Università degli Studi di Padova

Dipartimento di Ingegneria Industriale DII

Department of Industrial Engineering

Corso di Laurea in Ingegneria Aerospaziale

Performance of CFD packages for flow simulations in aerospace applications

Relatore:

Prof.

Francesco Picano

Laureando:

Artur Antipca

1104473

Anno accademico 2016/2017

Contents

List of Figures

List of Tables

Symbols

If otherwise does not specified the used symbols are:

α	phase fraction in multiphase simulations or angle of attack in aerofoil simulations
BC	boundary conditions
C_d	drag coefficient
C_l	lift coefficient
C_{mu}	turbulence model constant by default equal to 0.09
η	Kolmogorov scale
ϵ	turbulent dissipation
e	energy
E	strain tesnor
Φ	Flux term
$\phi^{\diamond}$	an instantaneous arbitrary quantity ϕ
g	denotes the gas phase (continuous phase)
IC	initial conditions
l	denotes the liquid (discrete phase) or can denote a characteristic length scale
μ	dynamic viscosity
ν	kinematic viscosity
p	pressure
P	production term in kinetic energy equation
k	fluctuation specific kinetic energy
K	average specific kinetic energy (Reynolds averaged)
ρ	density
Re	Reynolds number
R	stress tensor
σ	Prandtl number or the turbulence dispersion model's constant
τ_η	characteristic time of Kolmogorov scale
U	velocity
$\overrightarrow{\overrightarrow{I}}$	identity matrix

$<>$	ensemble average operator in a single phase flow
$<>$	volume average operator in two phase flow
δ_{ij}	Kronecker delta
capital letter	Reynolds averaged quantity used in single phase flow
lower case letter	Turbulence fluctuation quantity used in single phase flow
$\tilde{\phi}$	intrinsic phasic average (or χ weighted average) of an arbitrary quantity
$\grave{\phi}$	arbitrary deviation quantity ϕ used in two phase flow
$\bar{\phi}$	time average of an arbitrary quantity ϕ used in two phase flow

Introduction

The design in aerospace field undergoes changes if it is compared with the last century. In origin the field of aerospace engineering did not disposes of computational tools existing today. In past systems were built to attempts, preforming a lot of experiments. Since the fluid mechanics can seldom be linearized it was difficult to perform exact calculus, except very simple systems. One of the tool that is capable to deal with non linearity is asymptotic theory. Asymptotic expansion theory was widely used in boundary - layer theory, sonic boom theory, multiple scale technique, Kolmogorov law $E \propto k^{-\frac{5}{3}-\mu}$ ... and it was developed after the First Word War.

To understand the physics that stays at the roots of fluid mechanics it is necessary to build up experiments and to observe the behaviour of the quantities that describe the physics of the problem. Even if the technological progress brought new tools not always these tools are suitable for research purpose. Experiments are usually limited by the device that is used for the observation of the physics. For example it is very difficult to view what happens inside the primary break up of liquids because the flow is extremely dense. And so other methods are required.

Computational fluid dynamics (CFD) is a branch of virtual experiments and it consist in simulation a physical behaviour of a fluid and it's interaction with other fluids or solids. CFD is a recent tool with powerful capability and under certain assumption can be considered a tool to investigate the reality As any tool its strength depends on the user competence in employing it. CFD changed its meaning over the years. Now engineers usually refer to the CFD as to a three-dimensional simulation but in origin it was more a monodimensional theory. As its name hints at, computational fluid dynamics uses the power of a computer to solve the equations that describe the fluid dynamics. From a theoretical point of view, the Navier Stokes equations are able to describe the real behaviour of fluids under external forces. And so if these equations are solved exactly the solution reveals the real behaviour of the system. To solve exactly the NS equations it is necessary to settle the problem at microscopical scale. In a such way it is possible to obtain an

exact solution to problem which concern general fluid dynamics of relatively small systems such as the motion of objects into fluids, interaction of fluids of different nature ... but also problems of a large scale as collapse of a nebula which leads to the formation of stars and planets. Solving a problem at a microscopical scale[1] requires an enormous quantity of computational power and memory. For this reason the exact solution can be obtained only for very small system; it depends on the computer in use but generally the solved domain have a total volume of order of cm^3. The simulations that give the exact solution are called Direct Numerical Simulations.

From an engineering point of view it is unacceptable to solve fluid dynamics problems with DNS (or LES), because it requires too much computational power and time. Since a fast solution is required the exact equations are drastically simplified and models are used to solve the problems. This equations are called Reynolds Averaged Navier Stokes equation. The main simplification in RANS solver is linked to the turbulence but this is not the only model used within. It is important to understand if the used models are able simulate the physical reality or not. Taking in consideration the simplicity of the model it is important to understand if it is accurate or not, it is important to understand the limitations of the models. The purpose of this thesis is to consider two CFD software applied to two different problems: aerofoil drag problem and liquid jet break up; to understand and to point out the limitations and the strength of the two solvers analysing the accuracy by comparing the results with experimental data.

[1] Kolmogorov scale

Chapter 1

Theory

In this chapter NS equations are presented and averaging operator is applied to obtain RANS equations. Two different cases are analysed. The first part of this chapter deals with Newtonian fluids in hypothesis of only one phase, for example air. The second part deals with Newtonian fluids but considers the case in which two phases are involved, for example gas and liquid. Because this study is focused on engineering applications only RANS equations are considered. Before proceeding it is useful to define a phase. The Fluent Users Services Centre provides the following definition: "A phase is a class of matter with a definable boundary and a particular dynamic response to the surrounding flow /potential field. Phases are generally identified by solid, liquid or gaseous states of matter but can also refer to other forms e.g. particles of different size.

1.1 Single phase averaged equations

The Navier Stokes equations are: continuity, momentum and energy equation. Since one of the main hypothesis is constant density only averaged continuity and momentum equations are deduced.

The **equation of mass continuity** is expressed as:

$$\frac{\partial \rho^{\diamond}}{\partial t} + \overrightarrow{\nabla} \cdot (\rho^{\diamond} \overrightarrow{u^{\diamond}}) = 0 \tag{1.1}$$

The equation of continuity describes the transport of mass. The first term is linked to the variation in time of mass while the second term is a flux term. The continuity equation can have also a source term; for example in a system with gas phase and liquid phase a source term is linked to the evaporation.

The second equation is **momentum equation**:

$$\frac{\partial \rho^{\diamond} \overrightarrow{u}^{\diamond}}{\partial t} + \overrightarrow{\nabla} \cdot (\rho^{\diamond} \overrightarrow{u}^{\diamond} \overrightarrow{u}^{\diamond}) = \rho^{\diamond} \overrightarrow{f}^{\diamond} + \overrightarrow{\nabla} \cdot \overrightarrow{\overrightarrow{\tau}}^{\diamond} \tag{1.2}$$

The first term on the left side is the variation of momentum in time, the second term on the left side is a flux term of momentum; the first term on the right side is linked to volume forces[1] while the second term on the right side is linked to the surface forces.

The last equation is the **equation of energy**. It is possible to distinguish between internal energy[2] $e^{\diamond}$ and the macroscopic energy $\frac{u^{\diamond}u^{\diamond}}{2}$. The total energy is $e^{\diamond} + \frac{u^{\diamond}u^{\diamond}}{2}$ and the energy transport equation is:

$$\begin{aligned} &\frac{\partial \rho^{\diamond}(e^{\diamond} + 0.5 \cdot u^{\diamond}u^{\diamond})}{\partial t} + \overrightarrow{\nabla} \cdot \rho^{\diamond}(e^{\diamond} + 0.5 \cdot u^{\diamond}u^{\diamond}) \overrightarrow{u}^{\diamond} = \rho^{\diamond} Q^{\diamond} - \overrightarrow{\nabla} \cdot \overrightarrow{q}^{\diamond} + \\ &+ \rho^{\diamond} \overrightarrow{u}^{\diamond} \cdot \overrightarrow{f}^{\diamond} + \overrightarrow{\nabla} \cdot (\overrightarrow{u}^{\diamond} \cdot \overrightarrow{\overrightarrow{\tau}}^{\diamond}) \end{aligned} \tag{1.3}$$

As previous the first term of the left side is the variation of energy in time, the second term is the flux term of energy; the first term on the right side is the volume heat, the second term is the surface heat and the last two terms are respectively the work done by volume and surface forces. All the terms inside equations 1.1, 1.2 and 1.3 are instantaneous quantities. The reason why it is necessary to transform instantaneous quantities in average quantities is explained in the next sub-section. To make the equation more elegant the tensor notation will be preferred to vector notation.

1.1.1 Ensemble-averaging

In computational fluid dynamics the instantaneous quantities, especially in flows with a high Reynolds number, need a huge computational memory to be stored. It is possible to reduce the required memory if only averaging quantities are stored. An average operation is like a low pass filter; a high frequency portion of the field is removed reducing the computational effort. From an engineering point of view it is an acceptable simplification. Almost always it is sufficient to describe the behaviour of a system statistically and not instantaneously. In one phase flow, speaking about statistical description is the same thing as speaking about average description. From the experimental analysis on fluids behaviour it is possible to conclude what follows:

[1] for example gravity

[2] for example atomic o chemical reactions

- real flows with high Reynolds number instantly are not symmetric, even if they are characterized in time by a symmetrical geometry and symmetrical boundary conditions
- the flows are characterized by a mean behaviour that preserves it's characteristics even if Reynlods number increases, and instantaneous fluctuation behaviour that is a function of Reynolds number.
- if the geometry is fixed but the initial conditions or boundary conditions change by a very small quantity the instantaneous behaviour of the flow is completely different[3]
- if the geometry is symmetrical and the boundary conditions are fixed the average behaviour of the flow does not change increasing the Reynolds number

Here ensemble average is defined as:

$$B = \lim_{N\to\infty} \frac{1}{N} \sum_{j=1}^{N} b_N \tag{1.4}$$

where N is the number of experiments performed to measure the quantity b_N. All the observed experiments are interpreted as an approximation of the ideal one. In this thesis the ensemble average operator will be indicated with $<>$. Notice that the ensemble average is not a spacial or temporal average but it is an average based on the number of experiments and the ensemble average is linked to the statistical behaviour of the measured quantity. It is not subject to spatial or temporal restriction. If the boundary condition don't change with time it is possible to prove that ensemble average is equal to time average.

The ensemble average operator $<>$ has the following property:

- *Linearity*: $< U + aV >=< U > +a < V >$ where a is a constant
- *Derivatives and average commute* $< \frac{\partial U}{\partial x} >= \frac{\partial < U >}{\partial x}$
- *Double average* $<< U >>=< U >$
- *Product average* $< U < V >>=< U >< V >$
- Attention: $< u^2 > \neq < u >^2$

[3]the theory of chaos can explain turbulence

1.1.2 RANS equations for incompressible Newtonian single phase flow

It is possible to write any quantity[4] of the flow as a contribution of a mean and fluctuation amount. To obtain the ensemble averaged equations from Navier-Stokes equation the exact quantity must be replace by the sum of fluctuation and average contribution. For simplicity an incompressible flow $\rho = constant$ is considered

$$u^{\diamond}(x,t) = U(x,t) + u(x,t) \tag{1.5}$$

where $U(x,t)$ is the average quantity while $u(x,t)$ is the fluctuation quantity.

1.1.3 Continuity transport equation

In hypothesis of constant density, substituting 1.5 within continuity equation (1.1) achieves:

$$\begin{aligned} < \frac{\partial \rho}{\partial t} > + < \frac{\partial \rho (U_j + u_j)}{\partial x_j} > &= \frac{\partial \rho}{\partial t} + \frac{\partial \rho < U_j >}{\partial x_j} + \frac{\partial \rho < u_j >}{\partial x_j} \\ &= \frac{\partial \rho}{\partial t} + \frac{\partial \rho U_j}{\partial x_j} \\ &= \frac{\partial U_j}{\partial x_j} \\ &= 0 \end{aligned} \tag{1.6}$$

The equation 1.6 is the averaged continuity equation and within there is not fluctuation quantities. If $\rho = constant$ from continuity equation

$$\frac{\partial u_j^{\diamond}}{\partial x_j} = 0$$

substituting the instantaneous velocity with average and floating velocity $u_j^{\diamond} = U_j + u_j$

$$\frac{\partial (U_j + u_j)}{\partial x_j} = \frac{\partial U_j}{\partial x_j} + \frac{\partial u_j}{\partial x_j} = 0$$

considering the result of equation 1.6 yields:

$$\frac{\partial u_j}{\partial x_j} = 0 \tag{1.7}$$

The equation 1.7 is an important result that will be used to determine the transport equation of macroscopic turbulence energy.

[4] such as pressure, velocity, density ...

1.1.4 Momentum transport equation

If a Newtonian fluid is considered it is possible to express the surface forces by:

$$\overrightarrow{\overrightarrow{\tau^{\diamond}}} = -p^{\diamond} \cdot \overrightarrow{\overrightarrow{I}} + 2\mu \overrightarrow{\overrightarrow{E^{\diamond}}} + \lambda(\overrightarrow{\nabla} \cdot \overrightarrow{u^{\diamond}}) \overrightarrow{\overrightarrow{I}} \tag{1.8}$$

where

$$\overrightarrow{\overrightarrow{E}} = \frac{1}{2}\left(\frac{\partial u_j}{\partial x_k} + \frac{\partial u_k}{\partial x_j}\right)$$

If the flow is incompressible, from the continuity equation emerges that:

$$\overrightarrow{\nabla} \cdot \overrightarrow{u} = \frac{\partial u_j}{\partial x_j} = 0$$

and so:

$$\tau^{\diamond}_{jk} = -p^{\diamond}\delta_{jk} + 2\mu E^{\diamond}_{jk}$$

The ensemble averaged momentum equation is:

$$< \frac{\partial \rho(U_k + u_k)}{\partial t} > + < \frac{\partial \rho(U_k + u_k)(U_j + u_j)}{\partial x_j} >=$$
$$- < \frac{\partial (P + p)}{\partial x_k} > + < \rho(F_k + f_k) > + < \frac{\partial (E_{jk} + e_{jk})}{\partial x_j} >=$$

which brings to:

$$\frac{\partial \rho U_k}{\partial t} + \frac{\partial \rho U_k U_j}{\partial x_j} = -\frac{\partial P}{\partial x_k} + \rho F_k + 2\mu \frac{\partial E_{jk}}{\partial x_j} - \frac{\partial \rho < u_j u_k >}{\partial x_j} \tag{1.9}$$

Examining 1.9 it is possible to observe the analogy with the exact Navier-Stokes equation; the difference between the two equations is a new term called *Reynolds stress* $< u_j u_k >$. This term is composed by fluctuation velocity. Till now the average quantities were introduced but averaged NS equation are still written in exact expression. Indeed the equation 1.9 depends on instantaneous velocity. An average operation usually can not free the transport equation from the instantaneous values but it can isolate the instantaneous contribute. Once the instantaneous term is isolated it is possible to model it. The *Reynolds stress* term that appears in 1.9 keeps the information about instantaneous velocity.

1.1.5 Kinetic energy

To model Reynolds stress it is important to understand its nature. Reynolds stress is a product of two velocities and so it is linked to the kinetic energy.

Now the equations of turbulence kinetic energy and average kinetic energy will be deduced. For clearness the turbulence kinetic energy is the energy related to the turbulence velocity while average kinetic energy is the energy of the mean flow.

Mean kinetic energy

The equation 1.9 can be rewritten as:

$$\frac{\partial U_k}{\partial t} + \frac{\partial U_k U_j}{\partial x_j} = -\frac{1}{\rho}\frac{\partial P}{\partial x_k} + \rho F_k + 2\nu\frac{\partial E_{jk}}{\partial x_j} - \frac{\partial < u_j u_k >}{\partial x_j} \tag{1.10}$$

it is possible to define the fluctuation specific kinetic energy as:

$$< u_j u_j >= 2k$$

and the specific averaged kinetic energy as:

$$U_j U_j = 2K$$

Multiplying the left and right part of the equation 1.10 by U_k and developing the simplifications brings to:

$$\frac{\partial U_k}{\partial t}U_k + \frac{\partial U_k U_j}{\partial x_j}U_k = -\frac{1}{\rho}\frac{\partial P}{\partial x_k}U_k + \rho F_k U_k + 2\nu\frac{\partial E_{jk}}{\partial x_j}U_k - \frac{\partial < u_j u_k >}{\partial x_j}U_k \tag{1.11}$$

Developing the equation 1.11 and considering the derivative properties, equation 1.6 and neglecting the volume forces yields:

$$\begin{aligned}\frac{\partial K}{\partial t} + U_j\frac{\partial K}{\partial x_j} =& -\frac{\partial}{\partial x_j}\left(U_j\frac{P}{\rho}\right) + 2\nu\left(\frac{\partial E_{jk}U_k}{\partial x_j} - E_{jk}E_{jk}\right) + \\ & - \frac{\partial U_k < u_j u_k >}{\partial x_j} + \frac{\partial U_k}{\partial x_j} < u_j u_k >\end{aligned}$$

In this equation it is possible to distinguish one term on the left side and three terms on the right side:

i. $\frac{\partial K}{\partial t} + U_j\frac{\partial K}{\partial x_j} = \frac{DK}{Dt}$

ii. $\Phi_K = -\frac{\partial}{\partial x_j}\left(U_j\frac{P}{\rho} - 2\nu E_{jk}U_k - U_k < u_j u_k >\right)$ is a term of flux composed by the contribution of pressure, viscosity and fluctuation velocity

iii. $\Sigma_K = 2\nu E_{jk}E_{jk}$ is a term of viscous dissipation; from the dimensional analysis it infers that increasing the Reynolds number the viscous dissipation vanish

iv. $P_K =< u_j u_k > \frac{\partial U_k}{\partial x_j}$ is a production term, it is possible to show that its rate is negative hence it extract the energy from the average fluid field

The equation 1.11 can be rewritten as:

$$\frac{D(K)}{Dt} = \overrightarrow{\nabla} \cdot \overrightarrow{\Phi_K} + P_K - \Sigma \tag{1.12}$$

Transport equation for turbulence kinetic energy

It is possible to obtain a similar equation for turbulence kinetic energy. Considering equation 1.2, substituting within equation 1.8 and the decomposition of instantaneous quantity in mean and fluctuation contribution, from the resulted equation subtract equation 1.10, expanding the terms taking in mind the equation 1.7, multiplying the obtained equation by turbulent velocity u_j and applying the average equation it is possible to write the next expression:

$$\frac{\partial k}{\partial t} + U_k \frac{\partial k}{\partial x_k} =< -\frac{1}{\rho}\frac{\partial < (pu_j) >}{\partial x_j} + 2\nu \frac{\partial < u_j e_{jk} >}{\partial x_k} - 2\nu < e_{jk} e_{jk} > + \\ - < u_j u_k > \frac{\partial U_j}{\partial x_k} - < \frac{1}{2}\frac{\partial u_k u_j u_j}{\partial x_k} >$$

As for mean kinetic energy it is possible to isolate three terms on the right side and one term on the left side and one term on the right side:

i. the term on the left side is

$$\frac{Dk}{Dt} = \frac{\partial k}{\partial t} + U_k \frac{\partial k}{\partial x_k}$$

ii. the first term on the right side is a flux term and is composed by a pressure, viscous and turbulence transport contribution

$$\Phi_k =< -\frac{1}{\rho}\frac{\partial < (pu_j) >}{\partial x_j} + 2\nu \frac{\partial < u_j e_{jk} >}{\partial x_k} - < \frac{1}{2}\frac{\partial u_k u_j u_j}{\partial x_k} >$$

iii. the second term on the right side is a deformation flux term

$$\epsilon_k = -2\nu < e_{jk} e_{jk} >$$

iv. the third and last term on the right side is a production term

$$P_k = - < u_j u_k > \frac{\partial U_j}{\partial x_k}$$

v. putting together all the parts mentioned in the last 4 passages the transport equation for the turbulent kinetic energy becomes

$$\frac{Dk}{Dt} = \overrightarrow{\nabla} \cdot \overrightarrow{\Phi_k} + P_k + \epsilon_k \tag{1.13}$$

Analysis of mean and turbulence kinetic energy

Let consider the two equations (1.13 and 1.12) previously derived. Both equations were written in a such way to have only one term on the left side and three terms on the right side.

A The terms $\frac{Dk}{Dt}$ and $\frac{DK}{Dt}$ are temporal and spacial variations of the kinetic energy. The variation of energy is influenced by the terms on the right side of equations 1.12 and 1.13

B The first term that produces a variation of energy is a flux term Φ. This term is composed by three contributions: *pressure, viscosity and fluctuation velocity.* Let compare for an instance Φ_K and Φ_k:

$$\Phi_K = -\frac{1}{\rho}\frac{\partial(PU_j)}{\partial x_j} + 2\nu\frac{\partial E_{jk}U_k}{\partial x_j} + \frac{\partial U_k < u_j u_k >}{\partial x_j}$$

$$\Phi_k = -\frac{1}{\rho}\frac{\partial < (pu_j) >}{\partial x_j} + 2\nu\frac{\partial < u_j e_{jk} >}{\partial x_k} - \frac{1}{2}\frac{\partial < u_k u_j u_j >}{\partial x_k}$$

the first two terms in Φ_k are correlated to average pressure and velocity. Since average pressure and velocity are unknown and calculated with equations 1.6 and 1.9 no model is needed. The last term of Φ_K is the Reynolds stress and as it was mentioned before, this term has to be modelled. The Φ_k is very similar to Φ_K as form but all his components are linked to floating quantities. In the field of RANS simulations these quantities are not unknowns of the problem and so have to be modelled.

C The second two terms are P_k and P_K

$$P_K = < u_j u_k > \frac{\partial U_j}{\partial x_k}$$

$$P_k = - < u_j u_k > \frac{\partial U_j}{\partial x_k}$$

As it can be noticed $P_K = -P_k$. The purpose of this term is to transfer the energy from the mean field to the turbulence field; $P_K = -P_k$ is the result of the internal friction inside in the mean flow. In particular, if the inertia of the particles is dominating than it is provable that $(< u_j u_k >) < 0$; the term P_K becomes negative and subtract the energy from the mean flow and since P_k becomes positive the subtracted energy is transferred to the fluctuating field. This term is especially important at high Reynolds number. Indeed if $Re \to \infty$ the mean field can not dissipate the energy by means of viscosity but the floating field can do it, the reason why it is possible is explained in the next point.

D The last term of are dissipation terms Σ_K and ϵ_k

$$\epsilon_k = -2\nu < e_{jk} e_{jk} >$$

$$\Sigma_k = -2\nu E_{jk} E_{jk}$$

In the dimensionless form 2ν is replaced by $\frac{1}{Re}$ and so if $Re \to \infty$ for sure $\Sigma_k \to 0$ because $E_{jk} = \frac{\partial U_j}{\partial x_k} + \frac{\partial U_k}{\partial x_j}$ and the spatial variation of the average velocity is to small to elide $Re \to \infty$; meanwhile $e_{jk} = \frac{\partial u_j}{\partial x_k} + \frac{\partial u_k}{\partial x_j}$, the high spatial turbulence velocity gradient can elide the effect of high Reynolds number and so $\frac{1}{Re} < e_{jk} e_{jk} >$ is not null, as consequence ϵ_k dissipate energy. If Re number is small (laminar case), Σ_K is not null and so the average field can dissipate energy. It is important to notice that ϵ_k is composed by the contribution of turbulence velocity $\overrightarrow{u}$ and so it must be modelled.

So considering a small volume (one cell of the mesh in RANS simulations) the energy can enter or exit throw the Φ_K term. Any cell of the domain has to dissipate some energy, if the domain il fully turbulent the average velocity can not dissipate in heat the macroscopic energy. The kinetic energy is transported from mean field to turbulent field where it is transformed in heat and so dissipates.

1.1.6 K-41 Theory

The mechanism of energy dissipation was proposed by Kolmogorov in 1941. The hypothesis of the theory are: universal, isotropic and homogeneous statistically distribution for small scales in high Reynolds number incompressible turbulence. The theory predicts a cascade mechanism that transport the energy from mean field (large scale) to the turbulence field (small scale) where the energy turns into heat.

Before proceeding it is necessary to point out that from an experimental point of view the behaviour of turbulence scale is not a function of the geometry. Indeed let consider a river, if an observant looks to a portion of area that is comparable with the river's width he will notice that the flow of water is influenced by the river's boards (geometrical boundary conditions). Let consider now the same river and imagine an observant that is inside the river and look to a very small portion of area; in this case the observant will notice that the flow is independent by the boarder of the the river. If this remark is true also for turbulence scale it infers the existence of an universal law of micro scales that can be applied to any case. Kolmogorov made an attempt to deduce this universal law.

The K-41 theory is based on two assumptions:

1 Two points $a(\overrightarrow{x})$ and $b(\overrightarrow{x}+\overrightarrow{r})$ are characterized respectively by velocity $\overrightarrow{u}(\overrightarrow{x},t)$ and $\overrightarrow{u}(\overrightarrow{x}+\overrightarrow{r},t)$ if the the vector $\overrightarrow{r}$ is small compared with macro-scales of the turbulence then the n-variate distribution of velocity difference $\overrightarrow{u}(\overrightarrow{x}+\overrightarrow{r},t)-\overrightarrow{u}(\overrightarrow{x},t)$ is an universal isotropic function solely of the difference vector $\overrightarrow{r}$, kinematic viscosity ν and mean rate of energy dissipation ϵ.

2 When $\overrightarrow{r}$ is large compared with the dissipation length scale the velocity difference is independent of ν.

Considering the second hypothesis $<\overrightarrow{u}(\overrightarrow{x}+\overrightarrow{r},t)-\overrightarrow{u}(\overrightarrow{x},t)>^n = f(\overrightarrow{r},\nu,\epsilon)$. Applying the dimensional analysis it is possible to state that the n-variate distribution of velocity differences in inertial range becomes:

$$u_r =<\overrightarrow{u}(\overrightarrow{x}+\overrightarrow{r},t)-\overrightarrow{u}(\overrightarrow{x},t)>^n = B_n(\epsilon r)^{\frac{1}{3}n} \tag{1.14}$$

where B_n is an universal constant. The equation 1.14 is independent from ν. Assuming $B_n = 1$ and $n = 1$ it is possible to express the Reynolds number of inertia rage as:

$$Re_r = \frac{u_r r}{\nu} = \frac{\epsilon^{\frac{1}{3}} r^{\frac{1}{3}} r}{\nu} = \frac{\epsilon^{\frac{1}{3}} r^{\frac{4}{3}}}{\nu}$$

so Re_r varies with $r^{\frac{4}{3}}$. The ϵ_r of the inertial scales can be expressed as:

$$\epsilon_r = \frac{{u_r}^2}{\tau_r} \quad \text{where } \tau_r = \frac{r}{u_r} \text{ is the characteristic time of scale } r$$

hence

$$\epsilon_r = \frac{{u_r}^2}{\tau_r} = \frac{{u_r}^3}{r} = \frac{(\epsilon r)^{\frac{3}{3}}}{r} = \epsilon$$

The macroscopic field is characterized by $\epsilon \approx \frac{U^3}{l}$, epsilon passes from the macroscopic field to microscopic field through the inertial range where it was previously showed that $\epsilon_r = \epsilon$ and so no dissipation occurs at inertial range.

Now let consider the dissipation range. With dimensional analysis it is possible to extract the wave length η of the dissipation range:

$$\eta = \left(\frac{\nu^3}{\epsilon}\right)^{\frac{1}{4}}$$

The length scale η of the dissipation range is called also Kolmogorov scale. In a similar way it is possible to extract the characteristic velocity u_η and characteristic time τ_η of the dissipation range:

$$u_\eta = (\epsilon\nu)^{\frac{1}{4}}$$

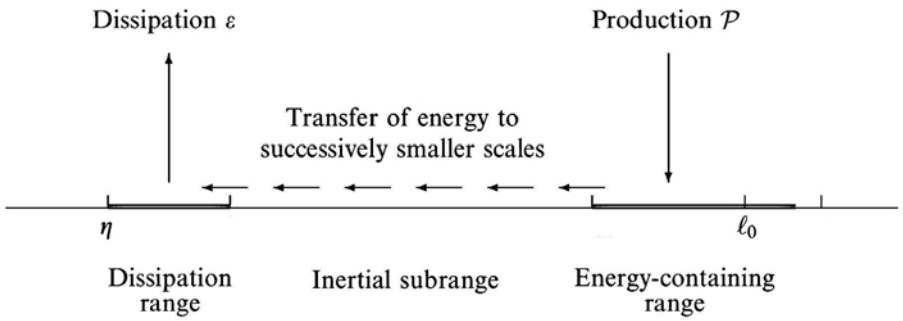


Figure 1.1: A schematic diagram of the energy cascade at high Reynolds number

and

$$\tau_\eta = \left(\frac{\nu}{\epsilon}\right)^{\frac{1}{2}}$$

Considering the previous equation for u_η it is possible to show that the characteristic Reynolds number of the dissipation range is of unity order, indeed:

$$Re_\eta = \frac{u_\eta \eta}{\nu} = \frac{\nu^{\frac{1}{4}}\epsilon^{\frac{1}{4}}\nu^{\frac{3}{4}}\epsilon^{-\frac{1}{4}}}{\nu} = \frac{\nu}{\nu} = 1$$

$Re_\eta = 1$ means in hypothesis of $K-41$ theory energy is dispersed ant the η scale. The ϵ that arrives at the dissipation scale is

$$\epsilon_\eta = \nu\frac{{u_\eta}^2}{\eta^2} = \nu\frac{\epsilon\nu^{\frac{2}{4}}}{\nu^{\frac{3}{2}}\epsilon^{-\frac{1}{2}}} = \epsilon = \epsilon_r$$

The previous equation explains why the $K-41$ theory is called cascade theory; the energy on time length scale is conserved through all the inertial ranges; so a cascade of energy is theorized from macro to micro scale where the energy is dissipated. The figure 1.1 is a schematic diagram of this mechanism. Qualitatively it implies the fission of whirls into smaller whirls till their dimension reaches the dimension η. This picture of cascade was described by Richardson in 1922:

> *Big whorls have little whorls, which feed on their velocity*
> *And little whorls have lesser whorls, and so on to viscosity.*

However this picture is in contrast with the experimental qualitative structure of high Reynolds number flows.

The K-41 theory is a kinetic theory and it is used to model the Reynolds Stress. The Reynolds stress is modelled later. It is important to understand

how the K-41 theory is linked to the Reynolds Stress. The specific turbulence kinetic energy is defined as $k = \frac{1}{2} < u^2 >$ and it is possible to link the energy spectrum to turbulence kinetic energy as follows:

$$k = \frac{1}{2} < u^2 >= \int_0^\infty E(N)dN \tag{1.15}$$

The quantity $E(N)$ is the scalar energy distribution function per mass unit for a certain wave number N. Kolmogorov used dimension analysis to link this quantity to the energy dissipation ϵ and wave number N. In the assumption that the spectrum in the dissipation range falls off rapidly enough with increasing the wave number, $E(N)$ is expressed as:

$$E(N) = C\epsilon^{\frac{2}{3}}N^{-\frac{5}{3}} \tag{1.16}$$

with C a constant. Equation 1.16 is called $-\frac{5}{3}$ law. Substituting equation 1.16 in equation 1.15 and integrating it from the wave number $N_l = \frac{2\pi}{l}$ and further supposing that the portion of the turbulent kinetic energy is proportional to the turbulent kinetic energy k yields:

$$l = \frac{C_u k^{\frac{2}{3}}}{\epsilon} \tag{1.17}$$

where C_u is a constant. The characteristic length l is used to model the Reynolds stress. The $-\frac{5}{3}$ law was found also in flow with small Reynolds number where there is not a clear distinction between inertial and dissipation range as well as in boundary layers where the hypothesis of isotropy is not confirmed. So in 1962 Kolmogorov changed the first hypothesis of $K-41$ theory and added a third hypothesis creating the $K-62$ theory. Nevertheless some problems of $K-41$ theory, it can't be logically disqualified, at the level of crude dimensional analysis it is a candidate such as 1962 theory.

