POLITECNICO DI MILANO
School of Industrial and Information Engineering
Master of Science in Mechanical Engineering

CFD analysis of spark-ignition combustion in engine-like configurations

Supervisor: Prof. Tommaso Lucchini
Co-Supervisor: Lorenzo Sforza

Master Degree Thesis of:
Federico Clerici, ID 836929

Academic Year 2016-2017

Acknowledgements

In conclusione a questo percorso vorrei ringraziare tutta la mia famiglia: i miei genitori per aver reso possibile tutto questo e per avermi accompagnato in questi anni sostenendomi sempre, Elena e Luca per i preziosi consigli e i bei momenti passati insieme e anche ad Aurora e Tommaso che riescono sempre a strapparti un sorriso rallegrando anche i momenti più duri, i miei nonni Ambrogio, Anna e Agostino per avermi trasmesso i loro insegnamenti e aver vegliato su di me da lassù. Un particolare grazie va alla nonna Renata che con costanza mi ha sempre aiutato tutti i giorni con i suoi consigli e i suoi pranzi. Davvero grazie per essere stati la mia forza in questo percorso.

Un particolare grazie ad Alice che ha reso questi ultimi mesi pieni di felicità e gioia, capendomi sempre e trovando sempre il modo giusto per farmi affrontare le nuove sfide con serenità. Ringrazio moltissimo Alessandro e Cristina per i momenti trascorsi insieme, per avermi ascoltato sempre e per i loro preziosi consigli. Un grazie anche a Deborah ed Emanuele per le belle serate in compagnia.

Ringrazio tutti i miei amici: Marco, entrambi i Francesco, Giorgio, Davide, Walter, Edoardo per tutte le avventure trascorse insieme, per avermi spronato a dare sempre il massimo e per avermi dato il dono della loro amicizia.

Un ringraziamento va a tutti i professori del Politecnico di Milano per avermi aiutato a crescere culturalmente e personalmente in questi anni di università, in particolare ringrazio il Professor Tommaso Lucchini per avermi dato questa opportunità, e Lorenzo Sforza per avermi seguito con costanza fin dall'inizio, insegnandomi davvero molto.

Abstract

Objective of this work is the correct flow field initialization for combustion numerical analysis. To this end, suitable devices able to generate particular flow field in terms of velocity and turbulence intensity will be investigated finding a numerical set-up able to estimate the correct flow quantities. To support the numerical results the experimental findings available from Herweg et al. [14] for the test engine case and from Galmiche [11] for the vessel are used. The numerical analysis on these cases have been performed using OpenFOAM, an open-source CFD software, and Lib-ICE, a code based on OpenFOAM technology focused on internal combustion engine simulations developed by the Internal Combustion Engine (ICE) group of Politecnico di Milano. For both the analysed cases, a cold flow analysis has been carried out to initialize correctly the mixture, then after the numerical set-up is found, a reacting analysis have been performed for the test engine, adopting a recently updated solver for spark ignition combustion, developed by the ICE group of Politecnico di Milano and contained in the Lib-ICE library.

Keywords: CFD, internal combustion engine, combustion, cold-flow, OpenFOAM.

Sommario

L'obiettivo di questo lavoro è la corretta inizializzazione del campo di moto per l'analisi numerica della combustione. A tal fine, saranno studiati dispositivi idonei a generare un particolare campo di flusso in termini di velocità e intensità di turbolenza trovando un set-up numerico capace di stimare queste quantità correttamente. Per validare i risultati numerici vengono utilizzate misure sperimentali disponibili da Herweg et al. [14] per il caso del motore e da Galmiche [11] per il recipiente sferico. L'analisi numerica su questi casi è stata eseguita utilizzando OpenFOAM, un software CFD open source, e Lib-ICE, un codice basato sulla tecnologia OpenFOAM creato dal gruppo di Internal Combustion Engine (ICE) del Politecnico di Milano specificamente per simulazioni di motori a combustione interna. Per entrambi i casi analizzati è stata effettuata un'analisi del flusso non reagente per inizializzare correttamente la miscela, quindi dopo aver trovato il set-up numerico, è stata effettuata un'analisi di combustione per il motore, adottando un solutore recentemente aggiornato per la combustione ad accensione comandata, sviluppata dal gruppo ICE del Politecnico di Milano e contenuto nella Lib-ICE.

Parole chiave: CFD, motore a combustione interna, combustione, cold-flow, OpenFOAM.

Table of Contents

List of Tables

List of Figures

1. Introduction

Fuel consumption reduction is one of the key issues in the current Spark Ignition (SI) engine development. Some solutions could be adopted to increase global efficiency, like adopting direct injection and turbocharging with a significant reduction in pumping losses. However, further improvements are necessary, which involve the use of lean, highly diluted or stratified mixtures with high turbulence. In this context, we need to be mainly focused on initial combustion development, which is a complex process affected by ignition system properties, local flow, turbulence, fuel type and equivalence ratio. In particular, we have a greater need to focus on the effects of mixture turbulence and mean flow velocity on flame kernel formation. These effects were investigated by using a disc-shaped optically accessible side chamber showing that flame kernel is changed by turbulence: especially in an internal combustion engine, turbulence intensity is increased as the rotational speed increases. From these first considerations, it is possible to understand the importance of a correct flow initialization where a combustion process could start. A lot of time was dedicated to investigate the effect of the flow field in relation to turbulence and mean flow velocity also using particular devices suited to reproduce theoretical flow condition, like isotropic turbulence and quasi-homogeneous flow field in the combustion zone. Specifically, a constant volume vessel were used to investigate the effect of turbulence on flame kernel formation and development. The use of experimental rigs under different conditions in terms of equivalence ratio, turbulence intensity and mean flow velocity to analyse the effect of each parameter on the flame growth, require plenty of experimental test and time, which involves prohibitive costs. In the last decades, a new powerful technique has been developed to analyse systems related to fluid motion: Computational Fluid Dynamic (CFD). This technique allows to perform fluid dynamic studies without the necessity of expensive experimental tests: actually, there is no need to manufacture particular engine or equipment with optical access thus avoiding the use of very expensive measurement device. As CFD model needs to be validated by some experimental results, it is impossible to completely replace experimental test with CFD simulation. However, having few experimental data about a case it is possible to numerically validate the experimental findings and then carry out a lot of tests by changing certain parameters. In this way, CFD code is validated and can predict the flow field in similar conditions. In this work, two particular devices will be analysed: a test engine in which the combustion takes place in a side chamber and a stainless steel vessel with optical access, which is able to generate an isotropic turbulence and a quasi-homogeneous flow field in the central part where the combustion process occurs. A lot of experimental findings are available for the test engine while the numerical validation was not performed for all engine rotational speeds. As to the vessel, only experimental results are available and in this work we will make an attempt to tune a CFD model using a technique able to simulate the component rotation without having to move the mesh. This technique is required because the vessel has six fans able to generate a turbulent flow field and the blades' movement is managed by the Moving Reference Frame approach.

In the second chapter, we will briefly introduce the Computational Fluid Dynamic, then the fundamentals of fluid flows, turbulence models and discretization process will be presented, explaining the numerical models adopted in this work.

In the third chapter the structure of the code used is provided, OpenFOAM and Lib-ICE, i.e. a code developed by ICE Group of Politecnico di Milano, based on OpenFOAM.

In the fourth chapter we will illustrate the test engine experimental set-up adopted, then the mesh generation process is shown with a special focus on the different grids generated for validation, initialization and combustion analysis.

The fifth chapter is dedicated to the validation of numerical results inside a vessel that could generate isotropic turbulence for studying the initial stage of combustion process: also in this case, the experimental set-up is described, then the mesh generation and flow initialization procedure and validation is presented.

In the last chapter, some conclusions are drawn regarding the results obtained from this work.

2. Computational Fluid Dynamics

2.1 Introduction

Computational Fluid Dynamics (CFD) is the analysis of systems involving fluid flow, heat transfer and associated phenomena such as chemical reactions using computer-based simulation [12]. This technique is very powerful and spans a wide range of applications as follows:

- aircraft and vehicles' aerodynamics,
- ships hydrodynamics,
- power plant: combustion in Internal Combustion Engines (IC engines) and gas turbines,
- electrical and electronic equipment cooling,
- chemical process engineering,
- meteorology,
- biomedical engineering.

Aerospace industry has integrated CFD techniques into design and manufacture of aircraft and jet engines since 1960s more recently, thanks to the increase of computational power also the design of IC engines has used these methods. The ultimate aim of developments in the CFD field is to provide a capability comparable to other CAE (Computer-Aided Engineering) tools, such as stress analysis codes. The main reason why CFD has lagged behind is the tremendous complexity of the underlying behaviour, which precludes a description of fluid flows that is economical and sufficiently complete at the same time. The availability of affordable high-performance computing hardware and the introduction of user-friendly interfaces have led to a recent growth of interest and CFD is poised to entry into the wider industrial community in the 1990s. As to the economical aspect, the costs of suitable hardware for CFD analysis are smaller compared to those requested to buy good experimental facilities (test rig, measurements devices. . .). The variable costs of an experiment, in terms of facilities, hire and/or man-hour costs, is proportional to the number of data points and configurations tested. By contrast, CFD codes can produce extremely large volumes of results at virtually no added costs, and it is very cheap to perform parametric studies, for instance to optimise equipment performance. In addition to a substantial investment outlay, we also note that an organisation needs qualified people to run the codes, communicate their results and briefly consider the modelling skills required by CFD users. Moreover, there are several unique advantages of CFD over experiment-based approaches to fluid systems design:

- reduction of lead times and costs of new designs,
- ability to study systems where controlled experiments are difficult or impossible to perform (e.g. very large systems),
- ability to study systems under hazardous conditions at and beyond their normal performance limits (e.g. safety studies and accident scenarios),
- practically unlimited level of results' details.

2.2 CFD for Internal Combustion Engine applications

Although every single engine component has been exploited to give the maximum potential, the use of simulation tools such as CFD has opened ways in designing IC engines of tomorrow. Businesses, especially automotive manufacturers, look upon CFD as one of the very few tools able to decrease cost, increase revenues and contribute to innovation, as market is getting increasingly competitive.

Modern CFD solvers' ability to simulate the flow behaviour is what most engine designers are interested in. From a macroscopic flow behaviour to individual species study, combustion engineers can obtain enough performance data about the engine to deal with. This non-destructive testing approach predicts the effects of multiple physics on the engine construction and provides information on the amount of harmful pollutant generation. Interestingly, all this information can be obtained right from the early design stages, which directly reduce the number of prototyping test trials required.

The ability to predict the heat transfer process within the combustion chamber has helped designers to apply innovative approaches in developing engines that work even in the most extreme thermal conditions. On the other hand, conventional physical tests have not lost their importance, but serve as a source of validating the simulation results. There is no single approach that designers should rely upon, especially when their main concern is promoting innovation. Mathematical calculations do play a crucial role in the design process, but with the integration of computational fluid dynamics and physical tests, comprehensive information can be obtained; allowing design engineers to identify unexplored dimensions leading towards innovation [1].

In the light of these considerations, the efficiency is finally a very important parameter which designers are interested in, especially due to the new restrictive pollutant regulations. The efficiency is related to the ability of converting fuel chemical energy into mechanical power available at the engine crankshaft: this means that combustion has a fundamental role for having high efficiency engines. Combustion process is influenced by the turbulent flow's nature found in the combustion chamber, turbulence improves air and fuel mixing by promoting a faster and complete combustion, reducing the unburned species, increasing the fuel efficiency as more fuel is exploited for producing power. The generation of a suitable flow motion and turbulence is an aspect regarding the design of intake manifolds and valve ports to generate the desired flow field, in Figure 2.1 3 main different organised flow motions are shown:

- Swirl
- Tumble
- Squish

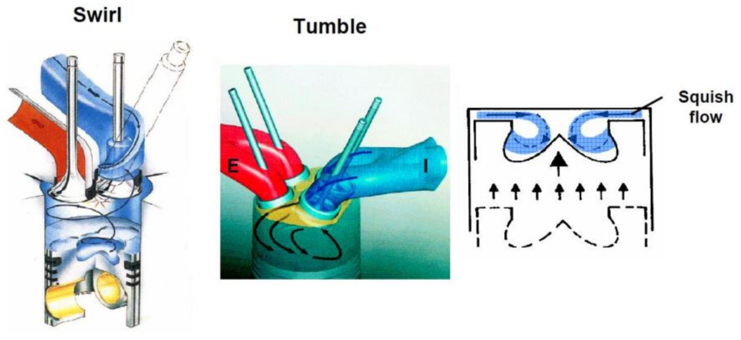


Figure 2.1: Organised flow motions: swirl, tumble and squish

Swirl and squish motions enhance mixing and combustion velocity while tumble motions are very helpful in increasing the cylinder filling. In this study, the interest is focused on the effect of the generated turbulence in the first stages of the combustion process. For this reason, it is very important to have a satisfactory validation of the non-reacting flow inside the combustion chamber up to the ignition time, before the combustion starts.

Due to the complexity of the processes taking place inside an engine, simplified models of the underlying physics are required, from the turbulence to the chemical model. Modelling is a tricky phase: when certain aspects are neglected, results might be not accurate and reliable. Nevertheless, making some assumptions is essential to reduce the complexity to a manageable level while preserving the main aspects of the problem, leading to acceptable computational costs. As to these last considerations, IC engines CFD modelling could be classify based on:

- Flow type.
- Turbulence model.
- Mesh types.

The flow inside an engine undergoes different phenomena; firstly it is introduced in the cylinder, then it is compressed, burnt and, finally, expanded. Before it enters in the cylinder, or when the intake valve is closed, the fuel-air mixing process takes place: this phenomenon requires a modelling strategy as well. To simplify the entire cycle, different phases could be identified:

- Cold flow: this part models the non-reacting phase of the flow, the different strokes (intake, compression, expansion and exhaust) could be simulated [22].
- Fuel-air mixing: this phase is related to the fuel injection simulation inside the air stream (direct or indirect injection) [21].
- Reacting flow: in which the ignition and the following flame propagation is modelled and simulated [17].

In this way, the entire cycle that takes place inside the cylinder could be studied in different phases, thus simplifying the modelling and reducing the computational effort.

The second important aspect is the turbulence model, if the interest is in predicting the mean cycle, the average flow field and expected turbulence intensity the **RANS modelling** could be adopted. The advantages of this model is the reliability and ability in obtaining results faster than LES modelling. On the other hand, if the interest is in predicting cyclic variability the **LES approach** could be adopted, this type of model is expected to provide more detailed results but the computational and numerical modelling effort is higher compared to RANS approach [32].

The third aspect is the mesh types since in an internal combustion engine piston and valves move during the cycle, suitable meshes need to be generated to accommodate this movement. Different categories of grids generator algorithm are available:

- **Body-fitted multiple deforming meshes [19]**: the grid is generated and remains valid in a certain crank angle interval, so there is the need to generate multiple meshes to cover the entire simulation. The mesh deformation is such that grid points are accommodated by boundary motion. Considering these aspects, we need to map the solution from one mesh to the next one.
- **Topological changes [18]**: with this mesh movement algorithm there is the possibility to run the entire simulation with only one grid or a very reduced set. Topology modifiers operate on the mesh by adding/removing cells, faces, points. This mesh type is characterized by an easy mesh motion of the grid since there is only the need to displace points close to the moving boundaries.
- **Local adaptive mesh refinement [29]**: the grid is continuously regenerated at each time-step to accommodate valve and piston motion. Mesh is locally refined close to the boundaries and, eventually, in regions where strong gradients are present (velocity, mixture fraction,...)
- **Immersed boundary [24]**: a reference geometry configuration is used as computational domain, then moving parts (piston, valves) are taken into account by introducing suitable body force fields in the computational domain.

As regards mesh types, sometimes, if the movement of some components inside the domain is a rotation, we have two options: move the mesh or change the fluid equation taking into account the rotation effect. This last modelling technique is known as **Moving Reference Frame (MRF)**.

An MRF assumes that an assigned volume has a constant rotation speed and the non-wall boundaries are surfaces of revolution (e.g., cylindrical, spherical, conical). MRF is equivalent to running a rotational simulation and then observing the results at the instant equivalent to the rotor's position within the MRF. MRF assumes a weak interaction between MRF volume and the surrounding stationary volumes [3].

Using this approach, the momentum equation is modified to incorporate the additional Coriolis acceleration term where the part that should be rotated is present. In particular, only the cells that are attached to/in between the rotating component (like the fans blades for the Orleans vessel) need to be modelled with MRF approach because they are the only cells that perceived the rotation. Consequently, by solving this modified momentum equation, the flow around rotating component could be modelled without mesh motion.

The implementation of this technique in OpenFOAM environment will be explained in Section 5.2.2.

Governing equations and, later, their discretization and numerical solution strategy are approached below.

2.3 Governing equations

Computational Fluid Dynamic codes are based on conservation laws formulation. Conservation laws can be obtained by considering a given quantity of matter or control mass (CM) and its extensive properties, such as mass, momentum and energy. This approach is used to study solid bodies' dynamics, where the CM (sometimes called the system) is easily identified. In fluid flows, however, it is difficult to follow a parcel of matter. It is more convenient to deal with the flow within a certain spatial region called control volume (CV), rather than in a parcel of matter that quickly passes through the region of interest. This analysis method is called control volume approach. The conservation equations are:

- conservation of mass: **continuity equation**,
- conservation of momentum (Newton Law): **momentum equation**,
- conservation of energy (first principle of thermodynamics): **energy equation**.

In 3D this is a 5 equations system, momentum equation is a vectorial quantity. To determine the state of the fluid, additional information is required: equation of state is used for this purpose. Under the assumption of thermodynamic equilibrium (acceptable if strong shockwaves do not occur), this equation provides the link between the thermodynamic variables ($p,\ T,\ e,\ \rho$).

This set of equations when applied to viscous flow is known as **Navier-Stokes equations** while when applied to an inviscid flow it is known as **Euler equations**.

A fundamental concept in the formulation of conservation laws is the **flux**: the variation of the total amount of a quantity inside a given domain is equal to the balance between the amount of that quantity entering and leaving the considered domain, plus the contribution from any sources generating that quantity. A conservation equation for a generic scalar quantity ϕ:

$$\frac{\partial}{\partial t}\int_{\Omega}\phi \mathrm{d}\Omega = -\oint_{S}\overrightarrow{F}\cdot \mathrm{d}\overrightarrow{S} + \int_{\Omega} Q_v \mathrm{d}\Omega + \oint_{S}\overrightarrow{Q_s}\mathrm{d}\overrightarrow{S} \tag{2.1}$$

Where:

- Ω is the controlled volume domain,
- S is the surface defining the volume domain,
- $\overrightarrow{F}$ is the flux,
- Q$_v$ and Q$_s$ are volume and surface source terms, respectively.

Conservation equation written like Eq. (2.1) is the integral form for a generic scalar quantity. Applying Gauss' theorem to surface integrals, it is possible to write the conservation equation in differential form:

$$\oint_S \overrightarrow{F} \cdot \mathrm{d}\overrightarrow{S} = \int_\Omega \overrightarrow{\nabla} \cdot \overrightarrow{F} \mathrm{d}\Omega \tag{2.2a}$$

$$\frac{\partial}{\partial t} \int_\Omega \phi \mathrm{d}\Omega + \int_\Omega \overrightarrow{\nabla} \cdot \overrightarrow{F} \mathrm{d}\Omega = \int_\Omega Q_v \mathrm{d}\Omega + \int_\Omega \overrightarrow{\nabla} \cdot \overrightarrow{Q}_s \mathrm{d}\Omega \tag{2.2b}$$

$$\frac{\partial \phi}{\partial t} + \overrightarrow{\nabla} \cdot \overrightarrow{F} = Q_v + \overrightarrow{\nabla} \cdot \overrightarrow{Q}_s \tag{2.2c}$$

Since Eq. (2.2b) is valid for an arbitrary volume Ω it must be valid for any point of the flow and it is possible to write them in a differential form like Eq. (2.2c). From Eq. (2.2c) it is possible to see that fluxes appears only under divergence operator, and they could be generated from two contributions:

- **Convective** contribution: it represents the amount of ϕ that is carried out away or transported by the fluid flow $\overrightarrow{F}_C = \phi\overrightarrow{U}$. The convective flux is proportional to the flow velocity and has directional properties. It appears as a first order partial derivative term and represents a non-linear term as the velocity field depends on the transported variables.

- **Diffusive** contribution: It represents the amount of ϕ that is carried away or transported by the presence of its gradient $\overrightarrow{F}_D = -k\rho\overrightarrow{\nabla}\phi$. The diffusive flux appears as a second order partial derivative term, i.e. under the Laplace operator.

The differential conservation equation written considering the two types of fluxes then becomes:

$$\frac{\partial \phi}{\partial t} + \overrightarrow{\nabla} \cdot \phi\overrightarrow{U} = \overrightarrow{\nabla} \cdot (k\rho\overrightarrow{\nabla}\phi) + Q_v + \overrightarrow{\nabla} \cdot \overrightarrow{Q}_s \tag{2.3}$$

If the conserved property is a vector, the flux becomes a tensor as well as the surface source term (stress tensor), whereas the volume source terms become a vector. The conservation equation for a vector property is represent by Eq. (2.4a).

$$\frac{\partial}{\partial t} \int_\Omega \overrightarrow{\phi} \mathrm{d}\Omega = -\oint_S \overline{\overline{F}} \cdot \mathrm{d}\overrightarrow{S} + \int_\Omega \overrightarrow{Q}_v \cdot \mathrm{d}\Omega + \oint_S \overline{\overline{Q_s}} \cdot \mathrm{d}\overrightarrow{S} \tag{2.4a}$$

$$\frac{\partial}{\partial t} \int_\Omega \overrightarrow{\phi} \mathrm{d}\Omega = -\int_\Omega \overrightarrow{\nabla} \cdot \overline{\overline{F}} \mathrm{d}\Omega + \int_\Omega \overrightarrow{Q}_v \cdot \mathrm{d}\Omega + \int_\Omega \overrightarrow{\nabla} \cdot \overline{\overline{Q_s}} \mathrm{d}\Omega \tag{2.4b}$$

As for scalar quantity, since the integral formulation is valid for any arbitrary control volume, the differential form is derived:

$$\frac{\partial}{\partial t}\overrightarrow{\phi} + \overrightarrow{\nabla} \cdot (\overline{\overline{F}} - \overline{\overline{Q}}_S) = \overrightarrow{Q}_V \tag{2.5}$$

From these general formulations, the conservation equations of fluid dynamics can be derived [12, 23].

Continuity equation: this conservation equation is referred to a scalar property $\phi = \rho$.

$$\frac{\partial}{\partial t}\int_{\Omega}\rho \mathrm{d}\Omega + \oint_{S}\rho\vec{U}\cdot \mathrm{d}\vec{S} = 0 \tag{2.6a}$$

$$\frac{\partial \rho}{\partial t} + \vec{\nabla}\cdot(\rho\vec{U}) = 0 \tag{2.6b}$$

Eq. (2.6b) is the differential form of continuity equation. Diffusion term is not included in continuity equation because mass does not diffuse.

Momentum equation: this conservation equation is referred to a vectorial property $\phi = \rho\vec{U}$. In the momentum equation the source terms account for internal force (stress) and external volume forces (gravity, applied forces). Internal forces are the expression of fluid's deformability and since it depends on the position and orientation of the surface it acts on, it must be represented by a tensor. Assuming Newtonian fluid, the stress tensor is:

$$\overline{\overline{\sigma}} = -p\overline{\overline{I}} + \overline{\overline{\tau}} \tag{2.7}$$

Where $-p\overline{\overline{I}}$ is the isotropic pressure component and $\overline{\overline{\tau}}$ is the viscous shear stress, representing the internal friction force of fluid layers against each other. Newton's constitutive law suggests the following model for the shear stress:

$$\overline{\overline{\tau}}_{ij} = \mu\left[\left(\frac{\partial U_j}{\partial x_i} + \frac{\partial U_i}{\partial x_j}\right) - \frac{2}{3}(\vec{\nabla}\cdot\vec{U})\delta_{ij}\right] \tag{2.8}$$

Inside the control volume the internal forces cancel out, whereas they do not have any counterpart on the volume surface. The integral form of the momentum conservation equation will be:

$$\frac{\partial}{\partial t}\int_{\omega}\rho\vec{U}\mathrm{d}\Omega + \oint_{S}(\rho\vec{U}\vec{U})\cdot \mathrm{d}\vec{S} = \int_{\Omega}\rho\vec{f}_e \mathrm{d}\Omega + \oint_{S}\overline{\overline{\sigma}}\cdot \mathrm{d}\vec{S} \tag{2.9}$$

Applying Gauss' theorem:

$$\frac{\partial}{\partial t}\int_{\omega}\rho\vec{U}\mathrm{d}\Omega + \int_{\Omega}\vec{\nabla}(\rho\vec{U}\vec{U})\mathrm{d}\Omega = \int_{\Omega}\rho\vec{f}_e \mathrm{d}\Omega + \int_{\Omega}\vec{\nabla}\cdot\overline{\overline{\sigma}}\mathrm{d}\Omega \tag{2.10}$$

Finally, as for the continuity equation, the differential form is derived:

$$\frac{\partial(\rho\vec{U})}{\partial t} + \vec{\nabla}\cdot(\rho\vec{U}\vec{U} + p\overline{\overline{I}} - \overline{\overline{\tau}})\mathrm{d}\Omega = \rho\vec{f}_e \tag{2.11}$$

Note that momentum diffuses into the fluid via viscous shear stresses, in Eq. (2.11) the Laplace operator is not explicitly written but it is expressed in the shear stress definition (Eq. (2.8))

Energy equation: the conserved quantity is the total energy E, defined as the sum of fluid internal energy plus its kinetic energy per unit mass $\phi = \rho E$, where $E = e + \frac{1}{2}\overrightarrow{U}^2$. According to previous considerations on the terms constituting a conservation equation, each term is described:

- Flux: for the energy equation two types of fluxes are considered: convective and diffusive (thermal conductivity of the fluid) fluxes:
  - Convective term $\overrightarrow{F}_C$ is: $\overrightarrow{F}_c = \rho E \overrightarrow{U}$.
  - Diffusive term $\overrightarrow{F}_D$ is: $\overrightarrow{F}_D = -k\overrightarrow{\nabla}T$, that is the thermal conductivity of the fluid.
- Volume source terms are represented by the work of the volume forces f_e and heat sources:
$$Q_V = \rho \overrightarrow{f}_e \cdot \overrightarrow{U} + q_H \tag{2.12}$$
- Surface sources $\overrightarrow{Q}_S$ result from the work done on the fluid by t pressure and internal shear stress acting on control volume surface:
$$\overrightarrow{Q}_S = \overline{\overline{\sigma}} \cdot \overrightarrow{U} = -p\overrightarrow{U} + \overline{\overline{\tau}} \cdot \overrightarrow{U} \tag{2.13}$$

Considering all these terms, the energy equation can be written in its integral form:

$$\frac{\partial}{\partial t}\int_\Omega \rho E \mathrm{d}\Omega + \oint_S \rho E \overrightarrow{U} \cdot \mathrm{d}\overrightarrow{S} = \oint_S k\overrightarrow{\nabla}T \mathrm{d}\overrightarrow{S} + \int_\Omega (\rho \overrightarrow{f}_e \cdot \overrightarrow{U} + q_H)\mathrm{d}\Omega + \oint_S (\overline{\overline{\sigma}} \cdot \overrightarrow{U}) \cdot \mathrm{d}\overrightarrow{S} \tag{2.14}$$

In differential form the energy equation will be:

$$\frac{\partial \rho E}{\partial t} + \overrightarrow{\nabla} \cdot (\rho E \overrightarrow{U}) = \overrightarrow{\nabla} \cdot (k\overrightarrow{\nabla}T) + \overrightarrow{\nabla} \cdot (\overline{\overline{\sigma}} \cdot \overrightarrow{U}) + \rho \overrightarrow{f}_e \cdot \overrightarrow{U} + q_H \tag{2.15}$$

Then, if the stress tensor is written in its isotropic and viscous shear stress and introducing enthalpy, the energy conservation equation becomes:

$$\frac{\partial \rho H}{\partial t} + \overrightarrow{\nabla} \cdot (\rho H \overrightarrow{U} - k\overrightarrow{\nabla}T - \overline{\overline{\tau}}) = \frac{\partial p}{\partial t} + \rho \overrightarrow{f}_e \cdot \overrightarrow{U} + q_H \tag{2.16}$$

These three conservation equations are sufficient to study the so-called non-reacting case, while when some complex phenomena such as combustion happen the mass conservation equation must be solved by the code. In particular, in this study combustion analyses are performed on the test engine presented in Section 4.1 so a briefly introduction on the conservation equation for reacting flow is given [31]. The three main differences between conservation equations for reacting flows and the usual Navier-Stokes equations for non-reacting cases are:

- a reacting gas is a non-isothermal mixture of multiple species (hydrocarbons, oxygen, water, carbon dioxide, etc.) which must be tracked individually. Also thermodynamic data are more complex than in classical aerodynamics because heat capacities in a reacting gas change significantly with temperature and composition,
- species react chemically, and the rate at which these reactions take place requires specific modelling,
- since the gas is a mixture of gases, transport coefficients (heat diffusivity, species diffusion, viscosity, etc.) require specific attention.

Combustion involves multiple species reacting through multiple chemical reactions. Species are characterized through their mass fractions Y_k for k = 1 to N, where N is the number of involved species in the reactions. The mass fractions Y_k are defined by:

$$Y_k = \frac{m_k}{m} \tag{2.17}$$

where m_k is the mass of species k present in a given volume V and m is the total mass of gas in this volume. The primitive variables for a three-dimensional compressible reacting flow are:

- the density $\rho = \frac{m}{V}$,
- the three-dimensional velocity field u_i,
- one variable for energy (or pressure or enthalpy or temperature T),
- and the mass fractions Y_k of the N reacting species.

Non-reacting cases require solving for 5 variables while reacting cases require solving for N+5 variables: this is the first significant effort needed to compute reacting flows. Total mass conservation equation is unchanged compared to non-reacting flows, since combustion does not generate mass.

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u_i}{\partial x_i} = 0 \tag{2.18}$$

The mass conservation equation for species k is written

$$\frac{\partial \rho Y_k}{\partial t} + \frac{\partial}{\partial x_i}(\rho(u_i + V_{k,i})Y_k) = \dot{\omega}_k \qquad \text{for k=1,N} \tag{2.19}$$

where $V_{k,i}$ is the i-component of the diffusion velocity V_k of species k and $\dot{\omega}_k$ is the reaction rate of species k. By definition, the following relation subsists:

$$\sum_{k=1}^{N} Y_k V_{k,i} = 0 \quad \text{and} \quad \sum_{k=1}^{N} \dot{\omega}_k = 0 \tag{2.20}$$

The diffusion velocities V_k of the N species are obtained by solving the system ([33]):

$$\nabla X_p = \sum_{k=1}^{N} \frac{X_p X_k}{D_{pk}}(V_k - V_p) + (Y_p - X_p)\frac{\nabla P}{P} + \frac{\rho}{p}\sum_{k=1}^{N} Y_p Y_k (f_p - f_k) \quad \text{for p=1,N} \tag{2.21}$$

where $D_{pk} = D_{kp}$ is the binary mass diffusion coefficient of species p into species k and X_k is the mole fraction of species k: $X_k = Y_k W / W_k$. The system (2.21) is a linear system of size N^2 which must be solved in each direction at each point and at each instant for unsteady flows. This task is difficult and costly, and most codes use simplified approaches.

