ABSTRACT

Title of Thesis:	DEVELOPMENT OF A SUITE OF VERIFICATION TESTS FOR THE CFD FIRE MODEL FIREFOAM
	Hao Li, Master of Science, 2017
Thesis Directed By:	Professor Arnaud Trouvé
	Department of Mechanical Engineering

Several benchmark cases are propelled to perform the verification and validation (V&V) of FireFOAM, a LES solver based on OpenFOAM. The Lamb-Oseen vortex with co-flow is used to verify several temporal and spatial schemes used in FireFOAM. The Taylor-Green vortex is implemented to verify the conservation of kinetic energy and enstrophy growth of FireFOAM. The Smagorinsky and one-equation turbulence models are validated by simulating the case of an isotropic decaying turbulence. Numerical solution of kinetic energy decay and energy spectrum are compared with the experimental data of Comte-Bellot and Corrsin (CBC). Several combinations of boundary conditions (BCs) are verified by studying the case of Lamb-Oseen vortex with co-flow and the case of a hot bubble with buoyancy. Several problems with BCs in FireFOAM are identified and corresponding reasons are analyzed. The Eddy-dissipation combustion model is evaluated in the McCaffrey's pool fire case. The classical -5/3 slope for the temperature rise and -1/3 slope for the velocity decay are observed in the plume zone.

DEVELOPMENT OF A SUITE OF VERIFICATION TESTS FOR THE CFD FIRE
MODEL FIREFOAM

By

Hao Li

Thesis submitted to the Faculty of the Graduate School of the

University of Maryland, College Park, in partial fulfillment

of the requirements for the degree of

Master of Science

2017

Advisory Committee:

Professor Arnaud Trouvé

Professor Amir Riaz

Professor Johan Larsson

Acknowledgements

I would like to thank Prof. Arnaud Trouvé who offer me the chance to do the research on computational fluid dynamics and his guidance, support and patience throughout my research.

I would like to thank Prof. Larsson for his interesting and useful guidance in the course of computational method for scientist and engineer, which make me believe my interest in computational science.

Thank also to all my current CFD group members: Salman Verma, Alexis Pascal Marchand, Cong Zhang, Le Van Minh. Thank you for your help when I meet with difficulties.

Thank all of the guys who I meet at the University of Maryland. This two-year experience make me learn a lot and I will treasure it forever.

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction:

Introduction to FireFOAM and OpenFOAM:

FireFOAM is a Large Eddy Simulation (LES) solver based on OpenFOAM, which is a general-purpose Open Source CFD Toolbox. FireFOAM is developed by FM Global for fire and explosion modeling applications [1].

OpenFOAM is a free, open source CFD software released and developed primarily by OpenCFD Ltd since 2004 [2]. It has a large user base across most areas of engineering and science, from both commercial and academic organizations. OpenFOAM has an extensive range of features to solve anything from complex fluid flows involving chemical reactions, turbulence and heat transfer, to acoustics, solid mechanics and electromagnetics. OpenFOAM is first and foremost a C++ library, used primarily to create executables known as applications. The applications fall into two categories: solvers, that are each designed to solve a specific problem in continuum mechanics; and utilities, that are designed to perform tasks that involve data manipulation [3]. The OpenFOAM distribution contains numerous solvers and utilities covering a wide range of problems.

The main benefits for people to choose OpenFOAM is listed below:

- Technologically equivalent to commercial solutions.

- Totally free – no licensing fees/unlimited number of users, jobs and cores.

- Friendly syntax for partial differential equations. Its syntax for tensor operations and partial differential equations closely resembles the equation being solved.

- Wide range of applications and models ready to use.

1

- Extensibility – Users can create custom objects, such as boundary conditions or turbulence models, which will work with existing solvers without having to modify or recompile the existing source code.

Structure of files in OpenFOAM:

There are three basic directories which are necessary to construct a case in OpenFOAM, the time directory, the constant directory and the system directory. Figure 1.1 shows the relationship between these directories and some common files contained in these directories.
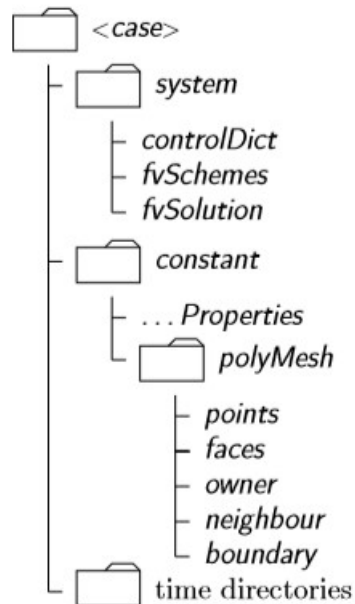
Figure 1.1: Case directory structure [4]

The time directory can be separated into two parts, the 0 directory and the directories for other times. 0 directory is used to set the initial and boundary conditions of each case. It contains individual files of data for particular fields, such as the pressure field and velocity field. Each OpenFOAM field must always be initialized even when the solution does not strictly require it. The other time directories are used to store the simulation result at each output time. Of course,

each of these can also be treated as the initial condition for the simulation running after this time. The name of each time directory is the simulation time at which the data is stored.

The constant directory is used to store the physical properties of the field that is simulated and the complete definition of the computational mesh generated for the case, which is stored separately in the polyMesh subdirectory. Some common files in the constant directory of FireFOAM are listed in Table (1.1) with their utilities.

Table (1.1) Basic properties defined in the constant directory

File Name	Utility
thermophysicalProperties	Define the energy/heat properties of the simulation
combustionProperties	Control the type of combustion modeling
thermo.compressibleGas	Specify the species properties of the simulation
turbulenceProperties	Control the type of turbulence modeling.
transportProperties	Specify the model type of kinetic viscosity ν and related necessary constant.
radiationProperties	Specify the radiation model of simulation
reactions	Define the global combustion equation

The system directory is used to set the methods and parameters related to the solution procedure. It primarily contains 3 dictionaries, i.e. controlDict, fvScheme, fvSolution. All runs set up in OpenFOAM solvers start by setting up a database. This database controls the input and output of data. What controlDict does is to set up the parameters which are essential for the creation of the database, i.e. it is used to set the format of the input and output of time and data. fvScheme is

used to set up the spatial discretization and interpolation method of each term in the governing

equation of the solver. OpenFOAM uses the standard Gaussian finite volume integration which

is based on summing values on cell faces. The value on the faces are interpolated from the center

value. So the difference choice of interpolation method will produce different results. The user

has numerous choices of interpolation methods to choose in OpenFOAM, some of which are

constructed for certain derivative terms. OpenFOAM divide these terms into different types

which is shown in Table (1.2). Interpolation of all these terms need to be specified in fvScheme.

The fvSolution is responsible for the choice of the iterative solvers which are used to advance the

solution in time and for the choice of tolerances. There are three types of tolerances that can be

set in fvSolution: the absolute tolerance (tolerance), the relative tolerance (relTol) and the

maximum number of iterations (maxIter).

Table (1.2) Types of terms in governing equation

Scheme types in OpenFOAM	comment
timeScheme	First and second time derivatives, $\partial/\partial t, \partial^2/\partial t^2$
gradSchemes	Gradient ∇
divSchemes	Divergence $\nabla \cdot$
laplacianSchemes	Laplacian ∇^2
interpolationSchemes	Cell to face interpolations of values
snGradSchemes	Component of gradient normal to a cell face
wallDist	Distance to wall calculation, where required

Procedure of simulation using FireFOAM:

Preprocessing:

Mesh generation:

The first step in a simulation is the decomposition of space, i.e. the creation of a computational domain. In this case, we use the command *blockMesh* to do this. The principle behind *blockMesh* is to decompose the domain geometry into a set of three dimensional, hexahedral blocks. Edges of the blocks can be straight lines, arcs or splines. The mesh is ostensibly specified as a number of cells in each direction of the block. In order to use this command, we need to create a dictionary file called *blockMeshDict* located in the *constant/polyMesh* directory of the case. *BlockMesh* reads this dictionary, generates the mesh and writes out the mesh data to *points* , *faces*, *cells* and *boundary* files in the same directory.

The running status of *blockMesh* is reported in the terminal window. Any mistake in the *blockMeshDict* file is picked up by *blockMesh* and the resulting error message directs the user to the line in the file where the problem occurred.