1.1.7 The computational cost of DNS simulations

In the previous section it was deduced the dimension of Kolomogorov scale η. Direct Numeric Simulation solves the Navier Stokes equations on the scale η. It means that if the length of the domain is L_0 in x,y and z direction, for a DNS simulations the domain must be divided in $\frac{L_0}{\eta}$ elements in x,y and z directions. Considering that $\epsilon \propto \frac{{U_0}^3}{L_0}$, the total number of elements in one direction is:

$$N_1 = \frac{L_0}{\eta} = \frac{L_0\epsilon^{\frac{1}{4}}}{\nu^{\frac{3}{4}}} \propto \frac{{U_0}^{\frac{3}{4}}L_0}{{L_0}^{\frac{1}{4}}\nu^{\frac{3}{4}}} = Re^{\frac{3}{4}}$$

Because all DNS simulations are three-dimensional the total number of elements in the domain is:

$$N_{space} = {N_1}^3 \propto Re^{\frac{9}{4}}$$

Most of engineering applications are turbulent, so Reynolds number is of order 10^6 it means that the domain must be divided in $\propto 3 \cdot 10^{12}$ volumes. The temporal time step of DNS simulation can be expressed as:

$$N_t = \frac{T_0}{\tau_\eta} = \frac{T_0 \epsilon^{\frac{1}{2}}}{\nu^{\frac{1}{2}}} \quad \text{where } T_0 = \frac{L_0}{U_0}$$

thus

$$N_{temp} = \frac{T_0}{\tau_\eta} \propto \frac{L_0 {U_0}^{\frac{3}{2}}}{U_0 \nu^{\frac{1}{2}} {L_0}^{\frac{1}{2}} = Re^{\frac{1}{2}}}$$

It is possible to define computational cost as

$$C.C = N_{space} \cdot N_{temp} \propto Re^{\frac{11}{4}}$$

Such a high computational cost, considering relatively large systems, can not be supported by actual computers. And so DNS simulations impose a numerical restriction to the analysis. This is the reason why RANS simulations are used in engineering field. The computational cost of RANS is widely smaller. For example in general, in automotive simulations the domain is divided in circa $3 \cdot 10^6$ parts and it is usually used a steady flow model.

Besides numerical restrictions there is also a physical restriction. In DNS multiphase flow simulations the interface between phases is idealized to a surface over which the fluids properties change abruptly. This operation brings to intermittent structures with sharp discontinuities that in general are problematic. In cases of topological changes, as breakup, this idealization encounter several problems and the model becomes singular.

For the numerical and physical aspects previously mentioned the DNS simulations are restricted to a limited number of problems and are usually used for a basic understanding of physics. The Direct Numerical Simulations, for now, are disqualified as a possible solution for engineering problems.

1.1.8 Reynolds stress model

Once mean and fluctuating kinetic energy were introduced, the relation between the macro and micro scale was pointed out, the problem of closures of RANS equations is remained. The transport equation of momentum 1.9 has a term proportional to the fluctuating velocity called Reynolds stress

$2\rho < u_i u_j >$. The Reynolds stress is a tensor and so has 9 components that must be modelled.

Prandtl and Kolmogorov suggested to introduce a quantity named *eddy viscosity* and to parametrise it as the product of a characteristic length and a characteristic velocity.

$\nu_t = C_m l \cdot v$ where l was defined in equation 1.17 and C_m is a constant

Since $k^{\frac{1}{2}}$ is proportional to the characteristic velocity v, inserting this assumption in Kolmogorov-Prandtl relation and considering the equation 1.17:

$$\begin{aligned} \nu_t &= l \cdot v \\ &= l k^{\frac{1}{2}} \\ &= \frac{k^{\frac{1}{2}} C_m C_u k^{\frac{3}{2}}}{\epsilon} \\ &= C_d \frac{k^2}{\epsilon} \end{aligned} \tag{1.18}$$

so the turbulent viscosity correlates k and ϵ. If the particle's inertia is considered relevant the term $< \overrightarrow{u}\,\overrightarrow{u} >$ reduces the velocity gradient and therefore it could be modelled as a viscous term; this is the *Boussinesq's hypothesis* and it states that:

$$< u_j u_k >= \frac{2}{3} k \delta_{jk} - 2\nu_t E_{jk} \tag{1.19}$$

The last step is to give an equation for k and ϵ.

1.1.9 k ϵ or ω equations

Equation 1.13 is the exact equation for turbulent kinetic energy. In hypothesis of eddy viscosity theory the production term (P) and the dissipation term ϵ are written in close form. The flux term is still unknown. The hypothesis of gradient diffusion define a flux[5] of k down the gradient of k and so:

$$\overrightarrow{\nabla} \cdot \overrightarrow{\Phi_k} = -\frac{\nu_t}{\sigma_t} \overrightarrow{\nabla} k$$

where σ_k is the turbulent Prandtl number for kinetic energy. The final transport equation for k is:

$$\frac{Dk}{Dt} = \overrightarrow{\nabla} \cdot (\nu + \frac{\nu_t}{\sigma_k} \overrightarrow{\nabla} k) + C_{k1} P - C_{k2} \epsilon \tag{1.20}$$

[5] due to velocity and pressure fluctuations

The exact equation for ϵ is much complex than that of k. As showed in section 1.1.6 ϵ is the energy flow rate in the cascade determined by large-scale motion. For this reason the standard model equation for ϵ can be expressed empirically as

$$\frac{D\epsilon}{Dt} = \overrightarrow{\nabla} \cdot (\nu + \frac{\nu_t}{\sigma_\epsilon} \overrightarrow{\nabla} \epsilon) + C_{\epsilon 1} P \frac{\epsilon}{k} - C_{\epsilon 2} \epsilon \frac{\epsilon}{k} \tag{1.21}$$

where P is the rate of production

$$P_{jk} = 2\nu_t E_{jk} E_{jk}$$

while σ_k and σ_ϵ are respectively the Prandtl number that connects the diffusivity of k and ϵ to the eddy viscosity. In a such way the RANS equations are closed, indeed substituting the model for Reynolds Stress in equation 1.9 leads to:

$$\frac{D\overrightarrow{U}}{Dt} = \overrightarrow{\nabla} \cdot [-(\frac{P}{\rho} \overrightarrow{\overrightarrow{I}} + \frac{2k}{3}) \overrightarrow{\overrightarrow{I}} + 2(\mu + \mu_t) \overrightarrow{\overrightarrow{E}}]$$

if the mean pressure is defined as $P_m = \frac{P}{\rho} + \frac{2}{3}k$ it is possible to rewrite the previous equation in the following shape:

$$\frac{D\overrightarrow{U}}{Dt} = \overrightarrow{\nabla} \cdot [-P_m + \overrightarrow{\overrightarrow{I}} + 2(\nu + \nu_t) \overrightarrow{\overrightarrow{E}}] \tag{1.22}$$

Till now 5 constants where introduced and experimentally it was observe that for a wide number of application the suitable values are:

$$C_\mu = 0.09$$

$$\sigma_k = 1.0$$

$$\sigma_\epsilon = 1.3$$

$$C_{\epsilon 1} = 1.44$$

$$C_{\epsilon 2} = 1.92$$

The previous model is $k - \epsilon$ model and it is based on two equation, one for k and another one for ϵ.

A similar approach is used in $k - \omega$ model. In this model ω quantity is used instead of ϵ, ω is defined as:

$$\omega = \frac{\epsilon}{k}$$

The equation for ω is:

$$\frac{D\omega}{Dt} = \overrightarrow{\nabla} \cdot (\nu + \frac{\nu_t}{\sigma_\omega}\overrightarrow{\nabla}\omega) + C_{\omega 1} P \frac{\omega}{k} - C_{\omega 2}\omega^2 \tag{1.23}$$

where

$$\nu_t = C_{mu}\frac{k}{\omega}$$

Virtually the $k-\epsilon$ and $k-\omega$ models are identical (unless for the constants) but from numerical point of view it was seen that the two models have different behaviour. The main shortcomings of the two models are:

$k-\epsilon$ is characterized by the lack of sensitivity to adverse pressure gradients and overpredicts the shear-stress levels. It was supposed that this is due to the ovepredictions of the length scale in the near wall region. Also the model undergoes problems related to numerical stiffness of the equations when integrated through the viscous sublayer. A third trouble is related to the non zero value of ϵ at a no-slip surface.

$k-\omega$ the main inconvenient of this model is the strong dependency on the free stream values ω that are specified outside the shear-layer.

Menter proposed a solution to overcome the shortcomings of the two model. His solution consists in using $k-\epsilon$ and $k-\omega$ equations together but in different regions of the flow. This solution proved to be robust and not consuming excessive amount of computational time. In order to achieve his purpose Menter rewrote the $k-\epsilon$ model in terms of $k-\omega$ model. The k equation of the transformed $k-\epsilon$ model remains unchanged while the ϵ equation becomes:

$$\frac{D\omega}{Dt} = \overrightarrow{\nabla} \cdot (\nu + \frac{\nu_t}{\sigma_\omega}\overrightarrow{\nabla}\omega) + C_{\omega 1} P \frac{\omega}{k} - C_{\omega 2}\omega^2 + 2\frac{\omega_t}{\sigma_k}\overrightarrow{\nabla}\omega \cdot \overrightarrow{\nabla}k$$

so an extra cross diffusion term appears. The original $k-\omega$ model is multiplied by a function F_1 and the transformed model by a function $1-F_1$. The aim of the function F_1 is to activate the origin model near the solid surface ($F_1 = 1$) and the modified model away from the walls ($F_1 = 0$). The equations of the new model become:

$$\begin{aligned}
\frac{Dk}{Dt} &= \overrightarrow{\nabla} \cdot (\nu + \frac{\nu_t}{\sigma_k}\overrightarrow{\nabla}k) + C_{k1}P - C_{k2}\omega k \\
\frac{D\omega}{Dt} &= \overrightarrow{\nabla} \cdot (\nu + \frac{\nu_t}{\sigma_\omega}\overrightarrow{\nabla}\omega) + C_{\omega 1} P \frac{\omega}{k} - C_{\omega 2}\omega^2 + \\
&\quad + 2\frac{\omega_t}{\sigma_k}\overrightarrow{\nabla}\omega \cdot \overrightarrow{\nabla}k(1-F_1)
\end{aligned} \tag{1.24}$$

Once the equations of the model is adjusted the constants must be chosen. The process proposed is to match the constants of the original models. If A_1 is any constant of the origin $k-\omega$ model and A_2 is any constant of $k-\epsilon$ model and A the corresponding constant of the new model then:

$$A = A_1 F_1 + (1 - F_1) A_2$$

In order to close the model it is necessary to define the function F_1. The value of F_1 should go to zero near the boundary layer edge to guarantee the free-stream independence. It is superfluous in this work to derive the function F_1 however it is conceivable to write F_1 in terms of the turbulent length scale (equation 1.17) and the distance from the nearest surface, also $F_1 \neq 0$ must be guaranteed in the near surface region.

Reconsidering the equation 1.19 if $R_{ij} =< u_i u_j > -\frac{2}{3} k \delta_{ij}$ then:

$$R_{ij} = 2\nu_t E_{ij}$$

It is important to notice that Boussinesq's hypothesis imposes a linear relation between stress and strain tensor and it is analogous to the linear constitutive equation for Newtonian flows (see equation 1.8 and its simplifications). The proportionality of the two tensors can be tested if the tensors $\overrightarrow{\overrightarrow{R}}$ and $\overrightarrow{\overrightarrow{E}}$ are known and as was shown in "About Boussinesq's turbulent viscosity hypothesis: historical remarks and a direct evaluation of its validity" by Fran¸cois G Schmitt this parallelism is not verified in many applications, nevertheless Boussinesq's hypothesis is the heart of many turbulent models as $k-\epsilon$ who provides very satisfactory results. The inaccuracies of the models comes from transport equations and the linear constitutive equation based on kinetic energy theory.

1.1.10 Other models for Reynolds Stress

Previously the two equation model was presented, it is called two equation because it is based on the two transport equations: one for turbulent kinetic energy and one for turbulent dissipation. There are other possibilities to model the Reynolds stress. The first possibility is to link ν_t to mean velocity and to a length scale characteristic of the mean flow: $\nu_t = C_{zero} \Delta U b$. In the case of the jet flow ΔU can be the difference between the jet central-line velocity and the velocity of the external flow and length scale b can be the width of the jet. C_{zero} is a constant and varies from flow to flow, a typical value drops in the range of $[0.05, 0.1]$. The turbulent viscosity is constant across the shear layer. In the wall region of the turbulent boundary layer ν_t

must vary and different equations can be proposed. This model is called zero equation model because there is no need to write a transport equation for an added turbulent quantity.

The one equation model uses the Boussinesq's hypothesis as a constitutive equation and the turbulent viscosity is defined as $\nu_t = C_{one} q l$. The length l is prescribed by means of some shrewdness, a model equation for the turbulence velocity scale q is introduced after some manipulations of Navier-Stokes equations, the dissipation is modelled in terms of q and l.

Another alternative is to model every component of the Reynolds Stress tensor, or to introduce model with more than two equations; usually these models are complex and used in particular problems such as swirled flows because in such cases there are two direction with a viscosity gradient. It is interesting to note that the most aerospace combustion chambers work with axial or transverse injection. One of the reason can be the difficulty of swirled flow simulation. This is an example how the design can be influenced by the capability in simulating the system's physics.

1.2 Two phase averaged equations

Two phase flow is the simplest case of multiphase flow. Multiphase flow is defined as simultaneous flow of materials with different states or phases. In this case it is interested to the gas-liquid type of multiphase flow, particularly to the gas-droplet flows. Yet a universal multiphase model has not been invented. It is necessary to know a priori certain characteristics of the flow to chose which model to use. In this section the averaged equations for multiphase flow are deduced considering Euler model[6]. There are some difference between the RANS equations previously deduced and the averaged equation that are obtained in this section. In order to discriminate one phase from another a new quantity, in charge of fluid fraction, has to be defined. Two types of averages will be applied to the Navier Stokes equations. The first average operator applied to NS is the volume average; the second applied operator is the time average, its aim is to elide the temporal fluctuations caused by turbulence. The result of double averaging process is a set of governing equations that manifests smooth transition in temporal and spatial domain. How it was noticed in the previous section, an average operation applied on the momentum equation introduces new terms that must be modelled.

The two phase flow is analysed in the hypothesis of:

1 incompressible two-phase flow

2 steady-state condition.

3 no mass transfer between the two phases

4 Newtonian fluid

The letter "l" indicates the dispersed phase and "g" indicates the continuous phase; the letter "a" indicates an arbitrary phase. To avoid confusion with tensor notation the belonging of an arbitrary quantity ϕ to the phase a is indicated as ϕ^a. In what follow χ_a is a distribution that marks the phase a; the phase a is present in an arbitrary small portion of the domain if $\chi_a = 1$, otherwise $\chi_a = 0$. Usually χ_a is named *phase indicator function.* An arbitrary quantity ϕ is called conditioned quantity if it is multiplied by phase indicator function. The bounding condition in a two phase flow is:

$$\chi^g + \chi^l = 1 \tag{1.25}$$

This condition means that only one phase indicator needs to be considered in order to describe the phases within a two-phase system.

[6]another technique is VOF model

The quantity χ_a is related to the interface surface between the two phases. Hence to describe the temporal behaviour of χ_a it is sufficient to observe the interface surface which is moving with velocity $\hat{u}$, so also χ moves with $\hat{u}$. The interface transport equation is:

$$\frac{\partial \chi^a}{\partial t} + \overrightarrow{\hat{u}} \cdot \nabla \chi^a = 0 \tag{1.26}$$

1.2.1 Volume average

If χ^a is the phase indicator function and ϕ is a flow variable belonging to phase a, the average of the quantity ϕ over a volume is defined as:

$$< \chi^a \phi > \equiv < \phi^a > = \frac{1}{V} \int_V \chi^a \phi dV.$$

V is the averaging volume defined by an averaging scale l_Δ that is bigger than the length scale of microscopic fluctuations and smaller than the macroscopic characteristic length scale. Considering a volume V^a that contains the phase a it is possible to define the volume phase fraction as:

$$\alpha^a = \frac{V^a}{V} \quad \text{where } V = \sum V^a \text{ is the total averaging volume}$$

The intrinsic phase volume average is defined as:

$$\tilde{\phi}^a \equiv < \phi^a >^a \equiv \frac{1}{V^a} \int_{V^a} \phi^a dV = \frac{1}{V^a} \int_V \chi^a \phi dV = \frac{V}{V^a V} \int_V \chi^a \phi dV = \frac{< \chi^a \phi >}{\alpha^a} \tag{1.27}$$

Note that the average defined in equation 1.27 is an integral over the volume V^a that contains the phase a and also volume average make explicit the contribution of volume fraction; α_a is only a volume fraction if volume average is employed. The volume average has the same properties as Reynolds average and so linearity, derivatives and average commute, double average and product average. From the definition of volume fraction:

$$\alpha^a \equiv < \chi^a > = \frac{V}{V^a}$$

It is important to note that

$$< \chi^a \phi > \neq < \chi^a >< \phi > \quad \text{and so } < \chi^a \phi > \neq \alpha^a < \phi >$$

To obtain the volume averaged NS equations it is necessary to multiply the exact NS equations by χ^a in order to distinguish between the two phases; also

a random instantaneous quantity $\phi^{\diamond}$ (velocity for example) is decomposed in mean ($\tilde{\phi}$) and fluctuation ($\grave{\phi}$) quantity, it is multiplied by χ^a and volume averaged. Keeping in mind the definition of volume average (equation 1.27:

$$< \chi^a \phi >= \alpha^a \tilde{\phi}$$

as for ensemble average quantities the average of fluctuation part is zero $<\chi^a \grave{\phi} >= 0$.

1.2.2 Time average

The time average operation is indicated with a bar: $\overline{\phi}$ and it is defined as

$$\overline{\phi} \equiv \frac{1}{T} \int_T \phi dT.$$

The time averaging period T needs to be larger than the characteristic time scale of microscopic fluctuations and smaller than the characteristic time scale of the flow. The volume averaged quantity is decomposed in time mean ($\tilde{\overline{\phi}}$) and time fluctuation quantity ($\grave{\overline{\phi}}$). Applying time average operator to the volume average decomposition yields:

$$\overline{\alpha\tilde{\phi}} = \overline{\alpha}\tilde{\overline{\phi}}$$

and so

$$\tilde{\overline{\phi}} = \frac{\overline{\alpha\tilde{\phi}}}{\overline{\alpha}} \tag{1.28}$$

It is important to point out that:

$$\overline{\alpha\grave{\phi}} = 0 \quad \text{but } \overline{\grave{\phi}} \neq 0 \tag{1.29}$$

Indeed considering the equation 1.28 it emerges that:

$$\overline{\grave{\phi}} = \frac{\overline{\grave{\phi}\alpha}}{\overline{\alpha}} \tag{1.30}$$

where $\frac{\overline{\grave{\phi}\alpha}}{\overline{\alpha}}$ is a quantity related to time fluctuations and so has to be modelled. This aspect is reopened when drag force is modelled, because of equation 1.30 the drag force has an extra term.

1.2.3 Volume averaged equations

The first average equations are obtained multiplying the equations 1.1 and 1.2 by χ^a and applying the operator of volume average <>. Starting with the equation 1.2:

$$\left\langle \chi^a \frac{\partial \rho u_j^\diamond}{\partial t} \right\rangle + \left\langle \chi^a \frac{\partial \rho u_j^\diamond u_k^\diamond}{\partial x_k} \right\rangle = \left\langle \chi^a \rho f_j^\diamond \right\rangle + \left\langle \chi^a \frac{\partial \tau_{jk}^\diamond}{\partial x_k} \right\rangle$$

because ρ is a constant it is possible to pick it out from the average operation moreover considering the derivative properties and the equation 1.26 it is possible to conclude that:

$$\rho \frac{\partial}{\partial t} \left\langle \chi^a u_j^\diamond \right\rangle + \rho \left\langle \frac{\partial \chi^a u_j^\diamond u_k^\diamond}{\partial x_k} \right\rangle = \left\langle \frac{\partial \chi^a \tau_{jk}^\diamond}{\partial x_k} \right\rangle + \rho < \chi^a f_j^\diamond > + \left\langle (\rho u_j^\diamond (u_k^\diamond - \hat{u}_k^\diamond) - \tau_{jk}^\diamond) \frac{\partial \chi^a}{\partial x_k} \right\rangle$$

The term $\rho u_j^\diamond (u_k^\diamond - \hat{u}_k^\diamond) \frac{\partial \chi^a}{\partial x_k}$ represents the instantaneous mass transfer between the two phases and since the phase mass transfer is not consider this term is null. Considering the equation 1.27, the volume average momentum equation can be rewritten in terms of phase volume fraction as:

$$\rho \frac{\partial}{\partial t} \alpha^a (\tilde{u}_j)^a + \rho \frac{\partial < (\chi^a u_j^\diamond u_k^\diamond)^a >^a}{\partial x_k} = \frac{\partial \alpha^a (\tilde{\tau}_{jk})^a}{\partial x_k} + \rho \alpha^a (\tilde{f}_j)^a - \left\langle \tau_{jk}^\diamond \frac{\partial \chi^a}{\partial x_k} \right\rangle$$

As was done for one phase flow, the instantaneous velocity $u_j^\diamond$ is decomposed in two quantities: *volume average quantity* $\tilde{u}_j$ and *deviation quantity* $\grave{u}_j$

$$u_j^\diamond = \tilde{u}_j + \grave{u}_j$$

so substituting this term inside $< (\chi^a u_j^\diamond u_k^\diamond)^a >^a$ yields:

$$\begin{aligned} < (\chi^a u_j^\diamond u_k^\diamond)^a > &=< \chi^a (\tilde{u}_j + \grave{u}_j)^a (\tilde{u}_k + \grave{u}_k)^a > \\ &= (\tilde{u}_j \tilde{u}_k)^a + < (\chi^a \grave{u}_j \grave{u}_k)^a > \end{aligned}$$

$< (\chi^a \grave{u}_j \grave{u}_k)^a >$ is very similar to the Reynolds stress. It is appropriate to express this term in correlation with α^a. So let define the *pseudo turbulent Reynolds stress* as:

$$R_{jk}^{pt} = -\rho^a < (\chi^a \grave{u}_j \grave{u}_k)^a >^a$$

considering equation 1.27 it results that $\rho^a < (\chi^a \grave{u}_j \grave{u}_k)^a >= \alpha^a R_{jk}^{pt}$ it is called pseudo turbulent because other factors than turbulent can generate

$\grave{u}_j$. Substituting the last decomposition in the volume averaged momentum equation yields:

$$\rho^a \frac{\partial}{\partial t}\alpha^a(\tilde{u}_j)^a + \rho^a \frac{\partial \alpha^a(\tilde{u}_j\tilde{u}_k)^a}{\partial x_k} = \frac{\partial \alpha^a(\tilde{\tau}_{jk} + R^{pt}_{jk})^a}{\partial x_k} + \rho^a\alpha^a(\tilde{f}_j)^a - \left\langle \tau^\diamond_{jk}\frac{\partial \chi^a}{\partial x_k}\right\rangle \tag{1.31}$$

Proceeding in the same manner it is possible to obtain the volume average continuity equation; multiplying the equation 1.1 by χ_a and applying the volume average from:

$$\left\langle \chi^a \frac{\partial \rho^\diamond}{\partial t}\right\rangle + \left\langle \chi^a \frac{\partial \rho^\diamond u^\diamond_j}{\partial x_j}\right\rangle = 0$$

yields:

$$\rho_a \frac{\partial \alpha^a}{\partial t} + \rho^a \frac{\partial \alpha^a(\tilde{u}_j)^a}{\partial x_j} = 0 \tag{1.32}$$

With a single average approach it is not possible to derive an expression of Reynolds stress in which the volume fraction α is included inside the averaged operation. Single averaging does not produces terms involving gradients of α. Such terms may be modelled a posteriori to describe the forces observed in particular flow conditions but it is considered more satisfactory to derived these terms through a second average.

1.2.4 Time averaged equations

The equation 1.32 and 1.31 must be averaged for a second time applying the *time averaging operator* indicated by an over-bar. For simplicity will be omitted the index a, but it is important to remember that the following equations are written for only one phase.

$$\rho\frac{\partial}{\partial t}\overline{\alpha(\tilde{u}_j)} + \rho\frac{\partial\overline{\alpha(\tilde{u}_j\tilde{u}_k)}}{\partial x_k} = \frac{\partial\overline{\alpha(\tilde{\tau}_{jk}+R^{pt}_{jk})}}{\partial x_k} + \rho\overline{\alpha(\tilde{f}_j)} - \overline{\left\langle \tau^\diamond_{jk}\frac{\partial\chi}{\partial x_k}\right\rangle} \quad \text{Momentum}$$

$$\rho\frac{\partial\overline{\alpha}}{\partial t} + \rho\frac{\partial\overline{\alpha(\tilde{u}_j)}}{\partial x_j} = 0 \quad \text{Continuity}$$

Considering the decomposition of averaged velocity in α-average velocity plus fluctuation velocity

$$\tilde{u}_j = \tilde{\tilde{u}}_j + \grave{\tilde{u}}_j$$

and substituting it in the second term of the left side of momentum equation yields:

$$\begin{aligned}\overline{\alpha(\tilde{u}_j)(\tilde{u}_k)} &= \overline{\alpha(\tilde{\tilde{u}}_j + \grave{\tilde{u}}_j)(\tilde{\tilde{u}}_k + \grave{\tilde{u}}_k)} \\ &= (\tilde{\tilde{u}}_j\tilde{\tilde{u}}_k) + \overline{(\grave{\tilde{u}}_j\grave{\tilde{u}}_k)}\end{aligned}$$

At last, substituting the equation 1.28 in momentum and continuity equations it is obtained:

$$\rho\frac{\partial}{\partial t}\overline{\alpha}(\tilde{\tilde{u}}_j)+\rho\frac{\partial\overline{\alpha}(\tilde{\tilde{u}}_j\tilde{\tilde{u}}_k)}{\partial x_k}=\frac{\partial\overline{\alpha}(\tilde{\tilde{\tau}}_{jk}+\tilde{\tilde{R}}^{pt}_{jk}+R^t_{jk})}{\partial x_k}+\rho\overline{\alpha}(\tilde{\tilde{f}}_j)-\overline{\left\langle\tau^{\diamond}_{jk}\frac{\partial\chi}{\partial x_k}\right\rangle} \quad (1.33)$$

$$\rho\frac{\partial\overline{\alpha}}{\partial t}+\rho\frac{\partial\overline{\alpha}(\tilde{\tilde{u}}_j)}{\partial x_j}=0 \quad (1.34)$$

where $(R^t_{jk})=-\frac{\overline{\rho\alpha(\tilde{\tilde{u}}_j\tilde{\tilde{u}}_k)}}{\overline{\alpha}}$ is the α-weighted turbulent stress tensor.

1.2.5 Pressure and Viscous contribution

Let consider the stress term τ in the equation 1.33

$$\frac{\partial\overline{\alpha}(\tilde{\tilde{\tau}}_{jk})}{\partial x_k}-\overline{\left\langle\tau^{\diamond}_{jk}\frac{\partial\chi}{\partial x_k}\right\rangle}$$

and remember that τ is expressed by constitutive equation as a sum between a pressure and viscous contribution in the hypothesis of Newtonian fluids such that $\tau_{jk}=-p\delta_{jk}+R_{jk}$ where p is the pressure term and R is the viscous term. The next expression is achieved substituting the constitutive relation in the stress term

$$\frac{\partial\overline{\alpha}(\tilde{\tilde{\tau}}_{jk})}{\partial x_k}-\overline{\left\langle\tau_{jk}\frac{\partial\chi}{\partial x_k}\right\rangle}=-\frac{\partial\overline{\alpha}(\tilde{p})}{\partial x_j}+\overline{\left\langle p^{\diamond}\delta_{jk}\frac{\partial\chi}{\partial x_k}\right\rangle}+\frac{\partial\overline{\alpha}(\tilde{\tilde{R}}_{jk})}{\partial x_k}-\overline{\left\langle R^{\diamond}_{jk}\frac{\partial\chi}{\partial x_k}\right\rangle}$$

As was mentioned by Oliveira in his Phd thesis, the physical meaning of $\phi\frac{\partial\chi}{\partial x_j}$ is to 'pick' out the value of ϕ at the a-side of the interface (acting as a delta function) so $\left\langle\phi\frac{\partial\chi}{\partial x_j}\right\rangle=-\frac{1}{V}\int_{S^i}\phi n_j dA$ where n_j is the j component of the normal vector. Applying this observation to the pressure part of the stress term it is possible to express the interface contribution (integral over interface surface S^i):

$$-\overline{\alpha}\frac{\partial\tilde{\tilde{p}}}{\partial x_j}-\overline{\frac{1}{V}\int_{S^i}(p^{\diamond}-\tilde{\tilde{p}})n_j dA}$$

The pressure part of the stress term can be further transformed assuming $\tilde{\tilde{p}}$ equals the pressure of the continuous phase and decomposing the pressure on interface in mean and fluctuation quantity.