The momentum conservation equation is the same in reacting and non-reacting flows:

$$\frac{\partial}{\partial t}\rho u_j + \frac{\partial}{\partial x_i}\rho u_i u_j = -\frac{\partial p}{\partial x_j} + \frac{\partial \tau_{ij}}{\partial x_i} + \rho \sum_{k=1}^{N} Y_k f_{k,j} = \frac{\partial \sigma_{ij}}{\partial x_i} + \rho \sum_{k=1}^{N} Y_k f_{k,j} \tag{2.22}$$

where $f_{k,j}$ is the volume force acting on species k in direction j. Even though this equation does not include explicit reaction terms, the flow is modified by combustion: the dynamic viscosity μ strongly changes because temperature varies in a ratio from 1:8 or 1:10. Density also changes in the same ratio and dilatation through the flame front increases all speeds by the same ratio. Consequently, the local Reynolds number varies much more than in a non-reacting flow: although the momentum equations are the same with and without combustion, the flow behaviour is very different [31].

The energy conservation equation requires the greatest attention because multiple forms exist. Starting from the conservation equation for total energy e_t ([16]):

$$\rho\frac{De_t}{Dt} = \frac{\partial \rho e_t}{\partial t} + \frac{\partial}{\partial x_i}(\rho u_i e_t) = -\frac{\partial q_i}{\partial x_i} + \frac{\partial}{\partial x_j}(\sigma_{ij} u_i) + \dot{Q} + \rho \sum_{k=1}^{N} Y_k F_{k,i}(u_i + V_{k,i}) \tag{2.23}$$

where $\dot{Q}$ is the heat source term (due for example to an electric spark). $\rho \sum_{k=1}^{N} Y_k F_{k,i}(u_i + V_{k,i})$ is the power produced by volume forces f_k on species k. The energy flux q_i is:

$$q_i = -\lambda \frac{\partial T}{\partial x_i} + \rho \sum_{k=1}^{N} h_k Y_k V_{k,i} \tag{2.24}$$

The energy flux consists of two terms:

- heat diffusion term expressed by Fourier's law $(-\lambda \frac{\partial T}{\partial x_i})$,
- diffusion term associated with the diffusion of species with different enthalpies, which is specific of multi-species gas.

As already done for non-reacting conservation equation, introducing the relation between total enthalpy and total energy ($h_t = e_t + p/\rho$), the energy conservation equation for reacting flow can be written:

$$\rho \frac{Dh_t}{Dt} = \frac{\partial \rho h_t}{\partial t} + \frac{\partial}{\partial x_i}(\rho u_i h_t) = \frac{\partial p}{\partial t} - \frac{\partial q_i}{\partial t} + \frac{\partial}{\partial x_j}(\tau_{ij} u_i) + \dot{Q} + \rho \sum_{k=1}^{N} Y_k f_{k,i}(u_i + V_{k,i}) \tag{2.25}$$

In the next section, the turbulence modelling is going to be discussed and a brief introduction on turbulence along with the description of some turbulence models is provided.

2.4 Turbulence models

There are many opportunities to observe turbulent flows in everyday surroundings, whether it could be smoke from a chimney, water in a river or waterfall, or a strong wind buffeting. Observing a waterfall, it is possible to understand that the flow is unsteady, irregular, seemingly random and chaotic, and surely the motion of every eddy or droplet is unpredictable [26]. Turbulence modelling is one of the three key elements in CFD, and due to its unpredictability, a considerable modelling effort is required. Turbulence is characterized by 5 quantities [25]:

1. **Fluctuations**: Turbulent flows contain fluctuations in the dependent-field quantities (velocity, pressure, temperature, etc.) even when the flow's boundary conditions are steady. Turbulent fluctuations appear to be irregular, chaotic, and unpredictable.

2. **Nonlinearity**: Turbulence is found to occur when the relevant nonlinearity parameter, i.e. the Reynolds number *Re*, the Rayleigh number *Ra*, or the inverse Richardson number Ri^{-1}, exceeds a critical value. The turbulence nonlinearity is evident since it is the final state of a nonlinear transition process.

Once the critical parametric value is exceeded, small perturbations can grow spontaneously and may equilibrate as finite amplitude disturbances. However, the new equilibrium state can become unstable to more complicated disturbances, and so on, until the flow eventually reaches a non-repeating unpredictable state (turbulence).

3. **Vorticity**: Turbulence is characterised by fluctuating vorticity. A cross-section view of a turbulent flow typically appears as a diverse collection of streaks, strain regions, and swirls of various sizes that deform, coalesce, divide, and spin. Turbulence always involves a range of eddy sizes and the size range increases as Reynolds number increases.

4. **Dissipation**: On average, the vortex stretching mechanism transfers fluctuation energy and vorticity to smaller and smaller scales via nonlinear interactions, until velocity gradients become so large that the energy is converted into heat (i.e., dissipated) by the action of viscosity and the motion of the smallest eddies. Persistent turbulence, therefore, requires a continuous supply of energy to make up for this energy loss.

5. **Diffusivity**: Due to the prevalence of agitation and overturning motions (macroscopic mixing), turbulent flows are characterized by a rapid rate of mixing and diffusion of species, momentum, and heat compared to equivalent laminar flows that lack fluctuations.

In turbulence theory the most widely accepted concept is the *energy cascade*: this concept plays a crucial role in the study of well-developed turbulence. This was first expressed by Lewis F. Richardson in the 1920s:
"Big whirls have little whirls that feed on their velocity, and little whirls have lesser whirls and so on to viscosity" [28].
Richardson's notion of turbulence was that a turbulent flow consists of "eddies" of different sizes. The size defines a peculiar length scale for the eddies, which are also characterized by flow velocity scales and time scales (turnover time) dependent on the length scale. Large eddies are unstable and eventually break up, originating smaller eddies, and the kinetic energy of the initial large eddy is divided into the smaller eddies stemmed from it. These smaller eddies undergo the same process, giving rise to even smaller eddies that inherit the energy of their predecessor eddy, and so on. In this way, the energy is passed down from the large scales of the motion to smaller scales until reaching a sufficiently small length scale such that the viscosity of the fluid can effectively dissipate the kinetic energy into internal energy. This concept is explained graphically in Figure 2.2 and in Figure 2.3 introducing also another concept: the level of approximation.

Three different level of approximation for the turbulence are available:

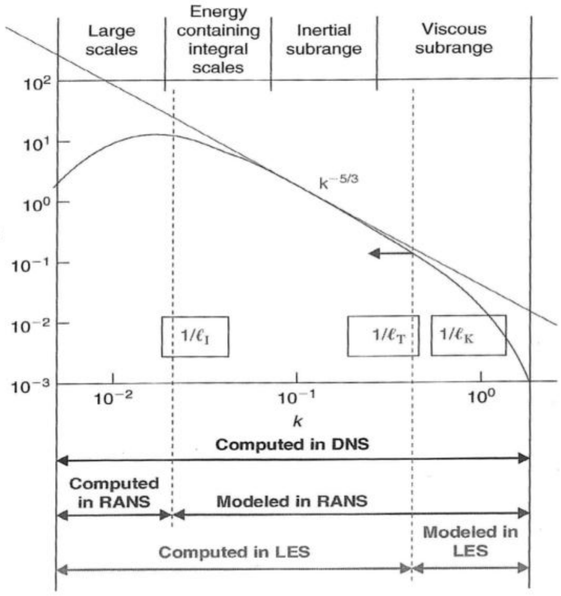


Figure 2.2: Energy cascade.

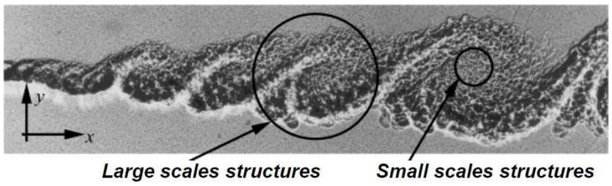


Figure 2.3: Eddy break up.

- Direct Numerical Simulation (**DNS**): simulates the whole range of the turbulent statistical fluctuations at all relevant physical scales. The size of the smallest turbulent eddies is inversely proportional to $Re^3/4$. This technique requires a notable computational effort, assuming a resolution of n points per direction per unit length, the number of operations to be computed is proportional to $n^3 * Re^9/4$. To solve even the smallest turbulent velocity fluctuations, the spatial grid needs to be extremely fine and time steps small enough to resolve the fastest fluctuations.

- Large Eddy Simulation (**LES**): it is similar to DNS approach because it computes directly the turbulent fluctuations in space and time but above a certain length scale - below that scale turbulence is modelled by semi-empirical laws. Also for this technique, computational effort remains still high.

- Reynolds-averaged Navier-Stokes (**RANS**) equations: this method involves time averaging of Navier-Stokes equations. From the averaging procedure some extra terms appear, which are the mean of the product between fluctuations component. In particular, in the momentum equation appears a second order tensor, the so-called Reynolds stress tensor that, according to Boussinesq, is proportional to mean deformation rates:

  $$\overline{\overline{r}} = -\rho \overline{\overrightarrow{u} \otimes \overrightarrow{u}} \tag{2.26}$$

  where $\overrightarrow{u}$ is the velocity fluctuation. Writing Eq. (2.26) for the *i-j* component of the fluctuation:

  $$r_{i,j} = -\rho \overline{u_i u_j} = \mu_t \left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right) - 2\rho k \delta_{ij} \tag{2.27}$$

  μ_t is the turbulent viscosity and must be determined. Purpose of turbulence model is to estimate the value of μ_t, and closure equations are required.

  $$\mu_t = \rho C_\mu \frac{k^2}{\varepsilon} \tag{2.28}$$

  where C_μ is a dimensionless constant.

In ICE application, RANS solvers are commonly used, and the most widely turbulence model applied is k-ε, RNG k-ε and k-ω SST models. In this study three different turbulence models are tested and presented:

- *k-ε*,
- *RNG k-ε*,
- *k-ω SST*.

In the next sections, these models are presented showing the governing equations and the modelling constant adopted.

2.4.1 k-ε model

It is a two-equation model, solving a transport equation for k and one for ε. The introduced unknown μ_t is estimated according to Eq. (2.28), where k is the turbulent kinetic energy and ε is rate of dissipation of turbulent kinetic energy. The mentioned transport equations for k and ε are:

$$\frac{\partial \rho k}{\partial t} + \overrightarrow{\nabla} \cdot \left(\rho k \overrightarrow{U} \right) = \overrightarrow{\nabla} \cdot \left(\frac{\mu_t}{\sigma_k} \overrightarrow{\nabla} k \right) + \overline{\overline{r}} : \overrightarrow{\nabla} \overrightarrow{U} - \rho \varepsilon \tag{2.29a}$$

$$\frac{\partial \rho \varepsilon}{\partial t} + \vec{\nabla} \cdot \left(\rho \varepsilon \vec{U} \right) = \vec{\nabla} \cdot \left(\frac{\mu_t}{\sigma_\varepsilon} \vec{\nabla} \varepsilon \right) + C_{1\varepsilon} \frac{\varepsilon}{k} \overline{\overline{r}} : \vec{\nabla} \vec{U} - C_{2\varepsilon} \rho \frac{\varepsilon^2}{k} \tag{2.29b}$$

In both equations, the first term on the left hand side is the rate of change of the variable (k or ε) and the second is the transport of the variable by convection, while on the right hand side the terms $\left(\vec{\nabla} \cdot \left(\frac{\mu_t}{\sigma_k} \vec{\nabla} k \right)\right.$ and $\left(\vec{\nabla} \cdot \left(\frac{\mu_t}{\sigma_\varepsilon} \vec{\nabla} \varepsilon \right)\right.$ represent the transport by diffusion of the variable k and ε, respectively, the rate of production $\left(\overline{\overline{r}} : \vec{\nabla} \vec{U} \right)$ for k and $\left(C_{1\varepsilon} \frac{\varepsilon}{k} \overline{\overline{r}} : \vec{\nabla} \vec{U} \right)$ for ε and finally the rate of destruction of the variable k $\left(\rho \varepsilon \right)$, and ε $\left(C_{2\varepsilon} \rho \frac{\varepsilon^2}{k} \right)$. The ε-equation is problematic in the near-wall region since the term ε^2/k is singular at the wall. Equations (2.29a) and (2.29b) contain five adjustable constant parameters evaluated by comprehensive data fitting for a wide range of turbulent flows:

$$C_\mu = 0.09 \tag{2.30a}$$

$$\sigma_k = 1.00 \tag{2.30b}$$

$$\sigma_\varepsilon = 1.30 \tag{2.30c}$$

$$C_{1\varepsilon} = 1.44 \tag{2.30d}$$

$$C_{2\varepsilon} = 1.92 \tag{2.30e}$$

Despite its advantages, the model shows a reduction of accuracy when describing flows containing large adverse pressure gradients. The k-ε model also poorly performs in a variety of important cases such as unconfined flows, curved boundary layers, rotating flows and flows in non-circular ducts.

2.4.2 RNG k-ε model

The RNG model was developed using Re-Normalisation Group (RNG) methods by [34] to renormalize the Navier-Stokes equations, in order to account for the effects of smaller scales of motion. In the standard k-ε model, the eddy viscosity is defined by a single turbulence length scale, so the calculated turbulent diffusion is the one occurring only at the specified scale, whereas in reality all scales of motion will contribute to the turbulent diffusion. The RNG approach, which is a mathematical technique that can be used to derive a turbulence model like the k-ε, results in a modified form of the ε equation, which attempts to account for the different scales of motion through changes to the production term.

$$\frac{\partial \rho \varepsilon}{\partial t} + \vec{\nabla} \cdot \left(\rho \varepsilon \vec{U} \right) = \vec{\nabla} \cdot \left[\left(\mu + \frac{\mu_t}{\sigma_\varepsilon} \right) \vec{\nabla} \varepsilon \right] + C_{1\varepsilon} \frac{\varepsilon}{k} P_k - C^*_{2\varepsilon} \rho \frac{\varepsilon^2}{k} \tag{2.31}$$

Turbulent viscosity being calculated in the same manner as with the standard k-ε model. In Eq. (2.31):

$$P_k = \overline{\overline{r}} : \overrightarrow{\nabla}\overrightarrow{\overline{U}} \tag{2.32a}$$

$$C^*_{2\varepsilon} = C_{2\varepsilon} + \frac{C_\mu \eta^3 (1 - \eta/\eta_0)}{1 + \beta\eta^3} \tag{2.32b}$$

$$\eta = S\frac{k}{\varepsilon} \tag{2.32c}$$

$$S = (2S_{ij}S_{ij})^{1/2} \tag{2.32d}$$

Where S is the modulus of the mean rate-of-strain tensor. Different constants' values are:

$$C_\mu = 0.0845 \tag{2.33a}$$

$$\sigma_k = 0.71942 \tag{2.33b}$$

$$\sigma_\varepsilon = 0.71942 \tag{2.33c}$$

$$C_{1\varepsilon} = 1.42 \tag{2.33d}$$

$$C_{2\varepsilon} = 1.68 \tag{2.33e}$$

$$\eta_0 = 4.38 \tag{2.33f}$$

$$\beta = 0.012; \tag{2.33g}$$

2.4.3 k-ω SST model

The *k*-ω *SST* model is a combination of two models: the *k*-ε and the *k*-ω turbulence models. Menter (1994) proposed a modified version of the *k*-ω model, i.e. the Shear Stress Transport (SST). This modified version combines the advantages of the two main models, thus minimising their weakness. It combines the ε and the ω equations, thanks to an extra term that, if null, makes the equation identical to the one for ω, while, if entirely computed, makes the equation identical to the one for ε.

The governing equations for k and ω are:

$$\frac{\partial \rho k}{\partial t} + \overrightarrow{\nabla} \cdot \left(\rho k \overrightarrow{\overline{U}}\right) = \overrightarrow{\nabla} \cdot \left(\left(\mu + \frac{\mu_t}{\sigma_k}\right)\overrightarrow{\nabla} k\right) + \overline{\overline{r}} : \overrightarrow{\nabla}\,\overrightarrow{\overline{U}} - \beta^* \rho\omega k \tag{2.34a}$$

$$\frac{\partial \rho \omega}{\partial t} + \overrightarrow{\nabla} \cdot \left(\rho \omega \overrightarrow{\overline{U}}\right) = \overrightarrow{\nabla} \cdot \left(\left(\mu + \frac{\mu_t}{\sigma_\omega}\right)\overrightarrow{\nabla} \omega\right) + \frac{\gamma\rho}{\mu_t}\overline{\overline{r}} : \overrightarrow{\nabla}\,\overrightarrow{\overline{U}} - \beta\rho\omega^2 + (1 - F_1)\frac{2\overrightarrow{\nabla}\omega\overrightarrow{\nabla}k}{\sigma_\omega \omega} \tag{2.34b}$$

In which the closure coefficient and auxiliary relations are:

$$\mu_t = \frac{\rho a_1 k}{max(a_1\omega; F_2\overrightarrow{\nabla}U)} \tag{2.35a}$$

$$CD_{k\omega} = max\left(\frac{2\vec{\nabla}\omega\vec{\nabla}k}{\sigma_{\omega 1}\omega}, 10^{-20}\right) \tag{2.35b}$$

$$F_1 = \tanh\left(min\left(max\left(\frac{k^{1/2}}{\beta^*\omega y}; \frac{500\mu}{\rho y^2\omega}\right); \frac{4\rho\sigma_{\omega 1}k}{CD_{k\omega}y^2}\right)\right)^4 \tag{2.35c}$$

$$F_2 = \tanh\left(max\left(2(\frac{k^{1/2}}{\beta^*\omega y}; \frac{500\mu}{\rho y^2\omega}\right)\right)^2 \tag{2.35d}$$

The constant values in Eqs. (2.34a), (2.34b) are:

$$\beta^* = 0.09 \tag{2.36a}$$

$$\beta_1 = 0.075 \tag{2.36b}$$

$$\sigma_k = 0.85 \tag{2.36c}$$

$$\sigma_\omega = 0.5 \tag{2.36d}$$

$$a_1 = 0.31 \tag{2.36e}$$

$$\gamma = \frac{\beta_1}{\beta^*} - \sigma_{\omega 1}k^2/\sqrt{\beta^*} \tag{2.36f}$$

In Eq. (2.34b), a blending factor *1-F_1* smoothly shifts the dissipation equation from Wilcox ω to ε when moving away from the wall. In this way, the model exploits the ability of Wilcox *k*-ω model to correctly predict the near wall region, while working like a standard *k*-ε model in case of free stream conditions.

2.5 Ignition and combustion models

More attention should be given to initial combustion development, which is a complex process affected by ignition system properties, local flow, turbulence, fuel type and equivalence ratio. Furthermore, different combustion regimes might be expected near the spark, depending on engine load and speed. Because of its major influence on engine efficiency and cyclic variability, the initial flame development process in spark ignition engines has been extensively studied over the years, to clarify the relevant governing parameters and to provide general guidelines for numerical model development [14, 13]. In this section, we will introduce the combustion process in Spark Ignition (SI) engines, explaining the fundamentals of combustion focusing on turbulent premixed combustion, then an overview of the numerical model adopted to simulate the ignition process, and the following turbulent flame development is presented.

2.5.1 Introduction

Combustion is a thermo-chemical process in which fuel and oxidiser react to produce a significant amount of heat. This process is affected by:

- reactant's physical-chemical properties,
- initial reacting mixing conditions,
- flow conditions,
- initial thermodynamic conditions,
- geometrical constraints.

Turbulence plays a crucial role in this context, and it has a dominant effect on the burning rate. Combustion in SI engines involves a very complex sequence of elementary reactions that oxidize fuel. Most of them depend on mixture temperature T and, to a lesser extent, on reacting concentrations. In first approximation the velocity of the overall oxidation process could be expressed by an Arrhenius law:

$$w_c = Cp^n e^{(-\frac{E_a}{RT})} \tag{2.37}$$

Where R is the universal gas constant and C, n and E_a (activation energy) are constants depending on the properties of fuel/air mixture). Depending on the flow condition, two distinct types of combustion could be distinguished:

- Laminar combustion: flame propagation requires thermal conduction and thermal diffusion of species.
- Turbulent combustion: molecular processes are augmented by turbulent mixing and flame distortion. This type of combustion is primarily a fluid dynamic issue, except for mixtures near flammability limit, ignition or flame quenching.

In IC engines the flow is always turbulent inside the cylinder, and its characteristics influence the flame propagation: in particular, the eddies generated are larger than the laminar flame thickness. Under these conditions, the flame is said to display a structure known as **wrinkled laminar flame**. The turbulent burning velocity depends on the turbulent intensity u' and can be up to 30 times higher that the laminar burning velocity. A flame is spark ignited in a flammable mixture only if the spark energy is greater than the minimum ignition energy E_{ign}; it is found experimentally that the ignition energy is inversely proportional to the square of the mixture pressure:

$$E_{ign} \propto \frac{1}{P^2} \tag{2.38}$$

In spark ignition engines the combustion is started, close to the end of the compression stroke, by the electrical discharge produced between the spark plug electrodes by the ignition system. Two types of charge could be burnt in a SI engine:

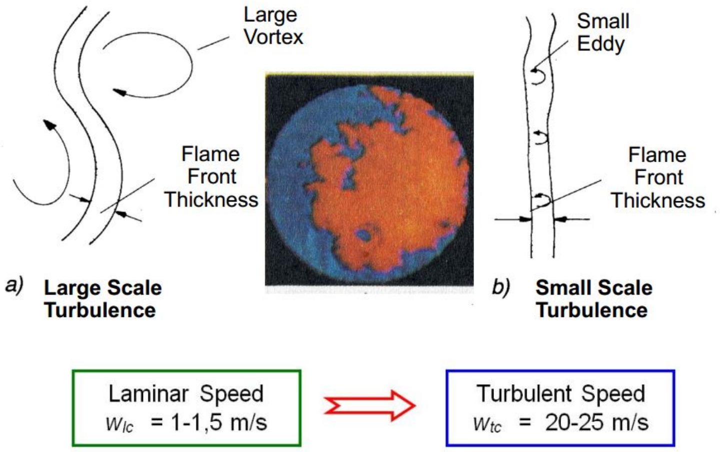


Figure 2.4: Effects of large and small turbulence vortexes on flame front.

- Homogeneous charge,
- Stratified charge.

In this study, a homogeneous mixture of air and propane is ignited in the side chamber of the test engine, so some information about this combustion type is given below.

2.5.2 Homogeneous charge combustion

In conventional spark ignition engine, the fuel is completely vaporised and homogeneously premixed with air and eventually with residual gas, before entering the cylinder. If the combustion is normal, the mixture is ignited by the spark plug in an optimised moment of the engine cycle, toward the end of the compression stroke. Normal combustion occurs when:

- the ignition of the mixture is controlled by the spark plug, with a pre-set timing,
- after the ignition the flame regularly propagates to the whole mixture, without any sudden increase in its velocity.

A normal combustion process could be subdivided into three different phases:

1. Initial flame development: a small ignited volume of mixture is gradually transformed into a developed front of a turbulent flame.

2. Turbulent flame propagation: the flame rapidly spread over the main part of the combustion chamber.

3. Burnout: the mixture completes its oxidation process behind the flame front.

During the first period, the combustion speed is mainly determined by the physical and chemical properties of the combustible mixture, which influence the reactions kinetics. This phase begins with the ignition spark and ends when the pressure line of the fired cycle (Figure 2.5) departs from the compression line of the motored cycle. The main phase is the second and it is characterized by a rapid burning of the mixture. In this phase, a completely turbulent flame, greatly influenced by the turbulent motions, is spreading through the combustion chamber with an increasing speed, until it approaches the walls. Conventionally, this second phase begins at the separation of the compression, and combustion line (Figure 2.5) in the p - θ diagram and it ends when the flame front comes close to the walls, and the maximum pressure is reached. At this stage, the combustion is not yet completed, because afterburning goes on near the walls and behind the front until the oxidation process is completed. The rate of this last combustion completion phase is lower than the previous one, and it depends more on the physical and chemical properties of the mixture than on the turbulence intensity.

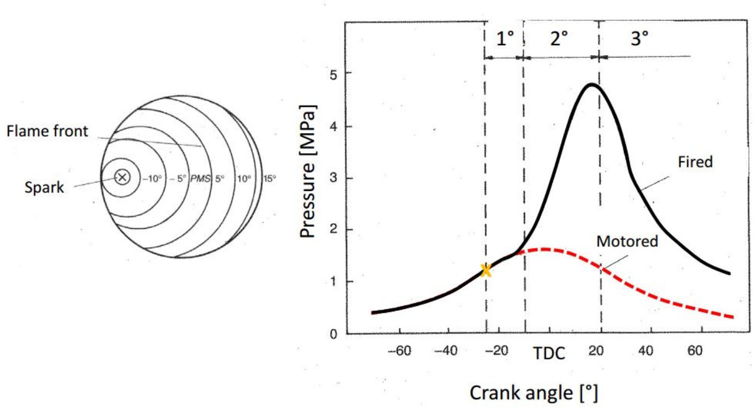


Figure 2.5: Schematic of the combustion phases in a SI engine.

In Figure 2.5 these three phases are showed graphically in a p - θ diagram, and it is possible to observe also the effect that the combustion has on the pressure inside the cylinder (fired cycle) compared to the motored cycle in which the pressure increase is only due to the compression stroke.

2.5.3 Numerical models

In this study, suitable models are adopted for simulating the combustion process in a spark ignition engine with a homogeneous premixed charge of air and propane under different rotational speeds and for different mixture composition. In particular, the combustion modelling involved two different sub models for the ignition phase and the following turbulent flame development. The ignition model adopted is based on the AKTIM model (Arc and Kernel Tracking Ignition Model [27]) while the turbulent combustion propagation is based on a *Coherent Flamelet Model* (CFM) according to Choi and Huh [8] and modifications to the stretch of the flame proposed by Sforza et al. [30]. In the following sections, the equations supporting the models adopted are presented.

2.5.4 Ignition model

In this section, the fundamentals of the ignition model adopted in this study for the combustion simulations is described. As mentioned in the introduction, in a spark ignition engine an electrical spark generated with electrodes initiates the combustion. This phenomenon could be modelled by imposing at the ignition time t$_{ign}$ an initial profile of flame surface density (FSD). Observing this initial stage of combustion, the diameter of the initial sphere of burned gases is very small - it is in the order of the spark gap distance. Consequently, such small flame front cannot be resolved on the mesh and the total reaction rate should be correctly estimated as long as the volume occupied by the burned gases is too small to be resolved. For this purpose, the AKTIM model is adopted in this study according to what proposed by Richard et al. [27]. The difference between what proposed by Richard et al. [27] and the AKTIM model is that no lagrangian tracking is used for describing the flame kernels. The initial flame kernel is tracked by the Eulerian progress variable c. The ignition model combines three phases:

1. **Imposition of a spherical profile of the progress variable:** the initial profile of the progress variable $\tilde{c}$ imposed at the spark timing is:

$$\tilde{c}(x, t_{ign}) = \frac{c_0}{2}(1 - \tanh\left(\frac{|x - s_{spk}|}{\delta_{ign}}\right) \tag{2.39}$$

   where δ_{ign} is the mean value of mesh spacing near the spark plug, x$_{spk}$ is the position of the spark plug electrodes and c$_0$ is an arbitrary small parameter.

2. **Definition of the reaction rate:** after t$_{ign}$ the reaction rate (or FSD) has to be defined for the progress variable, since the flame front is not fully established an ignition FSD model called Σ_{ign} is reconstructed. As in the AKTIM model, it is assumed that the mean flame kernel surface S$_{mean}$ has

roughly a spherical shape during the first combustion stage described by the ignition model. S$_{mean}$ is related to the progress variable by the following relation:

$$V^b(t) = \int_\Omega \overline{c}(x,t)\mathrm{d}V = \frac{4}{3}\pi(r^b(t))^3 \tag{2.40a}$$

$$S_{mean}(t) = 4\pi(r^b(t))^2 \tag{2.40b}$$

where V^b is the total volume occupied by burned gases on the computational domain and r^b is the radius of the sphere of volume V^b. In the first stage of combustion it could be assumed that the wrinkling of the reacting front remains small and the total flame surface is given by:

$$S_{tot} = \Xi_{spk}(t)S_{mean}(t) \tag{2.41}$$

where Ξ_{spk} is the mean wrinkling factor over this surface. Initially, since the flame is laminar Ξ_{spk}=1, then increases due to the turbulent stretching of the flame. A model evolution equation for Ξ_{spk} is given by:

$$\frac{\mathrm{d}\Xi_{spk}}{\mathrm{d}t} = \langle a_t\rangle_s \left\langle \frac{\Xi_{eq} - \Xi_{spk}}{\Xi_{eq} - 1} \right\rangle_s \Xi_{spk} \tag{2.42}$$

Equations 2.40 and 2.42 allow to calculate the total flame surface S$_{tot}$ at each instant. As this surface needs to be spread spatially over the computational grid, the FSD is expressed by:

$$\Sigma_{ign}(x,t) = \alpha\tilde{c}(x,t)(1 - \tilde{c}(x,t)) \tag{2.43}$$

Where α is a global coefficient which is defined by imposing the integral of Σ_{ign}(x,t) over the domain to be equal to the total flame surface S$_{tot}$:

$$\alpha = \frac{S_{tot}}{\int_\Omega \tilde{c}(x,t)(1 - \tilde{c}(x,t))\mathrm{d}V} \tag{2.44}$$

3. **Transfer of the FSD$_{ign}$ to the FSD transport equation:** the reaction rate is defined by Σ_{ign} until the progress variable reaches the value of 1 somewhere in the domain, at this instant called $t_{transition}$ the flame surface density defined by Σ_{ign} is transferred to the FSD transport equation $\overline{\Sigma}_{\overline{c}}$ given by:

$$\partial\overline{\Sigma}_{\overline{c}}t + T_{res} + T_{sgs} = S_{res} + S_{sgs} + C_{res} + C_{sgs} + P \tag{2.45}$$

where T$_{res}$, S$_{res}$, C$_{res}$ and P are, respectively, the transport, strain, curvature and propagation terms due to resolved flow motions, and T$_{sgs}$, S$_{sgs}$ and C$_{sgs}$ are, the unresolved transport, strain and curvature terms, respectively.