After the generation of the mesh, we can use the checkMesh command to check the mesh quality. This command will get the mesh statistics, the topology and geometry information of the mesh as well as test their validity. The graphics package Paraview is also a useful tool to check the mesh since it can plot the grid lines and thereby help inspect the structure of the mesh.

Initialization of the simulation:

Once the mesh generation is completed, the user can pay attention to the initialization of the fields for this case. The case is set up to start at time t = 0 s, so the initial field data is stored in a 0 sub-directory of the case directory. The 0 sub-directory in FireFOAM contains 12 files. 9 of them concern physical properties as illustrated in Table (1.3). We need to specify the values of these fields at each grid point as well as the boundary conditions. Table (1.3) shows the fields which need to be defined in the 0 directory.

Table (1.3) Fields that need to be defined in the 0 directory

Properties	SI unit	Comments
α_{sgs}	$kg.m^{-1}.s^{-1}$	Mass density times the Sub-Grid Scale diffusivity
I	$kg.s^{-3}$	Radiation
K	$m^2.s^{-2}$	Kinetic energy
μ_{sgs}	$kg.m.s^{-1}$	Sub-Grid Scale dynamic viscosity
P	$kg.m^{-1.}s^{-1}$	Pressure
P_{rgh}	$kg.m^{-1.}s^{-1}$	$P_{rgh} = P - \rho gh$
T	K	Temperature
U	$m.s^{-1}$	Velocity
Y	-	Mass fraction

Processing:

After the preprocession of the input files for the case, we can run the simulation by typing the solver's name on the terminal. OpenFOAM also supports parallel computing. It uses the public domain openMPI implementation of the standard message passing interface (MPI) by default, other libraries can also be used [5].

Postprocessing:

The main post-processing tool provided with OpenFOAM is a reader module to run with Paraview, an open-source, visualization application. The module is compiled into 2 libraries, PV4FoamReader and vtkPV4Foam using version 4.4.0 of paraview supplied with the OpenFOAM release [6]. It uses the visualization Toolkit (VTK) as its data processing and rendering engine, so it can read any data in VTK format. You can visualize the data at each grid point, as well as get some spatial and temporal statistics by using the utility of Paraview. You can also generate the animation of the simulated fields in Paraview.

The other way to postProcess data consists in using the OpenFOAM post-Processing functionality. It can be executed while the simulation is running by adding object functions in the system/controlDict dictionary as well as be executed after the simulation by running the postProcess command. Both approaches have advantages. Post-simulation post-processing allows the user to choose the way to analyze data after the simulation are completed. Run-time processing provides flexibility since users can access to all the data in the database at all simulation times [7].

Governing Equations:

The instantaneous continuity, momentum, energy for a compressible fluid can be written, in indicial notations, as:

$$\frac{\partial \rho}{\partial t} + \frac{\partial (\rho u_j)}{\partial x_j} = 0 \tag{1.1}$$

$$\frac{\partial (\rho u_i)}{\partial t} + \frac{\partial (\rho u_i u_j)}{\partial x_j} = -\frac{\partial p}{\partial x_j} + \frac{\partial \tau_{ij}}{\partial x_j} + \rho g_i \tag{1.2}$$

$$\frac{\partial (\rho e_0)}{\partial t} + \frac{\partial (\rho u_j e_0)}{\partial x_j} = -\frac{\partial (u_j p)}{\partial x_j} - \frac{\partial q_j}{\partial x_j} + \frac{\partial (u_i \tau_{ij})}{\partial x_j} \tag{1.3}$$

where ρ is the mass density, u_i is the ith component of velocity, p is the pressure, g_i is the ith component of the gravity vector, p is the pressure, T is the temperature. R is the specific gas constant which is defined by $R = R_u/MW$, where R_u is the universal gas constant $(8315 J kmol^{-1} K^{-1})$ and MW is the molecular weight. PIMPLE (merged PISO-SIMPLE) method has been used to solve the pressure-velocity coupling in the low-Mach compressible flow. The Semi-Implicit Method for Pressure-linked equations (SIMPLE) algrorithm allows the calculation of pressure on a mesh from velocity components by coupling the Navier-Stokes equations with an iterative procedure. The Pressure Implicit Splitting Operator (PISO) has been applied in the PIMPLE algorithm to rectify the pressure-velocity correction [8]. More information about SIMPLE and PSIO algorithm can be found in Feriger and Peric (1999) [9] and Issa (1985) [10] respectively. The τ_{ij} is the viscous stress tensor which for a Newtonian fluid given by:

$$\tau_{ij} = 2\mu (D_{ij} - \frac{1}{3} D_{kk} \delta_{ij}) \tag{1.4}$$

where μ is the dynamic viscosity and the rate of strain tensor D_{ij} is defined by:

$$D_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \tag{1.5}$$

e_0 is the total energy defined by:

$$e_0 = e + \frac{u_k u_k}{2} \tag{1.6}$$

where, e is the specific internal energy. The heat-flux, q_j, is given by Fourier's law:

$$q_j = -\lambda \frac{\partial T}{\partial x_j} \tag{1.7}$$

Here T is the temperature and λ is the thermal conductivity.

Code verification and validation:

Computational Fluid Dynamics (CFD) users and developers today face an important question: How to assess the confidence in modeling and simulation? During the last three or four decades, a large number of computer simulation tools have been developed and are routinely used in the design and analysis of engineering projects. Highly complex fluid flow processes can be described by CFD tools, such as turbulent combustion, chemical vapor deposition and multiphase flows. However, the underlying physics of most of these proceses are not understood completely. For such complex processes, it is still hard to achieve a predictive capability based on the first principles and we are not sure if simply increasing computing power can give us a correct solution [11]. To address this problem, several approaches have been developed to

evaluate the performance of CFD solvers and introduce quality control. The two principal approaches are called verification and validation (V&V).

The definition of verification and validation varies among different authors. One compact definition is proposed by Boehm [12]:

Verification: Are we building the model right?

Validation: Are we building the right model?

In other words verification is ensuring that the product has been built according to the requirement and design specifications, while validation ensures that the product meets the user's needs, and that the specifications were correct in the first place. Verification is the assessment of the accuracy of the solution to a computational model, primarily by comparison with known solutions, including exact analytic solution, computational results from previously verified codes and codes that address simplified or specialized cases. Validation is the assessment of the accuracy of a computational simulation by comparison with experimental data [11]. The relationship of the simulation to the real world is not tested in verification. In contrast, in validation, the relationship between the computation and the real world is the main focus.
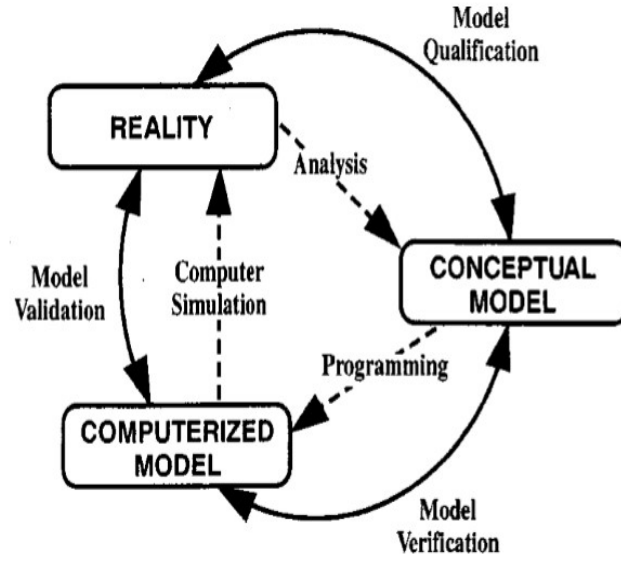
Figure 1.1 Phases of modeling and simulation and the role of V&V[13]

One graphical representation of the fundamental meaning of V&V was proposed by the Society for Computer Simulation (SCS) in 1979, as shown in Fig. 1.1. Two types of model are defined in this figure: a conceptual model and a computerized model. All mathmetical equations are included in Conceptual model. It also consists of its corresponding initial and boundary conditions. We observe and analyze the system in reality to obtain the conceptual model. In CFD, the conceptual model is mainly given by partial differential equations (PDE) for conservation of mass, momentum and energy. All of the auxiliary equations are also included in the conceptual model, for example, the combustion models, turbulence models and constitutive models for materials. The corresponding initial and boundary conditions are included in the conceptual model too. The computerized model is the computer program which is used to solve the conceptual model. Nowadays, the conceptual model can be referred to as mathematical model and the computerized model is often called the code. Figure 1.1 shows that verification is used to handle the consistence between the computerized model and conceptual model and the validation clearly shows the consistence between the computerized model and the real world.