The viscous part of the stress tensor is more complicated to handle. Sometimes it is possible to neglect the viscosity. The difficulty to express the viscous terms rises from the choice to write or not the volume fraction inside the divergence terms. The stress term is expressed as:

$$\begin{aligned} &-\overline{\alpha}\frac{\partial\tilde{\tilde{p}}}{\partial x_j}+(\Delta p)\frac{\partial\overline{\alpha}}{\partial x_j}+\overline{F}_j^p \quad \text{pressure contribution} \\ &\overline{\alpha}\frac{(\tilde{\tilde{R}}_{jk})}{\partial x_k}-<\overline{(\dot{R}_{jk})}>\frac{\partial\overline{\alpha}}{\partial x_k}+\overline{F}_j^R \quad \text{viscous contribution} \end{aligned} \tag{1.35}$$

Only the terms $-\overline{\alpha}\frac{\partial\tilde{\tilde{p}}}{\partial x_j}$ and $\overline{\alpha}\frac{(\tilde{\tilde{R}}_{jk})}{\partial x_k}$ are kept and inserted in the momentum equation 1.33, the other terms contribute to create drag, virtual mass and lift forces that are modelled. It is important to mention that $\overline{F}_j^R$ and $\overline{F}_j^p$ appears in continuous phase with opposite sign with which them appears in the dispersed phase and usually these two forces are modelled as a unique drag term $\overline{F}_j^D$. The equation 1.36 is attained by replacing the stress term with it's viscous and pressure part inside the momentum equation 1.33

$$\begin{aligned} \rho\frac{\partial}{\partial t}\overline{\alpha}(\tilde{\tilde{u}}_j)+\rho\frac{\partial\overline{\alpha}(\tilde{\tilde{u}}_j\tilde{\tilde{u}}_k)}{\partial x_k}=&-\overline{\alpha}\frac{\partial\tilde{\tilde{p}}}{\partial x_j}+\overline{\alpha}\frac{\partial(\tilde{\tilde{R}}_{jk})}{\partial x_k}+\frac{\partial\overline{\alpha}(\tilde{\tilde{R}}_{jk}^{pt}+R_{jk}^t)}{\partial x_k}+ \\ &+\rho\overline{\alpha}(\tilde{\tilde{f}}_j)+\overline{F}_j^D \end{aligned} \tag{1.36}$$

1.2.6 $k-\epsilon$ equation

All the hypothesis that were made to model the turbulence for one continuous phase are still valid, so also for two phase flow the Kolmogorov's theory and Boussinesq's hypothesis are suitable. The big difference from one phase flow is that in two phase flow the dispersed an continuous phase interact one with each other. The turbulence of each phase is depicted by two equations, one for turbulence kinetic energy and one for dissipation. The equations, as it is expectable, are very similar to equations 1.21 and 1.20

$$\begin{aligned} \rho^a\left(\frac{\partial\overline{\alpha}^a k^a}{\partial t}+\frac{\partial(\overline{\alpha}\tilde{\tilde{u}}_j)^a k^a}{\partial x_j}\right)&=\frac{\partial\overline{\alpha}^a\frac{(\mu^t)^a}{\sigma_k}\frac{\partial k^a}{\partial x_j}}{\partial x_j}+\overline{\alpha}^a(G-\rho^a\epsilon^a) \\ \rho^a\left(\frac{\partial\overline{\alpha}^a\epsilon^a}{\partial t}+\frac{\partial(\overline{\alpha}\tilde{\tilde{u}}_j)^a\epsilon^a}{\partial x_j}\right)&=\frac{\partial\overline{\alpha}^a\frac{(\mu^t)^a}{\sigma_\epsilon}\frac{\partial\epsilon^a}{\partial x_j}}{\partial x_j}+\overline{\alpha}^a\frac{\epsilon^a}{k^a}(C_1G-C_2\rho^a\epsilon^a) \end{aligned} \tag{1.37}$$

where

$$G_{jk}=\mu^t\left(\frac{\partial\tilde{\tilde{u}}_j}{\partial x_k}\left(\frac{\partial\tilde{\tilde{u}}_j}{\partial x_k}+\frac{\partial\tilde{\tilde{u}}_k}{\partial x_j}\right)\right)^a$$

In the transport equations for k and ϵ, the parenthesis with index a $()^a$ indicates that the quantity belongs to the phase a. If these equations are applied to both phases, for example liquid $a = l$ and gas $a = g$ and if those equations are summed than[7]:

$$\left(\frac{\partial \overline{\rho}^m k}{\partial t} + \frac{\partial (\overline{\rho}\tilde{\tilde{u}}_j)^m k}{\partial x_j}\right) = \frac{\partial \frac{\overline{\mu}^{t,m}}{\sigma_k}\frac{\partial k}{\partial x_j}}{\partial x_j} + (G^m - C_2\overline{\rho}^m\epsilon) \tag{1.38}$$

$$\left(\frac{\partial \overline{\rho}^m \epsilon}{\partial t} + \frac{\partial (\overline{\rho}\tilde{\tilde{u}}_j)^m \epsilon}{\partial x_j}\right) = \frac{\partial \frac{\overline{\mu}^{t,m}}{\sigma_\epsilon}\frac{\partial \epsilon}{\partial x_j}}{\partial x_j} + \frac{\epsilon}{k}(C_1 G^m - C_2\overline{\rho}^m\epsilon) \tag{1.39}$$

with mixture characteristics:

$$\begin{aligned}
&\overline{\rho}^m = \overline{\alpha}^g\rho^g + \overline{\alpha}^l\rho^l \\
&\overline{\mu}^{t,m} = \overline{\alpha}^g\mu^{t,g} + \overline{\alpha}^l\mu^{t,l} = C_\mu\overline{\rho}^m\frac{k^2}{\epsilon} \\
&\nu^t = C_\mu\frac{k^2}{\epsilon} \\
&(\tilde{\tilde{u}}_j)^m = \frac{\overline{\alpha}^g(\rho\tilde{\tilde{u}}_j)^g + \overline{\alpha}^l(\rho\tilde{\tilde{u}}_j)^l}{\overline{\rho}^m} \\
&G^m = \overline{\rho}^g G^g + \overline{\alpha}^l G^l
\end{aligned} \tag{1.40}$$

The turbulent stress is modelled with the Boussinesq's hypothesis as was done in the one phase case (equation 1.19) Equations 1.39 and 1.38 are an extension of the single phase $k-\epsilon$ turbulence model to a mixture of the two phases.

1.2.7 Drag force and turbulence dispersion

In the equation 1.36 $\overline{F^D}$ is the drag force obtained by time averaging the interface drag force. It is important to mention that F^D was defined as a sum of a pressure contribute F^p and a viscous contribute F^R; the pressure force generates not only the drag but also contributes to the virtual mass and inviscid lift forces that are not considered here. Traditionally the drag force is modelled proportional to the instantaneous relative velocity which is the difference between the particles phase velocity and the continuous phase

[7] $\rho^a = constant$

velocity. Considering one of the three components of drag vector:

$$\begin{aligned}
\overline{F_j^D} &= F^D\overline{\alpha^l(\tilde{u}_j^l - \tilde{u}_j^g)} \\
&= F^D\overline{\alpha^l((\tilde{\tilde{u}}_j^l + \grave{u}_j^l) - (\tilde{\tilde{u}}_j^g + \grave{u}_j^g))} \quad \text{where was used } \tilde{u}_j = \grave{u}_j + \tilde{\tilde{u}}_j \\
&= F^D\overline{\alpha^l(\tilde{\tilde{u}}_j^l - \tilde{\tilde{u}}_j^g)} - F^D\overline{\alpha^l\grave{u}_j^g} \quad \text{because } \overline{\alpha^l\grave{u}_j^l} = 0 \text{ see equation 1.29} \\
&= F^D\overline{\alpha^l(\tilde{\tilde{u}}_j^l - \tilde{\tilde{u}}_j^g)} - F^D\overline{(1-\alpha_j^g)\grave{u}_j^g} \quad \text{bounding condition } \alpha^l + \alpha^g = 1 \\
&= F^D\overline{\alpha^l(\tilde{\tilde{u}}_j^l - \tilde{\tilde{u}}_j^g)} - F^D\overline{\grave{u}_j^g} \quad \text{because } \overline{\alpha^l\grave{u}^l} = 0 \\
&= F^D\overline{\alpha^l}(\tilde{\tilde{u}}_j^l - \tilde{\tilde{u}}_j^g) - F^D\overline{\grave{u}_j^g} \quad \text{see equation 1.28}
\end{aligned}$$

From equation 1.30:

$$\overline{\grave{u}}_j = \frac{\overline{\breve{\tilde{u}}_j\alpha}}{\overline{\alpha}}$$

where the term $\overline{\breve{\tilde{u}}\alpha}$ can be assumed proportional to the gradient diffusion of the volume fraction:

$$\overline{\breve{\tilde{u}}_j\alpha} = -\mu^g\frac{\partial\overline{\alpha}^g}{\partial x_j} \quad \text{therefore } \overline{\grave{u}}_j = -\frac{\mu^g}{\overline{\alpha^g}}\frac{\partial\overline{\alpha}^g}{\partial x_j}$$

Once it is substituted inside the drag force and considering $F^D = A^l\rho^l$ it is obtained that:

$$A^l\rho^l\overline{\alpha^l}(\tilde{\tilde{u}}_j^l - \tilde{\tilde{u}}_j^g) + A^l\rho^l\frac{\mu^g}{\overline{\alpha^g}}\frac{\partial\overline{\alpha}^g}{\partial x_j} \tag{1.41}$$

where A^l depends on the physics of the case. The equation 1.41 contains two terms for drag: the first term $A^l\rho^l\overline{\alpha^l}(\tilde{\tilde{u}}_j^l - \tilde{\tilde{u}}_j^g)$ is a mean drag contribution, the second term $A^l\rho^l\frac{\mu^g}{\overline{\alpha^g}}\frac{\partial\overline{\alpha}^g}{\partial x_j}$ is called *turbulent drag* or *turbulent dispersion* and is proportional to the fraction gradient which arises from turbulent fluctuation of the velocity. The turbulent dispersion can be modelled with different expressions as is presented in the chapter concerning the implementation of averaged Navier Stokes equations.

Chapter 2

Numerical approach

The chapter starts with a description of the Lagrangian model and presents the motivations why an Eulerian approach was used in the breakup simulations. Than it is given an outline about how the partial equations are implemented in CFD codes. And finally some information about wall functions, multiphase turbulence modelling and turbulence dispersion is provided.

2.1 Lagrangian approach

There are two possibilities to describe a phase flow. Euler approach concentrates the attention on what happens at a fix point in a space as time progresses. This approach is interesting in applications where the shape of body of fluid is not of interest and the attention is put on the rate at which change is taking place. Opposed to Euler description is the Lagrangian approach; every particle is followed as long as it moves in space and the change of physical properties is remarked. From the combinations of Euler and Lagrangian point of views come up: Lagrangian-Lagrangian model, Euler-Lagrangian model and Euler-Euler model. Usually the Eulerian description is related to the continuous phase and the Lagrangian description is related to the discrete phase.

In the Lagrangian model every individual droplets are considered. In one specie case each droplet is marked with a index and is characterized by a centre of gravity x, by a mass m and enthalpy h. One of the problem of the Lagrangian model is the amount of computational memory required for tracking every single droplet. To reduce the computational cost so called *small scale models* have to be implemented. These models make a huge gain in computational cost and a widely used also in DNS simulations. In a point

droplet approximation theory one of the main famous model is $D^2 - law$:

$$\frac{\partial D^2}{\partial t} = -8\frac{\rho_g}{\rho_l} D ln(B_m + 1)$$

where B_m is a function of mass fraction. This model describes the behaviour of the droplet diameter amenable to evaporation in a steady environment. Evaporation produces:

$$\dot{m}^k = \frac{\partial m^k}{\partial t} \leq 0$$

while heat diffusion and latent heat effects produces

$$\frac{\partial h^k}{\partial t} = \dot{h}^k.$$

The centre of gravity of every droplet evolves with droplet velocity U^k and so:

$$\frac{\partial^2 x_j^k}{\partial t^2} = \frac{\partial U_j^k}{\partial t} = \frac{1}{m^k} \int_\Gamma^k f_j^k dX - g\frac{\partial z}{\partial x_j}$$

where Γ is the interface and f_j^k is the droplet gas-momentum coupling (composed by a pressure and viscous contribution, see equation 1.35 of Euler approach). In RANS simulations only mean field quantities are solved and so the previous droplet evolution equations have to be written in terms of mean averaged quantities. In RANS simulations the main purpose is to understand the mean effect of droplets on the gas phase. Therefore the particles provides a statistical description of the dispersed phase and does not depict the picture of instantaneous state. Because of the statistical approach, CFD software uses a computational droplet[1] that carries a statistical weight of a set of particles with the same physical characteristics.

Since the two phase flow simulations analysed in this work deals with high phase fraction zones and in these zones the coupling between droplets and continuous phase can introduce numerical instability, since the number of the droplets in a parcel may have a considerable effect on some obtained statistics, since the statistical characteristics of individual droplet is not of interest, since the two-fluid model[2] by definition incorporates two-way coupling the Euler-Euler model is adopted in this study. Finally a Lagrangian description requires a 3D domain and can not be implemented with axial symmetry; if a fast solution is required Euler description is the best choice. It is important to emphasize that since the most inter-facial terms are non-linear the convergence of Euler model may be slow therefore it is recommended to use always a 2D geometry if it is possible.

[1] also called parcel

[2] Euler Euler model

2.2 Finite Volume discretisation

From the analysis of any equation previously derived it is possible to notice some important aspects. Let consider for example equation 1.10 :

$$\frac{\partial U_k}{\partial t} + \frac{\partial U_k U_j}{\partial x_j} = \frac{1}{\rho}\frac{\partial P}{\partial x_k} + \rho F_k + 2\nu\frac{\partial E_{jk}}{\partial x_j} - \frac{\partial < u_j u_k >}{\partial x_j}$$

The equation is composed by spatial $\frac{\partial}{\partial x_j}$ and temporal $\frac{\partial}{\partial t}$ operators, therefore two domains must be postulated: a spatial domain and a temporal domain. Each domain has to be separated in a finite number of elements.

A discretisation of spacial domain is called *mesh*. A mesh for CFD simulation is a geometry divided in a finite number of control volumes. Every cell (or control volume) is bounded by faces through which it send the information to adjacent elements. Every face is characterized by an area vector $\overrightarrow{s}$ normal to the face. Its magnitude is equal to the area A_s of the face. The face versor $\overrightarrow{n}$ is defined by $\frac{\overrightarrow{S}}{A_s}$. The flow variables are stored at the centre p of volume V_p whose position vector $\overrightarrow{x_p}$ satisfies $\int_{V_p}(\overrightarrow{x} - \overrightarrow{x_p})dV = 0$. This arrangement is called *collocated*[3]. The vector between two adjacent volumes V_{p1} and V_{p2} is $\overrightarrow{x_{p1}} - \overrightarrow{x_{p2}} = \overrightarrow{d}$. A mesh is called orthogonal when $\overrightarrow{d}$ is normal to the relative face for each face in the domain.

The discretisation of temporal domain involves to broke the time interval t in a finite number of time steps. The partition of the both domains in smaller elements can be uniform or not, it depends on the physical behaviour. Each equation has to be solved in each spatial and time element so an integration over volume and time is required. Usually the integration over a control volume is turned into an integration over the volume's boundary using Gauss theorem.

2.2.1 Volume integrations

Time derivative

The first term inside the 1.10 is a time derivative operator. In hypothesis of static mesh, assuming linear variation of U_k within a time step, the integral over a control volume V_p is:

$$\int_{V_p}\frac{\partial U_k}{\partial t}dV_p \approx \frac{U_k^n - U_k^o}{\Delta t}\Delta V_p$$

[3]Earlier, because difficulties with pressure velocity coupling, the velocities where stored at the cell vertices or faces. This choice is called *staggered* arrangement and is problematic in discontinuous boundary conditions

where U_k^n denotes the new value at the time step the calculation is computed while U_k^0 denotes the value from previous time step.

Convection term

The second term inside the 1.10 is a convection term. To compute this term Gauss theorem is used, the volume integral is transformed into a surface integral

$$\int_{V_p} \frac{\partial U_k U_j}{\partial x_j} dV_p = \int_S U_k U_j dS_k \approx \sum_f (U_k)_f (U_j)_f S_k = \sum_f F(U_j)_f$$

where S_k is the k element in the surface vector, $F = S_k(U_k)_f$ is the specific flux through the face f, $(U_j)_f$ is the interpolation of the quantity U_j on the face f

Gradient

The third term inside the equation 1.10 is the gradient term. Different methods exist to implement this operator. Considering the Gauss theorem:

$$\int_{V_p} \frac{\partial P}{\partial x_k} dV_p = \int_S dS_k P_f \approx \sum_f S_k P_f$$

where P_f is the interpolation of the pressure on the face f

Diffusion term

The diffusion term is treated as convection term, the term is converted in a surface integral

$$\begin{aligned}
\int_{V_p} \frac{\partial E_{jk}}{\partial x_j} dV_p &= \int_{V_p} \frac{1}{2} \frac{\partial}{\partial x_j} \left(\frac{\partial U_j}{\partial x_k} \right) + \frac{1}{2} \frac{\partial}{\partial x_j} \left(\frac{\partial U_k}{\partial x_j} \right) dV_p \\
&= \int_S \frac{1}{2} \left(\frac{\partial U_j}{\partial x_k} \right) dS_j + \frac{1}{2} \left(\frac{\partial U_k}{\partial x_j} \right) dS_j \\
&\approx \sum_f \frac{1}{2} S_j \left(\frac{\partial U_j}{\partial x_k} \right)_f + \frac{1}{2} S_j \left(\frac{\partial U_k}{\partial x_j} \right)_f
\end{aligned}$$

where $\left(\frac{\partial U_k}{\partial x_j} \right)_f$ is the gradient at the face f. The product $S_j \left(\frac{\partial U_k}{\partial x_j} \right)_f$ can be expressed as:

$$\left(S_j \frac{\partial U_k}{\partial x_j} \right)_f = \frac{1}{d} \left[\left(\frac{\partial U_k}{\partial x_j} \right)_P - \left(\frac{\partial U_k}{\partial x_j} \right)_N \right] A_s n_j$$

where d is the magnitude of the vector $\overrightarrow{d}$ between the centre of the cell of interest N and the adjacent cell P, $n_j = \frac{S_j}{A_s}$. If $\overrightarrow{d}$ is parallel to $\overrightarrow{S}$ the scheme is second order accurate. When this condition is not satisfied a non-orthogonal correction is added:

$$\left(\frac{\partial U_k}{\partial x_j}\right)_f S_j = B_1 \frac{1}{d}\left[\left(\frac{\partial U_k}{\partial x_j}\right)_P - \left(\frac{\partial U_k}{\partial x_j}\right)_N\right] A_s n_j + k_j \left(\frac{\partial U_k}{\partial x_j}\right)_f$$

where the constant B_1 and the vector k_j are determined by the non-orthogonality treatment.

2.2.2 Temporal discretisation

Once the integral over volume is solved, remains to consider the time integral. There are several alternatives to express temporal integral. However there are two possibilities to evaluate an arbitrary quantity ϕ_p: to evaluate it in terms of new values or in terms of old values.

Explicit finite difference scheme: the value at time n depends explicitly on the value at previous time 0, hence old value ϕ^0 is considered: $\phi_p = \phi_p^0$ and $\phi_f = \phi_f^0$. When non orthogonal mesh is considered, the diffusion term becomes: $\left(\frac{\partial \phi^o}{\partial x_j}\right)_f S_j = B_1 \frac{1}{d}\left[\left(\frac{\partial \phi^0}{\partial x_j}\right)_P - \left(\frac{\partial \phi^0}{\partial x_j}\right)_N\right] A_s n_j + k_j \left(\frac{\partial \phi_0}{\partial x_j}\right)_f$. It is first order accurate and it is unstable if the Courant number is greater than 1. If the velocity characteristic velocity is $\hat{U}$, the Courant number on the face is defined as:

$$C_o = \frac{\hat{U}_j S_j}{d_j S_j}\delta t$$

In openFoam, the solver *twoPhaseEulerFoam* chooses the characteristic velocity equal to the relative velocity between gas and liquid phases, however the software computes three different C_o: considering the velocity of the two phases and the relative velocity between phases.

Implicit finite difference scheme: the spatial derivatives are evaluated at the new time step: $\phi_p = \phi_p^n$ and $\phi_f = \phi_f^n$. For non orthogonal mesh the diffusion term becomes: $\left(\frac{\partial \phi^n}{\partial x_j}\right)_f S_j = B_1 \frac{1}{d}\left[\left(\frac{\partial \phi^0}{\partial x_j}\right)_P - \left(\frac{\partial \phi^n}{\partial x_j}\right)_N\right] A_s n_j + k_j \left(\frac{\partial \phi_0}{\partial x_j}\right)_f$. Because the non-orthogonal correction is treated explicitly this scheme guarantees the boundedness of the solution. The implicit scheme is unconditionally stable and this is a very significant to reduce the time of the simulation. For example the solver *twoPhaseEulerFoam* is build as an explicit, this means

that it is bounded by a low Currant number. However a semi-implicit scheme was implemented. The both schemes (explicit and implicit) are first order accurate in time.

2.2.3 System of linear Algebraic Equations

Reconsidering the momentum equation, neglecting the Reynolds stress and volume forces, counting the volume and time integral, adopting an implicit scheme the momentum equation becomes:

$$\frac{(U_k^n - U_k^o)_N}{\Delta t}\Delta V_p + \sum_f (U_k^n)_f (U_j^n)_f S_k = -\sum_f S_k \frac{P_f^n}{\rho} + \frac{1}{d}\left[\left(\frac{\partial U_k^n}{\partial x_j}\right)_P - \left(\frac{\partial U_k^n}{\partial x_j}\right)_N\right] A_s n_j + \\ + \frac{1}{d}\left[\left(\frac{\partial U_j^n}{\partial x_k}\right)_P - \left(\frac{\partial U_j^n}{\partial x_k}\right)_N\right] A_s n_j$$

where the quantities on the discrete volume face are a function of the volume of centre P or N according to the interpolation scheme. The equation can be rewritten in a generic form:

$$a_N U_N^n + \sum_P a_P \phi_P^n = R_N \tag{2.1}$$

The equation 2.1 is written considering the volume with centre N. The velocity U_N at time step n depends on the velocities of neighbour cells denoted with index P and on the explicit quantities (velocity at the previous time step) placed in source term R_N. Each cell of the domain requires an algebraic equation as equation 2.1, hence a system a generic system of algebraic equations can be developed;

$$[A][U] = [R]$$

The diagonal of the matrix $[A]$ is composed by a_N elements while a_P are off diagonal elements. $[U]$ is the vector of unknown velocities and $[R]$ is the source vector. The matrix $[A]$ can be expressed as a sum of diagonal (matrix $[D]$) and off-diagonal (matrix $[N]$) contribute:

$$[A] = [D] + [N]$$

Usually the matrix $[A]$ is solved using an iterative method because iterative methods are more economical and since the discretisation error is usually much larger than the accuracy of computer arithmetic the extreme tolerance of iterative method is not required. The convergence of the solver depends on

the dispersion of eigenvalues and can be improved with preconditioner. To guarantee the convergence, iterative solver requires a diagonal dominant matrix. It means that the magnitude of the diagonal elements is larger or equal than the sum of magnitudes of the off-diagonal elements: $|a_N| \geq \sum_P |a_P|$. The time derivative terms creates only diagonal and source coefficient hence this term increases the diagonal dominance. Indeed one technique to reach the convergence is to use transient solver instead of steady state solver. Another technique is to use under-relaxation factors, in this case the diagonal dominance is created through additional artificial terms. The equation 2.1 becomes:

$$a_N U_N^n + \frac{1-\lambda}{\lambda} a_N U_N^n + \sum_P a_P \phi_P^n = R_N + \frac{1-\lambda}{\lambda} a_N U_N^0$$

2.3 Turbulence treatment in multiphase simulations

The turbulence in multiphase simulations is solved using standard $k-\epsilon$ or $k-\omega$ models. There are some possibilities how to implement the models inside the software. The main difference between the implementations consists in writing one or two equations of k and ϵ (or ω). Once it was chosen how many equations have to be used, the turbulence parameters are defined as described in section 2.4. Fluent is provided by a third option for viscous treatment: Reynolds Stress, composed by 5 equations. However this model has not been considered in this thesis.

2.3.1 Mixture treatment

Turbulence mixture treatment is implemented writing only one equation for k and another one for ϵ. The equations are 1.38 and 1.39. The mixture properties are calculated using the 1.40 These equations are not referred to a particular phase but to the mixture. So the value of turbulence are imposed in mixture menu. In the menu of primary and secondary phase it is possible to chose the magnitude of the velocity, thermal conditions or other options according to the boundary condition. It is suggested to apply this model when phases separate in nearly stratified multiphase flow and when the density ratio between phases is close to one. In the break up simulations the ration between the densities is of order of 10. Since the cases does not refer to atmospheric pressure this model was not discarded.

2.3.2 Dispersed treatment

In this case the set of k and ϵ (or ω) equations are written for the primary phase. The equations are similar to equations 1.37 where the index a has to be substituted by g , gas continuous index phase. However equations 1.37 are not complete, indeed it misses the contribution of interphase turbulent momentum transfer. The menu of primary and secondary phase gives the possibility to chose other flow characteristics such as temperature, volume fraction ... according to the boundary conditions. It is suggested to use this model in a flow characterised by a dilute concentration of secondary phase.

2.3.3 Per phase

Per phase model is the most complex model because it is composed by two sets of turbulence equations. Each phase is characterised by it's own value of k and ϵ (or ω); the equations are similar to 1.37 but extra terms are added. The extra account the difference between the turbulent kinetic energy, the difference between the mean velocities and the volume fraction gradient of two phases.

2.4 Turbulence Boundary Conditions

The precision of the results are a function of the mesh and used models. Turbulence $k-\epsilon$ is the model that introduces the strongest approximation and so that influences in the major part the results. The boundary conditions for the turbulent kinetic energy and turbulence dissipation must be chosen realistic as much as possible. The wrong boundary conditions can bring to divergence or in the worst case to unrealistic results. Aerofoil and breakup simulations are characterized by inlet boundary, the values of k and ϵ on the inlet are of primary importance. It is possible to define such quantities through a user define function or a constant value can be imposed. The values can't be decided arbitrary but must be representative of the physics[4].

The turbulence in inlet can be defined in different modes, defining:

- turbulence intensity and turbulence length scale
- turbulence viscosity ratio and turbulence intensity
- k and ϵ

[4]it is possible to extrapolate the values for k and ϵ from experiments or empirical data

Turbulence intensity defined at the inlet is a function of the upstream conditions while the turbulence intensity defined on walls is 0 because of no slip conditions. Turbulence intensity is defined in terms of average velocity rate, a value smaller of 1% is considered a low value and characterized of flows that are slightly disturbed; 10% is a high intensity value. Turbulence intensity can be defined as:

$$I = \frac{u_{rms}}{u_{avg}}$$

with u_{rms} root mean square of fluctuation velocity and u_{avg} average velocity.

Tubulence length scale l is linked to the largest whirls inside the flow that hold the energy of turbulene flow. In the case of liquid jet breakup the turbulence length scale can be defined as a percentage of nozzle diameter.

Turbulence viscosity ratio $\frac{\mu_t}{\mu}$ is a function of turbulence Reynold number:

$$Re_t = \frac{k^2}{\epsilon\eta}$$

in general the turbulence viscosity ratio must be smaller than 10:

$$1 < \frac{\mu_t}{\mu} < 10$$

If this condition is not verified, Fluent and OpenFoam bounds the ratio[5].

The following expressions can be used to calculate the k, ϵ and ω on the inlet[6].

$$k = \frac{3}{2} u_{avg}^2 I^2 \tag{2.2}$$

$$\epsilon = C_\mu^{\frac{3}{4}} \frac{k^{\frac{3}{2}}}{l} \tag{2.3}$$

$$\omega = \frac{1}{C_{mu}^{0.25}} \frac{k^{0.5}}{l} \tag{2.4}$$

If the absence of further specification the Reynolds stress components on the inlet of the boundary are defined approximately as $< u_i u_j >= 0$ and $< u_i u_i = \frac{2}{3} k >$

[5]Fluent message informs that turbulence boundary ratio is bounded while OpenFoam bound k or ϵ

[6]this expression are valid for Fluent as for OpenFoam

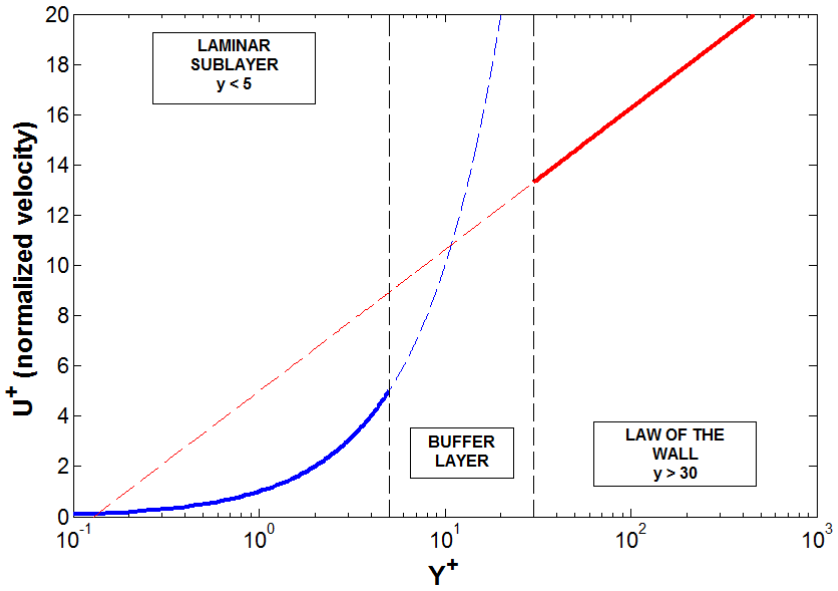


Figure 2.1: Composite regions of the turbulent boundary layer

2.5 Boundary layer and the importance of y^+ value

Considering external aerodynamics with the purpose to calculate force coefficient as C_l and C_d it is important to solve adequately the computational zone close to the wall of interested body. It means that the selected turbulence model, the created mesh and the way in which the boundary layer is solved are suitable for the results the user are looking for. The turbulent boundary layer is composed by three main regions as indicated in figure 2.1. If $y(1)$ is the distance of the first node above the wall, U_t is the wall parallel component of the velocity than the y^+ and U^+ are defined as:

$$y^+ = \frac{y(1)U_\tau\rho}{\mu} \quad \text{and } U^+ = \frac{U}{U_\tau} \text{ where } U_\tau = \left(\frac{\mu}{\rho}\frac{\partial U_t}{\partial n}\right)^{0.5} \tag{2.5}$$

The first region is characterized by $y^+ < 5$ and within it the flow exhibits laminar characteristics. As already pointed out in section 1.1.9 the $k-omega$ model works fine in the near wall region and so it can be exploited to solve the laminar sub-layer. More exactly the best model to use for this purpose is $k-\omega\ SST$ as this model solve accurately also the far wall region. As general recommendation the average y^+ value must be of unity order if the $k-\omega\ SST$ model is used without wall function but this is not the only

requirement. The second demand is to have at least 10 nodes of the mesh inside the boundary layer.

In the law of the wall region the inertia forces dominate the viscous forces. It is possible to use a semi-empirical expressions, this expression does not solve the boundary layer but bridge the viscosity region and the fully turbulent region. The main benefits of such technique consists in the reduction the mesh size and simulation time however the final results can be less realistic as those obtained solving the boundary layer. The y^+ in the near wall region must be bigger than 15 and smaller 300. A small value of y^+ generally is translated in unbounded errors. Using $k-\epsilon$ model with the standard wall function implies that the first node near the wall lies inside the log-law region of the boundary layer. However it is possible to use a so called *scalable wall function*, it imposes a virtually a displacement of the mesh to a $y^+ \sim 11.225$ avoiding bounding problems.

The user must find a trade off between accuracy, speed and stability; once the requirement are fixed it is possible to choose if to use a wall function or to solve the boundary layer. The CFD users community usually recommend to solve the laminar sub-layer if the interest is to calculate the drag force, boundary layer velocity, thermal profile or in situations with high changing in pressure gradient[7]. It is important to mention that from a convergence point of view $k-\epsilon$ with wall function and $k-\omega$ SST are at the same level.

2.6 Interfacial models for turbulent dispersion

In Euler model, as in VOF model, the pressure and volume fraction are computed solving phase and total continuity but the result is the probability where phase may exist. Euler ,unlike VOF model, does not solve exactly the interface. The interface is described through unknowns terms within NS averaged equations. These terms are direct consequence of boundary conditions at the interface which are solved exactly in VOF model. The main differences with the VOF mass conservation is that Eulerian mass conservation varies smoothly[8] and each phase has its own velocity field. From simulations it was deduced that the term that mostly influence the liquid spray breakup is the turbulence dispersion model, introduced with the equation 1.41. Turbulent dispersion accounts for the interface turbulent momentum transfer. Thanks to its mechanism the particles can be caught up in continuous phase turbulent eddies and carried from region of high concentration to region of low

[7] where the separation is expected; however if the separation is due to sharp changing of the geometry the $k-\epsilon$ model may be sufficient

[8] in VOF it has a jump

concentration.

Turbulent dispersion arises from averaging the interface drag model; considering a dispersed phase l and a continuous phase g if $K^{l,g}(\vec{\tilde{U}^l} - \vec{\tilde{U}^g})$ is the instantaneous drag, $K^{l,g}(\vec{U^l} - \vec{U^g})$ is the mean momentum exchange between phases and $\vec{T^{td,g}} = -\vec{F^{td,l}} Klg \vec{U^{dr}}$ is the turbulent dispersion force then:

$$K^{l,g}(\vec{\tilde{U}^l} - \vec{\tilde{U}^g}) = K^{l,g}(\vec{U^l} - \vec{U^g}) - Kl, g\vec{U^{dr}}$$

where $K^{l,g}$ is a interphase exchange coefficient. Usually in CFD codes the turbulent dispersion force is multiplied by a limiting factor that is not considered in this case. The drift velocity U^{dr} accounts the dispersion of the secondary phase due to transport by fluid.