2.5.5 Flame Surface Density Model

The turbulent combustion analysis could be performed by several models: the eddy breakup model (EBU), the assumed probability density function (PDF) and the coherent flamelet model(CFM). In this work the CFM model is adopted according to what proposed by Choi and Huh [8] with some modifications regarding the flame stretch proposed by Sforza et al. [30]. The CFM views the turbulent flame as a collection of flamelet elements embedded in a turbulent flow. The CFM applies to both premixed and non-premixed flame based on the laminar flamelet concept. Its major advantage lies in the decoupled treatment of the chemical reaction and turbulent flow. The laminar flamelet regime exists for $Da > 1$ and $K < 1$, where Da is the Damköhler number and K is the Karlovitz number. The Damköhler number is defined as the ratio of the integral time scale to the flame time scale as:

$$Da = \frac{\tau_t}{\tau_F} \tag{2.46a}$$

$$\tau_t = \frac{l_t}{u^{‘}} \tag{2.46b}$$

$$\tau_F = \frac{\delta_L}{U_L} \tag{2.46c}$$

where l$_t$ is the integral length scale, u$^{‘}$ is the turbulence intensity, δ_L is the laminar flame thickness and U$_L$ is the unstrained laminar burning velocity. The Karlovitz number is defined as:

$$K = \tau_F \left(\frac{1}{A} \frac{\mathrm{d}A}{\mathrm{d}t} \right) \tag{2.47}$$

where A is the flamelet surface area. The terms within brackets in Eq. (2.47) represents the flame stretch, that will be explained in the following section. The mass burning rate per unit volume in the unstretched laminar flamelet regime is simply given by:

$$\dot{w} = \rho_u U_L \Sigma \tag{2.48}$$

Where ρ_u is the fresh gas density, and Σ is the flame surface density. Chemical and molecular effects are represented by U$_L$, while turbulence effects are represented by Σ. A conservation equation for the flame surface density Σ is:

$$\frac{\partial \Sigma}{\partial t} + \frac{\partial}{\partial x_i}(U_i \Sigma) = \frac{\partial}{\partial x_i}\left(\frac{\nu_t}{\sigma_\Sigma} \frac{\partial \Sigma}{\partial x_i} \right) + P_{FSD} - D_{FSD} \tag{2.49}$$

where P_{FSD} is the production of flame surface by turbulent rate of strain and D_{FSD} is the annihilation of flame surface by mutual collision, ν_t is the turbulent kinematic viscosity and σ_Σ is the turbulent Schmidt number ($\sigma_\Sigma = 1$). Production and annihilation terms are modelled by several models. In this study the model adopted is computed as follows [20, 30, 8]:

$$P_{FSD} = \alpha_{FSD} \frac{u^{`}}{l_{tc}} \tag{2.50a}$$

$$D_{FSD} = \beta_{FSD} s_{u0} \frac{\Sigma^2}{c(1-c)} \tag{2.50b}$$

The transient curvature and flame stretch effect of eddies of different length scales on flame surface generation are considered by the Intermittent Turbulence Net Flame Stretch (ITNFS) model, while the total effect is assumed to be equal to the sum of each individual scale neglecting any nonlinear interaction between different scales [8]. In summary, the model adopted solves two transport equations:

1. the combustion normalized progress variable c,

2. the flame surface density Σ.

Furthermore, it should be noted that the ignition model and the CFM model work separately Initially, only the ignition model (AKTIM) is active, then, when the progress variable c reaches a threshold value (0.98) this model is switched off, and the CFM model is activated for proceeding the combustion process. For coupling these two models, a variable that in this work is called α_{AKTIM} is used to calibrate the flame surface in the ignition phase. This parameter multiplies the wrinkling factor Ξ in Eq. (2.41) and permits to couple the two models.

2.5.6 Flame stretch model

The Gülder correlation is used in this model to compute the unstretched laminar flame velocity s_{u0} as a function of fuel type, local equivalence ratio, temperature and pressure. This correlation does not consider 2 effects:

1. Curvature effects due to the flame front geometry.

2. Strain effects due to the flame front flow field interaction.

Both these effects are considered in the so-called flame stretch process and are taken into account by the stretch factor I_0, defined as:

$$I_0 = \frac{s_u}{s_{u0}} \tag{2.51}$$

This process is modelled following the approach proposed in [35], the actual laminar flame speed s_u was computed according to Bradley et al. [6]:

$$s_u = s_{u0} \frac{1 - KMa}{1 + \frac{c}{R}\delta_l Ma} \tag{2.52}$$

where δ_l is the laminar flame thickness and c=2 is the parameter to fit Eq. 2.52 for spherical flame shapes, since the initial flame kernel is supposed to be spherical. The Karlovitz stretch factor K was computed as in [30]:

$$K = 0.157\left(\frac{u^{‘}}{s_{u0}}\right)^2 R_L^{-0.5} \tag{2.53a}$$

$$R_L = \frac{u^{‘}L_i}{\nu} \tag{2.53b}$$

Here λ is the Taylor microscale, R$_L$ is the turbulent Reynolds number, ν is the kinematic viscosity and L$_i$ is the integral length scale of turbulence. The parameter $1/R$ in Eq. (2.52), known as the local flame curvature is defined by the following relation:

$$\frac{1}{R} = \left|\nabla\cdot\left(\frac{\overrightarrow{\nabla}c}{\|\overrightarrow{\nabla}c\|}\right)\right| \tag{2.54}$$

where $\overrightarrow{\nabla}c/\parallel\overrightarrow{\nabla}c\parallel$ is the flame front perpendicular direction. The Markstein number Ma was evaluated into fully burnt products according to [30]:

$$Ma = \left[\frac{1}{\gamma}\ln\left(\frac{1}{1-\gamma} + \frac{1}{2}Ze(le-1)\frac{1-\gamma}{\gamma}\int_0^{\frac{\gamma}{1-\gamma}}\frac{\ln(1+x)}{x}\mathrm{d}x\right] - \ln\left(\frac{1}{1-\gamma}\right) \tag{2.55}$$

where γ is the expansion ratio defined as $\gamma = (\rho_u - \rho_b)/\rho_u$, being ρ_u and ρ_b the burned and unburned density respectively, x is a dummy variable of integration, and Ze is the Zeldovich number computed as [30]:

$$Ze = \frac{T_{act}(T_{ad} - T_u)}{t_{ad}^2} \tag{2.56}$$

where T_{act} is the activation temperature, Le is the Lewis number and T_{ad} is the adiabatic flame temperature. Both the Lewis number and the activation temperature are evaluated by suitable models explained in the next section.

2.5.7 Lewis number model

Considering the premixed combustion phenomenon, several instability modes could affect the flame front. One of these is the thermo-diffusive instability, which is controlled by the relative importance of the heat diffusion in relation to the deficient reactant one, i.e. by the Lewis number, defined as the ratio of those magnitude. In the presence of stoichiometric mixture, where the deficient reactant cannot be clearly defined, some trouble could occur, but as mentioned in [30] and based on experiment of Joulin and Mitani [15],it was observed that the flame behaviour gradually changes as the equivalence ratio varies across the stoichiometric condition. This means that the combustion process is controlled not only by the deficient component of the mixture but also by the abundant one; consequently, also the effect on the stability of this last component should be considered. Under these considerations, the Lewis number of a premixed mixture can be considered a weighted average of the Lewis number values computed only in relation to the deficient and abundant reactant. The Lewis number is defined according to [30] as:

$$Le = Le_{def} + H(Le_{abu} - Le_{def}) \tag{2.57}$$

In which Le_{def} and Le_{abu} are the Lewis numbers defined with respect to the deficient and abundant reactant diffusivity, respectively. The parameter H is defined as:

$$H = n\frac{G(m, n-1, A)}{G(m, n, A)} \tag{2.58a}$$

$$G(m, n, A) = \int_0^\infty x^m (x + A)^n \exp^{-x} \mathrm{d}x \tag{2.58b}$$

Eqs. (2.58a) and (2.58b) are both function of m and n which are the order of reaction of the deficient and abundant reactant. The value of A is evaluated as:

$$A = Ze\frac{y}{Le_{abu}} \tag{2.59}$$

where y is:

$$y = \phi - 1 \qquad \text{if } \phi > 1 \tag{2.60a}$$

$$y = \frac{1}{\phi} - 1 \qquad \text{if } \phi < 1 \tag{2.60b}$$

2.5.8 Activation temperature model

The activation temperature of a premixed mixture was estimated by using the model proposed in [30]:

$$\ln s_{u0} = c - \frac{T_{act}}{2}\frac{1}{\overline{T}_f} \tag{2.61}$$

where $\overline{T}_f$ is the average flame temperature defined as:

$$\overline{T}_f = T_u + 0.74(T_{ad} - T_u) \tag{2.62}$$

Values of T_{act} were computed before running any numerical simulation and written into a look-up table as function of pressure (p), equivalence ratio (ϕ) and exhaust gas recirculation (EGR).

2.6 Discretization

In computational fluid dynamics modelling, the discretization of the equations is implemented following the *Finite Volume Method* [10]. This method uses the integral form of the conservation equations as its starting point. The solution domain is subdivided into a finite number of contiguous control volumes (CVs), and the conservation equations are applied to each CV. At the centroid of each CV lies a computational node at which the variable values are to be calculated. Interpolation is used to express variable values at the CV surface regarding the nodal (CV-centre) values. The FV method can accommodate any type of grid, so it is suitable for complex geometries. The grid defines only the control volume boundaries and does not need to be related to a coordinate system.
The first step is the discretisation of the computational domain into non-overlapping sub-domains: this means generating a suitable grid or mesh of cells (i.e. the control volumes) as the one shown in Figure 2.6. The physical variables are calculated and stored in the control volume centroid (node **P**) and not at their boundary faces, thus allowing to easily implement the boundary conditions. In Figure 2.6 **N** is the centroid of the neighbouring cell, and every neighbouring cell shares a face with the considered control volume.

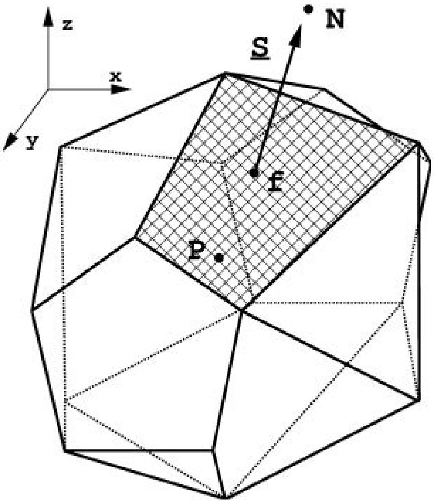


Figure 2.6: Control volume.

The centroid of the cell $\vec{x}_P$ is defined as follows:

$$\int_V (\vec{x} - \vec{x}_P)\mathrm{d}V = 0 \tag{2.63}$$

Considering a general property ϕ, a linear variation of that property within each control volume is assumed:

$$\phi = \phi_P + (\vec{x} - \vec{x}_P) \cdot \left(\vec{\nabla}\phi\right)_P + O\left((\vec{x} - \vec{x}_P)^2\right) \tag{2.64}$$

Moving from the centroid to the cell's face, it is possible to define the face center x_f as:

$$\int_S (\vec{x} - \vec{x}_f)\mathrm{d}S = 0 \tag{2.65}$$

And linear variation of ϕ is assumed within each internal face:

$$\phi(\vec{x}) = \phi_f + (\vec{x} - \vec{x}_f) \cdot \left(\vec{\nabla}\phi\right)_f + O\left((\vec{x} - \vec{x}_f)^2\right) \tag{2.66}$$

The following step consists in discretizing the partial differential equation into algebraic ones. This procedure is carried out by integrating the differential equations over each discrete element and finally iteratively solved. A peculiarity of the FV method is that it is intrinsically conservative As proof of this, the flux entering a given cell is identical to that leaving the adjacent volume, this property makes this method perfectly suitable for CFD applications.

2.6.1 Approximation of surface integrals

The net flux through the CV boundary is the sum of integrals over the four (in 2D) or six (in 3D) CV faces:

$$\int_S f\mathrm{d}S = \sum_k \int_{S_k} f\mathrm{d}S \tag{2.67}$$

where f is the component of the convective ($\rho\phi v \cdot n$) or diffusive ($\Gamma grad\phi \cdot n$) flux vector in the direction normal to CV face. To calculate the surface integral in Eq. (2.67) exactly, we would need to know the integrand f everywhere on a surface cell S. This information is not available, as only the nodal (CV centre) values of ϕ are calculated, so an approximation must be introduced. This is best done using two levels of approximation:

- the integral is approximated in terms of variable values at one or more locations on the cell face;
- the cell-face values are approximated in terms of nodal (CV centre) values.

The simplest approximation of surface integral is the *midpoint rule*, which consists in approximating the integral as a product of the integrand at the cell-face centre and the cell-face area. This approximation is second-order accurate: to increase the accuracy order more than two locations are required to evaluate the value of *f*.

2.6.2 Approximation of volume integrals

Some terms in the transport equations require the integration over the volume of a CV. The simplest second-order accurate approximation is to replace the volume integral with the product of the mean value of the integrand and the CV volume and approximate the former as the value at the CV centre:

$$Q_p = \int_\Omega q \mathrm{d}\Omega = \overline{q}\Delta\Omega \approx q_P \Delta\Omega \tag{2.68}$$

where q_P stands for the value of q at the cell centre. This quantity is easily calculated; since all variables are available at node **P**, no interpolation is necessary. The above approximation becomes exact if q is either constant or linearly varies within the CV; otherwise, it contains a second-order error. An approximation of higher order requires the values of q at more locations than just the centre. These values have to be obtained by interpolating nodal values or, equivalently, by using shape functions.

2.6.3 Interpolation and Differentiation Practices

The integrals approximation requires to know the value of variables at locations different from the cell centre. The integrand, denoted with *f*, involves the product of different variables at those locations. We assume that the velocity field and the fluid properties p and Γ are known at all locations. To calculate the convective and diffusive fluxes, the value of ϕ and its gradient normal to the cell face at one or more locations on the CV surface are needed. Volume integrals of the source terms may also require these values. They have to be expressed in terms of nodal values by interpolation. Different possibilities are available, the most widely used are:

- upwind interpolation: Upwind Differencing Scheme (UDS), approximating ϕ by its value at the node upstream.
- linear interpolation: Central Differencing Scheme (CDS), the value at control volume face centre is a linear interpolation between the two nearest nodes.
- quadratic upwind interpolation: Quadratic Upstream Interpolation for Convective Kinematics (QUICK), the value of ϕ between the cell centre and control volume face centre is approximated by a quadratic interpolation.

2.6.4 Time discretization

When unsteady cases are studied, also a time discretisation is required. The time domain is subdivided into time steps Δt. There are two different methods to discretise the time domain:

- explicit: the value of ϕ at time n and cell centre **P** (ϕ_P^n) depends on the value of ϕ at previous time and at neighbour cell **N** (ϕ_N^0);
- implicit: the value of ϕ at time n and cell centre **P** (ϕ_P^n) depends on the value of ϕ at same time n and at neighbour cell **N** (ϕ_N^n);

The implicit method is always stable (thus the time step can be larger), but computationally more expensive, while explicit method is only conditionally stable, thus meaning that to have stability the time step cannot be larger than a maximum value, which depends on the Courant number defined in Eq. (2.69).

$$Co = \frac{u}{\Delta x / \Delta t} \tag{2.69}$$

where u is the velocity magnitude, $\Delta\ x$ is the grid dimension and $\Delta\ t$ is the time step. To ensure stability the condition to be respected is that $Co < 1$. Among the implicit methods, the most used are **Euler** (accuracy of the first order) and **Crank Nicolson** (second order).

3. OpenFOAM and Lib-ICE

In this chapter we will offer a brief introduction of the code adopted for running all the simulations, then an overview of the main solvers used is presented.

OpenFOAM stands for Open Source Field Operation and Manipulation: it is a framework for developing application executables that uses packaged functionality included in a collection of approximately 100 C++ libraries [5]. The great advantages of this tool are that it is free and new solvers and utilities can be created by its users and compiled in addition to standard ones. An overview of OpenFOAM structure is shown in Figure 3.2. Thanks to the advantages mentioned before, the ICE group of Politecnico di Milano has developed a set of libraries, solvers and utilities grouped under Lib-ICE denomination: it is a code based on OpenFOAM environment focused on internal combustion engines simulations. This code provides modified version of the base OpenFOAM solver and utilities for mesh management, combustion, direct injection and other typical engine applications.

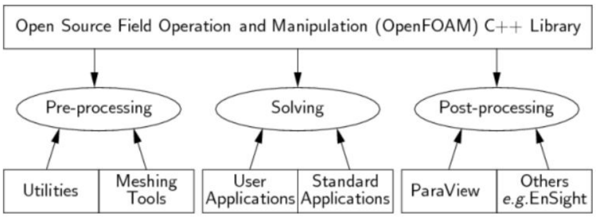


Figure 3.1: Overview of OpenFOAM structure.

3.1 Solvers

In the current work, different solvers are used. For the incompressible and steady state analysis performed on the Orleans vessel (see Section 5.3.2) the *simpleFoam* solver is used, while for the compressible and unsteady simulations performed on the test engine and on the Orleans vessel different solvers are used. In particular, for the Orleans vessel a modified version of *rhoPimpleFoam* has been used, while for the test engine the *pimpleColdEngineDyMFoam* for cold flow analysis and *pimpleCFMEngineDyMFoam* for reacting analysis were utilized. The steady state and incompressible solver *simpleFoam* uses a SIMPLE algorithm. SIMPLE is an acronym for Semi-Implicit Method for Pressure Linked Equations and the algorithm follows an iterative process explained here. Starting from the assigned boundary condition, the gradients of velocity and pressure are computed. Then the discretised momentum equation is solved to calculate the intermediate velocity field, so it is possible to compute the uncorrected mass fluxes at faces. The next step is to solve the pressure correction equation to produce cell values of the pressure correction, in this way the pressure fields are updated according to Eq. (3.1a) in which **urf** is the under-relaxation factor for pressure, then the pressure corrections are updated and the mass fluxes are rectified according to Eq. (3.1b). The cell velocities are corrected according to Eq. (3.1c) where $\nabla p^{'}$ is the gradient of the pressure corrections and $\overrightarrow{a}_p^v$ is the vector of central coefficients for the discretised linear system representing the velocity equation and Vol is the cell volume.

$$p^{k+1} = p^k + urf \cdot p^{'} \tag{3.1a}$$

$$\dot{m}_f^{k+1} = \dot{m}_f^* + \dot{m}_f^{'} \tag{3.1b}$$

$$\overrightarrow{v}^{k+1} = \overrightarrow{v}^* - \frac{Vol \nabla p^{'}}{\overrightarrow{a}_p^v} \tag{3.1c}$$

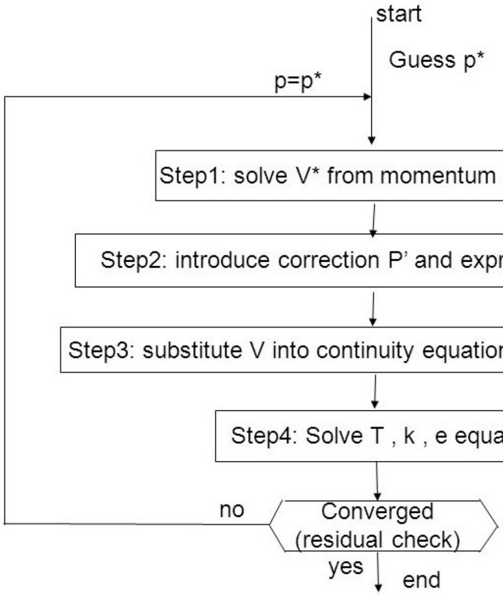

start
Guess p*
p=p*
Step1: solve V* from momentum equations
Step2: introduce correction P' and express V = V* + f(P')
Step3: substitute V into continuity equation solve P' and then V
Step4: Solve T , k , e equations
no
Converged (residual check)
yes
end

Figure 3.2: SIMPLE algorithm.

For the unsteady and compressible and, finally, for the combustion analysis the solvers adopted and mentioned before are based on the PIMPLE algorithm.

This method merges PISO and SIMPLE algorithms. PISO is the acronym for Pressure Implicit with Splitting of Operators, whereas the SIMPLE loop is the same explained before. The main differences from the SIMPLE algorithm are the following:

- No under-relaxation is applied.
- The momentum corrector step is performed more than once.

In particular, it solves for the momentum only once at any time step. The corrector loops are required to couple the velocity field with the pressure filed. Each corrector loop, after solving the pressure equation, updates the values of flux that are used in the following correction to solve the pressure equation once again. The number of correctors must be chosen in order not to excessively slow down the simulation but to have low initial residuals for the last correctors as well. In Figure 3.3 the PIMPLE cycle is shown. We can see that the outer iteration is the SIMPLE loop, whereas the inner one is the PISO.

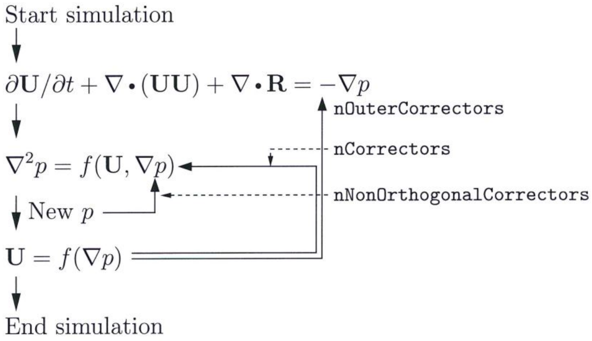

Start simulation
$\partial\mathbf{U}/\partial t + \nabla\bullet(\mathbf{UU}) + \nabla\bullet\mathbf{R} = -\nabla p$
nOuterCorrectors
nCorrectors
$\nabla^2 p = f(\mathbf{U}, \nabla p)$
nNonOrthogonalCorrectors
New p
$\mathbf{U} = f(\nabla p)$
End simulation

Figure 3.3: PIMPLE algorithm.

4. Test Engine

This chapter introduces and explains the characteristics and development of a numerical set-up of an experimental test engine with optical access [14].

First, the geometrical features of the engine are presented, then the generation of different meshes for different purpose (cold flow analysis and combustion simulation) is reported and, finally, the non-reacting and reacting analyses are performed comparing numerical with experimental results available from Herweg et al. [14] and Herweg and Maly [13].

4.1 Description of the experimental test engine

Figure 4.1 shows a scheme of the test engine. The cylinder head is linked to a side chamber by a pipe called "connecting throat" that has a diameter of 12 mm and it is tangential to the surface of the disc-shaped side chamber (diameter 45 mm, height 19 mm), in order to produce intense vortex motion during the compression stroke. It is possible to view the flame kernel region from two orthogonal directions. Optical access is provided by two large quartz windows with 17 mm in diameter. Additional ports for piezo-electric pressure transducer, conventional spark plug electrodes or ionization probes can be used. Two pressure transducers in the main and side chamber measured pressure traces during engine operation. A piezo-resistive transducer in the inlet port system defined the inlet pressure at inlet valve closing point. In the study performed in [14], thin wire electrodes (0.8 mm in diameter) were inserted in the side chamber via the quartz front windows to minimise flow field interference. Two different electrode positions were tested: a central position and a peripheral position 12.5 mm distant from the centre as indicated in Figure 4.1. The engine characteristics are summarized in Table 4.1. Two conventional ignition systems were applied: a transistorised coil ignition system (TCI) and a capacitor discharge system (CDI). Briefly, the TCI system delivers most of the energy during the glow discharge mode, lasting 0.6 to 1.2 ms depending on flow velocity level. During this time, current drops from 80 mA to zero and voltage amounts to over 400 V. The CDI discharge characteristics substantially differ from TCI system's, since most of the electrical energy is released within the arc discharge mode as short (0.1 to 0.2 ms), high current, low voltage pulse.

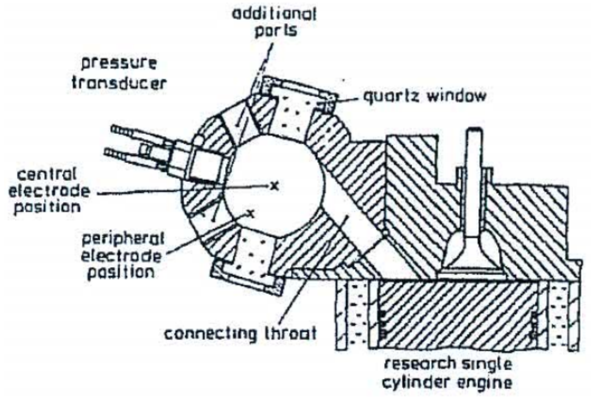

additional ports
pressure transducer
quartz window
central electrode position
peripheral electrode position
connecting throat
research single cylinder engine

Figure 4.1: Side chamber engine with central and peripheral ignition

Bore	73 mm
Stroke	67 mm
Connecting road length	112 mm
Clearance height	1 mm
Compression ratio	7.3
Diameter of side chamber	45 mm
Clearance of side chamber	19 mm
Diameter of connecting throat	12 mm
Thin wire electrodes diameter	0.8 mm
Spark gap	1 mm
Spark locations	central and peripheral (12.5 mm)
EGR	No
Engine speeds	300, 500, 750, 1000,1250
Normalized air-fuel ratio	1.0, 1.3, 1.5
Ignition timing (CAD)	-10
Intake Valve Closing IVC (Crank Angle [deg])	-168
Mixture inlet temperature	298 K
Pressure at ignition timing	5 bar
Fuel	Propane

Table 4.1: Operating characteristics of the side combustion engine

Accurate engine control and recorded data synchronisation was achieved by an electronic control unit shown in Figure 4.2. This device controls the data acquisition system, two rotating mirror cameras, spark flash light source, ignition timing and charge control. The unit was driven by input signals from a crank angle mounted shaft encoder delivering signals every half crank angle degree. An electric motor was used to control the engine speed. The engine was preheated and operated in a single-fired mode to eliminate residual gas. Therefore, a constant propane-air mixture was flowing through the combustion chamber, while motoring the engine at the determined engine speed. The air and propane flow rates were monitored by a Brooks mass flow meter and mass flow controller with an accuracy of 1%. Mixing of both was performed via a Venturi nozzle along with a special mixing tank reduces pumping effects in the inlet system.

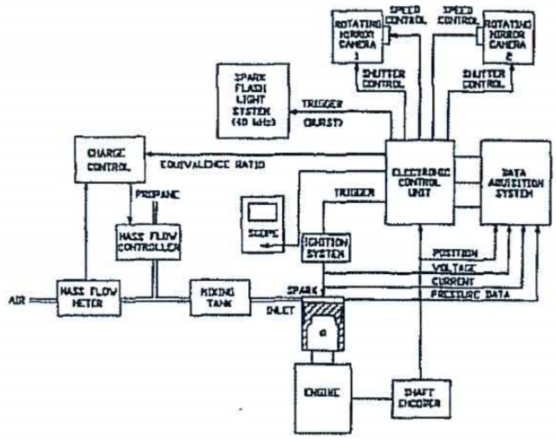


Figure 4.2: Electronic instrumentation

4.1.1 Measurement system

Figure 4.3 shows a schematic presentation of the high speed schlieren photography. A purpose-built spark flash light source is used to illuminate the combustion chamber from two orthogonal directions with a frequency of 40 kHz and a pulse duration of 20 ns. For each direction, a schlieren system built of condenser, schlieren lens, circular schlieren stop and camera optics is set up. Two rotating mirror cameras are used to get the pictures with a 20 mm negative size. Image processing is performed after negative processing to define the schlieren contour of every single frame. For measuring flow quantities, Laser Doppler Anemometer (LDA) is used at the peripheral spark location.

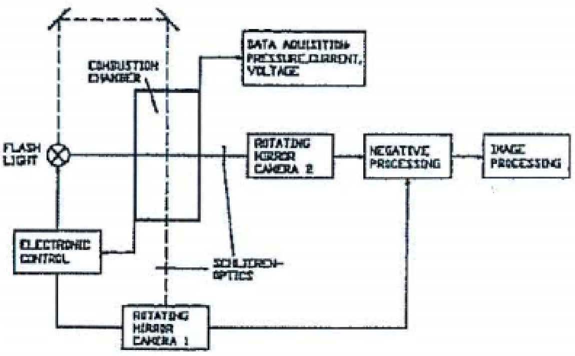


Figure 4.3: Schematic of high-speed schlieren photography

4.1.2 Flow prediction

In [14] also flow predictions were available and a multi dimensional method was used to perform flow predictions in the combustion chamber. It is a finite-volume procedure which solves the conservation equations of mass, momentum, energy and the chosen turbulence model (in [14] a suitably-adapted version of the k-ε model is used). The solution domain for the flow predictions study was constrained (limited) to the swirl chamber due to the non-availability of a grid-generator technique to generate a body-fitted orthogonal grid for the whole of the cylinder-connecting-throat-swirl-chamber configuration. Simulation starts at IVC (which occurred 12 crank angle degree after bottom dead centre of induction, and, according to this work, occurred at -168 crank angle degree) and covered the whole of the compression stroke (i.e. up to 0 crank angle degree, top dead centre). Since we had no information about the initial field, all velocity components were initialized to zero, and spatial uniform scalar fields were assumed.
Boundary conditions at the walls are managed by standard wall functions, while at the inlet of the swirling chamber, due to the lack of experimental data, a method that assumes the density homogeneous in the main cylinder and swirl chamber (justified if the throat Mach number is less than 0.3) is used to calculate the inlet mass flow rate.
The experimental and numerical set up explained is used as guideline to generate a suitable grid and implement a numerical model in OpenFOAM environment. In this study, a non-reacting, unsteady and compressible simulations campaign was implemented to validate the flow field inside the side chamber by comparing the numerical results with the experimental findings and flow predictions of [14].

4.2 Herweg and Maly test engine mesh

In this section, the generation of different types of mesh (structured and unstructured) for the test engine is described: first the generation of a mesh without spark plug electrodes used for finding the numerical set-up is presented, then the generation of 2 grids with central and peripheral electrodes is explained and, finally, the mesh adopted in the combustion analysis is shown.

After the presentation of the mesh generation process, the grids require the piston movement, so also this aspect is illustrated.

Once the geometry is reproduced by a suitable mesh, the non-reacting flow analysis and validation based on the experimental findings is performed and thanks to the availability of plenty of experimental findings regarding the combustion process in the side chamber, a reacting analysis was performed as well.

4.2.1 Structured mesh

Structured mesh is identified by regular connectivity. This type of grid is highly space-efficient since neighbourhood relationships are defined by storage arrangement. These considerations mean that simulations are faster compared to an unstructured grid. For computational time and power available, the grid adopted to validate the flow field in the side chamber engine is of structured type.

