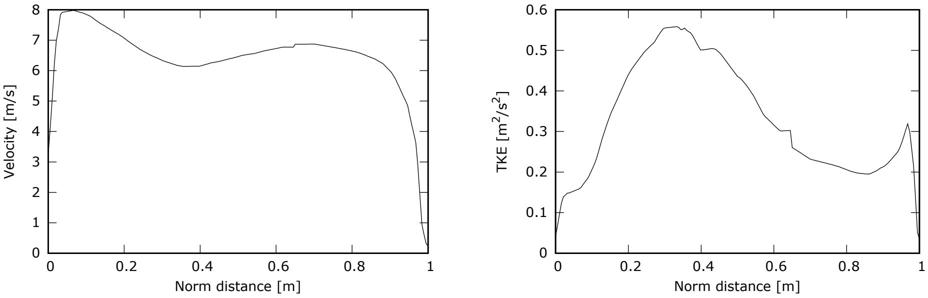
(a) Mean velocity. (b) Mean TKE.

Figure 3.71: Left main bronchus (segment 4).



(a) Mean velocity. (b) Mean TKE.

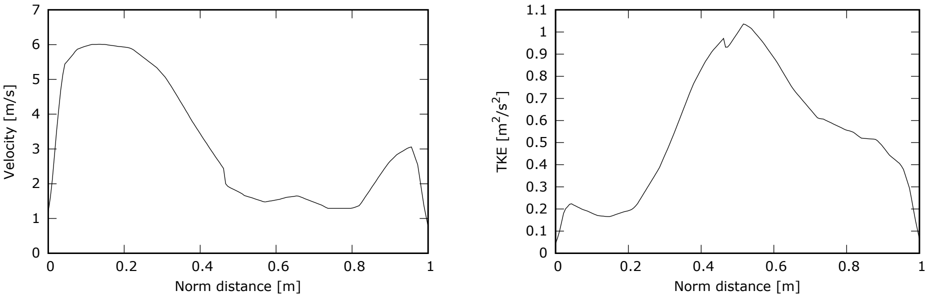
Figure 3.72: Right main bronchus (segment 8).



(a) Mean velocity.

(b) Mean TKE.

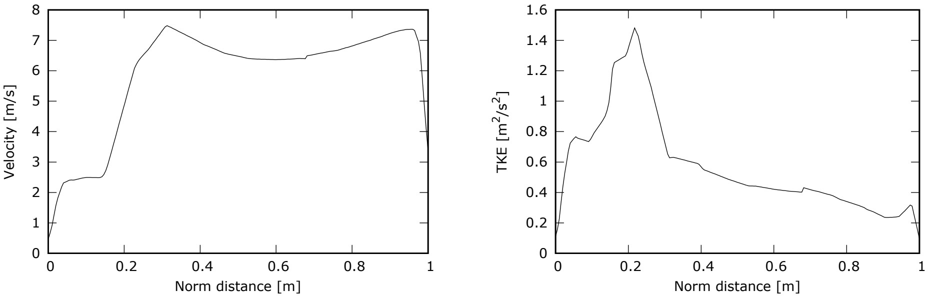
Figure 3.73: Cross-section G.



(a) Mean velocity.

(b) Mean TKE.

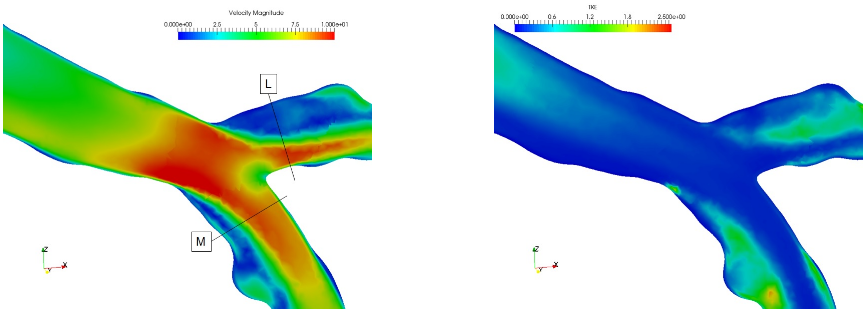
Figure 3.74: Cross-section H.



(a) Mean velocity.

(b) Mean TKE.

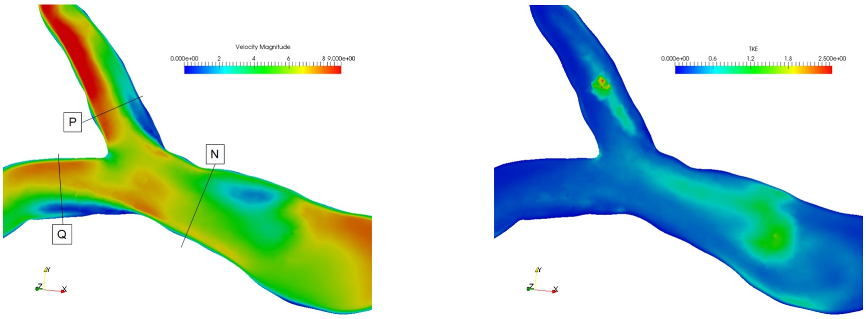
Figure 3.75: Cross-section J.



(a) Mean velocity.

(b) Mean TKE.

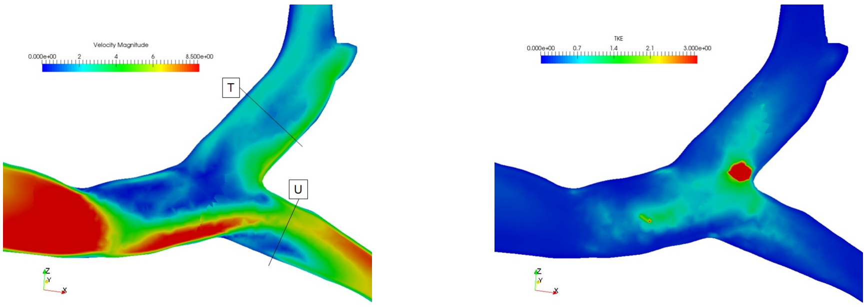
Figure 3.76: Segment 5 in left lung.



(a) Mean velocity.

(b) Mean TKE.

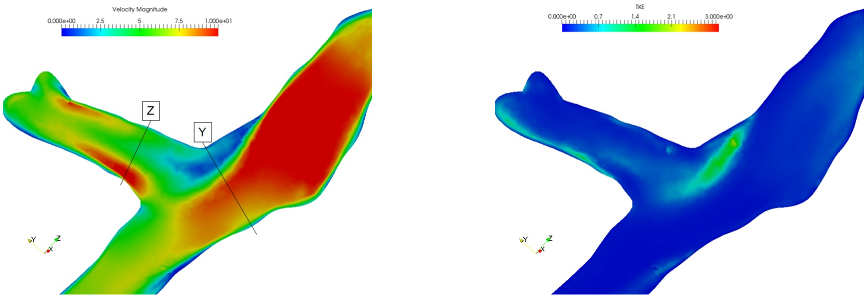
Figure 3.77: Segment 9 in right lung.



(a) Mean velocity.

(b) Mean TKE.

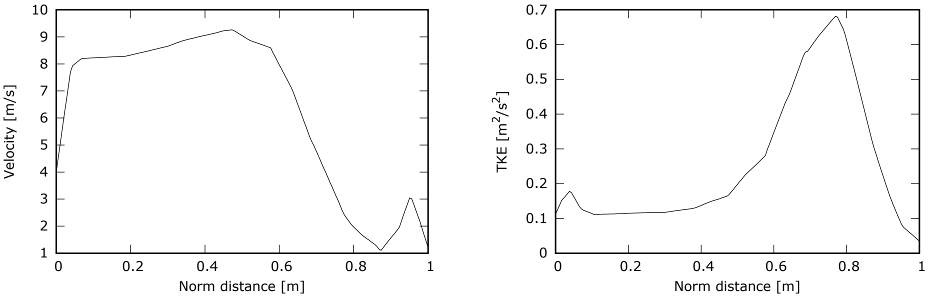
Figure 3.78: Segment 7 in left lung.



(a) Mean velocity.

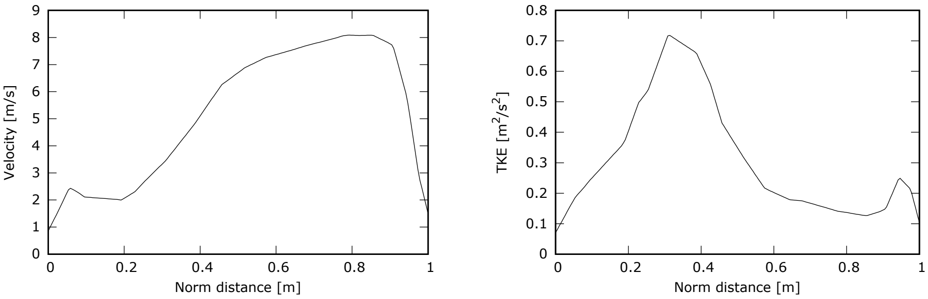
(b) Mean TKE.

Figure 3.79: Segment 12 in right lung.



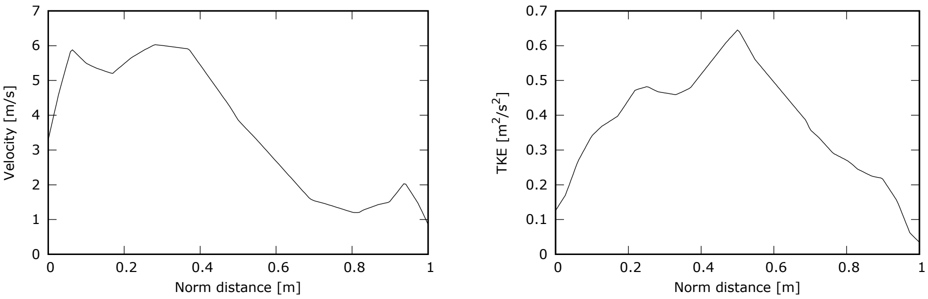
(a) Mean velocity. (b) Mean TKE.

Figure 3.80: Cross-section M.



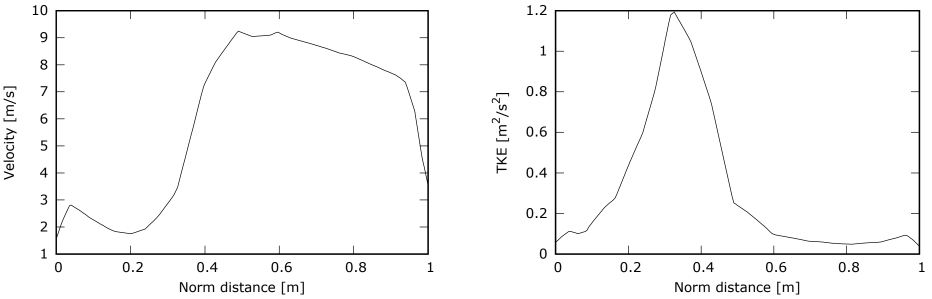
(a) Mean velocity. (b) Mean TKE.

Figure 3.81: Cross-section P.



(a) Mean velocity. (b) Mean TKE.

Figure 3.82: Cross-section U.



(a) Mean velocity.

(b) Mean TKE.

Figure 3.83: Cross-section Y.

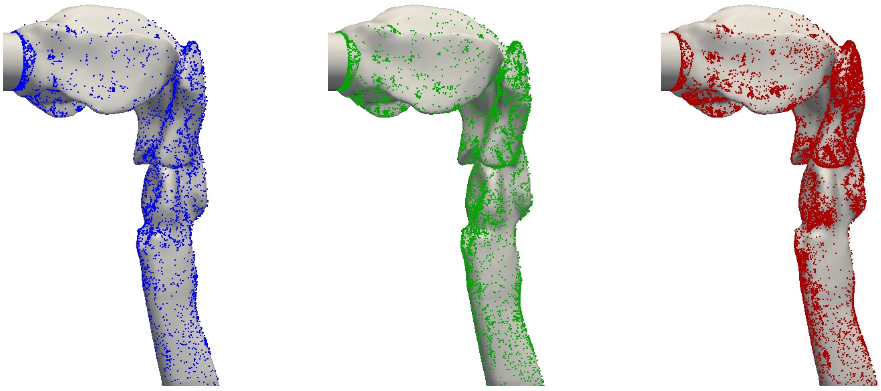
pendix B.1.

Aerosol deposition

In the present section it is examined the deposition of the aerosol particles along the respiratory airways in the simulation of the *in vitro* experiment. The deposition patterns in the extrathoracic airways, trachea and major bronchi are shown in Figs. 3.84 to 3.87, for particle sizes 1, 4.3 and 8 µm. The same main deposition hotspots are observed across all three particle sizes. In the extrathoracic airways, major deposition occurs along the side walls of the pharynx. Additional hotspots are observed on the sharp step in the larynx, due to the sudden change in flow directions, and on the anterior wall slightly downstream, due to the separated shear layer. Some deposition also occurs in front of the tongue, where the incoming flow from the inlet pipe impinges.

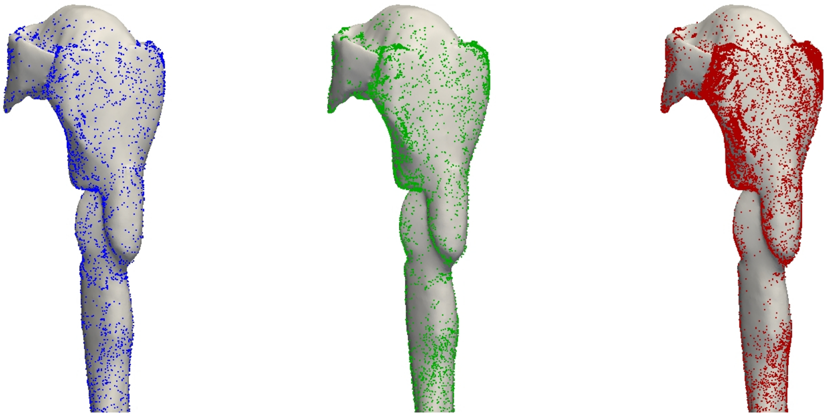
In the trachea, low deposition is observed for small particles. As particle size increases, deposition increases significantly. Particles deposit primarily along the upper half of the trachea, where the separated shear layer impinges on the anterior wall, and at the bend further downstream, where the high-speed velocity shifts from the anterior to the posterior wall (see Fig. 3.63a). Downstream of the trachea, three main hotspots are observed: (i) just left of the carina, at the location of the stagnation point; (ii) along the upper wall of the right bronchus; and (iii) along the bottom wall of its daughter airway, as it branches off at a sharp angle.

In order to examine regional deposition patterns, a more detailed information of the deposition results are shown in Figs. 3.88 to 3.97, where the deposition fractions per segment of the airways are plotted for the ten different particle sizes injected in the numerical simulation. Segment numbers are shown in Fig. 3.61(c). In Fig. 3.93 the deposition results obtained in the simulations for particles with a diameter of d_p = 4.3 µm are compared against the results obtained in the experiment. As can



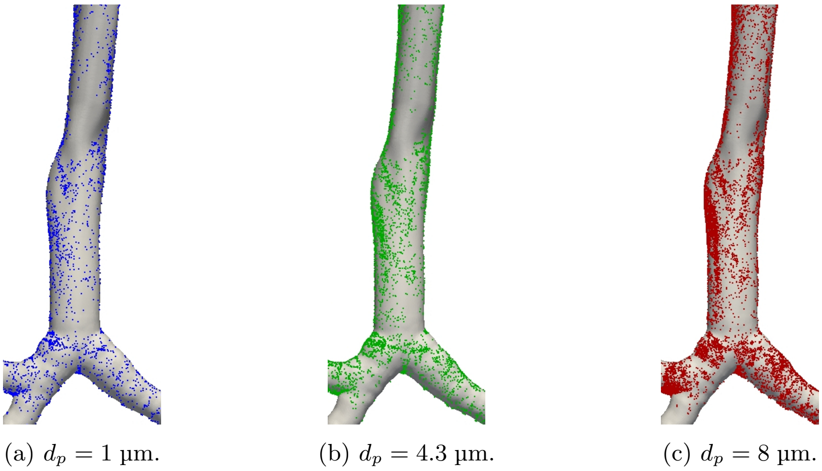
(a) $d_p = 1$ µm. (b) $d_p = 4.3$ µm. (c) $d_p = 8$ µm.

Figure 3.84: Sagittal view of the extrathoracic ways.



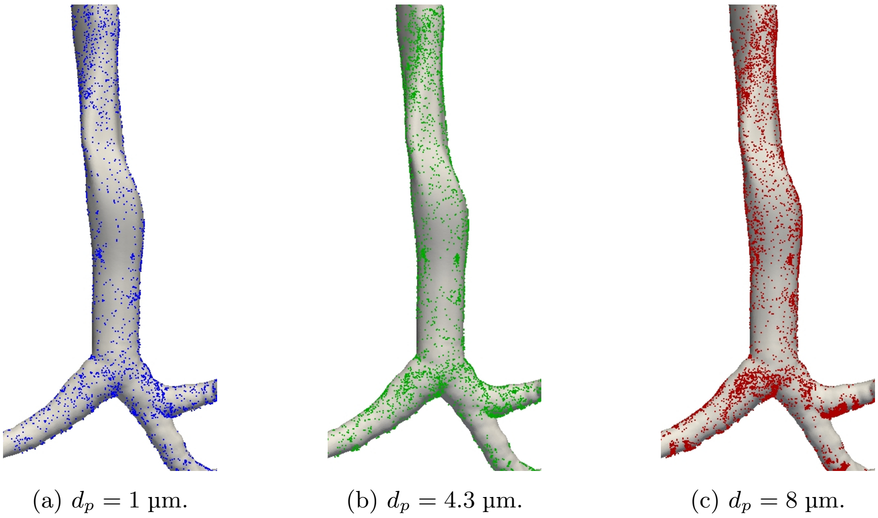
(a) $d_p = 1$ µm. (b) $d_p = 4.3$ µm. (c) $d_p = 8$ µm.

Figure 3.85: Posterior view of the extrathoracic ways.



(a) $d_p = 1$ µm. (b) $d_p = 4.3$ µm. (c) $d_p = 8$ µm.

Figure 3.86: Anterior view of the trachea and major bronchial ways.



(a) $d_p = 1$ µm. (b) $d_p = 4.3$ µm. (c) $d_p = 8$ µm.

Figure 3.87: Posterior view of the trachea and major bronchial ways.

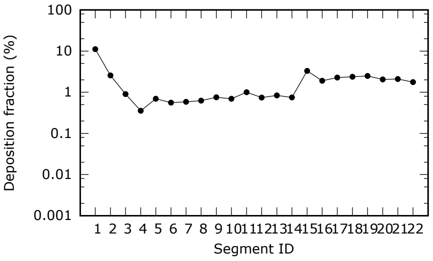


Figure 3.88: Deposition $d_p = 0.5$ µm.

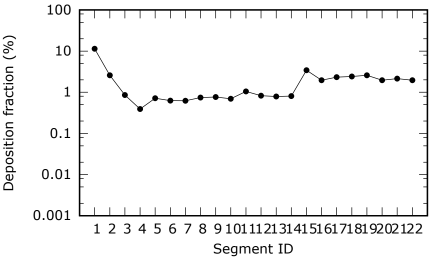


Figure 3.89: Deposition $d_p = 1$ µm.

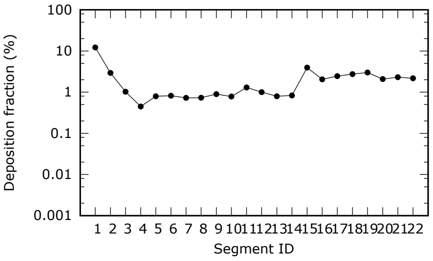


Figure 3.90: Deposition $d_p = 2$ µm.

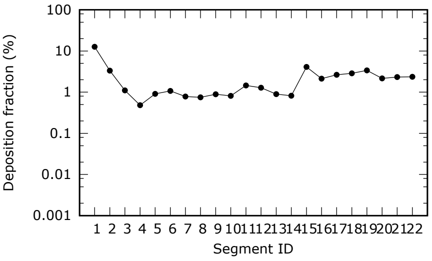


Figure 3.91: Deposition $d_p = 2.5$ µm.

be seen, in general there is a slight over-prediction of the overall deposition in all the segments, although the deposition pattern is very well captured. As can be noticed, heavier particles are mainly deposited in the mouth and the throat (segment 1), while lighter particles tend to be deposited in the tracheobronchial region. This is due to the main deposition mechanism for heavier particles is inertial deposition, and these type of particles are deposited by impaction at the mouth and the throat. On the other hand, lighter particles, with small Stokes number, are basically deposited by diffusion along the trachea and the bronchial airways, since these particles follow more easily the flow pattern. This deposition pattern is better seen in Fig. 3.98, where the deposition fraction is shown as function of particle size in the entire geometry. As can be seen, deposition increases with particle size, which suggests that inertial impaction is the dominant deposition mechanism in the upper airways. The decrease in the tracheobronchial deposition for particles with a diameter bigger than $d_p = 6$ µm is a result of the significant filtering that occurs upstream in the mouth-throat region.

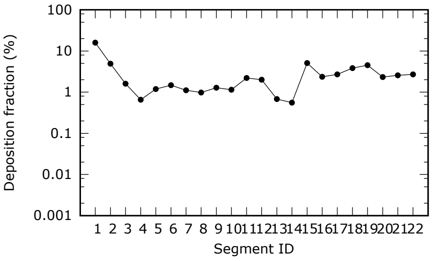


Figure 3.92: Deposition $d_p = 4$ µm.

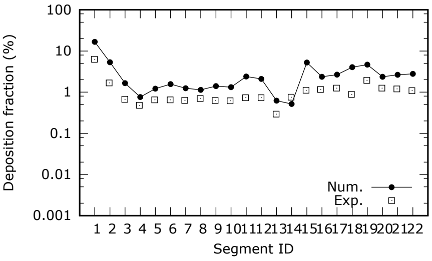


Figure 3.93: Deposition $d_p = 4.3$ µm.

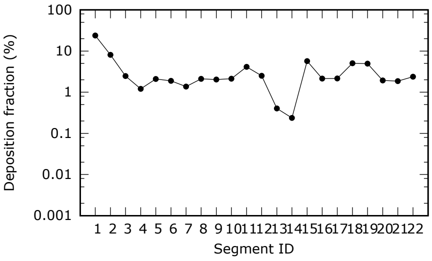


Figure 3.94: Deposition $d_p = 6$ µm.

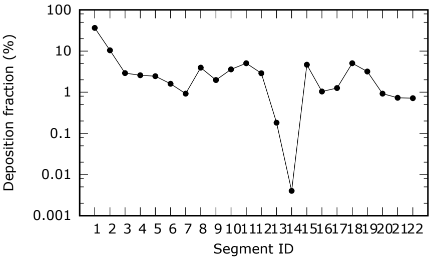


Figure 3.95: Deposition $d_p = 8$ µm.

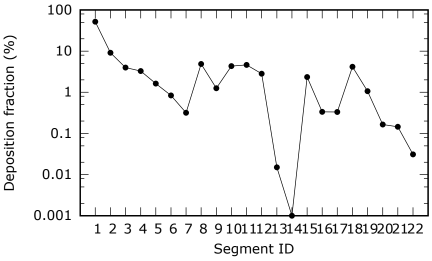


Figure 3.96: Deposition $d_p = 10$ µm.

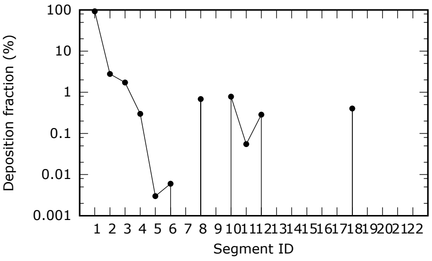


Figure 3.97: Deposition $d_p = 20$ µm.

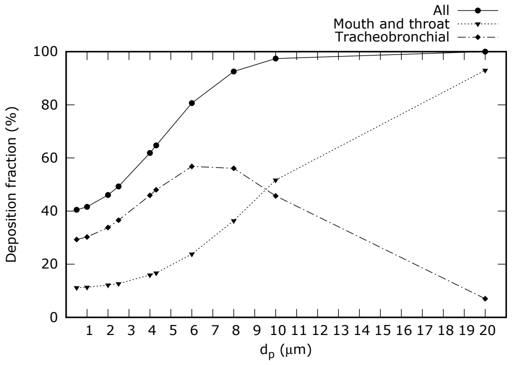


Figure 3.98: Deposition fraction as function of particle size.

3.7 Conclusions

In the present chapter, all the mathematical models and numerical algorithms required to build the numerical library of TermoFluids [92] devoted to perform complex numerical simulations of dispersed multi-phase flows has been detailed and presented. Throughout the chapter, several aspects have been covered, from the very basics like the assumptions and mathematical formulation of the developed numerical tool, to more advanced topics like the development of a new particle sub-grid dispersion model in LES modelling. Some of these aspects like the time-integration strategy or the interpolation method, that have a big impact in the accuracy and overall performance of the model, have been carefully studied.

It has been shown that when integrating the velocity and position equations of particles' trajectory, if an explicit scheme is employed for the velocity equation, an explicit scheme should be also employed for the position equation, although the velocity in the new time-step is ready-available. Moreover, the accuracy of various time-integration schemes of 1^{st} and 2^{nd} order have been compared using both constant time-step and variable time-step. As expected, 2^{nd} order schemes present a better accuracy, although special careful must be placed when dealing with variable time-step, since high-order linear multi-steps methods are derived assuming a constant time-step, and when they are employed in cases with a variable time-step their accuracy is diminished. It is shown that Runge-Kutta schemes do not present this downside, although at the expense of a higher computational cost.

Since the numerical library for dispersed multi-phase flows developed in this thesis is aimed to work in non-structured meshes, interpolation methods for scattered data must be employed. When dealing with cartesian grids it is relatively easy to obtain interpolation methods with a high-order of accuracy, and several options are available in the literature [48]. On the other hand, for unstructured meshes, it is difficult to construct general high-order accurate interpolation methods at a reasonable computational cost. The correct determination of the continuous phase properties at particles' location is a crucial aspect in Lagrangian-Eulerian simulations, and hence, both accuracy and computational cost of the interpolation method employed are a key aspect. Therefore, a detailed study of different interpolation methods for scattered data has been presented in this chapter. It is shown that the Pseudo-laplacian method is able to retrieve 2^{nd} order accuracy in both structured and unstructured meshes, while the Shepard method is only 1^{st} order accurate. Moreover, it is also shown that when using a 1^{st} order Taylor interpolation method, the method employed to compute the gradient has a direct effect in the accuracy of the method. Between the different gradient calculations methods analysed, the one offering a better accuracy in all the situations is the least-squares method with vertex connectivity. Regarding the computational cost of the interpolation method, the Taylor method is the one with the smallest computational cost if the cost of calculating the gradient is not considered. Hence, if the gradient can be considered *free*, i.e., is already calculated for other purposes or requirements, the Taylor method is the method computationally most efficient. If this is not the case, the method presenting the smallest computational cost will depend on the number of particles and the size of the mesh. Between the other studied methods, the one presenting a lower computational load is the Shepard method, followed by the Pseudo-laplacian method. The one with the highest cost is the RBF method.

Another important aspect addressed in this chapter is the influence of the sub-grid scale effects over the dispersed phase in LES modelling. In LES only the large scales of the flow are resolved, while the smallest ones are modelled. Hence, the effect of these unresolved scales over the Lagrangian particles, which interaction with the continuous phase is modelled at sub-grid scale, can be relevant in order to correctly predict the behaviour of the dispersed phase. Therefore, in this thesis the behaviour of Lagrangian particles in an isotropic turbulent flow has been studied and characterized, as well as the impact of the filtering procedure on the velocity of the continuous phase over the statistics for position and velocity of the dispersed phase. Furthermore, the performance and behaviour of two different stochastic models designed to reconstruct the influence of the sub-grid velocities over Lagrangian particles lost by filtering in LES have been examined. Both studied stochastic models seem well-suited to reconstruct particle-kinetic energy for high-inertia particles, although the models have a model constant that is case-dependant and should be properly setted. For small-inertia particles, these models introduce a stirring effect that dissipates the particle concentration effect. Both models have a similar computational cost.

Seeing the shortcomings and deficiencies of the studied stochastic methods, the development of a new sub-grid dispersion model without this downsides is intended.

This new proposed model is based on the reconstruction of the sub-grid scales lost due to LES filtering using PDFs. The idea is to employ the PDFs in order to recover statistical information of the sub-grid scales from magnitudes ready available in LES. The new model has been tested in a homogeneous isotropic turbulent case for particles with $\mathrm{St} = 0.5$. The obtained preliminary results are promising and clearly show the potential of this new method. The model recovers very well the kinetic energy of the particles lost with LES filtering, almost matching DNS values. Furthermore, it is able to preserve more flow scales than stochastic models, better conserving the preferential concentration effect. Moreover, it is shown that if the sub-grid time scale of the residual motions is properly determined, this method is able to present a closer particle concentration effect to the DNS case than a LES case without sub-grid dispersion model, which is a clear improvement respecting the current sub-grid dispersion models. This new model is still under development and further research must be carried out, but these preliminary results are very promising and encouraging.

Finally, two real-test cases have been simulated employing the numerical models and algorithms presented throughout the chapter.

The first case is the well-known flow loop Hercule of Borée et al. [75], where an axisymmetric confined bluff body flow is generated and solid particles with different sizes are injected through the inner pipe. The simulated experiment has a mass loading ratio of $M_i = 22\%$, allowing to assume one-way coupling. The results for the carrier phase velocity agree well with the experimental results, although the mean velocities are slightly over-predicted and the stagnation points are not perfectly matched. Nonetheless, these small differences can be attributed to the boundary conditions, since the flow configuration of this case is extremely sensitive to the inlet boundary conditions [82], and is very difficult reproduce the exact inlet conditions of the experiment. Regarding the results of the dispersed phase, the mean velocity is well captured, although the results are influenced by the small over-prediction in the velocity of the carrier phase. The simulations using sub-grid stochastic models for the velocity of the dispersed phase show that these models increase the kinetic energy of the particles. Other interesting physical effects of the dispersed phase like particle clustering and particle concentration effect are well-captured by the simulations.

The second benchmark case simulated in this thesis is an *in vitro* experiment designed to study the regional aerosol deposition of inhaled medicines in the respiratory airways, and which reproduces realistic human airways. The simulation presented in this chapter is part of a wider study carried out by different research groups in the framework of the SimInhale COST Action MP1404 [84]. The objective of the study is to present a benchmark case designed to serve as a reference for quality assurance of computational models in the upper airways. In order to do so, in the collaborative work [85] there are provided the deposition measurements obtained in an *in vitro* experiment reproducing a complex realistic geometry, as well as the results obtained by six different numerical approaches simulating the experiment. The comparison of the different simulations shows that the prediction of the flow in the upper ways is sensitive to mesh size and the turbulence model, but that the influence of the inflow condition is very low downstream the throat. Regarding the particle deposition, it is shown that this is strongly sensitive to mesh size and particle-tracking scheme, specially for the lightest particle with a small Stokes number. For these kind of particles, very sensitive to the flow field, it is a key aspect a good resolution of the near-wall flow as well as an accurate interpolation of the flow field velocity at particle location, showing that a very fine mesh is required to properly capture the deposition pattern of small particles. The results presented in the current chapter show a slight over-prediction of the particle deposition, basically due to that the mesh resolution near the wall is not fine enough. Nonetheless, the obtained results show that CFD simulations using a Lagrangian-Eulerian approach are a reliable and powerful tool in order to provide detailed information about regional deposition of inhaled aerosols in the human airways.

References

[1] Shankar Subramaniam. Lagrangian-Eulerian methods for multiphase flows. *Progress in Energy and Combustion Science*, 39(2):215–245, 2013.

[2] Swedish Industrial Association for Multiphase Flows (SIAMUF) and Martin Sommerfeld. *Best Practice Guidelines for Computational Fluid Dynamics of Dispersed Multi-Phase Flows.* European Research Community on Flow, Turbulence and Combustion (ERCOFTAC), 2008.

[3] Rodney O. Fox. Quadrature-Based Moment Methods for Fluid-Particle Flows. Department of Chemical and Biological Engineering, Iowa State University, US–DOE Ames Laboratory, February 2017.

[4] Rodney O. Fox. Large-Eddy-Simulation Tools for Multiphase Flows. *Annual Review of Fluid Mechanics*, 44(1):47–76, 2012.

[5] Olivier Desjardins, Rodney O Fox, and Philippe Villedieu. A quadrature-based moment method for dilute fluid-particle flows. *Journal of Computational Physics*, 227(4):2514–2539, 2008.

[6] A. B. Basset. On the Motion of a Sphere in a Viscous Liquid. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 179:43–63, 1888.

[7] Joseph Boussinesq. Sur la résistance qu'oppose un liquide indéfini en repos. *CR Acad. Sci. Paris*, 100:935–937, 1885.

[8] Carl Wilhelm Oseen. Hydrodynamik. *Akademische Verlag, Leipzig*, 1927.

[9] M. R. Maxey and J. J. Riley. Equation of motion for a small rigid sphere in a nonuniform flow. *Physics of Fluids*, 26:883–889, April 1983.

[10] George Gabriel Stokes. *On the effect of the internal friction of fluids on the motion of pendulums*, volume 9. Pitt Press, 1851.

[11] L. Schiller and Z. Naumann. A drag coefficient correlation. *Z. Ver. Deutsch. Ing*, 77(1):318–320, 1935.

[12] M. C. Yuen and L. W. Chen. On Drag of Evaporating Liquid Droplets. *Combustion Science and Technology*, 14(4-6):147–154, 1976.

[13] Phillip P Brown and Desmond F Lawler. Sphere drag and settling velocity revisited. *Journal of environmental engineering*, 129(3):222–231, 2003.

[14] Marian Zastawny, George Mallouppas, Fan Zhao, and Berend van Wachem. Derivation of drag and lift force and torque coefficients for non-spherical particles in flows. *International Journal of Multiphase Flow*, 39:227–239, 2012.

[15] Fuat Odar and Wallis S. Hamilton. Forces on a sphere accelerating in a viscous fluid. *Journal of Fluid Mechanics*, 18(2):302–314, 1964.

[16] Vincenzo Armenio and Virgilio Fiorotto. The importance of the forces acting on particles in turbulent flows. *Physics of Fluids*, 13(8):2437–2440, 2001.

[17] JGM Kuerten. Point-Particle DNS and LES of Particle-Laden Turbulent flow-a state-of-the-art review. *Flow, Turbulence and Combustion*, 97(3):689–713, 2016.

[18] Warren H Finlay. *The mechanics of inhaled pharmaceutical aerosols: an introduction.* Academic Press, 2001.

[19] Clayton T. Crowe, John D. Schwarzkopf, Martin Sommerfeld, and Yutaka Tsuji. *Multiphase flows with droplets and particles.* CRC press, 2011.

[20] SA Schaaf and PL Chambré. *Flow of Rarefied Gases.* Princeton University Press, 1961.

[21] C N Davies. Definitive equations for the fluid resistance of spheres. *Proceedings of the Physical Society*, 57(4):259, 1945.

[22] E Loth. Compressibility and rarefaction effects on drag of a spherical particle. *AIAA journal*, 46(9):2219–2228, 2008.