Five types of models were analysed in this thesis:

- **Burns**: the equation that describes the force is

$$\vec{F}^{td,g} = -\vec{F}^{td,l} = -\bar{C}^{gl} \frac{k^{tg}}{\sigma^{\alpha g}} \left(\frac{1}{\bar{\alpha}^g} - \frac{1}{\bar{\alpha}^l} \right) \bar{\nabla} \alpha^g$$

  where $C^{g,l}$ can be defined as:

$$C^{g.l} = \frac{3}{4} C^d \frac{\alpha^l \rho^g}{d^l} |\vec{U^l} - \vec{U^g}|$$

  In previous equations l and g indicate the two phases, U is the velocity, d is the Sauter diameter, ρ is the density, C^d is a constant, $\sigma^{\alpha t}$ is the turbulent Prandtl number for volume fraction dispersion[9], α is the volume fraction, $k^{t,g}$ is the kinetic eddy viscosity of phase g

- **Lopez de Bertodano** model is described by the equation

$$\vec{T}^{td,g} = -\vec{T}^{td,l} = C^{TD} \rho^g k^g \nabla \alpha^l$$

  where k^g is turbulent kinetic energy in continuous phase ρ^g is the continuous phase density and $\nabla \alpha^l$ is the gradient of dispersed volume phase fraction.

- **Simonin model**: Simonin and Viollet proposed to calculate the turbulent dispersion force as:

$$\vec{T}^{td,g} = -\vec{T}^{td,l} = C^{TD} K^{lg} \frac{D^{t,lg}}{\sigma^{lg}} \left(\frac{\nabla \alpha^l}{\alpha^l} - \frac{\nabla \alpha^g}{\alpha^g} \right)$$

  where D^{pq} is the fluid-particulate dispersion tensor and σ^{pq} is the dispersion Prandtl number whose default value is 0.75.

[9] expected to be of order unity

- **Diffusion in VOF model** does not consider $\vec{T}^{td}$ as an interface force but model it as a turbulent diffusion term in the governing equation of the volume fraction. Considering equation 1.34 that is written for zero source term and for no mass transfer from one phase to another, the zero on the right part must be substituted by the turbulent dispersion term $\nabla(\gamma^g \nabla \alpha^l)$ where γ is the diffusion coefficient.

- **Constant model** is very similar as formulation to Lopez de Bertodano model. The expression that account turbulent dispersion is:

$$\vec{T}^{td,g} = -\vec{T}^{td,l} = C^{TD} \alpha^l \rho^g k^g \nabla \alpha^l$$

In Fluent are implemented the following models: Burns, Lopez de Bertodano, Diffusion in Vof and Simonin model while in OpenFoam are implemented Burns, Lopez de Bertodano and Constant model.

Chapter 3

Aerofoil simulations

This chapter is pertinent to the first part of simulations concerning aerofoil drag and lift calculation. It is divided in three main parts. The first part deals with the Fluent simulations and the second part deal with the OpenFoam simulations. Each of these two part are subdivided in subsection related to the geometry, mesh and CFD simulations. The third part gathers the conclusion. To validate the results was used a NACA0012 aerofoil section because it is the AGARD standard used in the testing of numerical solution; experimental data provided by Charles L. Ladson (1988) was considered. All the points of the profile were obtained using a modified equation provided by NASA which gives a unit chord. Experimental data are characterized by a Reynolds number that is varied from $2 \cdot 10^6$ to $6 \cdot 10^6$ and the Mach number that is varied from 0.05 to 0.36. For the purpose of this thesis was considered a low number of Mach and so the $\rho = constant$ hypothesis. It guarantees the independence from Mach number, so the independence from velocity. The experimental data regards a free transition and a forced transition from laminar to turbulence. A fixed transition of the boundary layer was obtained with carborundum strips[1] applied to upper and lower surfaces at the $0.05c$ station; the strips were about $0.01c$ in width. All experimental data includes corrections for the standard low-speed wind tunnel boundary effects.

In simulations the following parameters are imposed alike in Fluent and OpenFoam: boundary conditions are inlet, outlet and the aerofoil is a wall with no slip condition, the k is calculated imposing a turbulence intensity value $I = 0.05\%$ and the turbulence length scale is of unity order; the velocity value and density are fixed to unity value.

[1] Test and Procedures

3.1 Fluent Simulations

In this section it is presented how to create the geometry and the mesh for aerofoil using Ansys tools. It is supposed that the reader is familiar with Ansys interface and no screenshots are provided.

Fluent does not have a pre-processor so an external software must be adopted: Design Modeler. Usually externals CAD presents a too complex and full of details geometry to simulate. One of the requirement of CFD user is to recreate a simple geometry, poor of details but enough rigorous to capture the physics. For this reason ANSYS developed Design Modeler, very user friendly CAD. Ansys Meshing was chosen for the purpose to create the mesh.

3.1.1 Aerofoil geometry created with design modeler

Design modeler is provided by two windows: sketch window and model window. Inside sketch window the sketches are created; so only lines and points can be generated within it. The sketches is transported automatically inside the model window where 2D and 3D bodies can be created and managed.

The first step is to create the aerofoil's shape. On the first sketch the points of the profile must be generated.

Sketching → Draw → Construction Point

Once a point is created it is necessary to assign the distance from axis.

Sketching → Dimension → General → "select point" → "assign dimension"

The number of points to create depends on the required accuracy. In zones with emphasized curvature the density of points must be higher than in more linear zones. The profile's trailing edge was cut to avoid problems with skewness (see figure 3.1). If the profile geometry is maintained with a sharp tail, the automatic mesh software creates in that zone elements with a non acceptable value of skewness[2]. Once the points are created it is necessary to joint them with a curve using a spline.

Sketching → Draw → Spline → "select the points"

The spline must be transformed in a 2D surface inside modelling window.

Modeling → Concept → Surface from Sketches → "select the sketch"

[2]it can be avoid using a different software, for example *ICEM* where elements with high skewness can be deleted. The sharp angles are a real challenge for Ansys Meshing and must be avoid.

Once the operation is done the internal space delimited by the aerofoil profile is coloured in grey, this means that a surface is created. At this step a new item is compared inside the *body* menu, it is the surface previously created. Once the aerofoil shape is created, it is necessary to immerse it in the air. The air is represented by another surface. To create a new surface it is necessary to freeze the first surface, otherwise the modification to the model will be appended to the already existing surface.

Modeling → Tools → Freeze

Now it is possible to create a second sketch in the sketch menu.

Sketching → New Sketch

Within the new sketch it is possible to create the constructive points of the domain, join all the points with a spline (or line) and create a new surface. The shape of the domain is showed in figure 3.1. At this step there are 2 bodies in the body menu. Since it is not necessary to mesh the surface inside the aerofoil, this surface can be deleted using a boolean operation.

Modeling → Create → Boolean → Operation → Subtract

Once the second body was selected as target body and the first created body was selected as tool body it is sufficient to apply the modification. After this operation the aerofoil surface disappeared and the result is a surface with a hole whose shape is identical to the aerofoil profile. The geometry of the case is done. Ansys Meshing software can use virtual defined surfaces to bound the mesh's cell dimension. These virtual objects are called *bodies of influence*. Two bodies of influence were created;in figure 3.1 can be noticed two zones with different cell density, created employing the virtual bodies. The steps to create these virtual objects are equal to the sequence followed to create the geometry hence: the existing geometry must be frozen; new sketch must be generated; inside the sketch the geometry of the boundary of influence must be defined; the surface have to be generated from the sketch and in the end the applied modifications must be frozen.

3.1.2 Meshing

The next step consists in importing the geometry inside Ansys Meshing and mesh it. Non structural mesh was created. The following itemization provides a detailed information about the settings applied to the mesh.

- solver: CFD Fluent[3]
- maximum acceptable size: 7.1 $[m]$
- minimum acceptable size: $1e-06$ $[m]$
- inflation: smooth transition, transition ratio $7e-02$, Maximum layers 25, Growth rate 1.2
- dimension of elements on the surface of the aerofoil: $5e-04$ $[m]$
- dimension of elements inside the first body of influence: $5e-02$ $[m]$
- dimension of elements inside the second body of influence: 0.1 $[m]$
- dimension of elements in the far field zone: 0.5 $[m]$
- growth rate: 1.2
- mesh technique based on proximity and curvature
- type of elements: quadrilateral

Ansys meshing is very user friendly. To generate a *sizing*, an *inflation* or to define a *boundary of influence* it is enough to select with the right click of the mouse the item "mesh" and a new window will appear with previously mentioned operations. In 2D cases the body of influence operation are not presented by default; to apply a body of influence to a 2D case it is necessary to select the surface not with "surface tool" but with "body tool" and apply a size condition on it.

$$\boxed{\text{Mesh} \rightarrow \text{Insert} \rightarrow \text{Sizing} \rightarrow \text{Type} \rightarrow \text{Body of Influence}}$$

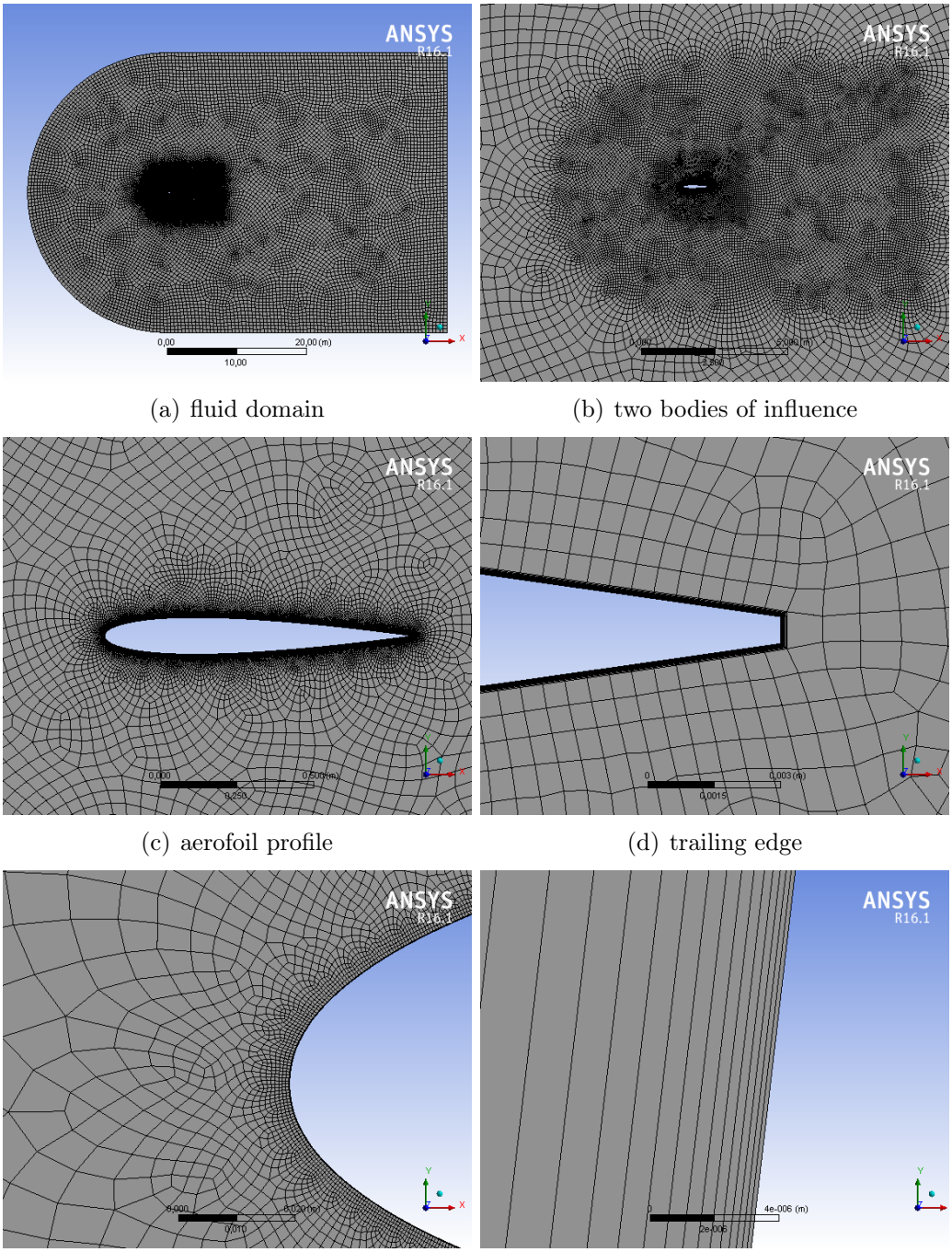
now it is possible to apply the desired condition selecting target and tool body. "Body of influence" is useful to control the size of the mesh relatively far from the profile. "Inflation" is a operation that manage the cells very closed to the wall. It extrudes a fixed number of cells in normal direction when the viscous boundary layer has to be solved.

$$\boxed{\text{Mesh} \rightarrow \text{Insert} \rightarrow \text{Inflation}}$$

Once the necessary dimensional bounds are applied and the mesh is generated it is necessary to name the boundaries of the domain. If the name of the

[3]Ansys Meshing create automatically the mesh placing the nodes according to the used solver

boundary is equal to one name in fluent data-base boundary condition than fluent assigns automatically the relative boundary condition; if a surface (or line)is called $pressure - outlet - 1$ fluent will assign to it a pressure outlet boundary condition. The figure 3.1 shows the main details that characterized all the created meshes. The mesh in figure 3.1 is characterized by 169691 nodes, 167548 elements and a value of maximum skewness equal to 0.9.



(a) fluid domain

(b) two bodies of influence

(c) aerofoil profile

(d) trailing edge

(e) part of extruded zone distantly

(f) part of extruded zone nearly

Figure 3.1: General aspects of the aerofoil mesh used in fluent

3.1.3 CFD results obtained with Fluent

In this section the results obtained with fluent are presented. The section is divided in 2 parts. The first part deals with the mesh sensitivity analysis considering a null angle of attack. The second part presents the result obtained with non zero angles of attack

Mesh sensitivity

A first mesh with poor quality was created and the C_d and C_l were analysed. The mesh was changed increasing the number of cells in the far zone and changing the y^+ value acting on the dimension and on the number of the extruded elements from the aerofoil surface. A null angle of attack was considered $\alpha = 0$. The iterative process to create the mesh is reported:

- *mesh1* is the case with the highest value of y^+. Only 2 layers where extruded in the normal direction from the wall surface. Since the y^+ value is too high the software automatically switch to $k - \epsilon$ turbulence model with wall function providing a value of $C_d = 0.0087$

- *mesh2*: was used the parameters of the first mesh but the number of extruded layers was brought from 2 to 10. The transition ratio was set from 0.272 to 0.17. A *k-w SST* model was used. The $C_d = 0.83$ was obtained in this configuration.

- *mesh3*: in this case all the parameters were decreased to create a finer mesh, the result was a $C_d = 0.008$.

- *mesh4*: further reduction of values were performed, nevertheless the drag coefficient remained the same.

- *mesh5*: in this case were reduced the dimension of elements near the profile, the C_d coefficient decreased to 0.0077

- *mesh6*: because of small reduction of drag obtained with the fifth mesh, were preferred to extrude more elements from the wall surface, the number of elements were brought from 15 to 25. This choice brought to a $C_d = 0.0078$ very similar as the result obtained with mesh number five. This mean that additional nodes inside the viscous sublayer does not change the results

- *mesh7*: Was decided to reduce the value of y^+ by a further reduction of transition ratio, was obtained a $C_d = 0.0072$.

-	D_{max}	D_{b1}	D_{b2}	D_{0012}	N_l	N_{tr}	N_{gr}	Skew	Nodes
mesh1	0.7	0.15	0.1	0.001	2	0.272	1.2	0.6933	38366
mesh2	0.7	0.15	0.1	0.001	10	0.17	1.2	0.7970	54175
mesh3	0.6	0.13	0.07	0.0007	13	0.15	1.2	0.7214	84794
mesh4	0.5	0.1	0.05	0.0005	15	0.13	1.2	0.8122	127423
mesh5	0.5	0.07	0.03	0.0003	15	0.1	1.2	0.7257	214104
mesh6	0.5	0.1	0.05	0.0002	25	0.1	1.2	0.9094	384832
mesh 7	0.5	0.1	0.05	0.0005	25	0.07	1.2	0.7645	170030
mesh8	0.5	0.1	0.05	0.0005	30	0.05	1.2	0.81118	190749

Table 3.1: Mesh characteristics related to the mesh sensitivity analysis in fluent

- *mesh8*: was considered the same setups of mesh seven but the number of layeres was increased to 30 and transition ratio decreased to 0.05. The resulting C_d is 0.007. At this step it is possible to conclude the the result of C_d does not depend on any more from the number of the elements in the viscous sub layer, from the value of y^+ and from the size of the element far from the aerofoil

The table 3.1 summarizes the mesh alteration described in the previous itemization. In the table 3.1 D_{max} is the maximum dimension of cells in the domain, it characterizes the far zone from the mesh; D_{b1} is the dimension of cells in the outer zone identified by the biggest body of influence; D_{b2} is the dimension in the inner zone identified by the smallest body of influence; D_{0012} is the distance between nodes on the aerofoil surface. The mesh within viscous boundary layer is characterized by N_l the number of layers extruded in normal direction from the surface of the wall with a N_{tr} transition ratio between the layers and growth rate equal to N_{gr}. All the dimensions reported in table 3.1 are in meters. The table 3.2 presents the results in terms of C_d and C_l obtained in configuration $\alpha = 0$. From tables 3.2 and the table 3.3 it is plain that the results obtained with the model $k - \omega$ SST is very close to experimental data in free transition. Mesh number seven and mesh number eight provide the best results. Mesh number seven was chosen for simulations with a non zero angle of attack since it has less elements comparing it with mesh number eight and so less computational effort is required for simulations.

mesh	C_d Fluent	C_d free transition	C_d 80-W Grit	y^+ max
mesh1	0.0087	0.0065	0.00826	40
mesh2	0.0083	0.0065	0.00826	6
mesh3	0.0080	0.0065	0.00826	2
mesh4	0.0080	0.0065	0.00826	0.9
mesh5	0.0077	0.0065	0.00826	0.4
mesh6	0.0078	0.0065	0.00826	0.045
mesh7	0.0072	0.0065	0.00826	0.08
mesh8	0.007	0.0065	0.00826	0.02

Table 3.2: Fluent CFD results in configuration $\alpha = 0$ compared with experimental data in free transition and force transition (80-W Grit)

Mach	Re	α	C_d	C_l	type of transition
0.15	$6 * 10^6$	0.05	0.0065	-	free transition
0.15	$6 * 10^6$	0.00	0.00895	-	fixed transition 60-W Grid
0.15	$5.95 * 10^6$	-0.05	0.00809	-	fixed transition 80-W Grid
0.15	$6 * 10^6$	0.01	0.00804	-	fixed transition 120-W Grid
0.15	$5.95 * 10^6$	0.04	0.00811	-	fixed transition 180-W Grid

Table 3.3: Experimental data in configuration $\alpha = 0$ for NACA0012

3.1.4 Results in the configuration: non zero attack angle

The table 3.4 presents the results of the simulations considering a non zero angle of attack. The first observation is that the lift coefficient is very similar in simulations and fix transition experiments despite drag coefficient is not; indeed the simulations consider a turbulent fully developed flow and this is one of the reason why C_d is higher. The second observation is that nevertheless the mesh number eight gave good results in agreement with free transition experimental data for $\alpha = 0$, for a non zero angle of attack the

angle	simulation		free transition		fixed transition 80-W Grit	
	C_d	C_l	C_d	C_l	C_d	C_l
$\alpha = 6.20$	0.00908	0.6918	0.00680	0.6630	0.00913	0.6895
$\alpha = 10.18$	0.01315	1.1118	0.01050	1.0880	0.01420	1.1231
$\alpha = 13.10$	0.0184	1.39	0.0130	1.3680	0.02360	1.39

Table 3.4: Fluent CFD results in different α configurations compared with experimental data in free transition and force transition (80-W Grit)

results are closer to fix transition. Comparing the C_d value in experimental fix transition results and simulations an error of 0.5% for a $\alpha = 6.20$, an error of 7% is obtained for $\alpha = 10.18$ and an error of 22% is obtained in the case of $\alpha = 13.10$. For a null angle of attack comparing the Rans simulation and experimental 80-W Grit C_d value the error is 13%. The model $k - \omega SST$ with the chosen boundary conditions gives a relative small error for small angle of attack. For $\alpha = 13.10$ the error doubled its value. The y^+ value, in all the simulations whose results are presented in this section, is proved to be smaller than 1. It is important to mark that the mesh was created to fit free transition data instead it provided results close to fix transition data for a non null angle of attack.

3.2 OpenFoam Simulations

In this section is presented how to create a mesh inside OpenFoam without using an external pre-processor. The results obtained are compared with experimental data.

3.2.1 Geometry and Mesh for OpenFoam

OpenFoam provides different possibilities to mesh a geometry. A mesh can be created with an external software and imported, otherwise a fluid domain can be defined and meshed inside OpenFoam. Two commands can be used for meshing in OpenFoam: *blockMesh* and *snappyHexMesh* usually used for complex geometry imported from a CAD. To discretize the aerofoil fluid domain was used only the blockMesh command and was created a structural mesh. Due to a relative big quantity of information within the input file, it was created with a script in Matlab. The mesh input file must contain all the vertices of the fluid domain, the definition of all the surfaces and boundaries, the definition of all the blocks, grading information, maximum number of elements in any block and other information about the mesh. For all zones described bellow the spatial distribution of nodes was chosen to have a maximum increase ratio of 2 between to neighbour cells. All the input items were defined in Matlab and to print it within .txt file was used the matlab command *fprintf*. The output *blockMeshDict* file is used as input file to create the mesh inside OpenFoam. An encode problem was find while reading with OpenFoam the matlab generated file. Inside *fprintf* string, the command used by a matlab user to type a "space" between symbols is */b*. Since OpenFoam does not recognize the "space" inserted by matlab it is necessary to replace it manually by a "space" inserted with the keyboard.

Bounds imposed in matlab script to create the mesh

The upper and lower surface of the profile is divided in three parts, each one can be discretized with an N number of elements characterized by a previously decided dimension:

- On the first part it was allocated nodes apart dx_1 to describe with a good approximation the curvature of the profile.
- On the second part, if as in our case there is not a conspicuous curvature, it is possible to allocate a small number of cells gaining resource during simulations; the distance between nodes is dx_2

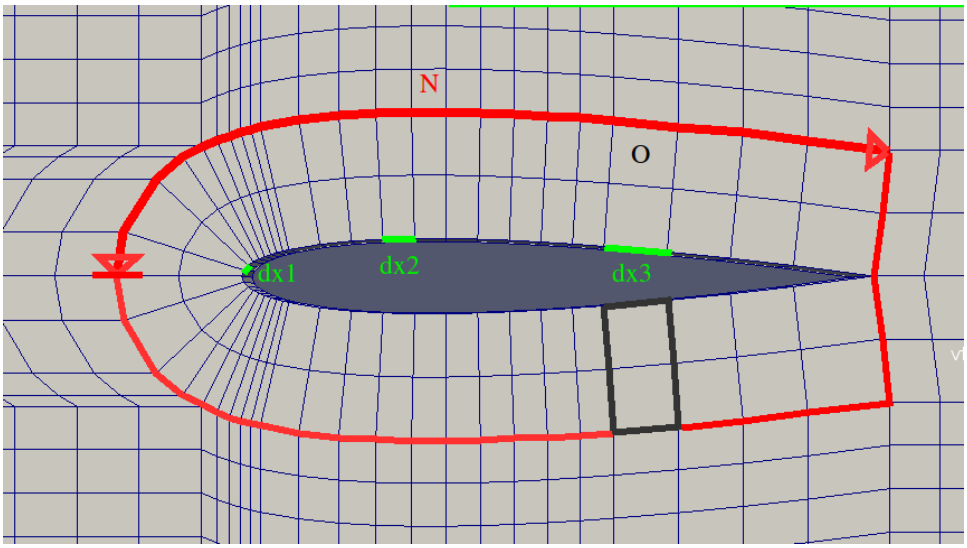


Figure 3.2: Details of the mesh employed in the aerofoil OpenFoam simulations

- The third and last zone on the profile is the most linear, it was allocated the last number of nodes with a dx_3 distance one from each other. The dimension of the last element in the tail of the profile is calculated to reach a unity chord.

The Naca profile ends with a sharp tail; no cut was done as for the mesh simulated in fluent[4]. The aerofoil coordinates are saved in a matrix, the first and second column is respectively the x and y coordinates of upper portion of the profile while the third and fourth column are the x and y coordinates of lower portion of the profile. This organization of the matrix allows to use also a non symmetric profile. As mentioned in section 2.5 the turbulence model is generally linked to the mesh; indeed the dimension of the first element near the wall has to be controlled and to satisfy the required y^+ value. For this reason was created a first zone of influence on the mesh. This zone is named **O** and is characterized by $2 \cdot (N-1)$ blocks allocated in the direction of the curvilinear coordinate. One of these blocks is marked in black on figure 3.2 Each block has n cells (in figure 3.2 $n = 2$) in the normal direction of the profile and one cell in the direction that connects two neighbour nodes on the profile. The thickness of this n cells is calculated in order to reach a desired y^+ value considering a magnitude of velocity with unity order. It is recommended to control the y^+ value after the simulation to be sure about

[4]With AnsysMeshing a cusp tail is the origin of elements with a high value of skewness. In this case the position of nodes are decided to avoid high skew cells

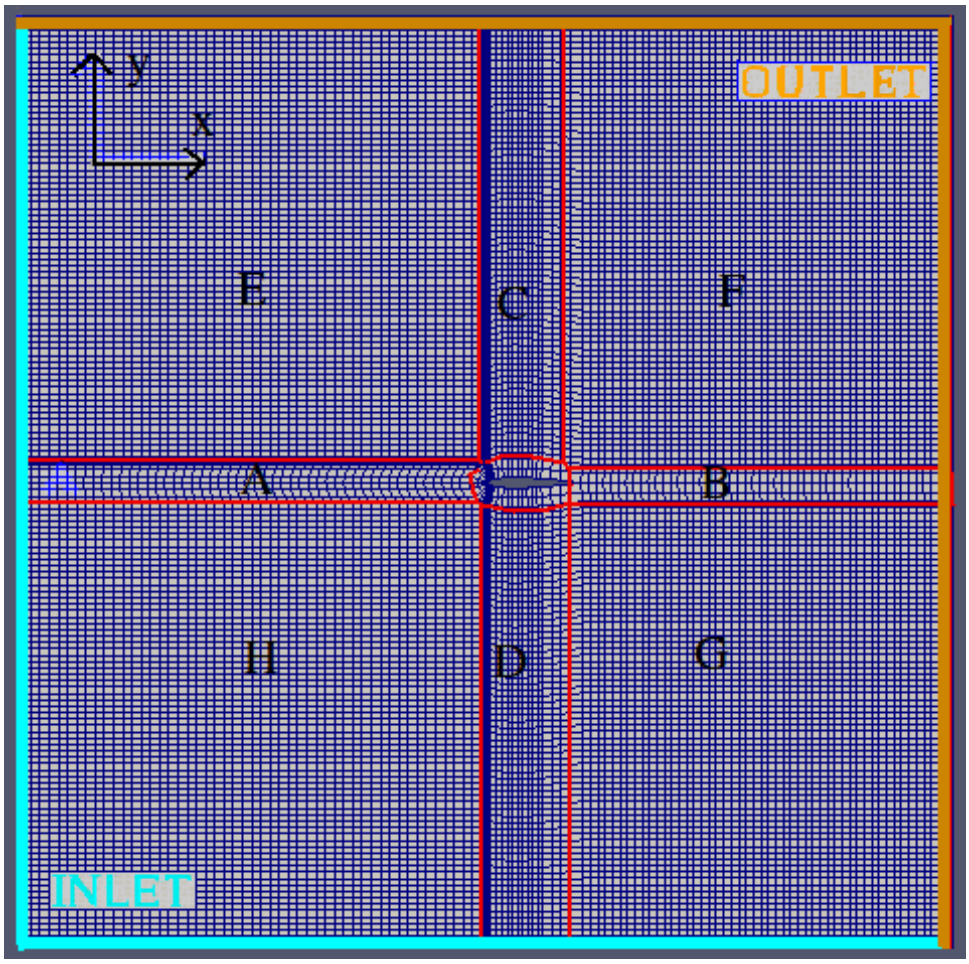


Figure 3.3: Details of the mesh employed in the aerofoil OpenFoam simulations

it's value. The distance between the nodes inside each block is controlled through *grading* command and because two neighbour blocks share half of their boundary points inside this zone the grading is equal for all blocks.

Once the distance of inlet and outlet from the aerofoil profile was chosen it is possible to create the boundaries of the air domain. The frontal part of the mesh (zone **A** on figure 3.3) is created by a p number of horizontal blocks. The number p is linked to the value of skewness in this zone; indeed it is chosen automatically to have a maximum angle of 45 degree between the cells in Zone **O** and cells in zone **A** (figure 3.2). Each of p blocks shares a face with one block in zone **O** while the opposite face is an inlet of the domain. The grading scheme of each block has at least 100 hundred nodes in x direction and 1 element in y direction.

The zone **B** of the mesh is the one which connects the zone **O** and the outlet. This zone is created with $2 \cdot n$ horizontal blocks. The grading scheme of the blocks is the same as the grading scheme in zone **O**. Because of the small thickness value of the n cells in zone **O** particular attention have to be paid to zone **B**. Indeed the thickness of the each cell is equal with the dimension calculated to achieve the desired y^+ value. If the blocks in zone **B** have a small number of cells in x direction it is possible to have cells with high aspect ratio inside this zone. OpenFoam gives a possibility to check the high aspect ratio value with *checkMesh* command. If the aspect ratio is more than 1000 a warning will be displayed in the terminal. The ability to solve zones with high aspect ration depends on the CFD code implementation but it is recommended to avoid high aspect ratio elements.

The zones **C** and **D** connect the zone **O** with upper and lower surface of the domain which are, respectively, outlet and inlet. Each of this zone is composed by vertical blocks, every block share a face with one block of the zone **O** and the opposite face is a boundary for the air domain. This blocks has at least 75 elements in y direction and 1 element in the x direction. These are the regions with high cell density because of the shared face with the zone **O**. The spacial grading in y direction is equal for all blocks because two neighbour blocks share half of their nodes.

The zone **E** and **H** have the same number of cells in x direction as the zone number **A** while the zones **F** and **G** have an equal number of cells to zone **B** in the horizontal direction. The zones **E** and **F** has equal number of nodes to zone **C** in y direction while zones **G** and **H** have the same number of cells as the zone **D** in the vertical direction.

All the bonds previously described are imposed inside the matlab script. The script is useful because permits to change the geometry of the aerofoil and the mesh in some seconds. A similar script can be made by Octave, an open source matlab clone. The combination of Octave and OpenFoam can be interesting because in cases with relatively simple geometry it avoids the usage of an external CAD software.

3.2.2 Mesh transfer from OpenFoam to Fluent

To make a comparison between two softwares it is appropriate to use the same mesh. The peculiarity of the OpenFoam is that even a 2D case needs a three dimensional mesh. Indeed the 2D case that lies on xy plane has 1 cell in z direction. On the contrary Fluent uses a 2D mesh for two-dimensional simulations. Since there is not a command to translate the $3D$ OpenFoam format mesh in a $2D$ Fluent mesh in this section is explained how to transform a 3D mesh in a 2D mesh using ICEM software. The 3D aerofoil mesh was

created inside OpenFoam with the Matlab algorithm previously described.

The first step is to translate the mesh from OpenFoam format to fluent format [.msh], it is easily done by opening the terminal within the main folder[5] and typing the string *foamMeshToFluent*. A new folder will appear containing the $3D$ mesh in .msh format.

Now the mesh must be imported in Fluent, so a new three dimensional case must be opened in Fluent. Now the $3D$ OpenFoam mesh must be imported inside Fluent.

File → Read → Mesh → name.msh

The case must be saved in fluent format *.cas*

File → Write → Case → Ok

Finally the fluent format *.cas* can be imported in ICEM.

File → Import Mesh → From Fluent → name.cas

Once the mesh is imported in ICEM it must be modified removing half of the nodes, the nodes to remove are showed in figure 3.4 (b).