11

The definition of model qualification proposed by SCS is the determination of the adequacy of the conceptual model to provide an acceptable level of agreement for the domain of intended application. We do not do the model qualification in this thesis.

The verification processes consists of static/structural and dynamic/behavioral aspects. Static code verification is the analysis of the code which is performed without actually running a simulation. In most cases the analysis is performed on some versions of source code, and in the other cases, some forms of the object code. Dynamic code verification is performed by executing the code built from that software on a real or virtual processor. For dynamic code verification to be effective, the target code must be executed with sufficient test inputs to produce interesting behavior. Also, it is also important to minimize the effect that instrumentation has on the execution. Inadequate testing can lead to huge failures similar to the maiden flight of the Ariane 5 rocket launcher where dynamic execution errors resulted in destruction of the vehicle [14].

As for validation, it can only be done dynamically, i.e. the code is tested by running benchmark cases.

Outlines of this thesis:

In this thesis, dynamic verification of FireFOAM is performed using several classic CFD cases. Chapter 2 is a verification case focused on the quality of numerical schemes implemented in FireFOAM using the case of the Lamb-Oseen vortex with co-flow. The numerical solution obtained with FireFOAM is compared with an analytical solution [15], which is used to evaluate the numerical error/dissipation of FireFOAM. Chapter 3 is a verification case focused on kinetic energy conservation and enstrophy growth using the Taylor-Green vortex. The enstrophy obtained with FireFOAM is compared with the semi-analytical solution for the enstrophy growth

by Brachet et al. [16] and the kinetic energy is compared with an exact solution, which is trivial since there is no viscosity. Chapter 4 is a validation test of the sub-grid turbulent stress model in FireFOAM using decaying isotropic turbulence, which is a classic turbulent flow problem often used for validation of LES models. The solutions obtained with FireFOAM are compared with the low Reynolds number data of Comte-Bellot and Corrison(CBC) [17]. Chapter 5 is a verification test of the accuracy of boundary conditions using Lamb-Oseen vortex with inflow/outflow boundary conditions (BCs). Several combinations of BCs are used and the solutions are compared with a reference case witch has no boundary influence. Chapter 6 is a validation case of the fire modeling using the classical McCaffrey pool fire experiment [18]. The agreement between the numerical results and experiment data shows that the current CFD model performs well for small-scale fires.

Chapter 2: Lamb – Oseen Vortex with Co-flow

The aim of this case:

We use this case to evaluate the numerical errors (both in time and space) of FireFOAM when solving the Navier-Stokes equation. An analytical solution is available for the Lamb-Oseen vortex model [19], a line vortex that decays due to viscosity. Convective influence is taken into account by adding a co-flow, which can also influence the simulation accuracy. Therefore, this case is relevant to the simulation of a turbulent reacting flow in which errors caused by the numerical dissipation might exist in the part of turbulent flow.

Case initialization:

The domain of this simulation is $0.1\text{m} \times 0.1\text{m}$. The boundary condition in streamwise the x-direction is periodic. The boundary condition in y-direction is zeroGradient, which sets the normal gradient at boundary to zero. We set the co-flow velocity as 0.1m/s. The velocity of the vortex is initialized using Eq. (2.1) and parameters in table 2.1.

$$u_x(x,y) = u_{max}\left(\frac{y-y_c}{r_c}\right)exp\left(\frac{1}{2} - \frac{(x-x_c)^2 + (y-y_c)^2}{2r_c^2}\right) + u_{coflow}$$

$$u_y(x,y) = -u_{max}\left(\frac{x-x_c}{r_c}\right)exp\left(\frac{1}{2} - \frac{(x-x_c)^2 + (y-y_c)^2}{2r_c^2}\right) \quad (2.1)$$

Table (2.1) Vortex parameter

Parameter	Value	Units	Comments
r_c	0.005	m	Core radius of vortex
u_{max}	0.5	m/s	Maximum Velocity
u_{coflow}	0.1	m/s	Co-flow Velocity
(x_c, y_c)	(0, 0)	(m, m)	Centre location
ν	1×10^{-5}	m^2/s	Kinetic viscosity

Figure 2.1 shows the x and y components of the initial velocity field and the magnitude of velocity as well as the magnitude of the initial vorticity field respectively. Initialization of velocity field is used by the command called funkySetFields. We can use this command to set the value of both scalar and vector fields by defining an expression in a dictionary or a command line on terminal. Field values on certain patches can also be set by this command. This command can only be used after the downloading of swak4Foam. In the simulation, we make an approximation that the initial pressure is uniform. As we can see later, such approximation will not significantly influence the results.
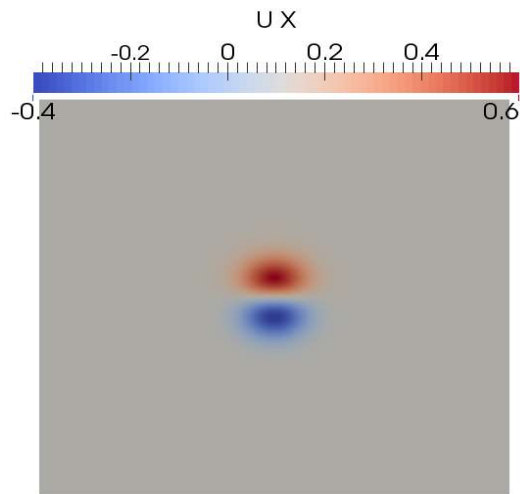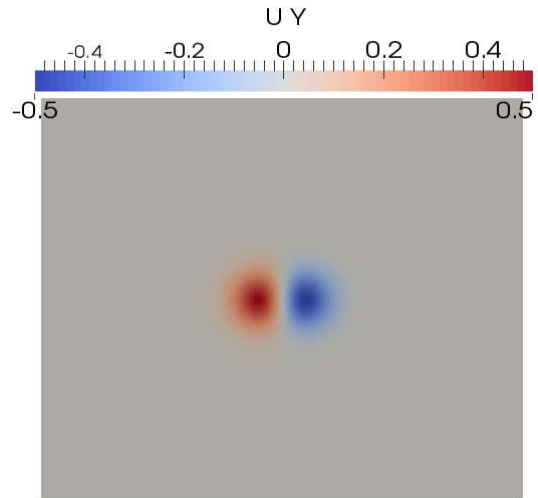
(a)   (b)
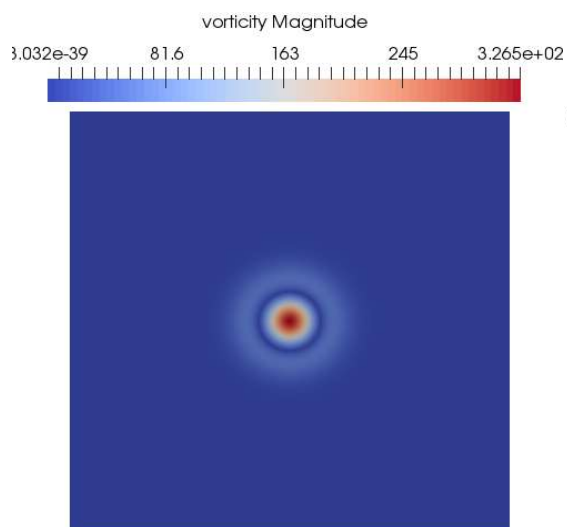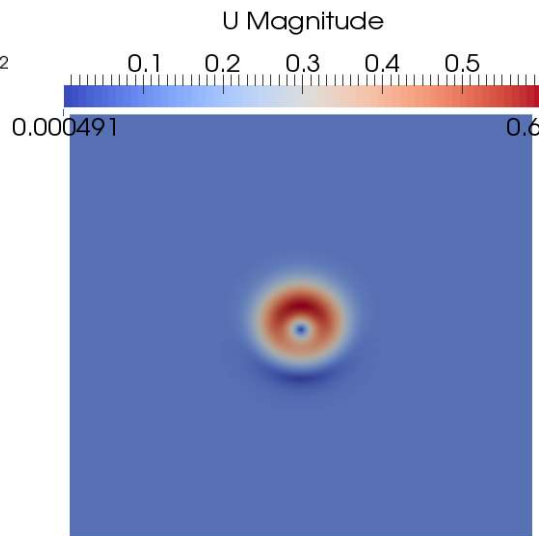
(c)   (d)

Figure 2. 1 Initialization of a single vortex:a) u_x b) u_y

c) magnitude of vorticity d) magnitude of velocity

Analytical Solution:

In our case, the main variable that is used to evaluate the results of the simulation is the vorticity.