The mesh does not include the spark plug electrodes because usually when flow quantities are experimentally measured they are removed. To build this grid, a big effort was made to ensure connectivity between all blocks. For the sake of simplicity, the origin of the cartesian reference system is placed at the centre of the side chamber. This choice is justified by the fact that the ignition points are two:

- one at the centre of the side chamber,
- one in the peripheral part of the side chamber, 12.5 mm distant from the centre.

In this way it is very straightforward to extrapolate the numerical results in the ignition points that are the position in which the experimental flow characteristics are measured and to assign the ignition position for further combustion simulation. Starting from the origin (0,0,0) the side chamber is created, on the x-y plane, the side chamber develops in radial and circumferential directions, while the thickness of the side chamber is directed like the z axis while the cylinder axis is directed parallel to y axis.

In Figure 4.4 the engine geometry is reported with its reference frame. From that picture it is also possible to see different coloured parts. Each one represents a

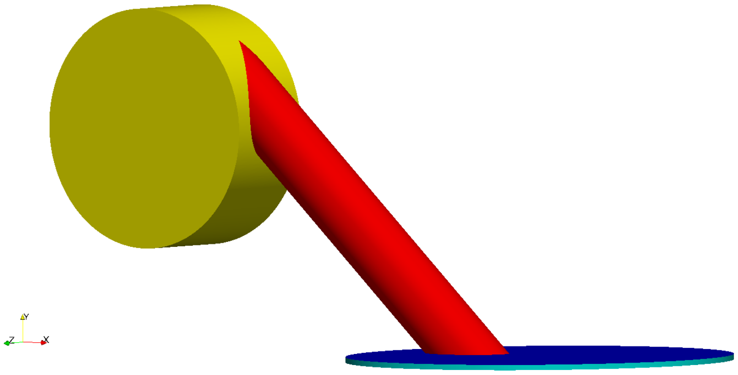


Figure 4.4: Engine geometry and reference frame adopted for the mesh generation.

different named patch, in yellow the side chamber, in red the connecting throat, in blue the cylinder head, in light blue the liner and parallel to the cylinder head there is the piston patch that it is not visible for practical reason. The cylinder patches need to be named as described, because the utility **moveEngineTopoMeh** for moving the mesh needs specific names for the patches as follows:

- *cylinderHead* for the head of the cylinder
- *piston* for the side of the cylinder that moves
- *liner* is the external cylindrical surface connecting the cylinder head to the piston surface.

In particular, for the structured mesh, the side chamber part is divided into five parts presented in different colours in Figure 4.5. In Table 4.2 the name of each coloured part is presented.

Colour	**Patch name**
Blue	SideChamberOgridFront
Light blue	SideChamberOgridBack
Green	SideChamberFront
Yellow	SideChamberBack
Red	SideChamberLateral

Table 4.2: Side chamber patches

All the others patches are the same as those presented in Figure 4.4.

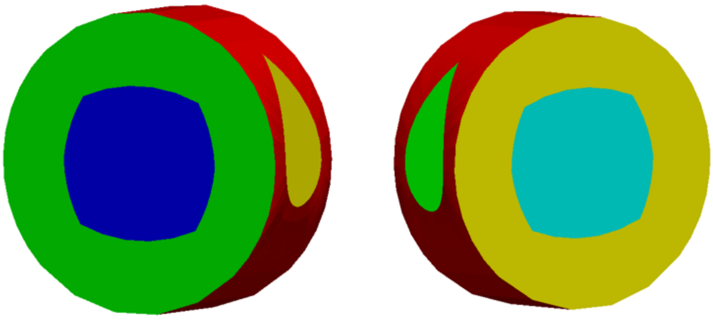


Figure 4.5: Patches of the side chamber for the structured mesh

4.2.2 Blocks generation procedure

Looking at the engine's geometry and considering that a structured mesh requires connectivity between all different blocks - meaning that a vertex of another cell corresponds to each cell vertex , and all the cells are hexahedral - it is possible to understand that the connecting throat requires a big effort to be designed, it is connected to the side chamber and to the cylinder head. Besides, in those parts a suitable blocks arrangement is required to ensure connectivity between all the blocks, so when we start building the side chamber also the blocks allowing the subsequent building of the connecting throat need to be taken into account. In particular, the geometry could be divided into three main parts:

- Side chamber
- Connecting throat
- Cylinder

Each part has a cylindrical geometry and to represent this type of geometry a common procedure is used: the creation of an internal block (called *O-grid*) that allows the creation of four other blocks connected to each side of the O-grid creating the final cylindrical geometry. Figure 4.6 presents a typical example of the construction of a structured mesh using an O-grid.

The O-grid technique is used for creating all three different parts of the engine, but it requires some improvements since the three cylindrical components lie in three different planes. Specifically, the side chamber lies in the x-y plane while the connecting throat O-grid, contained in the side chamber, lies in the y-z plane and the O-grid of the side chamber is a cube lying in the x-y plane, so there is the need to generate 2 concentric O-grid that allow a sufficient number of vertex to build the internal O-grid of the side chamber. In details, the internal O-grid consists of four blocks arranged as in figurename 4.7. According to the reference

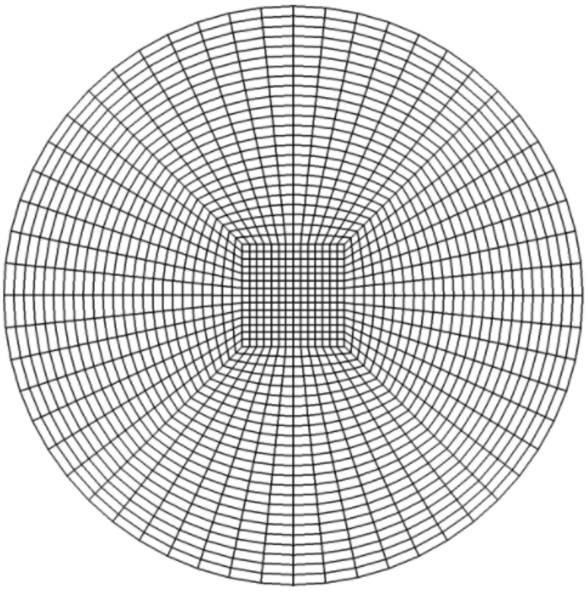


Figure 4.6: Example of cylindrical structured mesh with its O-grid

system presented at the beginning of this chapter, four blocks in the y-z plane are created and extruded along the x direction.

Figure 4.7: Internal O-grid for generating the O-grid of the side chamber

The next step is to use each side of the blocks shown in Figure 4.7 to generate others eight blocks constituting together a circular geometry in the y-z plane, (Figure 4.8) i.e. the connecting throat part that has to be subsequently extruded according to the engine geometry.

The generation of curved edges is possible by specifying the shape of the edges connecting different point in the **blockMeshDict**. Having eight vertexes on the external edge, it is possible to build the O-grid for generating the side chamber O-

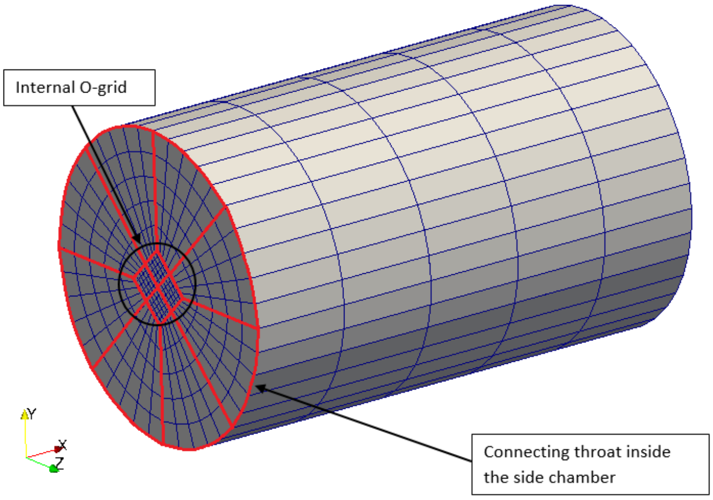


Figure 4.8: Connecting throat y-z plane

grid in the x-y plane, adding further eight blocks. In Figure 4.9 the resulting O-grid is shown. We can see that if only one O-grid was used instead of two concentric O-grids after the generation of the connecting throat, there would be only two points instead of three for each quarter of the side chamber O-grid, making it impossible to generate the O-grid of the side chamber.

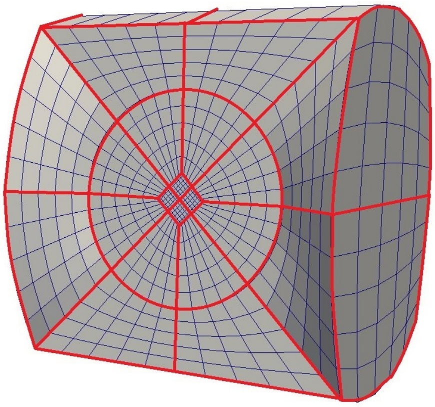


Figure 4.9: O-grid to generate the side chamber of the engine

The following step consists in extruding the connecting throat contained in the side chamber O-grid at both extremities: on one side it is extruded up to the lateral circular surface of the side chamber, while on the other side a suitable extrusion to accommodate the junction to the connecting engine throat is created. This junction section has been source of some issues related to high skew and non-orthogonal faces, because in that part the connecting throat needs to turn in the correct direction according to the geometry presented in Figure 4.4. In Figure 4.10 it is possible to see three pictures that show the addition of twenty blocks to complete the connecting throat inside the side chamber.

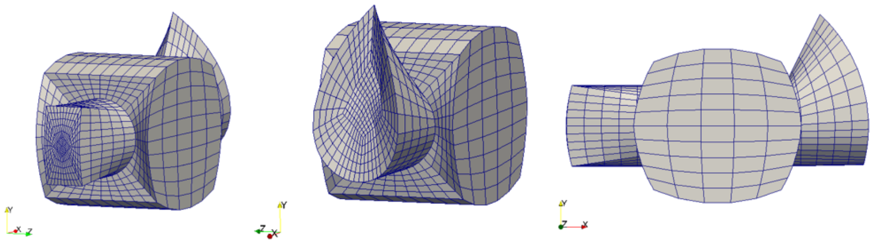


Figure 4.10: Extrusion of the connecting throat inside the side chamber

The central view of Figure 4.10 shows the extrusion allowing to connect the side chamber to the connecting throat part. Here we can see that the external faces do not lie in a plane: this was necessary to reduce as much as possible the number of skew and non-orthogonal faces. Another important aspect visible in Figure 4.10 is the complex edge necessary to correctly represent the side chamber and connecting throat junction. These edges were created defining the edges as spline, in such a way the points that lie on the edges are specified. The coordinate of the points for the spline generation are obtained from a CAD software. First, the three engine geometry parts (side chamber, throat and cylinder) were designed and then it was possible to extrapolate the coordinate of the edge intersection between throat and side chamber and between throat and cylinder, which are the most difficult edge to be reproduced. At this point, it is possible to add twenty-two blocks to complete the generation of the side chamber part as presented in Figure 4.11.

The connecting throat is subdivided in three parts to better arrange the curvature in the two junctions section:

- one on the side chamber side
- one in-between side chamber and cylinder
- one on cylinder head side

Using this arrangement, a straight central part is created, in which the cells have the same direction as the flow, a condition that decrease the numerical diffusion. In Figure 4.12 the results of the procedure described are reported.

The last part that constitutes the engine is the cylinder. First, the connecting throat is extruded of 1 mm (engine clearance) parallel to the y axis, then, as done for the other cylindrical components of the engine, an O-grid was generated to

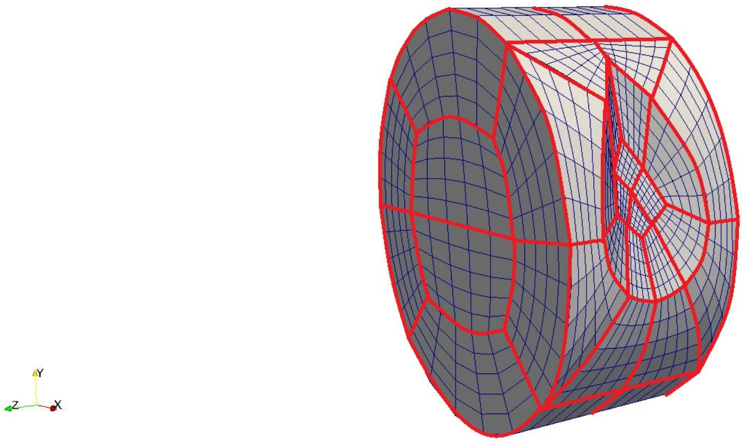


Figure 4.11: Blocks constituting the side chamber

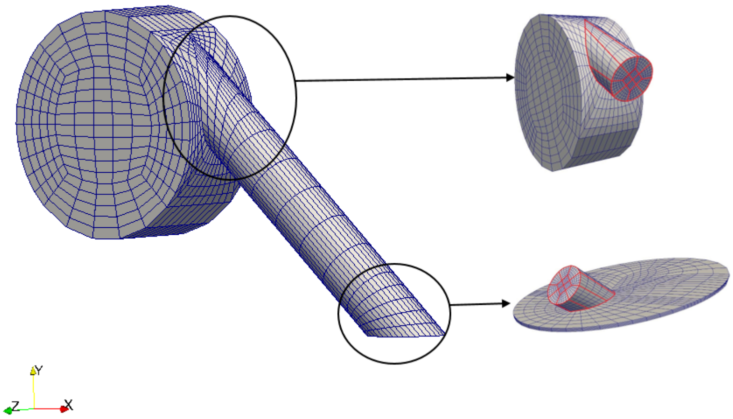


Figure 4.12: Junctions part of the connecting throat

have the possibility of creating the blocks constituting the cylinder. Particular attention is paid to the edges of the cylinder O-grid because by curving them it is possible to avoid high non-orthogonal faces thus improving the quality of the mesh. Another important aspect is the number of cells along y direction, the axis of the cylinder, since using a mesh deformation and remapping method to move the piston, there is the need to avoid a big number of cells in only 1 mm, otherwise the aspect ratio of that cells would be very high producing a poorquality mesh. Concluding, there are five cells parallel with the cylinder axis. In Figure 4.13 the results of the cylinder mesh generation are shown.

The final mesh obtained is presented in Figure 4.14

The grid has 15625 cells and non-orthogonality and skewness checks are ok.

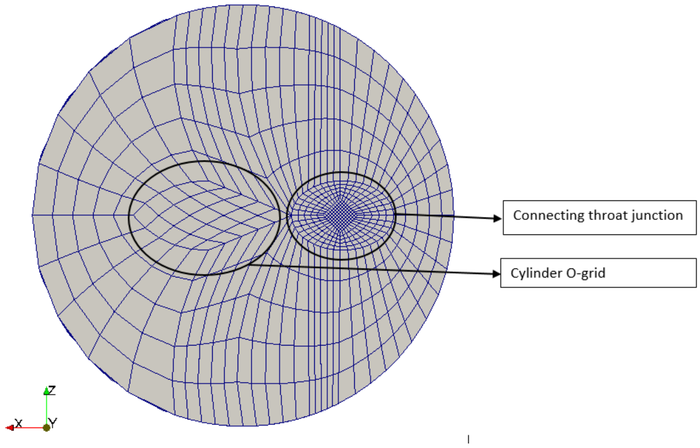


Figure 4.13: Detail of the blocks constituting the cylinder part

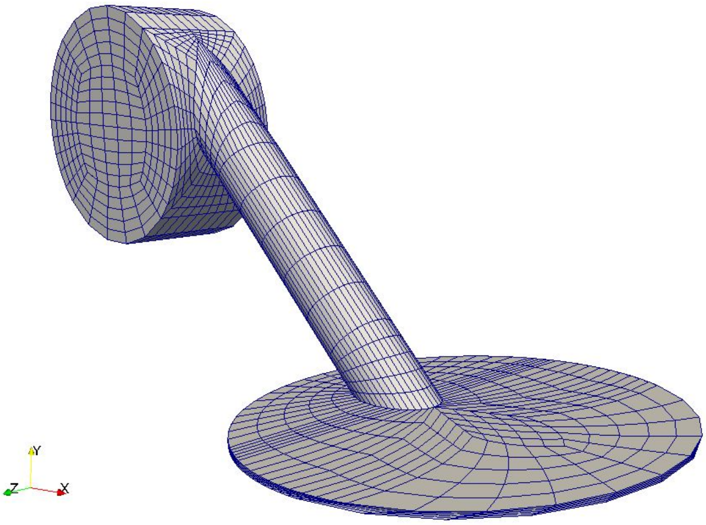


Figure 4.14: Coarse mesh

The last step is to rotate the mesh to have the z axis directed like the cylinder axis: for doing this, is necessary to use **moveEngineTopoMesh**. This task was performed switching y axis with z axis by means of an OpenFOAM utility (**rotateMesh**).

4.2.3 Refined structured mesh

In CFD simulations, it is a common practice to verify if the results change increasing or decreasing the mesh refinement. This procedure is called mesh sensitivity analysis and ensure the mesh independence of the results. For this purpose, a refined version of the mesh described in Section 4.2.1 is built. The refinement is obtained by increasing the number of cells of the internal O-grid in the x direction according to what shown in Figure 4.7. In this way a circumferential refinement in the side chamber is obtained. In the connecting throat the number of cells along its axis is increased to obtain cells with a better aspect ratio. Mesh structure is identical to the one present in Section 4.2.1 and the resulting grid is presented in Figure 4.15.

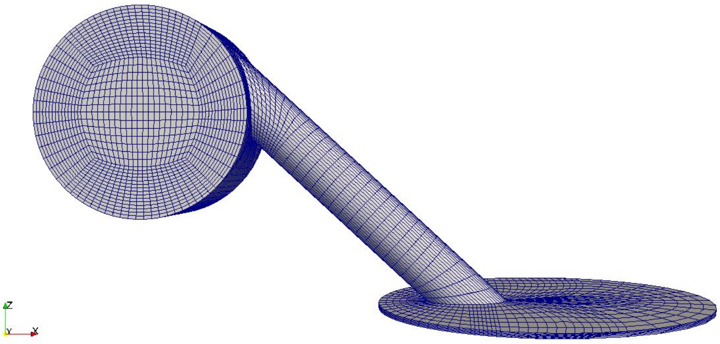


Figure 4.15: Refined mesh

The refined mesh has 81977 cells, and all checks are ok.

4.2.4 Mesh for flow initialization

Reacting simulations were performed on the test engine and to this end it is necessary to initialize the flow field for the combustion case adding the spark plug electrodes to the mesh. The base mesh used is the one presented in Section 4.2.1. The electrodes geometry is designed according to [14] with a CAD software able to generate a STereoLithograhy file (.STL) that reproduces in detail the geometry of the thin wire electrodes. The spark plug electrodes are two cylindrical wires of 0.8 mm diameter, and the spark gap is 1 mm. The ignition positions are two, the central and the peripheral one, so two STL files were created - one for each different ignition point. In Figure 4.16 the detail of the spark plug electrodes is shown.

Using the automatic mesh generator **snappyHexMesh** the spark plug electrodes geometry is generated in the side chamber. In the next sections, a detailed overview of the procedure for generating the central and peripheral ignition meshes is described.

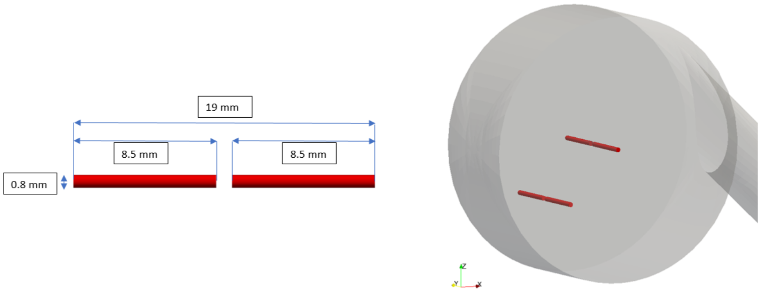


Figure 4.16: Detail of the spark plug electrodes geometry (left) and position in the mesh

4.2.5 Peripheral ignition mesh

The mesh generation with peripheral ignition starts from the coarsest structured mesh.

The *polyMesh* folder of the coarsest structured mesh is taken as the starting point for the generation of a grid that has the spark plug electrodes. To this end, the STL file, representing the peripheral spark plug electrodes, is put in the *triSurface* folder. Then, using the utility **snappyHexMesh**, that remove the cells inside the STL file, the generation of the grid with peripheral spark plug electrodes is performed.

Refinement zones are added:

- a cylinder that has the axis coincident with spark plug electrodes, 1 mm radius, the length equal to the clearance of the side chamber (19 mm) and a minimum level of refinement of 2 and a maximum of 3.
- a cylinder of 1 mm length and 1 mm radius that lies in the spark gap zone, with a minimum level of refinement of 3 and a maximum of 4.

On the spark plug electrodes surface, a minimum level of refinement of 4 and a maximum of 5 is specified. These refinements are a good compromise between the increasing number of cells, and a sufficiently good representation of the spark plug electrodes geometry. The mesh obtained has 26661 cells and is shown in Figure 4.17 where it is also possible to see the refinement level.

4.2.6 Central ignition mesh

The mesh with central ignition is generated according to the same procedure as the one followed for the peripheral ignition mesh, and the same level and zones of refinement are used. There was an issue in the generation of the mesh since

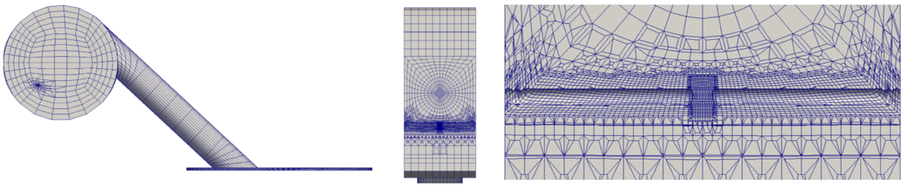


Figure 4.17: Detail of the mesh with peripheral ignition (left) and refinement levels (centre and right)

in the central part of the side chamber the aspect ratio of the base mesh cells is too high and **snappyHexMesh** is not able to reproduce the spark plug electrodes geometry acceptably. To solve this issue the base mesh was refined. In particular, the cells in the x direction of the O-grid in Figure 4.7 were increased since the mesh is structured, the number of cells also increase in the circumferential direction of the side chamber.

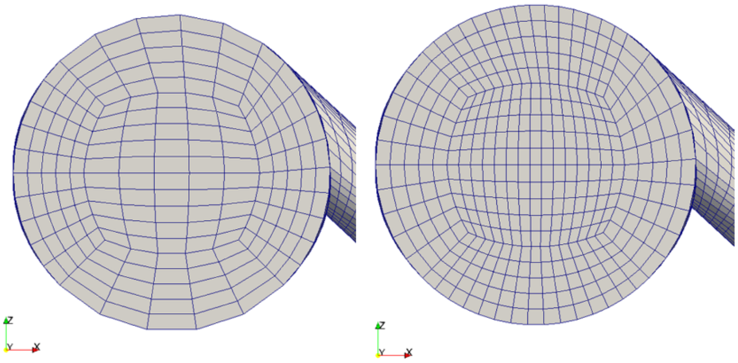


Figure 4.18: Base mesh (left) and base mesh refined in the x direction (right)

In Figure 4.18 the base mesh is compared before (left) and after (right) the refinement necessary to obtain a good representation of the central spark plug electrodes geometry. The resulting mesh has a greater number of cells compared to the one with the peripheral spark plug electrodes due to this further refinement of the base mesh. Finally, the grid has 36695 cells and it is shown in Figure 4.19

4.2.7 Mesh for combustion cases

The grid for combustion simulations requires a peculiarity: the mesh has to be regular as much as possible in the zone where the combustion takes place (side chamber). This requirement implies the creation of a new mesh that satisfies this latter observation since the structured mesh used to initialise the flow field

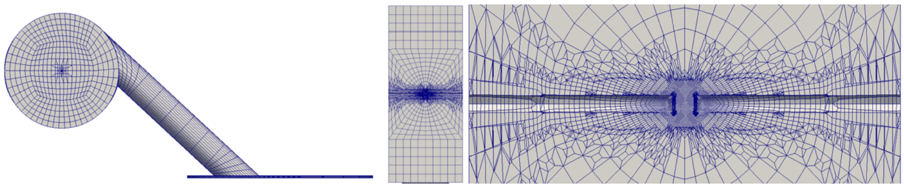


Figure 4.19: Detail of the mesh with central ignition (left) and refinement levels (centre and right)

features cells with different size and high aspect ratio in the central part of the side chamber.

The generation of a suitable mesh was performed thank to the **snappyHexMesh** utility according to the following steps:

- Creation of a base block mesh with 160x80x80 mm side featuring 80 cells along each edge(Figure 4.20).
- Extraction of the final mesh by keeping only the cells inside the actual engine geometry reproduced in detail by a STereoLithography (STL) file.
- Addition of one refinement cylinder of 1 mm, featured by a refinement level of 2. The cylinder axis lies on the spark plug electrodes cylinder.
- Addition of:
  - a minimum level of 0 and a maximum of 1 on the connecting throat surface,
  - two levels of refinement on the liner surface,
  - 1 level of refinement on the piston surface.

The base block mesh was generated using multiple cells grading along x direction (the one that start from the side chamber and moves towards the cylinder). In detail, the 35% of the length in that direction contains the 70% of the cells in x direction, the remaining 65% of the side length contain 30% of the cells (Figure 4.20). This procedure allows having more cells in the side chamber part where the combustion takes place, and less cells in the connecting throat and cylinder where the flow has been already initialised and does not change since the piston is close to the top dead centre and its movement is very small.

The combustion solver does not work well with concave faces, so to avoid this issue, a parameter controlling the maximum concaveness of the faces is set to 5°: concavity is allowed below this angle, if it is 0° the faces are straight, if it is less than 0° the faces would be convex. The mesh generation procedure described

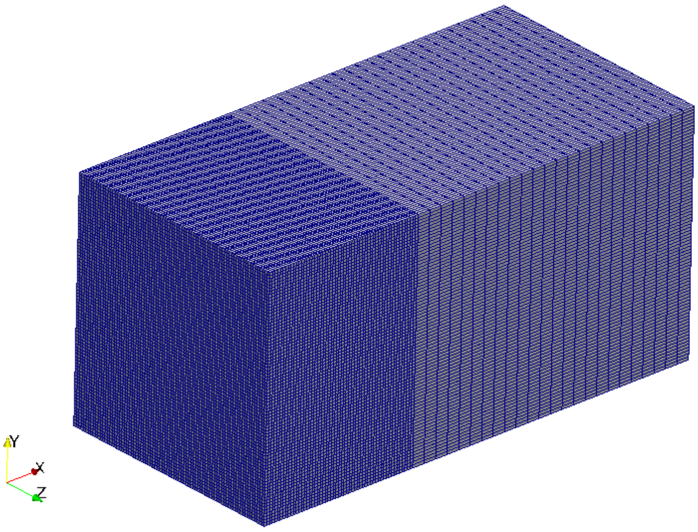


Figure 4.20: Base block mesh used for generating the mesh for combustion simulations

is the same for both grids with central and peripheral ignition. The mesh with central ignition has 64974 cells and is presented in Figure 4.21, while the one with peripheral ignition has 65119 cells (Figure 4.22).

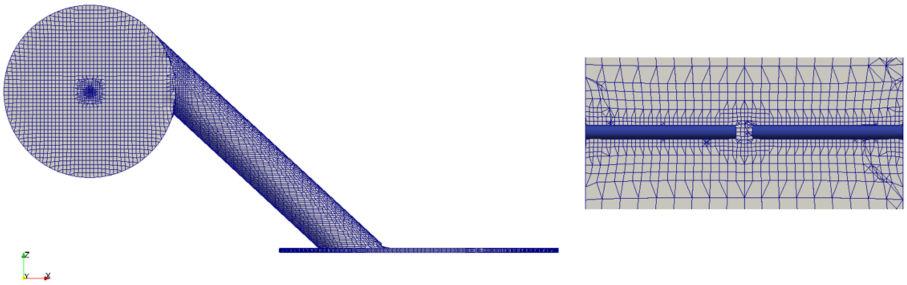


Figure 4.21: Detail of the mesh for combustion simulations with central ignition

After some attempts to calibrate the combustion model, it was found that the cells side length inside the combustion chamber needs to be equal at least to 0.4 mm to avoid mesh refinement dependence of the results. The required mesh was generated starting from the same block used for the previous mesh with the same arrangement of the cells in percentage per each side, but the number of cells along the three edges was increased to 100. In this configuration, the 35% of the x length has the 70% of the total cell along the x direction, 35% of 160 mm is equal

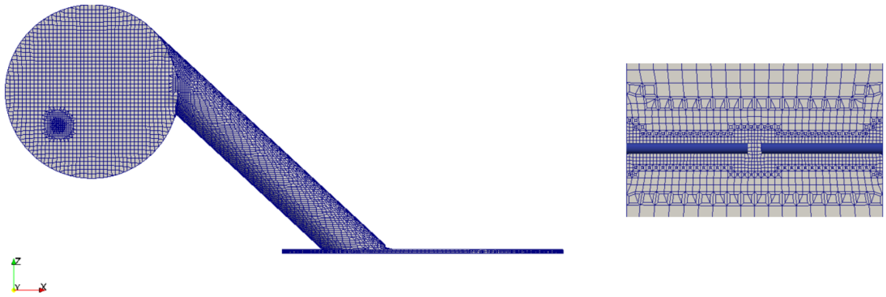


Figure 4.22: Detail of the mesh for combustion simulations with peripheral ignition

to 56 mm, which are subdivided into 70 cells leading to a base mesh with 0.8 mm side for each cell. The block side length along y and z direction is equal to 80 mm that are subdivided into 100 cells leading to 0.8 mm side length for each cell. After the generation of a block with these features (presented in Figure 4.23), *snappyHexMesh* utility was used to generate the engine mesh, in particular to satisfy the requirement to have 0.4 mm length side inside the side chamber, a level of refinement equal to 1 is applied to a cylindrical region according to the following geometrical properties:

- the axis of the cylindrical region lays on the same axis of the side chamber (y axis),
- the radius of the cylinder is equal to 23 mm, allowing covering all the side chamber part
- the length of the cylinder is equal to the length of the side chamber, 19 mm.

Further refinement is implemented near the spark plug electrodes, a cylindrical region that has the length of the side chamber (19 mm), with the axis that is coincident with spark plug electrodes (central or peripheral ignition) and with 1 mm radius is generated with a level of refinement equal to 2. This level is a good compromise between a satisfactory geometry representation and a not too large number of cells that would increase computational time. The procedure described was adopted for generating both the mesh with central and peripheral ignition. The final mesh obtained is presented in Figure 4.24, in which it is possible to see the mesh with central (Figure 4.24.a) and peripheral (Figure 4.24.b) ignition. The number of cells for the mesh with central ignition is equal to 523968 while for the one with peripheral ignition is equal to 525052.