[23] Amy Li and Goodarz Ahmadi. Dispersion and deposition of spherical particles from point sources in a turbulent channel flow. *Aerosol science and technology*, 16(4):209–226, 1992.

[24] GAE Godsave. Studies of the combustion of drops in a fuel spray—the burning of single drops of fuel. In *Symposium (International) on Combustion*, volume 4, pages 818–830. Elsevier, 1953.

[25] Dudley Brian Spalding. The combustion of liquid fuels. In *Symposium (international) on combustion*, volume 4, pages 847–864. Elsevier, 1953.

[26] B Abramzon and WA Sirignano. Droplet vaporization model for spray combustion calculations. *International journal of heat and mass transfer*, 32(9):1605–1618, 1989.

[27] J Bellan and K Harstad. Analysis of the convective evaporation of nondilute clusters of drops. *International journal of heat and mass transfer*, 30(1):125–136, 1987.

[28] SS Sazhin, T Kristyadi, WA Abdelghaffar, and MR Heikal. Models for fuel droplet heating and evaporation: comparative analysis. *Fuel*, 85(12):1613–1630, 2006.

[29] SS Sazhin, A Elwardany, PA Krutitskii, G Castanet, F Lemoine, EM Sazhina, and MR Heikal. A simplified model for bi-component droplet heating and evaporation. *International Journal of Heat and Mass Transfer*, 53(21):4495–4505, 2010.

[30] R.S. Miller, K. Harstad, and J. Bellan. Evaluation of equilibrium and non-equilibrium evaporation models for many-droplet gas-liquid flow simulations. *International Journal of Multiphase Flow*, 24(6):1025 – 1055, 1998.

[31] WE Ranz and WR Marshall. Evaporation from drops. *Chemical Engineering Progress*, 48(3):141446, 1952.

[32] CK Law and FA Williams. Kinetics and convection in the combustion of alkane droplets. *Combustion and Flame*, 19(3):393–405, 1972.

[33] E. Knudsen Shashank and H. Pitsch. Spray evaporation model sensitivities. Annual Research Briefs of the CTR, 2011.

[34] Edward Knudsen, Heinz Pitsch, et al. Modeling partially premixed combustion behavior in multiphase LES. *Combustion and Flame*, 162(1):159–180, 2015.

[35] Carl G Downingm. The evaporation of drops of pure liquids at elevated temperatures: Rates of evaporation and wet-bulb temperatures. *AIChE Journal*, 12(4):760–766, 1966.

[36] Shwin-Chung Wong and Ar-Cheng Lin. Internal temperature distributions of droplets vaporizing in high-temperature convective flows. *Journal of fluid mechanics*, 237:671–687, 1992.

[37] C Maqua, G Castanet, F Grisch, F Lemoine, T Kristyadi, and SS Sazhin. Monodisperse droplet heating and evaporation: experimental study and modelling. *International Journal of Heat and Mass Transfer*, 51(15):3932–3945, 2008.

[38] Sourabh V Apte, Krishnan Mahesh, and Parviz Moin. Large-eddy simulation of evaporating spray in a coaxial combustor. *Proceedings of the Combustion Institute*, 32(2):2247–2256, 2009.

[39] Mouldi Chrigui, James Gounder, Amsini Sadiki, Assaad R Masri, and Johannes Janicka. Partially premixed reacting acetone spray using LES and FGM tabulated chemistry. *Combustion and flame*, 159(8):2718–2741, 2012.

[40] D Mazumdar and RIL Guthrie. An analysis of numerical methods for solving the particle trajectory equation. *Applied mathematical modelling*, 12(4):398–402, 1988.

[41] O. Lehmkuhl, C.D. Pérez Segarra, R. Borrell, M. Soria, and A. Oliva. Termofluids: A new Parallel unstructured CFD code for the simulation of turbulent industrial problems on low cost PC cluster. *Proceedings of the Parallel CFD Conference*, pages 1–8, 2007.

[42] F.X. Trias and O. Lehmkuhl. A Self-Adaptive Strategy for the Time Integration of Navier–Stokes Equations. *Numerical Heat Transfer, Part B: Fundamentals*, 60(2):1116–134, 2011.

[43] S Elghobashi. On predicting particle-laden turbulent flows. *Applied scientific research*, 52(4):309–329, 1994.

[44] Said Elgobashi. An updated classification map of particle-laden turbulent flows. *Fluid Mechanics and Its Applications*, 81:3, 2006.

[45] F Bianco, Sergio Chibbaro, Cristian Marchioli, MV Salvetti, and Alfredo Soldati. Intrinsic filtering errors of Lagrangian particle tracking in LES flow fields. *Physics of Fluids*, 24(4):045103, 2012.

[46] WR Michałek, Johannes GM Kuerten, JCH Zeegers, R Liew, J Pozorski, and Bernardus J Geurts. A hybrid stochastic-deconvolution model for large-eddy simulation of particle-laden flow. *Physics of fluids*, 25(12):123302, 2013.

[47] Richard S Miller and Josette Bellan. Direct numerical simulation and subgrid analysis of a transitional droplet laden mixing layer. *Physics of Fluids*, 12(3):650–671, 2000.

[48] William H. Press. *Numerical recipes 3rd edition: The art of scientific computing.* Cambridge University Press, 2007.

[49] O. Estruch, O. Lehmkuhl, R. Borrell, C.D. Pérez Segarra, and A. Oliva. A parallel radial basis function interpolation method for unstructured dynamic meshes. *Computers & Fluids*, 80:44–54, 2013.

[50] DG Holmes and SD Connell. *Solution of the 2D Navier-Stokes equations on unstructured adaptive grids.* American Institute of Aeronautics and Astronautics, 1989.

[51] CD Perez-Segarra, C Farre, J Cadafalch, and A Oliva. Analysis of different numerical schemes for the resolution of convection-diffusion equations using finite-volume methods on three-dimensional unstructured grids. part i: Discretization schemes. *Numerical Heat Transfer, Part B: Fundamentals*, 49(4):333–350, 2006.

[52] M. Chrigui, J. Gounder, A. Sadiki, A.R. Masri, and J. Janicka. Large Eddy Simulation of a Polydisperse Ethanol Spray Flame. *Flow, Turbulence and combustion*, 90:813–832, 2013.

[53] JGM Kuerten. Point-Particle DNS and LES of Particle-Laden Turbulent flow-a state-of-the-art review. *Flow, turbulence and combustion*, 97(3):689–713, 2016.

[54] B. Shotorban and F. Mashayek. Modeling subgrid-scale effects on particles by approximate deconvolution. *Physics of Fluids*, 17:081701, 2005.

[55] J.G.M. Kuerten. Subgrid modeling in particle-laden channel flow. *Physics of Fluids*, 18:025108, 2006.

[56] J.P. Minier, S. Chibbaro, and S.B. Pope. Guidelines for the formulation of Lagrangian stochastic models for particle simulations of single-phase and dispersed two-phase turbulent flows. *Physics of Fluids*, 26:113303, 2014.

[57] J.P. Minier. On lagrangian stochastic methods for turbulent polydisperse two-phase reactive flows. *Progress in Energy and Combustion Science*, 50:1–62, 2015.

[58] Jacek Pozorski and Sourabh V Apte. Filtered particle tracking in isotropic turbulence and stochastic modeling of subgrid-scale dispersion. *International Journal of Multiphase Flow*, 35(2):118–128, 2009.

[59] I Pesmazoglou, AM Kempf, and S Navarro-Martinez. A dynamic model for the Lagrangian stochastic dispersion coefficient. *Physics of Fluids*, 25(12):125108, 2013.

[60] Cristian Marchioli, Maria Vittoria Salvetti, and Alfredo Soldati. Appraisal of energy recovering sub-grid scale models for large-eddy simulation of turbulent dispersed flows. *Acta mechanica*, 201(1-4):277–296, 2008.

[61] Ch Gobert. Analytical assessment of models for large eddy simulation of particle laden flow. *Journal of Turbulence*, 11:N23, 2010.

[62] T.S. Lundgren. Linearly forced isotropic turbulence. In *Annual Research Briefs*, Center for Turbulence Research (CTR), Stanford, 2003.

[63] R. Borrell, O. Lehmkuhl, F.X. Trias, and A. Oliva. Parallel direct poisson solver for discretisations with one fourier diagonalisable direction. *Journal of Computational Physics*, 12:4723–4741, 2011.

[64] E.W. Saw, J.P.L.C. Salazar, L.R. Collins, and R.A. Shaw. Spatial clustering of polydisperse inertial particles in turbulence: I. comparing simulation with theory. *New Journal of Physics*, 14:105030, 2012.

[65] J. Pozorski and S.V. Apte. Filtered particle tracking in isotropic turbulence and stochastic modeling. *International Journal of Multiphase Flow*, 35:118–128, 2009.

[66] M. Bini and W.P. Jones. Particle acceleration in turbulent flows: A class of nonlinear stochastic models for intermittency. *Physics of Fluids*, 19:035104, 2007.

[67] M. Bini and W.P. Jones. Large-eddy simulation of particle-laden turbulent flows. *Journal of Fluid Mechanics*, 614:207–252, 2008.

[68] CW Gardiner. Handbook of stochastic methods for physics, chemistry and the natural sciences. *Applied Optics*, 25:3145, 1986.

[69] Jacek Pozorski and Jean-Pierre Minier. Probability density function modeling of dispersed two-phase turbulent flows. *Physical Review E*, 59(1):855, 1999.

[70] L.P. Wang and M.R. Maxey. Settling velocity and concentration distribution of heavy particles in homogeneous isotropic turbulence. *Journal of Fluid Mechanics*, 256:27–68, 1993.

[71] G. Jin and G.W He. A nonlinear model for the subgrid timescale experienced by heavy particles in large eddy simulation of isotropic turbulence with a stochastic differential equation. *New Journal of Physics*, 15(3):035011, 2013.

[72] George I Park, J Urzay, M Bassenne, and P Moin. A dynamic subgrid-scale model based on differential filters for LES of particle-laden turbulent flows. *Annual Research Briefs, Center for Turbulence Research, Stanford University*, pages 17–26, 2015.

[73] M Germano. Differential filters for the large eddy numerical simulation of turbulent flows. *The Physics of fluids*, 29(6):1755–1757, 1986.

[74] M Germano. Differential filters of elliptic type. *The Physics of fluids*, 29(6):1757–1758, 1986.

[75] J Borée, T Ishima, and I Flour. The effect of mass loading and inter-particle collisions on the development of the polydispersed two-phase flow downstream of a confined bluff body. *Journal of Fluid Mechanics*, 443:129–165, 2001.

[76] Jean-Pierre Minier. On Lagrangian stochastic methods for turbulent polydisperse two-phase reactive flows. *Progress in Energy and Combustion Science*, 50:1–62, 2015.

[77] Lars Davidson. Large-Eddy Simulation: A Dynamic One-Equation Sub-grid Model for Three-Dimensional Recirculating Flow. In *11th International Symposium on Turbulent Shear Flow*, volume 3, pages 26–1, 1997.

[78] Sinisa Krajnovic and Lars Davidson. Large-eddy simulation of the flow around a surface-mounted cube using a dynamic one-equation subgrid model. In *TSFP DIGITAL LIBRARY ONLINE*. Begel House Inc., 1999.

[79] Ahmad Sohankar, Lars Davidson, and Christoffer Norberg. Large eddy simulation of flow past a square cylinder: comparison of different subgrid scale models. *Journal of Fluids Engineering*, 122(1):39–47, 2000.

[80] Siniša Krajnovic and Lars Davidson. Large-eddy simulation of the flow around a bluff body. *AIAA journal*, 40(5):927–936, 2002.

[81] M Klein, A Sadiki, and J Janicka. A digital filter based generation of inflow data for spatially developing direct numerical or large eddy simulations. *Journal of computational Physics*, 186(2):652–665, 2003.

[82] Eleonore Riber, Vincent Moureau, Martha García, Thierry Poinsot, and Olivier Simonin. Evaluation of numerical strategies for large eddy simulation of particulate two-phase recirculating flows. *Journal of Computational Physics*, 228(2):539–564, 2009.

[83] A. Nokhodchi and G.P. Martin. *Pulmonary Drug Delivery: Advances and Challenges*. Advances in Pharmaceutical Technology. Wiley, 2015.

[84] Siminhale COST Action MP1404. http://www.siminhale-cost.eu/. Accessed: 21/12/2017.

[85] P. Koullapis, S.C. Kassinos, J. Muela, C. Perez-Segarra, J. Rigola, O. Lehmkuhl, Y. Cui, M. Sommerfeld, J. Elcner, M. Jicha, I. Saveljic, N. Filipovic, F. Lizal, and L. Nicolaou. Regional aerosol deposition in the human airways: The Siminhale benchmark case and a critical assessment of in silico methods. *European Journal of Pharmaceutical Sciences*, pages –, 2017.

[86] Andreas Schmidt, Stephan Zidowitz, Andres Kriete, Thorsten Denhard, Stefan Krass, and Heinz-Otto Peitgen. A digital reference model of the human bronchial tree. *Computerized Medical Imaging and Graphics*, 28(4):203–211, 2004.

[87] K.H. Cheng, Y.S. Cheng, H.C. Yeh, and D.L. Swift. Measurements of airway dimensions and calculation of mass transfer characteristics of the human oral passage. *Journal of biomechanical engineering*, 119(4):476–482, 1997.

[88] Frantisek Lizal, Jakub Elcner, Philip K Hopke, Jan Jedelsky, and Miroslav Jicha. Development of a realistic human airway model. *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine*, 226(3):197–207, 2012.

[89] Frantisek Lizal, Miloslav Belka, Jan Adam, Jan Jedelsky, and Miroslav Jicha. A method for in vitro regional aerosol deposition measurement in a model of the human tracheobronchial tree by the positron emission tomography. *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine*, 229(10):750–757, 2015.

[90] Thomas JR Hughes, Luca Mazzei, and Kenneth E Jansen. Large eddy simulation and the variational multiscale method. *Computing and Visualization in Science*, 3(1):47–59, 2000.

[91] C. Kleinstreuer and Z. Zhang. Airflow and particle transport in the human respiratory system. *Annual Review of Fluid Mechanics*, 42:301–334, 2010.

[92] O. Lehmkuhl, C.D. Perez-Segarra, R. Borrell, M. Soria, and A. Oliva. TERMOFLUIDS: A new Parallel unstructured CFD code for the simulation of turbulent industrial problems on low cost PC Cluster. *Parallel Computational Fluid Dynamics 2007*, pages 275–282, 2009.

4

Sliding Meshes

Main contents of this chapter have been published at international journals:

J. Muela, D. Martínez, O. Lehmkuhl, C.D. Pérez-Segarra and A. Oliva. *New parallel method for adjacent disconnected unstructured 3D meshes.* International Journal of Computational Fluid Dynamics Vol. 30(6), pp. 388-394, 2016.

and presented at international conferences:

J. Muela, D. Martínez, O. Lehmkuhl, C.D. Pérez-Segarra and A. Oliva. *New parallel method for adjacent disconnected unstructured 3D meshes.* In Proceedings of the *27th International Conference on Parallel Computational Fluid Dynamics 2015*, Montreal, Quebec, Canada, 17-20 May 2015.

Abstract. A new parallel method for Computational Fluid Dynamic simulations with non-overlapping disconnected mesh domains but adjacent boundaries is presented and studied. The developed algorithm generates dynamically during the simulation time the required links and connections between the adjacent boundaries of the meshes, allowing to perform numerical simulations using multiple 3D unstructured meshes that are independent. Moreover, since this methodology is able to generate the connection between the meshes at each iteration, this allows simulations where one or more of the independent meshes are moving. Hence, the developed methodology is specially well-suited for technological applications having both static and moving parts like fans, wind turbines, axial compressors, turbines and all kind of turbomachinery.

The methodology uses explicit convective and diffusive operators. For the current implementation these operators are interpolated in the neighbouring mesh and the information is sent to the boundary node. For the pressure, that is implicit, the topology of the Poisson matrix is rebuilt at each iteration in function of the neighbouring connectivity.

4.1 Introduction

In Computational Fluid Dynamics (CFD), an Eulerian framework is traditionally employed since the majority of the problems of interest are studied in static volumes of the space. But this traditional approximation may not be the best option for some specific cases like for example wind turbines, tidal-stream turbines or other rotary machines. Consider a wind turbine: it would be useful to work with a mesh *attached* to the rotating blades (Eulerian-Lagrangian framework), but also a static mesh would be desirable for the tower, the nacelle, and the ground (Eulerian framework). This duality caused by the presence of a moving part close to a static one makes impossible CFD simulations using as reference frame the rotating turbine rotor. Therefore, it is of clear interest to develop a method able to perform simulations where a mesh domain Ω_a is moving with respect to a second static mesh domain Ω_b. This method will allow to link a moving mesh including the rotating rotor and the blades and a static mesh containing the statics parts.

There are two main types of methods aimed to perform this kind of simulations: the Chimera technique and the sliding mesh technique. The former was originally developed by Stager et al. [1] while the latter is mainly based on the works of Rai [2, 3]. In the Chimera method there is an overlap between the static and the moving meshes and the communication between both meshes is done through this overlapped areas. Each mesh interpolates in its nodes located inside this overlapped area the required values from the values stored in the nodes of the neighbour cell. On the other hand, in the sliding mesh technique both domains are not overlapped $\Omega_a \cap \Omega_b = 0$ but they are adjacent and share some or all boundaries $\partial\Omega_{a_i} = \partial\Omega_{b_i}$. Most of the methods based on the sliding technique make the communication of the info between the meshes using a computation of fluxes through the sliding interface.

An example of these flux-based methods is the Generalized Grid Interface (CGI) method, developed by Beaudoin and Jasak [4] implemented in OpenFOAM [5]. This method is based on the projection of one of the boundary faces over the other one, obtaining some weighting factors which are basically the percentage of surface intersection between the different faces in the sliding boundary of the two meshes. Then, these weighting factors are employed to calculate the fluxes from a mesh to the other. The CGI method allows the computation of fluxes through non-conforming meshes, but although being flux-based, it is not fully-conservative. For example, in circular boundaries like the one represented in Fig. 4.1, the areas of the faces do not overlap and at each iteration the area of the projected surface will be slightly different. In [4] a rescaling of the weighting factors is proposed in order to mitigate this issue, although it cannot be completely eliminated since is inevitable due to the mesh geometry. Another sliding mesh technique is the *attach-detach* method. In this method the topology of the cells at the interface is reconstructed at each iteration. This reconstruction of the faces at each iteration eliminates the gap existing between the two

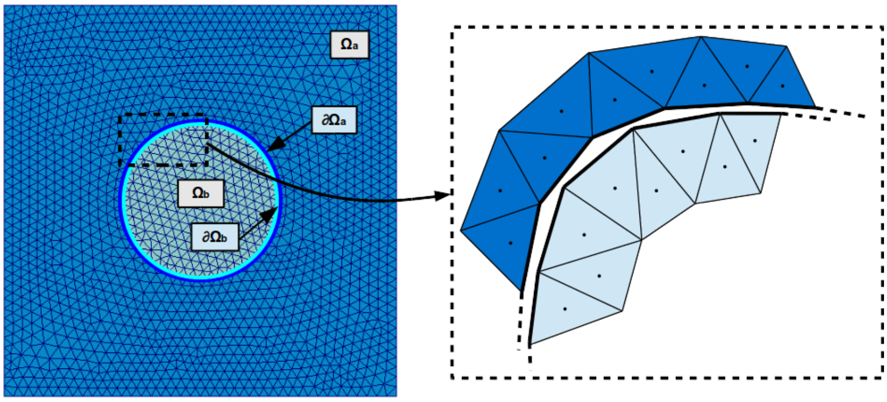


Figure 4.1: Example of a domain composed by a static mesh Ω_a plus a rotary mesh Ω_b and a detail of the sliding boundary between them.

meshes, avoiding the discrepancies in the surfaces' areas found in the CGI method. Nonetheless, this reconstruction of the mesh topology implies a modification of the cells' volume, making the method non-conservative too. Obviously, the CGI method is faster than the *attach-detach* method. These two method were compared by Petit et al. [6] for the study of a centrifugal pump and both methods offered very close results.

One more sliding mesh approach is the Shear-Slip Mesh Update Method (SSMUM) presented by Behr and Tezduyar [7, 8]. In this method the slipping boundary is handled letting a thin layer of mesh elements to undergo deformation and performing frequent re-meshing steps via regeneration of element connectivity before the deformation becomes too big. This re-meshing avoids de surface projection of other sliding mesh methods.

In order to overcome the issue of the surface projection discrepancies in non-conformal meshes for rotational applications using finite element methods (FEM), Bazilevs et al. proposed to employ Non-Uniform Rational B-Spline (NURBS)-based isogeometric analysis for the discretization [9]. This discretization allows to exactly represent circular and cylindrical surfaces, in contrast to standard finite elements discretizations, resulting in a very attractive method for the analysis of rotating machines. In this method, the interface between the static and the rotating mesh is unique and it remains circular (or cylindrical) at all time-steps. This is possible because the interface is exactly represented by NURBS functions of second or higher order. This method was successfully employed for simulations of full wind turbines by Hsu et al. using FEM [10, 11].

A method that also falls inside the category of sliding mesh technique is the one developed by McNaughton et al. [12], although oppositely to the previous presented methods, this one is not flux-based. The basic idea of the method developed by McNaughton et al. is to treat the sliding interface as an internal boundary where the values are imposed using a Dirichlet-type boundary condition. This values set in the boundary nodes are obtained from interpolation of the variables in the boundary cells of the neighbour mesh, but in contrast to halo methods this one do not use overlap cells.

A comparison between both methods, the Chimera and the Sliding Mesh techniques, for a counter rotating open-rotor was carried out by B. Francois [13]. It was found that the results obtained with the two methods were very close, and moreover, both were able to generate reliable and accurate numerical results.

As can be seen, several methods aimed to handle problems with moving boundaries and interfaces have been developed. Each method presents its owns pros and cons, but all of them have been successfully employed for certain applications. Te current chapter of this thesis is aimed to develop and implement a new method able to perform numerical simulations using two or more non-overlapping meshes with relative motion between them. The main requirements set for the development of this new algorithm are: accuracy, reliability, generality, performance and parallel scalability. Hence, the method should be able to produce accurate and reliable numerical results, having the lowest computational impact as possible, presenting a good scalability for parallel simulation and must be able to deal with all kind of meshes and geometries.

4.2 Methodology

The aim of the developed methodology is to *stitch* two independent meshes Ω_a and Ω_b that share a common boundary $\partial\Omega_{a_i} = \partial\Omega_{b_i}$. In order to do this, the implemented algorithm works in the following way: at the beginning of the simulation, each boundary cell of $\partial\Omega_{a_i}$ places a *mirror* node in mesh Ω_b, and vice versa. This mirror node is a lagrangian particle that is moved with the relative velocity between both meshes in order to preserve the relative position between the parent cell and the mirror node (see Fig. 4.2). This mirror node allows to easily interpolate the value of any scalar or gradient on the neighbouring domain and transfer the information to the boundary node. Furthermore, this mirror node is employed to reconstruct at each iteration the topology of the Poisson matrix.

4.2.1 Mathematical formulation

The Navier-Stokes equations for incompressible flows in an Arbitrary Lagrangian Eulerian (ALE) formulation are:

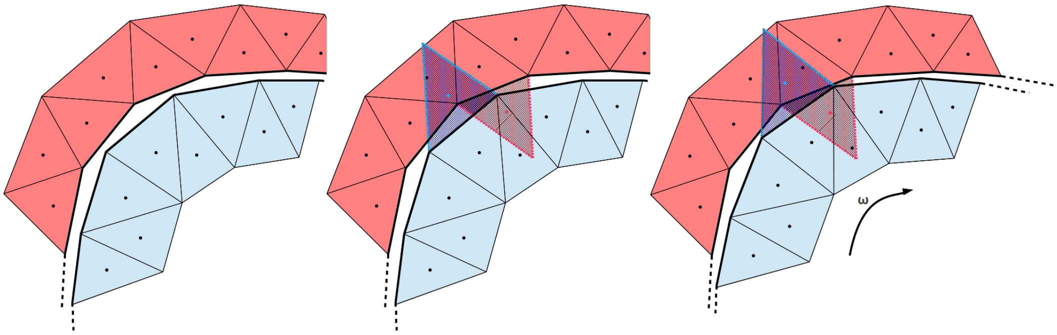


Figure 4.2: Schematic representation of a mirror node in the sliding boundary.

$$\nabla \cdot \mathbf{u} = 0 \tag{4.1}$$

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \left(\left(\mathbf{u} - \mathbf{v}_d \right) \mathbf{u} \right) = -\frac{1}{\rho} \nabla p + \nu \nabla^2 \mathbf{u} \tag{4.2}$$

where $\mathbf{u}$ is the velocity vector, ρ the density, p the pressure, ν the kinematic viscosity and $\mathbf{v}_d$ the displacement velocity of the domain with respect to the Eulerian reference framework. The Finite-Volume Method (FVM) is employed to discretize Eq. (4.1) and Eq. (4.2) on a general arbitrary mesh. Moreover, the velocity-pressure coupling is solved by means of the Fractional Step Method (FSM):

$$\mathbf{u}^p = \mathbf{u}^n + \Delta t \left[-\nabla \cdot \left(\mathbf{u}^n \mathbf{u}^n \right) + \nabla \cdot \left(\mathbf{v}_d^n \mathbf{u}^n \right) + \nu \nabla^2 \mathbf{u}^n \right] \tag{4.3}$$

$$\mathbf{u}^{n+1} = \mathbf{u}^p - \frac{\Delta t}{\rho} \nabla p^{n+1} \tag{4.4}$$

where the superscript n refers to the time instant, $\mathbf{u}^p$ is the predicted velocity and Δt the integration time-step. The convective and the diffusive terms are solved explicitly, while the pressure is solved implicitly in order to guarantee that $\mathbf{u}^{n+1}$ is divergence-free. Using a collocated mesh scheme, the predictor step equation is discretized integrating Eq. (4.3) over a control volume c and applying the divergence theorem to its faces:

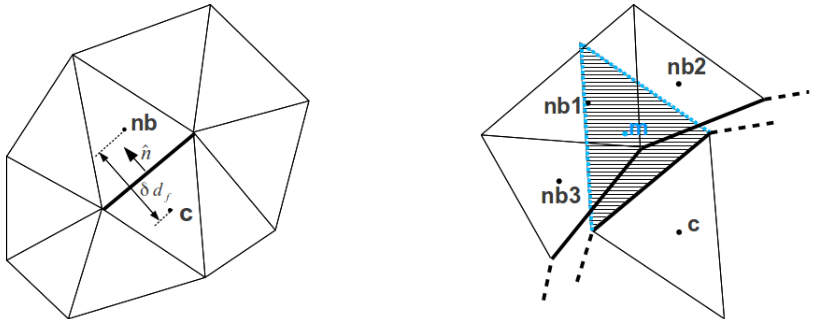


Figure 4.3: 2D Full (left) and Sliding (right) mesh detail.

$$\mathbf{u}_c^p = \mathbf{u}_c^n + \frac{\Delta t}{V_c}\left[- \sum_{f\in F(c)} \mathbf{u}_f^n \hat{U}_f^n A_f + \sum_{f\in F(c)} \mathbf{u}_f^n \hat{V}_{df}^n A_f \right. \\ \left. + \nu \sum_{f\in F(c)} \left(\mathbf{u}_{nb}^n - \mathbf{u}_c^n\right) \frac{A_f}{\delta d_f} \right] \tag{4.5}$$

where V_c is the volume of cell c, $\mathbf{u}_f$ is the velocity vector at the face f, $\hat{U}_f$ the normal face velocity, A_f the surface of f, the subscript nb refers to the face-neighbouring cells of c, δd_f is the distance between the cell centroids of c and nb projected onto the normal of the face f, and $\hat{V}_{df}$ is the normal face displacement velocity respect the reference frame. Applying the divergence operator $\nabla\cdot$ to Eq. (4.4) and imposing the divergence-free condition of Eq. (4.1) the discretized Poisson equation employed to solve the pressure reads:

$$\sum_{f\in F(c)} \hat{U}_f^p A_f = \frac{\Delta t}{\rho} \sum_{f\in F(c)} \left(p_{nb}^{n+1} - p_c^{n+1}\right) \frac{A_f}{\delta d_f} \tag{4.6}$$

Once the solution of p^{n+1} is obtained, the discrete form of Eq. (4.4) is used to calculate $\mathbf{u}^{n+1}$:

$$\mathbf{u}_c^{n+1} = \mathbf{u}_c^p - \frac{\Delta t}{\rho V_c} \sum_{f\in F(c)} p_f^{n+1} \hat{\mathbf{n}}_f A_f \tag{4.7}$$

where $\hat{n}_f$ is the unit normal vector of face f pointing outwards and p_f the pressure interpolated at the face f. The values for velocity and pressure at the face $(\mathbf{u}_f, \hat{U}_f, p_f)$

have to be interpolated. The face velocity is calculated using a symmetry-preserving scheme $\mathbf{u}_f = \frac{1}{2}\left(\mathbf{u}_c + \mathbf{u}_{nb}\right)$. The normal face velocity and the face pressure are calculated as $\hat{U}_f^n = \frac{1}{2}\left(\mathbf{u}_c + \mathbf{u}_{nb}\right) \cdot \hat{\mathbf{n}}_f$ and $p_f = \frac{1}{2}\left(p_c + p_{nb}\right)$ in order to minimize the kinetic energy conservation error [14]. The normal face displacement velocity is $\hat{V}_{df} = \mathbf{v}_{df} \cdot \hat{\mathbf{n}}_f$ (i.e. in rotating meshes $\mathbf{v}_{df} = \boldsymbol{\omega}_f \times (\mathbf{x}_f - \mathbf{x}_0)$, where $\boldsymbol{\omega}_f$ is the rotating velocity, $\mathbf{x}_f$ the face centroid and $\mathbf{x}_0$ the centre of rotation). It is obvious that the presented methodology cannot be applied straightforwardly in the sliding face, since as can be seen in Fig. 4.3, in the sliding face two *neighbouring* cells are not connected by a unique shared face f. As explained before, each cell with a face in the common boundary $\partial\Omega_{a_i} = \partial\Omega_{b_i}$ places a mirror node in the neighbour mesh. Each mirror node is assumed to belong to a cell that is a mirror of the parent cell. This assumption allows this face (f_b) to be treated as a pseudo-inner face between the cells c and m (see Fig. 4.3), which at the same time allows the calculation of the face values for velocity and pressure in a similar fashion to the inner faces. For the explicit operators, i.e. the convective and the diffusive operators, the value in the mirror node is interpolated using any interpolation scheme for unstructured data $\phi_m^n = f\left(\phi_{nb_1}^n, \phi_{nb_2}^n, ..., \phi_{nb_n}^n\right)$, and once the value ϕ_m^n has been calculated, the values for velocity and pressure in the sliding face can be calculated similarly to the inner faces via:

$$\mathbf{u}_f^n = \frac{1}{2}\left(\mathbf{u}_c^n + \mathbf{u}_m^n\right) \tag{4.8}$$

$$\hat{U}_f^n = \frac{1}{2}\left(\mathbf{u}_c^n + \mathbf{u}_m^n\right) \cdot \hat{\mathbf{n}}_f \tag{4.9}$$

$$p_f^{n+1} = \frac{1}{2}\left(p_c^{n+1} + p_m^{n+1}\right) \tag{4.10}$$

Nonetheless, this strategy cannot be employed for the implicit Poisson equation. In an unique static mesh, if Eq. (4.6) is applied to all the control volumes of the mesh, a system of type $[A]\mathbf{p^{n+1}} = \mathbf{b}$ is obtained, where matrix $[A]$ only depends on the geometry and therefore does not change during simulation time. On the other hand, when two (or more) non-overlapping independent meshes with relative motion are considered, as in the present case, this matrix $[A]$ is no longer constant, and it must be reconstructed at each iteration applying the Eq. (4.11) to all the faces in the sliding boundary. Basically, the main idea is that the neighbour cell of cell c is the cell where the mirror node is living in the current time step.

$$\frac{1}{2}\left(\mathbf{u}_c^p + \mathbf{u}_m^p\right) \cdot \hat{\mathbf{n}}_f A_f = \frac{\Delta t}{\rho V_c}\left(p_{nb1}^{n+1} - p_c^{n+1}\right)\frac{A_f}{\delta d_f} \tag{4.11}$$

4.2.2 Conservation analysis

The objective of the present section is to study and analyse the conservative properties of the implemented methodology. Since the discrete Navier-Stokes equations are derived from mass and momentum conservation equations, a proper discretization method should conserve mass and momentum. As demonstrated by Jofre et al. [14], when the symmetry-preserving scheme is employed for the convective scheme, momentum and mass are numerically conserved. However, the conservation of derived quantities like kinetic energy, vorticity or enstrophy are not obtained directly just by using a proper discretization. For example, as demonstrated by Veldman et al. [15], in order to preserve kinetic-energy the convective operator must be discretized using the Symmetry-Preserving (SP) scheme in order to obtain a discrete convective operator being skew-symmetric, as the continuous convective operator. When using a collocated mesh scheme as in the present formulation, appears a kinetic energy error of the form $\mathcal{O}(\Delta t^m, \Delta h^2)$ due to formulation of the pressure term. Nonetheless, the conservation of these quantities is important in order to achieve reliable physical results from numerical simulations.