The vorticity vector $\vec{w} = \nabla \times \vec{u}$ is defined in cylindrical coordinates by:

$$\vec{\omega} = \begin{pmatrix} \dfrac{1}{r}\dfrac{\partial u_z}{\partial \theta} - \dfrac{\partial u_\theta}{\partial z} \\[2mm] \dfrac{\partial u_r}{\partial z} - \dfrac{\partial u_z}{\partial r} \\[2mm] \dfrac{1}{r}\left(\dfrac{\partial r u_\theta}{\partial r} - \dfrac{\partial u_r}{\partial \theta} \right) \end{pmatrix}$$

In the Lamb-Oseen vortex, the only component of velocity which is not zero is u_θ which may be shown to satisfy:

$$u_\theta(r,t) = u_{max} \left(\frac{r}{r_c}\right) exp\left(\frac{r_c^2 - r^2}{2r_c^2}\right) \tag{2.2}$$

Where r_c and u_{max} are functions of time:

$$r_c^2(t) = r_c^2(t = 0) + 2vt \tag{2.3}$$

$$u_{max}(t) = u_{max}(t = 0) \left(\frac{2vt}{r_c^2(t = 0)} + 1\right)^{-3/2} \tag{2.4}$$

The maximum vorticity is at the center of the vortex, i.e. $r = 0$:

$$\omega_{max}(t) = \frac{2U_{max}(t)}{r_c(t)} \exp\left(\frac{1}{2}\right) \tag{2.5}$$

Results:

Figure 2.2 shows the evolution of the vorticity magnitude during the simulation. The maximum vorticity of the vortex is decreasing with respect to time due to viscous dissipation, the simulation is performed over a period of 2s which contains 2 loops of the vortex motion.

(a)   (b)

vorticity Magnitude

3.032e-39   81.6   163   245   3.265e+02

vorticity Magnitude

3.032e-39   81.6   163   245   3.265e+02

(c)   (d)
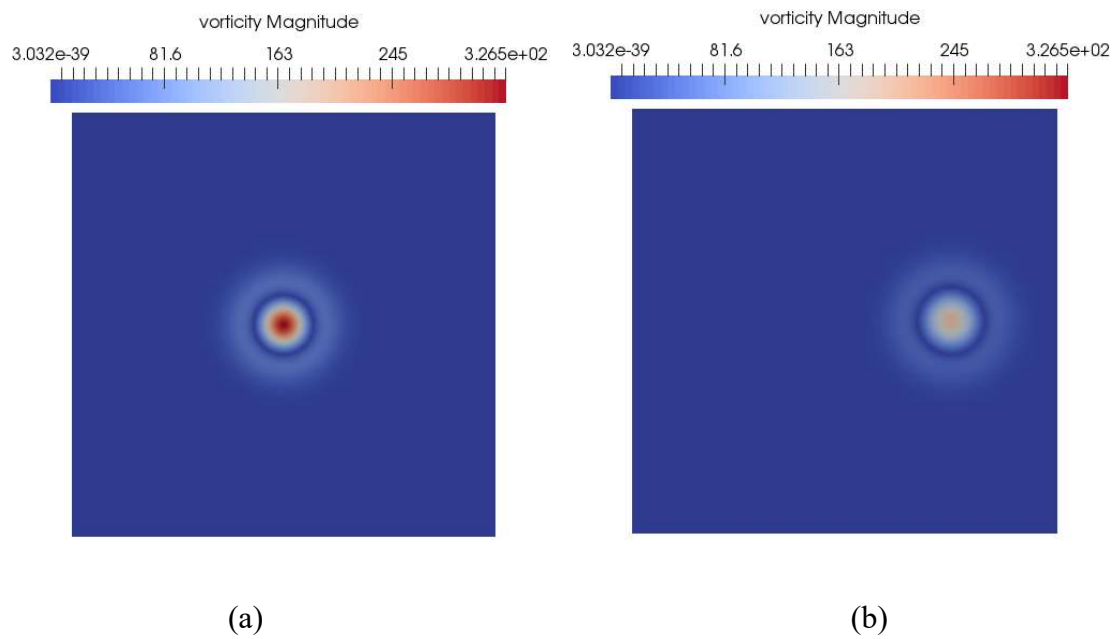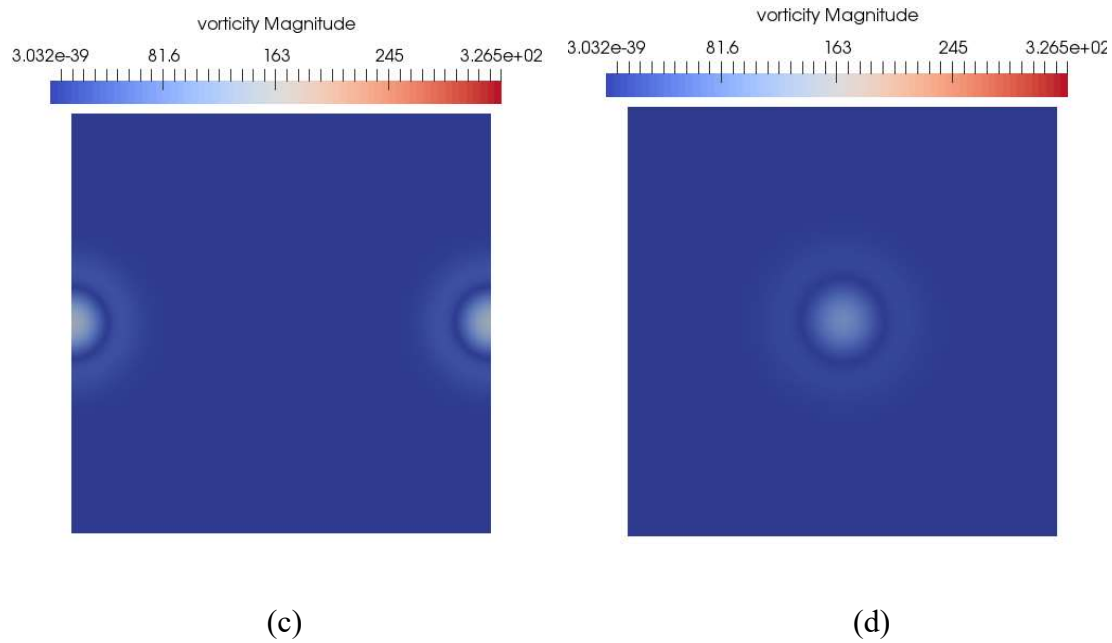
Figure 2. 2 The evolution of the vorticity magnitude with respect to time: a) t=0s b) t =0.25s

c) t=0.5s d) t=0.75s

We run this case using two different temporal schemes, one with a variable time step controlled

by CFL number, i.e. $\Delta t = \frac{CFL \times \Delta x}{u}$, we set $CFL = 0.5$. The other case is ran with a fixed time step

$\Delta t = 2.5 \mu s$.

CFL=0.5

The first case we discuss here is the one with $CFL = 0.5$, the corresponding initial time step of

$125 \mu s$ for a grid resolution $\frac{r_c}{\Delta x} = 40$. First order (Euler) or second order (backward) temporal

schemes have been used in these simulations. The spatial scheme is second order accurate.

Figure 2.3 shows qualitatively that the dissipation of the Euler scheme is larger than that of the

backward method.