Edit Mesh → Delete Nodes → "select all the nodes on one face" → Apply

Previous operation deleted one face and transformed the 3D mesh into a 2D mesh (figure 3.4). Within "section parts" window disappeared all the boundary conditions with exception of two. The first item in the "section parts" window contains the internal information of the mesh, the second item contains all the boundaries of the mesh. Now the mesh is two-dimensional and can be saved in *.msh* format.

Output → Select Solver → Output Solver → Ansys Fluent → Apply

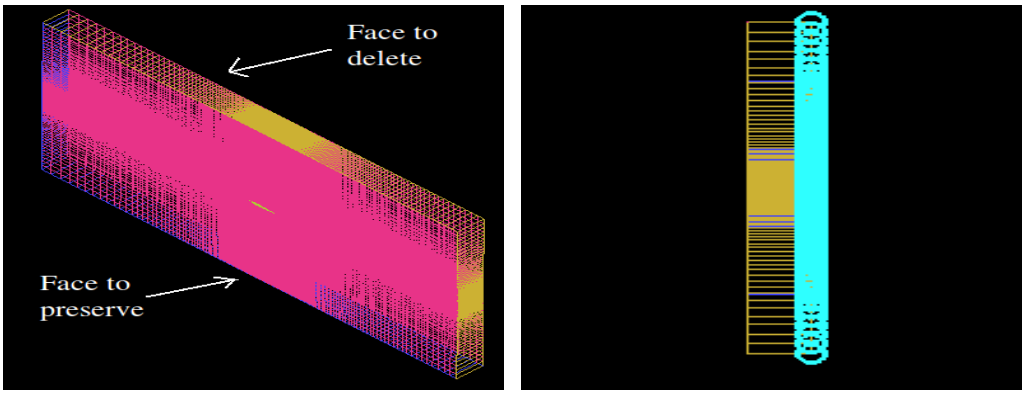
Output → Save → Open → 2D → Done

At this step the mesh can be imported in fluent as a two dimensional case. Once the mesh is imported only two surfaces exist: interior and boundary. It is possible to split the boundary surface in inlet, outlet and wall. To do so it is necessary to follow the sequence:

Mesh → Separate → Faces

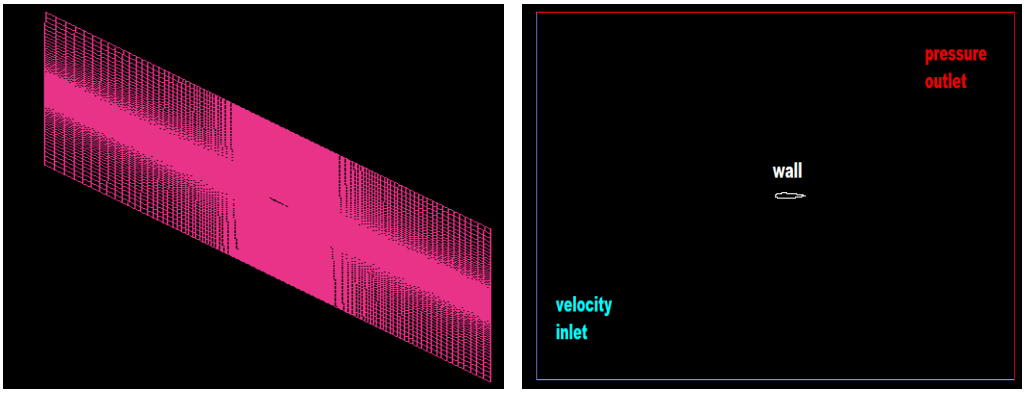
a new window appears; select "angle option", set angle degree equal to 90 and select only boundary as "zone". This operation will split the boundary in 5 zones that can be renamed inside the boundary conditions menu. The result of all this operations is showed in figure 3.5.

[5]where are located folder 0, constant, system ...



(a) (b)

Figure 3.4: Transform a 3D mesh to a 2D mesh



(a) (b)

Figure 3.5: Fluent mesh and boundary conditions

3.2.3 Setting the case in OpenFoam

All the information about the simulations is stored inside at least three folders: folder **0**, folder **constant** and folder **system**. Each folder contains some files with the simulation's detail. These folders are necessary because OpenFoam is not provided with an interface and so the user communicates with the software through files in the previous mentioned folders. The folders with relative files are described below:

- folder **0**: the files inside this folder contains the information about boundary conditions and the initial conditions. Each flow property has it's own file that contains the physical dimension, BC, IC and physical dimension.
  - file k: is the file that contains the information about the turbulent kinetic energy, the formula 2.2 can be used to evaluate the values at inlet. It has been recommended never to define $k = 0$ but to give a very small value if k is null.
  - file *nut*: contains the information about turbulent eddy viscosity. If $k - \omega\ SST$ model is used than this file is not necessary. The presence of *nut* file in 0 folder ensure that the value of eddy viscosity will be saved each time that the case is saved. On all the boundaries *nut* must be set to "type calculated".
  - file *omega*: refers to the specific turbulent dissipation rate. It's value can be calculated using the formula 2.4. On the wall surfaces where a wall function is applied it a Menter's empirical expressions can be used to evaluate ω, for example $omegaWall = \frac{60\nu}{0.075y^2}$
  - file p: is te file containing the pressure information. Particular attention have to be paid to the dimension in this file indeed the pressure within is defined[6] as $\frac{p}{\rho}$
  - file U: is the velocity file. Contains the magnitude and the direction of the velocity vector; within a non slip condition can be imposed on wall surface.
  - file *yPlus*: is a file of post-process family, it contains the value of y^+. This file appears after a y^+ check on the terminal and is not necessary to begin the simulation
- folder **constant**: inside this folder are collocated two files and one folder

[6] the solver does not require the density value

  - file *transportProperties*: in this file a Newtonian transport model is specified and the value for kinematic viscosity (nu) is assigned
  - file *turbulenceProperties*: inside this file the RAS $k-\omega$ SST model is chosen and so turbulence is activated
  - folder *polyMesh*: this folder is generated automatically once the command blockMesh is run on the terminal. There are 5 files inside this folder with the geometry and mesh information

- folder **system**: inside this folder four files are collocated
  - file *blockMeshDict* contains the information about the geometry and the mesh[7].
  - file *controlDict* set the solver's type[8], the initial and final time of the simulation, the time interval between two time steps and other information[9], it follows an example of the options:
    * application simpleFoam
    * startFrom startTime
    * startTime 0
    * stopAt endTime
    * endTime 2000
    * deltaT 1
    * writeControl timeStep
    * writeInterval 50
    * purgeWrite 0
    * writeFormat ascii
    * writePrecision 7
    * writeCompression off
    * timeFormat general
    * timePrecision 6
    * runTimeModifiable true
    * functions include 'forceCoeffs'

- file *forceCoeffs*: this file is needed to calculate C_d and C_l. The options set are:

[7] this file was generated by matlabas explained in the previous section

[8] the solve is chosen accordingly to the physics

[9] the function used to calculate the C_d and C_l must be called inside this file (see the last item in the controlDict file)

  - lifDir $-sin(\alpha)$ $cos(\alpha)$ 0
  - dragDir $cos(\alpha)$ $sin(\alpha)$ 0
  - magUInf 1
  - lRef 1
  - Aref 1

  It is very important to put identical setup for the calculation of C_d and C_l in Fluent and OpenFoam, particular attention have to be paid to reference values as length and area.

- *fvSchemes* contains the following information
  - ddtScheme default steadyState
  - gradScheme default Gauss linear
  - gradScheme grad(U) cellLimited Gauss linear 1
  - divScheme default none
  - divScheme div(phi U)bounded Gauss linearUpwindV grad(U)
  - divScheme div(phi K) bounded Gauss upwind
  - divScheme div(phi omega) bounded Gauss upwind
  - divScheme div((nuEff*dev2(T(grad(U))))) Gauss linear
  - laplacianSchemes default Gauss linear corrected
  - interpolationSchemes default linear
  - snGradSchemes default corrected
  - wallDist method meshWave
- *fvSolution* contains the following solvers information
  - p
    - solver GAMG
    - tolerance $1e-7$
    - relTol 0.01
    - smoother GaussSeidel
  - phi
    - $p
  - U

        * solver smoothSolver
        * tolerance $1e-8$
        * relTol 0.1
        * smoother GaussSeidel
        * nSweeps 1
    - k
        * solver smoothSolver
        * tolerance $1e-8$
        * relTol 0.1
        * smoother GaussSeidel
        * nSweeps 1
    - omega
        * solver smoothSolver
        * tolerance $1e-8$
        * relTol 0.1
        * smoother GaussSeidel
        * nSweeps 1
    - SIMPLE
        * nNonOrthogonalCorrectors 0
        * consisten YES
    - potentialFlow
        * nNonOrthogonalCorrectors 10
    - relaxationFactors equations
        * U 0.9
        * k 0.7
        * omega 0.7
    - cache
        * grad(U)

- **postProcessing**: in this folder is saved the file *forceCoeffs* with the data relative to drag and lift coefficients

	OpenFoam		free transition		80-W Grit	
	C_d	C_l	C_d	C_l	C_d	C_l
$\alpha = 4.18$	0.00817	0.458	0.00760	0.4520	0.00832	0.4527
$\alpha = 6.20$	0.01117	0.667	0.00680	0.6630	0.00913	0.6895
$\alpha = 8.22$	0.01489	0.887	0.00800	0.8800	0.01060	0.9065
$\alpha = 10.18$	0.02593	1.060	0.01050	1.0880	0.01420	1.1231

Table 3.5: openFoam and experimental data

3.2.4 Simulations results

All the cases run in OpenFoam have the same setup, the only parameter which varies is the angle of attack. The first attempt was to use $k-\omega$ SST model without wall functions. A similar mesh sensitivity analysis as for fluent (see section 3.1.3) was done. One of the biggest problem was to create a mesh with y^+ value close to one. A such mesh involve a big number of cells, indeed to have a good quality mesh it is recommended to maintain the area ratio between two neighbour cells near a value of 0.5[10], by the way in fluent this bond was not kept[11]. Different simulations were done with different meshes trying to solve the viscous boundary layer. All the simulations failed because of divergence problem. The OpenFoam model *k-w SST* without wall function for ω is very sensitive to boundary conditions, initial conditions and mesh[12]. To reach the convergence the mesh was changed, the y^+ value was increased and a wall function was used. The turbulence parameters are defined in the opening of this chapter; in particular on the wall of the aerofoil profile omega was set equal to the value calculated for the inlet.

The table 3.5 presents the results obtained with OpenFoam and the experimental data. Also in this case the CFD results are closer to fix transition experimental data. Considering the drag coefficient the error at $\alpha = 4.18$ is 2%; the error increases increasing the angle indeed at $\alpha = 6.20$ it becomes 22.34%, at $\alpha = 8.22$ it arrives at 40% and finally at $\alpha = 10.18$ the error reaches the value of 80%. As in fluent the behaviour is similar, the error in evaluating the C_d increasing with the angle. This increment is much more emphasized in OpenFoam, the reason is that the used model does not solve the viscous boundary layer but uses a wall function.

[10] area ratio = $\frac{smaller_area}{bigger_area}$

[11] it was concluded that even a smaller ratio between the cells in a close region near the profile bring to acceptable results

[12] this problem was find also by other users as reported on CFDonline community

	OpenFoam		free transition	
	C_d	C_l	C_d	C_l
$\alpha = 4.18$	0.00604	0.4647	0.0076	0.4520
$\alpha = 6.20$	0.00778	0.6683	0.0068	0.663
$\alpha = 8.22$	0.00889	0.9082	0.008	0.88
$\alpha = 10.18$	0.01294	1.1061	0.01050	1.0880

Table 3.6: Sensitiveness of OpenFoam results regarding boundary conditions

3.2.5 The importance of boundary conditions

If the aim is to evaluate the forces acting on the aerofoil it is of primary importance to impose the right turbulent boundary conditions. The choice of the turbulence intensity and turbulence length scale can be crucial to obtain a desired result. The simulations in this section does not consider the values of turbulence intensity and length scale defined in the opening of this chapter. A turbulence intensity was decreased to a value of 2% while the ω value on the wall was calculated with the Menter's empirical equation. The results are reported in the table 3.6 The results are closer to free transition data[13] In this case the drag coefficient error remains much more steady in the studied range: for $\alpha = 4.18$ it's value is 22%, for $\alpha = 6.20$ the error decreases a little to the value of 14%, for $\alpha = 8.22$ the error becomes 11% and finally for $\alpha = 10.18$ the error increases to 23%.

[13]in literature there is different Menter's empirical expressions according to the wall roughness

3.3 Closure

3.3.1 Comparison between two simulations with identical conditions executed with OpenFoam and Fluent

The OpenFoam mesh was translated in the Fluent *.msh* format to perform a comparison between the two software. Same boundary conditions were applied (figure 3.5) and same order numerical scheme was used. The following itemize provides the information about the settings

- **precision**: double precision[14]
- **solver**: pressure based steady solver
- **divergence scheme**: second order
- **gradient scheme**: gauss cell centred
- **scheme for k and ϵ**: first order
- **coupling pressure velocity**: yes[15]
- **velocity**: a unity velocity magnitude with an angle of 4.18 respect to the x axis.
- **turbulence**: 2% turbulence intensity and unity turbulence length scale
- **boundaries**: the same boundaries as in OpenFoam simulations

It is interesting to note that fluent solver with coupled equations for pressure and velocity converged in 230 iterations while the OpenFoam software converged in quasi 9000 iterations. So Fluent is much faster in convergence with these numerical schemes. Changing OpenFoam numerical schemes it was possible to make converge the case in quasi 700 iterations. The values of drag coefficient are reported in the table 3.7

[14]in fluent the user can arbitrary choose a single precision o double precision solver; in OpenFoam the double precision is set by default, to switch to single precision it is necessary to recompile the solver.

[15]too couple pressure and velocity in simple foam it is necessary to set *consistent yes* in system file under SIMPLE entry

	OpenFoam	Fluent
	C_d	C_d
$\alpha = 6.20$	0.0112	0.0107

Table 3.7: Lift and drag coefficients obtained with OpenFoam and Fluent considering identical geometry, boundary conditions and similar numerical schemes

3.3.2 Conclusion

Both software gave acceptable results for lift coefficient; the main difference lies in the evaluation of drag coefficient. Drag force is due to the pressure and shear force acting on the profile and so it is more complex to predict drag force rather than lift force. Two different models was used to predict drag: solving the viscous boundary layer with Fluent or using OpenFoam with wall function.

Fluent used for the simulations a non structural mesh with a $y^+ < 1$ and $k-\omega\ SST$ model without wall functions. It must be marked that if the first nodes do not lie in viscous sub-layer (for example $y^+ = 40$) fluent $k-\omega\ SST$ automatically impose a wall function conditions. It is important to check in post-process the y^+ in order to understand if the obtained result is due to a wall function technique or not. When $k-\omega\ SST$ model is used fluent does not notify to the user if a wall function was applied or not.

In OpenFoam it was not possible to solve the boundary layer because of problems with convergence. For this reason a mesh with $y^+ > 15$ and $k-\omega\ SST$ model with wall function was used, this permitted to have a mesh with a less number of cells against fluent case. The drag coefficient was evaluated for angles smaller than 10. The error respect the free transition data in evaluating C_d lies in a range of 10% - 20%. It was showed how sensitive is OpenFoam to the boundary conditions and how important is to set the write BC in order to obtain realistic results. As figure 3.6 shows, changing the boundary conditions at the aerofoil's wall can improve, in a significant manner the results Both softwares show similar behaviour for high angle of attack: the C_d error increases too much for angles higher than 10 degree in OpenFoam and 13 degree in Fluent. The main difference regarding boundary conditions comparing OpenFoam and Fluent is that OpenFoam require a boundary condition of k and ω not only at inlet and outlet but also near the wall; Fluent does not require a value of k and ω near the wall but only information about the slip condition and roughness (even if a wall function is used). Using a $k-\omega\ SST$ model in OpenFoam the user must switch manually

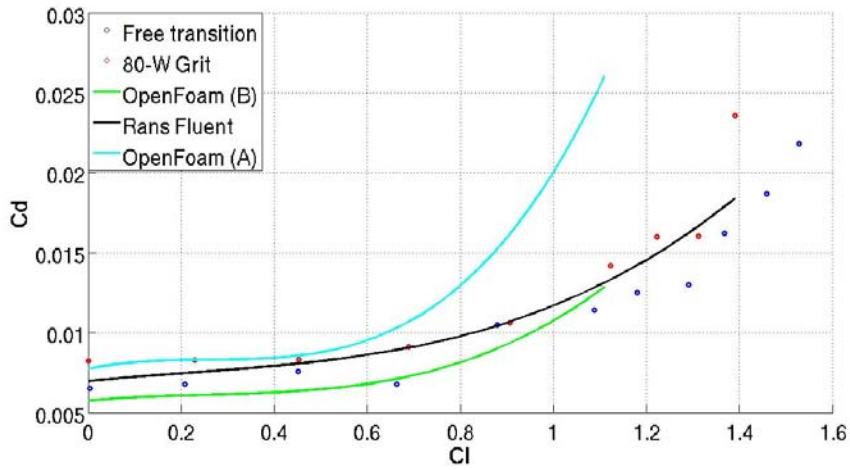


Figure 3.6: Drag and Lift coefficient obtained with CFD and compared with experimental results

from solving the boundary layer or applying a wall-function[16], Fluent does this operation automatically.

[16]a mesh which solves viscous boundary layer can not be used with wall-function and vice versa

Chapter 4

Breakup simulations

In this chapter the results concerning the near nozzle liquid jet breakup are presented. The aim is to study the liquid penetration length and the aperture of the liquid cone. Here, the liquid penetration length is defined as the axial position where the liquid volume fraction is bellow $\alpha = 0.05$. The simulations was performed with Fluent and OpenFoam. In all the simulations the computational domain is a cylinder with axial liquid injection. An axis symmetric model was used because of the symmetry in geometry and boundary conditions. Equal initial and boundary conditions were imposed in OpenFoam and Fluent. The simulation starts at time $t = 0$ seconds when the injection of liquid is switched on hence as initial condition the air phase fraction was set to 1 while fuel phase fraction was set to 0 everywhere inside the domain. At initial instant the domain is filled with still gas. The liquid phase enter inside the computational domain through a velocity inlet. A uniform value for ϵ and k was imposed on inlet while wall functions were imposed on walls. The density of the air is an input parameter and so even the pressure (the temperature field is uniform and constant). The pressure inside the computational domain was calculated using ideal gas state expression. The temperature of the two phases is ambient.

In both CFD a multiphase euler solver was used. Since no evaporation occurs the energy equation can be switched off. In fluent it is easy to switch off the energy equation and impose constant density for the gas phase; in OpenFoam this is not possible, to switch off the energy equation modifications to the code must be carried. To avoid modifications of the OpenFoam solver the relative temperature residual is set to 2, in this manner the OpenFoam does not solve the energy equation. The DNS data were provided by R. Lebas et al (2009) and the experimental data were provided by A. I. Ramirez (2009)

4.0.1 VOF Model

The first attempt was to simulate the break up using Volume Of Fluid (VOF) technique. Since similar qualitative results were obtained with Fluent and OpenFoam only Fluent results are presented in this section.

VOF model is used to simulate two or more immiscible fluids; the flow is described by only one momentum equation. It is applied to simulate jet breakup[1], motion of large bubble in liquid, dam break, tracking of liquid gas interface. VOF formulation as Euler-Euler formulation are used to compute a time-dependent solution, but a steady solver is available too. The main hypothesis of this model is that the two fluids are not inter-penetrating. VOF description requires a phase indicator function hence the equation 1.25 is still valid. The model solves secondary phase and total continuity equations; the result is pressure and volume fraction. Knowing the volume fraction it is possible to identify the position of the interface. It also solve one set of momentum equations using mixture properties. All variables or properties are shared by the two phases, since the phase volume fraction in any cell is known the variables are representative of one of the phases ($\alpha^a = 1$ within cell) or the variables are representative of a mixture of the phases. It is interesting to notice that since the relative velocity between phases is not computed it is not possible to calculate the drag acting on phases and the equation 1.41 can not be solved.

Several simulations where executed. The characteristics of domain geometry are described by table 4.1, the boundary conditions are summarized in table 4.3 and table 4.4. Turbulence $k - \epsilon$ and $k - \omega$ models were used, implicit and explicit time schemes were adopted but the qualitative results remain always the same. The results are represented in figure 4.1. From the qualitative imagine it is clear that jet breakup does not occurs. The reasons is that VOF model (in OpenFoam it is called *interFoam* solver)does not supply inter-phases momentum transfer. A sharp interface between gas and liquid is preserved. RANS VOF model is not adjusted to simulate liquid jet breakup. It is a direct method of predicting interface shape between immiscible phases, for example stratified flow regimes can be modelled by direct tracking of interface with Volume of Fluid model. It is also important to emphasize that VOF model solves exact multiphase equation and boundary conditions at interface.

[1] see tutorial 18 on inkjet printer

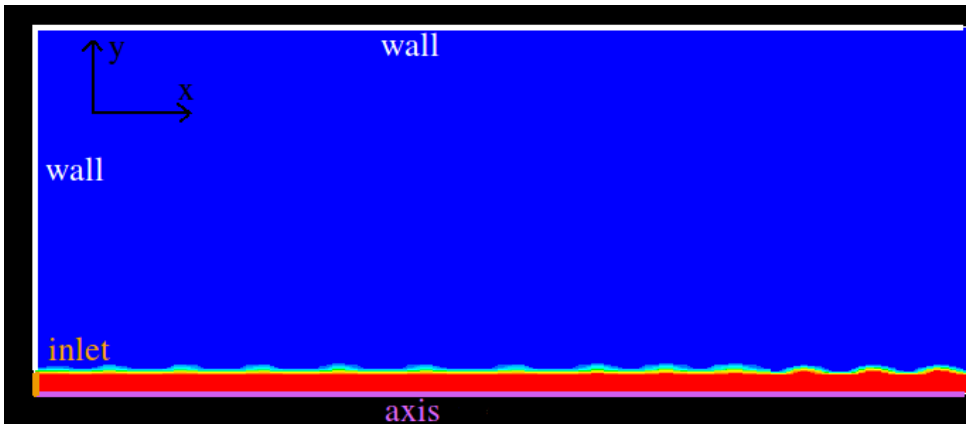


Figure 4.1: Qualitative results of breakup modelled with volume of fluid technique (red zone is $\alpha^l = 1$, blue zone is $\alpha^g = 1$)

4.0.2 Geometry, boundary conditions and mesh dependency

The geometry is a cylindrical chamber whose length and radius was chosen to avoid the situation in which the liquid phase hit the walls. To create the geometry and mesh for Fluent and OpenFoam the pre-processor ICEM was used. There are big differences between ICEM and Ansys Meshing[2]. Ansys Meshing does not guarantee the total control on the mesh parameters but it is useful when it is necessary to create a mesh in short time; it is more user friendly and is perfect for a new user. ICEM guarantees a total control on mesh's elements; for example it is possible to choose the number of nodes on each surface.

In ICEM before proceeding with the geometry it is necessary to define the units an the geometrical tolerance.

Settings → Model/Units

Meter was chosen as length unit and the minimum tolerance is set to $1e{-}07$. Every geometrical part smaller than the imposed tolerance is neglected. The used geometry is very simple. The first step to create the geometry is to define the points.

Geometry → Create Points → Explicit Coordinates

The inserted coordinates are itemized in the table 4.1 (or the table 4.2) The

[2]used to mesh the airfoil profil

point number	x [μm]	y [μm]
1st	0	0
2nd	0	50
3rd	0	1000
4th	5000	0
5th	5000	50
6th	5000	1000

Table 4.1: Points coordinate of the domain used to simulate and compare RANS breakup results with the DNS data

point number	x [μm]	y [μm]
1st	0	0
2nd	0	85
3rd	0	2000
4th	15000	0
5th	15000	85
6th	15000	2000

Table 4.2: Points coordinate of the domain used to simulate and compare RANS breakup results with the experimental data

geometry is concluded when the points are joined together in the shape of two rectangular blocks.

Geometry → Create Curve → From Points

Once the geometry is done it is necessary to link each line to its boundary name; for example line that joints point 1 and 2 is named *fuel inlet*.

Parts → Create Part → "Select the curve" → Apply → Rename part

The next step is to use blocking technique to create the mesh. A block referred to the geometry is created.

Blocking → Create Block → Type → 2D planar → Apply

The block must be split in two parts because the geometry is composed by two rectangles.

Blocking → Split Block → Select Edges → "select the *inlet fuel*" → Apply

Because the mesh is created on the blocks and not on the geometry the user must associate points to vertexes and edges to curves. Vertexes and curves

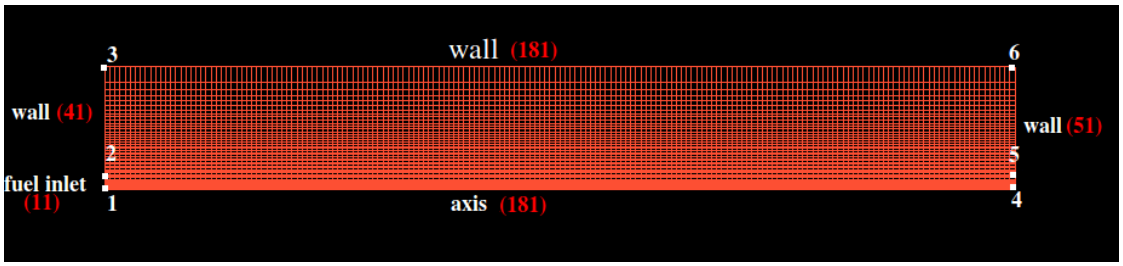


Figure 4.2: Mesh and the boundary conditions for breakup simulations

are part of the geometry while points and edges are parts of the blocks previously created.

Blocking → Associate → Associate Vertex → "select vertex" → "select point"

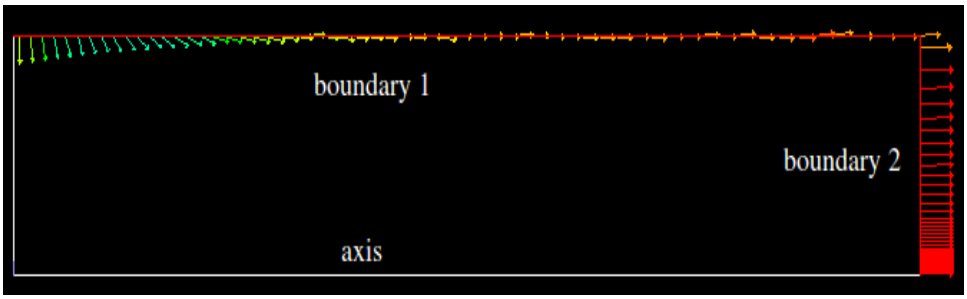
A similar operation must be followed for edges, it is sufficient to select in menu "Associate Edge to Curve" instead of "Associate vertex". To associate a vertex to a point it is sufficient to select with the left button of the mouse firstly the point and after the edge to whom it is associated. To associate an edge to a curves the operation is slightly different: the user must select the edge with left button of the mouse, accept the selection with the middle button of the mouse and than select the curve to which the edge is associated. Once the two blocks are associated to the two geometrical rectangles finally the user can proceeds with mesh generation. The number of nodes is imposed on each edge of the blocks

Blocking → Pre-Mesh Params → Edge Params → Edge → "select edge"

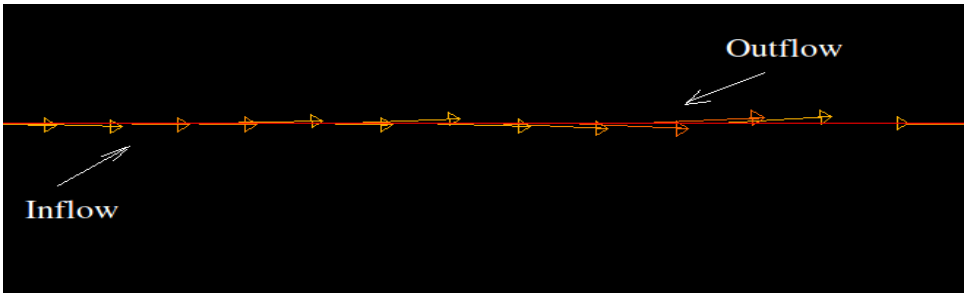
When the edge is selected it is possible to impose the number of nodes and the mesh law. In this case was used a *linear mesh law*. The size of elements increases from the symmetry axis to the top. The mesh, boundary conditions and the number of nodes (in red) of each surface are presented in the figure 4.2.

Geometry and mesh for OpenFoam simulations

Since the OpenFoam requires 3D mesh to perform the simulations there are to possibilities. The first possibility is to use the same mesh reated for fluent and to translate it in OpenFoam format using the command *FluentMeshToFoam* in the terminal. The second option is to create a mesh with *blockMeshDict*; in this case the y coordinates of the points in table 4.2 and table 4.1 have to be multiplied by $cos(5°)$ and $sin(5°)$ in order to obtain an y and z.



(a)

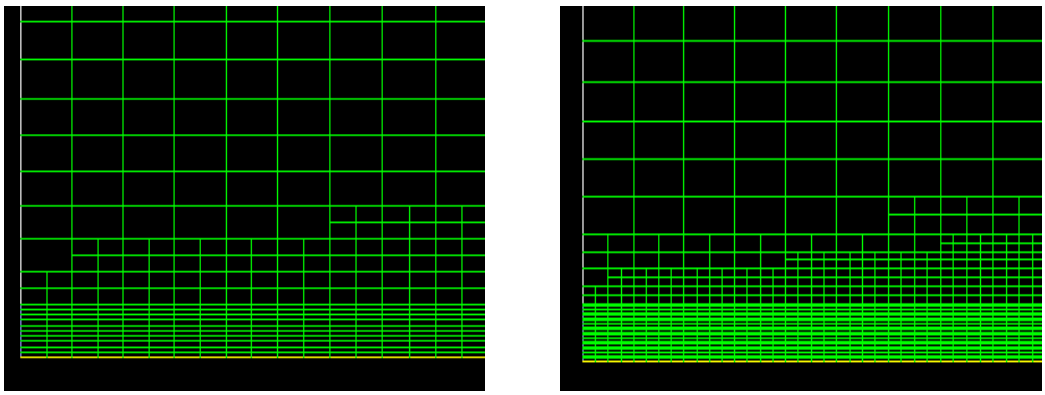


(b)

Figure 4.3: Possible problems with pressure outlet boundary conditions for break up simulations

Boundary Conditions for Breakup simulations

The boundary conditions of all the simulations have been chosen in the beginning and maintained constant for all the cases. The boundary conditions are reported in the figure 4.2. An alternative can seem to change the boundary wall condition with pressure outlet condition. This alternative is not recommended for break up simulation; indeed, the simulation with pressure outlet conditions, does not converge in terms of continuity and hence also in terms of volume fraction. The problems with the convergence is caused by wrong boundary conditions, Fluent call it "reversed flow". The figure 4.3 presents the velocity field of continuous gas phase on boundaries. The boundary 1 (upper surface) can not be defined as an outlet or inlet in unambiguous mode because in some regions the phase enters while in other exits from the computational domain. If the boundary 1 (figure 4.3) is changed in wall and boundary 2 remains pressure outlet reversed flow appears on the boundary when the dispersed phase approaches the outlet. Moving the outlet in the distance can avoid this situation but it will increase the number of cells and computational effort. For these reasons wall boundary conditions were chosen the most appropriate with the shrewdness to place the walls far



(a) second mesh (b) third mesh

Figure 4.4: Refined mesh for breakup simulations

enough to avoid the contact with liquid phase. Since the computational domain is big enough the pressure of gas phase does not increase considerably, the local maximum increment in absolute pressure is of 0.1% order.