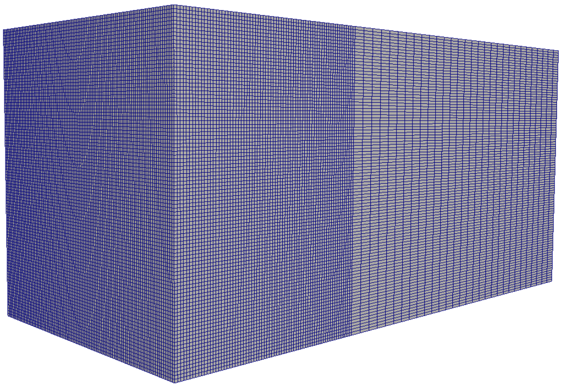


Figure 4.23: Detail of the starting block used for generating the required mesh characteristics for combustion simulations.

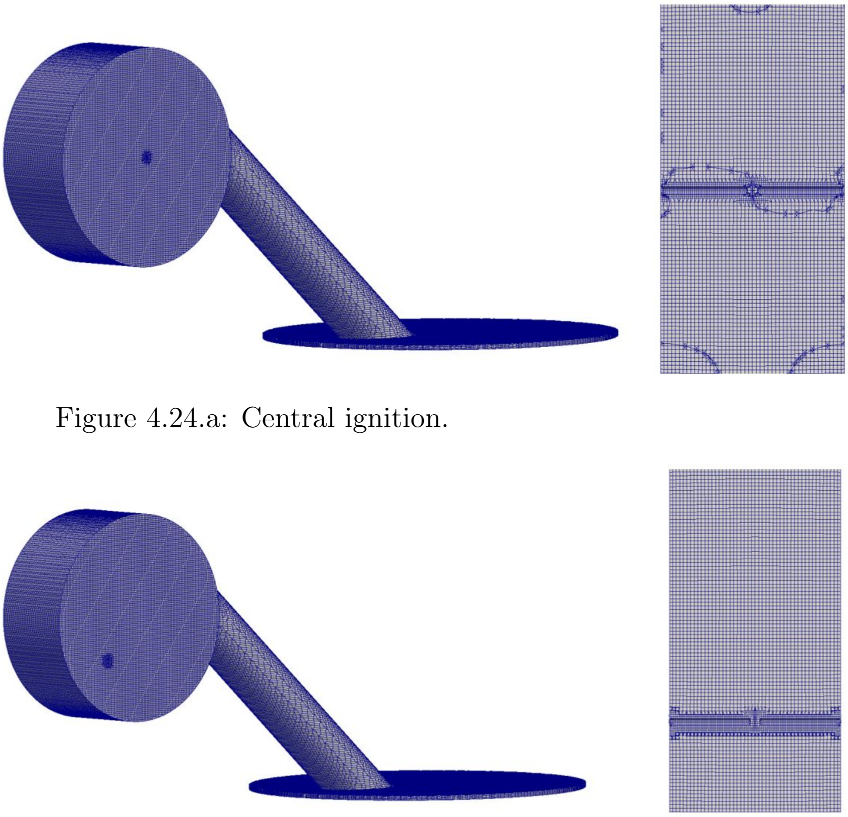


Figure 4.24.a: Central ignition.

Figure 4.24.b: Peripheral ignition.

Figure 4.24: Final mesh adopted for combustion cases.

4.2.8 Mesh movement

The cold flow simulations performed on this test engine starts from IVC and ends at ignition timing: this choice is justified by our interest in studying the effect of the generated flow field inside the lateral swirl chamber. Thus no interest is on the intake stroke but only the compression stroke, starting from the closure of the intake valve, is simulated because in that part the mixture contained in the cylinder is pushed through the connecting throat in the side chamber generating the flow field. This consideration involves the need to move the piston up to the IVC and then start the simulation compressing the cylinder mesh up to the ignition timing. For this task, two different methods were considered to allow the piston movement for the compression stroke:

- Layering,
- Mesh deformation.

Layering changes the topology of the mesh, layers of cells are removed during compression and added during expansion phase. A peculiar aspect that needs to be taken into account for this method is the necessity of a region with structured mesh where to perform the topology change. Not all engine geometries could be modelled with this tool, especially those with high geometrical complexity.
Mesh deformation technique does not change the topology of the mesh, connectivity and matrix shape remain the same. Moving from an initial mesh to a new deformed one is possible in a discrete manner once a set of grids is provided. Thanks to these properties, this approach has no limitations on mesh complexity. Mesh movement is managed with an automatic utility implemented in the LibICE library, ***moveEngineTopoMesh***. This application allows the movement of a mesh according to the two mentioned techniques. The tool automatically creates the new mesh as specified by the user. In particular, to avoid having a large number of cells in the cylinder part the mesh deformation approach is used. In this way, the number of cells in the cylinder remains constant - they are only deformed. This choice is justified by the limited computational resource, so to avoid long computational time it is better to have few cells in the cylinder where there is no interest in characterizing the flow motion and have a sufficient number of cells in the connecting throat and in the side chamber where the swirl motion is generated. In the utility ***moveEngineTopoMesh*** the layering approach is called **flatLayerAR**, while mesh deformation and remapping is named **simplePiston**.

moveEngineTopoMesh works according to the following procedure:

- creation of the mesh at Top Dead Centre (TDC),
- specification of bore, stroke, clearance and engine speed in a proper dictionary: *engineGeometry*,

- specify the approach to move the mesh (for the utility the mesh deformation method is called **simplePiston**),
- set up the start and end time in crank-angle degree (CAD),
- specify CAD increment between one mesh and the next,
- set the time step for writing the output folder in which the coordinate of the points of the deformed mesh are contained

This last step is very important because there is the necessity to have the mesh at the CAD of interest, when not the whole engine cycle is simulated.

In this section, the numerical settings of the utility ***moveEngineTopoMesh*** are described, explaining the choice made and the implementation of the case to generate the mesh at different CAD. To use ***moveEngineTopoMesh*** utility the set-up of some dictionaries is required. The dictionaries are in the **constant** and in the **system** folders of the case. In Figure 4.25 the structure of the case for move the mesh is shown.

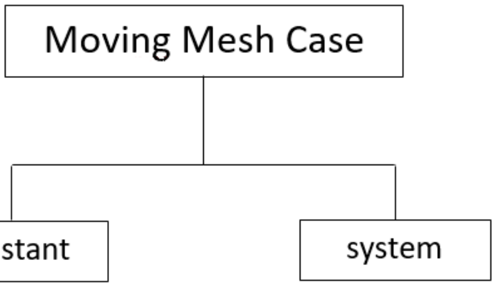


Figure 4.25: Structure of moving mesh case

The two main folders mentioned before are described:

- **constant** folder: it contains the base mesh at TDC that need to be deformed up to the IVC, the *engineGeometry* dictionary in which bore, stroke, clearance, connecting rod length and the engine speed are specified. Besides, the **simplePiston** method to deform the mesh needs to be specified. It has to be noticed that the base mesh is not generated in this process but it has been already created.
- **system** folder: it contains the dictionaries that manage the case. In particular, the *controlDict* dictionary requires the specification of start (-360 CAD) and end time (0 CAD): in this way the utility keeps the piston patch, pulls it stretching the liner patch from the TDC to the BDC. The *writeInterval*

and *deltaT* entries are required and need to be specified so that the interest CAD is written by the utility. As to the case presented in this work, the *writeInterval* and *deltaT* are equal to 1, so the application deforms the mesh every 1 CAD and writes the correspondent **polyMesh** for each CAD increment. In the generated folder, named with the number of the correspondent CAD, there is a **polyMesh** folder that contains only the file *points*, because the mesh is only deformed. As a result, the only thing that changes is the coordinate of the vertex of each cell in the cylinder. In the **system** folder there are also the *fvSchemes* and *fvSolution* dictionary, that are required as base dictionary to run an application in OpenFOAM, but they are not read by the utility.

Finally, the interest CAD is selected and the so-called file *points*, stored in the **polyMesh** folder of the CAD of interest, is copied to the **polyMesh** folder of the case in which the cold flow simulation will be done, substituting the coordinate of the piston point at TDC with the one generated by the mesh deformation. The structure of the cold flow or non-reacting case is the same as the one presented in Figure 4.25; there is also a time folder called with the number of the initial time step of the simulation and is -168 (as mentioned in Table 4.1 the IVC is at -168 CAD and the simulation starts from that time). In this time folder there are all initial conditions that need to be specified to the solver.

4.3 Cold flow analysis

In this section the procedure to set up the engine numerical model, the cold flow validation and, finally, the flow initialisation for the following combustion analysis is explained.

The validation of the numerical set up was done only for the peripheral spark plug electrodes location since experimental Laser Doppler Velocities (LDV) were defined only at this point([14]).

The working fluid inside the engine is a mixture of air and propane, as mentioned in [14], three different air fuel ratios (AFR) were tested: 1, 1.3 and 1.5. The mesh adopted to set up the engine numerical is the one with no spark plug electrodes (the one presented in Section 4.2.1), this choice was made because this mesh has lower cells compared to the one with electrodes leading to a reduced computational time and because, at this point, the interest was not in initializing the flow for combustion, but it was important to find the correct numerical set-up.

The cold flow analysis to choose the correct initial conditions, turbulence model and mesh (coarse or refined) is carried out at different rotational speeds for the mixture with AFR equal to 1. It is not necessary to validate the flow field regarding velocity and turbulence for different AFR because the low differences in terms of density and viscosity between the three different mixtures do not affect the flow field properties mentioned before. After the validation of the flow field quantities, the mesh with spark plug electrodes (Sections 4.2.5, 4.2.6) is adopted to initialise all the fields required for the combustion simulations. For this task, it is important to run simulations for the different engine speeds and AFR to obtain the proper flow initialisation.

4.3.1 Pressure calibration procedure

Initial pressure is required as initial condition for all simulations. A correct estimation of its value is crucial because at spark timing the pressure inside the engine must be equal to the one given in [14]. The pressure at ignition timing (5 bar [14]) is known but the initial pressure it is not specified: since the IVC is not at -180 CAD it is not possible to calculate the initial pressure using the compression ratio only. The procedure adopted to find the initial pressure's value is the following:

1. Run a simulation starting from IVC and finishing at spark timing with a guessed initial pressure of 1 bar.

2. Take the value of the pressure at spark timing and perform the ratio between the final and initial pressure to get an equivalent compression ratio between the IVC and the spark timing (Eq. (4.1)).

3. Divide pressure value at spark timing by the compression ratio to obtain the value of the initial pressure.

4. Run a simulation to verify the final pressure value at spark timing.

$$\beta = \frac{\text{Final pressure}}{\text{Initial pressure}} \tag{4.1}$$

This procedure was implemented on the coarse mesh to obtain results quicker, then it was also verified on the refined mesh and the results were as expected. Obviously, the rotational speed of the engine has no effects on the pressure at spark timing.

The Table 4.3 reports the value obtained from the procedure described.

Figure 4.26 shows the pressure evolution inside the engine starting from an initial value of 0.3513 bar at IVC and reaching a value of almost 5 bar at spark timing. The value is considered acceptable because the error on the pressure (defined in Eq. (4.2)) is less than 1%.

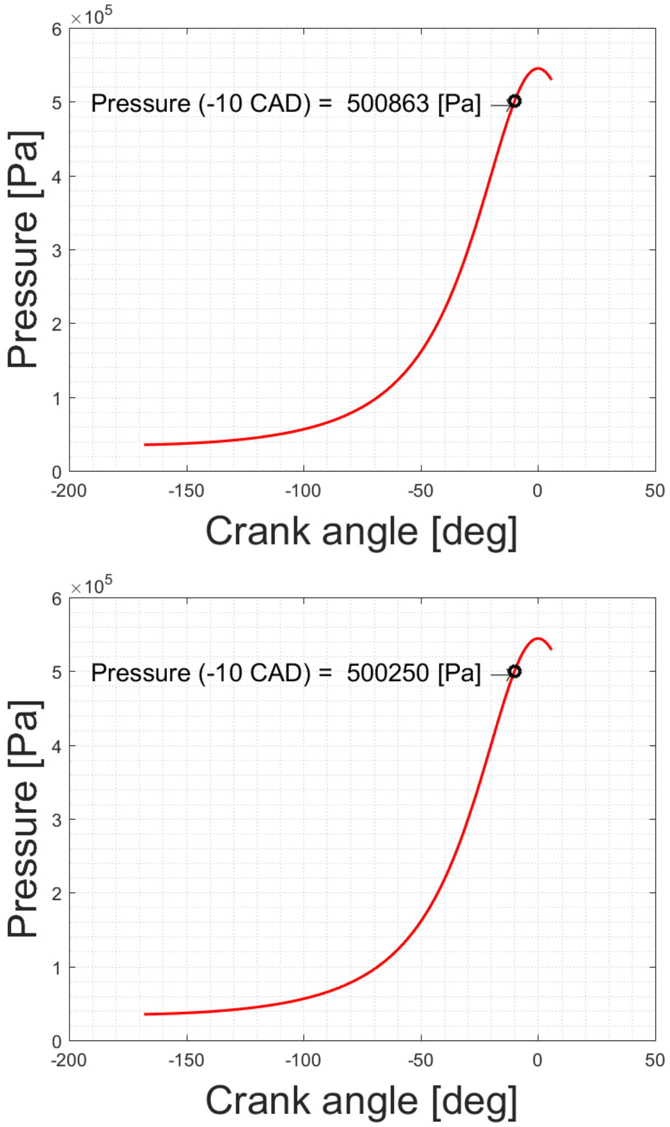


Figure 4.26: Pressure evolution inside the engine for the coarse (top) and for the refined mesh (bottom)

Initial pressure [bar]	**Final pressure [bar]**	β
1	14.2311	14.2311
0.3513	5	14.2311

Table 4.3: Pressure calibration

$$\varepsilon = \frac{\text{Final pressure-Target pressure}}{\text{Target pressure}} \ast 100 \qquad (4.2)$$

The value of initial pressure found by this procedure was used for all simulations.
In this first analysis, a comparison between the experimental measurements and the obtained numerical results is made, the quantities compared are the ones presented in [14]:

- Normalized mean swirl velocity at peripheral spark location.
- Normalized turbulence intensity (u') at peripheral spark location.

Normalized means that the interested quantity is divided by the mean piston speed (u_p) defined as:

$$u_p = \frac{2 * S * n}{60} \tag{4.3}$$

In Figure 4.27 it is possible to see that numerical mean swirl velocity has a very good agreement with experimental measurements and flow predictions while the normalised turbulence intensity is overestimated. The source of this significant difference could be the initial value chosen for the turbulence variables, the turbulence model adopted or the low refinement of the mesh. These aspects are going to be analysed in the following sections.

4.3.2 Turbulence initial condition sensitivity analysis

The turbulence model adopted in this initial part of the non-reacting analysis was the k-ε, i.e. a two equation model, reliable and stable.

To understand the reasons of the significant difference in turbulence intensity some initial values of k and ε were tested. In detail, three different tests were performed starting from values obtained in different ways:

1. Case 1: k and ε are found from a test case similar to the test engine.
2. Case 2: k and ε are estimated based on the mean piston speed through the CUP coefficient.
3. Case 3: k and ε are extrapolated from [14] and used.

In Table 4.4 the values for the three different cases are reported.

Case 2 requires some calculations, according to Eq. (4.3) the mean piston speed is calculated, then a value for CUP equal to 1 is chosen and from Eq. (4.4) the value of u' is calculated. Using Eq. (4.5) the value of turbulence kinetic energy is found. Finally, the value of ε is calculated with Eq. (4.6), where C_μ is equal to 0.09 and CL=0.07.

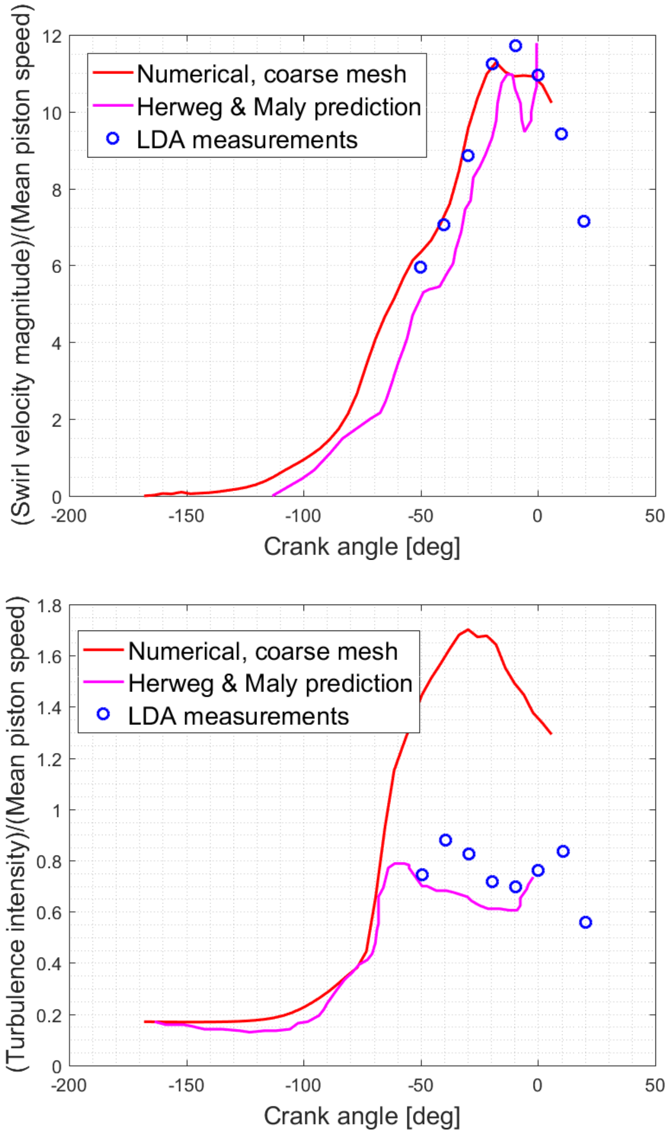


Figure 4.27: Comparison between numerical, LDA measurements and flow predictions of mean swirl velocity (top) and turbulence intensity (bottom)

$$u' = CUP * u_p \tag{4.4}$$

$$k = \frac{3}{2} * u'^2 \tag{4.5}$$

$$\varepsilon = C_\mu^{0.75} * \frac{k^{1.5}}{\text{CL*bore}} \tag{4.6}$$

Case	**1**	**2**	**3**
k $[\mathrm{m}^2/\mathrm{s}^2]$	31.74	0.67335	0.0882
ε $[\mathrm{m}^2/\mathrm{s}^3]$	8541.56	17.7673	1.9595×10^{-2}

Table 4.4: Turbulence initial conditions

Values of k and ε for Case 3 are extrapolated from [14] and used. The comparison between these three different initial conditions is shown in Figure 4.29. The three curves related to the three different initial conditions collapse onto the same line at a certain CAD, meaning that at the spark timing the different initial conditions do not influence the flow field at this moment. For this reason and to have agreement with the experimental findings the value of k and ε adopted for the next simulations are those found in [14]. The extrapolation process requires some calculations as the experimental data are normalized with respect to the mean piston speed. To calculate the value of k and ε the following procedure is adopted:

- Extrapolate the value of $\frac{u'}{u_p}$ from [14].
- Multiply the extrapolated ratio by the mean piston speed u$_p$.
- Calculate the value of k using Eq. (4.5).
- Calculate the value of ε using the Eq. (4.6).

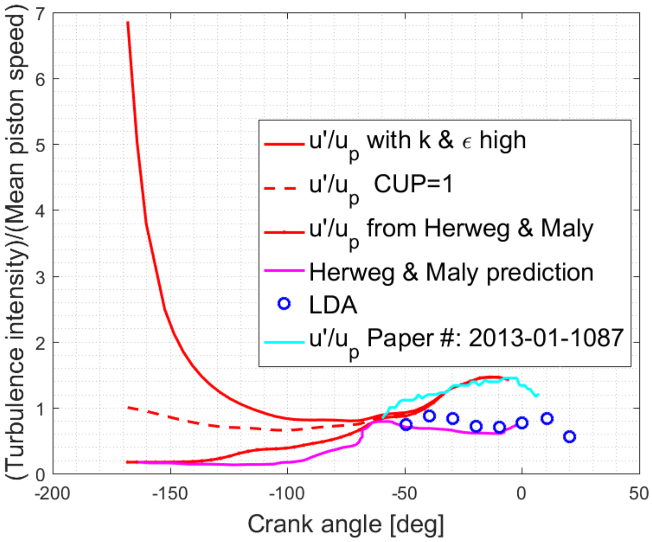


Figure 4.28: Comparison between *Case1, 2 and 3*

In conclusion, modifying the initial conditions on turbulence implies obtaining the same results at the spark timing, overestimating the experimental findings of u'. As mentioned before, another source of this overestimation could be the mesh refinement that could be too low, so the mesh described in Section 4.2.3 is used to try to increase the agreement with the experimental data and flow predictions. In the next section, a mesh sensitivity analysis is performed, comparing the results also with the data available from [20].

4.3.3 Mesh sensitivity analysis

To have a better agreement on turbulence intensity u' a mesh sensitivity analysis was performed. The grids compared are the coarse (Section 4.2.2) and the refined (Section 4.2.3) structured mesh. The comparison is made at the peripheral ignition location (where data are available) and the fields compared are the velocity magnitude and the turbulence intensity.

To have a quantitative measure of the error on the difference between experimental findings and numerical results, an error ϵ is calculated according to Eq. (4.7).

$$\epsilon = \frac{\text{Experimental U} - \text{Numerical U}}{\text{Experimental U}} * 100 \tag{4.7}$$

Equation (4.7) could also be used to estimate the error on u' and between numerical data available in [20] to have a quantitative idea of the overestimation. In Table 4.5 are reported the errors calculated for u' and U. It is possible to understand that the flow velocity is perfectly captured by the computational code while the turbulence intensity is also overestimated even with the refined grid. The refinement does not produce better agreement with the experimental findings, so the mesh adopted for continuing the flow validation is the coarse one, reducing the computational time and reaching satisfactory results.

	[14] VS Numerical	**[14] VS [20]**
U magnitude $[\frac{m}{s}]$	1.91 %	25.49 %
u' $[\frac{m}{s}]$	117.35 %	99.19 %

Table 4.5: Overestimation error

As the turbulence influences the initial stage of combustion, there is a great interest in improving the numerical results of the turbulence intensity, to obtain a better agreement with the experimental results. It is known ([10]) that in RANS approach, turbulence modelling is an approximation so that some differences could be expected but an attempt to reduce as much as possible the error is done. The

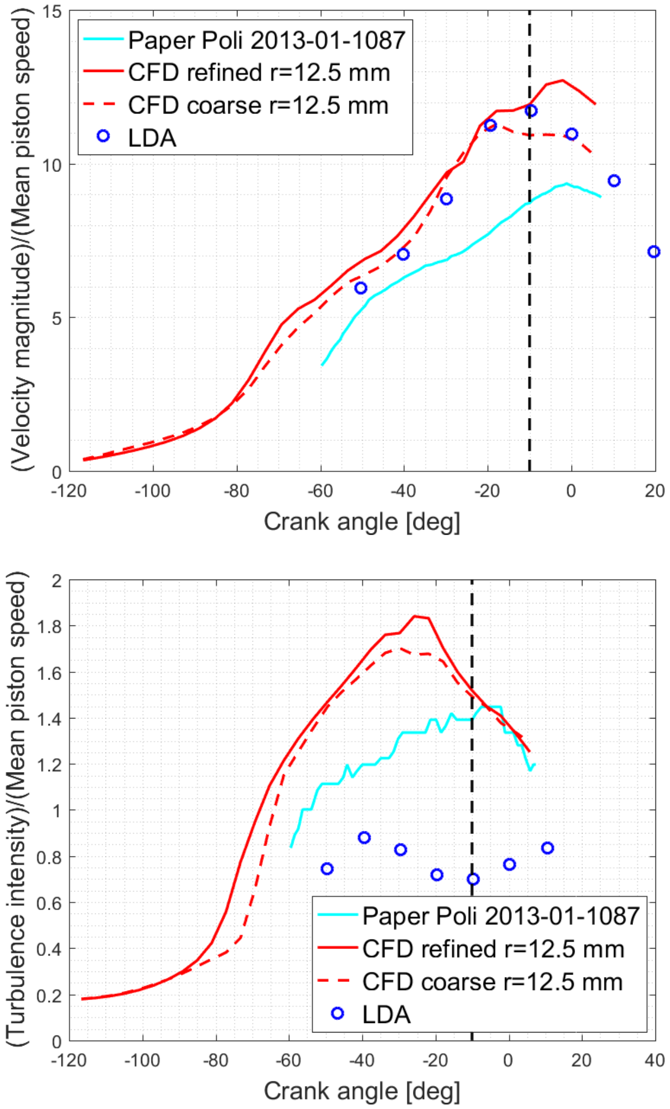


Figure 4.29: Comparison between numerical and measured flow velocity (top) and turbulence intensity (bottom) for 2 different meshes.

last possible source of error in estimating the turbulence intensity could be the adopted turbulence model. In the next section three different turbulence models are compared and, finally, the one that gives the smaller error is chosen for all simulations.

4.3.4 Turbulence models analysis

To improve the turbulence intensity agreement, the last possibility is to change the turbulence model. Three different turbulence models are compared:

- k-ε model.
- RNG k-ε model.
- k-ω SST model.

These models are used very often and give reliable results. The mesh adopted for the turbulence model analysis is the coarse one, and the validation of the results is done as previously described. The value of k and ε used are the same as *Case 3* and are used both for the k-ε and RNG k-ε turbulence model. For k-ω SST model the value of k is the same as for the other two models while the value of ω is calculated according to Eq (4.8):

$$\omega = C_{\mu}^{-0.25} * \frac{\sqrt{k}}{\text{CL*bore}} \tag{4.8}$$

Firstly, in Figure 4.30 the flow velocity and turbulence intensity are compared to the one presented in [14] and [20], then the error, calculated according to Eq. (4.7), is computed and the turbulence model that gives better agreement, and therefore the lowest error, is chosen.

It is possible to see from Figure 4.30 that *k-ω SST* turbulence model is the one that gives the best agreement compared to the other two models as well as to the numerical results of [20]. The error committed is almost halved concerning the k-ε model as shown in Table 4.6.

	RNG k-ε	**k-ω SST**
U magnitude $[\frac{m}{s}]$	9.01 %	5.67 %
u' $[\frac{m}{s}]$	104.89 %	72.20 %

Table 4.6: Error for different turbulence model

Concluding, the turbulence model adopted is the k-ω SST model, that gives results very close to experimental findings. The agreement with turbulence intensity, u' is not very good but it is the best results obtained comparing different turbulence models. The overestimation of normalised turbulence intensity with k-ω SST model at different rotational speed remains around 70%, in Figure 4.31 is shown the almost constant difference between normalised numerical and experimental *u'*. Whereas if the value of *u'* is considered (obtained multiplying the normalized value by the mean piston speed) it is possible to notice that the

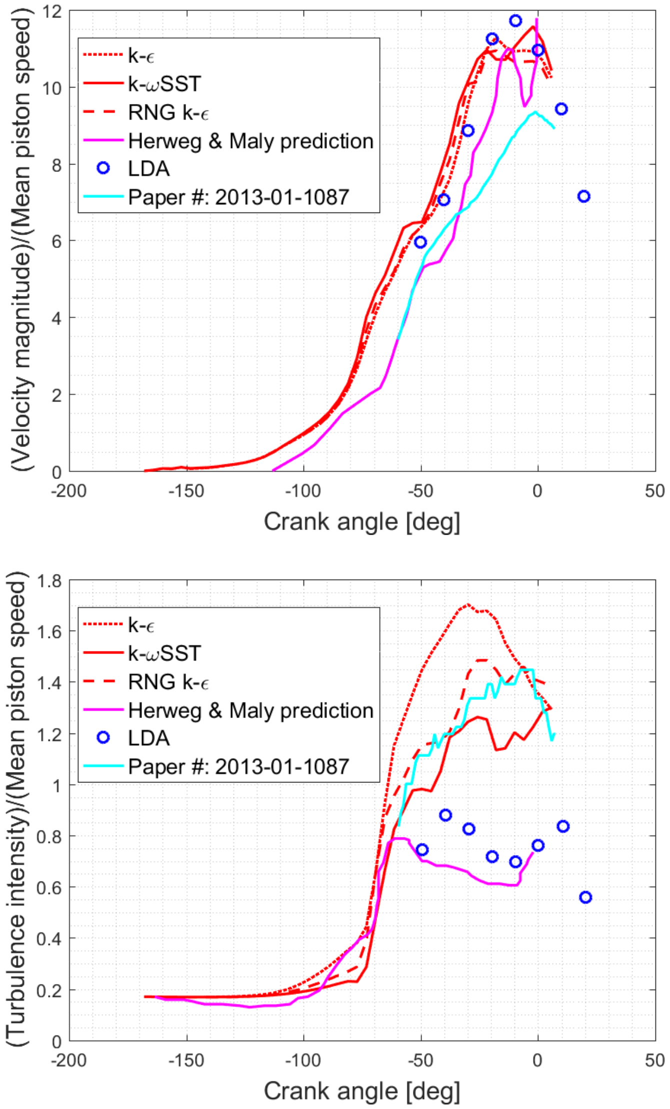


Figure 4.30: Comparison between different turbulence model for U (top) and u' (bottom).

difference between the numerical and experimental value of u' at spark timing increases with rotational speed, meaning that the magnitude of the overestimation is greater at higher rotational speed compared to the one at the lowest rotational speed (300 RPM). Turbulence intensity trend with increasing rotational speed and the increasing value of overestimation is presented in Figure 4.32.

These considerations on turbulence intensity lead to predict the behaviour of the burnt volume variable in the combustion cases. According to [9] the turbulent

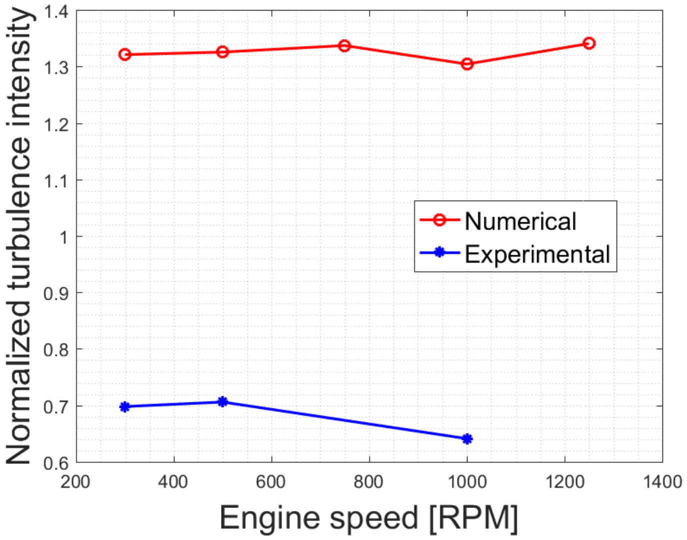


Figure 4.31: Numerical and experimental comparison of normalized *u'* at spark timing for different rotational speed.