19
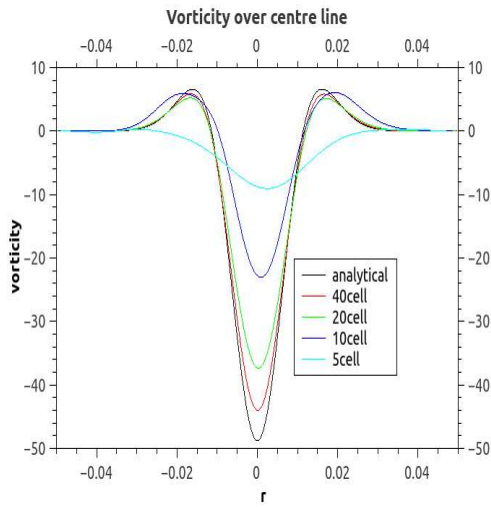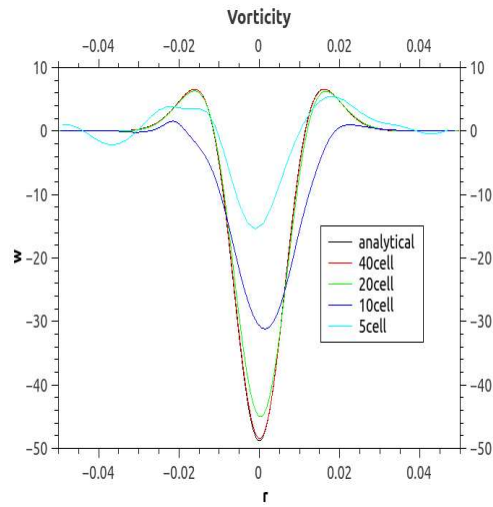
(a)   (b)

Figure 2.3 Vorticity of a single vortex for different grid size when t=2s: a) Euler method b)

backward method.

Figure 2.4 shows the maximum velocity with respect to time, which also shows the Euler scheme

is more dissipative than backward scheme.
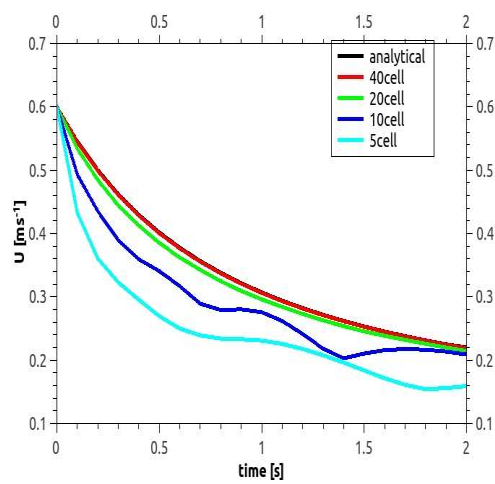
(a)   (b)

Figure 2. 4 Maximum velocity with respect to time (CFL=0.5): a) Euler method b) backward method

Figure 2.5 shows the maximum vorticity with respect to time.

(a)   (b)

Figure 2.5 Maximum vorticity with respect to time (CFL=0.5): a) Euler method b) backward method

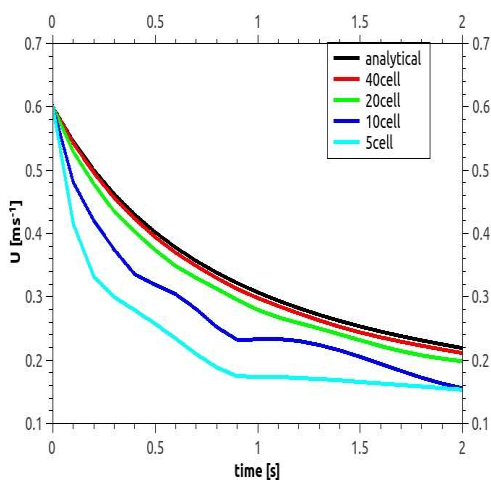Figure 2.6 shows the rms error of the simulations with respect to the grid resolution (5, 10, 20 and 40 cells across the core radius). The Euler scheme is a first order temporal scheme whose error is $O(\Delta t)$. The backward scheme is a second order temporal scheme and the truncation error is $O(\Delta t^2)$. From this figure, we can see that the spatial accuracy of the cases with both schemes are not straight lines with respect to Δx. This indicates that the temporal error plays an important role in this case and we need to reduce such influence of the temporal error to check the scaling of the spatial error.
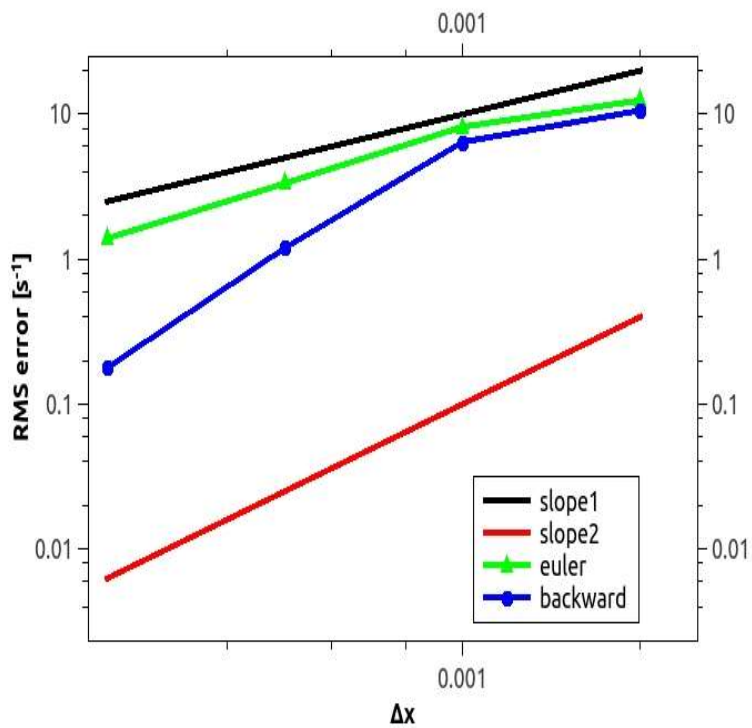
Figure 2.6 RMS error (CFL =0.5): Triangle symbols correspond to the euler numerical scheme and circle symbols to the backward numerical scheme.
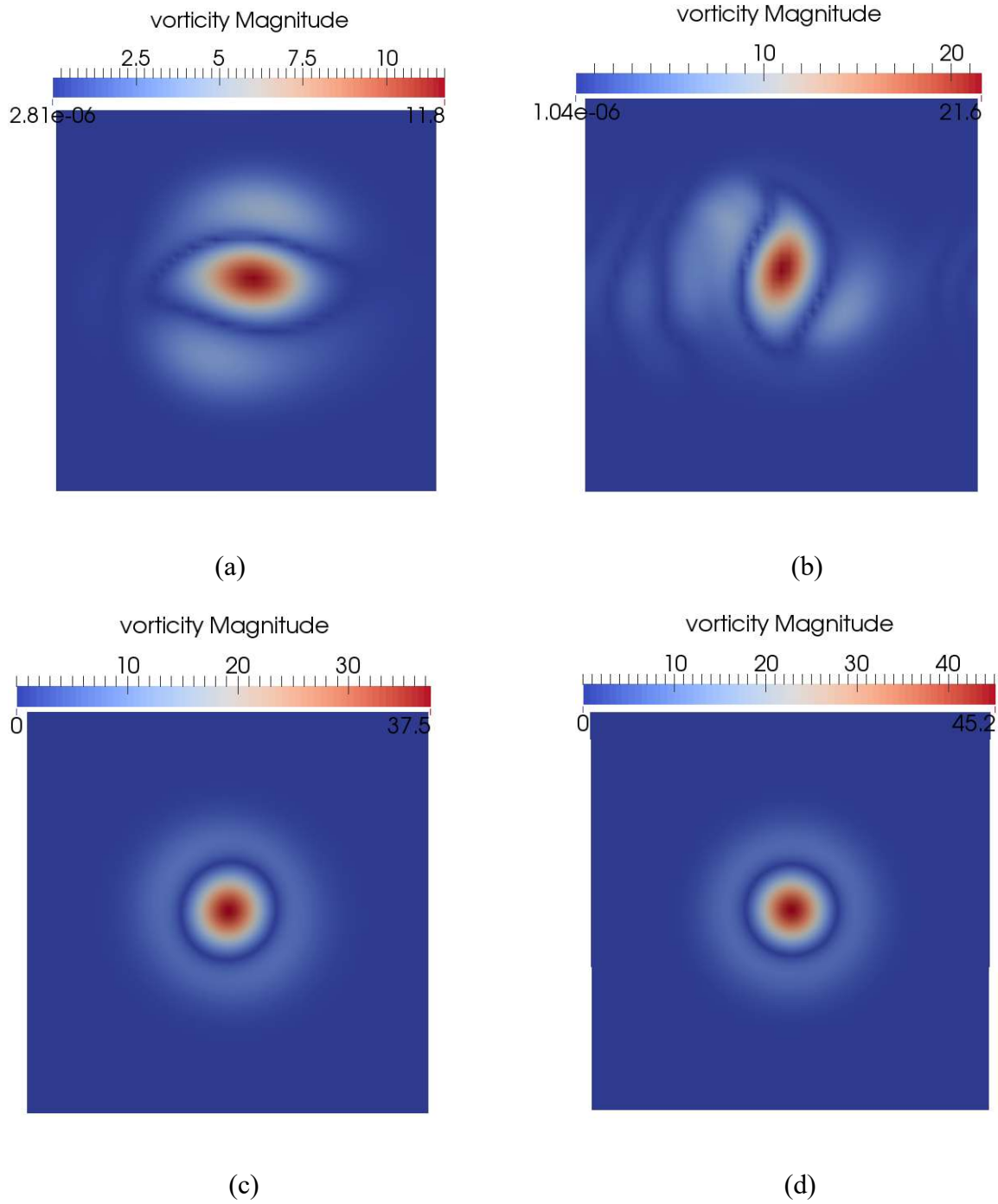
Figure 2.7 Vorticity magnitude for a single vortex with coflow at t=2s : a) 5 cells (Euler); b) 5 cells (backward); c) 20 cells (Euler); (d) 20 cells (backward).