Case sensitivity to the number of nodes

As a first attempt was created a mesh characterised by 60 nodes in axial direction and 20 nodes in the radial direction; 5 nodes were allocated on the inlet. This first mesh is characterized by 1586 cells. The mesh was refined in zones where was expected to find the liquid fraction. Other two meshes were created whose near inlet zone is showed in figure 4.4. The second mesh has 2265 elements while the third mesh has 4881 elements. The axial behaviour of the liquid fraction is showed in figure 4.5 From the plot of the volume fraction it is evident that the result change slightly from the second to the third mesh so for this reason the second mesh will be used in Fluent and OpenFoam simulations (to compare the results with DNS data). Because in the OpenFoam an explicit solver is used a constrain on the time step is imposed, for this reason the minimal cell size within mesh is very important since it is linked to the execution time. In all the simulations the mesh is coarse near the walls and with good resolution in the in the zone near the symmetry axis.

4.1 Breakup OpenFoam

To solve multiphase problems *OpenFoam* provides solvers suitable for different physical problems. The solver *twoPhaseEulerFoam* was considered to be

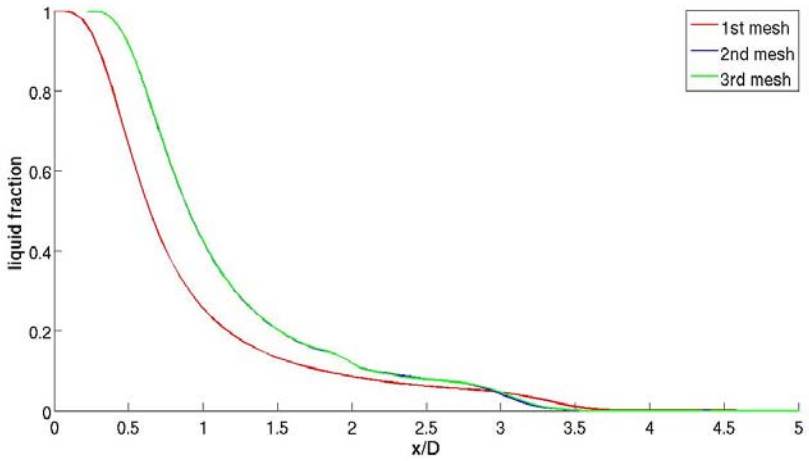


Figure 4.5: Axial behaviour of the liquid volume fraction considering different meshes

the best choice for this purpose. The solver includes:

- compressibility effects;
- heat-transfer;
- generalised two-phase turbulence modelling, including specialised dispersed-phase generation terms
- generalised run-time model selection
- new models, mainly for flows containing gas bubble.

Since the OpenFoam solver is explicit it is important to maintain Courant number beneath a unity value. In all the simulations in OpenFoam the Courant number was less than 0.5. As explained in aerofoil chapter three folders contain all the information about the case and the solver. Since liquid jet breakup involves two different fluid phases (liquid and the gas) it is necessary to define the characteristics of both phases and also the characteristics of the mixture.

- **folder 0** this folder include the information at time $t = 0$ seconds hence the initial condition and boundary conditions. In this case 17 files were stored within folder:
  - *file alpha.air, alpha.fuel*: concern the air and fuel phase fraction.

  - *epsilon.air, epsilon.fuel, epsilonm*: concern ϵ parameter respectively for gas, liquid and mixture.
  - *k.air, k.fuel, km* concern the turbulent kinetic energy.
  - *nut.air, nut.fuel* concern the turbulence viscosity
  - p concern the pressure; the pressure is calculated considering the ideal gas expression
  - p_rgh concern the contribution of the term $\rho \cdot g \cdot h$ zero in our case because the gravity contribution was not considered $g = 0$
  - *T.air, T.fuel* concern the temperature of the two phases
  - *U.air, U.fuel* concern the velocity of gaseous and liquid phase

- **folder constant** contains the information about phase properties hence how the phase interacts one with each other, thermophysical and turbulence properties for each phase and the gravity file. Six files were stored in this folder:
  - the file g contains the information about the gravity field
  - *phaseProperties* file contains the information about the way of interacting between phases. Drag model, virtual mass model heat transfer model, lift model, wall lubrication model and turbulent dispersion model were specified.
  - *thermophysicalProperties.air, thermophysicalProperties.fuel* holds the information about thermal model and mixture constants
  - *turbulenceProperties.air, turbulenceProperties.fuel* holds the information about the turbulence model

- **folder system** contains inside the usual files *blockMesh, controlDict, fvSchemes, fvSolution*. The nature of this files has been already discussed within previous chapter.

  In OpenFoam it is always important to check the dimensions of the parameters because different solvers are implemented differently. An example is the pressure input file in folder **0**. Indeed the solver *simpleFoam* requires in input the ratio between pressure and density $\frac{P}{\rho}$ while the *twoPhaseEulerFoam* requires in input pressure P.

Property	Fuel
Density $[\frac{kg}{m^3}]$	696
Viscosity $[\frac{kg}{ms}]$	0.0012
Surface Tension $[\frac{N}{m}]$	0.06

Table 4.3: Fuel properties used in DNS simulations

Velocity $[\frac{m}{s}]$	100
Injector diameter $[\mu m]$	100
Turbulent length scale $[\mu m]$	10
Turbulent intensity	0.05
Gas density $\frac{kg}{m^3}$	50

Table 4.4: Characteristics of liquid injection in DNS simulation

4.1.1 Comparison of RANS data with DNS data

The DNS fuel properties are listed in the table 4.3 The characteristics of the liquid injection are presented in the table 4.4. These data were used to set up the geometry and boundary conditions in RANS simulations.

Particle Diameter

OpenFoam imposes the user to define the particle diameter for all the phases even in a case of only two phases with one that acts as continuous. Different sizes where used for the simulations. Firstly turbulent dispersion model was switched off and the flow was simulated with several combinations of the two particle diameters. The converged combinations are presented in table 4.5 where D_{air} and D_{fuel} is respectively air and fuel particle's diameters. It was noticed that the diameter of gas must be at least 10 times the diameter of the liquid particle to avoid the divergence. An exception is the case 2 in

combination	D_{fuel} $[\mu m]$	D_{air} $[\mu m]$
case 1	$1 \cdot 10^{-7}$	$1 \cdot 10^{-6}$
case 2	$1 \cdot 10^{-6}$	$3 \cdot 10^{-6}$
case 3	$1 \cdot 10^{-6}$	$1 \cdot 10^{-6}$
case 4	$2 \cdot 10^{-6}$	$2 \cdot 10^{-5}$
case 5	$3 \cdot 10^{-6}$	$4 \cdot 10^{-5}$
case 6	$4 \cdot 10^{-6}$	$6 \cdot 10^{-5}$

Table 4.5: Particle diameter for air and fuel phase

case	dispersion model	model constant	D_{fuel} $[\mu m]$	D_{air} $[\mu m]$	note
1	Lopez de Bertodano	$Ctd = 0.1$	$1 \cdot 10^{-6}$	$3 \cdot 10^{-6}$	N
2	Lopez de Bertodano	$Ctd = 0.2$	$1 \cdot 10^{-6}$	$3 \cdot 10^{-6}$	N
3	Lopez de Bertodano	$Ctd = 0.4$	$1 \cdot 10^{-6}$	$3 \cdot 10^{-6}$	N
4	Lopez de Bertodano	$Ctd = 0.5$	$1 \cdot 10^{-6}$	$3 \cdot 10^{-6}$	D
5	Lopez de Bertodano	$Ctd = 0.6$	$1 \cdot 10^{-6}$	$3 \cdot 10^{-6}$	N
6	Lopez de Bertodano	$Ctd = 0.5$	$2 \cdot 10^{-6}$	$2 \cdot 10^{-5}$	Y
7	Lopez de Bertodano	$Ctd = 0.6$	$2 \cdot 10^{-6}$	$2 \cdot 10^{-5}$	Y
8	Lopez de Bertodano	$Ctd = 0.7$	$2 \cdot 10^{-6}$	$2 \cdot 10^{-5}$	Y
9	Lopez de Bertodano	$Ctd = 0.8$	$2 \cdot 10^{-6}$	$2 \cdot 10^{-5}$	Y
10	Lopez de Bertodano	$Ctd = 0.9$	$2 \cdot 10^{-6}$	$2 \cdot 10^{-5}$	D
11	Lopez de Bertodano	$Ctd = 0.4$	$3 \cdot 10^{-6}$	$4 \cdot 10^{-5}$	Y
12	Lopez de Bertodano	$Ctd = 0.5$	$3 \cdot 10^{-6}$	$4 \cdot 10^{-5}$	Y
13	Lopez de Bertodano	$Ctd = 0.6$	$3 \cdot 10^{-6}$	$4 \cdot 10^{-5}$	Y
14	Lopez de Bertodano	$Ctd = 0.7$	$3 \cdot 10^{-6}$	$4 \cdot 10^{-5}$	Y
15	Lopez de Bertodano	$Ctd = 0.8$	$3 \cdot 10^{-6}$	$4 \cdot 10^{-5}$	Y
16	Lopez de Bertodano	$Ctd = 0.9$	$3 \cdot 10^{-6}$	$4 \cdot 10^{-5}$	Y
17	Lopez de Bertodano	$Ctd = 1$	$3 \cdot 10^{-6}$	$4 \cdot 10^{-5}$	Y
18	Lopez de Bertodano	$Ctd = 1.1$	$3 \cdot 10^{-6}$	$4 \cdot 10^{-5}$	D
19	Lopez de Bertodano	$Ctd = 0.7$	$4 \cdot 10^{-6}$	$6 \cdot 10^{-5}$	Y
20	Lopez de Bertodano	$Ctd = 0.8$	$4 \cdot 10^{-6}$	$6 \cdot 10^{-5}$	Y
21	Lopez de Bertodano	$Ctd = 0.9$	$4 \cdot 10^{-6}$	$6 \cdot 10^{-5}$	Y
22	Lopez de Bertodano	$Ctd = 1$	$4 \cdot 10^{-6}$	$6 \cdot 10^{-5}$	Y
23	Lopez de Bertodano	$Ctd = 1.1$	$4 \cdot 10^{-6}$	$6 \cdot 10^{-5}$	Y
24	Lopez de Bertodano	$Ctd = 1.2$	$4 \cdot 10^{-6}$	$6 \cdot 10^{-5}$	D

Table 4.6: Lopez de Bertodano dispersion model settings

the table 4.5; in this case the air particle diameter is only 3 times bigger than the liquid particle diameter. This behaviour was confirmed also when the dispersion model was switched on. Was tried also a bigger diameter for fuel particles as for example $5 \cdot 10^6$ but divergence was detected with the following air particle diameters: $7 \cdot 10^{-5}$, $9 \cdot 10^{-5}$, 10^{-4}, $2 \cdot 10^{-4}$, $3 \cdot 10^{-4}$, $4 \cdot 10^{-4}$. It was decided to execute the simulation switching on the dispersion model and using the combinations of diameters listed in the table 4.5. The tables, 4.6, 4.7 and 4.8 summarize the turbulence dispersion settings used for simulations. The meaning of the notes in the previously mentioned tables are:

- **Y**: no divergence was detected and was noticed the primary break up

case	dispersion model	model constant	D_{fuel} $[\mu m]$	D_{air} $[\mu m]$	note
1	constant	$Ctd = 0.5$	$1 \cdot 10^{-6}$	$3 \cdot 10^{-6}$	N
2	constant	$Ctd = 0.7$	$1 \cdot 10^{-6}$	$3 \cdot 10^{-6}$	N
3	constant	$Ctd = 1$	$1 \cdot 10^{-6}$	$3 \cdot 10^{-6}$	N
4	constant	$Ctd = 5$	$1 \cdot 10^{-6}$	$3 \cdot 10^{-6}$	Y
5	constant	$Ctd = 7$	$1 \cdot 10^{-6}$	$3 \cdot 10^{-6}$	Y
6	constant	$Ctd = 10$	$1 \cdot 10^{-6}$	$3 \cdot 10^{-6}$	Y
7	constant	$Ctd = 15$	$1 \cdot 10^{-6}$	$3 \cdot 10^{-6}$	Y
8	constant	$Ctd = 2$	$2 \cdot 10^{-6}$	$2 \cdot 10^{-5}$	Y
9	constant	$Ctd = 2.4$	$2 \cdot 10^{-6}$	$2 \cdot 10^{-5}$	Y
10	constant	$Ctd = 3$	$2 \cdot 10^{-6}$	$2 \cdot 10^{-5}$	Y
11	constant	$Ctd = 7$	$2 \cdot 10^{-6}$	$2 \cdot 10^{-5}$	Y
12	constant	$Ctd = 10$	$2 \cdot 10^{-6}$	$2 \cdot 10^{-5}$	Y
13	constant	$Ctd = 1$	$3 \cdot 10^{-6}$	$4 \cdot 10^{-5}$	Y
14	constant	$Ctd = 1.5$	$3 \cdot 10^{-6}$	$4 \cdot 10^{-5}$	Y
15	constant	$Ctd = 2$	$3 \cdot 10^{-6}$	$4 \cdot 10^{-5}$	Y
16	constant	$Ctd = 2.4$	$3 \cdot 10^{-6}$	$4 \cdot 10^{-5}$	Y
17	constant	$Ctd = 3$	$3 \cdot 10^{-6}$	$4 \cdot 10^{-5}$	Y
18	constant	$Ctd = 1$	$4 \cdot 10^{-6}$	$6 \cdot 10^{-5}$	Y
19	constant	$Ctd = 1.5$	$4 \cdot 10^{-6}$	$6 \cdot 10^{-5}$	Y
20	constant	$Ctd = 2$	$4 \cdot 10^{-6}$	$6 \cdot 10^{-5}$	Y
21	constant	$Ctd = 2.4$	$4 \cdot 10^{-6}$	$6 \cdot 10^{-5}$	Y
22	constant	$Ctd = 3$	$4 \cdot 10^{-6}$	$6 \cdot 10^{-5}$	Y

Table 4.7: constant dispersion model setting

case	dispersion model	model constant	D_{fuel} $[\mu m]$	D_{air} $[\mu m]$	note
1	Burns	$\sigma = 1$	$1 \cdot 10^{-6}$	$3 \cdot 10^{-6}$	D
1	Burns	$\sigma = 1.5$	$1 \cdot 10^{-6}$	$3 \cdot 10^{-6}$	D
1	Burns	$\sigma = 2$	$1 \cdot 10^{-6}$	$3 \cdot 10^{-6}$	D
1	Burns	$\sigma = 2.5$	$1 \cdot 10^{-6}$	$3 \cdot 10^{-6}$	D
1	Burns	$\sigma = 3$	$1 \cdot 10^{-6}$	$3 \cdot 10^{-6}$	D
2	Burns	$\sigma = 2$	$1 \cdot 10^{-6}$	$1 \cdot 10^{-5}$	Y
3	Burns	$\sigma = 2.5$	$1 \cdot 10^{-6}$	$1 \cdot 10^{-5}$	Y
4	Burns	$\sigma = 2.7$	$1 \cdot 10^{-6}$	$1 \cdot 10^{-5}$	Y
5	Burns	$\sigma = 2.8$	$1 \cdot 10^{-6}$	$1 \cdot 10^{-5}$	Y
6	Burns	$\sigma = 2.9$	$1 \cdot 10^{-6}$	$1 \cdot 10^{-5}$	Y
7	Burns	$\sigma = 3$	$1 \cdot 10^{-6}$	$1 \cdot 10^{-5}$	Y
8	Burns	$\sigma = 1.5$	$2 \cdot 10^{-6}$	$2 \cdot 10^{-5}$	Y
9	Burns	$\sigma = 2$	$2 \cdot 10^{-6}$	$2 \cdot 10^{-5}$	Y
10	Burns	$\sigma = 2.5$	$2 \cdot 10^{-6}$	$2 \cdot 10^{-5}$	Y
11	Burns	$\sigma = 2.7$	$2 \cdot 10^{-6}$	$2 \cdot 10^{-5}$	Y
12	Burns	$\sigma = 1.5$	$3 \cdot 10^{-6}$	$4 \cdot 10^{-5}$	Y
13	Burns	$\sigma = 2$	$3 \cdot 10^{-6}$	$4 \cdot 10^{-5}$	Y
14	Burns	$\sigma = 2.5$	$3 \cdot 10^{-6}$	$4 \cdot 10^{-5}$	Y
15	Burns	$\sigma = 2.7$	$3 \cdot 10^{-6}$	$4 \cdot 10^{-5}$	Y

Table 4.8: Burns dispersion model setting

- **N**: no divergence was detected but no break up was observed
- **D**: divergence was detected

In *phaseProperties dictionary* the user can specify the blending model for the two phases. If the blending model is switched off the drag is done using the dispersed phase. If the blending model is switched on the drag is independently calculated with each phase as "dispersed phase", the overall drag contribution applied to the momentum equations is volume fraction weighted average of the two values. The blending scheme is useful approximation for flows with regions in which either phase is primary phase as in cases studied in this work. In all the simulations the blending model was activated. To evaluate the error in liquid volume fraction the DNS, experimental and RANS data concerning volume fraction were interpolated by a polynomial functions. Let consider $f(s)$ the polynomial that interpolate DNS data and $g(s)$ the polynomial that interpolate RANS data. To compare $g(s)$ and $f(s)$ over the interval [a,b] it is possible to consider the inner product of the two functions:

$$< f(s), g(s) >= \int_a^b f(s)g(s)ds$$

from which the error between the two curves is:

$$||f(s) - g(s)|| = \sqrt{\int_a^b (f(s) - g(s))^2 ds} \tag{4.1}$$

where $f(s)$ and $g(s)$ give the value of fuel volume fraction for any s position. The tables 4.9, 4.10 and 4.11 give the approximate error related to phase fraction for different dispersion models:

- the first column gives the error in axial direction considering a 20 mm distance from the inlet
- the second column gives the error in radial direction at 0.5 mm from the injector
- the third column gives the error in radial direction at 10 mm from the injector
- the fourth column gives the error in radial direction at 20 mm from the injector
- the fifth column is the error sum of previous errors

case	axial	$x = 5 \cdot D$	$x = 10 \cdot D$	$x = 20 \cdot D$	sum
6	17.32	1.09	4.83	8.29	31.54
7	14.13	1.12	4.20	6.49	25.95
8	13.79	0.57	2.46	5.03	21.85
9	10.23	0.50	1.98	4.35	17.07
11	13.03	1.36	4.97	5.09	24.45
12	10.37	1.49	3.99	4.63	20.50
13	5.41	0.90	1.21	4.17	11.70
14	4.88	3.74	1.99	3.40	14.02
15	3.44	0.63	2.68	3.05	9.80
16	2.67	0.69	1.73	2.57	7.67
17	3.29	2.82	2.06	2.64	10.81
19	3.77	2.42	2.45	2.91	11.55
20	4.43	0.91	2.75	2.70	10.79
21	7.47	0.88	3.58	3.20	15.13
22	7.95	0.97	3.04	3.34	15.31
23	9.71	3.74	3.76	3.57	20.77

Table 4.9: Error with Lopez de Bertodano model

From these tables the case 16 of Lopez de Bertodano model gives the smallest error compared with DNS data. Taking any model apart the best results are:

- *Lopez de Bertodano model* case 19 with an axial error equal to 2.67 and a total error equal to 7.70
- *Constant model* case 8 with an axial error equal to 2.62 and a total error equal to 10.79
- *Burns model* case 3 with an axial error equal to 3.83 and a total error equal to 9.36

The figure 4.6 presents the plot of fuel volume fraction in the position where RANS simulations were compared with DNS simulations. As it can be noticed from the plot 4.6 the Burns model presents a more regular behaviour than the other two models. The figure 4.7 presents qualitative RANS and DNS results at $t = 0.01\ ms$. The temporal behaviour shows that Burns model presents less oscillations on liquid surface in time compared with other two models; also in the simulations with Burns model the liquid shape is very uniform. In the simulations where the dispersions is modelled with Lopez de

case	axial	$x = 5 \cdot D$	$x = 10 \cdot D$	$x = 20 \cdot D$	sum
4	3.23	2.36	3.55	5.48	14.63
5	7.78	2.60	3.96	4.64	18.98
6	10.61	2.14	5.24	4.39	22.39
7	12.47	3.09	5.22	3.47	24.25
8	2.62	0.92	2.78	4.46	10.79
9	6.54	1.60	3.84	4.58	16.58
10	5.83	0.85	3.53	4.87	15.08
11	20.10	8.60	4.51	4.23	37.44
12	23.35	8.83	7.40	4.25	43.84
13	7.027	1.14	4.27	6.14	18.58
14	2.50	1.96	3.59	3.69	11.74
15	8.09	2.07	4.54	4.26	18.96
16	33.93	1.71	3.70	4.51	43.87
17	14.44	3.45	5.92	4.51	28.32
18	4.28	1.20	3.75	4.70	13.94
19	4.70	1.02	3.53	4.17	13.42
20	8.57	1.20	4.32	3.26	17.35
21	11.44	2.57	4.64	3.96	22.60
22	14.55	4.60	5.55	3.87	28.59

Table 4.10: Error with Constant model

case	axial	$x = 5 \cdot D$	$x = 10 \cdot D$	$x = 20 \cdot D$	sum
2	4.31	1.74	2.06	2.18	10.29
3	3.82	1.35	2.16	2.02	9.36
4	4.22	1.27	2.27	2.07	9.82
5	4.47	1.24	2.34	2.11	10.16
6	4.75	1.21	2.40	2.17	10.54
7	5.04	1.20	2.47	2.24	10.93
8	12.79	4.17	4.45	2.37	23.78
9	9.26	2.91	3.61	2.24	18.02
10	6.56	2.16	4.59	2.34	15.65
11	5.82	1.97	3.07	2.45	13.31
12	12.32	4.08	4.32	2.43	23.15
13	9.22	2.92	3.76	2.52	18.42
14	6.88	2.27	3.50	2.82	15.46
15	6.20	2.11	3.48	2.98	14.77

Table 4.11: Error with Burns model

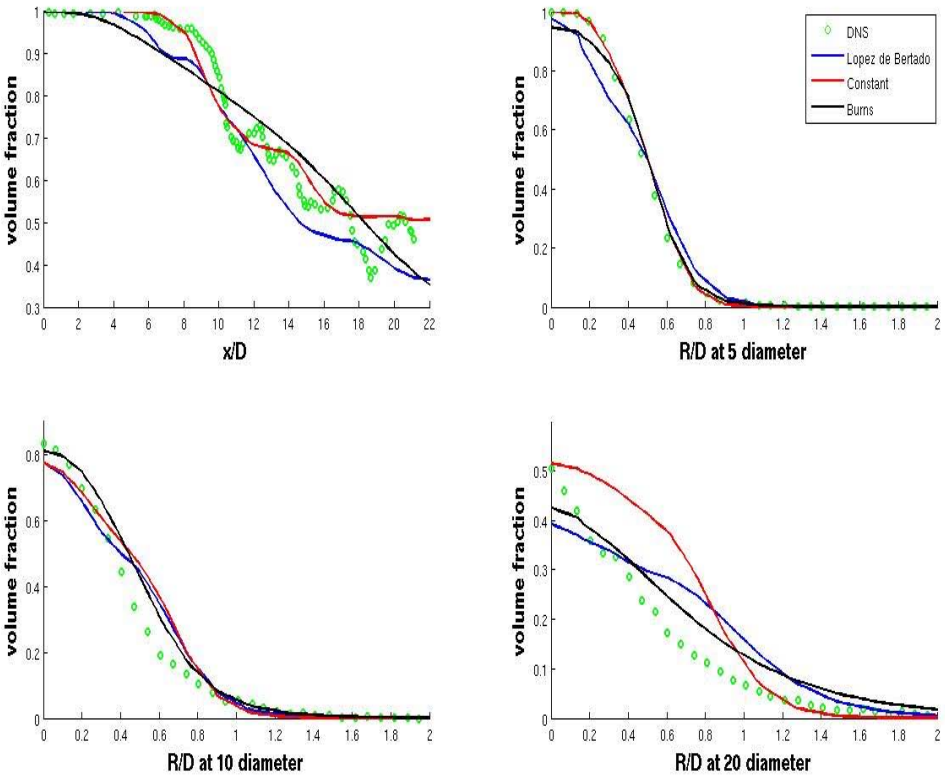
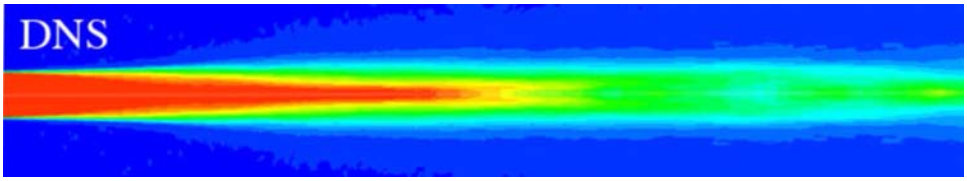
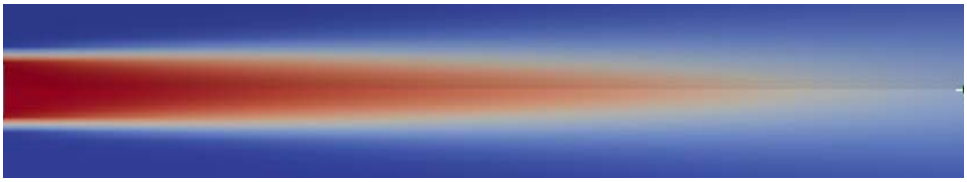


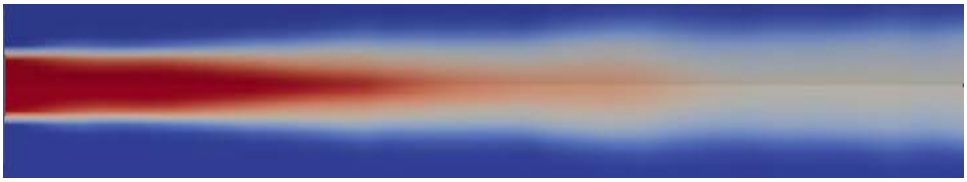
Figure 4.6: plot of volume fraction and comparison with DNS data



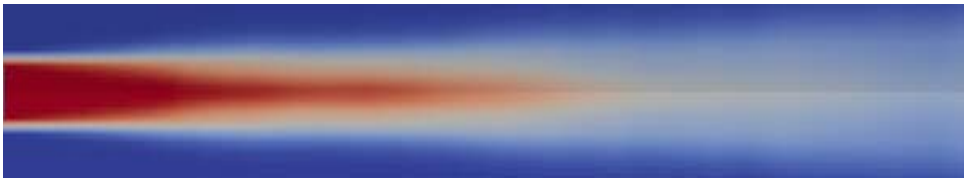
(a) DNS



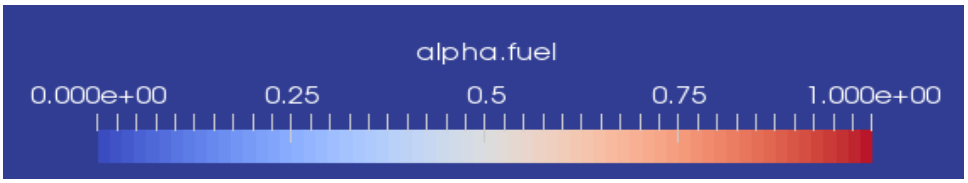
(b) Burns model



(c) Constant model



(d) Lopez de Bertodano model



(e) Rans simulations colour map

Figure 4.7: Qualitative behaviour of the volume fraction obtained with RANS simulations and compared with DNS simulation at time $t = 0.1\ ms$

Property	Viscor Cerium blend
Density ρ_f $[\frac{kg}{m^3}]$	865.4
Viscosity $[\frac{kg}{ms}]$	0.0029
Surface Tension $[\frac{N}{m}]$	0.026
Vapor pressure $[Pa]$	1057

Table 4.12: Fuel properties at 317 K

Bertodano model there is a separation of a conspicuous quantity of liquid in the first instants of the simulation. This quantity preserves a high value of liquid fraction till $t = 0.1\ ms$. The zone between the injector and this liquid quantity is characterized by small fluctuations on the inter-facial surface. The Constant turbulent dispersion model has a behaviour intermediary between Lopez de Bertodano model and Burns model. Indeed this model presents fluctuations on the surface of the dispersed liquid cone but no high fraction zone that precedes the spray is observed. Taking in consideration the liquid penetration, Burns model has a bigger penetration length than other two models while the Lopez de Bertodano model the smallest one. However data concerning liquid penetration are not provided with DNS simulation. Lopez de Bertodano model presents the smallest error in terms of liquid fraction distribution.

4.1.2 Experimental data Vs Rans Simulations

Once the dispersion models were set and results with tolerable errors were obtained the next step consists in understanding if the dispersion model's setting is independent or not from the boundary conditions. New data are necessary for this purpose. In this case experimental data were used to compare the Rans results. The experimental data are collected in the paper *Quantitative X-ray measurements of high pressure fuel sprays from production heavy duty diesel injector* by A.I.Ramìrez, S.Som, Suresh K. Aggarwal and others. To check the behaviour of the models the settings of the best configurations in tables 4.9, 4.10 and 4.11 were kept while the boundary conditions were changed to overlap the experimental configuration. New simulations with the new BC were carried out and comparisons with experimental data were performed. The experimental fuel properties are presented in the table 4.12 while the characteristics of liquid injection are presented in 4.13. Comparing the tables 4.12 and 4.13 with 4.4 and 4.3 the liquid properties and injection diameter are similar. However the velocity of DNS data is $100\frac{m}{s}$ while the experimental velocity varies from 0 to $240\frac{m}{s}$. Experimentally the X-ray radiography technique has been used to investigate the behaviour of fuel sprays

Velocity $[\frac{m}{s}]$	varies from 30 to 250
Injector diameter D_f $[\mu m]$	169
Gas density ρ_g $\frac{kg}{m^3}$	34.13

Table 4.13: Characteristics of liquid injection in DNS simulation

in the near injector region. The experimental 21 MPa oil rail pressure case was chosen to validate the RANS data. The paper presents data concerning the mass flow rate which was used to impose the boundary conditions. The results obtained with Rans respect the liquid penetration and liquid volume fraction behaviour are compared with provided experimental data.

Boundary conditions

The boundary conditions are fundamentals for the simulations, changing BC changes is equivalent to change the physical situation. All the simulations treated in this subsection has the following computational domain: a cylindrical box with axial length equivalent to 15 mm and the height from the symmetry axis equal to 4 mm. The same type of boundary conditions was maintained as in previous case: axis, inlet and wall. The big difference consists in the conditions applied at the inlet. The inlet conditions of the RANS simulations are reportable to experimental data in figure 4.8 The figure 4.8 provides the flow rate in some exact instances but also a linear interpolation of this data. From the figure 4.8 and the table 4.12 and 4.13 in hypothesis of $\rho_f = constant$ it is possible to extract the velocity using:

$$U(t) = \frac{\dot{m}(t)}{\rho_f \pi \frac{D_f^2}{4}}$$

The figure 4.9 is the plot of velocity profile of the case characterized by an oil pressure of 21 MPa. The curve in blue is the polynomial function that fits the experimental data. The line in red is the equivalent of the red line in figure 4.8.