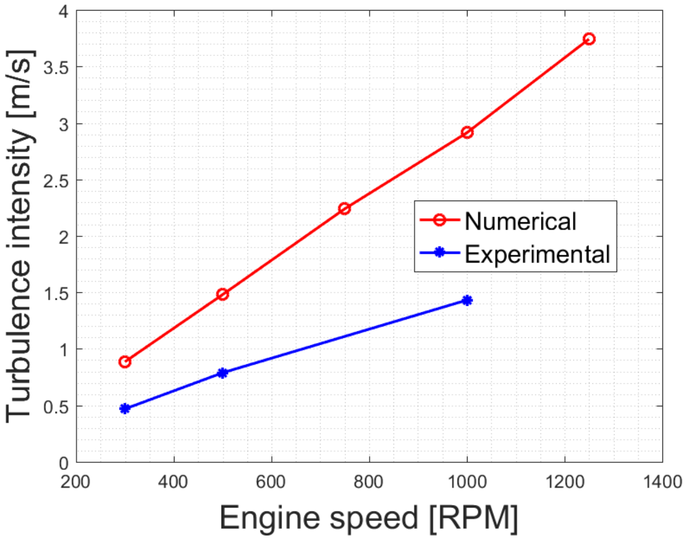


Figure 4.32: Numerical and experimental comparison of *u'* at spark timing for different rotational speed.

combustion velocity is proportional to the turbulence intensity (Eq. (4.9)), so the turbulence intensity overestimation results in higher turbulent combustion velocity. Consequence of this consideration is that the burnt volume is higher when the turbulent combustion velocity is higher considering a fixed time interval (all these considerations are explained in detail in Chapter 2.5 where the reacting simulations set-up and results are presented). Eq. (4.9) is the relation that link the turbulent combustion velocity (w_{tc}) with the turbulent intensity:

$$w_{tc} = w_{lc} + a \cdot w_{lc} \left(\frac{u^{‘}}{w_{lc}} \right)^{b} \tag{4.9}$$

Where w_{lc} is the laminar combustion velocity, a and b are constant depending on the engine and on its operating conditions.

4.3.5 Numerical schemes

Numerical schemes represent how the equation is discretized, the choice of numerical schemes is a very important task because it could bring a simulation to be stable or unstable and could increase or decrease the computational time.

In Table 4.7 and 4.8 are reported all the terms included in the equations solved with the numerical scheme adopted to discretize these terms and their numerical characteristics.

Term	**Scheme**
ddtSchemes	Euler
gradSchemes	default: Gauss linear
laplacianSchemes	default: Gauss linear limited 0.5
interpolationSchemes	linear
snGradSchemes	corrected

Table 4.7: Applied numerical schemes for time derivative, gradient, Laplacian, interpolation and surface normal gradient terms

Term	**Scheme**
div(phi, rho)	Gauss limitedLinear 0.5
div(phi, U)	Gauss limitedLinear 0.5
div$(phiU, p)$	Gauss linear
div(U)	Gauss linear
div(phi, k)	Gauss limitedLinear 0.5
div$(phi, epsilon)$	Gauss limitedLinear 0.5
div(phi, Yi_h)	Gauss limitedLinear 0.5
div$(phid, p)$	Gauss limitedLinear 0.5
div$((muEff * dev2(T(grad(U)))))$	Gauss linear

Table 4.8: Applied numerical schemes for divergence terms

Some considerations are made to explain the characteristics of the different numerical schemes adopted:

- Time derivative term is a first order, bounded and implicit method.
- Gradient schemes is set equal for all gradient terms, and it is a second order with Gaussian integration discretisation scheme.
- Laplacian scheme requires two entries: the interpolation and the surface normal gradient schemes (linear and limited, respectively) and it is a blend between first order bounded and second order unbounded scheme.

- Interpolation scheme adopted is a central differencing scheme (linear interpolation).
- Surface normal gradient apply an explicit non-orthogonal correction.
- Divergence terms have different schemes; the Gauss scheme is the only choice of discretisation and requires a selection of interpolation scheme for the dependent field. The schemes adopted are linear and limitedLinear: the first one is a second order, unbounded scheme while the second one is Total Variation Diminishing (TVD) scheme characterised by limited linear differencing.

The characteristics of numerical schemes available in OpenFOAM and used in this study are available in [4].
The numerical solvers for matrices are declared in the *fvSolution* sub dictionary. These solvers are associated to numerical algorithm used to solve the equations. The correct set-up of these solvers could increase the stability of the simulations also reducing numerical oscillations present (found) in the results. In particular, in this study, the turbulence intensity at a certain crank angle degree presents a peak that has no physical meaning. Initially, the Courant number was investigated to be sure that it is always less than 1, avoiding divergence in the solution. Since the Courant number always remains less than 1, a different solver was used. In Figure 4.33 the problem encountered is shown and the solvers adopted are presented in Table 4.9.

solver	PCG
preconditioner	DIC
tolerance	1×10^{-9}
relTol	0.1

(a) Solver for p|rho.

solver	PCG
preconditioner	DIC
tolerance	1×10^{-9}
relTol	0

(b) Solver for p|rho final.

Table 4.9: Non-Stabilized solvers

The ICE group develops a stabilized version of the solvers presented in Table 4.9 and has been used to avoid spurious numerical oscillations in the turbulence intensity. The results gained adopting this solution are presented in Figure 4.33 and in Table 4.10 is shown the stabilized solvers adopted.

The set-up of numerical model is done, the coarse mesh gives satisfactory results, the turbulence model adopted is the k-ω SST and the initial pressure is correctly estimated. The next task now is to initialize the flow for the three different mixtures (AFR=1, 1.3, 1.5) at the five different rotational speeds using the mesh with spark plug electrodes (Section 4.2.5,4.2.6). This initialization is necessary to have all the fields required for the combustion case initialized adopting the

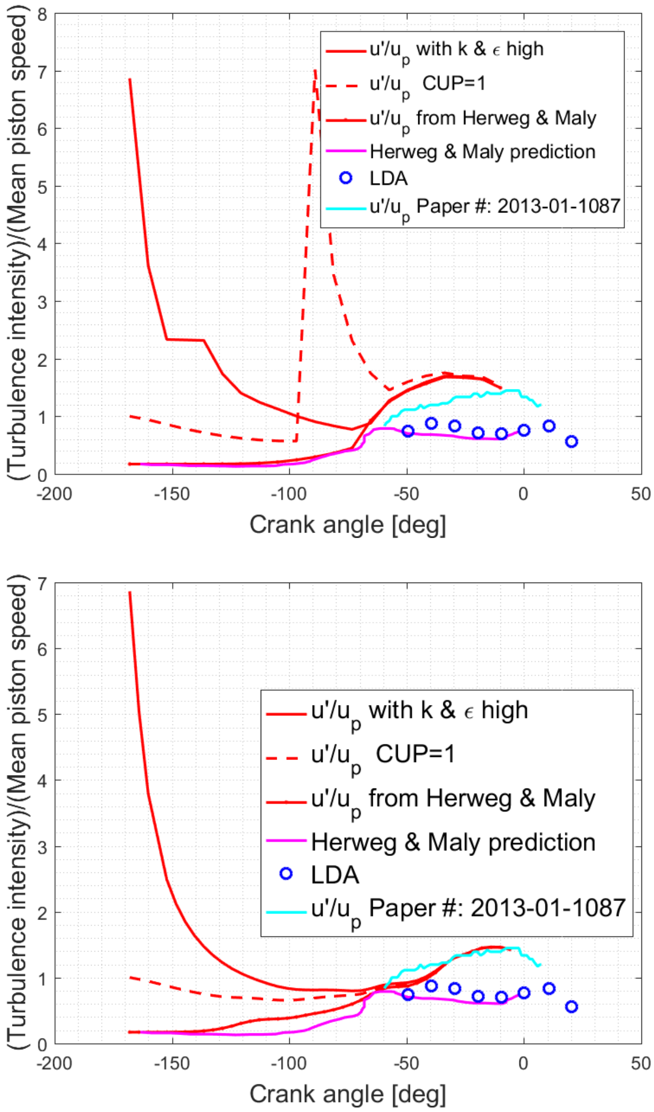


Figure 4.33: Comparison between turbulence intensisity obtained without stabilized solvers (top) and with stabilized solvers (bottom).

final geometry used for reacting simulations. Since the mesh adopted for combustion cases is not the same - it is more refined especially in the side chamber where the combustion process takes place - the flow field initialization was done on two meshes (one for the central and one for the peripheral ignition) coarser than the one used for combustion to avoid long computational time. Finally, the results for non-reacting flow initialization on the mesh with spark plug electrodes are mapped (using a LibICE utility: *engineMapFields*) on the mesh used for combustion.

solver	PBiCGStab
preconditioner	DILU
tolerance	1×10^{-9}
relTol	0.1

(a) U|k|epsilon|h|Z|Z2|Yi.

solver	PBiCGStab
preconditioner	DILU
tolerance	1×10^{-9}
relTol	0

(b) U|k|epsilon|h|Z|Z2|Yi Final.

Table 4.10: Stabilized solvers

4.3.6 Flow initialization

Once the numerical set up is found and the results obtained from the simulations give satisfactory agreements with the experimental findings, it is possible to initialize the flow for combustion simulations. For this task the meshes with spark plug electrodes presented in Section 4.2.5 and 4.2.6 are used. The simulations were performed with *pimpleColdEngineDyMFoam*, the same solver used for setting up the numerical model. The fields required in the initial time step folder are shown in Table 4.11 and 4.12.

Some specifications need to be made on *fuel, N2 and O2* fields. These three entries are related to the mixture composition, so when the air fuel ratio is changed from 1 to 1.3 and 1.5 the mixture becomes leaner, so the fuel content decreases while the air increases. In Table 4.13, the mixture composition for the three different air fuel ratios is presented.

Since the numerical model was already validated, in this part only the correct flow initialisation with different mixture composition for different rotational speed is done.

Boundary field	**Type**
alphat	HuhChangWallFunction
CO	zeroGradient
CO2	zeroGradient
fuel	zeroGradient
H	zeroGradient
H2	zeroGradient
H2O	zeroGradient
k	compressible::kqRWallFunction
mut	mutkWallFunction
N	zeroGradient
N2	zeroGradient
NO	zeroGradient
O	zeroGradient
O2	zeroGradient
OH	zeroGradient
omega	compressible::omegaWallFunction
p	zeroGradient
T	zeroGradient
Ydefault	zeroGradient
Yto2	zeroGradient
Z	zeroGradient
Z2	zeroGradient

Table 4.11: Fields required for initialize combustion cases for all geometry patches

Boundary field	**piston**	**other patches**
U	movingWallVelocity	fixedValue

Table 4.12: U field required for initialize combustion cases

AFR	**1**	**1.3**	**1.5**
fuel	0.060219	0.046975	0.040968
O2	0.218978	0.222064	0.223464
N2	0.720803	0.730961	0.735568

Table 4.13: Mixture composition for the analysed air fuel ratio

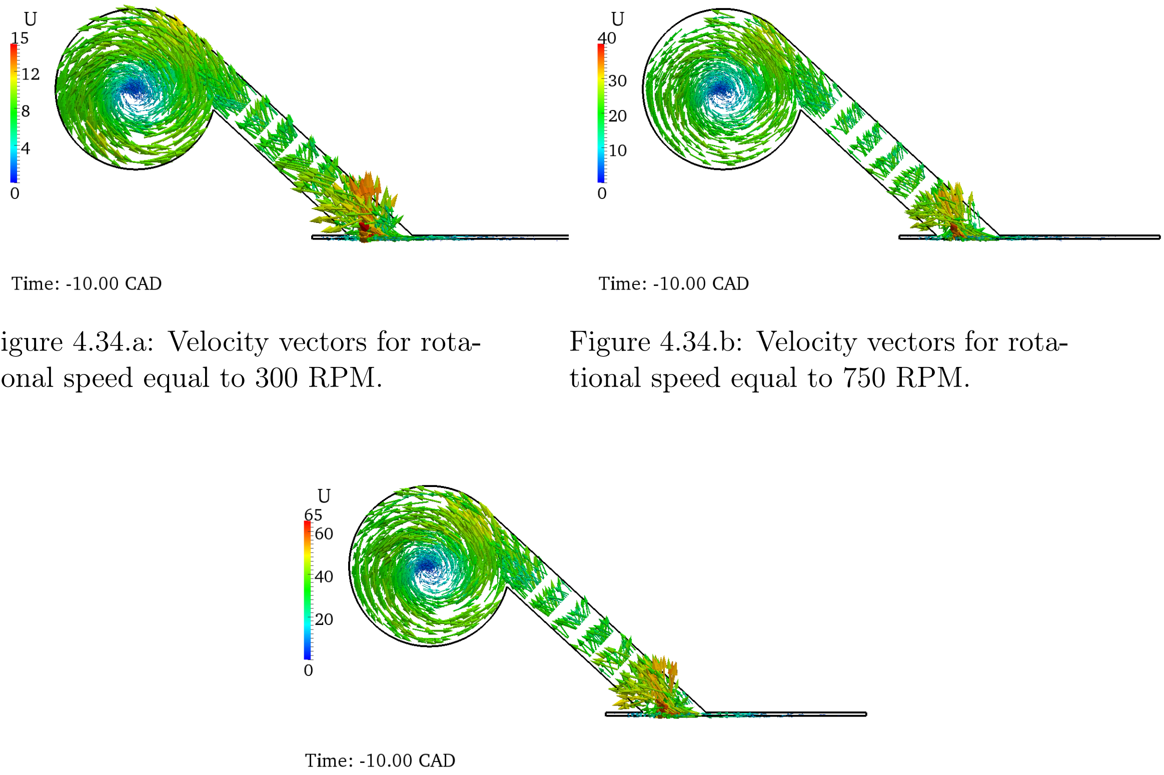


Figure 4.34.a: Velocity vectors for rotational speed equal to 300 RPM.

Figure 4.34.b: Velocity vectors for rotational speed equal to 750 RPM.

Figure 4.34.c: Velocity vectors for rotational speed equal to 1250 RPM.

Figure 4.34: Velocity field.

4.3.7 Conclusions

The numerical set-up for the test engine to correctly capture the flow inside the engine is found. Some overestimation on the turbulence intensity remains, which, as explained, could produce some differences for what concerns the burnt volume when reacting analysis is done. However, it is known from [9] that these two variables are linked and so if it is not possible to improve the turbulence intensity agreement also the burnt volume presents some differences between the experimental findings. In Figure 4.34 is shown the velocity magnitude vectors inside the engine for 3 rotational speeds (the extreme cases and the medium one in terms of rotational speed). In particular, it is possible to observe that the side chamber generates an intense swirl motion and in the centre of the lateral chamber the velocity is almost zero according to [14].

In the next section, combustion analysis is performed, starting from the cold flow simulations. The numerical results are then compared with the experimental findings available in [13].

4.4 Combustion Results

In this section, the numerical results obtained from reacting simulations are presented.

The numerical models used were explained in Section 2.5, and the validation of such model is performed comparing the simulations results with the experimental findings from [13]. The field investigated is the burnt volume as function of the time after ignition.

Simulations were performed on 1 processor to avoid problems derived from processors boundaries. The computational time required for simulate 0.5 CAD is approximately 12 hours.

The calibration of the ignition and combustion models was performed on the case at 750 RPM with air-fuel ratio equal to 1, the constants that need to be calibrated are the one for the ignition model, α_{AKTIM} (see Section 2.5.5), and the one for the CFM model, α_{CFM}. This last constant is associated to the production of flame surface density and it is the term α_{FSD} present in Eq. (2.50a).

The calibration of the two values for the combustion model was carried out on the first mesh presented in Section 4.2.7. Initially, a sensitivity analysis on the α_{AKTIM} value was done and the results are reported in Figure 4.35 where it is possible to see that the value equal to 0.5 gives the best agreement in the first stage of combustion.

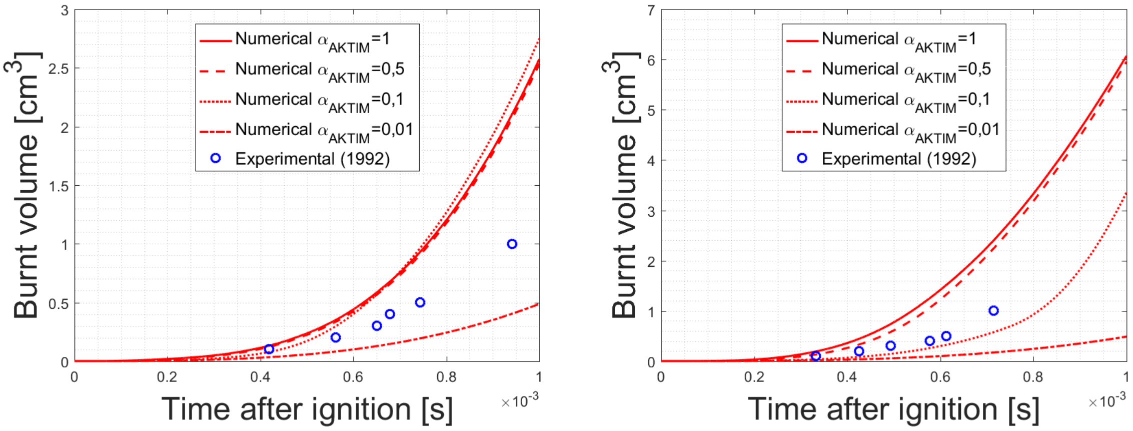


Figure 4.35.a: Central ignition.

Figure 4.35.b: Peripheral ignition.

Figure 4.35: α_{AKTIM} calibration for engine speed equal to 750 RPM and air-fuel ratio equal to 1.

These results were obtained with a value of α_{CFM} equal to 10. The next calibration was carried out on the α_{CFM} value, fixed the α_{AKTIM} value equal to 0.5.

In Figure 4.36 it is possible to see that a unique value of α_{CFM} that satisfies both the ignition positions does not exist. To improve the numerical results agreement, the second mesh presented in Section 4.2.7, that is more refined in the combustion side chamber, is adopted.

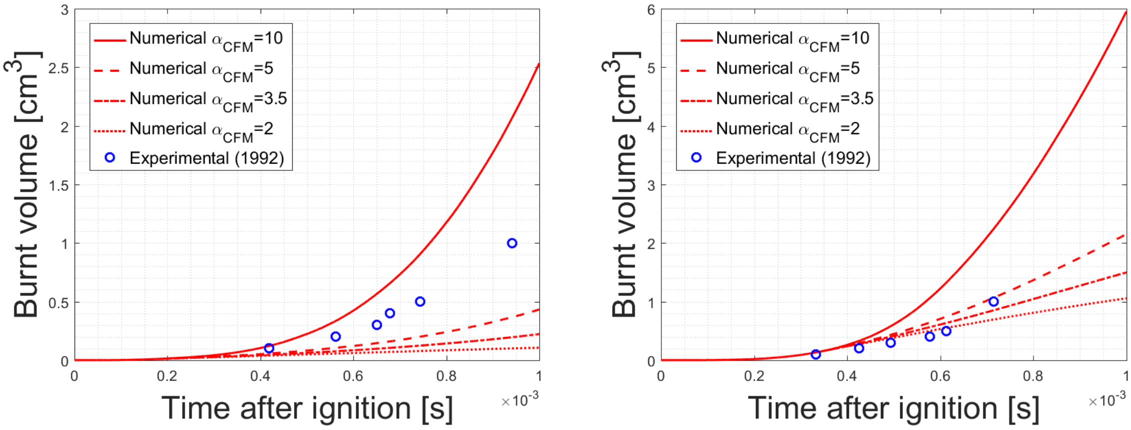


Figure 4.36.a: Central ignition.

Figure 4.36.b: Peripheral ignition.

Figure 4.36: α_{CFM} calibration for engine speed equal to 750 RPM and air-fuel ratio equal to 1.

Calibration is always done at the same test conditions (750 RPM, AFR=1) and the results are presented in Figure 4.37. The value of α_{AKTIM} is equal to 0.5 since it gives the best agreement with the experimental findings for both meshes in the first stage of combustion.

Finally, the values of the two coefficients for the combustion model calibration are presented in Table 4.14 and are adopted for all the rotational speeds, air-fuel ratio and for both ignition sites.

α_{AKTIM}	α_{CFM}
0.5	8

Table 4.14: Final values of combustion model constants.

The numerical results compared to the experimental findings as function of the time after the ignition are presented in Figure 4.38.

In Section 4.3.4, when the turbulence model analysis was presented, some observations were made about the overestimation in turbulence intensity, which could lead to an overestimation of the burnt volume since the two quantities are related. In Figure 4.38 this prediction is confirmed, but some clarifications are required. For what concerns the peripheral ignition site, it is possible to observe a good

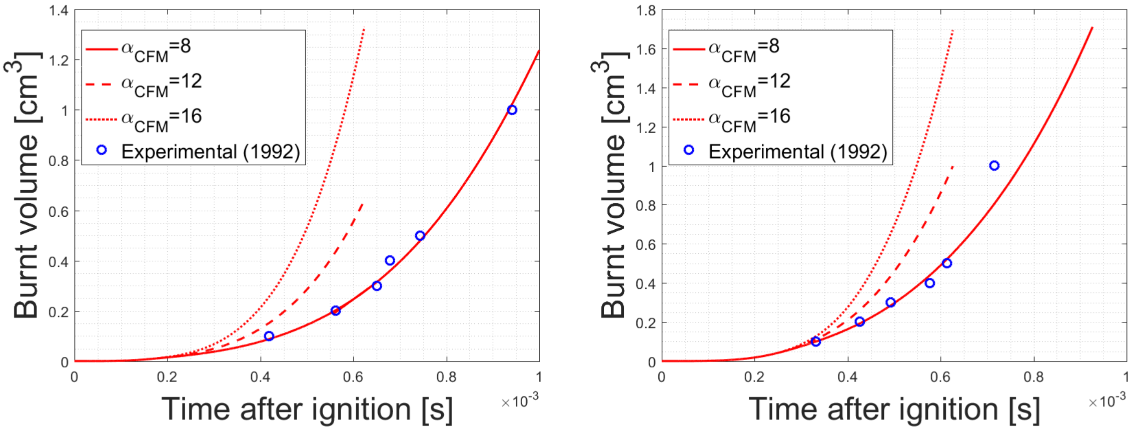


Figure 4.37.a: Central ignition.

Figure 4.37.b: Peripheral ignition.

Figure 4.37: α_{CFM} calibration for engine speed equal to 750 RPM and air-fuel ratio equal to 1 for the refined mesh.

agreement at 750 RPM at different air-fuel ratios. This is because the combustion model was calibrated at this engine speed. Considering that the turbulence intensity overestimation grows as the engine speed increases and since the combustion model is calibrated at an intermediate turbulence intensity, it could be noticed that the numerical burnt volume will be underestimated at engine speed lower than 750 RPM and overestimated at engine speed higher than 750 RPM. This is in agreement with what was presented in Figure 4.32 and with what was predict in Section 4.3.4. Considering the central ignition site, we see that the results are not in perfect agreement with the experimental findings. The reasons of this difference could lie in the lack of experimental measurements in the central part of the side chamber. The cold flow measurements were available only at the peripheral spark location, and in that position the best results obtained in terms of turbulence are quite higher compared to the experimental ones: probably in the centre of the side chamber the turbulence is more overestimated by the turbulence model thus leading to a better prediction of the burnt volume by the combustion model.

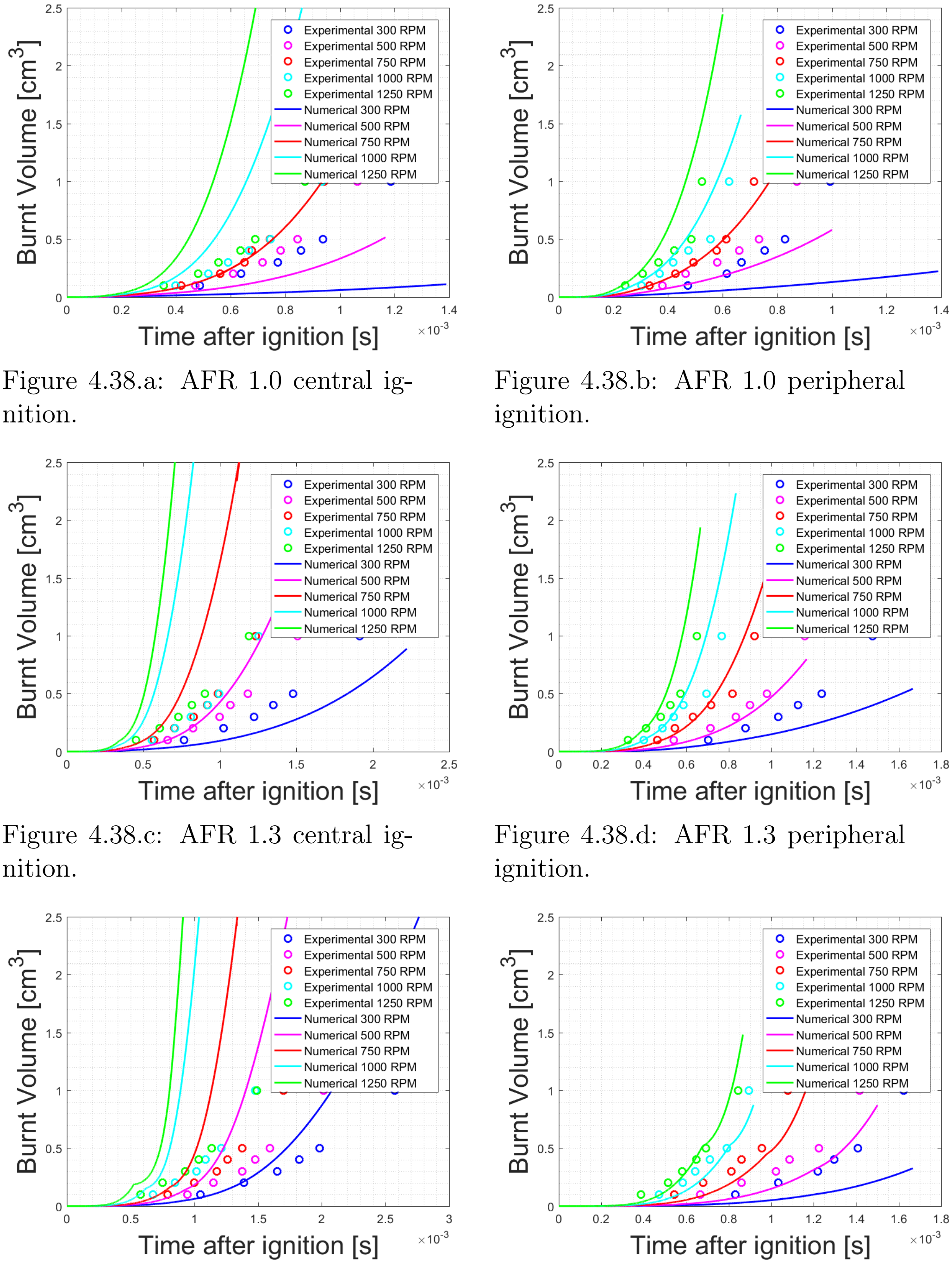


Figure 4.38.a: AFR 1.0 central ignition.

Figure 4.38.b: AFR 1.0 peripheral ignition.

Figure 4.38.c: AFR 1.3 central ignition.

Figure 4.38.d: AFR 1.3 peripheral ignition.

Figure 4.38.e: AFR 1.5 central ignition.

Figure 4.38.f: AFR 1.5 peripheral ignition.

Figure 4.38: Burnt volume.

Figure 4.38 shows the burnt volume trend, it is possible to observe that the numerical combustion model is able to represent the combustion process.

To confirm this last consideration it is possible to observe the effect that the ignition location and the mixture composition have on the combustion velocity and so on the burnt volume.

Moving from the central to the peripheral ignition location, since the flow velocity increase, it is expected that the combustion velocity increases, leading to a faster combustion. This consideration is supported both by the experimental and numerical results, in fact considering a fixed quantity of burnt volume, this is present earlier after the ignition for the peripheral location.

Increasing the air-fuel ratio means to produce a leaner mixture, consequently the combustion velocity reduces and also this aspect is confirmed by the numerical and experimental results. Looking Figure 4.38, and considering a fixed burnt volume quantity, it is possible to see that with increasing air-fuel ratio this quantity is present later, meaning that the combustion velocity is reduced.

These results show that the combustion model is able to correctly capture the combustion process, considering the effects of the flow velocity, turbulence intensity and mixture composition. Furthermore, it is possible to observe the importance in having a correct estimation of the turbulence before a combustion process take place because this quantity affect the combustion velocity.

5. Characterization of the flow inside a constant-volume vessel

In this chapter, a device able to generate a homogeneous and isotropic turbulence is analysed. Initially, the experimental rig is described, the characteristics of the mesh are introduced and, finally, the simulation strategy is explained.

In particular, the objective of this work was to find a solution using a steady-state and incompressible solver combined with the MRF technique to simulate the rotation of the fans: this set-up is a very simplified model that can give good and faster results compared to an unsteady solver.

The numerical results are finally compared with the experimental findings from [11] to validate the numerical model. Similarly to the test engine, this device was used to initialise a proper flow field for studying the initial flame kernel formation.

5.1 Description of the experimental device

The experimental study of combustion was carried out in a stainless steel-made spherical combustion chamber with a volume of 4.2 liters (internal diameter equal to 200 mm), represented in Figure 5.1. A heated wire resistor located on the exterior vessel surface (in orange in Figure 5.1), allows bringing gas temperature up to a maximum of 473 K. The installation is equipped with a vacuum pump enabling a residual pressure of less than 0.003 bar to be reached in the chamber before the introduction of the gases.