Figure 2.7 presents the vorticity magnitude for a single vortex with coflow at 2s for 5cells and 20 cells and for the Euler and backward schemes. For 5 cells across the core radius, the vortex structure is not correct and the magnitude of vorticity is also lower than the one in the case with 20 cells. This suggests that the dispersion error is important when the grid resolution is not high.

$\Delta t = 2.5 \mu s$

In order to reduce the influence of temporal errors, we choose very small value of time step, $\Delta t = 2.5 \mu s$.

Figure 2.8 shows the maximum vorticity with respect to time when $\Delta t = 2.5 \mu s$. Compare Figure 2.8 with Figure 2.5, we can see the dissipation is much smaller than the one in the case which has the fixed CFL.
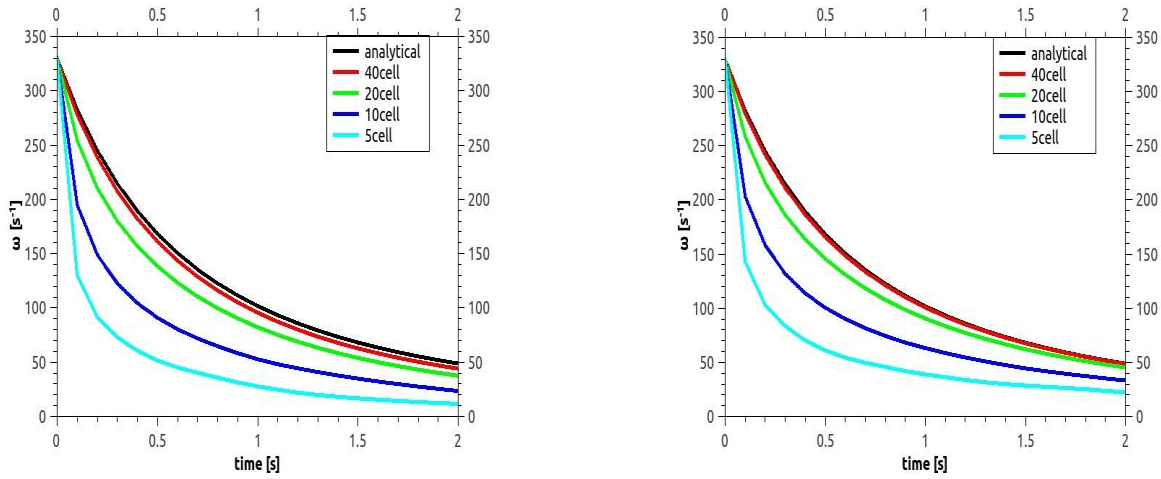

(a)

(b)

24
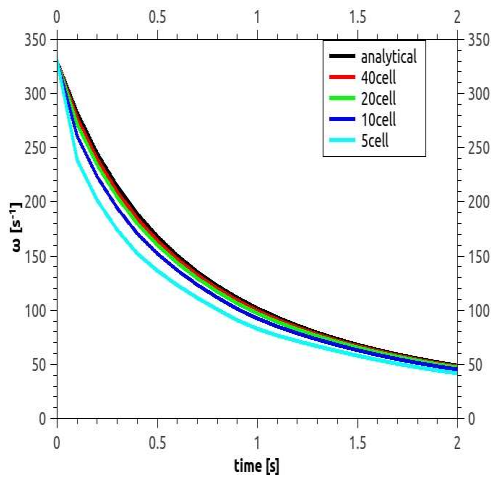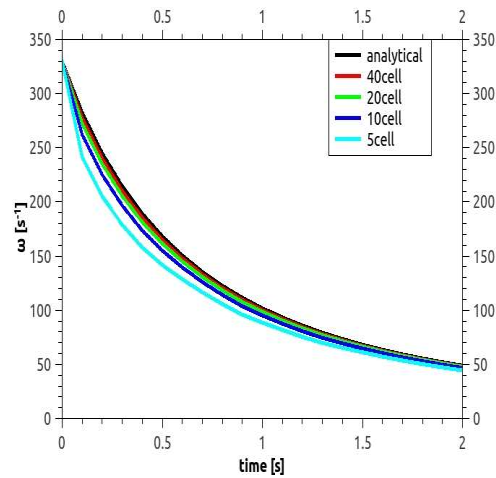
Figure 2.8 Maximum vorticity with respect to time(Δt=2.5μs): a) Euler method b) backward

method

Figure 2.9 shows the rms error with respect to space step. We can see the slope of grid

convergence are 2 which the same as the order of accuracy of spatial schemes used in this case.

The plot of Euler scheme almost overlaps the plot of backward scheme, which suggests the

dissipation caused by time scheme is very small because of the tiny time step.
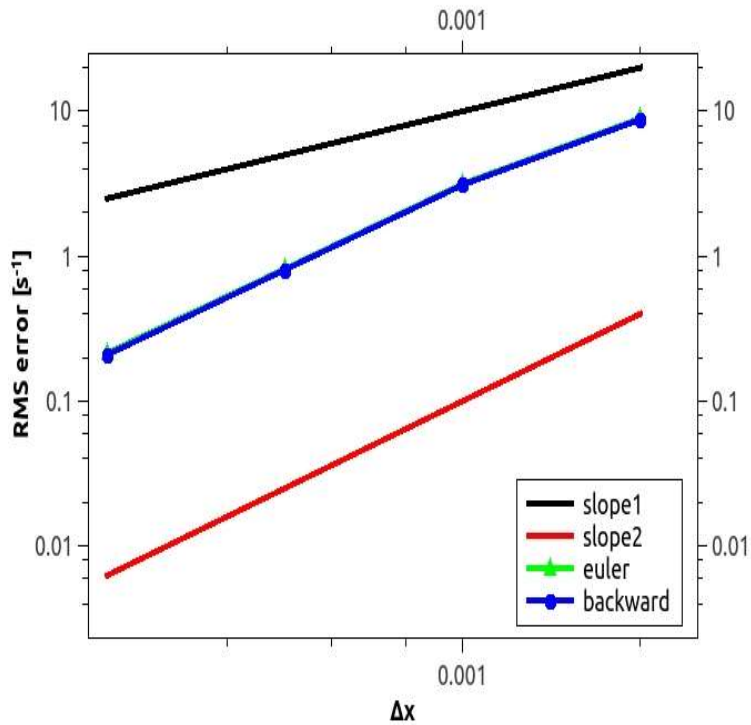
Figure 2.9 RMS error (Δt=2.5μs): Triangle symbols correspond to the Euler method and circle

symbols to the backward method.

Chapter 3: Taylor-Green Vortex

The aim of this case:

The Taylor-Green vortex flow is a simple configuration in which a turbulent energy cascade can

be observed numerically. From a well-resolved initial condition containing only a single length

scale, the flow field undergoes a rapid build-up of scales corresponding to a fully turbulent

dissipative spectrum because of non-linear interactions of the developing eddies [20]. It thus

constitutes a non-regularized problem with no lower bound on the length scale and is solved with

no regularization other than that provided by the numerical method. The resulting flow field

exhibits the features of an isotropic, homogeneous turbulence and is often used in code

verification [21]. The goal of this case is to measure the conservation of kinetic energy and the

enstrophy growth for different numerical schemes.