Due to the nature of the methodology is of interest to analyse how the method affects the conservation of these quantities. Hence, following is analysed how mass and kinetic-energy are conserved as function of different parameters. In order to check mass conservation, the continuity equation is integrated over the whole domain:

$$\int_{\Omega} \nabla \cdot \mathbf{u} d\Omega = \sum_{c\in\Omega} \int_{V_c} \nabla \cdot \mathbf{u} dV_c = \sum_{c\in\Omega} \int_{S_c} \hat{\mathbf{n}} \cdot \mathbf{u} dS_c = \sum_{c\in\Omega} \sum_{f\in F(c)} \hat{V}_f A_f \tag{4.12}$$

The specific kinetic energy is defined as $\frac{1}{2}\mathbf{u}\cdot\mathbf{u}$ and its conservation equation is obtained multiplying momentum equation by the velocity $\mathbf{u}$. For an incompressible flow using ALE formulation it reads as:

$$\frac{\partial\left(\frac{1}{2}\mathbf{u}\cdot\mathbf{u}\right)}{\partial t} + \nabla \cdot \left((\mathbf{u} - \mathbf{v}_d) \left(\frac{1}{2}\mathbf{u}\cdot\mathbf{u}\right)\right) = -\frac{1}{\rho}\nabla\left(p\mathbf{u}\right) + \nu\nabla\cdot\left(\mathbf{u}\times\omega\right) - \nu\omega\cdot\omega \tag{4.13}$$

where $\omega = \nabla \times \mathbf{u}$ is the vorticity. Notice that Eq. (4.13) is conservative except for the last term in the right-hand side ($\nu\omega\cdot\omega$) which acts as a source/sink. Hence, if there are not external forces and there is no viscosity, the kinetic energy is only redistributed but not generated or dissipated. In order to study the conservation of the kinetic energy for the developed methodology, the discrete momentum equation is multiplied by the velocity vector $\mathbf{u}$ and integrated in the whole domain Ω, resulting in:

$$\begin{gathered}
\sum_{c\in\Omega}\mathbf{u}_c\cdot\frac{\partial\mathbf{u}_c}{\partial t}V_c+\sum_{c\in\Omega}\mathbf{u}_c\cdot\sum_{f\in F(c)}\mathbf{u}_f\hat{U}_fA_f-\sum_{c\in\Omega}\mathbf{u}_c\cdot\sum_{f\in F(c)}\mathbf{u}_f\hat{V}_{df}A_f= \\
-\frac{1}{\rho}\sum_{c\in\Omega}\mathbf{u}_c\cdot\sum_{f\in F(c)}p_f\hat{\mathbf{n}}_fA_f+\nu\sum_{c\in\Omega}\mathbf{u}_c\cdot\sum_{f\in F(c)}\left(\mathbf{u}_{nb}-\mathbf{u}_c\right)\frac{A_f}{\delta d_f}
\end{gathered}\tag{4.14}$$

The problem chosen to carry out the study is a spatially periodic case of vortices described by equations:

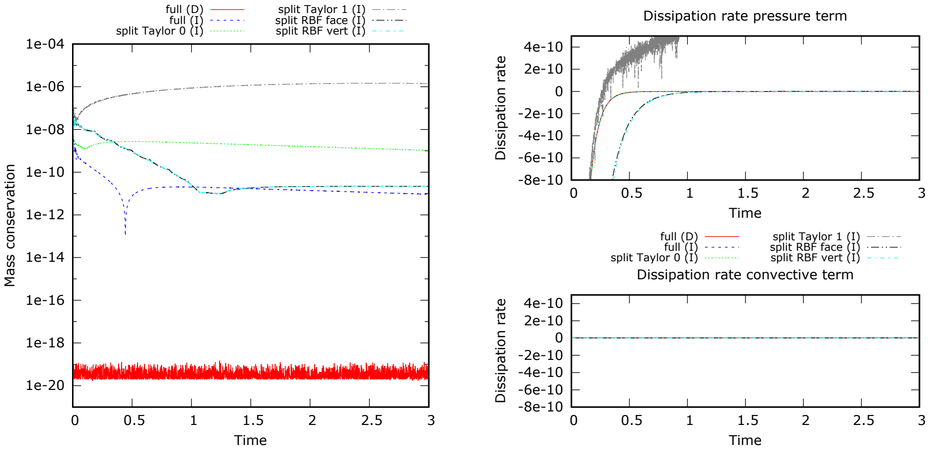
$$u=-C\sin\left(kx\right)\cos\left(ky\right)e^{-2k^2\nu t}\tag{4.15}$$

$$v=C\cos\left(kx\right)\sin\left(ky\right)e^{-2k^2\nu t}\tag{4.16}$$

where $C=3.2\times10^{-3}$ m/s is the velocity amplitude, $k=1$ is the wave number and the kinematic viscosity is set to zero ($\nu=0$ m^2/s) in order to eliminate the effects of the diffusive term. The problem is solved in a 2D mesh of size $2\pi\times2\pi$ and the density is set to unity $\rho=1$ kg/m^3. The conservation properties are function of different simulation parameters, mainly the interpolation scheme employed to calculate ϕ_m, the integration time-step Δt, the rotating velocity ω and the mesh size Δh. Therefore, a parametric study has been carried out for each one in order to characterise its influence in the conservative behaviour of the method. The first set of studies have been carried out using an unstructured mesh with 2033 of cells (see Mesh C in Fig. 4.7).

The conservation analysis for different interpolation schemes is depicted in Fig. 4.4. The conservation of a mesh without sliding boundary (full mesh) is compared using a direct solver (Direct Schur-complement based decomposition (DSD) [16]) and an iterative solver (Conjugate-Gradient (CG) method with Jacobi diagonal scaling) for solving the Poisson Eq. (4.6), since the presented method is only able to work with an iterative solver. The interpolation methods compared are the zero and first order Taylor, and the Normalized Radial Basis Function method (Shepard [17]) with a stencil of cells connected by faces and a stencil of cells connected by vertices. The methods are tested in a mesh with sliding boundary and null rotating velocity ($\omega=0$ rad/s) and as can be seen, the RBF methods are able to obtain the same order of conservation error than the ones obtained in a mesh without sliding boundary. Therefore, the following conservation tests have been carried out with the RBF with vertex connectivity.

In Fig. 4.5 the conservation properties are analysed for different time steps and two rotating velocities. As expected, the mass conservation is better for smaller time steps. Notice that for the rotating mesh ($\omega=0.1$ rad/s) the conservation error difference between time steps is smaller than for the static case ($\omega=0$ rad/s). The



(a) Mass conservation.

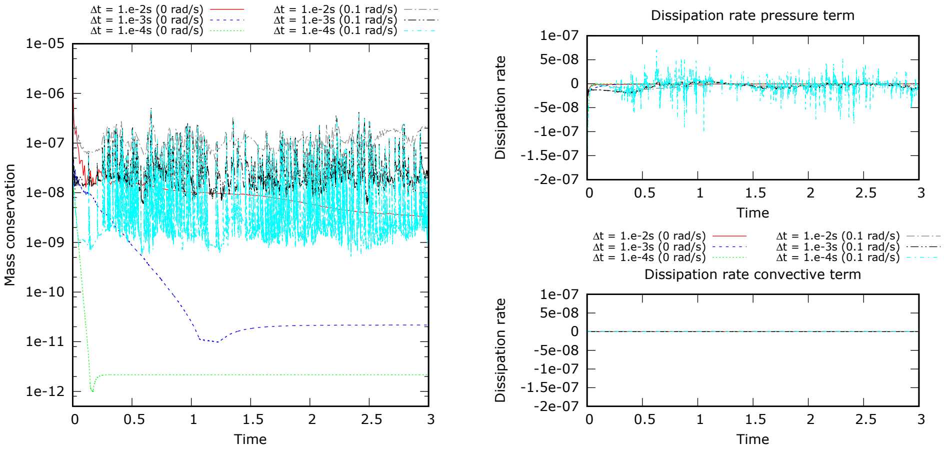
(b) Kinetic energy conservation.

Figure 4.4: Conservation for different interpolation schemes.

kinetic energy conservation error is in the same order of magnitude for all the cases. A similar trend is found in Fig. 4.6 where the effect of varying the rotating velocity (keeping constant the time step) is further studied. If the rotating velocity increases, the mass conservation is reduced. However, the mass conservation error seems to collapse at a certain value when increasing the rotating speed.

The last parameter influencing the conservation properties of the developed algorithm is the mesh. In order to asses the influence of the mesh resolution three different meshes shown in Fig. 4.7 have been studied. The three meshes have been studied using a static mesh ($\omega = 0$ rad/s) as well as with a rotating mesh ($\omega = 0.1$ rad/s). As depicted in Fig. 4.8, both mass and kinetic energy are better preserved in the finer mesh. Although this behaviour is found for both static and dynamic cases, in the rotating case the difference in the mass conservation errors are smaller than in the static one. Regarding the kinetic energy conservation, the rotating case in the coarsest mesh present the poorest values, while the finest mesh is able to achieve conservation values close to the static case.

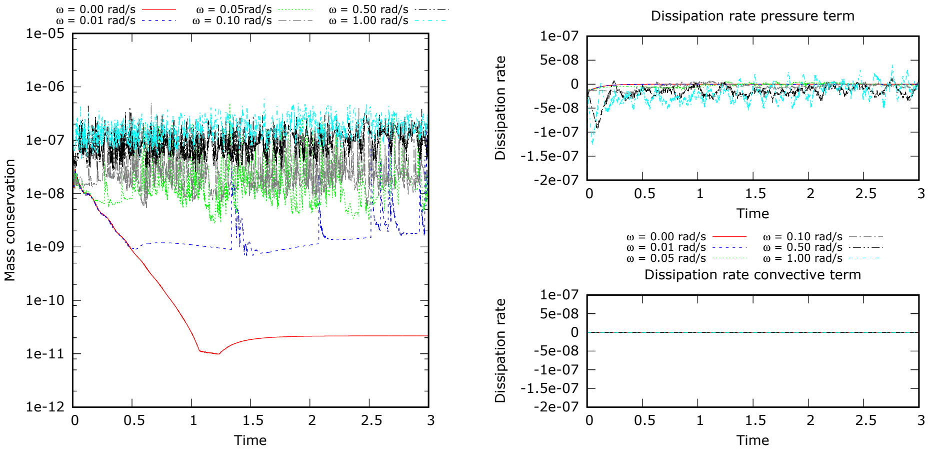
Following, the studies analysing the influence of the rotating velocity ω and the time step Δt have been repeated in an O-grid mesh type shown in Fig. 4.9. The results varying the time step Δt are shown in Fig. 4.10 and the study for different rotating velocities is depicted in Fig. 4.11. The first thing that caught the attention is the presence of some peaks appearing at regular periods. These peaks are due to the cell disposition in the O-grid meshes, and they appear because all the *ghost*



(a) Mass conservation.

(b) Kinetic energy conservation.

Figure 4.5: Conservation for different Δt.



(a) Mass conservation.

(b) Kinetic energy conservation.

Figure 4.6: Conservation for different rotating velocities ω.

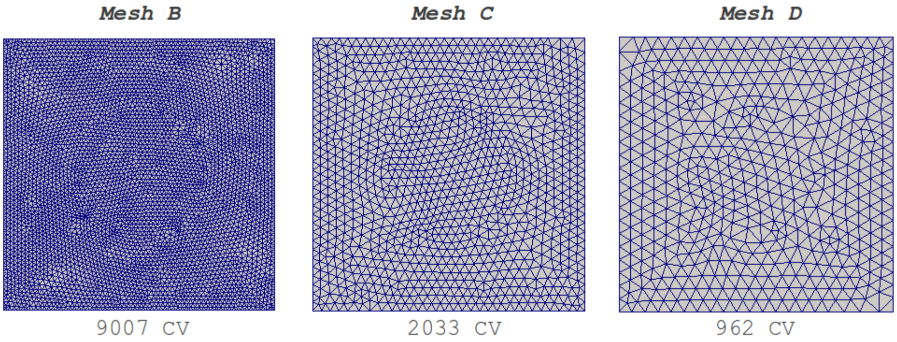
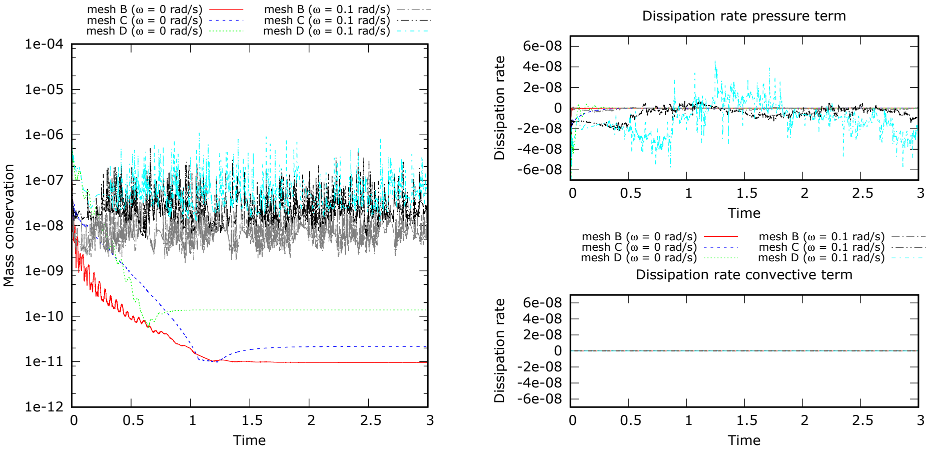


Figure 4.7: Detail of the three unstructured meshes employed in the conservation analysis.



(a) Mass conservation.

(b) Kinetic energy conservation.

Figure 4.8: Conservation for different rotating velocities ω.

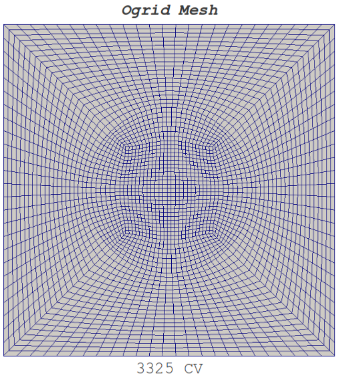


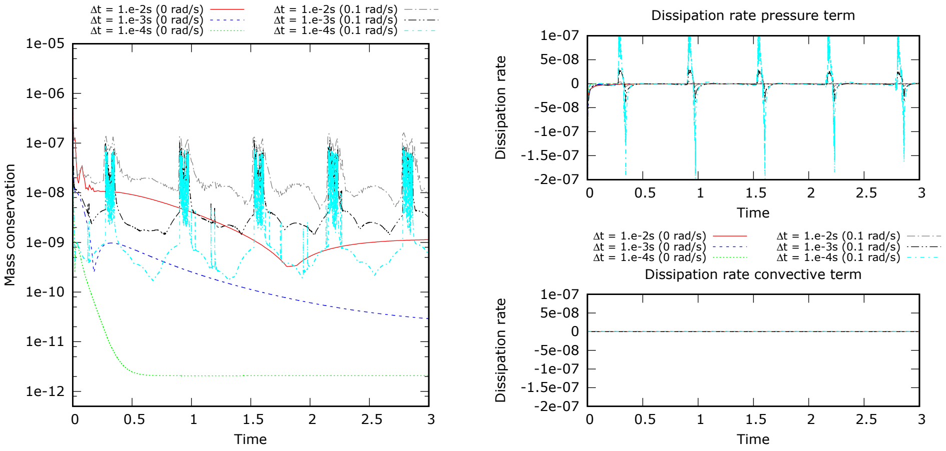
Figure 4.9: Detail of the O-grid mesh used in the conservation analysis.

nodes change simultaneously of neighbour cell, what causes a sudden modification of the Poisson matrix topology. As can be seen, when varying the integration time-step Δt all the cases have the peak at the same instant, since the change happens at the same time, while in the case for different rotating velocities, these peaks have a period inversely proportional to the rotating velocity ω. This effect does not appear in the previous studies where unstructured meshes were employed since in unstructured meshes only some of the cells modify their *neighbours* at each iteration, oppositely to what happens in the O-grid meshes, where all the cells changes their *neighbours* in the same iteration.

4.2.3 Parallelilzation strategy and speedup

As described previously, the mirror nodes are lagrangian particles that are moved during the simulation time. The mirror nodes probably will change of CPU several times along their trajectory, and therefore, a robust and efficient strategy of parallelilzation is required. Following, the parallelilzation strategy is detailed:

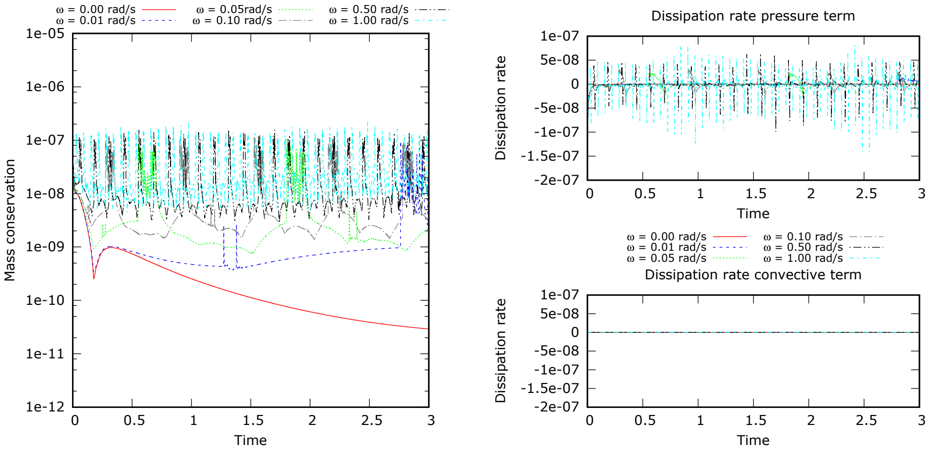
- Each injected mirror node stores the following information: its current position, the Cell ID where is living at the current time step, the parent Cell ID, and the parent Rank ID.
- Since a CPU knows how many mirror nodes has injected (basically, the number of parent cells that the CPU owns) and each mirror node knows his parent rank, the communication strategy at every iteration is as follows:
    1. Each CPU runs through all its mirror nodes performing the required interpolations.



(a) Mass conservation.

(b) Kinetic energy conservation.

Figure 4.10: Conservation for different Δt in the O-grid mesh.



(a) Mass conservation.

(b) Kinetic energy conservation.

Figure 4.11: Conservation for different rotating velocities ω in the O-grid mesh.

2. The interpolated data together with the parent Cell ID that is going to receive the information is stored in buffers with size $sizeBuf2Send_i$ (where subindex i refers to the parent Rank ID).
3. An MPI_Alltoall($sizeBuf2Send_i$,...,$sizeBuf2Rcv_i$,...) communication is performed. After this step, each CPU knows how many data have to send and receive from all the CPUs.
4. The buffers are communicated using point to point communications between the processors with $(sizeBuf2Send_i = sizeBuf2Rcv_i)! = 0$.

- In order to enhance the performance a special communicator is created at the beginning of the simulation time, including only the CPUs that have cells with a face in the sliding boundary and CPUs that are candidates to contain mirror particles at some point of the simulation time.

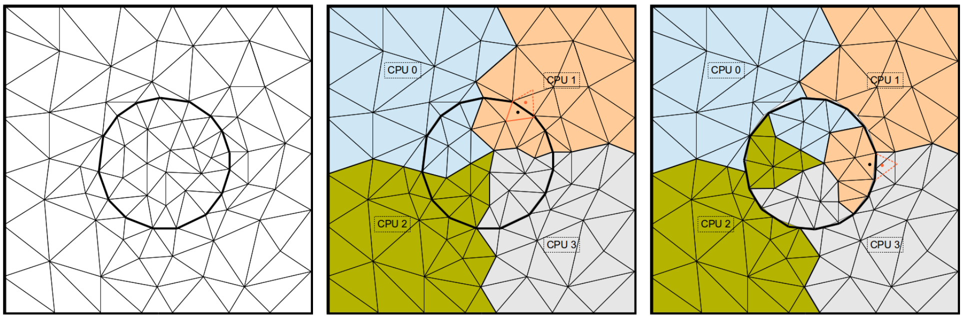
In order to show more clearly how the developed algorithm works, a simple example is presented: an initial domain like the one represented in Fig. 4.12a is divided in four parts aiming to perform a simulation using 4 processors. As detailed previously, each boundary cell of both domains injects one mirror node in the neighbour mesh. In Fig. 4.12b one of these mirror nodes has been represented. The inner mesh rotates in the clockwise direction, hence, after several iterations and for a certain Δt, the mirror particle represented in the example is transferred to a different CPU. Since each processor knows how many mirror particles belongs to it, and each mirror particle knows which is his parent processor, the communication strategy of the method in order to achieve a successful transmission of the interpolated data is detailed in the following algorithm:

Algorithm 1 Pseudo-code of the data communication in parallel computing.

```
1: for $CPU\, i = 0$ to $N$ do
2:     for $particle_{mirror}\, m = 0$ to $M$ do
3:         Interpolate scalarfields and/or gradients
4:         Store interpolated data and parent Cell ID in buffers with size $sizeBuf2Send_{i,j}$
5:     end for
6: end for
7: MPI_Alltoall($sizeBuf2Send$,...,$sizeBuf2Rcv$,...) comm.
8: Point-to-point comms between the processors with $(sizeBuf2Send_{i,j} = sizeBuf2Rcv_{j,i})! = 0$
```

where in step 3 the subindex j is the parent rank ID of the mirror particle. Notice that after step 7 each CPU knows the size of the data to send and receive.

Strong and weak speedup tests have been carried out in order to test the parallel efficiency of the algorithm. These scalability tests have been done solving the



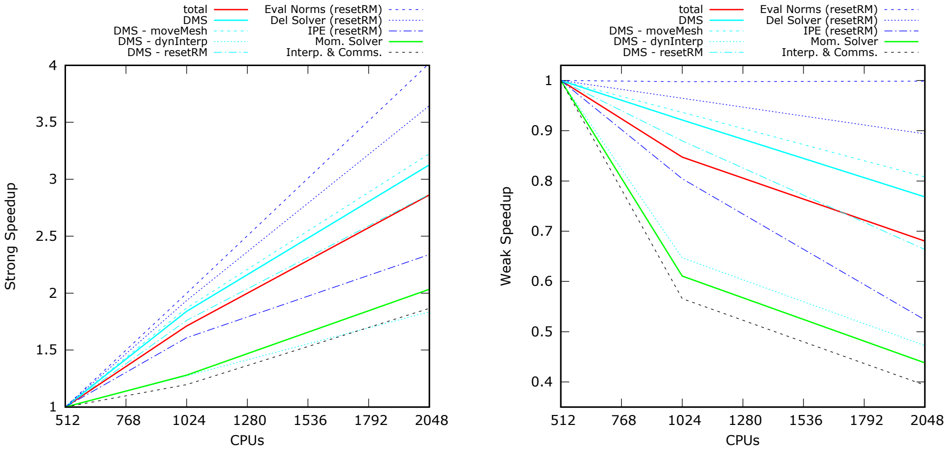
(a) Initial mesh. (b) Partitiones mesh in 4. (c) Mesh after a certain Δt.

Figure 4.12: Example of a 2D case with an external static mesh and an inner rotating mesh.

canonical Driven Cavity problem [18]. For the strong speedup a mesh of approximately 10.7M cells has been employed in 512, 1024 and 2048 CPUs, while the weak speedup has been computed starting in 512 CPU with the same mesh than for the strong speedup case, increasing the number of planes for the 1024 and 2048 CPU cases. These speedup tests have been carried out in the Vesta Cluster of the Argonne Leadership Computing Facility. The Vesta system configuration is based on an IBM BG/Q architecture, and has 2048 nodes with 16 1600 MHz PowerPC A2 cores per node (total cores: 32768). The memory per node is 16 GB RAM and the interconnection is done via a 5D Torus Proprietary Network.

The results for the strong and weak speedup are decomposed into different substeps of the algorithm (see Fig. 4.13). The two main steps are: the Dynamic Mesh Step (*DMS*) and the Momentum solver. The *DMS* includes all the steps related with the displacement of the mesh: the movement of the mesh (*moveMesh*), the displacement of the mirror nodes (*dynInterp*) and all the tasks required for the new mesh (*resetRM*), like re-evaluating the face normals, delete the solver linked with the old topology, and construct the new topology for the pressure equation (*IPE*). The Momentum step (Fig. 4.14) is decomposed in the evaluation of the convective and the diffusive operators (*momentum*), the evaluation of mass fluxes at faces (*mass*), the calculation of the velocities at the cell nodes (*vel*) and the resolution of the pressure equation (*pressEq*). Moreover, the steps of interpolation and communication related with the implemented methodology are also shown in Fig. 4.15.

The total running time for the baseline case (512 CPUs) is 1660 seconds (200 iterations). More specifically, the computational time for the *DMS* was 1375 seconds (82.8 %) while the Momentum step was 285 seconds (17.2 %). The time spent by the substeps of the *DMS* were: 1062 seconds (63.98 %) for the movement of the mesh



(a) Strong speedup.

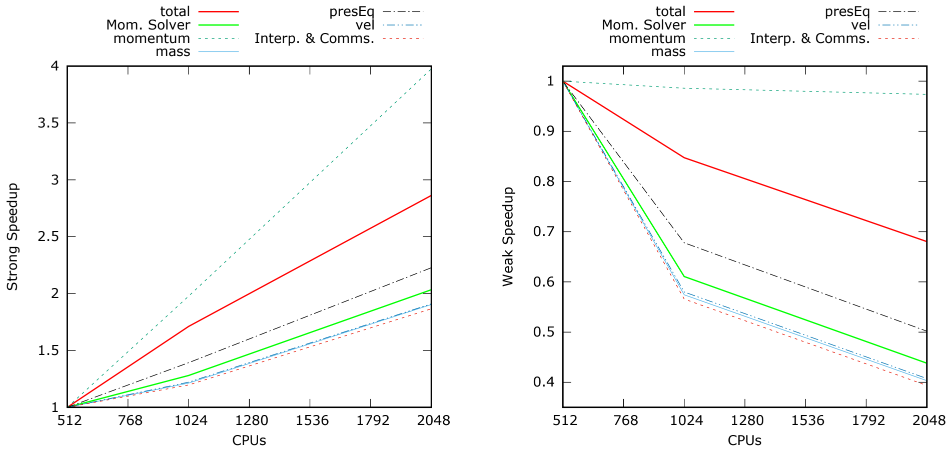
(b) Weak speedup.

Figure 4.13: Detailed speedup.

(*moveMesh*), 6 seconds (0.36 %) for the displacement and tracking of the mirror nodes (*dynInterp*), and 307 seconds (18.49 %) for all the subtasks involved in the reset of the mesh (*resetRM*).

The substeps with a lower speedup are the reconstruction of the Poisson (*IPE*), the interpolation and the communications (*Interp. & Comms.*), and the translation of the lagrangian particles (*dynInterp*). All these steps have a limited speedup because they are strongly linked to the distribution of the mirror nodes in the mesh. Depending on the mesh partitioning and the position of the sliding boundary $\partial\Omega_i$ in the domain $\Omega = \Omega_a \cup \Omega_b$, most of the mirror nodes could live in a few number of CPUs, unbalancing the computational effort and limiting the speedup of the steps involving the mirror nodes, as can be seen in the current example. Nevertheless, the total scalability is quite reasonable and promising, since the strong speedup for the 1024 CPUs case is 1.71 (parallel efficiency: 85.6 %) and for the 2048 CPUs case is 2.86 (parallel efficiency: 71.56 %).

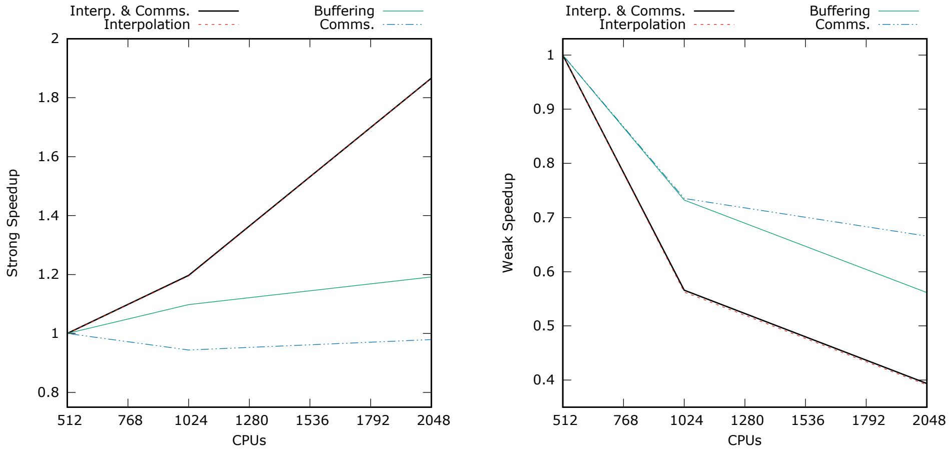
In the Momentum solver step (Fig. 4.14), the evaluation of the convective and the diffusive operators scales perfectly, since it is a sequential step that do not need any communications. The substeps for mass and velocity evaluation have a speedup that is strongly linked to the Interpolation & Communications speedup, since interpolations in the mirror nodes and communications between the meshes are required in these steps. Regarding the Interpolation & Communications, although the buffering and communications seems not to scale very well, their weight is negligible, since almost



(a) Strong speedup.

(b) Weak speedup.

Figure 4.14: Momentum solver speedup.



(a) Strong speedup.

(b) Weak speedup.

Figure 4.15: Interpolation & Communications speedup.

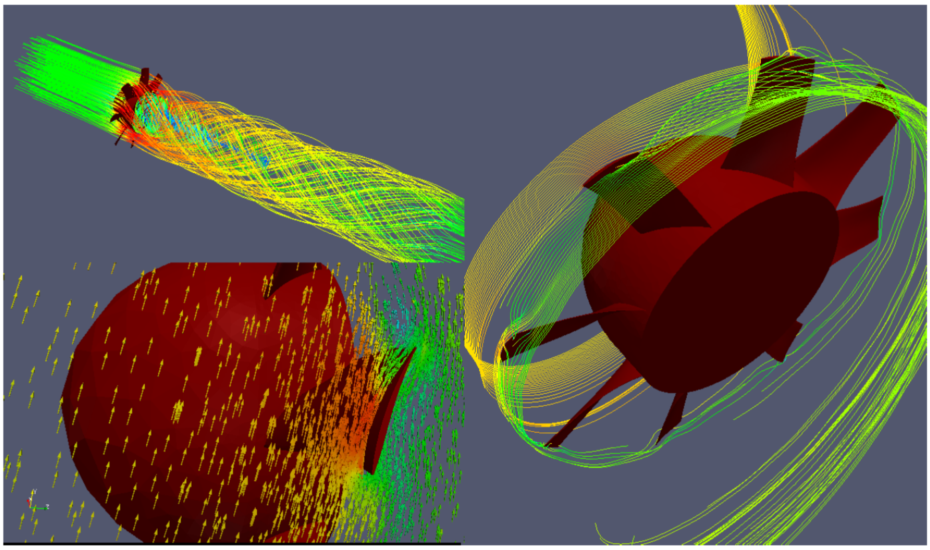


Figure 4.16: EFFAN Simulation.

all the computational effort is devoted to the interpolation step. As a matter of fact, for the baseline case, the Interpolation step took 230 seconds, while the time spent together for the Buffering and the Communications steps was less than a second. As can be seen, the speedup of the Interpolation & Communications is practically the same as the interpolation step speedup, and the interpolation step speedup is strongly dependent on the mesh partitioning and the distribution of the mirror nodes.

4.3 Real test case

In the following section an industrial application of the introduced methodology is presented. Concretely, the aim is the development of an alternative ram-air fan lay-out for the **More Electrical Aircraft (MEA)**. This task is inside the ongoing research project **EFficient FAN (EFFAN)**. The EFFAN project is being developed in the context of CleanSky project, in particular in the Systems for Green Operation (SGO) ITD. The new fan shall be capable to generate pressure drop whatever the flow without surge issues.

The simulations have been carried out employing the second order conservative discretization for unstructured meshes presented in Section 4.2.1 and Large Eddy Simulation (LES) modelling, with the unresolved Reynolds stresses closed by means of a Wall-adapting eddy viscosity model (WALE) [19] SGS model. The cases have been

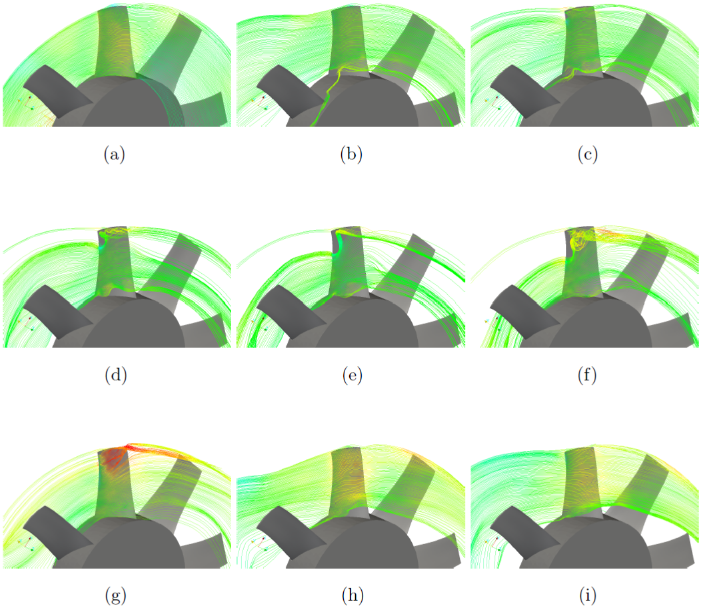


Figure 4.17: EFFAN Simulation.

computed in unstructured meshes from 100k to 10M control volumes in 512 CPUs of the UPC JFF cluster (Technical specifications: 40 cluster nodes, each node has 2 AMD Opteron with 16 Cores for each CPU linked with 64 Gigabytes of RAM memory and an infiniband QDR 4X network interconnection between nodes with latencies of 1.07 microseconds with a 40Gbits/s bandwith.). As can be seen in Fig. 4.18, the simulation results obtained using the presented method agree very well with the experimental data of the pressure rise across the fan rotor provided by the fan supplier.

4.4 Improving the method (enhancing conservation properties)

Although the previous presented methodology for sliding meshes has been proven to offer successful and reliable results, it is intended to upgrade and improve the method in order to better preserve mass, momentum and kinetic energy. The detailed analysis about the conservation properties presented in Section 4.2.2 shows that the main reason of mass losses and numerical dissipation of kinetic energy are due to the approximation in which is based the reconstruction of the Poisson matrix topology, since the method assumes that the neighbouring cell is a mirror of the cell itself, which is generally not true. Hence, the new method introduced in this section is not based

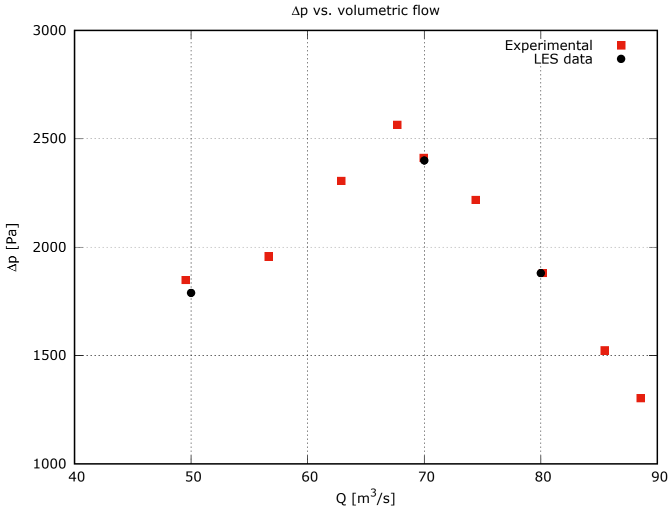


Figure 4.18: Simulation results compared against experimental data.

on this assumption and a new strategy is adopted.

The new methodology is based on the surfaces' projection of one of the sliding boundaries over the other one. The idea is to project one boundary over the other one and then obtain the intersected areas between all the boundary cells of both sliding boundaries. An schematic representation of the process is depicted in Fig. 4.19. Once the projection step has been done, each cell knows which are its neighbour cells in the neighbour mesh and which is the surface that they share between them. The idea is shown in Fig. 4.20.

This method presents two challenging aspects respect the previous one: the computational cost and the parallelization of the method. The increase in the computational effort is clear since in the new method the geometry is not supposed any more and there is an additional computational cost due to the algorithm calculating the geometry intersections. Regarding the parallelization, there is an *additional* level of complexity respecting the previous presented methodology, due to the possibility that when projecting the face of a cell over the other boundary, this face can be projected over two or more faces of cells that are owned by different processors. Take a look to Fig. 4.21a. Imagine that the blue face belongs to a sliding boundary that must be projected and intersected overt the stencil of white faces and that the red line shows the division between two processors. The projected blue face can belong to one of this two processors or to a third one. Therefore, a smart, robust and efficient

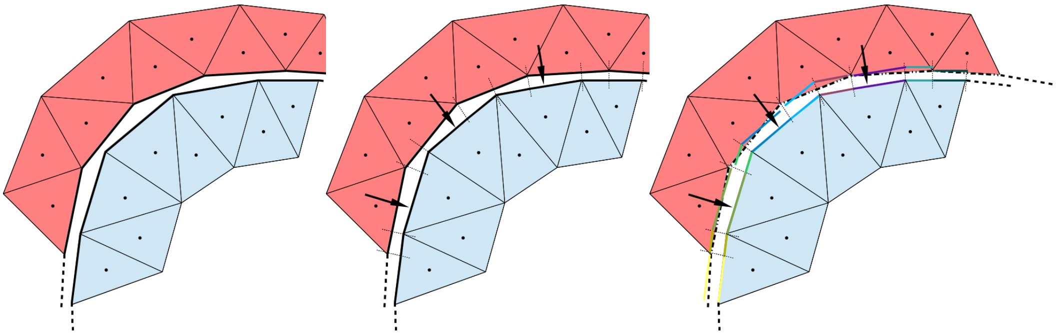


Figure 4.19: Schematic representation of the boundary projection process.