Simulations

The aim of the simulation is to verify the capability to predict the radial and axial liquid phase fraction. The experimental data of the liquid phase behaviour are provided in $\frac{\mu g}{mm^2}$ and non in volume phase fraction as in the case of DNS simulations. The experimental measurements of liquid fraction obtained with X-ray radiography are defined as the density of the spray (in mass per unit volume) integrated over the path length of the beam giving

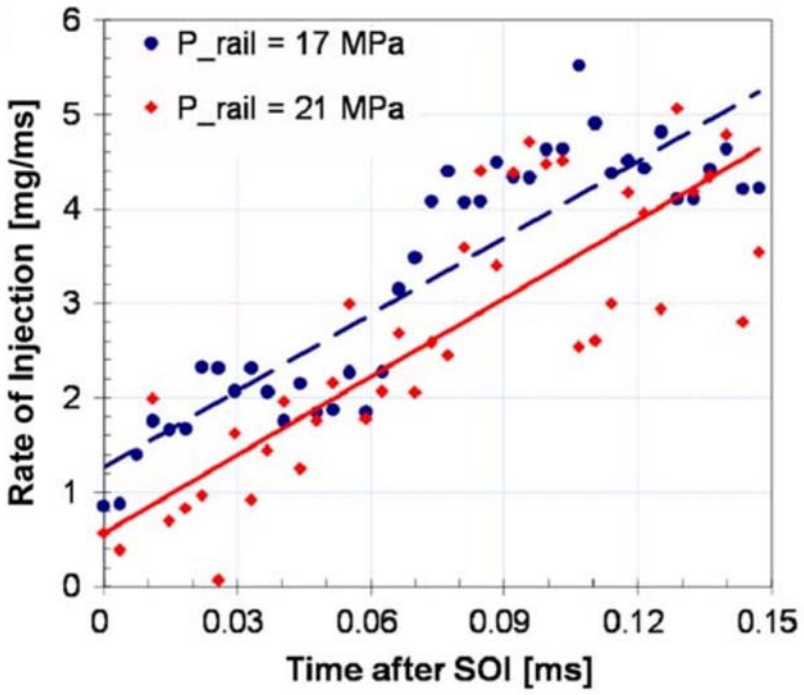


Figure 4.8: Rate of injection from experiments

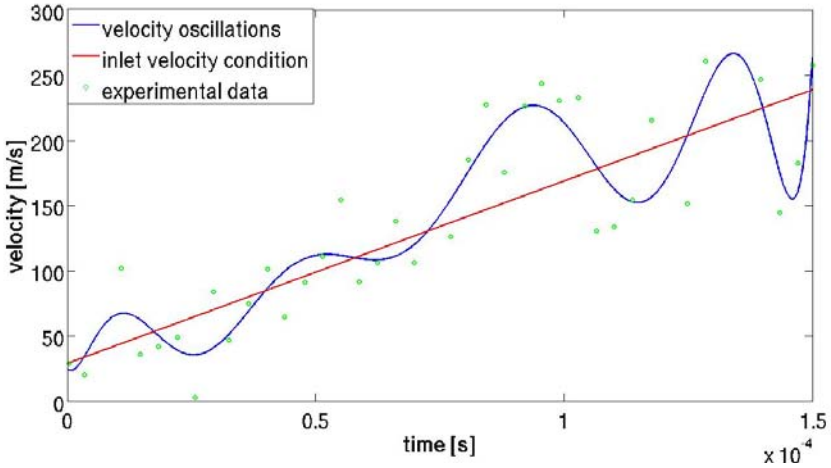


Figure 4.9: Velocity profile at the injector of experimental case

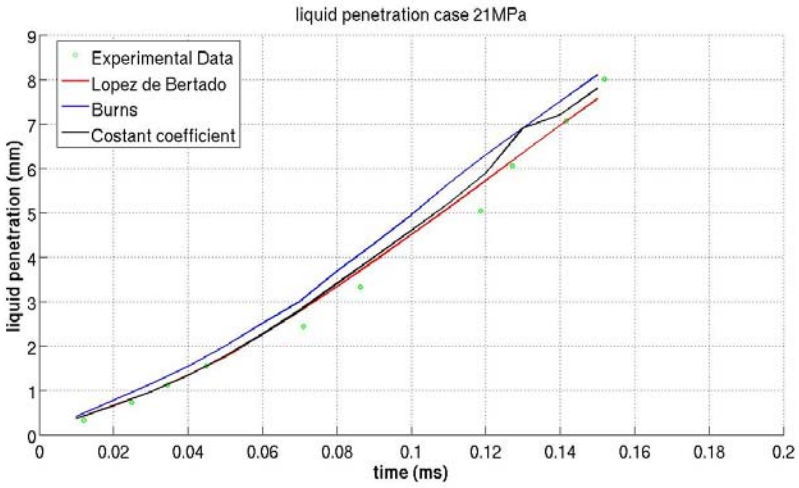


Figure 4.10: liquid penetration for 5% turbulence intensity

data in mass per unit area. Since the fuel and gas density are constant it is expected that the behaviour of the volume phase fraction is similar to the behaviour of the projected density quantity.

First of all new simulations were carried out with the new boundary conditions. The boundary condition at the inlet was imposed as *velocity inlet* where the value of velocity is reported in the figure 4.9. Holding fixed the inlet condition different turbulence properties were imposed:

- **first case** the turbulence parameters were kept equal to the DNS data so turbulence velocity equal to 5% of the averaged velocity and turbulence length scale equal to 10% of the diameter.

- **second case**: the figure 4.9 marks the fluctuations of the fuel velocity at the outlet of the injector. It is possible to link these fluctuations to the turbulence velocity u_{rms} indeed the fluctuation velocity can be defined as $u_{rms} = |U_2 - U_1|$ where U_2 is the real experimental velocity (green dots) while U_1 is the velocity extracted from the red line. Since the instantaneous behaviour of the velocity at any instant is unknown it is possible to assume as first approximation that u_{rms} is equal to 50% of U_1

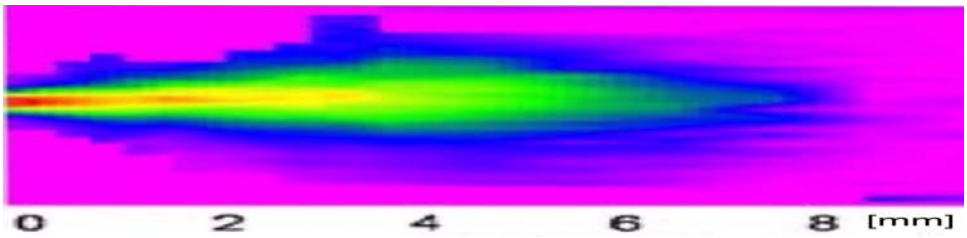
The simulation time is $t = 0.15\ ms$.

First case: turbulent intensity $u_{rms} = 5\%U_1$ The results concerning the liquid penetration are showed in figure 4.10. The liquid penetration was valued as the axial position, for any time step, where the liquid volume

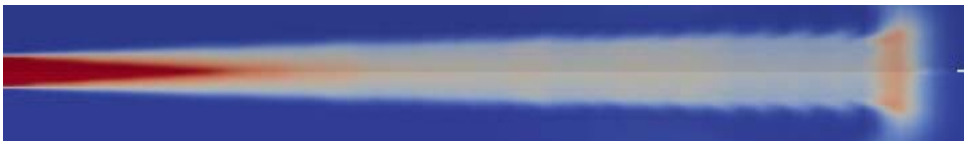
fraction $\alpha = 0.05$. As it can be noticed *Lopez de Bertodano model* presents the best results while *Burns model* has the biggest error.

If the qualitative liquid phase distribution is examined it is noticed that the cone aperture of the liquid phase in RANS simulations is lower than the experimental.

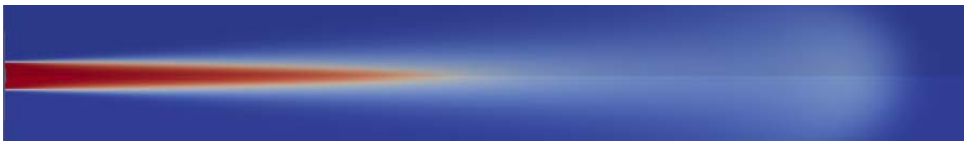
Turbulent intensity 50% The figure 4.11 shows the qualitative liquid phase distribution a time $t = 0.15\ ms$ obtained in experiments and with RANS simulations. A clear difference between experimental data and Lopez de Bertodano (and Constant coefficient model) is noticed. In Rans simulations an accumulation of liquid with high volume fraction proceed first; this is not noticed in experiments. This is the reason why the two dispersion models are considered not suitable in break up RANS simulations with a high value of turbulence intensity. On the other hand the Buns model presents a liquid distribution very similar to the experimental behaviour. This is the reason why only Burns model is considered in the next section about the influence of boundary and initial conditions on RANS simulations . The figure 4.12 compares the RANS liquid penetration at turbulence intensity equal to 50% with the experimental data. The plot marks a small error compared with experimental data till $t = 0.07\ ms$ but from this instant the error begin to increase. As a result, a bigger error is obtained in liquid penetration if the turbulence intensity is increased by an order of 10. The liquid penetration of the case with $I = 50\%$ is less than 6 mm. If the distribution of liquid fraction obtained with experiments and simulations is compared, a smaller error is noticed in simulations with $I = 50\%$ than in simulations with $I = 5\%$. So an enhancement of turbulence intensity causes a bigger cone opening. This behaviour is showed in figure 4.13 where is plotted the liquid distribution in radial direction at the position $x = 4.5\ mm$. From figure 4.11, 4.13 and 4.10 emerges that the high value of turbulence intensity increases the cone aperture but decreases the liquid penetration. Finally it is important to recall that the turbulence intensity at the inlet in the previous simulations is constant.



(a) Experimental data

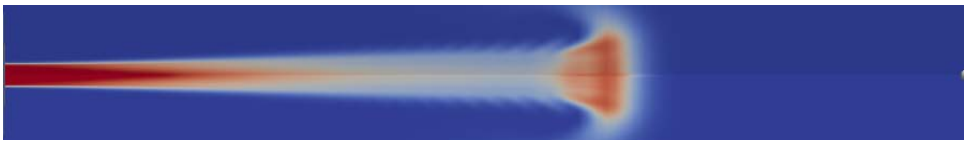


(b) Constant coefficient $I = 5\%$

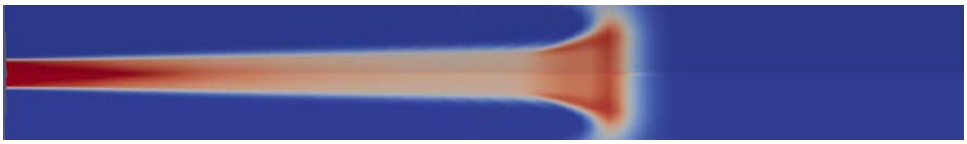


(c) Burns $I = 5\%$

(d) Lopez $I = 5\%$



(e) Lopez $I = 50\%$



(f) Constant coefficient $I = 50\%$

(g) Burns $I = 50\%$

(h) Rans simulations colour map

Figure 4.11: Rans simulations with different inlet turbulence intensity compared with experimental data at time $t = 0.15\ ms$

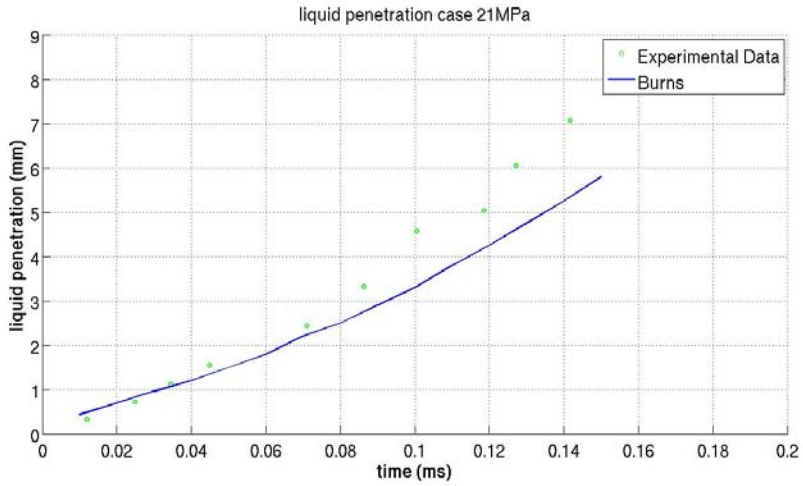


Figure 4.12: liquid penetration for 50% turbulence intensity

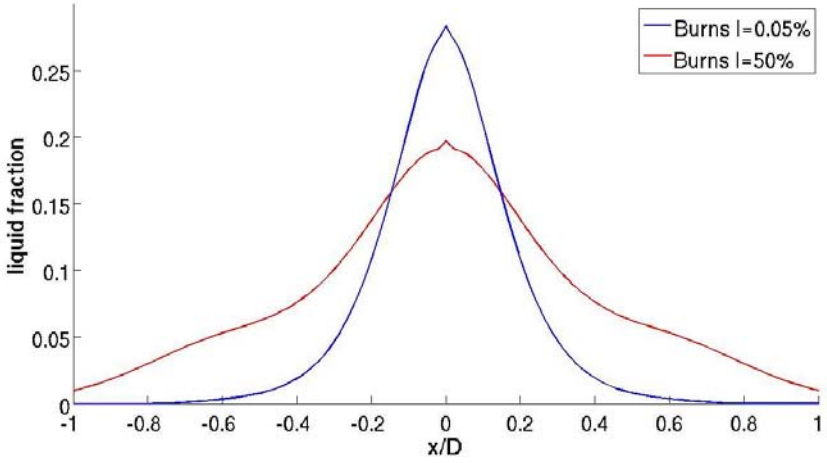


Figure 4.13: Liquid fraction calculated with RANS Burns model at different values of turbulence intensity defined on inlet.

4.1.3 The importance of boundary conditions

In the previous section was showed that, in hypothesis of constant turbulence parameters[3], raising turbulence intensity causes a smaller liquid penetration length but a bigger liquid cone aperture. The real velocity profile in figure 4.9 has a periodic behaviour. It is possible to interpolate the real velocity behaviour with a polynomial function and a linear function. If $U_3(t)$ is the velocity obtained with the polynomial function and $U_1(t)$ is the velocity obtained with linear interpolation of real velocity than $u(t)_{rms} = |U_3(t) - U_1(t)|$. Estimating u_{rms} in a such way, it is possible to impose turbulence boundary conditions as follows:

- the turbulence intensity value $I(t) = \frac{u(t)_{rms}}{U_1(t)}$. $I(t)$ was bounded by a minimum value of $I = 0.05$ and a maximum value of $I = 0.5$ to avoid problems with convergence
- the turbulence length scale value $L(t)$ is calculated from a linear function created on the assumption that $L = 0.1 \cdot D$ if $I(t) = 0.05$ and $L = D$ if $I(t) = 0.5$

Only Burns model was used for these simulations. The figures 4.15 and 4.14 exhibits the radial phase volume fraction behaviour and the liquid penetration length. The word transient is referred to the varying value of intensity.

It is possible to mark that the liquid penetration behaviour seems to interpolate the experimental data. The simulation also manifests a good cone aperture. The main difference between the simulation with transient and $I = 50\%$ is that the turbulence intensity has not a high value in the first instants of the injection. Because of relatively small turbulence intensity value (circa 20%) at $t = 0\ s$ the liquid penetration is acceptable. Indeed other simulations were performed. It was noticed that starting with a smaller value of turbulence intensity and than increasing I in the next instants bring to better results rather starting with a high value of turbulence intensity.

[3] turbulence intensity and turbulence length scale

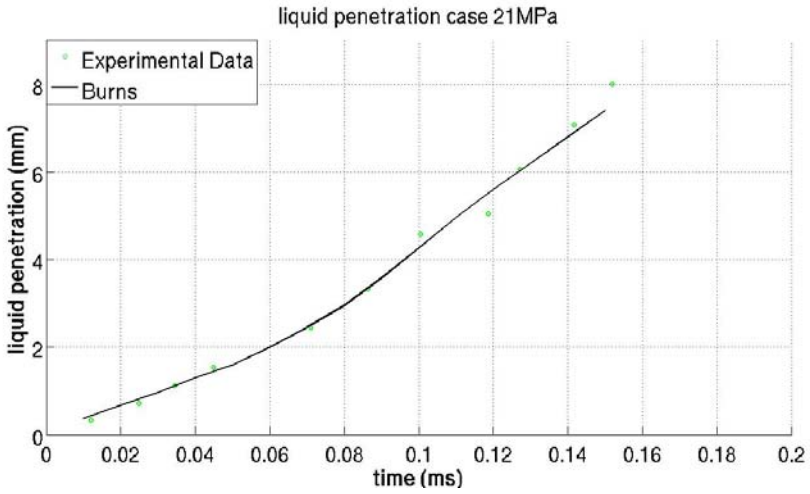


Figure 4.14: Liquid penetration obtained with Burns model and compared with experimental data.

4.2 Fluent results compared with DNS data

4.2.1 How to set the fluent case

It can be useful to define the steps to follow on the Fluent interface to set the simulations. Once fluent is launched in general options it is possible to impose the solver characteristics

$$\boxed{\text{General} \rightarrow \text{Solver}}$$

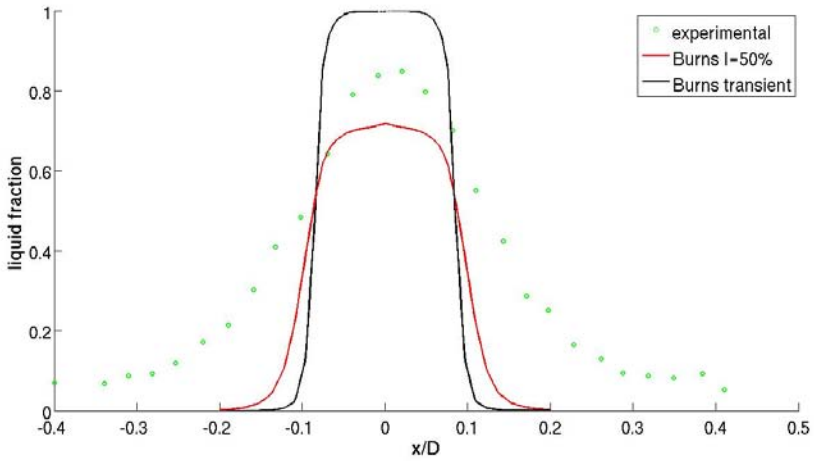
Inside this menu was chosen a pressure-based type solver with the absolute velocity formulation. The time scheme is transient and the case is 2D axis-symmetric. The second step is to chose the multiphase model.

$$\boxed{\text{Models} \rightarrow \text{Multiphase}}$$

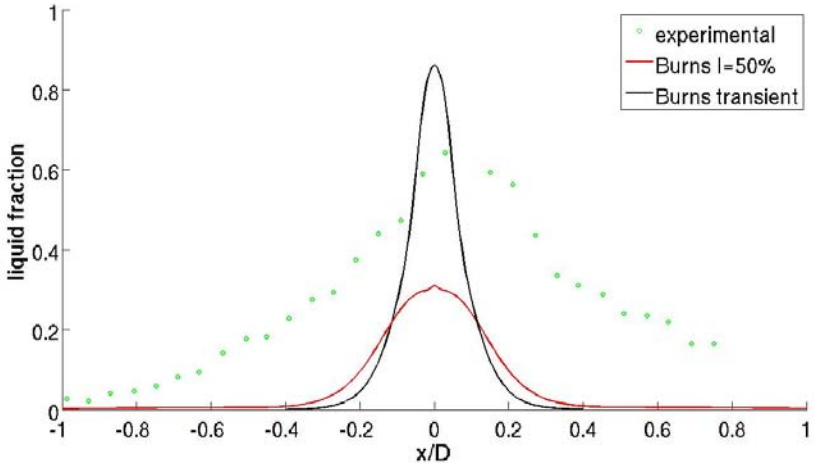
Eulerian model was chosen and the number of phases was imposed to 2. It is also possible to chose the numerical scheme. By default the numerical scheme is explicit; it was changed to implicit since with implicit scheme it is possible to reduce the solver time. All eulerian parameters[4] were switched off. The energy equation was switch off:

$$\boxed{\text{Models} \rightarrow \text{Energy} \rightarrow \text{off}}$$

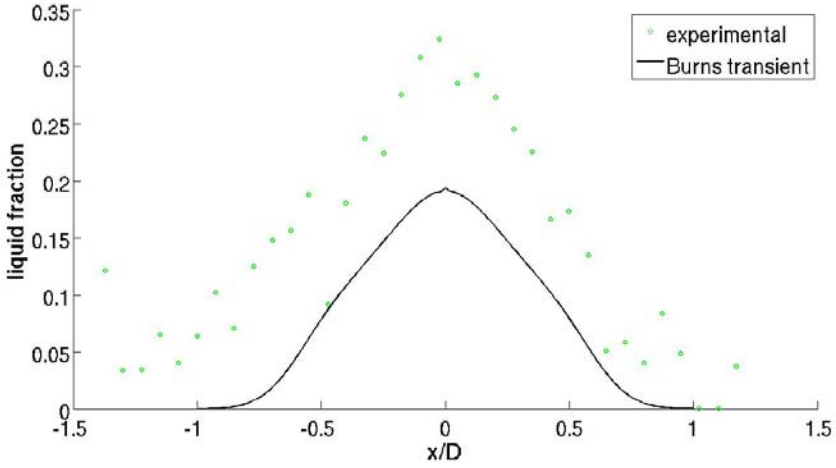
[4]Dense Discrete Phase model, Boiling model, Multi-Fluid VOF model



(a) $x = 0.283\ mm$



(b) $x = 2.483\ mm$



(c) $x = 7.083\ mm$

Figure 4.15: Liquid fraction obtained with OpenFoam solver and compared to experimental data at different axial position and 0.15 ms after the injection. Transient is referred to the transient value of turbulence intensity

In our case was chosen not to use the energy equation because it is not necessary in cases with constant ρ. Evaporation was not considered and so the two phases have the same temperature and constant density. The second reason why the energy equation was deactivated is that it complicate the set of equations and increase the required computational effort. Hence if it is possible to simplify the set of equation and obtain physical results it must be done. Without energy equation it is not possible to use ideal gas formulation and so not only the pressure but also the gas density must be imposed. The third model is viscous model.

$$\boxed{\text{Model} \rightarrow \text{Viscous}}$$

Standard $k - \epsilon$ formulation was chosen with standard near-wall treatment. One of the three, already described in section 2.3, turbulence multiphase model can be chosen. The model constants were maintained as default and no user defined function was used for primary or secondary phase. All the other models: radiation, heat exchanger, species, discrete phase and acoustics were switched off. It must be observed that if dense discrete phase model was chosen as eulerian parameter in multiphase model it would be necessary to activate discrete phase model. Once the models were chosen the next section is the material properties. Three materials must be defined: gas, liquid and solid. By default only two of these materials exists: gas air and solid aluminium. To create the third material:

$$\boxed{\text{Material} \rightarrow \text{Create/Edit} \rightarrow \text{Fluent Database} \rightarrow \text{diesel-liquid}}$$

Once all the materials are created it is possible to edit their properties:

$$\boxed{\text{Materials} \rightarrow \text{air (or diesel-liquid)} \rightarrow \text{edit}}$$

The material properties are reported in tables 4.12 and 4.4. The aluminium properties were maintained as default. The next step is to define the primary and secondary phase.

$$\boxed{\text{Phases} \rightarrow \text{primary-phase} \rightarrow \text{edit} \rightarrow \text{phase material} \rightarrow \text{air} \rightarrow \text{ok}}$$

The primary phase is always the continuous while the secondary phase is always the dispersed phase. In a similar way it is possible to chose the diesel liquid as secondary phase. When the secondary phase is chosen, in the same menu, it is possible to define the characteristic diameter of the dispersed phase. This diameter will be used to calculate the drag forces

acting on secondary phase[5]. The granular and inter-facial area concentration are switched off. The next step is to choose the boundary conditions

Boundary Conditions → "chose zone" → edit → "chose correct condition"

It is important to set the operating conditions inside the boundary menu. Selecting operating condition a new window appear where the operating pressure can be chosen. The pressure was calculated with real gas state equation. It is important to chose as reference pressure location a point where for sure $\alpha^g = 1$. Considering the actual geometry such coordinates are $[1e-05\ 0.0005\ 0]$ expressed in meters. The gravity force is switched off. For break up simulations the reference values are not so important as for aerofoil simulations. At this step the solution set-up is concluded. Selecting "solution methods" entry it is possible to chose the spatial discretization and transient formulation. The under relaxations factors stay inside "solution control" menu. All relaxation factor were preserved as default. In

Monitors → Residuals → edit

is possible to choose when stop the iteration within one time step. Two convergence criteria are possible: absolute and relative. It can be chosen a high number of iteration inside one time step and some absolute convergence criteria to stop the simulation. The value of convergence criteria depends on the physics that is simulated. A value of $1e-03$ is acceptable for continuity while for other equations it depends on the problem. The absolute criteria used for continuity and volume fraction is $1e-05$. It was observed that almost always the continuity residual had the highest value if compared with the other residual quantities. The solution initialization set-up is very important. In aerofoil simulations any case was initialized with hybrid initialization but this method is not suitable for breakup simulations. Indeed with hybrid initialization all the domain is filled with liquid at time $t = 0$. The case must be initialized with standard initialization.

Solution Initialization → compute from → "select the liquid inlet"

and

Solution Initialization → Initial values → phase 2 Volume Fraction → 0

The previous operation is very important indeed it imposes $\alpha^g = 1$ at time $t = 0$ in any cell of the domain. It is important to check all the values inside the

[5]fluent does not requires to define a characteristic diameter for primary phase.

initial values window and to ascertain that its are physically reasonable. After the check was done to initialize the case it is sufficient to select "Initialize" in solution initialization menu. Selecting "run calculation" menu it is possible to chose the temporal and iterative parameters of the simulation. Since no limit is imposed by Courant number, the time step size was imposed equal to $2.5e-06$. However it is recommended to calculate the time step size with a Courant number non above 200. If the simulation does not converge a possible solution can be to lower the time step size. The maximum iterations per time step can be chosen relatively high and depend by the problem. Usually the solution converge slowly in the first instants of the simulation.

The Fluent software presents three $k-\epsilon$ turbulent models for simulations: dispersed, mixture and per phase. All these models were tested. The tables 4.14, 4.15 and 4.16 present the configurations of these models of the most representative results. The error reported is calculated with the equation 4.1.

4.2.2 Dispersed turbulence treatment

The table 4.14 resumes the used parameters in simulations with *dispersed turbulence treatment*. Three main choices affects the simulation's results: dispersion model, model constant and the secondary phase characteristic diameter. In the table 4.14 the cases from 1 to 4 have the same dispersion model, same particle diameter but the model constant changes. The best results are obtained with the first case. If the particle diameter is increased to $D = 2{\cdot}10^{-6}$ as in case 6, than the maximum possible constant is $C = 0.2$, with constant higher values the case diverges. The same behaviour was noticed with $D = 3 \cdot 10^{-6}$. If the particle diameter is decreased to $D = 5 \cdot 10^{-7}$ and the constant is maintained $C = 0.2$ the results remains practically unaltered. The difference between the error of case 1 and case 7 is too small to conclude which case is better. Indeed the error calculated with equation 4.1 is based on the polynomial interpolation of the RANS and DNS data therefore the error itself depends on the grade of the polynomial interpolation. Moreover, the data used to create the interpolation was extracted from graphics using the open source software *Engauge-digitizer*, for this reason there is an intrinsic user error. It was disclosed from simulations that the maximum diameter to avoid the divergence is $D = 4 \cdot 10^{-6}$ whereas the minimum acceptable diameter is $D = 2 \cdot 10^{-7}$. Physical results were obtained only with diameters smaller or equal to 10^{-6}. In general, decreasing the constant value of Burns model, the breakup of liquid jet approaches to the inlet while no strong influence, in terms of breakup length, from diameter value was found.

Only two results of Lopez de Bertodano model are presented in the table 4.14. The model constant must be increased to improve the simulation's

case	dispersion model	model constant	D_{fuel} $[\mu m]$	error
1	Burns	$C = 0.2$	$1 \cdot 10^{-6}$	2.37
2	Burns	$C = 0.3$	$1 \cdot 10^{-6}$	4.93
3	Burns	$C = 0.4$	$1 \cdot 10^{-6}$	8.40
4	Burns	$C = 0.5$	$1 \cdot 10^{-6}$	11.06
5	Burns	$C = 0.2$	$2 \cdot 10^{-6}$	5.64
6	Burns	$C = 0.2$	$3 \cdot 10^{-6}$	10.77
7	Burns	$C = 0.2$	$5 \cdot 10^{-7}$	2.25
8	Lopez de Bertodano	$C = 1.5$	$1 \cdot 10^{-6}$	6.17
9	Lopez de Bertodano	$C = 1$	$1 \cdot 10^{-6}$	10.34
10	Simonin	$C = 0.55$	$1 \cdot 10^{-6}$	5.94
11	Simonin	$C = 0.65$	$1 \cdot 10^{-6}$	4.91
12	Simonin	$C = 0.75$	$1 \cdot 10^{-6}$	4.93
13	Simonin	$C = 0.85$	$1 \cdot 10^{-6}$	5.61
14	Simonin	$C = 1$	$1 \cdot 10^{-6}$	7.02
15	Simonin	$C = 2$	$1 \cdot 10^{-6}$	14.24
16	Simonin	$C = 0.55$	$2 \cdot 10^{-6}$	13.14
17	Simonin	$C = 0.55$	$3 \cdot 10^{-6}$	17.34
18	Simonin	$C = 0.65$	$5 \cdot 10^{-7}$	7.66

Table 4.14: Fluent turbulence treatment: dispersed model

result but increasing the constant the simulations diverges in the first time steps. If the droplet diameter is decreased and the constant is increased the simulation does not diverge in the very beginning but in the following time steps without reaching the last simulation's time step. The general behaviour of this model is: decreasing the diameter the breakup length increases; increasing the model constant the break up length decreases.

Simonin model gives better results but not so good as Burns model. Case 11, in the table 4.14, presents the best results. Simonin model is not influenced by the energy equation. Maintaining the same value of particle diameter it is possible to decrease the jet breakup length by increasing model constant. Preserving the value of the model constant and increasing the droplet diameter the breakup length increases.

The VOF model is not reported in the table respective to dispersed turbulent treatment because for some combinations of diameter and constant values the model diverged, for other combinations the residuals oscillate round non acceptable values.

4.2.3 Mixture turbulence treatment

The next $k-\epsilon$ multiphase turbulence treatment is *mixture*, the data relating RANS simulations are provided in table 4.15.

Considering Burns model the best result is obtained in case number 2. It is interesting to notice that if the diameter is increased by 2.5 times (as in case 6) the error scarcely changes. However if the diameter is decreased by a ratio of two (as in case 7) the error increases it's value by a fourth. This means that unlike in *dispersed turbulence treatment*, when the model converges, it is more sensitive to the diameter variation. Comparing the case 7 and 1 it is possible to notice that the errors are practical identical between the two cases (the same can be observed between case 6 and 3); it is possible to state that increasing or decreasing the diameter by a factor of two is quasi the same as decreasing or increasing the constant by a 0.1 quantity. Accordingly the variation of particle diameter produces less difference in results as the variation of model constant. The behaviour of the Burns model is to increase the breakup length scale if the model constant is decreased; to increase the breakup length scale if the particle diameter is decreased.

The next model is *VOF* turbulence dispersion. The best result was obtained in case number 10. It is the best result not only obtained between the cases studied with VOF model but is the best result obtained considering mixture treatment of turbulence. Comparing the case 10 to the case 12 in which the diameter was doubled it is possible to note a small increasing of the error due to an increase in liquid breakup length. The VOF turbulence dispersion model gives good results only considering the mixture turbulence treatment.

Considering Lopez de Bertodano model the best result is obtained with case 14. It is interesting to notice that the constant is much bigger than the constant of the best result with dispersed turbulence treatment (see table 4.14 case 8). Increasing the constant model it is possible to decrease the length scale of the liquid breakup. Decreasing the diameter the breakup length decreases while decreasing the model constant the breakup lenth increases. Also it was marked that considering a diameter $D = 1e-06$, the case converges with a constants 16, 22 but diverges with constant 20.