Case initialization:

The domain $x_i = [0,2\pi]$ is periodic and the grid resolution is $\Delta x_i = 2\pi/64$. And the initial

conditions are:

$$\rho = 1 \tag{3.1a}$$

$$u = \sin(x_1)\cos(x_2)\sin(x_3), \tag{3.1b}$$

$$v = -\cos(x_1)\sin(x_2)\cos(x_3), \tag{3.1c}$$

$$w = 0, \tag{3.1d}$$

$$p = 100 + \frac{[\cos(2x_3) + 2][\cos(2x_1) + \cos(2x_2)] - 2}{16} \tag{3.1e}$$

26

These initial conditions satisfy the continuity equation $\nabla \cdot \vec{u} = 0$. And the mean pressure is sufficiently high to make the problem essentially incompressible [22]. These conditions can be imposed by using the funkySetFields command.

(a)

(b)
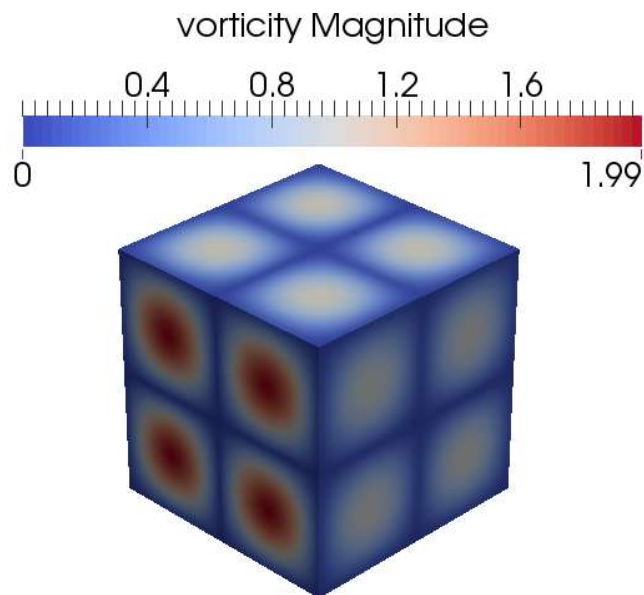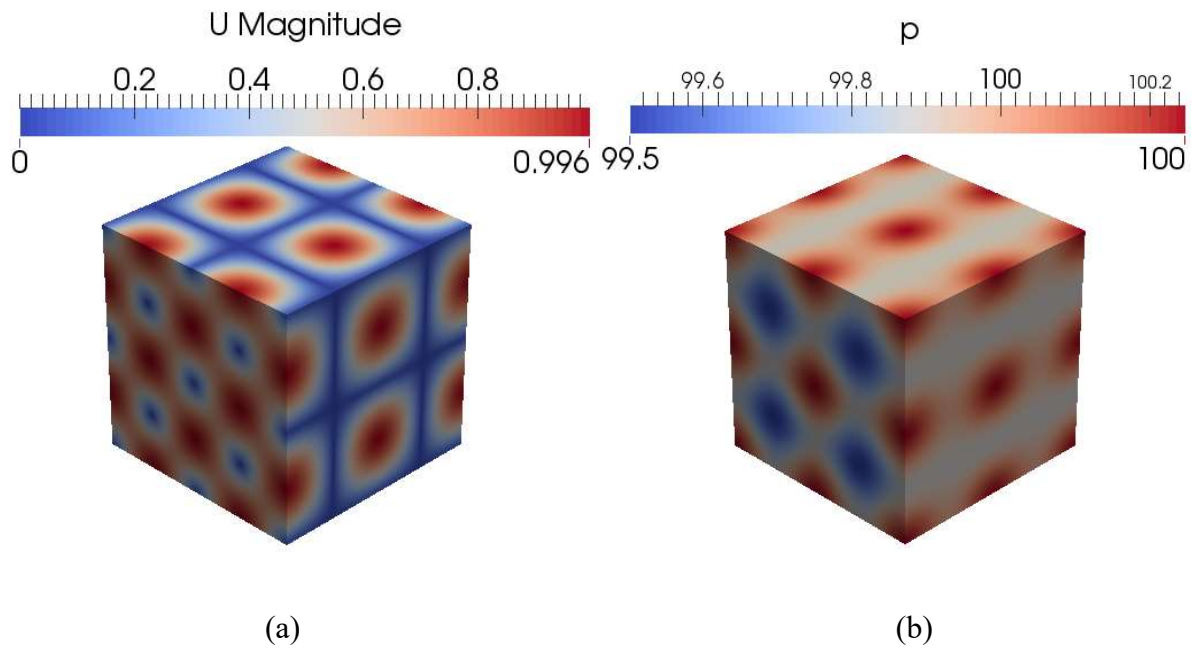
Figure 3.1 Initial condition for the Taylor-Green vortex case. a) velocity magnitude b) pressure magnitude c) vorticity magnitude

Results:

The reference solution of enstrophy growth I used is the semi-analytical solution given by Brachet et al [23]. for $t \leq 4$. In inviscid problems, the kinetic energy does not change while the enstrophy grows rapidly. Eq. (3.2) is the governing equation of vorticity [24]:

$$\frac{\partial \vec{\omega}}{\partial t} = -\vec{u} \cdot \nabla \vec{\omega} + \vec{\omega} \cdot \nabla \vec{u} + \nu \nabla^2 \vec{\omega} \tag{3.2}$$

Without viscosity, this equation turns into:

$$\frac{D\vec{\omega}}{Dt} = \vec{\omega} \cdot \nabla \vec{u} \tag{3.3}$$

Only the vortex stretch make the enstrophy grow. Figure 3.2 shows the temporal evolution of the mean kinetic energy, $< \rho u_i u_i >/2$, and enstrophy, $< w_i w_i >/2$, normalized by their initial values. The brackets denote averaging over all space: for a function q.

$$< q > = \frac{1}{L^3} \iiint\limits_{0}^{L} q(x_1, x_2, x_3) dx_1 dx_2 dx_3 \tag{3.4}$$

It is seen in Figure 3.2 that all the time schemes will cause dissipation. After 4s, all the schemes under-predict the kinetic energy and the enstrophy growth of the field. All temperal schemes agree well with the semi-analytical results for the enstrophy growth.

(a)   (b)
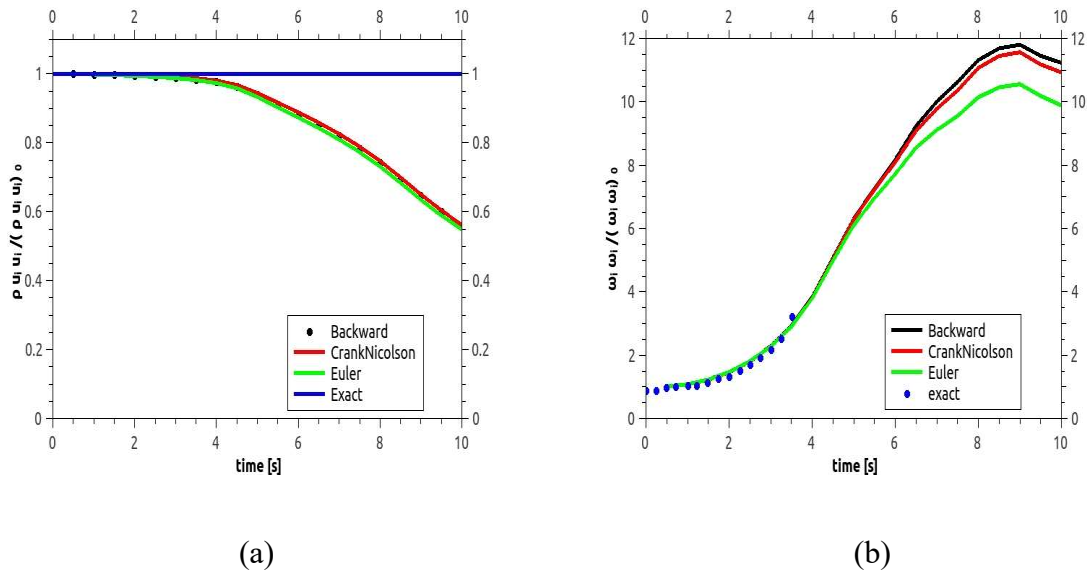
Figure 3.2: Mean quantities for the Taylor-Green vortex on a 64^3 grid. The zero subscripts

denotes the initial value. (a) Kinetic energy (b) Enstrophy. The semi-analytical solution of

Brachet et al. (1983) are the blue symbols.