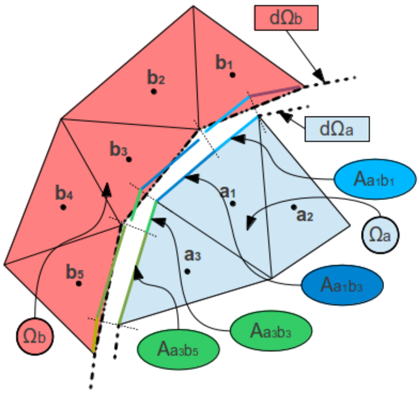
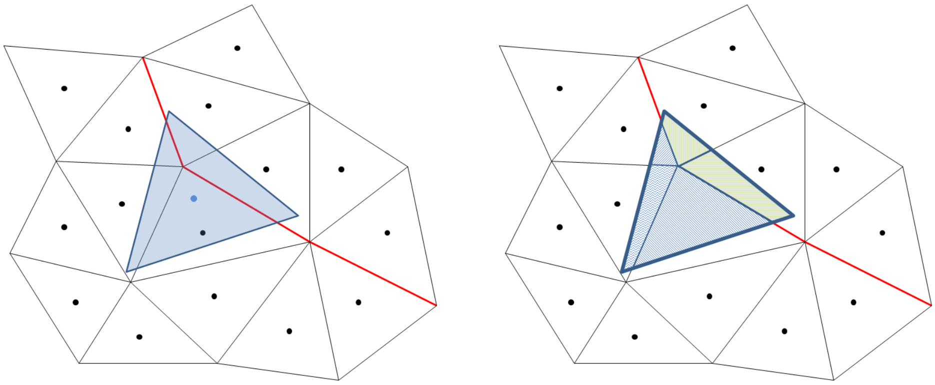


Figure 4.20: New surfaces and relations between de boundary cells after the projection and intersection of the sliding boundaries.

strategy must be employed in the projection and intersection method in order to minimize the calculations involved in the intersection operation, as well as minimizing the communications between the processors, but at the same time the algorithm must be robust enough to assure that all the faces are properly intersected. The intersection algorithm must give as result that the original surface results in five new surfaces as shown in Fig. 4.21b. Hence, after the intersection process, the cell that owns the original blue face will be connected with five cells of the neighbour mesh, three of them belonging to a processor and two other cells owned by another one.



(a) Face to be intersected (blue).

(b) Final result after the intersection.

Figure 4.21: Projection and intersection of a face over a stencil of faces belonging to two different processors.

Methodology

As mentioned so far, the method must be efficient and robust. In order to achieve these requirements the developed algorithm adopts a master-slave strategy. There is one *master* boundary which does the intersections and a *slave* boundary which *sends* all their faces to the proper processor of the *master* boundary in order to do the intersection. After the *master* boundary has calculated the resulting faces for a cell face sent by the *slave* boundary, the *master* boundary communicates the results to the implicated processors. This strategy assures that any intersection operation will be not replicated or repeated and that all the surfaces will be properly projected. It is important to state that in the pair master-slave boundaries, the master boundary is the one belonging to the static domain and the slave boundary is the one of the moving mesh. The reasons of this selection will be clarified in the detailed explanation of the method. Following, the steps performed by the algorithm are further detailed. There is an initial pre-processing step where the following actions are carried out:

1. Each face belonging to a slave boundary injects an auxiliary particle in the neighbour domain (the static one). As will be later explained, these particles moves at the same velocity than the moving domain, therefore, the pair face-auxiliary particle always maintains the same relative position.

2. The method requires an estimation of a maximum length characterizing the faces in the sliding boundaries, name it *charMaxLength*. In order to do so, for each face of the boundary it is calculated the maximum distance between their vertices.

3. Using this maximum characteristic length multiplied by a security factor f_s, each face of the master boundary saves in memory a *mini-mesh* of the surrounding faces found in a circle of radius $r_s = (f_s)(charMaxLength)$. This step is parallel, and a face will save in memory all the faces included in this area as shown in Fig. 4.22, no matter if they belong to the same processor or a neighbour one. Although this strategy represents an increase of the allocated memory, it allows to boost up the process of geometric intersections and avoids communications during this process.

Once this pre-processing step has been done, the method is ready to be employed during simulation time. Following there are detailed the steps carried out by the algorithm in order to connect both sliding and static meshes at each iteration:

1. The auxiliary particles living in the static mesh domain are moved at the same speed than the moving mesh. Hence, the relative position between the auxiliary particle and the *slave* face node which injected is preserved.

2. Each auxiliary particle is projected towards the position of the face node in the *slave* mesh which injected it until the particle arrives to the boundary of the static mesh. The intersected face of the *master* boundary will be the one in charge of perform the intersection process of the *slave* face associated to the auxiliary particle.

3. Once it is known the *master* face responsible of calculating the intersections, this info is communicated to the *slave* face. This latter, packs its geometry info and sends it to the processor owning the *master* face.

4. All the processors owning faces of the *master* boundary perform the intersection operations. It could happen that some *master* faces has not assigned any intersection operation, while other faces has to perform the intersection of two or more *slave* faces. It is not a problem, since all the faces of the *slave* boundary are intersected, and hence, all the surface of the boundary is covered.

5. Once all the intersection tasks has been performed, the resulting surfaces and relations are communicated to the involved boundary cells.

6. From these new relation between the cells of both meshes, it is created a new topology accounting for the new *neighbourhood* between cells and processors.

7. When performing the intersection of faces that are in different planes, as happens, for example, in circular boundaries (see Fig. 4.19 or Fig. 4.20), it may result that the sum of the areas of the new projected faces (Fig. 4.21b) are different to the area of the original face (Fig. 4.21a). Therefore, a normalization procedure is carried out.

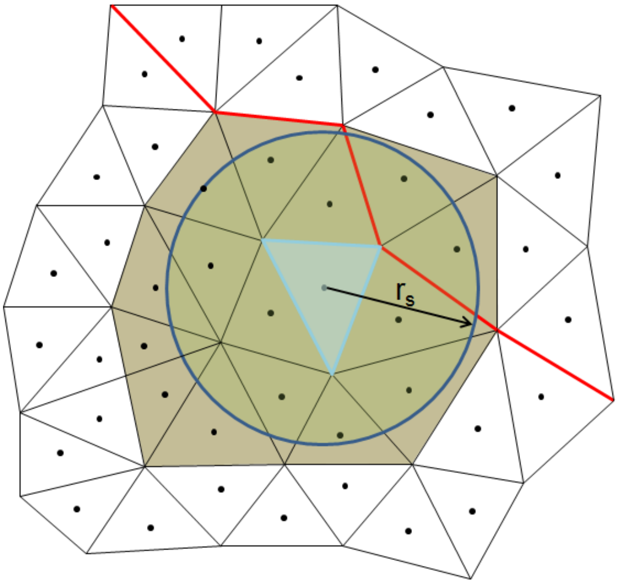
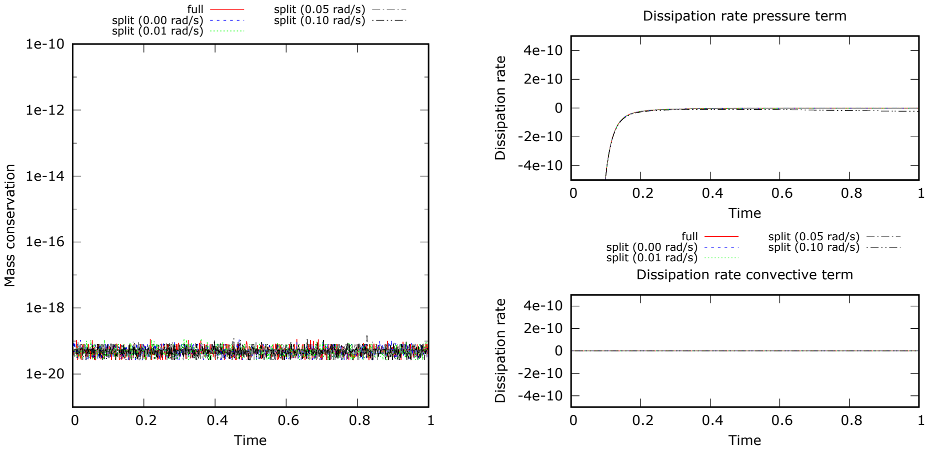


Figure 4.22: Mini mesh saved in the face coloured as blue, involving all the cells inside the radius r_s.

8. Once all the new relations and the new geometrical values (surfaces, normals, etc.) are known, the integration of the discrete Navier-Stokes are done treating this new generated faces in the sliding boundary as any other inner face of the mesh.

Mass and kinetic energy Conservation analysis

In order to demonstrate the conservation capabilities of this new methodology, the same numerical problem presented previously in Section 4.2.2 is chosen. The results for different rotating velocities in a split unstructured mesh are depicted in Fig. 4.23. The conservation properties of the method are also compared against a static case without sliding boundary, i.e., in a full mesh without division. As can be seen, the mass conservation of the method is almost *perfect*, since all the rotating cases present the same mass conservation than the static case solved in the full mesh. Regarding the conservation of the kinetic energy, shown in Fig. 4.23b, the preservation of the kinetic energy using the new method is also very good, and only for the highest rotating velocity there is a small dissipation of the kinetic energy associated to the discretization of the pressure term. However, the magnitude of this dissipation is very small and almost negligible.



(a) Mass conservation.

(b) Kinetic energy conservation.

Figure 4.23: Conservation properties of the new methodology for different rotating velocities ω compared against a static full mesh.

4.5 Conclusions

In this chapter a new methodology for CFD simulations with adjacent disconnected unstructured 3D meshes has been presented, studied and assessed. The developed methodology is able to perform numerical simulations with two or more non-overlapped meshes with common boundaries. The method is specially well-suited for cases where the meshes have a relative motion between them. Hence, the presented technique is very interesting for applications like wind turbines or turbo-machinery, where the interaction between the moving and the static parts is a crucial aspect in order to obtain reliable results and a full image of the physics taking place in these cases.

An exhaustive analysis on the conservation properties of the methodology has been presented, demonstrating that the method is able to conserve very well both mass and kinetic energy. The method has been analysed for different situations and using different meshes. In all the cases the conservation analysis shows good results. Moreover, the methodology has been applied to demanding test cases and has demonstrated to be able to obtain reliable and accurate results. Furthermore, the scalability and performance of the method in parallel simulations have also been carefully analysed. The results of this analysis show that the method is capable to handle demanding cases employing hundreds and thousands of processors in parallel computations, and also exhibiting a good scalability. Hence, it is demonstrated that the presented method is a powerful tool to carry out numerical simulations of fans, turbo-machinery, wind turbines, and so on.

Although the presented method obtains good results in the mass and kinetic energy conservation analysis, it is intended to improve it in order to obtain even better results. The improved method still relies on the auxiliary particles in order to perform efficiently the link of the independent meshes. However, unlike to the previous presented method, the new one does not suppose the geometry of the neighbour cells and performs a projection and intersection of the cell's faces in the sliding boundary. Therefore, a more accurate reconstruction of the discrete operators in the boundary cells can be carried out. This new methodology is currently under development, but the preliminary results obtained using this enhanced version of the algorithm are very promising. The mass conservation obtained using this method in sliding meshes presents the same values than the overall mass conservation obtained in an equivalent full stationary mesh. Regarding the kinetic energy, the conservation results are also very good, showing only a very small dissipation, almost negligible, associated to the discretization of the pressure term. Therefore, the aim of improving the already good conservation properties of the previous presented method has been achieved. These better conservation properties come at the expense of an increase in the computational load respect the former method, since there are additional calculations in order to obtain the geometrical relations between the boundary cells, while in the old method the geometry is supposed instead of being calculated.

References

[1] J. L. Steger, F. C. Dougherty, and J. A. Benek. A chimera grid scheme.[multiple overset body-conforming mesh system for finite difference adaptation to complex aircraft configurations]. 1983.

[2] M. M. Rai. A conservative treatment of zonal boundaries for euler equation calculation. 1984.

[3] M. M. Rai. A conservative treatment of zonal boundaries for euler equation calculations. *Journal of Computational Physics*, 62(2):472–503, 1986.

[4] Martin Beaudoin and Hrvoje Jasak. Development of a generalized grid interface for turbomachinery simulations with OpenFOAM. In *Open source CFD International conference*, volume 2. Berlin, 2008.

[5] H. G Weller, G. Tabor, H. Jasak, and C Fureby. A tensorial approach to computational continuum mechanics using object-oriented techniques. *Computers in physics*, 12(6):620–631, 1998.

[6] Olivier Petit, Maryse Page, Martin Beaudoin, and Håkan Nilsson. The ERCOFTAC centrifugal pump OpenFOAM case-study. In *Proceedings of the 3rd IAHR International Meeting of the Workgroup on Cavitation and Dynamic Problem in Hydraulic Machinery and Systems, Brno, Czech Republic*, 2009.

[7] M. Behr and T. Tezduyar. The shear-slip mesh update method. *Computer Methods in Applied Mechanics and Engineering*, 174(3-4):261–274, 1999.

[8] M. Behr and T. Tezduyar. Shear-slip mesh update in 3D computation of complex flow problems with rotating mechanical components. *Computer Methods in Applied Mechanics and Engineering*, 190(24):3189 – 3200, 2001. Advances in Computational Methods for Fluid-Structure Interaction.

[9] Y. Bazilevs and T. J. R. Hughes. NURBS-based isogeometric analysis for the computation of flows about rotating components. *Computational Mechanics*, 43(1):143–150, Dec 2008.

[10] Ming-Chen Hsu and Yuri Bazilevs. Fluid-structure interaction modeling of wind turbines: simulating the full machine. *Computational Mechanics*, pages 1–13, 2012.

[11] Ming-Chen Hsu, Ido Akkerman, and Yuri Bazilevs. Finite element simulation of wind turbine aerodynamics: validation study using NREL Phase VI experiment. *Wind Energy*, 17(3):461–481, 3 2014.

[12] J. McNaughton, I. Afgan, D. D. Apsley, S. Rolfo, T. Stallard, and P. K. Stansby. A simple sliding-mesh interface procedure and its application to the CFD simulation of a tidal-stream turbine. *International Journal for Numerical Methods in Fluids*, 74(4):250–269, 2014.

[13] Benjamin Francois, Michel Costes, and Guillaume Dufour. Comparison of chimera and sliding mesh techniques for unsteady simulations of counter rotating open-rotors. In *20th ISABE Conference*, 2011.

[14] Ll. Jofre, O. Lehmkuhl, F.X. Ventosa, J. Trias, and A. Oliva. Conservation properties of unstructured finite-volume mesh schemes for the Navier-Stokes equations. *Numerical Heat Transfer, Part B: Fundamentals*, 65(1):53–79, 2014.

[15] R.W.C.P. Verstappen and A.E.P. Veldman. Symmetry-preserving discretization of turbulent flow. *Journal of Computational Physics*, 187:343–368, 2003.

[16] R. Borrell, O. Lehmkuhl, F.X. Trias, and A. Oliva. Parallel direct poisson solver for discretisations with one fourier diagonalisable direction. *Journal of Computational Physics*, 12:4723–4741, 2011.

[17] William H. Press. *Numerical recipes 3rd edition: The art of scientific computing.* Cambridge University Press, 2007.

[18] U. Ghia, K.N. Ghia, and C.T. Shin. High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method. *Journal of computational physics*, 48(3):387–411, 1982.

[19] Franck Nicoud and Frédéric Ducros. Subgrid-scale stress modelling based on the square of the velocity gradient tensor. *Flow, turbulence and Combustion*, 62(3):183–200, 1999.

5

A self-adaptive strategy for the time-integration of convection-diffusion equations

Abstract. A self-adaptive strategy for the maximisation of the time-step for the numerical solution of convection-diffusion equations is discussed, specifically the Navier-Stokes equations. The presented algorithm smartly modifies the numerical integration method depending on the properties of the system matrix in order to maximize the stability region associated to the integration method. Furthermore, the method also evaluates the optimal time-step of the simulation. Commonly, a Courant-Friedrich-Lewy (CFL) condition with a fix Courant number is used. The Courant number is usually set through previous knowledge of the case of study and the temporal algorithm. The present self-adaptive strategy relies on the evaluation of the eigenvalues of the system to set the optimal time step. However, to avoid the costly evaluation of the eigenvalues, an efficient low-cost bounding method is presented. Previous works on self-adaptive strategies had been focused on explicit 2^{nd} order Linear Multi-Step schemes and symmetry-preserving schemes. In this thesis the methodology is extended to Predictor-Corrector methods and Runge-Kutta schemes. Furthermore, extension to non-symmetry-preserving schemes is carried on, namely upwind-like schemes. Additionally, the method is also applied to variable density cases. It is shown that the method is able to automatically modify the integration scheme to maximize the stability region and find the maximum allowable time-step at each iteration for any temporal and discretisation schemes.

5.1 Introduction

The Navier-Stokes equations describe the motion of flows. Their solution yields an accurate representation of the flow transient motion. However, no general analytical solution has been found and numerical solution of these equations has proven to be challenging and expensive in terms of required resources.

In transient simulations, sufficiently long temporal spans have to be simulated in order to correctly characterise the statistical behaviour of the flow. Also, it is often of interest to simulate long periods of time. Hence, there is a need of advancing in time as fast as possible. Then, besides decreasing the required effort per time-step, either increasing the computational power or reducing the number of operations per time step, using as large as possible time-steps is also a viable option. Still, the time-step cannot be chosen freely and is mainly limited for two reasons. First, flow dynamics must be properly represented. If a Direct Numerical Simulation (DNS) is to be performed, all temporal scales down to the Kolmogorov time-scale have to be resolved [1]. Therefore the integration time-step is limited by the time-scale of the problem to be simulated. Second, the time-step is limited due to stability reasons of the temporal discretisation scheme employed to integrate the problem. Depending on the governing equations, each temporal scheme has a maximum time-step that provides an accurate numerical solution. If this time-step is exceeded, the numerical integration will diverge. The latter constraint usually poses a much stricter limitation on the allowable time-step that can be used.

Each temporal discretisation scheme has its own stability region, allowing different time-steps for the same problem. Widely used schemes, such as explicit Adams-Bashforth methods (AB) family [2, 3], Runge-Kutta (RK) methods [4–7] or Predictor-Corrector methods (PC) [8–10], among others, allow for different time-steps. However, setting the time-step for a specific simulation is usually performed through the Courant-Friedrich-Lewy (CFL) condition. Through it, limitations on the time-step are set based on the domain of dependency. The evaluation is performed based on the convective speed and the diffusivity coefficient. Courant number (C) values are usually selected based on previous experience or on analysis of the behaviour of the implemented temporal scheme and physics being considered. Hence, usually the resulting time-step is not the optimal one, unless an exhaustive testing is performed for the particular case of study.

To overcome this limitation and to select an optimal time-step, it was proposed to perform an analysis of the stability region of the temporal scheme. Verstappen and Veldman [11] used a variant of the 2^{nd} order Adams-Bashforth scheme, where the coefficients of the temporal scheme were selected based on the properties of the matrix system for the specific problem of study. With this approach, the increased stability region allowed using larger time-steps. Following, Trias and Lehmkuhl [1] developed a self-adaptive strategy which modifies the temporal scheme coefficients and dynamically selects the optimal time-step during simulations. Time-step increases up to four times when compared to common applications of the CFL condition were reported.

Previous works were focused on convection-diffusion equations where the convective operator was constructed employing a Symmetry-Preserving (SP) discretisation scheme [12] and the diffusive operator was build using a Central Difference Scheme (CDS). This discretisation leads to an skew-symmetric convective operator and a symmetric negative-definite diffusive operator. These operators features are set by the need of preserving kinetic energy and avoiding numerical diffusion [13]. Hence, this requirement is placed to the momentum equation. However, for scalar's transport equations, such as temperature or species, this requirement is not placed. Furthermore, for cases with high Peclet numbers, centred approximations such as the SP may become unstable. Hence, in the present work the self-adaptive strategy is extended to handle non-skew-symmetric convective terms, which result from the use of Upwind-like schemes. In these cases, it is shown that the system matrix may still be decomposed into a skew-symmetric operator and a symmetric operator. Then, through the estimation of the eigenvalues of each matrix operator, the optimal temporal scheme coefficients and time-step may be determined.

Additionally, as discussed earlier, different temporal explicit schemes are commonly used in the numerical solution of the Navier-Stokes equations. Hence, the method is also extended to other temporal schemes, such as Predictor-Corrector methods and Runge-Kutta methods.

With the aforementioned aim, the rest of the chapter is organised as follows. In the second section, the mathematical background is presented. In its first subsection it is explained how the maximum time step of the system can be determined. It is done through the definition of the stability region of a temporal integration scheme, its stability polynomial and the eigenvalues of the system. Following, the procedure to maximise the time-step for any temporal integration scheme is described. As a first step, the method is applied to an explicit 2^{nd} order Linear Multi-step method. Then, the described methodology is applied to the Navier-Stokes equations. In this subsection, the properties of the convective and diffusive operators are analysed. Following, the system eigenvalues bounding methodology employing the Gershgorin Circle Theorem is detailed. Furthermore, it is numerically analysed how the eigenvalues of the convection-diffusion matrix are bounded by the eigenvalues of both convective and diffusive matrices. Both symmetry-preserving and non-symmetry-preserving schemes are considered. Moreover, the influence of the volume matrix in the system eigenvalues is mathematically analysed. Closing the section, extension of the methodology is performed to Predictor-Corrector methods and Runge-Kutta schemes. In the third section, in order to illustrate the behaviour of the proposed methodology, the eigenvalue bounding technique developed in previous sections is studied and assessed through CFD numerical simulations. The maximum eigenvalues obtained from the method are compared against the exact eigenvalues of the system matrix. Afterwards, the method is applied to the numerical simulation of a differentially heated cavity problem using different convective schemes, where the performance of the method is shown. Similarly, simulations are also carried out for different temporal schemes. Closing the section, the method is shown to be also effective for variable density cases. Finally, relevant results are summarised and conclusions are given in the last section.

5.2 Mathematical formulation

5.2.1 The stability region

In the field of physics, and more precisely in heat-transfer and fluid-dynamic problems, interest is placed in solving systems of Ordinary Differential Equations (ODEs) where the initial state of the system is known. These problems are commonly denoted as **Initial Value Problem** (IVP):

$$y' = f(t, y) \tag{5.1a}$$

$$y(t_0) = y_0 \tag{5.1b}$$

Usually these ODEs represent a physical phenomenon with spatio-temporal dependence. These problems can be expressed in discrete form as a linear system of ODEs:

$$\frac{d\mathbf{y}}{dt} = [\mathbf{A}]\,\mathbf{y} \tag{5.2}$$

where $[\mathbf{A}] \in \mathcal{M}_{NxN}$ is a square matrix and $\mathbf{y} \in \mathbb{R}^N$ is the unknowns vector. In order to solve this problem the system needs to be time-integrated. The most common numerical methods employed for the numerical solution of IVPs are Runge-Kutta, Linear Multi-Step and Taylor series methods. Focusing on Linear Multi-step (LM) methods, they can be written as:

$$\sum_{j=0}^{k} \alpha_j \mathbf{y}^{\mathbf{n+j}} = h \sum_{j=0}^{k} \beta_j \mathbf{f}(\mathbf{t}^{\mathbf{n+j}}, \mathbf{y}^{\mathbf{n+j}}) \tag{5.3}$$

where coefficients α_j and β_j are real constants, h is the time-integration step and k is the order of the linear k-step method. It is assumed that $\alpha_k \neq 0$ and that α_0 and β_0 are not both equal to zero. If $\beta_k = 0$ the k-step method is said to be **explicit**, since the value of y^{n+k} is obtained from known values. Otherwise, if $\beta_k \neq 0$ the method is **implicit**, because the value of $f(x^{n+k}, y^{n+k})$ has to be known in order to obtain y^{n+k}.

When solving a problem as defined by Eq. (5.2) using a LM method, the order k and the coefficients α_j and β_j are chosen at the beginning of the calculation and usually do not change. Besides, the integration step h should be chosen as large as possible in order to minimize the computational cost but small enough to guarantee a good numerical solution [14]. If the integration step h is too large the time integration process will lead to an unstable numerical solution and it might diverge. In order to determine the maximum stable step size h, it is useful to decompose Eq. (5.2) into a set of uncoupled equations via the diagonalization of the system's matrix $[\mathbf{A}] = [\mathbf{P}]^{-1}[\mathbf{\Lambda}][\mathbf{P}]$:

$$\frac{d\mathbf{s}}{dt} = [\mathbf{\Lambda}]\,\mathbf{s} \tag{5.4}$$

where $\mathbf{s} = [\mathbf{P}]\,\mathbf{y}$ and $[\mathbf{\Lambda}]$ is a diagonal matrix with the λ_i eigenvalues at the diagonal. Therefore, the system of equations becomes a set of N uncoupled equations of the form:

$$\frac{ds_i}{dt} = \lambda_i s_i \tag{5.5}$$

Applying the linear k-step method presented in Eq. (5.3) to problem in Eq. (5.5) will lead to:

$$\sum_{j=0}^{k} \alpha_j s_i^{n+j} = h\lambda_i \sum_{j=0}^{k} \beta_j s_i^{n+j} \tag{5.6}$$

From there on, assuming that s_i can be expressed in the form z^j, where z is a complex number, the previous equation can be rewritten as:

$$\pi(z) = \sum_{j=0}^{k} \alpha_j z^j - h\lambda_i \sum_{j=0}^{k} \beta_j z^j = 0 \tag{5.7}$$

The polynomial $\pi(z)$ is named the **stability polynomial**. Introducing, $\rho(z) = \sum_{j=0}^{k} \alpha_j z^j$ and $\sigma(z) = \sum_{j=0}^{k} \beta_j z^j$, the **first** and the **second characteristic polynomials**, respectively, the stability polynomial can be expressed as:

$$\pi(z, \bar{h}) = \rho(z) - \bar{h}\sigma(z) = 0 \tag{5.8}$$

where $\bar{h} = h\lambda_{max}$ and λ_{max} is the eigenvalue λ_i with the biggest complex modulus $\{\lambda_{max} \in \Lambda : \max(|\lambda_i|)\}$ being $\Lambda = \{\lambda_i\}$. The linear k-step method Eq. (5.6) is stable and converges for an integration step $h > 0$ if all the roots of the stability polynomial Eq. (5.8) have modulus less than one $|r_s| < 1$ [15].

Definition 1. *A linear k-step method is said to be* ***absolutely stable*** *in an open set* $\mathbf{R_A}$ *of the complex plane, if for all* $\bar{h} \in \mathbf{R_A}$ *all roots* r_s, s = *1,...,k, of the stability polynomial* $\pi(r, \bar{h})$ *associated with the method, and defined by Eq.* (5.8)*, satisfy* $|r_s| < 1$*. The set* $\mathbf{R_A}$ *is called* ***region of absolute stability*** *of the method.*

Hence, forcing $|r_s(\bar{h})| < 1$ the optimal step size h can be easily obtained if the eigenvalues λ_i of the system are known.

Application to the 2^{nd} order explicit method Adams-Bashforth

To illustrate the preceding discussion, an example of how to determine the maximum stable step size for the second-order explicit method Adams-Bashforth (AB2) is detailed. This LM is applied to the one-dimensional Cauchy problem $\phi' = f(\phi)$. The AB2 temporal scheme reads as:

$$\frac{\boldsymbol{\phi}^{n+2} - \boldsymbol{\phi}^{n+1}}{\Delta t} = \frac{3}{2} f(\boldsymbol{\phi}^{n+1}) - \frac{1}{2} f(\boldsymbol{\phi}^{n}) \tag{5.9}$$

Assuming that Δt is small enough, the non-linear function $f(\phi)$ can be linealized as $f(\phi_i) \approx \lambda_i \phi_i$, where $\lambda_i \in \mathbb{C}$ are the complex eigenvalues of the system. After linealization, Eq. (5.9) is rewritten as

$$\phi_i^{n+2} = \phi_i^{n+1} + \overline{h_i} \left[\frac{3}{2} \phi_i^{n+1} - \frac{1}{2} \phi_i^{n} \right] \tag{5.10}$$

where $\overline{h_i} = \Delta t \lambda_i = -e^{-i\varphi} \widetilde{\Delta t}$. Note that $-e^{-i\varphi} = \lambda_i / |\lambda_i| \in \mathbb{C}$ is a unitary vector and $\widetilde{\Delta t} = \Delta t \, |\lambda_i| \in \mathbb{R}$. In order to find the roots r_s of the stability polynomial associated to Eq. (5.10) for the most restrictive eigenvalue λ_{max}, the latter is expressed in matrix form:

$$\begin{pmatrix} \phi^{n+2} \\ \phi^{n+1} \end{pmatrix} = \begin{pmatrix} 1 + \frac{3}{2}\bar{h} & -\frac{1}{2}\bar{h} \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \phi^{n+1} \\ \phi^{n} \end{pmatrix} = [\mathbf{T}] \begin{pmatrix} \phi^{n+1} \\ \phi^{n} \end{pmatrix} \tag{5.11}$$

where ϕ is the value $\phi_i \in \boldsymbol{\phi}$ associated to λ_{max} (for the sake of clarity, the subindex i is omitted in Eq. (5.11)). The eigenvalues of matrix $[\mathbf{T}]$ are the roots of its characteristic polynomial, which is defined as $P_T(\bar{h}) = det \, |[\mathbf{T}] - \lambda_{ab2} I|$:

$$P_T(\bar{h}) = det \left| \begin{pmatrix} A & B \\ 1 & 0 \end{pmatrix} - \begin{pmatrix} \lambda_{ab2} & 0 \\ 0 & \lambda_{ab2} \end{pmatrix} \right| = \lambda_{ab2}^2 - A\lambda_{ab2} - B \tag{5.12}$$

where $A = 1 + \frac{3}{2}\bar{h}$ and $B = -\frac{1}{2}\bar{h}$. Therefore, the roots of Eq. (5.12) are:

$$r_s(\bar{h}) = \lambda_{ab2} = \frac{A \pm \sqrt{A^2 + 4B}}{2} \tag{5.13}$$

From Definition 1, the **region of absolute stability** ($\mathbf{R_A}$) can be drawn in the complex plane imposing $\left|r_s(\bar{h})\right| \leq 1$. For the 2^{nd} order Adams-Bashforth method the region $\mathbf{R_A}$ is depicted in Fig. 5.1. The boundary $\delta\mathbf{R_A}$ belongs to the expression $\left|r_s(\bar{h})\right| = 1$.

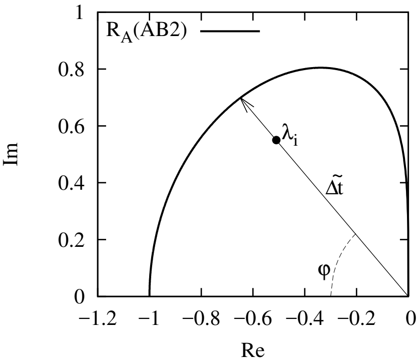


Figure 5.1: Stability Region R_A for 2^{nd} order Adams-Bashforth. The line $R_A(AB2)$ denotes $\left|r_s(\bar{h})\right| = 1$.

Once the Stability Region $\mathbf{R_A}$ of the linear multi-step method is known, the maximum stable integration step Δt is evaluated as follows: i) Determine φ from system's most restrictive eigenvalue λ_{max}, using $-e^{-i\varphi} = \lambda_{max}/\left|\lambda_{max}\right|$. ii) Look for the intersection between the line that crosses the complex origin $\mathcal{O}$ with angle φ and the boundary δR_A, obtaining $\widetilde{\Delta t}$. iii) Evaluate the maximum stable integration step from $\Delta t = \widetilde{\Delta t}/\left|\lambda_{max}\right|$.

5.2.2 Maximizing the integration step

In the previous section, the evaluation of the maximum step size h which guarantees that a LM method is stable has been shown. For a given linear k-step method the stability region $\mathbf{R_A}$ is immutable, since the stability polynomial of the method is constant. However, the integration time step could be maximised if the stability polynomial could be dynamically modified and the stability region $\mathbf{R_A}$ deformed depending on the system eigenvalues λ_i. The key point to generate a dynamic self-adapting LM method is to construct it using a bounded free-parameter κ and find a smart way to modify it. Hence, the parameter κ will be function of the system eigenvalues, i.e. $\kappa = \kappa(\lambda_i)$.

In order to construct a LM scheme four key aspects should be considered: accuracy, consistency, zero-stability and stability domain. The concept of stability domain has already been covered in Section 5.2.1.

Accuracy It can be shown [16] that a method has order of accuracy p if and only if:

$$C_0 = C_1 = ... = C_p = 0 \text{ and } C_{p+1} \neq 0 \tag{5.14}$$

where:

$$C_0 = \sum_{j=0}^{k} \alpha_j \tag{5.15a}$$

$$C_1 = \sum_{j=1}^{k} j\alpha_j - \sum_{j=0}^{k} \beta_j \tag{5.15b}$$

$$C_2 = \sum_{j=1}^{k} \frac{j^2}{2!}\alpha_j - \sum_{j=0}^{k} j\beta_j \tag{5.15c}$$

$$C_q = \sum_{j=1}^{k} \frac{j^q}{q!}\alpha_j - \sum_{j=0}^{k} \frac{j^{q-1}}{(q-1)!}\beta_j \tag{5.15d}$$

Consistency A LM method is consistent if the local error tends to zero when $\Delta t \to 0$. Any LM method of at least first order of accuracy is consistent. In other words, any LM method will be consistent if $C_0 = 0$ and $C_1 = 0$.

Zero-stability The zero-stability concept is related with the stability and sensitivity of a LM scheme to numerical errors in the starting values. A LM method is said to be zero-stable if it fulfils the *root condition*: all roots of the first characteristic polynomial of a LM scheme have to be inside the unit disc $|r_\rho| \leq 1$. Furthermore, if any root lies in the boundary $|r_\rho| = 1$, its multiplicity must be 1 (a simple root). This condition is named zero-stability because it is checked setting $\Delta t = 0$ (or equivalently $y' = 0$).