Inside the vessel, at the start of the intake phase, a mixer rotates to homogenise the reactive mixture. In addition, the various gases are introduced simultaneously into the chamber to ensure the best homogeneity as possible. The vaporisation temperature of the isooctane is 372 K at atmospheric pressure; to ensure fuel mixture vaporisation, the intake valve of the fuel/oxidant mixture is heated to 473 K as well. At the end of the intake phase, the agitator is stopped, and the premixed mixture is left to stand for 5 seconds before mixture ignition, to avoid turbulent disturbances during the propagation of the flame. The premixed gas is ignited in the centre of the chamber by an electrical discharge between two tungsten elec-

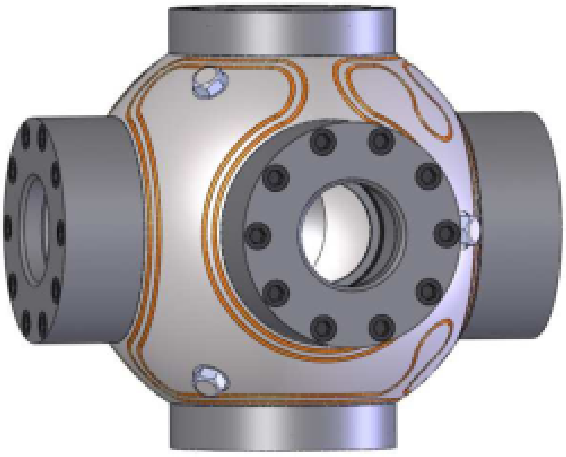


Figure 5.1: Schematization of the vessel

trodes (1.5 mm diameter, electrode spacing of the order of one millimetre), using a conventional ignition system. Since the voltage applied across the electrodes is limited, the electrodes must be close enough to create and maintain the electric arc. By that device, a maximum discharge energy of 95 mJ can be supplied between the two electrodes.

This device is equipped with four quartz circular cross-sections (thickness 60 mm, diameter 70 mm) allowing the installation of optical diagnostics devices. A photography of the experimental device is reported in Figure 5.2.

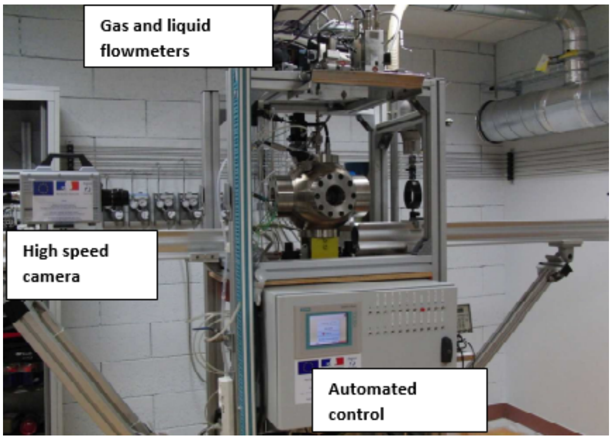


Figure 5.2: Experimental test rig configuration

5.1.1 Turbulence generation inside the combustion chamber

Turbulence is generated by six identical fans positioned near the inner wall of the vessel, which are arranged according to an octahedral configuration. Each fan has four blades, and the speed of rotation can be independently controlled over a wide range of speeds (0-15000 RPM). The rotational speed direction of the fans was chosen to direct the flow towards the centre of the vessel. In Figure 5.3 the vessel configuration and the fans arrangement is reported.

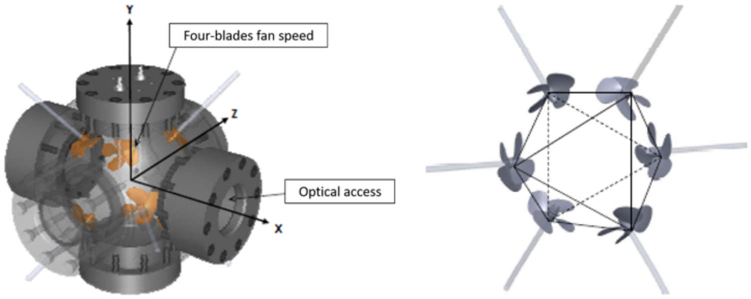


Figure 5.3: The Orleans vessel (left) and the detail of fans configuration (right),[11]

The non-reacting characterization of the vessel was carried out by testing different fans speeds (from *1000* to *15000 RPM*) under pressure and temperature conditions ranging between *1÷10 bar* and *300÷423 K*, respectively. In all tested conditions, the vessel was filled with suitable pressurized pure air.

The main results of the experimental activity were achieved inside a cubic region with *40 mm* side, centred with respect to the spherical vessel, where: an almost negligible mean flow velocity, a nearly constant and isotropic turbulence intensity were observed.

5.1.2 Measurement device: Laser Doppler Velocimeter LDV

The LDV device used in [11] is a two-component Dantec system, using green and blue wave lengths (respectively 541.5 nm and 488 nm) of an ionised Argon laser of 6W rated power. The air inside the vessel is seeded with droplets of silicone oil using a suitable device. The post-treatment of the Doppler puffs is carried out by the BSAFlowTM software to obtain the two speed components - U and V. The acquisition time of the signals is fixed at 100 s, which makes it possible, on the one hand, to ensure the convergence of the average speeds $\bar{U}$ and $\bar{V}$ and fluctuations u'and v', and, on the other hand, to enable the detection of the lowest frequencies during spectral analysis of velocity fluctuations. The average frequency of acquisition depends on the frequency of passage of the particles in the measurement volume. Depending on the case studied, the average acquisition frequency is between 500 kHz and 3000 kHz. Indeed, at low rotational speed of the fans, the turbulence intensity is low, speed fluctuations are low and a small number of particles passes through the measuring volume. Conversely, at high rotation speeds of the propellers, the fluctuations in speed are greater and the frequency signal acquisition is higher. In LDV, a statistical bias due to the random sampling of the velocity signals must be considered. Indeed, the number of high-speed particles passing through the measurement volume is greater than low speed particles' and the average speed, as well as higher moments of degree, are then biased towards high speeds. In [11], this bias is corrected by considering the transit time of the particles in the measurement volume, when calculating the average velocities $\bar{U}$ and $\bar{V}$, fluctuations u' and v' and moments of degree 3 and 4 (asymmetry and flattening of the velocity distributions). The LDV device is fixed on a crossbar allowing the 3D movements, in x, y and z directions (Figure 5.4). As with PIV measurements, for LDV measurements, the electrodes used for the ignition of the reactive mixture are removed.

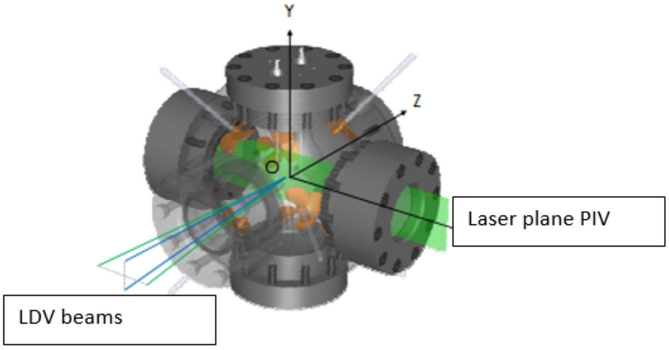


Figure 5.4: Representation of the LDV and PIV measurement configurations in the vessel

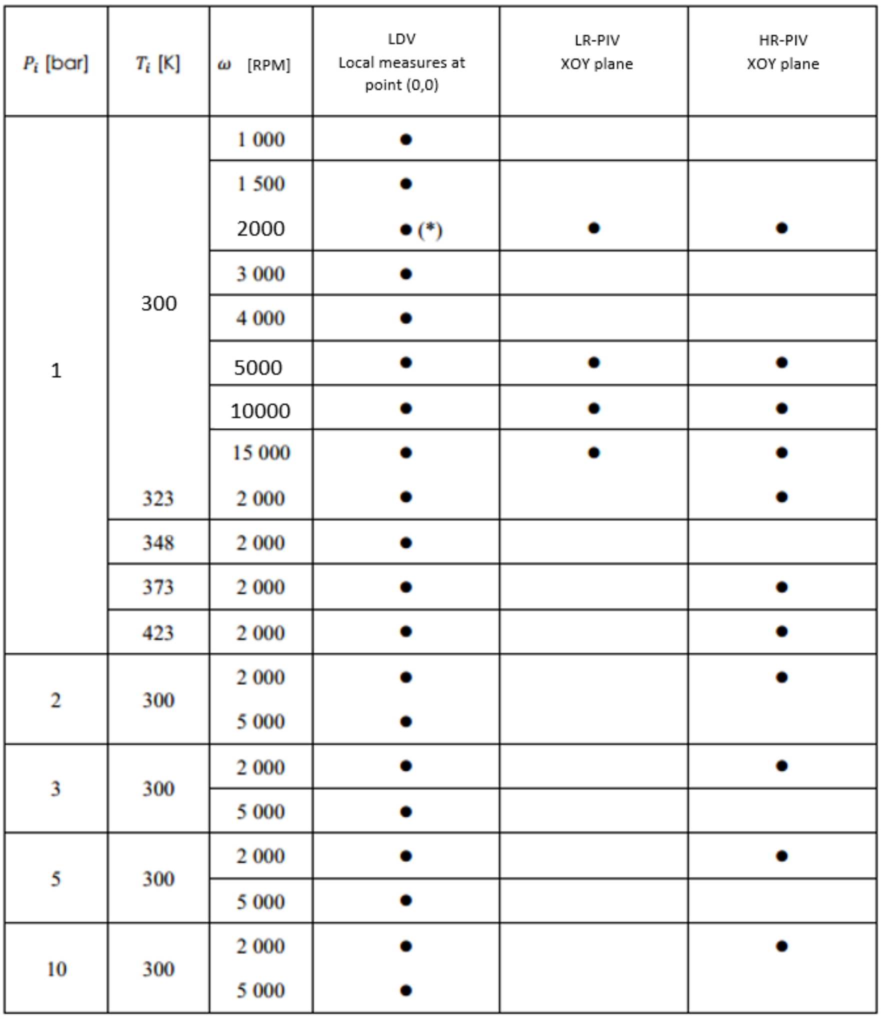
5.1.3 Measurement device: Particle Image Velocimeter PIV

The two speed components U and V can also be determined in the plane XOY (not intersecting any of the six fans), using a PIV device (Figure 5.4). To complete the information provided by the LDV measurements, two configurations of measures are adopted. The first uses the classic PIV techniques (low cadence, 2 Hz) and makes it possible to obtain 2D fields of the two speed components covering a wide viewing area (about 80 × 80 mm^2). This configuration will be called LR-PIV for Low Spatial and Temporal Resolution. The second uses the time-resolved PIV and allows obtaining velocity fields in a reduced area (about 20.5 × 20.5 mm^2)

associated with a high temporal resolution (3500 Hz). This configuration will be called HR-PIV, for High Spatial and Temporal Resolution.

5.1.4 Characterized experimental conditions

The turbulence parameters are characterized for different fans rotational speeds ω, different initial pressures and temperatures P_i and T_i. Conditions are presented in Figure 5.5. To compare the measurement techniques certain conditions are investigated by several techniques.



P_i [bar]	T_i [K]	ω [RPM]	LDV Local measures at point (0,0)	LR-PIV XOY plane	HR-PIV XOY plane
1	300	1 000	●		
		1 500	●		
		2000	● (*)	●	●
		3 000	●		
		4 000	●		
		5000	●	●	●
		10000	●	●	●
		15 000	●	●	●
	323	2 000	●		●
	348	2 000	●		
	373	2 000	●		●
	423	2 000	●		●
2	300	2 000	●		●
		5 000	●		
3	300	2 000	●		●
		5 000	●		
5	300	2 000	●		●
		5 000	●		
10	300	2 000	●		●
		5 000	●		

Figure 5.5: Experimental conditions for the characterization of turbulence in the vessel (* for this condition, the velocity is also measured at different points along the axes x, y and z

In this study, only three conditions at 1 bar and 300 K were simulated as shown in Table 5.1.

Test	Fan speed [rpm]	Mixture	Pressure [bar]	Temperature [K]
1	2000	Air	1	300
2	5000	Air	1	300
3	10000	Air	1	300

Table 5.1: Orleans test cases

This choice was made based on the large number of experimental data available for these three cases. The gas used in the simulation is simply air (20.9 % of oxygen and 79.1 % of nitrogen) instead of a mixture of air and propane because at 300 K and 1 bar the difference in density between only air or a mixture of air and propane is negligible . As a result, for characterizing the turbulence inside the vessel it is used pure air.

5.2 Orleans vessel mesh

The mesh used for the numerical analysis was the same for all tests in Table 5.2. It was created thanks to the following steps:

1. Creation of a rather coarse cubic mesh of 280 mm side, with hexahedral cells of 9 mm size.

2. Extraction of the final mesh by keeping only the cells inside the actual vessel geometry reproduced in detail by a STereoLithography (STL) file.

3. Addition, with respect to the basic mesh, of:

   - two levels of refinement at vessel walls and fan shafts, with the purpose to gain a satisfactory description of the whole geometry;

   - three levels of refinement at the fans blades, to improve the flow resolution where the MRF is active.

4. Addition of four centred refinement spheres of 25, 17.5, 10 and 5 mm diameter, featured by a refinement level of 3, 4, 5 and 6, respectively. Purpose of these spheres is to improve the flow velocity and turbulence description in the zone where experimental results are available [11]. The resulting mesh has 780650 cells.

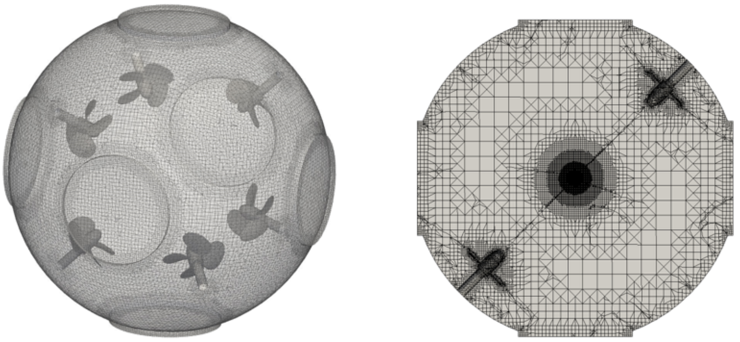


Figure 5.6: Orleans vessel mesh. Global view of the vessel (left) and mesh section (right), where it is possible to appreciate the adopted refinements.

The details of the mesh created are presented in Figure 5.6, from which it is possible to see a global view of the mesh and thanks to a section also the level of refinement is visible.

In this case, only one mesh was used for the non-reacting simulations, because the number of cells is quite high and the computational time becomes very long and not acceptable to obtain results in a pretty short time. The level of refinement at the fans blades is necessary, otherwise a large zone in which the Moving Reference Frame (MRF) approach (2.2) is active felt the rotation but they do not rotate causing an incorrect flow development. This consideration was observed in the early simulations of this numerical study when the cells sets generated include not only the cells attached to the fans blades and shafts and the one in-between the blades but also further cells in the direction normal to the fans blades/shafts. Such cell sets include parts of the mesh that do not rotate.

5.2.1 Numerical settings

The performed numerical analysis was carried out only on the three test conditions specified in Table 5.2. This choice was justified by two reasons:

1. The first is the larger quantity of experimental findings available for Tests 1, 2 and three than for other investigated setup.

2. The second is related to the further combustion investigation, which was performed by [7] at the fans speed range of 5000-7000 rpm. Hence, the range 2000-10000 rpm allowed a satisfactory turbulence analysis with reduced computational costs.

Numerical simulations were performed and the governing equations were solved with the RANS approach with the k-ε turbulence model.

Test	**Fans speed [RPM]**	**Mixture**	**Pressure [bar]**	**Temperature [K]**
1	2000	Air	1	300
2	5000	Air	1	300
3	10000	Air	1	300

Table 5.2: Investigated non-reacting test conditions selected from [11]

To handle the rotation of the fans, the Moving Reference Frame (MRF) approach was used, i.e. a CFD modelling technique used to simulate the rotating components.

Each test condition of Table 5.2 was first treated as a steady-state case because, at fixed fan speed, a constant recirculation flow rate was experimentally observed. Hence, incompressible steady-state simulations were initially performed with a first order discretisation accuracy, where the additional hypothesis of constant density is consistent with the low Mach number reached by the flow at a fan exit section. Then also steady-state simulations featured by a second order discretisation accuracy were performed to have better agreement between numerical results and experimental findings. Finally, to improve the experimental-numerical agreement, the steady-state simulation results were used as initial conditions for further compressible unsteady simulations featured by a second order discretisation accuracy. By averaging over time the investigated fields, the resulting steady-state solution was obtained.

5.2.2 Moving Reference Frame (MRF) implementation

The implementation of such modelling technique is divided into two main steps:

1. Creation of the cells sets in which activate the MRF technique (**topoSetDict**).

2. Assign the MRF to the created cells sets (**fvOptions**).

In OpenFOAM the **topoSetDict** dictionary is used to create six different cell sets, one for each fan with its shaft. In detail, each cell set includes three different regions:

1. One cylindrical cell set to include only the first cells attached to the fan shaft. The requirements are the starting and ending point for the cylinder axis and its radius.

2. One cell set containing the cells attached to the fan blades. This is possible using the *surfaceToCell* source that utilising the STereoLitography file of each fan specifies which cells need to be included in the cell set.

3. One cylindrical cell set that includes the cells in-between the fan blades. Also this cylindrical cell set requires the specification of the starting and ending point of the cylinder axis and its radius.

These three cell sets are added together to generate one cell set for each fan, so finally there are six cell sets. After the creation of the cell sets, there is the need to generate the relative cell zones that are used to assign the MRF approach to the cells that feel the rotation of the fans. Each cell set has a name (*cylinderFanX* where 1≤X≤6) that is used to create the relative cell zone. The cell sets created are groups of cells that need some specification: thanks to the **fvOptions** dictionary it is possible to activate the MRF approach. In particular, six different MRF sources are assigned to each cell zone created in the previous step. The *fvOptions* require 3 entries for (per) each MRF source:

1. Origin of the axis of rotation.
2. Axis of rotation directions (x-y-z).
3. Rotational speed in rad/s.

Finally, in Figure 5.7 the result of the described procedure is presented.

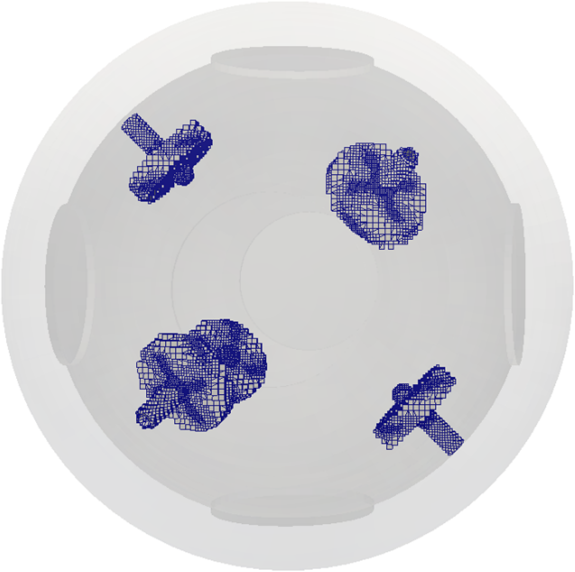


Figure 5.7: Detail of fans cell zones

After the set-up of these files, the utility **topoSet** is used to generate the cell sets and zones as well as to read the *fvOptions* dictionary activating the MRF on those cells.

5.3 Cold flow analysis

In this section, the experimental data available in [11] and [7] are compared with the numerical results obtained in this study. This comparison is necessary to have a correct numerical set up that gives the best agreement with the experimental finding having a correct flow initialisation regarding turbulence and flow velocity, for further combustion simulations. In the first part of this section some general considerations regarding the reference frame adopted are explained, then in the second part, the set up and results of steady state and incompressible simulations and of the unsteady and compressible simulations are presented.

5.3.1 Numerical data extrapolation procedure

A general procedure adopted for the post-processing part is here explained. The numerical data of the CFD simulations are available in a different frame of reference with respect to the one presented in [11]. The differences between the two reference systems are presented in Table 5.3 and graphically visible in Figure 5.8.

Experimental reference frame [11]	**Numerical reference frame**
x	-y
y	x
z	z

Table 5.3: Orleans reference frame differences

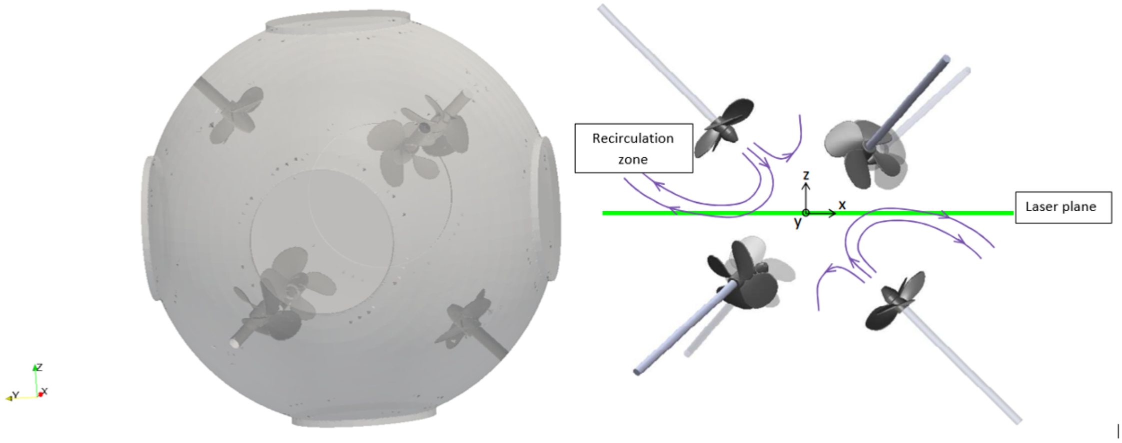


Figure 5.8: The numerical reference system (left) and the experimental one (right).

In [11], the central part of the vessel is the one in which the characterisation of the flow is important to have a correct flow initialisation because it is the ignition point for further combustion analysis. The numerical results are extrapolated in the central part of the vessel according to the experimental measurements. To extrapolate the data, a post-processing function, *plot over line* available in *Paraview* (the post-processing software of OpenFOAM), is used. This function requires both start and end points for drawing a line. Data are extrapolated from this line and the coordinates of the points for drawing the lines along the three directions and following the change of reference system, according to Table 5.3, are reported in Tables 5.4, 5.5 and 5.6.

Direction: x	x	y	z
Starting point	0	0.05	0
Ending point	0	-0.05	0

Table 5.4: Extrapolation point: x direction

Direction: y	x	y	z
Starting point	-0.05	0	0
Ending point	0.05	0	0

Table 5.5: Extrapolation point: y direction

Direction: z	x	y	z
Starting point	0	0	-0.05
Ending point	0	0	0.05

Table 5.6: Extrapolation point: z direction

The next Sections are devoted to the presentation of steady state and incompressible numerical results compared with the experimental findings. Finally, the unsteady state and compressible simulations results comparison is made.

5.3.2 Numerical results: incompressible steady-state simulations

A steady state and incompressible solver resolves the following equations:

- Continuity equation
- Momentum equation

without the time-derivative terms (steady-state) and imposing a constant density (incompressible). In this case, since the temperature is constant, the vessel walls do not exchange heat, and the enthalpy is constant, the energy equation is not solved. The steady state solver adopted is *simpleFoam* and is featured by an iterative process (SIMPLE algorithm explained in Chapter 2) that solves the previous equations until the residual between two consecutive iterations reaches a very low value (usually $10^{-7} \div 10^{-8}$). This means that a solution is reached when the residuals value is very small and does not oscillate.

The steady state solver requires some assignments as initial conditions and some physical properties of the working fluid to be assigned.

The value of pressure is assigned in the form of p/ρ and the air density at the tested conditions is calculated assuming the air composed only of nitrogen (79.1%) and oxygen (20.9%): from the air composition it is possible to calculate the molar mass of the air. As the universal constant of the gas R is 8314 $\frac{J}{K*kmol}$, the temperature is 300 K and the pressure is 10^5 Pa, knowing all these parameters and using the perfect gas law is possible to calculate the air density finding a value of ρ=1.15655 $[\frac{kg}{m^3}]$.

The kinematic viscosity is a physical property that needs to be assigned and to calculate this value at 1 bar and 300 K an interpolation procedure is implemented. For doing this task, different values of kinematic air viscosity at 1 bar and different temperature [2] are interpolated to find the value at 1 bar and 300 K. Table 5.7 reports the data used for the interpolation procedure found in [2], then using a Matlab code a second order polynomial interpolation is done. In Figure 5.9 the interpolated value is shown (red line) and is perfectly matched with the source value adopted. The value at 300 K is represented by the orange circle and it is equal to 1.564×10^{-5} $\frac{m^2}{s}$

Temperature [K]	**Kinematic viscosity** ν $\frac{m^2}{s}$
273.15	1.32×10^{-5}
283.15	1.41×10^{-5}
288.15	1.45×10^{-5}
293.15	1.50×10^{-5}
303.15	1.60×10^{-5}
313.15	1.69×10^{-5}
333.15	1.88×10^{-5}
353.15	2.09×10^{-5}
373.15	2.30×10^{-5}

Table 5.7: Air kinematic viscosity at different temperature and pressure equal to 1 bar

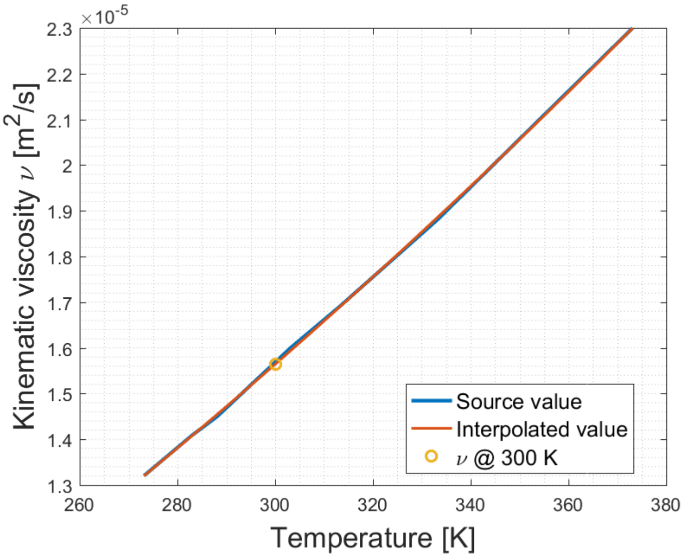


Figure 5.9: Interpolation of air kinematic viscosity data.

The interpolated kinematic viscosity value is used for all tests specified in Table 5.1 for the steady-incompressible and unsteady-compressible simulations.

At the beginning, inside the vessel the velocity field is null since the flow is at rest; on the wall for the no-slip condition (the flow is attached to solid body) the velocity is specified, and it is fixed to zero.

The turbulence variable k and ε are specified with the so-called wall function that allow saving plenty of cells near the walls of the vessel and near the fans body, because the boundary layer is approximated by model equations and it is not directly solved.

For what concern the numerical schemes, the first results were obtained with first order schemes to ensure stability, but the accuracy is penalized. In particular, after few attempts to reach stability the most important parameters to bring to a convergence solution are the under-relaxation factors for pressure and velocity. Under-relaxation limits the amount a variable changes from one iteration to the next, either by modifying the solution matrix and source before solving for a field or by modifying the field directly. An under-relaxation factor α, $0 \leq \alpha \leq 1$ specifies the under-relaxation amount, as described below.

- No specified α: no under-relaxation.
- α=1: guaranteed matrix diagonal equality/dominance.
- α decreases, under-relaxation increases.
- α=0: solution does not change with successive iterations.

The values of under relaxation factors for all the steady state cases are α=0.7 for pressure and α=0.3 for velocity. This choice is justified by the fact that the pressure inside the vessel is constant, while the velocity field is null everywhere and needs to be generated iteration after iteration. As a result, for pressure, there is no need to under relax the solution while for velocity the solution is reached slowly to avoid instability due to the change in velocity from one iteration to another. These values offer also a good compromise between computational time and agreement with the experimental results.

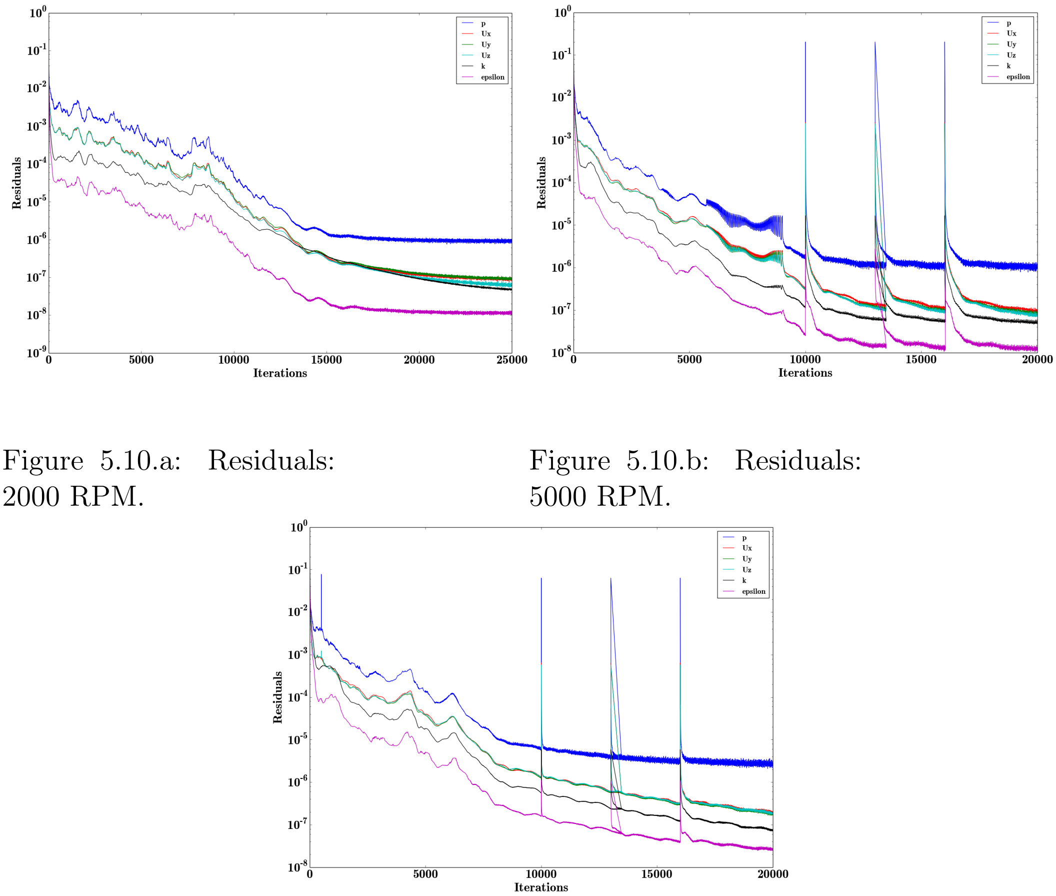


Figure 5.10.a: Residuals: 2000 RPM.

Figure 5.10.b: Residuals: 5000 RPM.

Figure 5.10.c: Residuals: 10000 RPM.

Figure 5.10: Residuals at different rotational speed.

The most important parameters in steady state simulations that need to be controlled are the residuals between consecutive iterations: when the residuals reach a value sufficiently low and stabilise or decrease continuously in a straight way it would be possible to observe a convergence solution.