Finally Simonin model gives the best results in the case 19. Increasing the model constant the liquid breakup length decreases. Comparing cases 21 and 22 it was noticed that a double diameter produces small increment in breakup length.

case	dispersion model	model constant	D_{fuel} $[\mu m]$	error
1	Burns	$C = 0.8$	$1 \cdot 10^{-6}$	4.92
2	Burns	$C = 0.7$	$1 \cdot 10^{-6}$	3.79
3	Burns	$C = 0.6$	$1 \cdot 10^{-6}$	3.95
4	Burns	$C = 0.5$	$1 \cdot 10^{-6}$	5.71
5	Burns	$C = 0.2$	$1 \cdot 10^{-6}$	16.23
6	Burns	$C = 0.7$	$2.5 \cdot 10^{-6}$	3.97
7	Burns	$C = 0.7$	$5 \cdot 10^{-7}$	4.98
8	Diffusion in Vof	$C = 2$	$1 \cdot 10^{-6}$	11.93
9	Diffusion in Vof	$C = 2.5$	$1 \cdot 10^{-6}$	5.41
10	Diffusion in Vof	$C = 3$	$1 \cdot 10^{-6}$	2.98
11	Diffusion in Vof	$C = 3.5$	$1 \cdot 10^{-6}$	6.22
12	Diffusion in Vof	$C = 3$	$2 \cdot 10^{-6}$	4.03
13	Lopez de Bertodano	$C = 8$	$1 \cdot 10^{-6}$	9.27
14	Lopez de Bertodano	$C = 11$	$1 \cdot 10^{-6}$	7.12
15	Lopez de Bertodano	$C = 16$	$1 \cdot 10^{-6}$	7.81
16	Lopez de Bertodano	$C = 22$	$1 \cdot 10^{-6}$	12.20
17	Lopez de Bertodano	$C = 16$	$2 \cdot 10^{-6}$	21.60
18	Simonin	$C = 0.45$	$1 \cdot 10^{-6}$	7.08
19	Simonin	$C = 0.65$	$1 \cdot 10^{-6}$	3.66
20	Simonin	$C = 0.85$	$1 \cdot 10^{-6}$	5.63
21	Simonin	$C = 0.65$	$2 \cdot 10^{-6}$	4.29

Table 4.15: Fluent turbulence treatment: mixture model

case	dispersion model	model constant	D_{fuel} $[\mu m]$	error
1	Burns	$C = 1$	$1 \cdot 10^{-6}$	7.04
2	Burns	$C = 0.8$	$1 \cdot 10^{-6}$	4.01
3	Burns	$C = 0.6$	$1 \cdot 10^{-6}$	4.60
4	Burns	$C = 0.4$	$1 \cdot 10^{-6}$	9.99
5	Diffusion in Vof	$C = 2$	$2 \cdot 10^{-6}$	17.95
6	Diffusion in Vof	$C = 4$	$2 \cdot 10^{-6}$	19.24
7	Simonin	$C = 0.65$	$1 \cdot 10^{-6}$	4.06

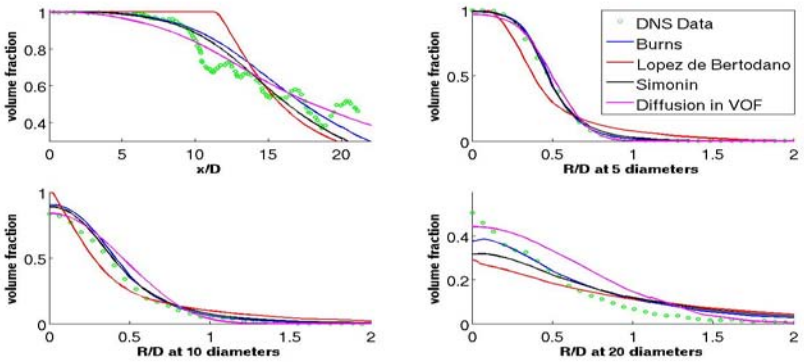
Table 4.16: Fluent turbulence treatment: per phase model

4.2.4 Per Phase turbulence treatment

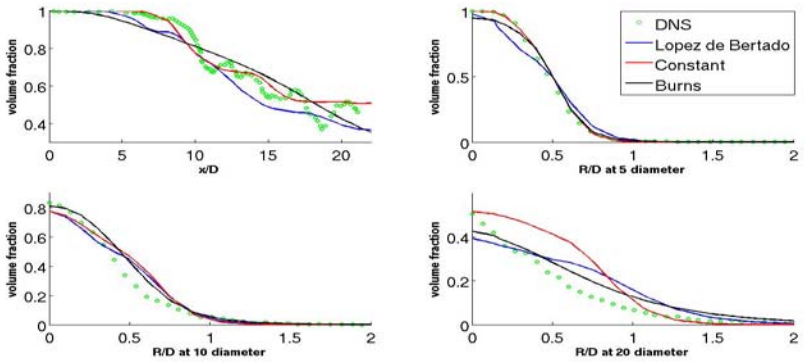
The last case is computationally the most complex because as has already been mentioned involves two set of $k - \epsilon$ equations. Only two models give physical results: Burns and Simonin. Lopez de Bertodano model does not converge. VOF model converges but gives physical wrong results as is expressed by the cases 5 and 6 in table 4.16

The first four cases in the table 4.16 are related to Burns model. The case 2 presents the smallest error. One important thing has to be pointed out: usually the models needed more than 100 iteration in the initial time steps to converge but converged very quickly in the last part of the simulation. Burns model with per phase turbulence description converges very quickly from the beginning to the end of the simulation. The similar behaviour to the previously Burns cases was noticed: small dependence from droplet diameter in terms of liquid breakup length. Increasing the droplet diameter the length scale increases. Increasing the model constant the liquid length scale decreases.

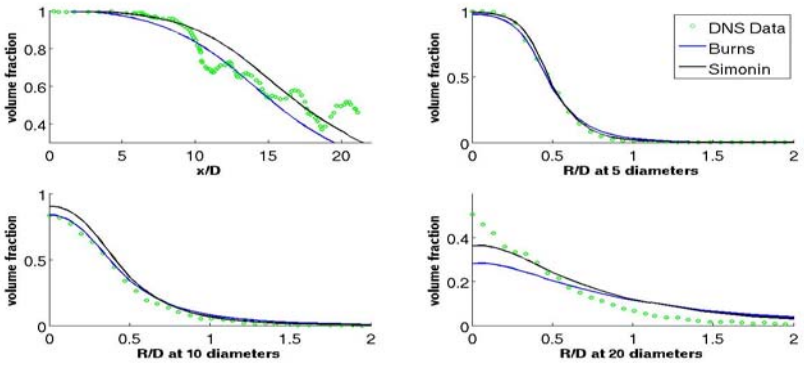
The Simonin model presents a behaviour very similar to the mixture cases. The behaviour of the case 7 in table 4.16 is very similar to the behaviour of case 19 in table 4.15. The consequence of the variation of model constant and particle's diameter qualitatively is identical to the cases already studied in mixture and dispersed turbulence multiphase treatment. The figure 4.16 shows axial behaviour of the volume fraction and radial behaviour at following axial positions $x = 5 \cdot D = 0.0005\mu m$, $x = 10 \cdot D = 0.001\mu m$, $x = 20 \cdot D = 0.002\mu m$. Considering the position $x = 20 \cdot D = 0.002\mu m$ it is possible to have an idea about the cone aperture of the liquid spray. All the models, except diffusion in VOF model, slightly over-predict the cone angle. In dispersed turbulence treatment, Lopez de Bertodano model predicts a non uniform axial behaviour of the volume fraction. If figures 4.7 and **??** are compared it is possible to conclude that Fluent software does not creates



(a) Mixture turbulence treatment



(b) Discrete turbulence treatment



(c) Per phase turbulence treatment

Figure 4.16: Breakup RANS data from fluent simulations at time $t = 0.1\ ms$ compared with DNS data for different turbulence treatment

fluctuating zones as OpenFoam, and so qualitatively fluent shows a better physical behaviour. The fluctuating behaviour was not verified in any of the Fluent simulation.

4.3 RANS Fluent results compared with experimental data

The models with the best results were considered to simulate the experimental case. The simulation's results are compared with experimental data to understand how boundary conditions and material properties condition the models. When convergence is not reached some changes to the model must be performed. It is possible to change the numerical schemes, slightly alter the turbulence parameters or the model constants[6]. The boundary conditions were changed accordingly to the experimental set-up while the turbulence dispersion model constants where chosen equal to the models discussed in previous section. None of models converged with dispersed turbulence treatment. When the divergence of some quantity[7] was verified, changing the numerical schemes it was always possible to avoid the divergence but not always was possible to make converge the simulation. In general exist one or more time steps within simulation when the residual remain high. The main problems were related to the continuity residuals that had the trend to oscillate around a value of order $1e-2$ that is too high to be accepted.

4.3.1 Burn turbulence dispersion model

Using mixture turbulence treatment model only two dispersion models converged: Burns model and Simonin model.

Mixture turbulent treatment

The model constant ($C = 0.7$) and liquid diameter $D = 1e-06$ were fixed to the same values of case 2 in the table 4.15. The boundary conditions were changed accordingly to the experimental set-up. The values of k and ϵ were calculated with a matlab function as explained in section 4.1.3. In what

[6]It is referred to model such turbulence dispersion and does not refer to turbulence model. It is not recommended to change turbulence constants; a modification of one constant in the turbulence model involves a variation of all other constants since its are interdependent. The turbulence constants are based on multiple experimental data and are chosen to make the model stiff.

[7]generally k was the must problematic quantity

follows different numerical schemes were adopted but it was inferred that changing numerical scheme can be a good attempt to avoid the divergence but not the oscillations of residual terms. In the first simulation it was not possible to reach the convergence, at time $t = 2e-05$ the continuity began to oscillate around a value of order $1e-02$. The turbulence length scale was slightly changed but without acceptable results. Modifications of k and ϵ values only shifted the instant when continuity begin to oscillates around unacceptable values. The last attempt was to use the droplet diameter equal to $2e-06$, constant turbulence intensity and constant turbulence length scale; the values were calculated as average of the values calculated as explained in section 4.1.3. With this choices of turbulence intensity ($I = 20\%$)and turbulence length scale ($l = 65\mu m$) the continuity was stabilized. Since the value of liquid penetration was less than $7mm$ the turbulence intensity and turbulence length scale were reduced respectively to 5% and $16.9\mu m$ and set time independent. The figure 4.17 shows the improvement in liquid penetration.

Per phase turbulent treatment

Comparing the per phase residuals with mixture residuals the conclusion is that per phase model (with same boundary conditions and numerical schemes) is more stable in convergence. However also in per phase model it was necessary to increase the droplet particle diameter to $2e-06$ for continuity stability. The boundary turbulence values are calculated as explained in section 4.1.3. The simulations converged but the liquid penetration is underestimated (less than $7mm$) **??**. Also in this case the boundary turbulence parameters were decreased to $I = 5\%$ and $l = 16.9\mu m$ to obtain a better result for liquid penetration length as showed in figure 4.17

4.3.2 Simonin turbulence dispersion model

Simonin model converges only with a Mixture and Per phase multiphase turbulence treatment.

Mixture turbulence treatment

The model turbulence constant ($C = 0.65$) and particle diameter ($D = 1e-06$) is reported in case 19 of table 4.15. The boundary conditions and geometry changed according to the experimental case. The first attempt was to introduce as turbulence boundary conditions the values calculated as explained in section 4.1.3 but the same behaviour, as for Burn model, was

observed: continuity does not converge. The continuity began to oscillate at $t = 2.25e-05$ and varying the numerical schemes was not possible to eliminate the continuity abnormal trend. The case does not converge neither using the averaged turbulence value for turbulent intensity ($I = 20\%$) and length scale ($l = 65\mu m$). Increasing the droplet diameter, the convergence behaviour did not change. It was verified that the model converge with lower values of turbulence ($I = 5\%$ and $l = 16.9\mu m$) and $D = 2e-06\mu m$. The figure 4.17 (b) shows the qualitative behaviour of liquid volume fraction and liquid penetration length.

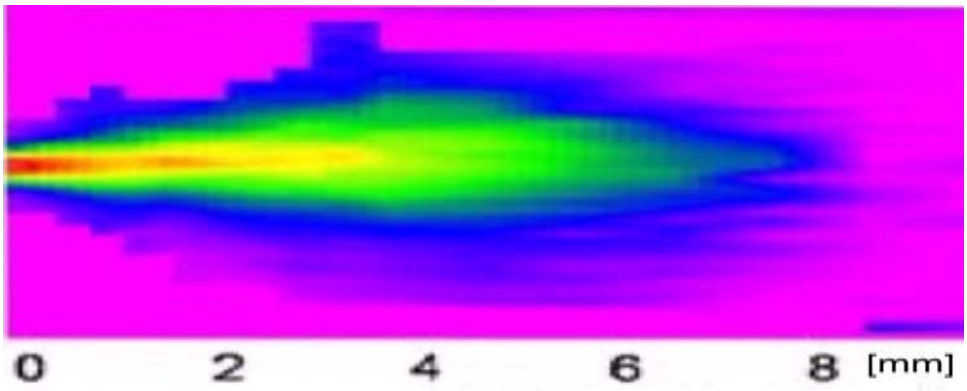
Per phase turbulence treatment

Considering per phase turbulence treatment with the Simonin constant model ($C = 0.65$) and particle diameter $D = 1e-06$ does not converged when was applied the turbulence boundary condition calculated as explained in section 4.1.3. The case converged with diameter $D = 2e-06$. However the liquid penetration length is not satisfactory because less than $7mm$.

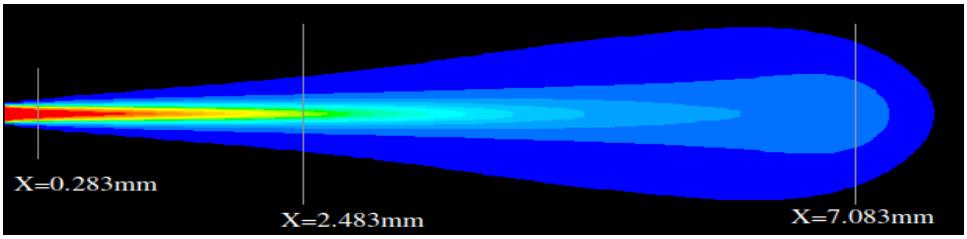
4.4 Conclusion

Fluent and OpenFoam gave acceptable results considering the liquid penetration and cone aperture. However there are some important differences between the settings of the two softwares. The first big difference is that in OpenFoam does not exist an implicit scheme to solve multiphase cases. For this reason the OpenFoam simulations are limited by the Courant number and so demand more time to finish. Fluent is supplied by an implicit solver hence the Courant number does not limit the simulation. The second big difference is that Fluent has more turbulence treatment for multiphase cases while OpenFoam has just the per phase turbulence model. However from simulations was concluded that per phase model has less problem with convergence and give acceptable results. Fluent does not have the blending model to calculate the drag force, for this reason the drag contribution in the momentum equation is calculated considering the dispersed phase characteristic diameter. OpenFoam is provided with blended model ad so it automatically weights the drag forces based on phase fraction.

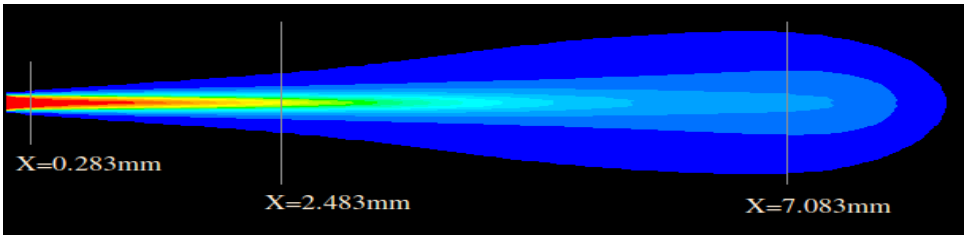
Several models to simulate the turbulence dispersion are presented within OpenFoam and Fluent. However only Burns model proved to be stable in Fluent and OpenFoam. Constant model, Lopez de Bertodano model and Diffusion in VOF model failed to simulate the breakup. Simonin model, presented only in Fluent, also gave acceptable results. It is interesting to



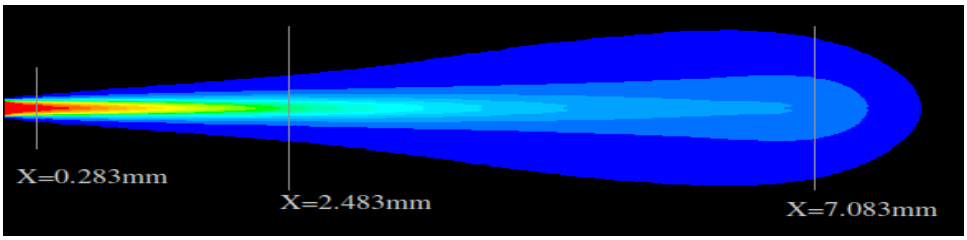
(a) experimental behaviour of liquid fraction



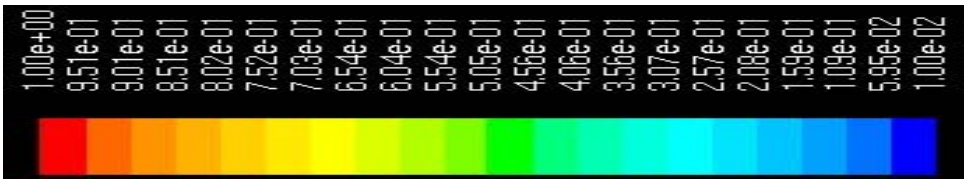
(b) turbulence mixture $I = 5\%$, turbulence dispersion: Simonin model



(c) turbulence per phase $I = 5\%$, turbulence dispersion: Burns model

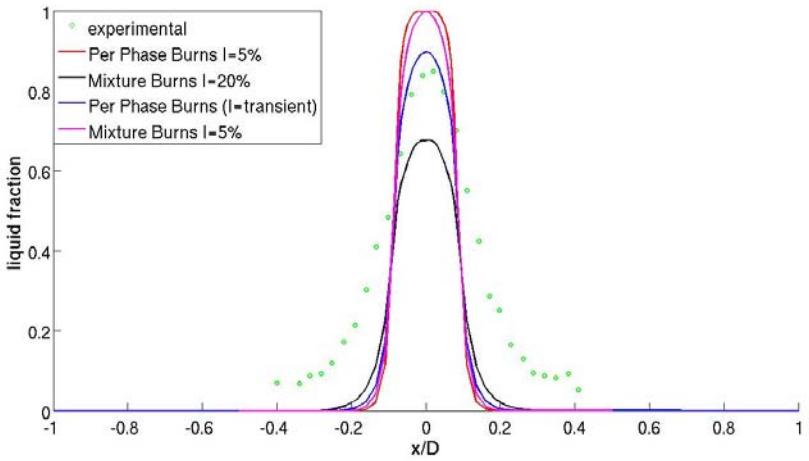


(d) turbulence mixture $I = 5\%$, turbulence dispersion: Burns model

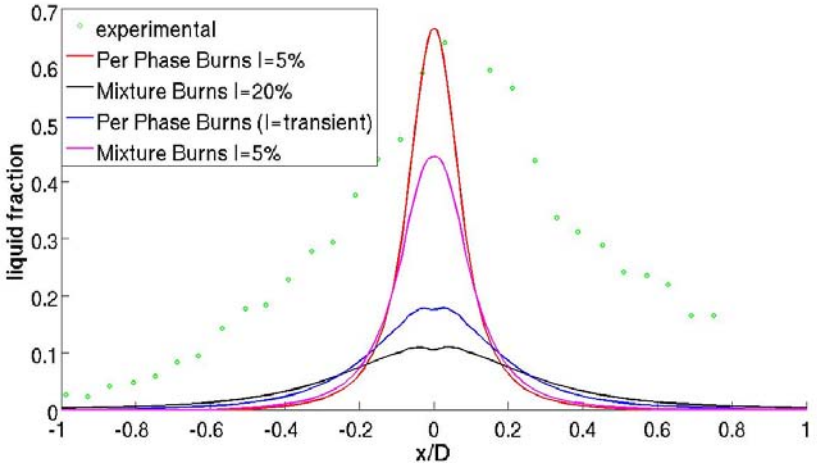


(e) the map refers fluent RANS simulations

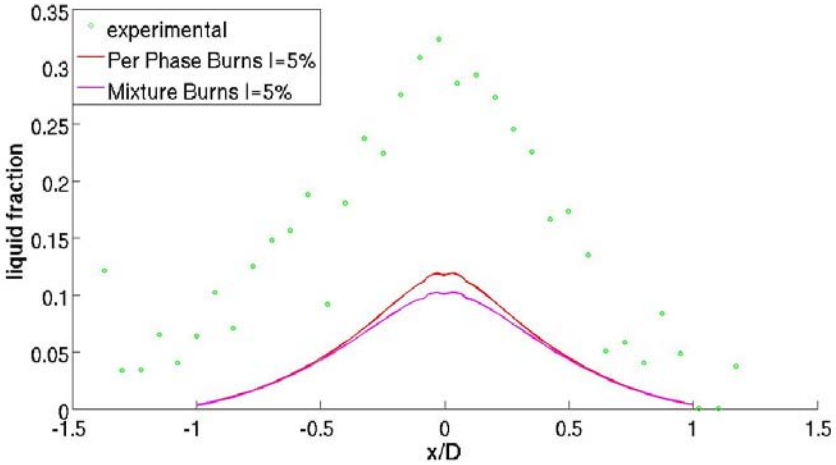
Figure 4.17: Qualitative behaviour of the liquid fraction compared with experimental data



(a) axial position $x = 0.283\mu m$

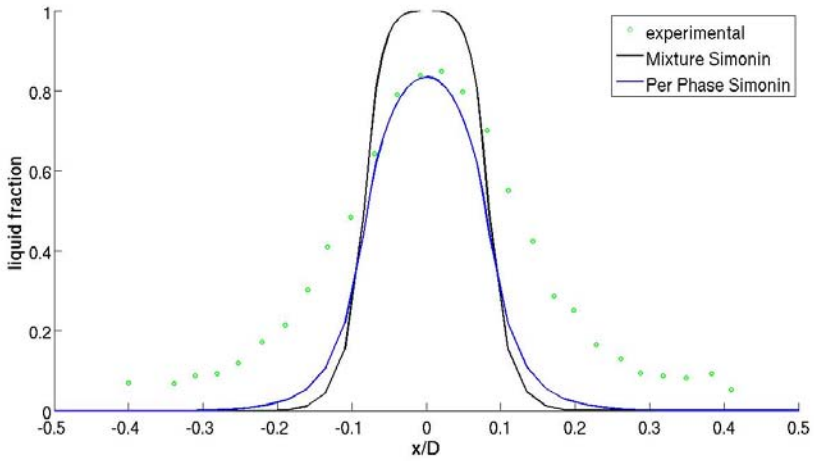


(b) axial position $x = 2.483\mu m$

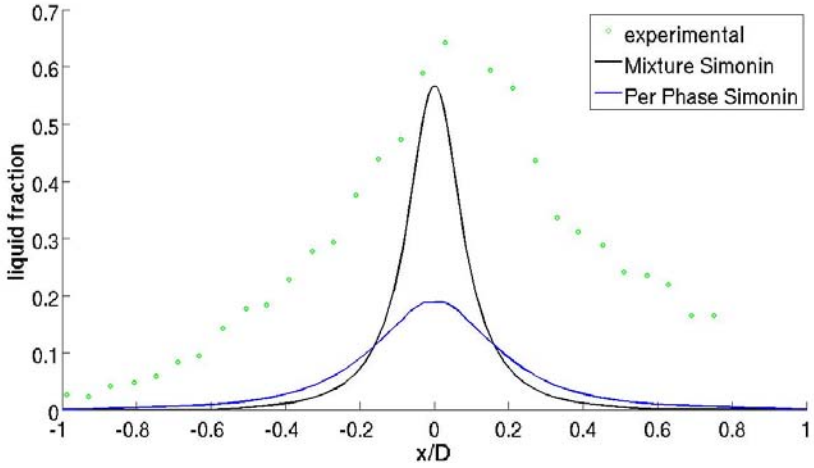


(c) axial position $x = 7.083\mu m$

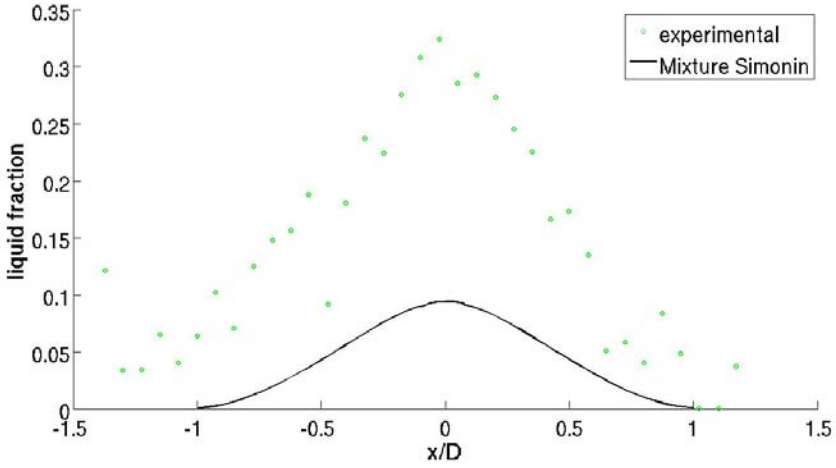
Figure 4.18: Radial distribution of volume fraction obtained with Burns model and compared with experimental data



(a) axial position $x = 0.283\mu m$



(b) axial position $x = 2.483\mu m$



(c) axial position $x = 7.083\mu m$

Figure 4.19: Radial distribution of volume fraction obtained with Simonin model and compared with experimental data

notice that the Simonin and Burns constants used in Fluent simulations are very similar. The liquid characteristic diameter that gave best results is $D = 2e - 06$; it was noticed that when the case does not converge varying a little the particle diameter can bring to convergence.

Transient boundary conditions was applied at the inlet. To approach the experimental results using OpenFoam the turbulent intensity defined at the inlet was increased to values of order of 20% for simulate the experimental case, and maintained $I = 5\%$ to simulate the DNS case. Using Fluent a constant turbulence intensity ($I = 5\%$) and constant turbulent length scale[8] (l=10%D) gave good results compared with DNS and experimental data. This different behaviour can be due to the different temporal scheme.

[8]D is the diameter

Conclusion

OpenFoam and Fluent was used to simulate and validate two aerospace problems: liquid jet breakup near the nozzle and drag force coefficient for a classical aerofoil's profile.

OpenFoam is an open source CFD. It does not have an interface and all the commands must be provided using *.txt* files. The absence of the interface can be a negative but also a positive note. The lack of interface impose the user to understand better how the software works; what numerical schemes it uses, what models are switched on. In this condition the CFD software does not obscure anything from the user who have the maximum control on what is happening. The lack of the interface is unfavourable when the case must be created from zero, when every file must be generated and tuned for the physical case that must be simulated. This operation can request a lot of time. Another positive aspect of OpenFoam is that it has practically all the numerical schemes used in commercial solvers and a lot of models for different physical problems. If the user has some knowledge about programming he can modify the solver, add new models or change the existing one. OpenFoam has a very user friendly post processor called paraFoam. Finally if the geometry is simple it is possible to define the mesh using blockMesh without the requirement of using an external preprocessor. The most important OpenFoam's shortcoming is the lack of an official documentation. The absence of the user and theory guide imposes the to examine the code and to understand it's implementation.

Fluent has an interface that accompanies the user in setting the case. The presence of the interface can be useful because save a lot of time but sometimes it can hide some options or some used models. Sometimes Fluent can converge to some plausible result even if some parameter was defined erroneously. Fluent seems to be much more fast and solid in convergence, but this depend on the user skill to set the case, to chose the boundary and initial conditions. Finally there is a lot of information about setting the classical problems in Fluent. There is an official user guide and a theory guide that drives the user inside the software's menu and settings.

Analysing the drag aerofoil results, with the configuration used and described in the third chapter it was not possible to solve with OpenFoam the viscous boundary layer using the created mesh. OpenFoam is very sensitive to transition ratio between cells of the grid and so it is difficult to make a mesh with y^+ less than one to solve the laminar zone in boundary layer. The use of wall function gave results on evaluating C_d with an error in a range of [10% , 20%]. Fluent managed to solve the viscous sub-layer providing results with an error less than 10%.

Considering the second part of the simulations the purpose was to simulate the breakup in the near nozzle zone. It was established that using RANS model and Volume of Fluid technique it is not possible to simulate the jet breakup and to study the radial behaviour of the liquid volume fraction. An Eulerian-Eulerian description was chosen. Euler-Euler equations are equal to VOF equations but averaged a second time. The interphase between the two fluids is not solved exactly (as in VOF) but is averaged, so extra terms appear. The extra term in Euler-Euler description must be modeled. Turbulent dispersion is the must important model that conditions the breakup. Different turbulent dispersion models were studied. Comparing OpenFoam and Fluent RANS results with experimental data it was concluded that the Burns turbulent dispersion model produces best results. It was also observed that the turbulent dispersion model's constant, the characteristic diameter of the disperse phase and the turbulence boundary conditions at the inlet affect mostly the results. It was established that the model's constant does not depend from the inlet jet velocity conditions and from the materials properties. However in OpenFoam (not in Fluent) it was necessary to change the turbulent conditions at the inlet to achieve good results in liquid jet cone aperture. It was also observed that the best results are obtained with "per phase" multiphase turbulence model. Both software gave realistic results compared with experimental and DNS data. Further investigation is require to understand how the models behave in evaporation and combustion.

Bibliography

[1] Alan D. Burns, Thomas Frank, Jan Hamill, Jun Mei-Shi *The favre averaged drag model for turbulent dispersion in eulerian multiphase flows*, 2004;

[2] Alex B. Liu, Daniel Mather, Rolf D. Reitz *Modeling of effects of drop drag and breakup on fuel sprays*, 1993;

[3] A. J. Ramirez et. al. *Quantitative X-ray measurements of high pressure fuel sprays from a production heavy duty diesel injector*, 2009;

[4] Charles L. Ladson *Effects of independent variation of Mach and Reynolds numbers on the low speed aerodynamic characteristics of the NACA 0012 airfoil section*, 1988;

[5] Fracois G. Schmitt *About Boussinesq's turbulent viscosity hypothesis: historical remarks and direct evolution of its validity*, 2007;

[6] Florian R. Menter *Zonal two equation $k-\omega$ turbulence models for aerodynamic flows*, 1993;

[7] Henrik Rusche *Computational Fluid Dynamics of Dispersed two-phase flow at high phase fraction*, 2002;

[8] Juan Alonso, David Darmofal, William Gropp, Elizabeth Lurie, Dimitri Mavriplis *CFD Vision 2030 study: A path to revolutionary computational aerosciences*, 2014

[9] Karsten Reuter et. al. *Towards the numerical simulations of multi-scale two-phase flows*, 2011;

[10] Katrin Kissling, Julia Springer et. al. *A coupled pressure based solution algorithm based on he volume of fluid approach for two or more immiscible fluids*, 2010;

[11] K. A. Sallam, Z.Dai, G.M.Faeth *Liquid breakup at the surface of turbulent round liquid jets in still gases*, 2001;

[12] Ph. Marmattant et E. Villermaux *Attomisation Primaire dans les jets coaxioux*, 2003;

[13] P. J. Oliveira *Computer Modelling of multidimensional multiphase flow and application to T-junctions*, 1992;

[14] Robert H. Kraichnan *On Kolmogorov inertial-range theories*, 1974;

[15] R. Lebas, T. Menard, P. A. Beau, A. Benlemont, F. X. Demoulin *Numerical simulation of primary breakup and atomization: DNS and modelling study*, 2009;

[16] R. J. Issa, P. J. Oliveira *Modelling turbulent dispersion in two phase flow jets*, 1993;

[17] S. P. Lin, R.D.Reitz *Drop and spray formation from a liquid jet*, 1998;

[18] S. San, P. K. Senecal, E. Pomraning *Comparison of Rans and Les Turbulence Moels agains Volume Diesel experiments*, 2012;

[19] Stephan B. Pope *Turbulent Flows*, 2000;

[20] W. C. Reynolds *Computation of turbulent flows*, 1976;