Construction of an explicit Linear Multi-step method with a free parameter

Once the basic mathematical aspects to construct a LM scheme have been covered, the construction of a 2^{nd} order explicit method with a free parameter κ is detailed ($\kappa 2LM$). The method can be expressed as:

$$\alpha_2\phi^{n+2} + \alpha_1\phi^{n+1} + \alpha_0\phi^n = \Delta t \left[\beta_1 f(\phi^{n+1}) + \beta_0 f(\phi^n)\right] \tag{5.16}$$

where it has been imposed $\alpha_2 = \kappa + 1/2$, $\alpha_1 = -2\kappa$ and $\alpha_0 = \kappa - 1/2$. These constraints fulfil the zero-stability condition for $0 \leq \kappa \leq 1$. Since the method is explicit, the condition $\beta_2 = 0$ has also been set. Consequently, there are two unknowns: β_0, β_1. Therefore, two equations are required, Eq. (5.15b) and Eq. (5.15c). Equation (5.15a) is only a constraint for α_i terms and is already fulfilled. Using Eq. (5.15) it results in:

$$C_0 = \sum_{j=0}^{k} \alpha_j = \kappa + 1/2 - 2\kappa + \kappa - 1/2 = 0 \tag{5.17a}$$

$$C_1 = \sum_{j=1}^{k} j\alpha_j - \sum_{j=0}^{k} \beta_j = (-2\kappa + 2\kappa + 1) - (\beta_1 + \beta_0) = 0 \tag{5.17b}$$

$$C_2 = \sum_{j=1}^{k} \frac{j^2}{2!}\alpha_j - \sum_{j=0}^{k} j\beta_j = 0.5(-2\kappa + 4\kappa + 2) - (\beta_1) = 0 \tag{5.17c}$$

From Eq. (5.17b) and Eq. (5.17c), it is obtained that $\beta_1 = \kappa+1$ and $\beta_0 = -\kappa$. Hence, the second order explicit method with a free parameter has the following form:

$$(\kappa + 1/2)\phi^{n+2} - 2\kappa\phi^{n+1} + (\kappa - 1/2)\phi^n = \Delta t\left[(\kappa + 1)f(\phi^{n+1}) - \kappa f(\phi^n)\right] \tag{5.18}$$

Notice that for $\kappa = 0.5$ the classical second order Adams-Bashforth is recovered. Repeating the process detailed previously in Section 5.2.1 to determine the region of absolute stability R_A for this method:

$$\phi^{n+2} = \frac{\bar{h} + \bar{h}\kappa + 2\kappa}{(\kappa + 1/2)}\phi^{n+1} - \frac{\bar{h}\kappa + \kappa - 0.5}{(\kappa + 1/2)}\phi^n = \chi\phi^{n+1} - \psi\phi^n \tag{5.19}$$

where $\chi = \left(\bar{h} + \bar{h}\kappa + 2\kappa\right) / (\kappa + 1/2)$ and $\psi = \left(\bar{h}\kappa + \kappa - 0.5\right) / (\kappa + 1/2)$. Analogously to Eq. (5.13), the eigenvalues of this linear system are given by:

$$\lambda_{\kappa 2LM} = \frac{\chi \pm \sqrt{\chi^2 + 4\psi}}{2} \tag{5.20}$$

Therefore, it is obvious that the absolute stability region $\mathbf{R_A}$ can now be modified via the bounded free parameter κ. The stability regions $\mathbf{R_A}$ for different values of κ are depicted in Fig. 5.2. Beginning at $\kappa = 0.5$ (the 2^{nd} order Adams-Bashforth), it is observed that when the free-parameter κ is reduced, the stability region $\mathbf{R_A}$ is pressed against the imaginary axis. This is a positive aspect if the linear system that is being solved has eigenvalues with a large imaginary component. On the other hand, if κ is increased, $\mathbf{R_A}$ is stretched along the real axis, being favourable for cases where the eigenvalues of the system have a large real part.

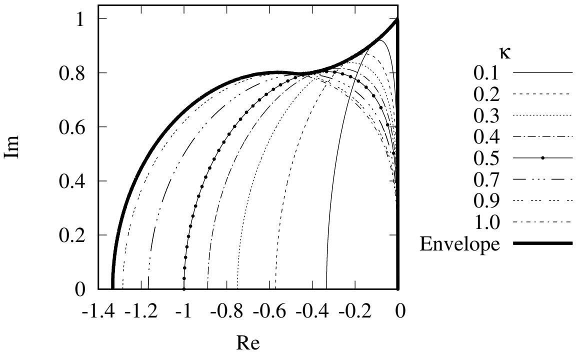


Figure 5.2: Stability Regions $\mathbf{R_A}$ for 2^{nd} order explicit LM method with free-parameter κ. The envelope depicts the maximum region of stability.

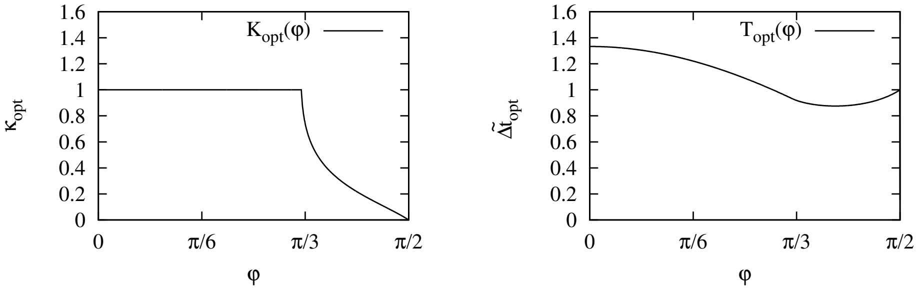
Time step and stability region maximisation algorithm

So far, the mathematical basis to construct a LM method with a free-parameter has been explained. As can be seen in Fig. 5.2, the optimal free-parameter κ will be a function of the system eigenvalues λ_i. Hence, the method is completed by defining two functions that return the optimal free-parameter $\kappa = K_{opt}(\varphi)$ and the *normalized* integration step $\widetilde{\Delta t} = T_{opt}(\varphi)$ as a function of the eigenvalue angle φ. Since the boundary of the stability region δR_A cannot be defined using a simple function, piecewise polynomial functions are fitted numerically. For example, the functions $K_{opt}(\varphi)$ and $T_{opt}(\varphi)$ for $\kappa \in (0, 1]$ and for the LM scheme developed in Section 5.2.2 are depicted in Fig. 5.3a and Fig. 5.3b, respectively. Hence, the basic steps are the following ones:

1. Determine the most restrictive eigenvalue λ_{max} of the system.
2. Evaluate φ as $\varphi = \frac{\pi}{2} - \arctan\left(\frac{\mathbb{R}(\lambda_{max})}{\mathbb{I}(\lambda_{max})}\right)$.
3. Find $\kappa = K_{opt}(\varphi)$ and $\widetilde{\Delta t} = T_{opt}(\varphi)$.
4. Solve the time integration step with $\Delta t = \widetilde{\Delta t} / |\lambda_{max}|$ and κ.

5.2.3 Evaluation of the system eigenvalues

The presented method requires the most restrictive system eigenvalue, denoted as λ_{max}, of the solved linear system (Eq. (5.2)). Unfortunately, except for few cases, matrix $[\mathbf{A}]$ changes every time step. Therefore, the largest system eigenvalue λ_{max}



(a) $K_{opt}(\varphi)$ function.

(b) $T_{opt}(\varphi)$ function.

Figure 5.3: Piecewise functions for 2^{nd} order explicit LM method with free parameter κ.

should be calculated at each iteration. Besides, the calculation of matrix's [**A**] eigenvalues would require a higher computational effort than solving the problem itself. Hence, calculating exactly λ_{max} is usually not a feasible option. An alternative approach is to bound the actual eigenvalues of the system. To perform this bounding, the **Gershgorin Circle Theorem** is used.

The Gershgorin Circle Theorem states that for a square matrix $[\mathbf{A}] = [a_{ij}] \in \mathcal{M}_{NxN}$ all the eigenvalues of [**A**] are located in the union of n discs:

$$\cup_{i=1}^{n} z \in \mathbb{C} : |z - a_{ii}| \leq R_i(A) \equiv G(A) \tag{5.21a}$$

where

$$R_i(A) = \sum_{\substack{j=1 \\ j \neq i}}^{n} |a_{ij}| \tag{5.21b}$$

denotes the *deleted absolute row sums* of [**A**]. Therefore, applying this theorem, the largest eigenvalue of matrix [**A**] can be bounded through matrix's [**A**] entries.

5.2.4 Application to Convection-Diffusion Equation

The technique described above can be applied to any linear system of equations. Still, the present work is focused on problems of heat transfer and fluid dynamics. The physical phenomena involved in this kind of problems is described by Partial Differential Equations (PDEs). In general, PDEs describing the transport of momentum, energy and species can be summarized in the Convection-Diffusion equation:

$$\frac{\partial \rho \phi}{\partial t} = -\nabla \cdot (\rho \mathbf{u} \phi) + \nabla \cdot (\Gamma (\nabla \phi)) + S(\phi) \tag{5.22}$$

where ϕ is the transported magnitude, ρ is the density, Γ is a diffusion coefficient and $\mathbf{u}$ is the velocity vector. The first term in the right-hand side of Eq. (5.22) is the convective transport, the second term accounts for the diffusive transport and the last one is the source term, $S(\phi)$. Except for very simple cases, Eq. (5.22) cannot be integrated analytically. Hence, numerical methods are required. In order to transform this PDE into a set of algebraic equations, the problem is discretized employing the Finite Volume Method (FVM) technique:

$$\mathbf{\Omega} \frac{\mathrm{d} \rho_h \boldsymbol{\phi_h}}{\mathrm{d}t} = -\mathbf{C}(\rho_h \mathbf{u_h}) \boldsymbol{\phi_h} + \mathbf{D}(\boldsymbol{\phi_h}) + \mathbf{S}(\boldsymbol{\phi_h}) \tag{5.23}$$

where the subindex $\mathbf{h}$ denotes discrete values, the diagonal matrix $\mathbf{\Omega}$ is the volume matrix, and matrices $\mathbf{C}$ and $\mathbf{D}$ are the convective and diffusive operators, respectively. The PDEs describing the motion of a fluid are the continuity and momentum equations, the Navier-Stokes (NS) equations. For a Newtonian fluid the NS equations read as:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \tag{5.24}$$

$$\frac{\partial \rho \mathbf{u}}{\partial t} = -\nabla \cdot (\rho \mathbf{u}\mathbf{u}) + \nabla \cdot (\boldsymbol{\tau}) - \nabla p \tag{5.25}$$

and the energy equation in its temperature form is:

$$\frac{\partial \rho T}{\partial t} = -\nabla \cdot (\rho \mathbf{u} T) + \frac{1}{c_p} \nabla \cdot (\lambda \nabla T) \tag{5.26}$$

where $\boldsymbol{\tau}$ is the viscous stress tensor, λ the thermal conductivity and c_p the specific heat capacity. Rewriting these equations in discrete form for constant physical properties and in the incompressible limit:

$$\mathbf{M}\mathbf{u_h} = 0 \tag{5.27}$$

$$\mathbf{\Omega} \frac{\mathrm{d} \mathbf{u_h}}{\mathrm{d}t} = -\mathbf{C}(\mathbf{u_h})\mathbf{u_h} + \nu \mathbf{D}(\mathbf{u_h}) - \frac{1}{\rho} \mathbf{\Omega} \mathbf{G} p_h \tag{5.28}$$

$$\mathbf{\Omega} \frac{\mathrm{d} T_h}{\mathrm{d}t} = -\mathbf{C}(\mathbf{u_h}) T_h + \alpha \mathbf{D}(T_h) \tag{5.29}$$

where $\mathbf{M}$ is the discrete divergence operator and $\mathbf{G}$ the discrete gradient operator. The kinematic viscosity is denoted by ν and the thermal diffusivity is defined as

$\alpha = \lambda/(\rho c_p)$. In order to handle the pressure-velocity coupling, the Fractional Step Method (FSM) method is applied. For the sake of simplicity, if the Euler method is employed, the FSM reads as:

$$\boldsymbol{\Omega}\frac{\mathbf{u_h^{pred}} - \mathbf{u_h^n}}{\Delta t} = -\mathbf{C}(\mathbf{u_h^n})\mathbf{u_h^n} + \nu\mathbf{D}(\mathbf{u_h^n}) \tag{5.30}$$

$$\mathbf{L}p_h^{n+1} = \frac{\rho}{\Delta t}\mathbf{M}\mathbf{u_h^{pred}} \tag{5.31}$$

$$\mathbf{u_h^{n+1}} = \mathbf{u_h^{pred}} - \frac{\Delta t}{\rho}\mathbf{G}p_h^{n+1} \tag{5.32}$$

where $\mathbf{L} = \mathbf{MG}$ is the discrete laplacian operator. Since the pressure term is treated implicitly, attention is placed on the terms treated explicitly, the convective $\mathbf{C}$ and diffusive $\mathbf{D}$ operators.

Properties of the convective and diffusive operators

The conservation properties of the discrete operators $\mathbf{C}$ and $\mathbf{D}$ depend on their symmetries. As shown by Verstappen and Veldman [12], kinetic energy is preserved if and only if the convective matrix is skew-symmetric, i.e $\mathbf{C} = -\mathbf{C}^*$. In order to construct a convective matrix operator that fulfils the skew-symmetric requirement, a Symmetry-Preserving scheme is employed [12]. The diffusive matrix operator, typically constructed using a central difference scheme, is symmetric and negative-definite. Hence, the diffusive matrix is equal to its transpose $\mathbf{D} = \mathbf{D}^*$ and $\mathbf{z}^*\mathbf{D}\mathbf{z} < 0$ for any non-zero $\mathbf{z}$. Therefore, focus is placed on applying the Gershgorin Circle Theorem to bound the eigenvalues of the system matrix $\mathbf{A} = -\mathbf{C} + \mathbf{D}$. It is known that:

- $\forall$ skew-symmetric matrices $\mathbf{C}$, all eigenvalues λ^C are $\mathbb{I}$.
- $\forall$ negative-definite matrices $\mathbf{D}$, all eigenvalues λ^D are $\mathbb{R}$ and $\lambda^D < 0$.

Eigenvalue bounding of the Convection-Diffusion equation

The properties of the convective and diffusive operator together with the Gershgorin Circle Theorem can be exploited to better bound the eigenvalues of matrix $\mathbf{A}$. First, instead of applying the Gershgorin theorem directly to matrix $\mathbf{A}$, the theorem is applied to matrices $\mathbf{C}$ and $\mathbf{D}$. Second, according to the Theorem attributed to Bendixson [17], the eigenvalue with the biggest complex modulus of the system matrix $\mathbf{A}$ is bounded as:

$$\max\left(\left|\mathbb{R}(\lambda_i^A)\right|\right) \leq \max\left(\left|\lambda_i^D\right|\right) \quad \& \quad \max\left(\left|\mathbb{I}(\lambda_i^A)\right|\right) \leq \max\left(\left|\lambda_i^C\right|\right) \tag{5.33}$$

Hence, defining $\left|\lambda^C_{max}\right| = \max\left(\left|\lambda^C_i\right|\right)$ and $\left|\lambda^D_{max}\right| = \max\left(\left|\lambda^D_i\right|\right)$, and exploiting the operators properties introduced in the preceding section, the bounded value of the most restrictive eigenvalue of the matrix $\mathbf{A}$ is defined as:

$$\lambda^A_{bounded} = -|\lambda^D_{max}| + i|\lambda^C_{max}| \tag{5.34}$$

In order to asses and study how this technique bounds and estimate the eigenvalue with the biggest modulus of the system matrix $\mathbf{A}$, a rigorous numerical analysis has been performed. Several pairs of convective and diffusive matrices of different sizes have been randomly generated. For each pair of $\mathbf{C}$ and $\mathbf{D}$, the following linear combination is defined $\mathbf{A} = -w_1\mathbf{C} + w_2\mathbf{D}$, where $w_i \in [0, 1]$. The real and imaginary part of the eigenvalue with the biggest complex modulus of matrix $\mathbf{A}$ are normalized by $|\lambda^D_{max}|$ and $|\lambda^C_{max}|$, respectively. Results are depicted in Fig. 5.4. As can be seen, the numerical results are in agreement with Eq. (5.33). Therefore, the most restrictive eigenvalue of $\mathbf{A}$ is always bounded by the eigenvalues of $\mathbf{C}$ and $\mathbf{D}$ with the biggest absolute value, as stated by Bendixson [17]. Notice that when one of the absolute values of the eigenvalues is much bigger than the other one ($\left|\lambda^D_{max}\right| \gg \left|\lambda^C_{max}\right|$ or $\left|\lambda^D_{max}\right| \ll \left|\lambda^C_{max}\right|$), the absolute value of the most restrictive eigenvalue of matrix $\mathbf{A}$, $\left|\lambda^A_{max}\right|$, has a similar absolute value than the most restrictive eigenvalue of matrices $\mathbf{C}$ and $\mathbf{D}$. On the other hand, if the complex modulus of both convective and diffusive matrices are similar ($\left|\lambda^D_{max}\right| \sim \left|\lambda^C_{max}\right|$), the absolute value of the most restrictive eigenvalue of matrix $\mathbf{A}$ will be smaller than the one estimated by this bounding technique $|\lambda^A_{max}| < |\lambda^A_{bounded}|$, where $|\lambda^A_{bounded}| = \sqrt{|\lambda^D_{max}|^2 + |\lambda^C_{max}|^2}$. Therefore, a consequence is that the estimated time-step is on the safe side. The complex modulus of the actual system's eigenvalues are always equal or lower than the ones estimated by the bounding technique. Hence, the resulting time-step calculated using $\lambda^A_{bounded}$ is equal or smaller than the one that would be calculated if λ^A_{max} were employed.

Moreover, this technique allows a proper estimation of position in the complex plane (distance and angle from the origin) of the most restrictive system matrix eigenvalue λ^A_{max}, which is not possible if the Gershgorin theorem is applied directly to matrix $\mathbf{A}$.

Accounting for the volume matrix

The final system matrix for a convection-diffusion equation discretised using the FVM technique, neglecting the source term, is $\mathbf{\Omega^{-1}A} = -\mathbf{\Omega^{-1}C} + \mathbf{\Omega^{-1}D}$ (see Eq. (5.23)). The volume matrix $\mathbf{\Omega}$ is a diagonal invertible matrix. Consequently, the product $\mathbf{\Omega^{-1}C}$ is not a skew-symmetric matrix. Similarly, the product $\mathbf{\Omega^{-1}D}$ is not a symmetric matrix. Hence, a priori, the properties that state $\forall\lambda^D \in \mathbb{R}$ and $\forall\lambda^C \in \mathbb{I}$ are not valid any more. However, in the following it is shown that all eigenvalues of a skew-symmetric matrix pre-multiplied by a diagonal matrix also lie on the imaginary axis. Analogously, it is also shown that all eigenvalues of a symmetric

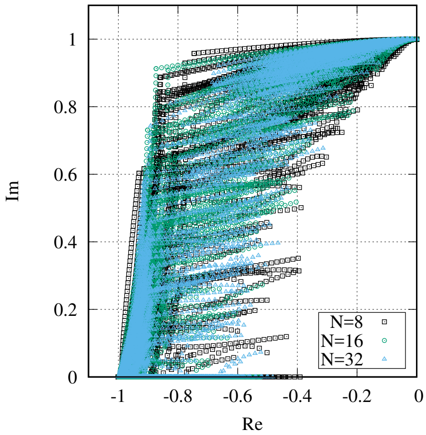


Figure 5.4: Most restrictive eigenvalues of matrix $\mathbf{A}$ normalized by $|\lambda^D_{max}|$ and $|\lambda^C_{max}|$.

matrix pre-multiplied by a diagonal matrix are real. Let $\mathbf{W}$ be a symmetric or a skew-symmetric matrix, and $\mathbf{d}$ a diagonal matrix:

$$\mathbf{dWx} = \lambda\mathbf{x} \tag{5.35}$$

where $\mathbf{x}$ represents the eigenvectors and λ the eigenvalues. The aim is to prove that all the eigenvalues of the system $\mathbf{dW}$ are real when $\mathbf{W}$ is symmetric, and that all the eigenvalues λ are purely imaginary if $\mathbf{W}$ is skew-symmetric. Beginning from Eq. (5.35) and post-multiplying $\mathbf{W}$ by the identity matrix written as $\mathbf{I} = \mathbf{d}^{1/2}\mathbf{d}^{-1/2}$:

$$\mathbf{dWd}^{1/2}\mathbf{d}^{-1/2}\mathbf{x} = \lambda\mathbf{x} \tag{5.36a}$$

$$\mathbf{d}^{1/2}\mathbf{d}^{1/2}\mathbf{Wd}^{1/2}\mathbf{d}^{-1/2}\mathbf{x} = \lambda\mathbf{x} \tag{5.36b}$$

Pre-multiplying the last expression by $\mathbf{d}^{-1/2}$ at both sides:

$$\mathbf{d}^{-1/2}\mathbf{d}^{1/2}\mathbf{d}^{1/2}\mathbf{Wd}^{1/2}\mathbf{d}^{-1/2}\mathbf{x} = \lambda\mathbf{d}^{-1/2}\mathbf{x} \tag{5.37}$$

Finally, let $\mathbf{d}^{-1/2}\mathbf{x} = \mathbf{x}'$ and since $\mathbf{d}^{-1/2}\mathbf{d}^{1/2} = \mathbf{I}$, the previous expression becomes:

$$\mathbf{d}^{1/2}\mathbf{Wd}^{1/2}\mathbf{x}' = \lambda\mathbf{x}' \tag{5.38}$$

Obviously, if $\mathbf{W}$ were symmetric, the product $\mathbf{d}^{1/2}\mathbf{Wd}^{1/2}$ is also symmetric, and the same applies in case $\mathbf{W}$ were skew-symmetric. Consequently, since $\mathbf{d}^{1/2}\mathbf{Wd}^{1/2}$ has the same eigenvalues λ as $\mathbf{dW}$, it has been proved that all the eigenvalues of $\mathbf{\Omega}^{-1}\mathbf{C}$ are pure imaginary, and that all the eigenvalues of $\mathbf{\Omega}^{-1}\mathbf{D}$ are pure real.

In case the system is pre-multiplied by more than one diagonal matrix, the same proof could be extended indefinitely:

$$\mathbf{d_1 d_2 ... d_n W x} = \lambda \mathbf{x} \tag{5.39a}$$

$$\mathbf{d_1}^{1/2}\mathbf{d_2}^{1/2}...\mathbf{d_n}^{1/2}\mathbf{W}\mathbf{d_1}^{1/2}\mathbf{d_2}^{1/2}...\mathbf{d_n}^{1/2}\mathbf{x}'' = \lambda \mathbf{x}'' \tag{5.39b}$$

Only the eigenvectors of the system will be modified, but the eigenvalues remain unchanged.

Eigenvalue estimation of the Convective and Diffusive operators

As previously stated, the system eigenvalues can be bounded through the maximum eigenvalues of the convective $\mathbf{C}$ and diffusive $\mathbf{D}$ operators. In the following, the evaluation of the maximum eigenvalue of each operator using the Gershgorin Circle Theorem is detailed. Initially, only discretisations which result in a skew-symmetric convective operator and a symmetric diffusive operator are considered. Afterwards, other discretisation schemes are analysed.

Convective operator The finite-volume discretisation of the convective term in a generic control volume i with cell volume V_i is:

$$C_i = \int_{V_i} (\rho \mathbf{u} \phi) dV_i = \sum_{f \in F(i)} \rho_f \mathbf{u_f} \phi_f \cdot \mathbf{n_f} A_f = \sum_{f \in F(i)} \dot{m}_f \phi_f \tag{5.40}$$

where subindex f indicates a value evaluated at a face of the control volume, $F(i)$ is a set of all faces of a control volume, $\mathbf{n_f}$ is the face normal vector and A_f is the surface area. The face mass flow is denoted by $\dot{m}_f$. The discretisation scheme chosen to evaluate ϕ_f will impact on the properties of the discrete convective operator and also will affect the matrix entries of the convective matrix $\mathbf{C}$. First, it is here considered a Symmetry-Preserving scheme, which, as the name indicates, preserves the skew-symmetry of the convective operator. Beginning with a Symmetry-Preserving scheme ($\phi_f = 0.5(\phi_i + \phi_j)$), where subindex j denotes the value in the neighbour cell, the convective matrix reads as:

$$\mathbf{C} = \sum_{f \in F(i)} 0.5 \dot{m}_f (\phi_i + \phi_j) \tag{5.41}$$

Therefore, the diagonal elements of the convective operator ($\mathbf{C}$) will be:

$$C_{ii} = 0.5 \sum_{f \in F(i)} \dot{m}_f \tag{5.42}$$

Considering a case with constant density, the null velocity divergence $\nabla\cdot\mathbf{u}=0$ results in all diagonal elements of matrix $\mathbf{C}$ being 0. On the other hand, the off-diagonal elements are:

$$C_{ij} = 0.5\dot{m}_{ij} \tag{5.43}$$

As stated before, the resulting matrix is skew-symmetric ($C_{ij} = -C_{ji}$). Following, applying the Gershgorin Circle Theorem to matrix $\mathbf{C}$ for row i results in:

$$|\lambda_i - C_{ii}| \leq \sum_{\substack{j=1 \\ j\neq i}}^{n} |C_{ij}| \tag{5.44}$$

Since $C_{ii}=0$, the Gershgorin circle is centred at the complex origin. Hence, all the eigenvalues, which are pure imaginary, lie inside the biggest radius, it is:

$$|\lambda^C_{max}| \leq \max \sum_{\substack{j=1 \\ j\neq i}}^{n} |C_{ij}|, \quad \text{for } i = 1...n \tag{5.45}$$

Diffusive operator Regarding the diffusive operator, the finite volume discretisation in a generic control volume i reads as:

$$D_i = \int_{V_i} \nabla\cdot(\Gamma\nabla\phi)dV_i = \sum_{f\in F(i)} \Gamma_f \nabla\phi_f \cdot \mathbf{n_f} A_f = \\ \sum_{f\in F(i)} \Gamma_f \left(\frac{\partial\phi}{\partial n}\right)_f A_f \approx \sum_{f\in F(i)} \Gamma_f \left(\frac{\phi_j - \phi_i}{d_{ij}}\right) A_f \tag{5.46}$$

where a 2^{nd} order Central Difference Scheme (CDS) has been employed. The distance d_{ij} is the distance between nodes i and j projected in the normal direction to the face, $d_{ij} = (\mathbf{s}_f\mathbf{n}_f)$, where $\mathbf{s}_f$ is the vector from the centroid of i control volume to the centroid of the neighbour control volume j. For the sake of simplicity, employing $d_w = \frac{A_f}{d_{ij}}$, Eq. (5.46) can be rewritten as:

$$D_i = \sum_{f\in F(i)} \Gamma_f d_w(\phi_j - \phi_i) \tag{5.47}$$

Therefore, the diagonal and off-diagonal terms of the diffusive operator $\mathbf{D}$ are:

$$D_{ii} = -\sum_{f\in F(i)} \Gamma_f d_w \tag{5.48}$$

$$D_{ij} = \Gamma_{ij} d_w \tag{5.49}$$

where the diagonal elements can also be expressed as $D_{ii} = -\sum_{f \in F(i)} D_{ij}$. If the Gershgorin Circle Theorem is applied to the diffusive matrix $\mathbf{D}$, the following expression is obtained:

$$|\lambda_i - D_{ii}| \leq \sum_{\substack{j=1 \\ j \neq i}}^{n} |D_{ij}| \tag{5.50}$$

Since the diagonal term D_{ii} is negative by definition, it is known that the Gershgorin circle centre will be placed at the left of the imaginary axis. Therefore, the eigenvalue with the biggest complex modulus is negative and its absolute value is:

$$|\lambda^D_{max}| \leq \max\left(|D_{ii}| + \sum_{\substack{j=1 \\ j \neq i}}^{n} |D_{ij}|\right), \quad \text{for } i = 1...n \tag{5.51}$$

System Matrix maximum eigenvalue bounding Once $|\lambda^C_{max}|$ and $|\lambda^D_{max}|$ have been estimated, the most restrictive eigenvalue of the system defined according to Eq. (5.34) is calculated, and the optimal time-step Δt is determined following the Self-Adaptive Time Integration Method, detailed in Section 5.2.2.

5.2.5 Dynamic eigenvalue bounding methodology extension

The bounding methodology introduced so far applies for skew-symmetric convective operators $\mathbf{C}$. However, there remain the question whether the described strategy can also be applied to convective operators which are not skew-symmetric. The skew-symmetric property of the convective operator matrix is lost in variable density cases ($\sum_f \dot{m}_f \neq 0$, hence $C_{ii} \neq 0$ in general) or employing non energy conservative discretisation schemes like Upwind, QUICK [18] or SMART [19]. Nevertheless, any square matrix $\mathbf{W}$ can be split in the sum of two matrices, its symmetric part plus its skew-symmetric part:

$$\mathbf{W} = symm(\mathbf{W}) + skew(\mathbf{W}) \tag{5.52a}$$

$$symm(\mathbf{W}) = \frac{\mathbf{W} + \mathbf{W^T}}{2} \tag{5.52b}$$

$$skew(\mathbf{W}) = \frac{\mathbf{W} - \mathbf{W^T}}{2} \tag{5.52c}$$

Since the sum of two symmetric matrices is another symmetric matrix, the analysis presented before for the system $\mathbf{A} = -\mathbf{C} + \mathbf{D}$ (when $\mathbf{C}$ skew-symmetric and $\mathbf{D}$

symmetric) can also be applied to the system $\mathbf{A} = -\mathbf{A}_{\mathbb{I}} + \mathbf{A}_{\mathbb{R}}$, where $\mathbf{A}_{\mathbb{I}} = skew(\mathbf{C})$ and $\mathbf{A}_{\mathbb{R}} = -symm(\mathbf{C}) + \mathbf{D}$. In other words, the symmetric (diffusive) part of the convective operator $\mathbf{C}$ is shifted to the diffusive operator $\mathbf{D}$. In Appendix A further detail is given regarding the evaluation of the symmetric and skew-symmetric parts of the convective operator for Upwind and QUICK schemes.

5.2.6 Extension to other temporal integration schemes

So far, the self-adaptive strategy has been introduced for explicit Linear Multi-step methods. Specifically, the construction of an explicit 2^{nd} order LM method with a free parameter has been detailed. Still, as previously stated, other temporal schemes are commonly used. In the following, the self-adaptive strategy is discussed in the context of Predictor-Corrector schemes and Runge-Kutta schemes.

Predictor-Corrector methods

Predictor-corrector (PC) schemes are based on the combination of an explicit method (the predictor step) and an implicit method (the corrector step). PC methods exploit the low computational cost of explicit methods and the larger stability domain of implicit schemes. A PC scheme of order k works as follows: first, a predictor step is computed employing an explicit scheme, obtaining a predicted value ϕ^* at time t^{n+k}. Then, a second step is calculated using an implicit scheme, but instead of iterating to calculate the converged value of $f(t^{n+k}, \phi^{n+k})$, the function at time t^{n+k} is evaluated using the predicted value as $f(t^{n+k}, \phi^*)$. The simplest Predictor-Corrector time integration scheme is one using an explicit Euler as predictor step $u^* = u^n + \Delta t[f(u^n)]$ followed by a corrector step employing a pseudo-implicit Euler step $u^{n+1} = u^n + \Delta t[f(u^*)]$.

As detailed by Süli and Mayers [16], if the predictor step has an accuracy order p^* and the corrector step one of p, the predictor-corrector method will have an order of accuracy of p. Hence, the development of a 2^{nd} Order Predictor/2^{nd} Order Corrector scheme ($\kappa 2P2C$) with a free parameter κ is investigated.

For the predictor step, the results obtained in Section 5.2.2 for the 2^{nd} order LM scheme are recovered (see Eq. (5.18)), and replacing ϕ^{n+2} by ϕ^* gives:

$$(\kappa + 1/2)\phi^* - 2\kappa\phi^{n+1} + (\kappa - 1/2)\phi^n = \Delta t \left[(\kappa + 1)f(y^{n+1}) - \kappa f(y^n)\right] \tag{5.53}$$

The process carried out for the LM method is now repeated for an implicit second order scheme. Therefore, analogously to Eq. (5.16), the second order implicit method will have the form:

$$\alpha_2\phi^{n+2} + \alpha_1\phi^{n+1} + \alpha_0\phi^n = \Delta t \left[\beta_2 f(\phi^*) + \beta_1 f(\phi^{n+1})\right] \tag{5.54}$$

where the α_i are defined as before, and $\beta_0 = 0$ is imposed. Therefore,

$$C_1 = \sum_{j=1}^{k} j\alpha_j - \sum_{j=0}^{k} \beta_j = (-2\kappa + 2\kappa + 1) - (\beta_2 + \beta_1) = 0 \tag{5.55a}$$

$$C_2 = \sum_{j=1}^{k} \frac{j^2}{2!}\alpha_j - \sum_{j=0}^{k} j\beta_j = 0.5(-2\kappa + 4\kappa + 2) - (2\beta_2 + \beta_1) = 0 \tag{5.55b}$$

From Eqs. (5.55a) and (5.55b), it is obtained that $\beta_2 = \kappa$ and $\beta_1 = 1 - \kappa$. Hence, the second order implicit method with a free parameter κ has the form:

$$(\kappa + 1/2)\phi^{n+2} - 2\kappa\phi^{n+1} + (\kappa - 1/2)\phi^n = \Delta t\left[(\kappa)f(\phi^*) + (1-\kappa)f(\phi^{n+1})\right] \tag{5.56}$$

Following, in order to determine the stability region of the developed PC method, Eq. (5.53) is substituted into Eq. (5.56), which after operating results in:

$$\phi^{n+2} = \frac{\bar{h}\kappa\chi + \bar{h}(1-\kappa) + 2\kappa}{(\kappa + 1/2)}\phi^{n+1} + \frac{\bar{h}\kappa\psi - (\kappa - 0.5)}{(\kappa + 1/2)}\phi^n \tag{5.57}$$

$$\phi^{n+2} = \chi_2\phi^{n+1} - \psi_2\phi^n \tag{5.58}$$

where χ, ψ are the same as in Eq. (5.19).

As carried out previously, the constraint $|\lambda_{\kappa 2P2C}| \leq 1$ is applied in order to assure that the PC method is absolutely stable. The obtained stability region R_A as function of the free-parameter κ is depicted in Fig. 5.5. As can be seen, the behaviour of this developed PC method differs from the 2^{nd} order explicit LM method developed before. In the $\kappa 2LM$ method for small values of κ the stability region is stretched against the imaginary axis, and increasing the value of κ, the stability region grows in the real axis direction. Meanwhile, it is observed that the maximum stability region of the PC method has a change in its behaviour for $\kappa = 0.14809$ (see Fig. 5.6). For lower values of κ, the behaviour is similar to the $\kappa 2LM$ method, since decreasing κ the stability region is pressed against the imaginary axis. For $\kappa = 0.14809$ the PC method reaches the maximum stability region in the real axis. For bigger values of κ, the stability region increases in the imaginary axis at the same time that decreases in the real axis when κ is incremented. Nevertheless, as for the $\kappa 2LM$ method, the benefits of employing a free-parameter κ are also really remarkable. For cases with eigenvalues only with real component, the step size can be increased more than two times with respect to the method with a fixed $\kappa = 0.5$, namely a PC method employing a 2^{nd} order Adams-Bashforth in the predictor step and an Adams-Moulton of 2^{nd} order in the corrector step.

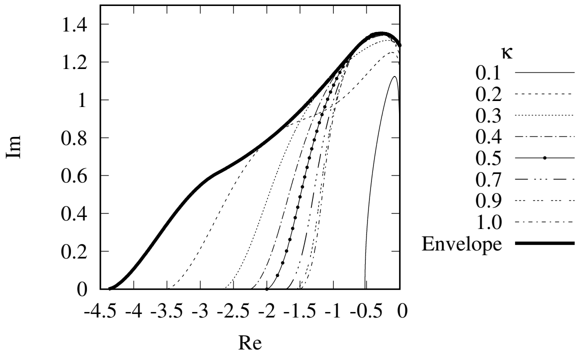


Figure 5.5: Stability Regions $\mathbf{R_A}$ for 2^{nd} Order Predictor-Corrector Method with free-parameter κ.