From Figure 5.10 it is possible to see that the convergence with first order schemes is reached. An observation is made on the presence of different peaks that are due to some stops and restarts in the simulations and not for a stability issue of the CFD model.

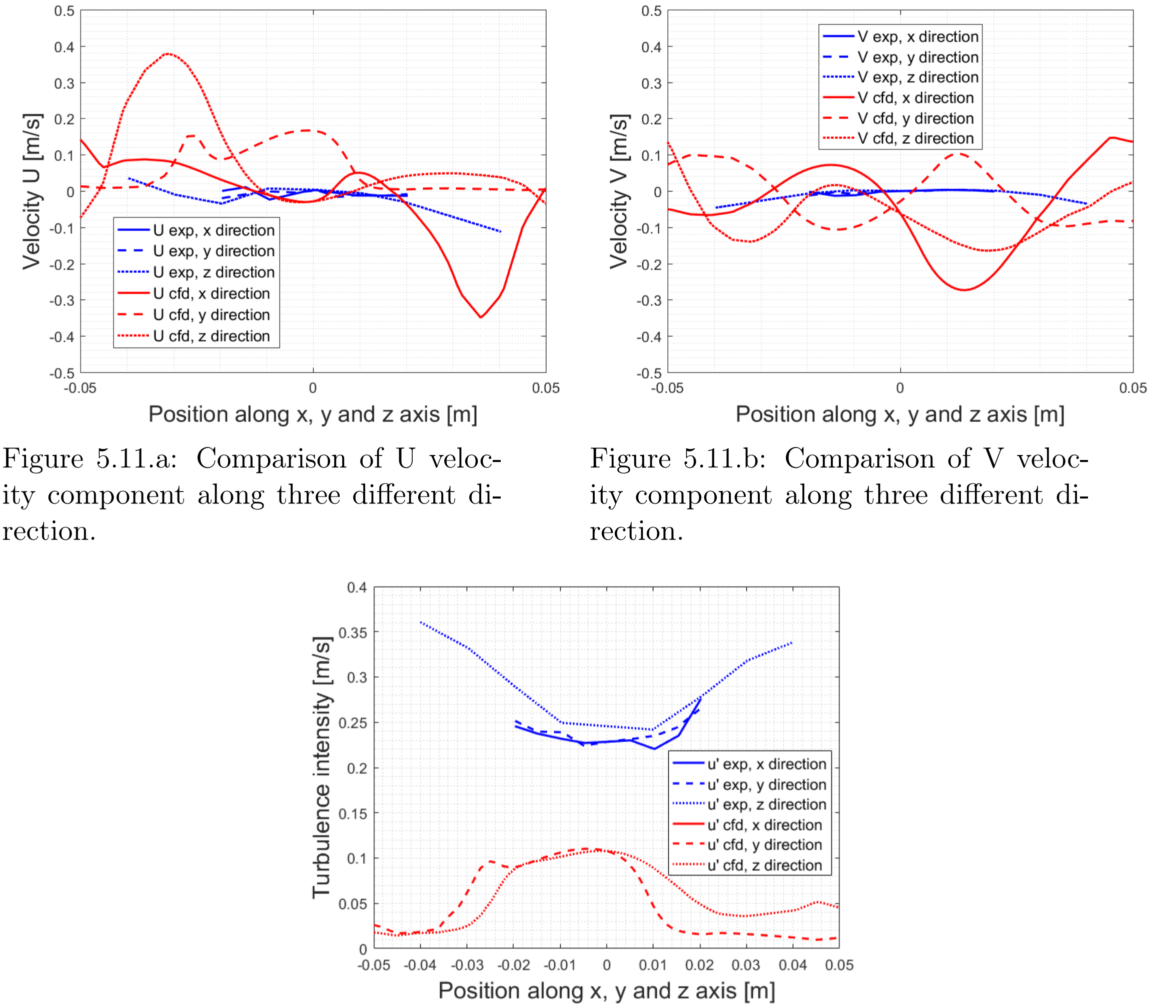


Figure 5.11.a: Comparison of U velocity component along three different direction.

Figure 5.11.b: Comparison of V velocity component along three different direction.

Figure 5.11.c: Comparison of u' along three different direction.

Figure 5.11: First order numerical schemes comparison.

Figure 5.11.a, Figure 5.11.b and Figure 5.11.c report the comparison between experimental data found in [11] and CFD results obtained from steady state and incompressible simulation with first order numerical schemes.

The comparison between velocity and turbulence intensity shows some differences between experimental findings and numerical results. A possible reason of these discrepancies could be the precision ensured by the first order numerical schemes, that is very stable but penalises accuracy, to increase theat aspect and have better agreement with the experimental findings, a second order numerical schemes simulations campaign was implemented. Increasing the order of numerical schemes means rising the accuracy of the results but this could involve numerical instability due to the higher order terms present in the approximation of the differential equations. The simulations featured by second order numerical schemes give a divergent solution considering residuals when starting from a situation not initialised. To avoid this situation, a possible solution is to take the solution reached by simulations featured by first order schemes and use it as initial condition for the ones featured by second order schemes. In this way, the simulation starts from a situation in which the differences in the flow quantity are not too large, and the residuals could remain low. This consideration is supported by the results presented in Figure 5.12 in which it is possible to see that the residuals initially decrease and then remain stable but too high in module.

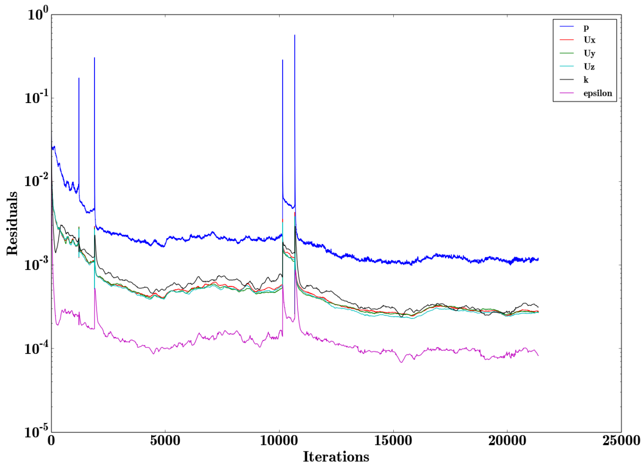


Figure 5.12: Residuals trend at fan speed equal to 2000 RPM for simulation featured by II order numerical schemes.

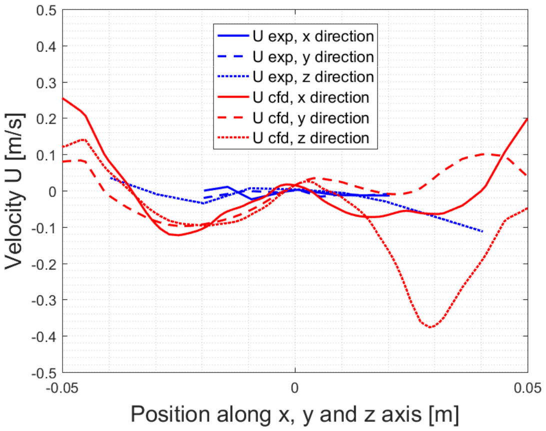


Figure 5.13.a: Comparison of U velocity component along three different directions for steady state simulation featured by II order numerical schemes.

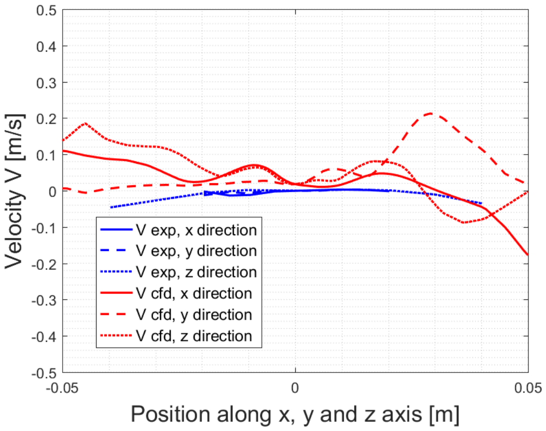


Figure 5.13.b: Comparison of V velocity component along three different direction for steady state simulation featured by II order numerical schemes.

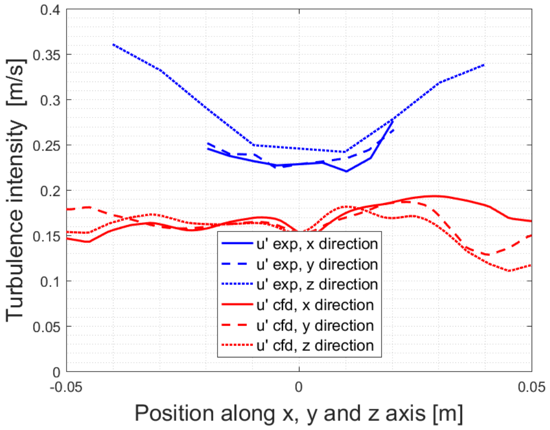


Figure 5.13.c: Comparison of u' along three different direction for steady state simulation featured by II order numerical schemes.

Figure 5.13: Second order numerical schemes comparison.

In Figure 5.13.a, Figure 5.13.b and Figure 5.13.c the comparison for the two velocity components and the turbulence intensity is made, second order numerical schemes were tested to try to obtain better residuals trend, but the convergence was not reached. As regards flow field, a comparison as for first order schemes is done.

Looking at the results of simulations with second order numerical schemes

(Figure 5.13) it is possible to see that the agreement with experimental data is better compared to first order ones.

The behaviour of the flow is correctly predicted by the numerical code, the velocity is almost zero in the centre of the vessel, in agreement with the experimental findings. The turbulence intensity shows a homogeneous zone in the central part and it is underestimated but the magnitude of the difference between the experiment and numerical u' is lower compared to the results obtained with first order numerical schemes. The choice to increase the accuracy gives the results expected, but as previously mentioned the residuals for this kind of numerical set-up are not perfectly satisfactory since they remain quite high in magnitude.

In the next paragraph, to try to obtain better correspondence between experimental and numerical results an unsteady and compressible simulation campaign was done.

5.3.3 Numerical results: compressible unsteady-state simulations

Unsteady and compressible simulations are featured by solving continuity, momentum and energy equations but the time derivative terms are present, and the density is not considered constant: consequently, the temperature needs to be specified as initial condition while the pressure is expressed as usual in Pascal and not as for the incompressible case.
The last iteration of the incompressible and steady state case was obtained by modifying the CFD code. The pressure for the incompressible case is divided by the density, but the compressible solver requires the static pressure: to this end, in the CFD code a function that calculate the static pressure is added. In summary, the procedure adopted for unsteady and compressible simulation is:

1. Take the last iteration of the steady state case for the different test case and use it as initial time step for the unsteady case.

2. Run the simulation until the velocity and turbulence fields remain almost constant between different time steps.

3. When fields do not significantly change, stop the simulation and restart from the last time step but using a different code that allows calculating the mean value of the interested fields to obtain mean value data that are to be compared with experimental results.

The test case 1 was run from the beginning because it takes too much time without reaching a solution that does not change too much between successive time steps.

Therefore, starting from the beginning, the flow field is generated avoiding long computational time.

Finally, all data are compared with the one present in [11] and are reported from Figure 5.14.a to Figure 5.16.b. The data sampled are compared with the numerical ones for 2000 RPM: the fluctuations of the component y of velocity (v') is practically the same as the x component (u'), showing the isotropy in turbulence, in accordance with [11]. For this reason and due the turbulence model adopted (isotropic) only u' is compared while the mean velocity components (U and V) are compared with the experimental data along the three axes of the reference system, according to the procedure explained in Section 5.3.1.

In Figure 5.14.a, Figure 5.14.b and Figure 5.14.a it is possible to see the comparison between experimental and numerical results for mean velocity components U and V and fluctuation u'.

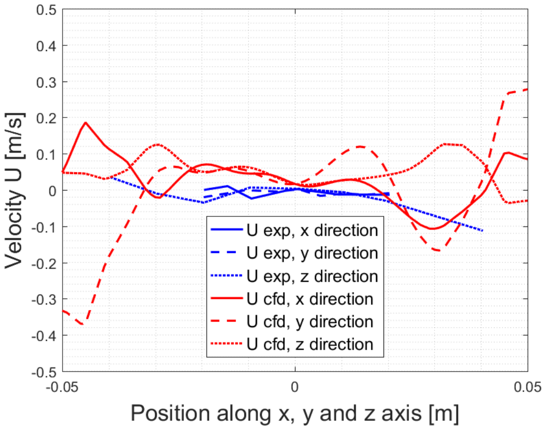


Figure 5.14.a: Comparison of U velocity component along three different direction for unsteady state simulation.

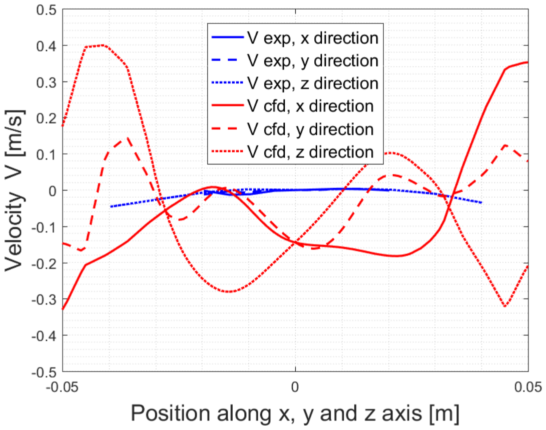


Figure 5.14.b: Comparison of V velocity component along three different direction for unsteady state simulation.

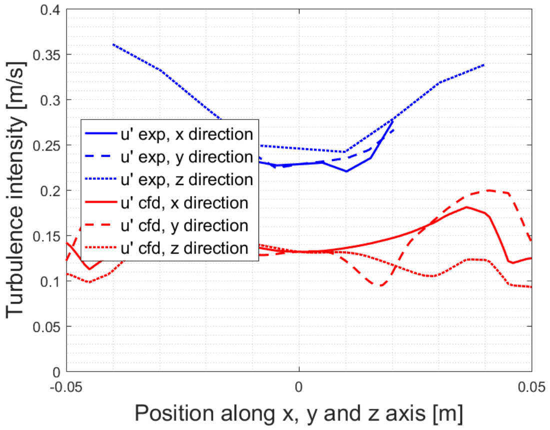


Figure 5.14.c: Comparison of u' along three different direction for unsteady state simulation.

Figure 5.14: Unsteady and compressible results comparison for 2000 RPM.

In Figure 5.14.a is possible to see that in the central part of the vessel the velocity is practically null and there is a homogeneous zone also in relation to turbulence (as shown in Figure 5.14.c).

The numerical results for the unsteady and compressible case present in Figure 5.14 show a better agreement with the experimental findings.

The numerical velocity component U is practically equal to the experimental trend and magnitude, while the component V oscillates a little bit but remain low in magnitude. The turbulence intensity is a little bit underestimated, probably this is due to the turbulence model adopted.

Experimental data are available only for test case 1, for the other two test cases, only numerical results are shown in the same form as the one for 2000 RPM.

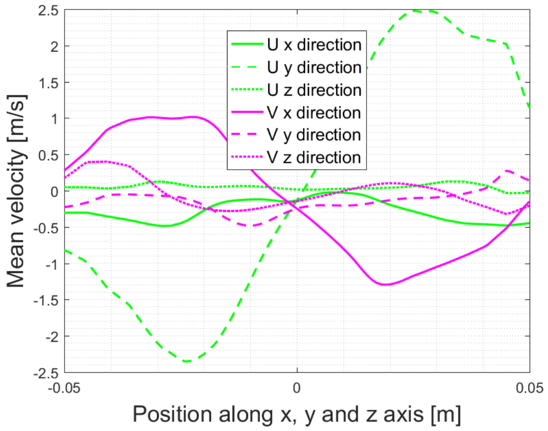


Figure 5.15.a: Mean velocity component U and V along three different direction for unsteady state simulation and fan speed equal to 5000 RPM.

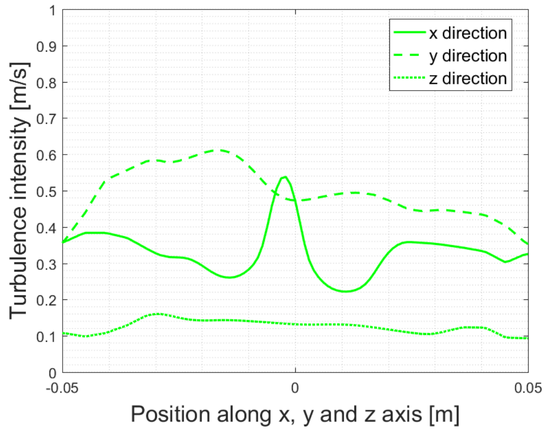


Figure 5.15.b: Comparison of u' along three different direction for unsteady state simulation and fan speed equal to 5000 RPM.

Figure 5.15: Unsteady and compressible results comparison for 5000 RPM.

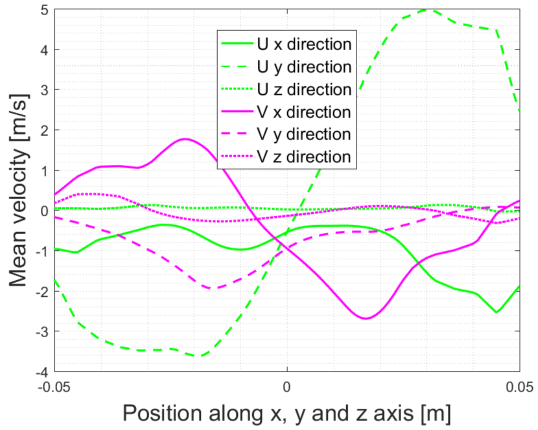


Figure 5.16.a: Mean velocity components U and V along three different directions for unsteady state simulation and fan speed equal to 10000 RPM.

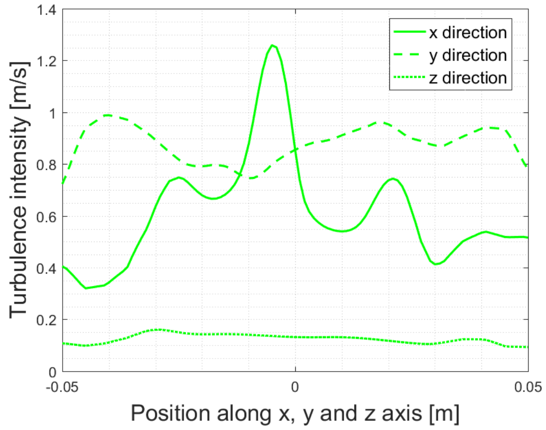


Figure 5.16.b: Comparison of u' along three different directions for unsteady state simulation and fan speed equal to 10000 RPM.

Figure 5.16: Unsteady and compressible results comparison for 10000 RPM.

The results at higher rotational speed are only for the numerical study but since the numerical model and set-up adopted at 2000 RPM is the same adopted for the study at higher rotational speed, it could be expected that the obtained results are in good agreement with the real conditions inside the vessel, thanks to the validation of the numerical model at 2000 RPM.

The numerical results at 5000 and 10000 RPM are reported in Figure 5.15 and Figure 5.16. Regarding the velocity fields the CFD results are in agreement with the experimental findings in [11]. The mean velocity components are almost constant in the central part of the vessel and this consideration remains valid also for the turbulence intensity, thus confirming the turbulence homogeneity in the central part of the vessel.

To support these considerations, it is possible to observe from Figure 5.15 and Figure 5.16 that the velocity components are almost null, this is due to the characteristic of the vessel able to generate a velocity field constant and practically at rest in central part.

Considerig the turbulence intensity, it is possible to see that increases with the rotational speed as it is expected, furthermore the turbulence intensity is homogeneous along the x direction, that coincide with the laser measurement plane, according to [11].

In conclusion the numerical results obtained are satisfactory and can be used for combustion analysis as done for the test engine in Chapter 4.

6. Conclusions

The work described in this thesis was mainly focused on the flow field initialisation for combustion analysis. For this purpose, two different devices able to generate particular flow condition in relation to generated turbulence and velocity field were investigated.

For the test engine, the focus was mainly on a correct flow initialization regarding velocity field and turbulence intensity. As to velocity field, very good agreement was found with experimental findings, while for the turbulence intensity the results are not in perfect agreement with experimental measurements. The turbulence intensity estimation was well captured compared to previous work adopting the k-ω SST turbulence instead of k-ε model. A big effort was done to further improve the turbulence intensity because it is well-known that combustion process is mainly influenced by this quantity: however, after many attempts, the best result obtained is the one presented in this work. Finally, combustion simulations were performed adopting an AKTIM ignition model for the flame kernel generation and a CFM model for the turbulent flame propagation. CFM model was recently modified adopting a different stretch factor formulation according to Sforza et al. [30]. The numerical results obtained seems to be not in agreement with the experimental ones, but this is due to the quite high overestimation in turbulence intensity that mainly affects the combustion process. The numerical model adopted for combustion analysis is able to correctly simulate the combustion process: as a proof of this, when we compare the effect of the ignition location and of the increasing air-fuel ratio, it is possible to see that also the numerical results are able to capture them (a fixed quantity of burnt mixture volume is present earlier after the ignition with increasing engine speed (increasin turbulence intensity), while the same burnt volume is present later with increasing air-fuel ratio (decreasing combustion velocity) as described by the experimental findings. Furthermore moving from the central to the peripheral ignition location the velocity increases producing a rise of combustion velocity that lead to a greater quantity of burnt volume gases after the ignition. All these observatio on the numerical results are supported by the experimetal findings.

For the Orleans vessel, a steady state solver was employed and the numerical results obtained were improved starting from simulations featured by first order and then by second order numerical schemes . The results are quite satisfactory compared to the experimental findings available from Galmiche [11], Brequigny [7], but to obtain the best possible agreement, even unsteady state simulations were carried out. The results obtained could be considered satisfactory, future improvements could be the use of a more refined mesh to verify if it will be possible to reach a convergent solution with low value of residuals with the steady state solver. Furthermore, combustion simulations could be performed to complete numerical and experimental comparisons.

As a future development of this work, it would be possible to reach a better agreement in terms of turbulence intensity by adopting new and improved turbulence models in relation to cold-flow simulation, while for the reacting case a comparison between a Lagrangian and Eulerian combustion model could be performed.

Concluding, this work shows how a very precise cold-flow characterization is important to obtain very good results when combustion analysis is performed.

Bibliography

[1] The rise of cfd in building internal combustion engines of tomorrow. URL http://www.hitechcfd.com/cfd-knowledgebase/the-rise-of-CFD-in-building-internal-combustion-engines-of-tomorrow.html.

[2] Kinematic viscosity of air. URL http://www.chegg.com/homework-help/convert-absolute-kinematic-viscosity-water-table-68-f-20-c-b-chapter-1.4-problem-3p-solution-9780134292380-exc.

[3] Moving reference frame for computational fluid dynamics. URL http://www.symscape.com/blog/moving-reference-frame-for-cfd.

[4] Openfoam user guide. URL https://cfd.direct/openfoam/user-guide/.

[5] Openfoam web site. URL https://openfoam.org/.

[6] D. Bradley, A. K. C. Lau, M. Lawes, and F. T. Smith. Flame stretch rate as a determinant of turbulent burning velocity. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 338(1650):359–387, 1992. ISSN 0962-8428. doi: 10.1098/rsta.1992.0012. URL http://rsta.royalsocietypublishing.org/content/338/1650/359.

[7] P. Brequigny. *Fuel influence on combustion in spark-ignition engine : flame stretch impact*. Theses, Université d'Orléans, Dec. 2014. URL https://tel.archives-ouvertes.fr/tel-01145273.

[8] C. R. Choi and K. Y. Huh. Development of a coherent flamelet model for a spark-ignited turbulent premixed flame in a closed vessel. *Combustion and Flame*, 114(3):336 – 348, 1998. ISSN 0010-2180. doi: https://doi.org/10.1016/S0010-2180(97)00194-6. URL http://www.sciencedirect.com/science/article/pii/S0010218097001946.

[9] G. Ferrari. *Internal combustion engines*. Società editrice ESCULAPIO, 2014.

[10] P. J. H. Ferziger and D. M. Perić. *COMPUTATIONAL METHODS FOR FLUID DYNAMICS*. Springer-Verlag, 2002.

[11] B. Galmiche. *Experimental characterization of expanding laminar and turbulent flames*. Theses, Université d'Orléans, Apr. 2014. URL `https://tel.archives-ouvertes.fr/tel-01069403`.

[12] W. M. H. Versteeg. *An introduction to computational fluid dynamics: the finite volume method*. Longman Scientific & Technical, 1995.

[13] R. Herweg and R. R. Maly. A fundamental model for flame kernel formation in s. i. engines. In *SAE Technical Paper*. SAE International, 10 1992. doi: 10.4271/922243. URL `http://doi.org/10.4271/922243`.

[14] R. Herweg, P. Begleris, A. Zettlitz, and G. Ziegler. Flow field effects on flame kernel formation in a spark-ignition engine. In *SAE Technical Paper*. SAE International, 10 1988. doi: 10.4271/881639. URL `http://doi.org/10.4271/881639`.

[15] G. Joulin and T. Mitani. Linear stability analysis of two-reactant flames. *Combustion and Flame*, 40(Supplement C):235 – 246, 1981. ISSN 0010-2180. doi: https://doi.org/10.1016/0010-2180(81)90127-9. URL `http://www.sciencedirect.com/science/article/pii/0010218081901279`.

[16] K. K. Kuo. *Principles of Combustion*. John Wiley, New York, 1986.

[17] A. Kösters, A. Karlsson, M. Oevermann, G. D'Errico, and T. Lucchini. Rans predictions of turbulent diffusion flames: comparison of a reactor and a flamelet combustion model to the well stirred approach. *Combustion theory and modelling*, 19:81–106. ISSN 1364-7830.

[18] T. Lucchini, G. D'Errico, H. Jasak, and Z. Tukovic. Automatic mesh motion with topological changes for engine simulation. In *SAE Technical Paper*. SAE International, 04 2007. doi: 10.4271/2007-01-0170. URL `https://doi.org/10.4271/2007-01-0170`.

[19] T. Lucchini, G. D'Errico, and M. Fiocco. Multi-dimensional modeling of gas exchange and fuel-air mixing processes in a direct-injection, gas fueled engine. In *SAE Technical Paper*. SAE International, 09 2011. doi: 10.4271/2011-24-0036. URL `https://doi.org/10.4271/2011-24-0036`.

[20] T. Lucchini, L. Cornolti, G. Montenegro, G. D'Errico, M. Fiocco, A. Teraji, and T. Shiraishi. A comprehensive model to predict the initial stage of combustion in si engines. In *SAE Technical Paper*. SAE International, 04

2013. doi: 10.4271/2013-01-1087. URL http://dx.doi.org/10.4271/2013-01-1087.

[21] T. Lucchini, M. Fiocco, A. Onorati, A. Montanaro, L. Allocca, P. Sementa, B. M. Vaglieco, and F. Catapano. Full-cycle cfd modeling of air/fuel mixing process in an optically accessible gdi engine. *SAE International Journal of Engines*, 6(2013-24-0024):1610–1625, 2013/9/8.

[22] A.-F. M. Mahrous, M. L. Wyszynski, T. Wilson, and H.-M. Xu. Computational fluid dynamics simulation of in-cylinder flows in a motored homogeneous charge compression ignition engine cylinder with variable negative valve overlapping. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 221(10):1295–1304, 2007. doi: 10.1243/09544070JAUTO355. URL https://doi.org/10.1243/09544070JAUTO355.

[23] D. M. Moukalled F., Mangani L. *The Finite Volume Method in Computational Fluid Dynamics.* Springer, 2015.

[24] T. M. Nguyen, F. Proch, I. Wlokas, and A. M. Kempf. Large eddy simulation of an internal combustion engine using an efficient immersed boundary technique. *Flow, Turbulence and Combustion*, 97(1):191–230, Jul 2016. ISSN 1573-1987. doi: 10.1007/s10494-015-9683-4. URL https://doi.org/10.1007/s10494-015-9683-4.

[25] D. R. D. Pijush K. Kundu, Ira M. Cohen. *Fluid Mechanics.* Elsevier Inc, 2012.

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

[27] S. Richard, O. Colin, O. Vermorel, A. Benkenida, C. Angelberger, and D. Veynante. Towards large eddy simulation of combustion in spark ignition engines. *Proceedings of the Combustion Institute*, 31(2):3059 – 3066, 2007. ISSN 1540-7489. doi: https://doi.org/10.1016/j.proci.2006.07.086. URL http://www.sciencedirect.com/science/article/pii/S1540748906001040.

[28] L. F. Richardson. *Weather prediction by numerical process.* Cambridge, University Press, 1922.

[29] P. K. Senecal, K. J. Richards, E. Pomraning, T. Yang, M. Z. Dai, R. M. McDavid, M. A. Patterson, S. Hou, and T. Shethaji. A new parallel cut-cell cartesian cfd code for rapid grid generation applied to in-cylinder diesel engine simulations. In *SAE Technical Paper*. SAE International, 04 2007.

doi: 10.4271/2007-01-0159. URL https://doi.org/10.4271/2007-01-0159.

[30] L. Sforza, T. Lucchini, A. Onorati, X. Zhu, and S.-Y. Lee. Modeling ignition and premixed combustion including flame stretch effects. In *SAE Technical Paper*. SAE International, 03 2017. doi: 10.4271/2017-01-0553. URL https://doi.org/10.4271/2017-01-0553.

[31] D. V. Thierry Poinsot. *Theoretical and Numerical Combustion*. R.T. Edwards, Inc., 2005.

[32] O. Vermorel, S. Richard, O. Colin, C. Angelberger, A. Benkenida, and D. Veynante. Multi-cycle les simulations of flow and combustion in a pfi si 4-valve production engine. In *SAE Technical Paper*. SAE International, 04 2007. doi: 10.4271/2007-01-0151. URL https://doi.org/10.4271/2007-01-0151.

[33] F. A. Williams. *Combustion theory*. The Benjamin/Cummings Publishing Company, Inc, 1985.

[34] V. Yakhot, S. A. Orszag, S. Thangam, T. B. Gatski, and C. G. Speziale. Development of turbulence models for shear flows by a double expansion technique. *Physics of Fluids A: Fluid Dynamics*, 4(7):1510–1520, 1992. doi: 10.1063/1.858424. URL http://dx.doi.org/10.1063/1.858424.

[35] X. Zhu, L. Sforza, T. Ranadive, A. Zhang, S.-Y. Lee, J. Naber, T. Lucchini, A. Onorati, M. Anbarasu, and Y. Zeng. Experimental and numerical study of flame kernel formation processes of propane-air mixture in a pressurized combustion vessel. *SAE Int. J. Engines*, 9:1494–1511, 04 2016. doi: 10.4271/2016-01-0696. URL https://doi.org/10.4271/2016-01-0696.