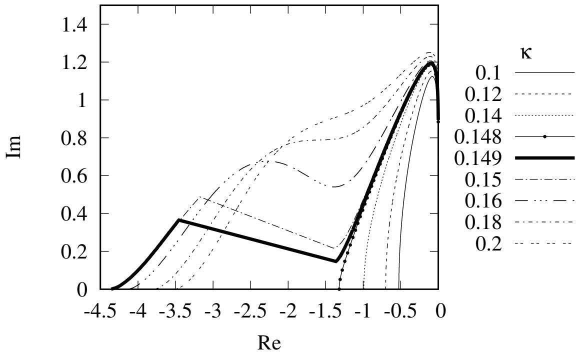


Figure 5.6: Stability Regions $\mathbf{R_A}$ for 2^{nd} Order Predictor-Corrector Method around $\kappa = 0.14809$.

Runge-Kutta methods

Runge-Kutta (RK) schemes are a family of numerical integration methods based on the evaluation of midpoints in the interval $f(t^n, \phi^n)$ and $f(t^{n+1}, \phi^{n+1})$. RK methods do not require previous time steps in order to be employed, like the 1^{st} order Euler explicit scheme, but achieving high order accuracy through evaluating $f(t, \phi)$ at intermediate steps between t^n and t^{n+1}. The family of explicit RK methods has the form:

$$\boldsymbol{\phi}^{n+1} = \boldsymbol{\phi}^{n} + \Delta t \sum_{s=1}^{S} b_s k_s \tag{5.59a}$$

$$k_1 = f(t^n, \boldsymbol{\phi}^n) \tag{5.59b}$$

$$k_s = f(t^n + \Delta t c_r, \boldsymbol{\phi}^n + \Delta t \sum_{r=1}^{s-1} a_{sr} k_r), \quad s = 2, ..., S \tag{5.59c}$$

where $\sum_{s=1}^{S} b_s k_s$ is a weighted average of function values $f(t, \phi)$ at the interval $t^n \leq t \leq t^{n+1}$ and S is the number of internal sub-steps of the method. A RK method will be consistent if and only if $\sum_{s=1}^{s} b_s = 1$. As shown by Lambert [20], if a RK method has the same accuracy order p than the number of steps S, Eq. (5.59) can be linealized with $f(\phi_i) \approx \lambda_i \phi_i$ as:

$$\phi_i^{n+1} = \left[\sum_{n=0}^{S} \frac{\overline{h_i}^{n}}{n!} \right] \phi_i^n \tag{5.60}$$

Therefore, the stability region of these RK methods is defined by:

$$r_s(\bar{h}) = \lambda_{RK} = \sum_{n=0}^{S} \frac{\bar{h}^n}{n!} \tag{5.61}$$

where $\bar{h} = h\lambda_{max}$. From Eq. (5.61) and Definition Definition 1, the stability region for RK methods up to $S \leq 4$ is depicted in Fig. 5.7.

In contrast to the Linear Multistep and Predictor-Corrector methods presented before, the introduced RK methods do not include a free-parameter κ. Hence, the RK method is not modified in order to increase the stability region as a function of the system eigenvalues. Nevertheless, as will be shown in section Section 5.3.3, the method developed in the present work still applies. When RK schemes are employed, the method evaluates the stability region and estimates the maximum stable time-step Δt for each iteration as a function of the system eigenvalues.

5.3 Numerical results

Numerical simulations are presented in the following to evidence the capacity of the self-adaptive strategy to be used with the different numerical schemes described so far. In all cases a Differentially Heated Cavity (DHC) is used as a test case. The DHC case features a square enclosure, with unit length, where the vertical left and

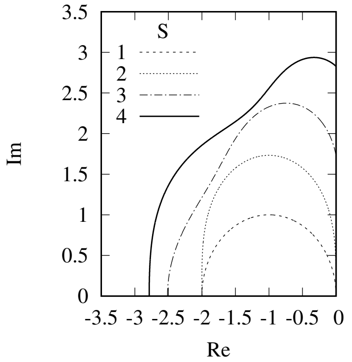


Figure 5.7: Stability Region R_A for RK Schemes with $S = p$.

right walls are at a hot and cold temperature, respectively. Bottom and top walls are adiabatic. In this case flow motions are governed by natural convection. Hence, a buoyancy force is added to the momentum equation Eq. (5.25). In the present work the buoyancy term is modelled using the Boussinesq approximation:

$$(\rho - \rho_{ref})\mathbf{g} \approx \beta(T - T_{ref})\mathbf{g} \tag{5.62}$$

where $\mathbf{g}$ is the gravity, T_{ref} is a reference temperature and $\beta = 1/T_{ref}$ is the thermal expansion coefficient for ideal gases.

Results for four numerical tests are presented. In the first one, the eigenvalues bounding is checked by comparing the values computed with the dynamic procedure and the exact eigenvalues. Second, numerical simulations using different numerical schemes for the convective term are presented. It is shown how changing the discretisation scheme affects the eigenvalues and the resulting allowable time-step. The third test case involves changing the time-integration scheme. The effects on the allowable time-step are presented. In the last test a variable density DHC is simulated where the buoyancy is evaluated using the left hand side of Eq. (5.62).

To perform the numerical simulations a finite-volume approach is used to solve the different transport equations, particularly 2D collocated meshes [13, 21]. In the construction of the discrete convective term of the momentum equation a Symmetry-Preserving (SP) scheme [12] is employed in order to preserve kinetic energy. For the convective term in the energy equation, besides the SP scheme two upwind-like schemes are also considered, a first order Upwind and a QUICK scheme. A second order centred difference scheme is used to construct the discrete diffusive term for all transported quantities. The Poisson equation appearing in the FSM, Eq. (5.31), is solved by means of a Schur decomposition. Numerical computations are performed using the general purpose unstructured and parallel object-oriented CFD code TermoFluids [22].

5.3.1 Eigenvalues estimation

The purpose of the current section is to assess the capability of the method to bound correctly the eigenvalues in a real case. In order to do so, the following eigenvalues are compared: (i) the most restrictive eigenvalue calculated using the dynamic bounding methodology detailed in section Section 5.2.4, (ii) the exact eigenvalues of matrices $\mathbf{C}$ and $\mathbf{D}$, (iii) the eigenvalues of matrices $\mathbf{A}_{\mathbb{I}} = skew(\mathbf{C})$ and $\mathbf{A}_{\mathbb{R}} = -symm(\mathbf{C}) + \mathbf{D}$ and (iv) the eigenvalues of matrix $\mathbf{A}$. The eigenvalues of matrices $\mathbf{C}$, $\mathbf{D}$, $\mathbf{A}_{\mathbb{I}}$, $\mathbf{A}_{\mathbb{R}}$ and $\mathbf{A}$ have been calculated employing the scientific software Scilab [23].

The DHC problem is solved in non-dimensional form, using a Prandtl number of 0.71 and the Rayleigh number has been set to $5 \cdot 10^6$. The domain has been discretised using a Cartersian non-uniform mesh stretched at the hot and cold boundaries. The mesh has 32^2 control volumes (CV). This coarse mesh allows calculating the exact eigenvalues of the system. In the DHC problem, the transport equations for momentum and energy have to be solved. For the current case the time step Δt is limited by the energy equation, therefore, the present analysis has been done for the energy equation.

Results obtained employing the Symmetry-Preserving scheme are depicted in Fig. 5.8. As can be seen, the maximum eigenvalue calculated using the dynamic methodology ($\lambda^{bounded}$) perfectly bounds all the eigenvalues of the system matrix $\mathbf{A}$ (black dots), including the most limiting one (λ^{A}). The eigenvalue $\lambda^{A_{Re,Im}}$ represents the maximum eigenvalue calculated as $\lambda^{A_{Re,Im}} = (-\max(\left|\lambda^{A_{\mathbb{R}}}\right|) + i\max(\left|\lambda^{A_{\mathbb{I}}}\right|))$, namely the eigenvalue obtained from the maxima eigenvalues of matrices $\mathbf{A}_{\mathbb{R}}$ and $\mathbf{A}_{\mathbb{I}}$. As stated before, if the SP scheme is employed for the convective operator, all its eigenvalues (yellow diamonds) lie in the imaginary axis. Therefore, the eigenvalues of its skew-symmetric (blue crosses) part are the same, and the eigenvalues of its symmetric part (purple squares) are null. Since the eigenvalues of $symm(\mathbf{C})$ are null, the eigenvalues of the diffusive operator (green circles) are the same as the eigenvalues of $\mathbf{A}_{\mathbb{R}}$ (red asterisks).

Results obtained using the Upwind Difference Scheme (UDS) for the convective operator in the energy equation are shown in Fig. 5.9. Unlike the SP scheme, in this case the eigenvalues of the convective operator have also a real part, thus revealing the diffusive behaviour of the UDS schemes. Comparing figures Fig. 5.8 and Fig. 5.9, it can be noticed that the skew-symmetric part of the convective matrix $\mathbf{C}$ in both cases is the same, as detailed in Appendix A. Although the eigenvalues of the diffusive operator are the same, now the highest eigenvalue of $\mathbf{A}_{\mathbb{R}}$ is bigger than when using SP, since the diffusive part of the convective operator has been included. In the same vein, the eigenvalues of the system matrix $\mathbf{A}$ are now shifted to the diffusive part.

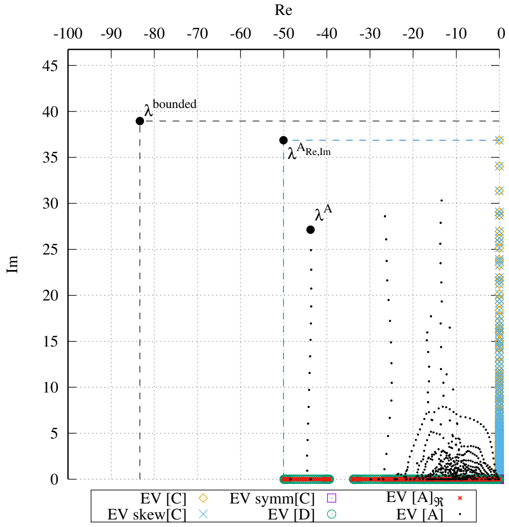


Figure 5.8: Eigenvalues employing the Symmetry-Preserving scheme.

As before, the most limiting eigenvalue (λ^A) is well-bounded by the the maximum eigenvalue estimated ($\lambda^{bounded}$).

5.3.2 Discretisation schemes

The next test consists of changing the discretisation scheme of the convective term of the energy equation. Simulations are run using the 2^{nd} order explicit Linear Multi-step method with a free parameter presented in Section 5.2.2, namely $K2LM$. An incompressible DHC case with $Ra = 10^7$ and $Pr = 0.71$ has been computed on a structured mesh of 128x64 CVs, also stretched close to the vertical walls. Transient simulations are run beginning from an initial state with a linear temperature profile and null velocities. Transient evolutions are expressed in terms of the case characteristic velocity and time, which are computed through:

$$u_{ref} = \frac{\alpha(Ra^{0.5})}{L_{ref}} \tag{5.63a}$$

$$\tau_{ref} = \frac{L_{ref}}{u_{ref}} \tag{5.63b}$$

where $L_{ref} = 1$.

Figures 5.10 to 5.12 show the φ angle, as defined in Section 5.2.2, and the computed

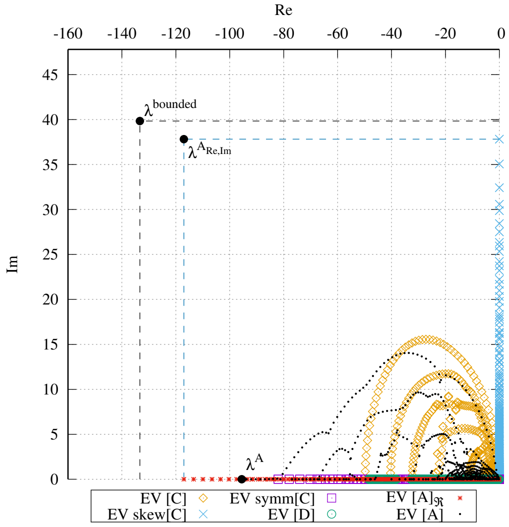


Figure 5.9: Eigenvalues employing the Upwind Difference scheme.

time step Δt obtained using SP, UDS and QUICK schemes, respectively. Moreover, the required Courant number C is also depicted, in case Courant–Friedrichs–Lewy condition (CFL) would be employed to estimate the time-step. The Courant number C is split into its different contributions, denoted here as *convective* C_c and *diffusive* C_d, and are evaluated as:

$$C_c = \frac{u\Delta t}{\Delta x} \tag{5.64a}$$

$$C_d = \frac{\Gamma\Delta t}{\Delta x^2} \tag{5.64b}$$

where Γ represents the diffusive coefficient $\Gamma = \nu$ or $\Gamma = \alpha$, of the momentum and energy equations, respectively.

Regarding φ, values close to 90 indicate a convective behaviour, since such an angle value indicates that the convective eigenvalue dominates. Focusing on the temporal variation of φ corresponding to the momentum equation, in all cases it can be observed that in the first time-units the flow features a transition from a pure diffusive behaviour to a more ”convective” one. The reason for this behaviour is that initially all the cavity is at the same temperature and the flow has null velocity. Hence, initially heat transfer is only through diffusion, namely conduction.

Regarding the energy equation, it can be observed that when the SP scheme is

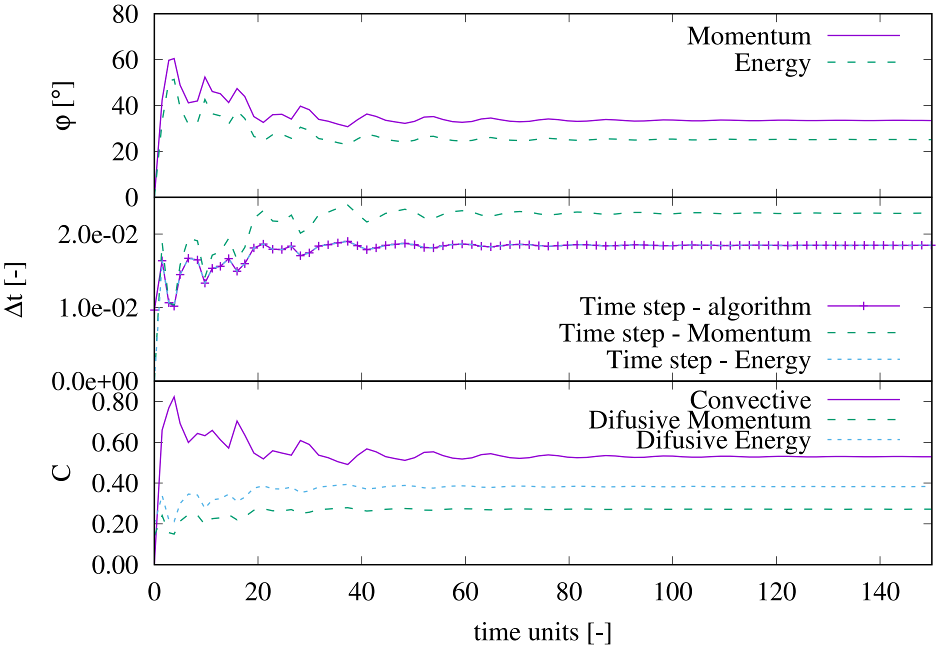


Figure 5.10: Transient evolution of the stability angle φ (top), Δt (middle) and Courant number C (bottom) obtained using a SP scheme. Time-step curves at the middle plot refer to the maximum allowable Δt for each equation. The time-step is expressed in non-dimensional form using τ_{ref}. Then, the lowest Δt of the two is selected at each iteration (Time step - algorithm). Courant numbers C are evaluated following Eq. (5.64).

used, a similar behaviour to the momentum equation is obtained. This is consistent because in this case the same numerical scheme is being used for the momentum and energy equations. However, when upwind-like schemes are used, a different temporal evolution is found. The difference between SP and the two upwind-like schemes can be explained by the diffusive behaviour introduced by the latter schemes, which do not preserve the skew-symmetry of the convective operator. Furthermore, comparing the results of the UDS and QUICK schemes, it can be seen that the φ angle for the UDS is 10° closer to the real axis than for the QUICK scheme. Thus, showing the more diffusive behaviour of the UDS scheme.

Considering the evaluated Δt and Courant number C, again a difference between the SP and upwind-like schemes can be observed. Focusing on the values related to

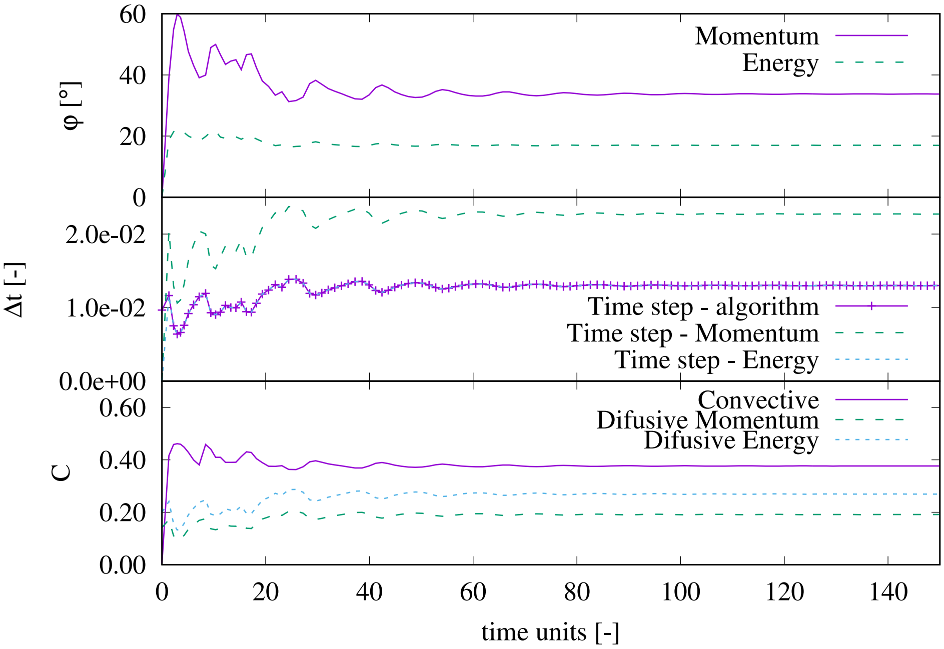


Figure 5.11: Transient evolution of the stability angle φ (top), Δt (middle) and Courant number C (bottom) obtained using an UDS scheme. See Fig. 5.10 for further details.

the energy equation, the SP scheme results in a higher time-step than either upwind-like scheme. However, although this higher allowable time-step with the SP for the present case, in cases where the local Peclet may become high, using this centred scheme in the scalars' transport equation may not be suitable [24], as it may lead to instabilities. Hence, in those cases upwind-like or flux-limiters would be required.

As a last remark, the time-step obtained with the QUICK scheme is lower than that obtained with the UDS scheme. The reason for this smaller time-step is attributed to the higher eigenvalues of the QUICK scheme (not shown here). Consequently, with larger eigenvalues, the resulting time-step cannot be as high as with the UDS, which features lower eigenvalues for this case. Nonetheless, results with the UDS show a more diffusive behaviour. Hence, in terms of accuracy of the solution, the QUICK scheme is better, since it introduces lower numerical diffusion.

Overall, the test shows the ability of the self-adaptive strategy to handle different discretisation schemes and dynamically adapt the time-step. Oppositely, if the time-

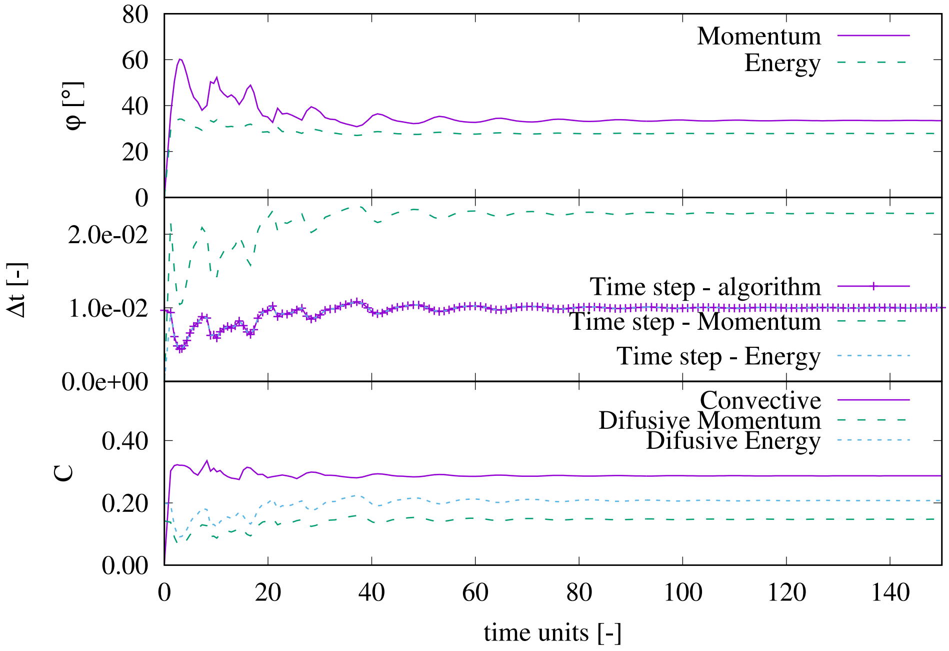


Figure 5.12: Transient evolution of the stability angle φ (top), Δt (middle) and Courant number C (bottom) obtained using an QUICK scheme. See Fig. 5.10 for further details.

step were to be calculated using the CFL condition, the Courant number C employed in Eq. (5.64) should be set to the most restrictive value required by the transient flow evolution. However, with the current approach, the simulation advances in time at the fastest pace possible, which is dictated by the stability of the temporal scheme and the system's eigenvalues at each iteration. It should be noted that, since explicit time-integration schemes are considered, the time step obtained from the temporal scheme stability analysis is more restrictive than the required physical time-step.

5.3.3 Temporal schemes

In the following the self-adaptive strategy is analysed for the temporal integration schemes presented in Section 5.2.6. Results for the 2^{nd} order explicit Linear Multi-step method with a free parameter ($\kappa 2LM$) are not repeated here. The test case is the same as in Section 5.3.2. Now, the convective scheme is fixed, only the SP is used. Still, changing the convective scheme with the different temporal schemes shows the same trends as described in the previous section.

The results obtained for a RK2 scheme and the PC method with a free parameter ($\kappa 2P2C$) developed in Section 5.2.6 are depicted in Figures Fig. 5.13 and Fig. 5.14 respectively. Comparing the evolution of the φ angle it can be observed that the temporal evolution is similar for both cases. As it would be expected, the converged solution does not depend on the chosen temporal scheme. Differences can be observed for the allowable time-step. The RK scheme shows an increased maximum time-step compared to the $\kappa 2LM$ solution, shown in Fig. 5.10. Analogously, Courant number C values computed are also higher for this case. The $\kappa 2P2C$ scheme also allows for a higher time-step than the $\kappa 2LM$ scheme. Still, not as high as for the RK scheme. Nonetheless, the increased time-step for these schemes comes at the cost of more operations per time-step.

Notice that in all the cases presented so far, the time-step of the simulation is limited by the energy equation, with the exception of the simulation using the PC scheme $\kappa 2P2C$ (Fig. 5.14), which is limited by the momentum equation. The reason is that the simulation was carried out employing a variant of the numerical algorithm presented by Ventosa et al. [25]. In the original algorithm the temporal integration of the scalars is performed using a 2^{nd} order Adams-Bashforth scheme in the predictor step and a Crank-Nicholson scheme in the corrector step. In the momentum equation, both sub-steps are solved by means of a 2^{nd} order Adams-Bashforth scheme. In the present work this algorithm is adapted to use the self-adaptive schemes with a free parameter κ. Hence, the energy equation is integrated using the $\kappa 2P2C$ scheme, while the momentum equation employs the $\kappa 2LM$ method in both sub-steps. Since the stability region $\mathbf{R_A}$ of the $\kappa 2P2C$ scheme is bigger than the stability region of the $\kappa 2LM$ scheme for the eigenvalues of the problem, the time-step of the simulation is limited by momentum. Therefore, a new algorithm where both momentum and energy equations are integrated employing the Predictor-Corrector Scheme with a free parameter $\kappa 2P2C$ is implemented and assessed. The results obtained with this new algorithm are depicted in Fig. 5.15. As can be seen, the time-step estimated for the energy equation is the same than the one obtained with the modified algorithm of Ventosa et al. [25]. However, in the new algorithm the time-step calculated for the momentum equation is bigger, allowing a bigger time-step for the simulation. The reason is that the time-step is limited by the energy equation and not by momentum.

Overall, the described time-step maximisation strategy has been shown to effectively bound the eigenvalues of the system and to find the maximum time-step for a stable simulation.

5.3.4 Variable density cases

A variable density DHC case has been simulated to ascertain the capacity of the described strategy to handle cases with variable density. This configuration corresponds with the DHC benchmark case studied by Le Quéré et al. [26]. In these cases

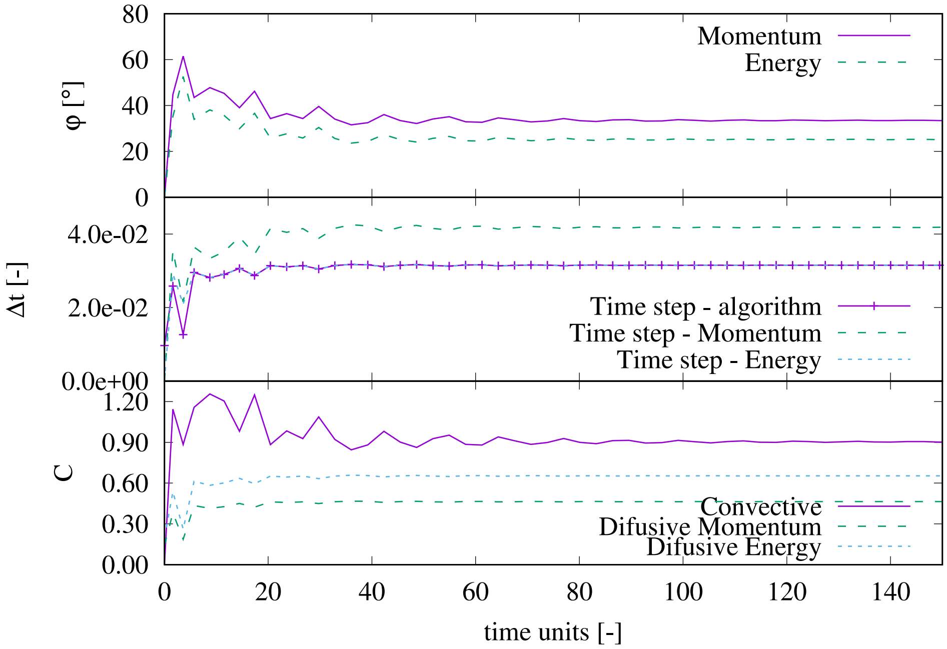


Figure 5.13: Transient evolution of the stability angle φ (top), Δt (middle) and Courant number C (bottom) obtained using a RK2 temporal scheme. See Fig. 5.10 for further details.

even when using a Symmetry-Preserving scheme the diagonal entries of the convective matrix are in general not null. To perform the simulations the same 128x64 CV mesh is used. Also, a Rayleigh number of 10^7 is set. Still, boundary conditions are changed. In the present case the hot wall is at $T_h = 960K$ and the cold wall at $T_c = 240K$. The density is evaluated using the ideal gas law and fluid properties are evaluated using Sutherland's law [27]. For the buoyancy term the Boussinesq approximation is not used. Instead, the product of gravity and density is retained.

A key characteristic of this case is the large temperature difference between vertical walls. The $\epsilon = \frac{T_h - T_c}{2T_0}$ parameter can be regarded as an indicator of the deviation from the Boussinesq case. Hence, even though the Rayleigh number is retained, results of this simulation differ from those performed with a lower temperature difference [28], and consequently with much lower ϵ values.

In Fig. 5.16 are shown the results of the simulation. As can readily be seen, the computed φ differs from previous results, which were computed in the incompress-

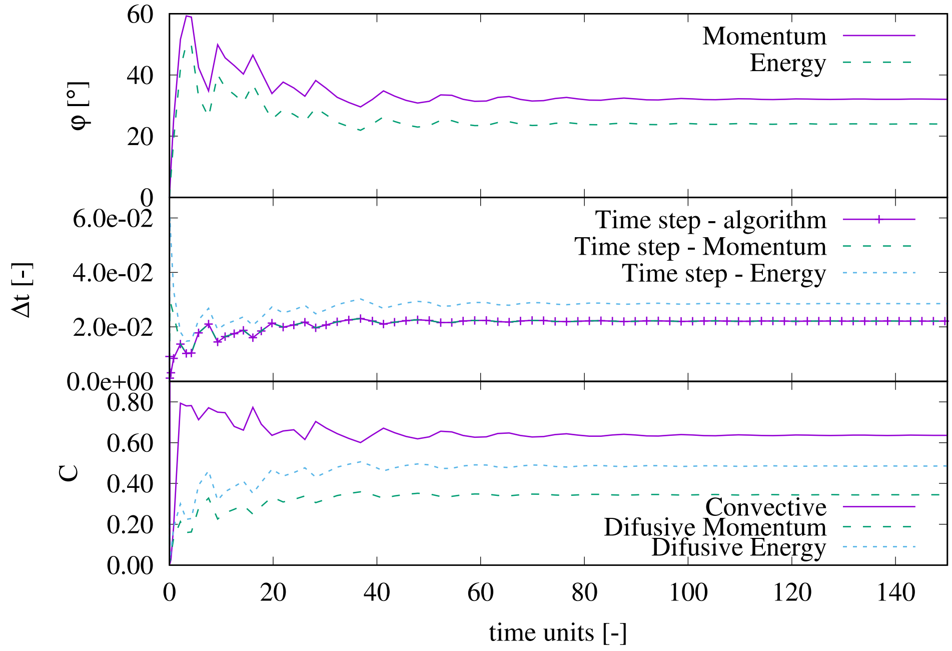


Figure 5.14: Transient evolution of the stability angle φ (top), time-step (middle) and Courant number C (bottom) obtained using a PC temporal scheme. See Fig. 5.10 for further details.

ible limit. Thus, the eigenvalues also change. From these results it can be inferred that diffusive transport is much more prominent than at low ϵ values, namely in the incompressible limit ones.

Regarding the time-step, again differences with respect to the incompressible simulations can be observed. Nonetheless, convergence is achieved after a similar number of time-units. With this test, the self-adaptive strategy is shown to be able to handle variable density cases.

5.4 Conclusions

A self-adaptive strategy for the evaluation of the time-step for the numerical solution of the Navier-Stokes equations has been presented. It has been argued that the increase in the time-step is a way to improve the computational performance of numerical codes devoted to solve the Navier-Stokes equations. The self-adaptive

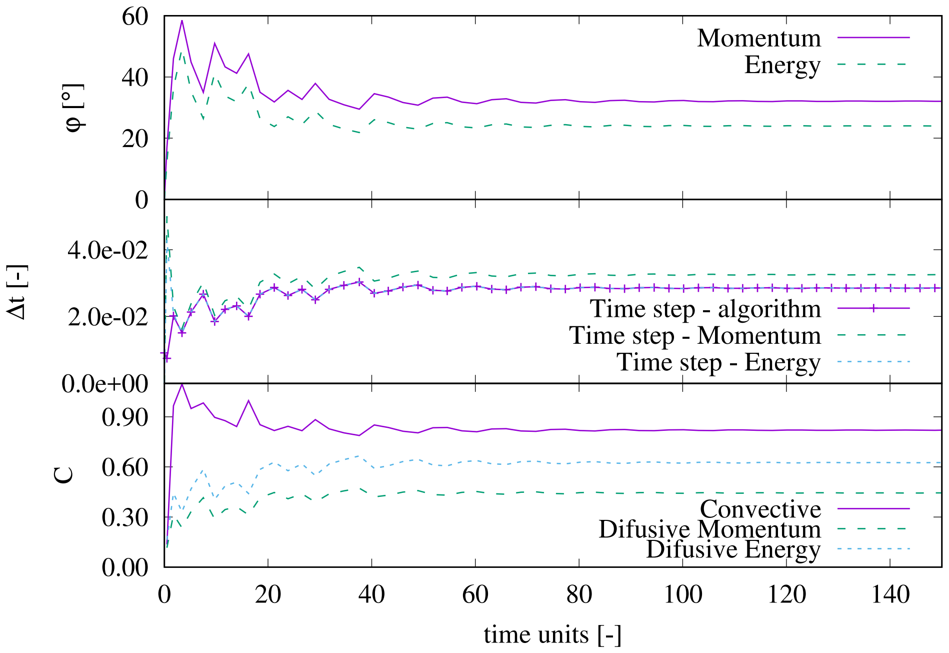


Figure 5.15: Transient evolution of the stability angle φ (top), time-step (middle) and Courant number C (bottom) obtained using a modified PC temporal scheme. See Fig. 5.10 for further details.

strategy was presented in a previous paper [1] where the convective and diffusive matrices were assumed to be of a certain type, skew-symmetric for the convective operator and symmetric and negative-definite for the diffusive operator. Extension has been made to convective operators which result in non-skew-symmetric convective matrices. Furthermore, variable density cases have also been considered, where in general the diagonal of the convective matrix is not null. The approach described in the paper requires the evaluation of the eigenvalues of the system, which is an expensive process. A method to bound these eigenvalues, and consequently avoid their direct evaluation, has been presented showing high efficiency. The bounding is based on the Gershgorin Circle Theorem. Still, two aspects have been numerically assessed. On the one hand, according to the Theorem by Bendixson [17], it has been numerically studied how the eigenvalues of a matrix are bounded by the eigenvalues of their symmetric and skew-symmetric matrices. On the other hand, it has been shown that for non-skew-symmetric convective matrices, the bounding technique can

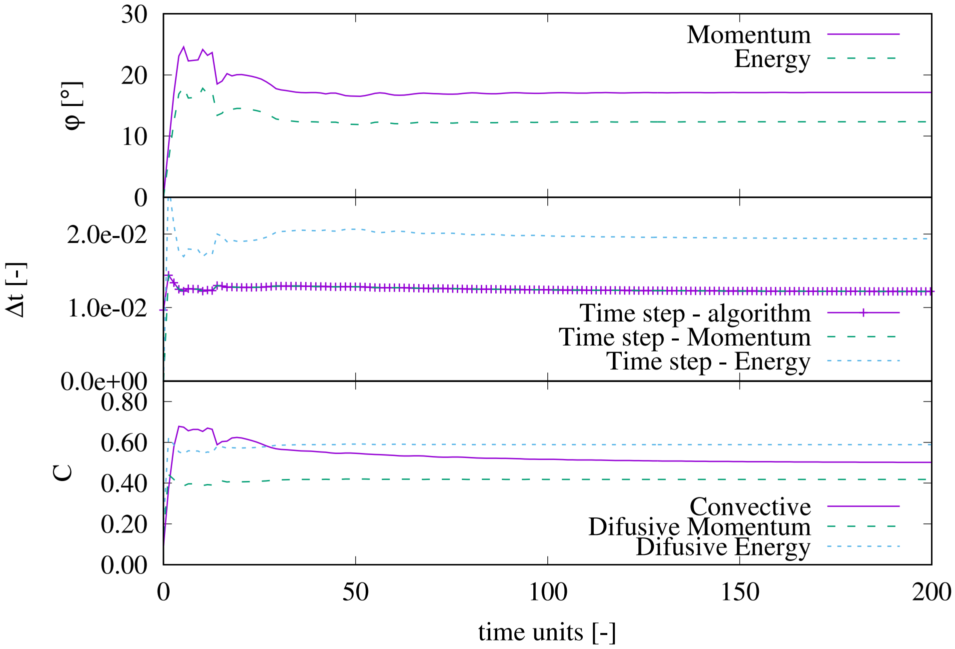


Figure 5.16: Transient evolution of the stability angle φ (top), time-step (middle) and Courant number C (bottom) obtained using a PC temporal scheme for a variable density DHC. See Fig. 5.10 for further details.

also be applied, provided that they are decomposed into their skew-symmetric and symmetric parts. The latter can be added to the diffusive matrix and the eigenvalues of the whole system are still correctly bounded.

Different temporal schemes and discretisation schemes have been tested. Regarding spatial discretisation schemes, the method has been extended to handle upwind-like schemes, such as the UDS and QUICK schemes. Nonetheless, with the presented methodology, extension to other schemes such as flux-limiters is straightforward. Regarding temporal schemes, extension to Predictor-Corrector (PC) and Runge-Kutta (RK) schemes has been performed. In the former, a new PC scheme with a free parameter has been presented. Concerning RK schemes, since no free parameter is present, the self-adaptive strategy only finds the maximum allowable time-step but it does not modify the region of stability. Nonetheless, it is shown that the method is capable of handling this type of schemes. The development of a self-adaptive RK scheme with free-parameter κ will be studied and developed in future works.

The described methodology is capable of finding an optimal time-step for each case. Therefore, the current approach can be implemented in any CFD code, allowing the calculation of the optimal time-step. This enables a stable and accurate simulation with a significant decrease of the computational cost.

References

[1] F.X. Trias and O. Lehmkuhl. A Self-Adaptive Strategy for the Time Integration of Navier–Stokes Equations. *Numerical Heat Transfer, Part B: Fundamentals*, 60(2):1116–134, 2011.

[2] 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.

[3] John Kim, Parviz Moin, and Robert Moser. Turbulence statistics in fully developed channel flow at low Reynolds number. *Journal of Fluid Mechanics*, 177:133–166, 4 1987.

[4] B. Sanderse. Energy-conserving Runge-Kutta methods for the incompressible navier-stokes equations. *Journal of Computational Physics*, 233:100–131, 2013.

[5] B. Sanderse and B. Koren. Accuracy analysis of explicit Runge-Kutta methods applied to the incompressible Navier–Stokes equations. *Journal of Computational Physics*, 231:3041–3063, 2012.

[6] B. Sanderse and B. Koren. New explicit Runge-Kutta methods for the incompressible Navier–Stokes. In *Seventh International Conference on Computational Fluid Dynamics (ICCFD7)*, Big Island, Hawaii, July 9-13, 2012.

[7] R. Knikker. A comparative study of high-order variable-property segregated algorithms for unsteady low Mach number flows. *International Journal for Numerical Methods in Fluids*, 66:403–427, 2011.

[8] Habib N. Najm, Peter S. Wyckoff, and Omar M. Knio. A Semi-Implicit Numerical Scheme for Reacting Flow. *Journal of Computational Physics*, 143:381–402, 1998.

[9] H. Pitsch and H Steiner. Large-eddy simulation of a turbulent piloted methane/air diffusion flame (Sandia flame D). *Physics of Fluids*, 12(10):2541–2554, 2000.

[10] B. Lessani and M.V. Papalexandris. Time-accurate calculation of variable density flows with strong temperature gradients and combustion. *Journal of Computational Physics*, 212:218–246, 2006.

[11] R.W.C.P. Verstappen and A.E.P. Veldman. Direct Numerical Simulation of Turbulence at Lower Costs. *Journal of Engineering Mathematics*, 32:143–159, 1997.

[12] R.W.C.P. Verstappen and A.E.P. Veldman. Symmetry-preserving discretization of turbulent flow. *Journal of Computational Physics*, 187:343–368, 2003.

[13] 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, 2014.

[14] L.F. Shampine. *Numerical solution of ordinary differential equations.* Chapman and Hall/CRC, 1994.

[15] Endre Süli. *Numerical Solution of Ordinary Differential Equations.* Mathematical Institute, University of Oxford, 2014.

[16] Endre Süli and David F Mayers. *An introduction to numerical analysis.* Cambridge university press, 2003.

[17] I. Bendixson. Sur les racines d'une équation fondamentale. *Acta Mathematica*, 25:359–365, 1902.

[18] Brian P Leonard. A stable and accurate convective modelling procedure based on quadratic upstream interpolation. *Computer methods in applied mechanics and engineering*, 19(1):59–98, 1979.

[19] M. S. Darwish and F. H. Moukalled. Normalized Variable and Space Formulation Methodology for High-Resolution Schemes. *Numerical Heat Transfer, Part B: Fundamentals*, 26:79–96, 1994.

[20] J.D. Lambert. *Numerical Methods for Ordinary Differential Systems: The Initial Value Problem.* Wiley, 1991.

[21] Ll. Jofre, O. Lehmkuhl, F.X. Ventosa, J. Trias, and A. Oliva. Conservation properties of unstructured finite-volume mesh schemes for the Navier-Stokes equations. *Numerical Heat Transfer, Part B: Fundamentals*, 65(1):53–79, 2014.

[22] O. Lehmkuhl, C.D. Pérez Segarra, R. Borrell, M. Soria, and A. Oliva. Termofluids: A new Parallel unstructured CFD code for the simulation of turbulent industrial problems on low cost PC cluster. *Proceedings of the Parallel CFD Conference*, pages 1–8, 2007.

[23] Scilab Enterprises. *Scilab: Free and Open Source software for numerical computation.* Scilab Enterprises, Orsay, France, 2012.

[24] S. V. Patankar. *Numerical heat transfer and fluid flow.* Hemisphere Publishing Corporation and McGraw Hill Book Company., Washington – New York – London and New York., 1980.

[25] J. Ventosa-Molina, J. Chiva, O. Lehmkuhl, J. Muela, C. D. Pérez-Segarra, and A. Oliva. Numerical analysis of conservative unstructured discretisations for low mach flows. *International Journal for Numerical Methods in Fluids*, 84(6):309–334, 2017. fld.4350.

[26] Patrick Le Quéré, Catherine Weisman, Henri Paillère, Jan Vierendeels, Erik Dick, Roland Becker, Malte Braack, and James Locke. Modelling of natural convection flows with large temperature differences: a benchmark problem for low Mach number solvers. Part 1. Reference solutions. *ESAIM: Mathematical Modelling and Numerical Analysis*, 39(3):609–616, 2005.

[27] W. Sutherland. The viscosity of gases and molecular force. *Philosophical Magazine*, 5(36):507–531, 1893.

[28] J. Vierendeels, B. Merci, and E. Dick. Benchmark solutions for the natural convective heat transfer problem in a square cavity with large horizontal temperature differences. *International Journal of Numerical Methods for Heat and Fluid Flow*, 13(8):1057–1078, 2003.

6

Conclusions

This thesis has been focused on the development and implementation of models and algorithms for numerical simulations of heat and mass transfer phenomena. The new codes and features have been specifically implemented in the in-house parallel Computational Fluid Dynamics (CFD) code **TermoFluids** (TF) [1]. This CFD code is a multi-physics code developed to work in unstructured meshes with a high parallel scalability. TermoFluids has been developed and maintained during the last years by the *Centre Tecnològic de Transferència de Calor* (CTTC) research group at the *Universitat Politècnica de Catalunya - BarcelonaTech* (UPC). The CTTC group has come a long way in the research of heat and mass transfer phenomena, a journey that began in the early 80s and continues today. Therefore, the work presented in this thesis can be seen as a *bridge* towards new physics and applications in the long road that the CTTC group has travelled in the scientific research of heat and mass transfer phenomena. As a *bridge*, the work carried out during the thesis aimed to overcome some obstacles and difficulties found in the *road* during this research task. Once the work has been finished, it can be stated that this new *bridge* built permits to continue the *journey*, allowing to set new goals and face new challenges in the research field of heat and mass transfer phenomena.

Each one of the previous chapters of this thesis has been focused on one problem to solve, presenting one or various possible solutions to face it. Although all the previous chapters of the thesis have their own conclusions section, where the main results and conclusions derived from the work done have been presented in deep, this chapter summarizes the main research goals and findings of the whole thesis.

First, Chapter 2 deals with numerical simulations of combustion processes. As it is argued throughout the chapter, the numerical simulations of combustion processes in turbulent flows present two main difficulties. The first one is related with the stiffness of the equations required to calculate the rates of production and destruction of all the chemical species when using the empirical Arrhenius expression [2, 3]. This stiffness forces to employ very small time integration steps when using explicit time integration schemes, or to employ non-classical implicit integration methods, that allow higher time integration steps, but at the expense of a greater computational load. The second difficulty arises when a modelling approach, i.e. RANS or LES, is employed to simulate turbulent combustion processes. For example, in LES only the large scales of the flow are solved, while the sub-grid scales are modelled, and as the chemical reactions occurs at molecular level, the flow-chemistry interaction must be modelled due to it occurs at sub-grid level, and a closure model has to be employed for the chemical reaction term. The first issue has been tackled from two different angles: one using a combustion model that allow to reduce the computational effort of computing chemical reactions during simulation time, and a second one developing a smart balancing algorithm for the evaluation of the chemical reaction rates in parallel combustion simulations.

The combustion model presented in this thesis is the Progress-Variable (PV) model. The method is based on the following: previous to the numerical simulation, all the detailed chemistry is computed and tabulated as a function of a reduced system of tracking scalars. During simulation time, instead of solving a transport equation for each species of the mixture, and computing the chemical reaction rates using finite-rate chemistry, only the tracking scalars are transported. The information regarding the state of the chemical reaction is recovered accessing to the pre-generated look-up table. Moreover, in order to overcome the modelling issue of the chemistry-flow interaction at sub-grid level in LES modelling, the model Presumed Conditional Moment (PCM) has been applied to the PV model in order to close the chemical reaction term in turbulent combustion simulations. This PCM model is a *simplified* version of the Conditional Moment Closure (CMC) model [4–6], as in PCM the conditional moments are presumed instead of being calculated. This modelling approach combining the PV and PCM models has been employed to simulate a benchmark case, namely the Cambridge autoignition experiment [7]. The results obtained from the numerical simulations show that the model is able to reproduce the combustion processes taking place in the experiment, including the auto-ignition and quenching of the ignition kernels. Nonetheless, the numerical simulations predicted a slightly smaller auto-ignition length than the experiment, due to this the model over-predicts the reaction rate of the progress variable. Hence, although the model produces reasonable accurate results and allows to simulate turbulent combustion processes at a very reasonable cost, it should be reviewed aiming to enhance its accuracy.

The second strategy applied to make combustion simulations more affordable is the development and implementation of a smart balancing algorithm for parallel combustion simulations. The method is based on a non-classical integration method specially well-suited for stiff sets of equations, specifically Gear's method [8]. This method allows higher time integration steps than conventional integration schemes, although at the expense of a higher computational cost. Nonetheless, this method is only applied to the computational cells with active chemical reactions, while non-reactive cells are integrated explicitly, as their equations do not present a stiff behaviour. Hence, only the cells close to the flame front are integrated using Gear's method, which means that in parallel simulations only a part of the total number of processors will have to perform this additional calculations, generating an imbalance in the simulations. The dynamic balancing algorithm avoids this imbalance in the computational load, and allows to notably speed-up the simulations, resulting in a notorious reduction of the simulation time. A rigorous analysis of the computational performance and the scalability of the developed algorithm has been presented. It is shown that the method scales very well, and is able to re-balance strongly imbalanced simulations, helping to increase the performance of parallel simulations. Moreover, this solution has been employed to simulate the same benchmark combustion presented above, the Cambridge experiment. Although in this case a perfect mixing at sub-grid scale is assumed, meaning that the sub-grid chemistry-flow interaction has not been modelled, the obtained results agree fairly well with the experimental ones. Moreover, and what is more relevant, it is demonstrated how the algorithm allows to notably speed-up the computations, making the combustion simulations using finite-rate chemistry more affordable.

In Chapter 3, another type of multi-physics problem has been treated: dispersed multi-phase flows. These types of flows are characterized by the presence of two (or more) phases, where one is continuous and the others are disperse, in the form of unconnected particles or droplets. There are several modelling approaches aimed to simulate this type of flows. Each one of these approximations has its owns strengths and limitations, being better suited and sometimes limited to a concrete type of dispersed multi-phase flows [9]. For example, if the motivation is to study in detail the fluid-dynamic behaviour of a few droplets or bubbles, the best choice is the fully resolved methods, where the forces acting over the dispersed phase are fully resolved. Some of these methods are the Level-set method, the Volume-of-Fluid method, the Front-tracking method, etc. Although they are very accurate, these type of methods are limited to a small number of particles or droplets.

On the other hand, if the aim is to study large systems with a high particle loading like fluidized-beds, the more convenient approach is the Euler-Euler model, where one assumes that the dispersed phase can be treated as a continuum. The objective of the current thesis is to study large systems where millions of particles or droplets can be present ranging from very dilute two-phase flows up to dense dispersed two-phase flows, i.e. the injection of coal in coal furnaces, fuel spray jets in combustion chambers, inhaled medicines, evaporative cooling applications or cyclone separators. For these type of flows the best suited option is the Lagrangian-Eulerian model. Hence, this model has been the one presented in this thesis and implemented in the CFD code TermoFluids.

Throughout the thesis some key aspects relevant for the implementation of the Lagrangian-Eulerian model have been deeply analysed. A rigorous analysis of different interpolation methods for unstructured meshes has been presented. The conclusions derived from this analysis shows which method is the most interesting depending on the simulation set-up, taking into account the accuracy offered by each interpolation method as well as its associated computational cost, which can be a limiting factor.

Another aspect studied in Chapter 3 is the relevance of the contribution of the sub-grid scales over the dispersed phase in LES modelling. As in LES the sub-grid scales are modelled instead of resolved, it is relevant to assess which is the influence of this unresolved scales over the particles or the droplets. In order to do so, several *a priori* LES studies have been carried out, where the DNS velocity field has been spatially filtered (FDNS simulations), obtaining LES-like velocity fields. The analysis has been done for different filtering levels and several Stokes numbers. Moreover, two sub-grid stochastic models available in the literature aimed to reconstruct the influence of the sub-grid scales over the particles have been assessed, namely the Bini and Jones (BJ) model [10, 11] model, and the Pozorski and Apte (PA) model [12]. The results of the analysis show that LES filtering have an influence over the particles depending on their Stokes number. For a Stokes number smaller than unity, LES filtering dissipates the preferential concentration effect. For particles with Stokes number larger than one, the preferential concentration effect seems to be enhanced by the filtering. Moreover, in both cases the total kinetic energy of the particles is diminished with filtering. Regarding the studied sub-grid stochastic models, both models are able to recover part of the kinetic energy lost with filtering, but they tend to dissipate the preferential concentration effect for all types of particles. This dissipation effect is desired for the heavy particles, but make results for particles with a small Stokes number worse.

From the detected deficiencies in the sub-grid stochastic models, the development of a new sub-grid dispersion model is investigated. The new proposed model is based on the reconstruction of statistics via Probability Density Functions (PDFs). The idea is to recover the influence of the sub-grid scales not resolved in LES over the particles from magnitudes ready-available in LES simulations. The first results obtained with this new model are very promising, and clearly outperform the results of the previous presented stochastic models. For the studied case, the new model is able to almost match the kinetic energy values of the DNS simulation. Furthermore, for particles with a Stokes number of 0.5, the method is able to recover part of the particle concentration effect lost with filtering, oppositely to the stochastic models which have a scattering effect over this class of particles.

Closing the chapter, two realistic cases have been simulated using the models and numerical libraries for dispersed multi-phase flows implemented during the development of this thesis. The first case is the flow loop Hercule of Borée et al. [13], which reproduces an industrial configuration aimed to control the mixing of a fuel, e.g. pulverized coal, with the surrounding air flow. The second simulated case is an *in vitro* experiment carried out in a human-based model of the upper airways [14], where the regional deposition of inhaled medicines in a realistic human airways configuration has been studied. The numerical results obtained from both cases are in good agreement with those of the experiments, demonstrating that the developed numerical tools are able to produce reliable results, and validating the numerical code.

The problem faced in Chapter 4 is the one found in CFD simulations where mobile parts moving close to other solid parts that are fixed exist. For example, when simulating applications like wind turbines or turbo-machines, it would be interesting to use moving meshes attached to the rotating parts, at the same time as static meshes are employed for the fixed parts. Therefore, it is desired to develop a method able to carry out simulations whose simulation domain is composed by one or more static meshes, and one ore several moving meshes. With that aim, an algorithm able to run parallel simulations using non-overlapping meshes which share some or all boundaries between them has been developed and implemented. The developed method falls inside the category of sliding meshes techniques.

The implemented algorithm employs auxiliary particles that help to reconstruct at each iteration the matrix operators of the discrete Navier-Stokes equations, allowing to *stitch* the independent meshes, and solve the numerical system as a whole. In order to analyse the properties of the algorithm, two studies have been carried out: the first one analysing the conservation properties of the method, and a second one studying its parallel scalability. The conservation analysis studied the influence of different parameters over the conservation properties of the method. The results obtained show that although there is a decrease in the conservation of mass and kinetic energy regarding a case using an unique static mesh, the obtained results are fairly good and validate the method. Regarding the parallel scalability analysis, it shows that the algorithm is able to scale up to thousands of CPUs with a good performance, meaning that the method can be used in big simulations of complex cases. The method has been applied to real test cases. The obtained numerical results are accurate and match the results of the experiments, showing the capability of the algorithm to generate reliable numerical results.

Furthermore, aiming to improve the conservation properties of the previous methodology, a new algorithm is developed. The main difference between the new method and the previous one is that the previous method supposes the geometry of the neighbour cell in the neighbour mesh, while the new one performs a surface' projection step where all the new faces connecting the boundary cells at the sliding boundary are calculated at each interaction. Obviously, this additional calculation makes the new method more computationally expensive, although it allows the method to preserve the mass and the kinetic energy better. A conservation analysis of the new method shows that this is capable of obtaining mass conservation values in the same order as a static case solved in a full mesh. In the same vein, the kinetic energy is also almost perfectly preserved, and only a minor dissipation occurs for the highest relative velocity considered, being almost negligible. Hence, it is demonstrated that the new method is able to perform simulations with sliding meshes preserving mass and kinetic energy just as well as a simulation in a full static mesh.

Finally, in Chapter 5 a new generic methodology for the time-integration of convection-diffusion equations aimed to increase as much as possible the integration time-step of explicit time-integration schemes is presented. This method allows to reduce the computational cost of numerical simulations. The presented work is an extension of a previous work carried out by Trias and Lehmkuhl [15], which is based on an earlier research carried out by Verstappen and Veldman [16]. In their work, Trias and Lehmkuhl presented a method able to modify dynamically the stability region of the temporal integration scheme depending on the properties of the system matrix. This allows the use of larger time-steps than when a usual time integration scheme is employed. The modification of the stability region is done thanks to the addition of a free-parameter in the temporal integration scheme. The method was implemented for a 2^{nd} order Linear Multi-step integration scheme. Moreover, the method was restricted to cases where the convective matrix operator was skew-symmetric, i.e., when using the symmetry-preserving discretization scheme [17].

In this thesis, the methodology has been extended to work in other time-integration schemes like Predictor-Corrector schemes and Runge-Kutta schemes, as well as to work with Upwind-like convective discretization schemes, and not only with the symmetry-preserving scheme. Moreover, the method has also been extended to work in variable density cases. In these cases, the symmetry-preserving scheme may not generate a skew-symmetric matrix operator, as the diagonal term is not null in general. Furthermore, it has been numerically assessed how the strategy to bound the matrix eigenvalues, based on the Gershgorin Circle Theorem and Bendixson's Theoren [18], is able to perfectly bound the exact matrix eigenvalues. It has been also demonstrated that the volume matrix does not affect the eigenvalues of the system matrix.

In order to ascertain the capabilities of the method, the chapter ends with various numerical simulations validating the different convective schemes and numerical integration methods presented throughout the chapter, as well as a variable density case. In order to validate the computations, the well-known Differentially Heated Cavity case has been used. The obtained results demonstrate the capability of the method to correctly determine a stable and optimal time-integration step for all the studied numerical schemes, and time integration methods. Moreover, it is proven that the method is able to dynamically select at each time-step the maximum allowable time-step that ensures the simulation stability. Therefore, the method demonstrates two clear advantages respect the classical CFL condition used to determine the time-step of the simulations: it dynamically adapts and maximizes the stability region of the time-integration schemes according to the system matrix eigenvalues, and it is capable to determine automatically the maximum stable time-integration step.

Future Work

Following the analogy presented at the beginning of the chapter, this thesis is a bridge that allows to continue the current research in the field of numerical simulations of heat and mass transfer phenomena at the *Centre Tecnològic de Transferència de Calor* (CTTC) beyond the current know-how. Moreover, the work developed during the thesis allows to face research challenges in new fields and topics. Nonetheless, these new possibilities also present new challenges and difficulties that encourage to continue and extend the work developed in this thesis. In this section, possible future works and research which can be carried out in order to widen the presented work are detailed.

There are several tasks which would be interesting to carry out in order to continue the research presented in Chapter 2. One interesting task would be the implementation of some sub-grid chemistry flow model for finite-rate chemistry. The developed smart-balancing algorithm has demonstrated to be a powerful tool to simulate combustion cases in parallel computation. Nevertheless, to properly simulate turbulent cases the sub-grid chemistry-flow interaction must be taken into account. Therefore, some modelling approaches available in the literature, i.e. Conditional Moment Closure (CMC) [6], a sub-grid joint probability density function (PDF) approach [19, 20], the dynamic thickened flame model [21] or the Partially Stirred Reactor approach [22] should be implemented in TermoFluids. This new feature will greatly increase the capability of the code to simulate combustion processes.

Other possible works are related with the implemented Progress-Variable model coupled with the Presumed Conditional Moment closure. The obtained results show that this approach is able to capture the physics in the auto-ignition process of a hydrogen flame, although some discrepancies regarding the experimental results are found. Hence, it would be interesting to carry out further research work trying to figure out the source of the discrepancies. Combustion simulations are very complex, and they involve different modelling assumptions and approaches, and sometimes it is difficult to figure out which assumption or simplification is affecting the results the most. Therefore, a rigorous quantification of these effects can be a very interesting research work, although at the same time very tough and complex.

The future work to be carried out starting from the work already developed in Chapter 3 can be very extensive. There are plenty of new physics and phenomenologies which can be added to the numerical models now implemented. For example, the current implementation considers the particle displacement, but not its possible rotation. When rotation is included due to torques acting on the particles, this particle may experience a lift force due to its rotation, known as the Magnus force. This force can be relevant at some flow configurations. Another lift force that is not yet implemented is the Saffman lift force, experienced by particles moving in a shear layer due to the non-uniform relative velocity over the particle and the resulting non-uniform pressure distribution. This transversal lift force pushes the particle towards the direction of the higher slip velocity in the shear-layer [23]. On the other hand, the motion of bubbles is much more complex than that of a rigid solid sphere. In bubbles, the interface is not rigid because there is a flow moving inside the bubbles, implying that the interface has a relative motion, which causes a drag reduction. Therefore, more sophisticated and complex drag correlations should be implemented. Moreover, in bubbles the transversal lift forces, the added mass force or the wall effects are really relevant. Therefore, bubbly flows are a big challenge [9].

More physics which can be included and can be relevant in some flow configurations are the particle-particle collisions and the particle-walls collisions. Both phenomena are sub-grid phenomenology which must be implemented using models. Several models can be found in the literature for both phenomena based on different approaches, and with different degrees of complexity, accuracy and computational requirements. For example, among the particle-particle collision models there are stochastic models [24], probabilistic models [25] or deterministic ones [26]. Regarding the particle-wall collision models, these can range from simple elastic collision models to more sophisticated approaches considering the roughness of the wall [27]

Other interesting physics that can be implemented for droplets in a future work are nucleation and coalescence. The implementation of these physics will lead the opportunity to study cases like steam near saturation conditions undergoing a rapid expansion in nozzles [28], or the wetness losses occurring at the low pressure part of a steam turbine due to condensation [29].

Furthermore, a clear objective is a transversal work coupling the multi-physics developed in Chapters 2 and 3. This link between these numerical models will allow the possibility to carry out complex CFD simulations of all the physics present in combustion chambers, from the injection of the fuel, its atomization and break-up, followed by the heating and evaporation of the fuel droplets, and once the fuel is evaporated, simulating the chemical reactions of the combustion process.

For the work carried out in Chapter 4, the future work is crystal clear: finish the development of the new algorithm, which enhances the conservation properties of the previous one. The current bottle-neck of the algorithm is in the projection and intersection step. Some heuristics and possible geometric intersections between faces are still not implemented, and sometimes the algorithm is not able to resolve the new face intersections. In order to solve this issue, two possible complementary strategies can be followed: the first and most intuitive is to try to find all casuistry of the possible geometries and implement a solution for their intersection. The second one can be thought of as a backup or safety strategy: if at some iteration a geometric configuration which is not covered by the intersection algorithm is found, it is possible to perform a smaller time-step, changing the relative position between the moving and the static meshes, thus avoiding the situation which generated the issue in the intersection process.

Regarding the work in Chapter 5, it can be extended to further time-integration schemes, like high order Linear Multi-step and Predictor-Corrector methods. Another future work is to study the possibility of developing a Runge-Kutta-like scheme including a free-parameter as done for the other integration schemes, adding the capability to this new RK-like method to adapt its stability region according to the matrix system eigenvalues. Moreover, from the results obtained in the detailed numerical analysis about how the current strategy bound the eigenvalues of the system, which shows that the eigenvalues are always safely bounded, it would be interesting to study if a new strategy able to bound the eigenvalues of the system more precisely can be derived. This will allow the use of higher time-steps in the simulations. Nonetheless, this new strategy should be computationally as cheap as the current one. If not, the savings in the simulation time obtained with this method can be lost if the strategy to estimate the eigenvalues requires a remarkable computational effort.

References

[1] O. Lehmkuhl, C.D. Perez-Segarra, R. Borrell, M. Soria, and A. Oliva. TERMOFLUIDS: A new Parallel unstructured CFD code for the simulation of turbulent industrial problems on low cost PC Cluster. *Parallel Computational Fluid Dynamics 2007*, pages 275–282, 2009.

[2] E. Hairer, S.P. Nørsett, and G. Wanner. *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems.* Lecture Notes in Economic and Mathematical Systems. Springer, 1993.

[3] Thierry Poinsot and Denis Veynante. *Theoretical and numerical combustion.* RT Edwards, Inc., 2005.

[4] A.Y. Klimenko. Multicomponent diffusion of various admixtures in turbulent flow. *Fluid dynamics*, 25(3):327–334, 1990.

[5] R.W. Bilger. Conditional moment closure for turbulent reacting flow. *Physics of Fluids A: Fluid Dynamics*, 5(2):436–444, 1993.

[6] Alex Y Klimenko and Robert William Bilger. Conditional moment closure for turbulent combustion. *Progress in energy and combustion science*, 25(6):595–687, 1999.

[7] C.N. Markides and E. Mastorakos. An experimental study of hydrogen autoignition in a turbulent co-flow of heated air. *Proceedings of the Combustion Institute*, 30(1):883–891, 2005.

[8] C. William Gear. *Numerical Initial Value Problems in Ordinary Differential Equations*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1971.

[9] Swedish Industrial Association for Multiphase Flows (SIAMUF) and Martin Sommerfeld. *Best Practice Guidelines for Computational Fluid Dynamics of Dispersed Multi-Phase Flows.* European Research Community on Flow, Turbulence and Combustion (ERCOFTAC), 2008.

[10] M. Bini and W.P. Jones. Particle acceleration in turbulent flows: A class of nonlinear stochastic models for intermittency. *Physics of Fluids*, 19:035104, 2007.

[11] M. Bini and W.P. Jones. Large-eddy simulation of particle-laden turbulent flows. *Journal of Fluid Mechanics*, 614:207–252, 2008.

[12] J. Pozorski and S.V. Apte. Filtered particle tracking in isotropic turbulence and stochastic modeling. *International Journal of Multiphase Flow*, 35:118–128, 2009.

[13] J Borée, T Ishima, and I Flour. The effect of mass loading and inter-particle collisions on the development of the polydispersed two-phase flow downstream of a confined bluff body. *Journal of Fluid Mechanics*, 443:129–165, 2001.

[14] Frantisek Lizal, Jakub Elcner, Philip K Hopke, Jan Jedelsky, and Miroslav Jicha. Development of a realistic human airway model. *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine*, 226(3):197–207, 2012.

[15] F.X. Trias and O. Lehmkuhl. A Self-Adaptive Strategy for the Time Integration of Navier–Stokes Equations. *Numerical Heat Transfer, Part B: Fundamentals*, 60(2):1116–134, 2011.

[16] R.W.C.P. Verstappen and A.E.P. Veldman. Direct Numerical Simulation of Turbulence at Lower Costs. *Journal of Engineering Mathematics*, 32:143–159, 1997.

[17] R.W.C.P. Verstappen and A.E.P. Veldman. Symmetry-preserving discretization of turbulent flow. *Journal of Computational Physics*, 187:343–368, 2003.

[18] I. Bendixson. Sur les racines d'une équation fondamentale. *Acta Mathematica*, 25:359–365, 1902.

[19] W.P. Jones, S. Navarro-Martinez, and O. Röhl. Large eddy simulation of hydrogen auto-ignition with a probability density function method. *Proceedings of the Combustion Institute*, 31(2):1765–1771, 2007.

[20] W.P. Jones and S. Navarro-Martinez. Study of hydrogen auto-ignition in a turbulent air co-flow using a Large Eddy Simulation approach. *Computers and Fluids*, 37(7):802–808, 2008.

[21] O. Colin, F. Ducros, D. Veynante, and T. Poinsot. A thickened flame model for large eddy simulations of turbulent premixed combustion. *Physics of Fluids*, 12(7):1843–1863, 2000.

[22] V. Sabelnikov and C. Fureby. Extended LES-PaSR model for simulation of turbulent combustion. In *Progress in Propulsion Physics*, volume 4, pages 539–568. EDP Sciences, 2013.

[23] Clayton T. Crowe, John D. Schwarzkopf, Martin Sommerfeld, and Yutaka Tsuji. *Multiphase flows with droplets and particles*. CRC press, 2011.

[24] Martin Sommerfeld. Validation of a stochastic Lagrangian modelling approach for inter-particle collisions in homogeneous isotropic turbulence. *International Journal of Multiphase Flow*, 27(10):1829–1858, 2001.

[25] David P. Schmidt and C.J. Rutland. A new droplet collision algorithm. *Journal of Computational Physics*, 164(1):62–80, 2000.

[26] Michael Breuer and Michael Alletto. Efficient simulation of particle-laden turbulent flows with high mass loadings using les. *International Journal of Heat and Fluid Flow*, 35:2–12, 2012.

[27] M Breuer, M Alletto, and F Langfeldt. Sandgrain roughness model for rough walls within eulerian–lagrangian predictions of turbulent flows. *International Journal of Multiphase Flow*, 43:157–175, 2012.

[28] M.J. Kermani and A.G. Gerber. A general formula for the evaluation of thermodynamic and aerodynamic losses in nucleating steam flow. *International journal of heat and mass transfer*, 46(17):3265–3278, 2003.

[29] J. Starzmann, M.V. Casey, and J.F. Mayer. Unsteady numerical study of wet steam flow in a low pressure steam turbine. *High Performance Computing in Science and Engineering '11*, pages 437–450, 2012.

Appendix A

Convective operator splitting

In order to bound the eigenvalues using the Gershgorin Circle Theorem presented in Section 5.2.3 is necessary to figure out the form of the convective matrix. Moreover, when Upwind-like schemes are employed, it is necessary to construct *on-the-fly* the symmetric and the skew-symmetric matrices. Following, the splitting of the convective operator **C** in its symmetric and skew-symmetric parts for Upwind-like schemes is detailed.

A.1 1^{st} order Upwind scheme

For the 1^{st} order Upwind scheme only two nodes per face will be linked in the convective matrix. Therefore, the relations between nodes i and j are studied. Hence, for node i:

$$\dot{m}_{ij}\phi_{ij} = \begin{cases} \dot{m}_{ij}\phi_i, & \text{if } \dot{m}_{ij} \geq 0 \\ \dot{m}_{ij}\phi_j, & \text{if } \dot{m}_{ij} < 0 \end{cases} \tag{A.1}$$

From Eq. (A.1) the matrix positions C_{ii} and C_{ij} are derived:

$$C_{ii} = \sum_{f \in F(i)} \dot{m}_{ij} \quad (\text{where } \dot{m}_{ij} = 0 \text{ if } \dot{m}_{ij} < 0) \tag{A.2}$$

$$C_{ij} = \begin{cases} 0, & \text{if } \dot{m}_{ij} \geq 0 \\ \dot{m}_{ij}, & \text{if } \dot{m}_{ij} < 0 \end{cases} \tag{A.3}$$

Regarding the node j, knowing that $\dot{m}_{ij} = -\dot{m}_{ji}$:

$$\dot{m}_{ji}\phi_{ji} = \begin{cases} \dot{m}_{ji}\phi_j, & \text{if } \dot{m}_{ji} \geq 0 \\ \dot{m}_{ji}\phi_i, & \text{if } \dot{m}_{ji} < 0 \end{cases} \begin{cases} -\dot{m}_{ij}\phi_i, & \text{if } \dot{m}_{ij} \geq 0 \\ -\dot{m}_{ij}\phi_j, & \text{if } \dot{m}_{ij} < 0 \end{cases} \tag{A.4}$$

So, the matrix position C_{ji} is:

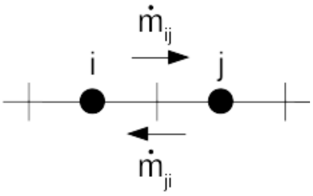


Figure A.1: Stencil scheme for 1^{st} order UW scheme.

$$C_{ji} = \begin{cases} -\dot{m}_{ij}, & \text{if } \dot{m}_{ij} \geq 0 \\ 0, & \text{if } \dot{m}_{ij} < 0 \end{cases} \tag{A.5}$$

From C_{ii}, C_{ij} and C_{ji} the symmetric and skew-symmetric part of the convective matrix are obtained:

$$S(C_{ii}) = C_{ii} \tag{A.6}$$

$$A(C_{ii}) = 0 \tag{A.7}$$

$$|S(C_{ij})| = \frac{C_{ij} + C_{ij}^T}{2} = \begin{cases} -0.5\dot{m}_{ij}, & \text{if } \dot{m}_{ij} \geq 0 \\ 0.5\dot{m}_{ij}, & \text{if } \dot{m}_{ij} < 0 \end{cases} = 0.5\,|\dot{m}_{ij}| \tag{A.8}$$

$$|A(C_{ij})| = \frac{C_{ij} - C_{ij}^T}{2} = \begin{cases} 0.5\dot{m}_{ij}, & \text{if } \dot{m}_{ij} \geq 0 \\ 0.5\dot{m}_{ij}, & \text{if } \dot{m}_{ij} < 0 \end{cases} = 0.5\,|\dot{m}_{ij}| \tag{A.9}$$

As can be seen, the skew-symmetric part of the upwind scheme is almost identical to the skew-symmetric matrix obtained for the Symmetry-Preserving scheme. Note that for the off-diagonal places the absolute value $|\cdot|$ is employed, since the Gershgorin Circle Theorem adds the absolute value of the off-diagonal values in the *deleted absolute row sum* (see Eq. (5.21b)).

A.2 QUICK scheme

The QUICK scheme in its non-dimensional form [1, 2] reads as:

$$\overline{\phi}_f = \overline{x}_f + \frac{\overline{x}_f(\overline{x}_f - 1)}{\overline{x}_C(\overline{x}_C - 1)}(\overline{\phi}_C - \overline{x}_C) \tag{A.10}$$

where:

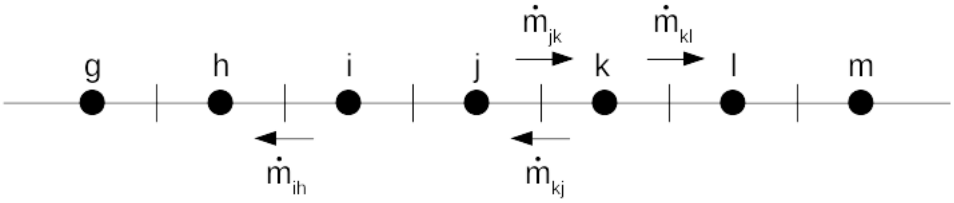


Figure A.2: Stencil scheme for QUICK scheme.

$$\overline{\phi} = \frac{\phi - \phi_U}{\phi_D - \phi_U} \quad \overline{x} = \frac{x - x_U}{x_D - x_U} \tag{A.11}$$

Since to figure out the topology of the convective matrix when using the QUICK scheme the relations between the nodes is required, the dimensional form is recovered combining Eqs. (A.10) and (A.11). Hence:

$$\begin{aligned}\phi_f = (\overline{x}_f - \frac{\overline{x}_f(\overline{x}_f - 1)}{\overline{x}_C(\overline{x}_C - 1)}\overline{x}_C)\phi_D + \frac{\overline{x}_f(\overline{x}_f - 1)}{\overline{x}_C(\overline{x}_C - 1)}\phi_C \\ + (1 - \overline{x}_f - \frac{\overline{x}_f(\overline{x}_f - 1)}{\overline{x}_C(\overline{x}_C - 1)}(1 + \overline{x}_C))\phi_U\end{aligned} \tag{A.12}$$

Expressing Eq. (A.12) in compact form:

$$\phi_f = w_D\phi_D + w_C\phi_C + w_U\phi_U \tag{A.13}$$

In the QUICK scheme, any face **f** of a cell **c** *links* four nodes: two *contact* nodes, and two *far* nodes. Following the scheme depicted in Fig. A.2, the face *jk* of cell **j** is analysed, therefore, the *contact* nodes are *j* and *k*, and the two *far* nodes are *i* and *l*. In this analysis two assumptions must be done:

- The relation between the nodes is bijective.
- The mass flux at the faces is the same: $\dot{m}_{jk} \approx \dot{m}_{kl} \approx \dot{m}_{hi} = -\dot{m}_{kj}$

Hence, for node **j**:

$$\dot{m}_{jk}\phi_{jk} = \begin{cases} \dot{m}_{jk}(w_D\phi_k + w_C\phi_j + w_U\phi_i), & \text{if } \dot{m}_{jk} \geq 0 \\ \dot{m}_{jk}(w_D\phi_j + w_C\phi_k + w_U\phi_l), & \text{if } \dot{m}_{jk} < 0 \end{cases} \tag{A.14}$$

From Eq. (A.14) the positions of the convective matrix C_{jj}, C_{jk}, C_{ji} and C_{jl} are obtained:

$$C_{jj} = \begin{cases} \dot{m}_{jk} w_C, & \text{if } \dot{m}_{jk} \geq 0 \\ \dot{m}_{jk} w_D, & \text{if } \dot{m}_{jk} < 0 \end{cases} \tag{A.15}$$

$$C_{jk} = \begin{cases} \dot{m}_{jk} w_D, & \text{if } \dot{m}_{jk} \geq 0 \\ \dot{m}_{jk} w_C, & \text{if } \dot{m}_{jk} < 0 \end{cases} \tag{A.16}$$

$$C_{ji} = \begin{cases} \dot{m}_{jk} w_U, & \text{if } \dot{m}_{jk} \geq 0 \\ 0, & \text{if } \dot{m}_{jk} < 0 \end{cases} \tag{A.17}$$

$$C_{jl} = \begin{cases} 0, & \text{if } \dot{m}_{jk} \geq 0 \\ \dot{m}_{jk} w_U, & \text{if } \dot{m}_{jk} < 0 \end{cases} \tag{A.18}$$

Analysing the other contact node **k** in face *jk* to obtain the matrix position C_{kj}:

$$\dot{m}_{kj}\phi_{kj} = \begin{cases} \dot{m}_{kj}(w_D\phi_j + w_C\phi_k + w_U\phi_l), & \text{if } \dot{m}_{kj} \geq 0 \\ \dot{m}_{kj}(w_D\phi_k + w_C\phi_j + w_U\phi_i), & \text{if } \dot{m}_{kj} < 0 \end{cases} = \begin{cases} -\dot{m}_{jk}(w_D\phi_k + w_C\phi_j + w_U\phi_i), & \text{if } \dot{m}_{jk} \geq 0 \\ -\dot{m}_{jk}(w_D\phi_j + w_C\phi_k + w_U\phi_l), & \text{if } \dot{m}_{jk} < 0 \end{cases} \tag{A.19}$$

$$C_{kj} = \begin{cases} -\dot{m}_{jk} w_C, & \text{if } \dot{m}_{jk} \geq 0 \\ -\dot{m}_{jk} w_D, & \text{if } \dot{m}_{jk} < 0 \end{cases} \tag{A.20}$$

Now the focus is placed in the *far* nodes **i** and **j** in order to obtain the matrix positions C_{ij} and C_{lj}:

$$\dot{m}_{ih}\phi_{ih} = \begin{cases} \dot{m}_{ih}(w_D\phi_h + w_C\phi_i + w_U\phi_j), & \text{if } \dot{m}_{ih} \geq 0 \\ (...), & \text{if } \dot{m}_{ih} < 0 \end{cases} = \begin{cases} (...), & \text{if } \dot{m}_{jk} \geq 0 \\ -\dot{m}_{jk}(w_D\phi_h + w_C\phi_i + w_U\phi_j), & \text{if } \dot{m}_{jk} < 0 \end{cases} \tag{A.21}$$

$$C_{ij} = \begin{cases} 0, & \text{if } \dot{m}_{jk} \geq 0 \\ -\dot{m}_{jk} w_U, & \text{if } \dot{m}_{jk} < 0 \end{cases} \tag{A.22}$$

$$\dot{m}_{kl}\phi_{kl} = \begin{cases} \dot{m}_{kl}(w_D\phi_l + w_C\phi_k + w_U\phi_j), & \text{if } \dot{m}_{kl} \geq 0 \\ (...), & \text{if } \dot{m}_{kl} < 0 \end{cases} = \begin{cases} \dot{m}_{kl}(w_D\phi_l + w_C\phi_k + w_U\phi_j), & \text{if } \dot{m}_{jk} \geq 0 \\ (...), & \text{if } \dot{m}_{jk} < 0 \end{cases} \tag{A.23}$$

$$C_{lj} = \begin{cases} \dot{m}_{jk}w_U, & \text{if } \dot{m}_{jk} \geq 0 \\ 0, & \text{if } \dot{m}_{jk} < 0 \end{cases} \tag{A.24}$$

Once C_{jj}, C_{jk}, C_{ji}, C_{jl}, C_{kj}, C_{ij} and C_{lj} have been derived, the symmetric and skew-symmetric part of the convective matrix when the QUICK scheme is employed are obtained:

$$S(C_{jj}) = C_{jj} \tag{A.25}$$

$$A(C_{jj}) = 0 \tag{A.26}$$

$$S(C_{jk}) = \frac{C_{jk} + C_{jk}^T}{2} = \begin{cases} 0.5\dot{m}_{jk}(w_D - w_C), & \text{if } \dot{m}_{jk} \geq 0 \\ 0.5\dot{m}_{jk}(w_C - w_D), & \text{if } \dot{m}_{jk} < 0 \end{cases} = |S(C_{jk})| = 0.5\,|\dot{m}_{jk}|\,|(w_D - w_C)| \tag{A.27}$$

$$A(C_{jk}) = \frac{C_{jk} - C_{jk}^T}{2} = \begin{cases} 0.5\dot{m}_{jk}(w_D + w_C), & \text{if } \dot{m}_{jk} \geq 0 \\ 0.5\dot{m}_{jk}(w_C + w_D), & \text{if } \dot{m}_{jk} < 0 \end{cases} = |A(C_{jk})| = 0.5\,|\dot{m}_{jk}|\,|(w_D + w_C)| \tag{A.28}$$

$$S(C_{ji}) = \frac{C_{ji} + C_{ji}^T}{2} = \begin{cases} 0.5\dot{m}_{jk}w_U, & \text{if } \dot{m}_{jk} \geq 0 \\ -0.5\dot{m}_{jk}w_U, & \text{if } \dot{m}_{jk} < 0 \end{cases} = |S(C_{ji})| = 0.5\,|\dot{m}_{jk}|\,w_U \tag{A.29}$$

$$A(C_{ji}) = \frac{C_{ji} - C_{ji}^T}{2} = \begin{cases} 0.5\dot{m}_{jk}w_U, & \text{if } \dot{m}_{jk} \geq 0 \\ 0.5\dot{m}_{jk}w_U, & \text{if } \dot{m}_{jk} < 0 \end{cases} = \\ |A(C_{ji})| = 0.5\,|\dot{m}_{jk}|\,w_U \quad \text{(A.30)}$$

$$S(C_{jl}) = \frac{C_{jl} + C_{jl}^T}{2} = \begin{cases} 0.5\dot{m}_{jk}w_U, & \text{if } \dot{m}_{jk} \geq 0 \\ 0.5\dot{m}_{jk}w_U, & \text{if } \dot{m}_{jk} < 0 \end{cases} = \\ |S(C_{jl})| = 0.5\,|\dot{m}_{jk}|\,w_U \quad \text{(A.31)}$$

$$A(C_{jl}) = \frac{C_{jl} - C_{jl}^T}{2} = \begin{cases} -0.5\dot{m}_{jk}w_U, & \text{if } \dot{m}_{jk} \geq 0 \\ 0.5\dot{m}_{jk}w_U, & \text{if } \dot{m}_{jk} < 0 \end{cases} = \\ |A(C_{jl})| = 0.5\,|\dot{m}_{jk}|\,w_U \quad \text{(A.32)}$$

As done for the Upwind scheme, in the off-diagonal values of the convective matrix the absolute value $|\cdot|$ has been derived.

References

[1] Brian P Leonard. A stable and accurate convective modelling procedure based on quadratic upstream interpolation. *Computer methods in applied mechanics and engineering*, 19(1):59–98, 1979.

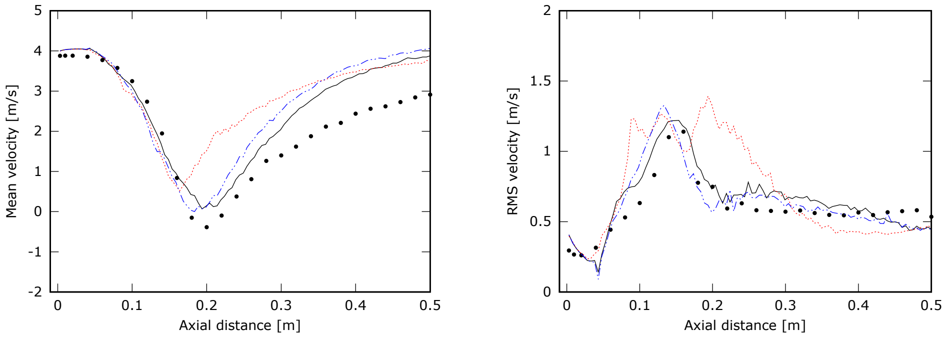
[2] M. S. Darwish and F. H. Moukalled. Normalized Variable and Space Formulation Methodology for High-Resolution Schemes. *Numerical Heat Transfer, Part B: Fundamentals*, 26:79–96, 1994.

Appendix B

Additional figures

B.1 Chapter 3

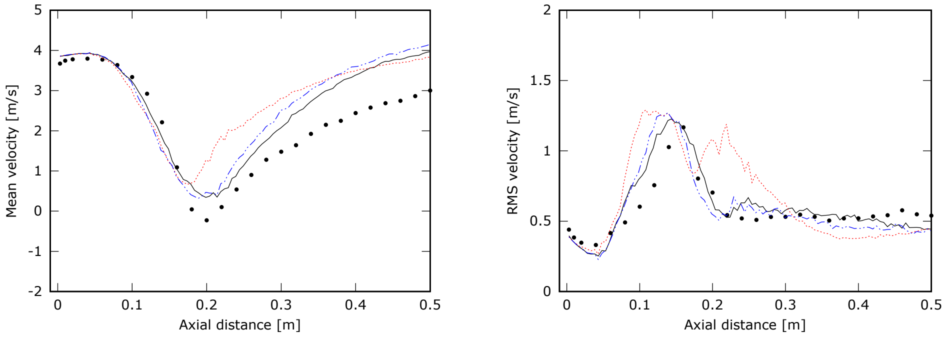
Additional figures for Chapter 3.



(a) Mean velocity.

(b) RMS velocity.

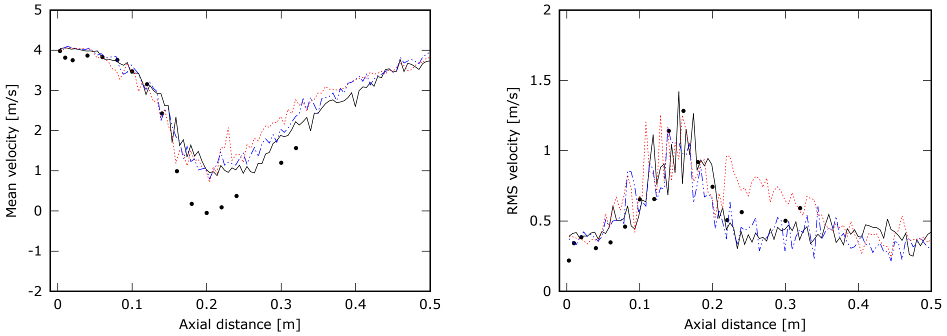
Figure B.1: Axial results of dispersed phase (particles 40 µm).



(a) Mean velocity.

(b) RMS velocity.

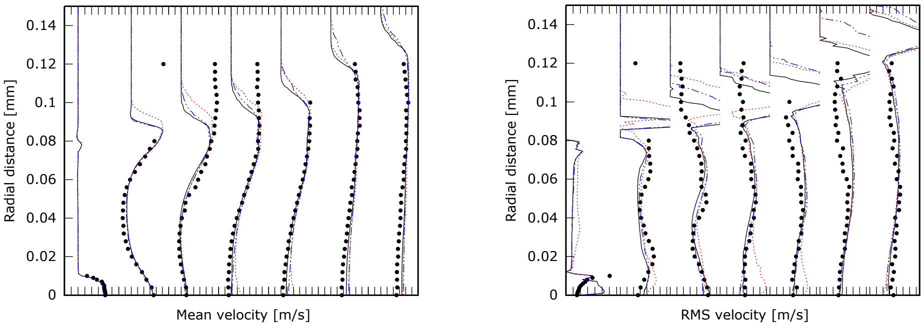
Figure B.2: Axial results of dispersed phase (particles 60 µm).



(a) Mean velocity.

(b) RMS velocity.

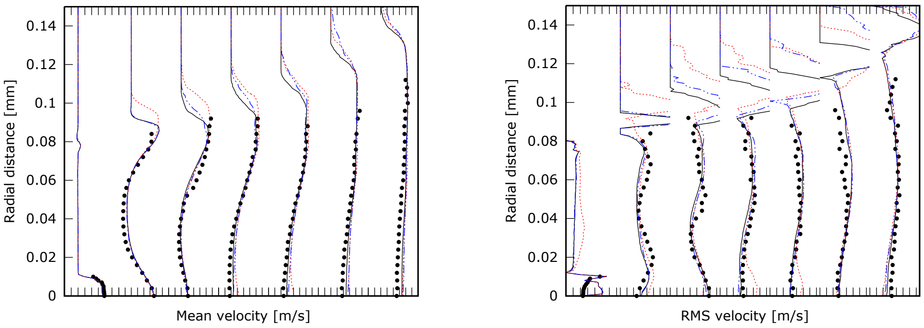
Figure B.3: Axial results of dispersed phase (particles 100 µm).



(a) Mean velocity.

(b) RMS velocity.

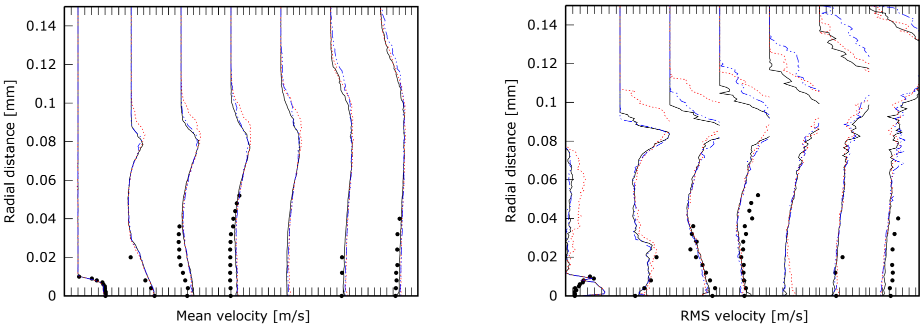
Figure B.4: Radial results of dispersed phase (particles $d_p = 40$ µm).



(a) Mean velocity.

(b) RMS velocity.

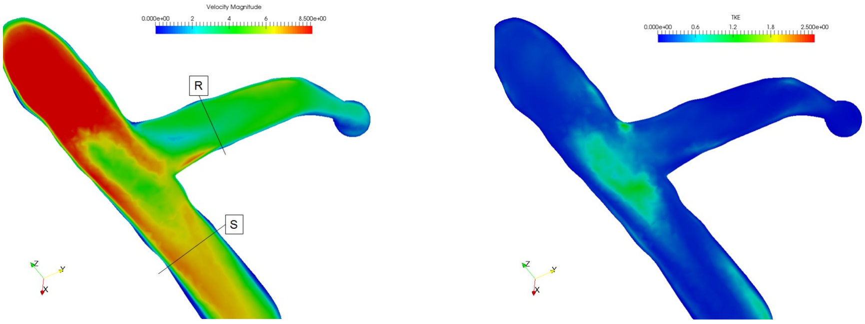
Figure B.5: Radial results of dispersed phase (particles $d_p = 60$ µm).



(a) Mean velocity.

(b) RMS velocity.

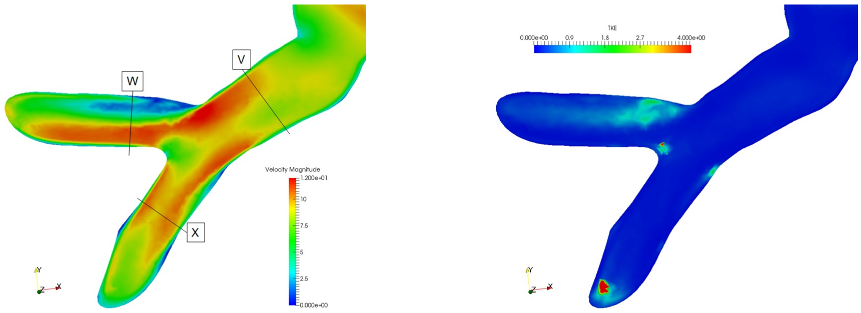
Figure B.6: Radial results of dispersed phase (particles $d_p = 100$ µm).



(a) Mean velocity.

(b) Mean TKE.

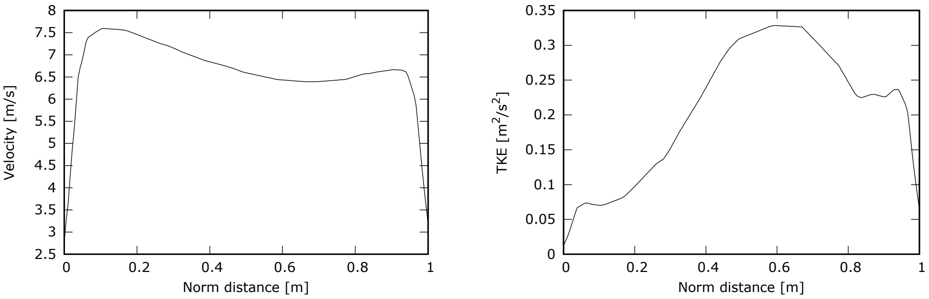
Figure B.7: Segment 6 in left lung.



(a) Mean velocity.

(b) Mean TKE.

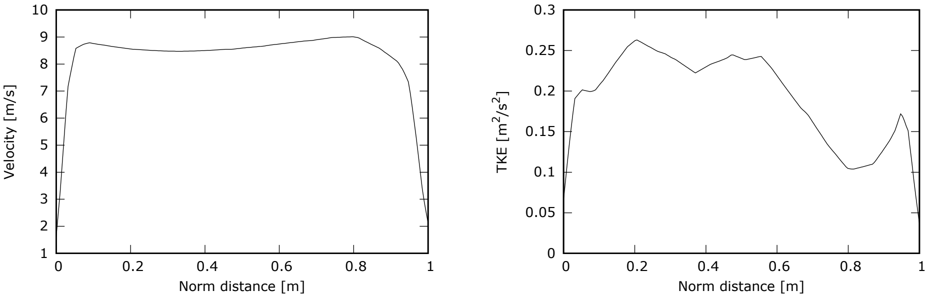
Figure B.8: Segment 11 in right lung.



(a) Mean velocity.

(b) Mean TKE.

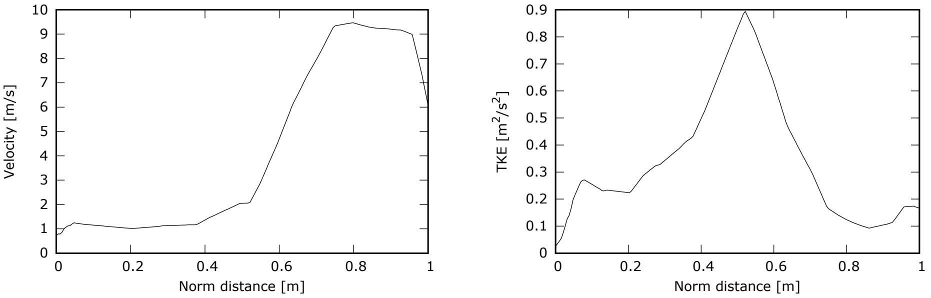
Figure B.9: Cross-section I.



(a) Mean velocity.

(b) Mean TKE.

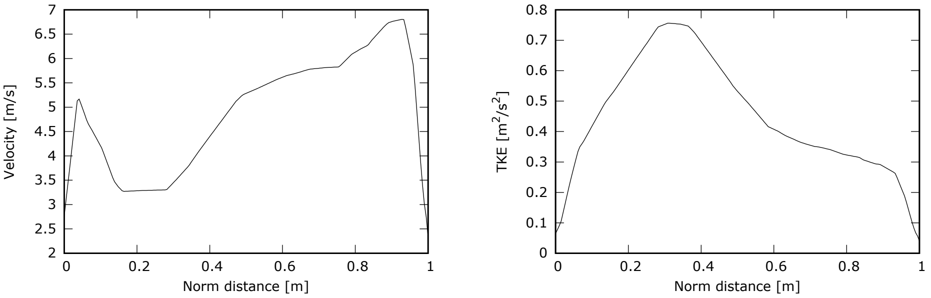
Figure B.10: Cross-section K.



(a) Mean velocity.

(b) Mean TKE.

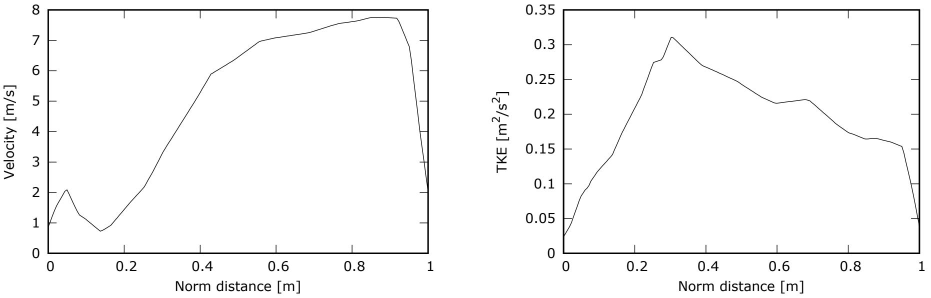
Figure B.11: Cross-section L.



(a) Mean velocity.

(b) Mean TKE.

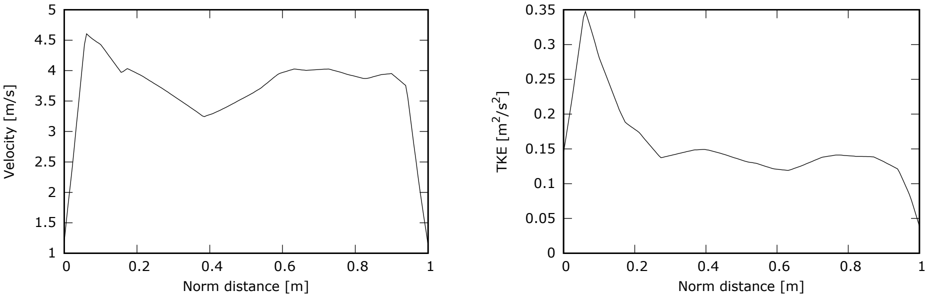
Figure B.12: Cross-section N.



(a) Mean velocity.

(b) Mean TKE.

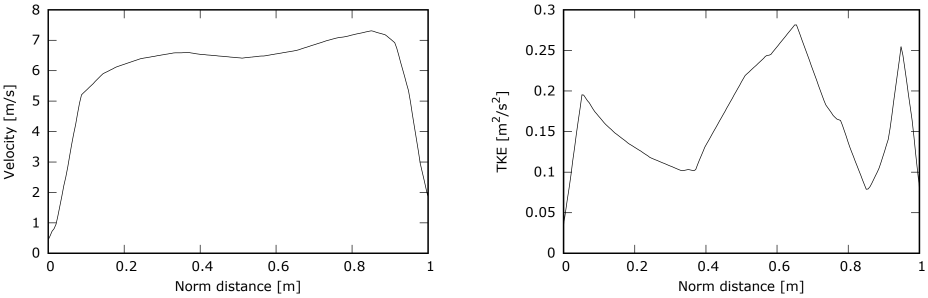
Figure B.13: Cross-section Q.



(a) Mean velocity.

(b) Mean TKE.

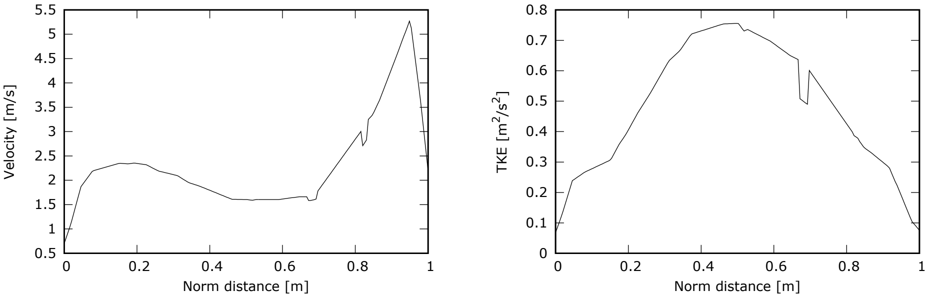
Figure B.14: Cross-section R.



(a) Mean velocity.

(b) Mean TKE.

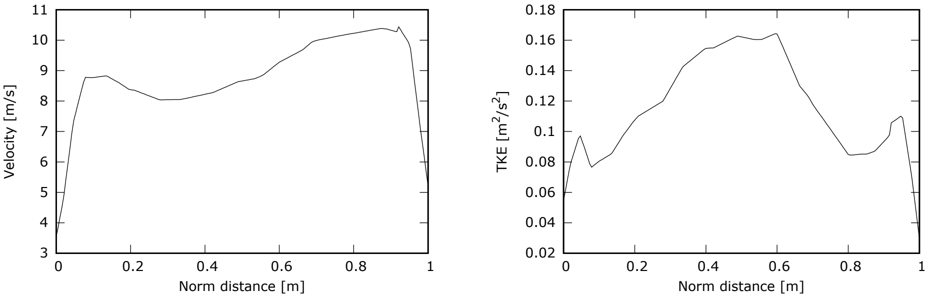
Figure B.15: Cross-section S.



(a) Mean velocity.

(b) Mean TKE.

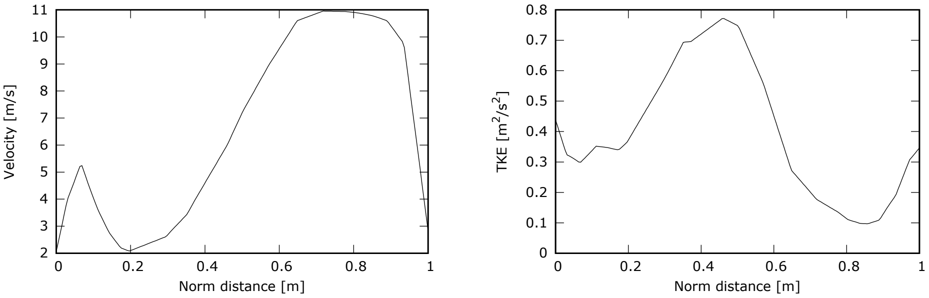
Figure B.16: Cross-section T.



(a) Mean velocity.

(b) Mean TKE.

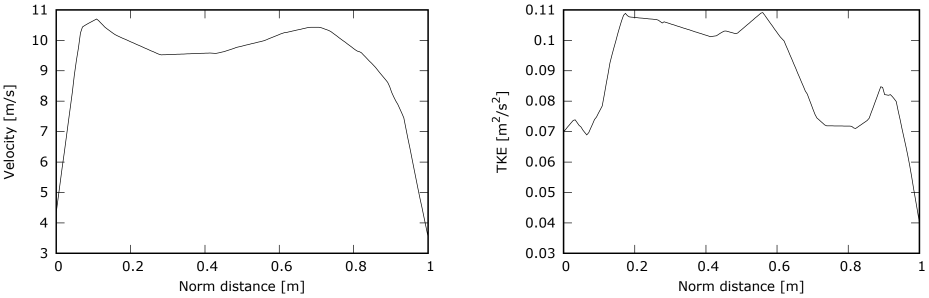
Figure B.17: Cross-section V.



(a) Mean velocity.

(b) Mean TKE.

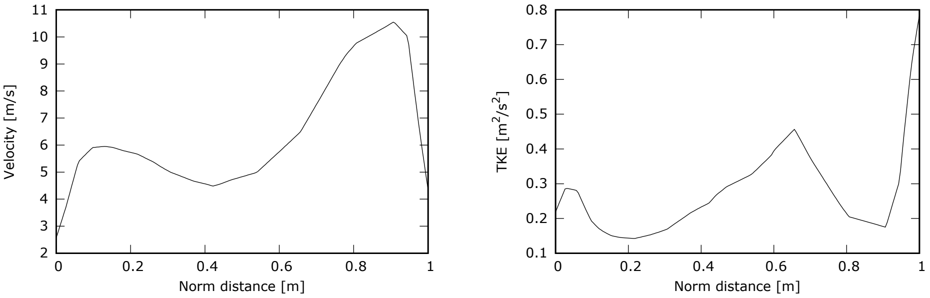
Figure B.18: Cross-section W.



(a) Mean velocity.

(b) Mean TKE.

Figure B.19: Cross-section X.



(a) Mean velocity.

(b) Mean TKE.

Figure B.20: Cross-section Z.

Appendix C

Main publications in the context of this thesis

International Journal Papers:

- J. Ventosa-Molina, J. Chiva, O. Lehmkuhl, J. Muela, C. D. Pérez-Segarra, and A. Oliva. *Numerical analysis of conservative unstructured discretisations for low mach flows.* International Journal for Numerical Methods in Fluids, 84(6):309–334, 2017.
- P. Koullapis, S.C. Kassinos, J. Muela, C. Perez-Segarra, J. Rigola, O. Lehmkuhl, Y. Cui, M. Sommerfeld, J. Elcner, M. Jicha, I. Saveljic, N. Filipovic, F. Lizal, and L. Nicolaou. *Regional aerosol deposition in the human airways: The sim-inhale benchmark case and a critical assessment of in silico methods.* European Journal of Pharmaceutical Sciences, pages -, 2017.
- J. Muela, D. Martínez, O. Lehmkuhl, C.D. Pérez-Segarra and A. Oliva. *New parallel method for adjacent disconnected unstructured 3D meshes.* International Journal of Computational Fluid Dynamics Vol. 30(6), pp. 388-394, 2016.

International Conference Papers:

- J. Muela, J. Ventosa, O. Lehmkuhl and A. Oliva. *Large eddy simulation of hydrogen autoignition in a preheated turbulent co-flow.* In Proceedings of the *8th Mediterranean Combustion Symposium 2013*, Çesme, Izmir, Turkey, 8-13 September 2013.
- J. Muela, J. Ventosa, O. Lehmkuhl and A. Oliva. *Study of the autoignition of a hydrogen jet in a turbulent co-flow of heated air using LES modelling.* In Proceedings of the *6th European Combustion Meeting 2013*, Lund, Sweden, 25-28 June 2013.

- J. Muela, R. Borrell, J. Ventosa, L. Jofre, O. Lehmkuhl, and A. Oliva. *An efficient parallelization method for the evaluation of chemical reaction rates.* In Proceedings of the *26th International Conference on Parallel Computational Fluid Dynamics 2014*, Trondheim, Norway, 22-24 May 2014.

- J. Muela, O. Lehmkuhl, C.D Pérez-Segarra and A. Oliva. *A new statistical model for subgrid dispersion in large eddy simulations of particle-laden flows.* Journal of Physics: Conference Series. Vol. 745. No. 3. IOP Publishing, 2016.

- J. Muela, O. Lehmkuhl and A. Oliva. *Study of stochastic models for subgrid dispersion in lagrangian-eulerian formulation.* In Proceedings of *10th International ERCOFTAC Workshop on Direct and Large-Eddy Simulation 2015*, Limassol, Cyprus, 27-29 May 2015.

- J. Muela, D. Martínez, O. Lehmkuhl, C.D. Pérez-Segarra and A. Oliva *New parallel method for adjacent disconnected unstructured 3D meshes.* In Proceedings of the *27th International Conference on Parallel Computational Fluid Dynamics 2015*, Montreal, Quebec, Canada, 17-20 May 2015.