Table (3.1) shows the normalized mean kinetic energy of different numerical schemes at $t = 5s$;

at this time, there are some vortices smaller than the grid size and the dissipation effect. The

mean enstrophy normalized by its initial value is also shown at $t = 3.5s$; this is the last time for

which the semi-analytical solution is known.

Table (3.1) Comparison of global measures of kinetic energy and enstrophy in FireFOAM

simulation performed with different temporal schemes for the Taylor-Green vortex case.

	Euler $O(\Delta t, \Delta x^2)$	Backward $O(\Delta t^2, \Delta x^2)$	Crank-Nicholson $O(\Delta t^2, \Delta x^2)$	Exact solutions Brachet et al.[5]
Kinetic Energy t = 5s	0.936	0.945	0.952	1.00
Enstrophy t = 3.5s	2.83	2.85	2.89	3.46

Table (3.2) reports quantitative metrics for the Taylor-Green vortex case for high-order solvers taken from the literature [22].

Table (3.2) Global measures for the Taylor-Green vortex case for other high-order solvers [22].

	Hybrid $O(\Delta t^4, \Delta x^6)$	Stan $O(\Delta t^4, \Delta x^6)$	WENO $O(\Delta t^4, \Delta x^5)$	Brachet et al.[23]
Energy t = 5s	1.00	0.976	0.916	1.00
Enstrophy t = 3.5s	3.33	3.23	3.13	3.46

Figure 3.3 presents the time evolution of the flow structure. The Q-criterion is used to visualize the vortical structure, with $Q = 0.1$. The initial Taylor-Green vortex is observed to stretch. Some smaller scales are produced. The 3D structure is colored by the vorticity magnitude to illustrate the enstrophy growth, i.e. the production of vorticity.

(a)

(b)

(c)

(d)

<div align="center">(e)   (f)</div>
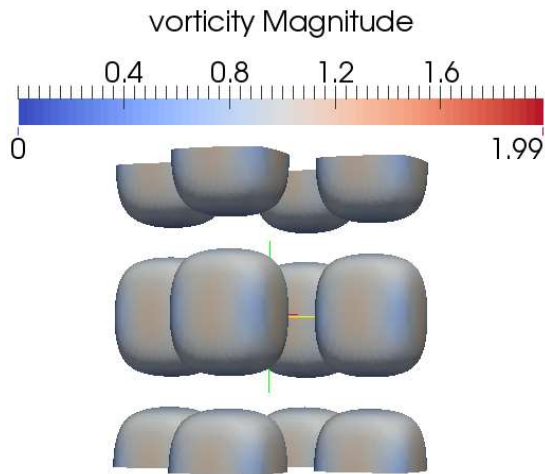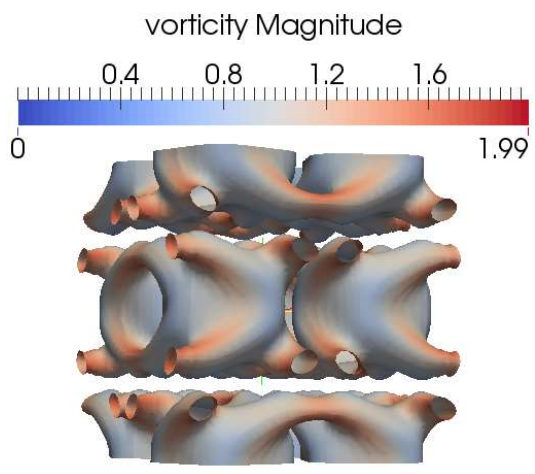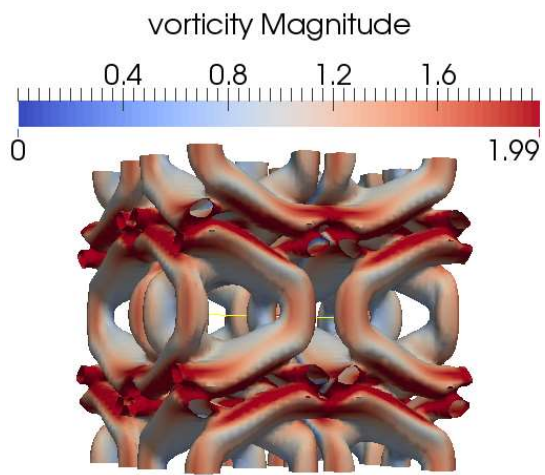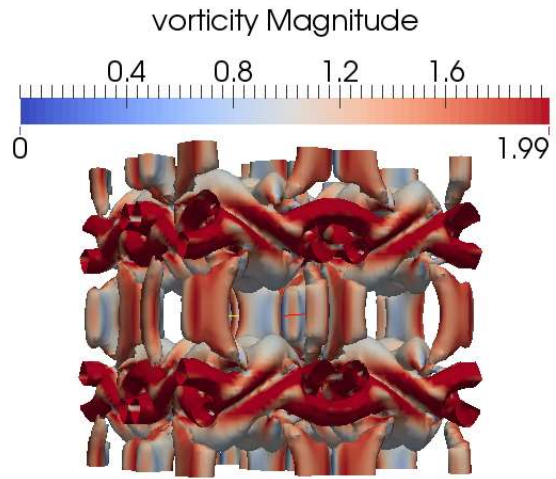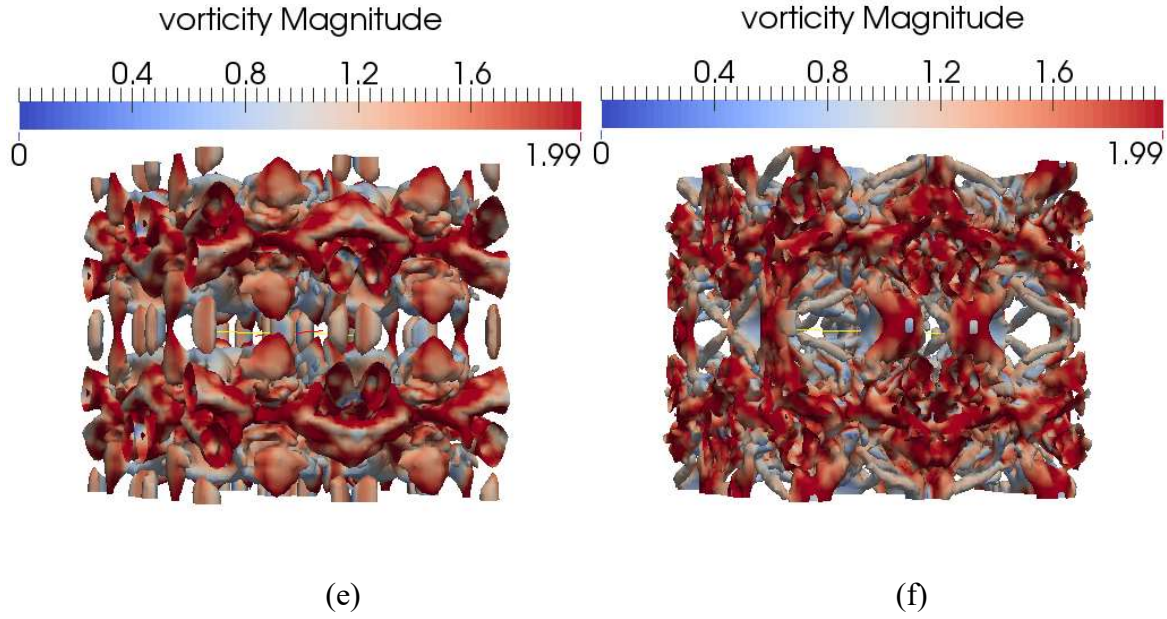
Figure 3. 3: Taylor-Green vortex at different times using the Q-criterion with Q=0.1 colored by vorticity magnitude. (a) $t = 0s$ (b) $t = 2s$ (c) $t = 4s$ (d) $t = 6s$ (e) $t = 8s$ (f) $t = 10s$

Grid Convergence

As we can see above (Table 3.1), the second order spatial and temporal schemes used in FireFOAM are slightly less accurate than other high order schemes. Therefore, we need to refine the grid to make the simulation achieve the same level of accuracy. Table 3.3 shows the normalized kinetic energy and enstrophy for a 64^3 and 128^3 grid.

Table (3.3) Global measures for the Taylor-Green vortex case for different mesh resolution [23].

	64^3 Cells $O(\Delta t^2, \Delta x^2)$	128^3 Cells $O(\Delta t^2, \Delta x^2)$	Exact solutions Brachet et al.[5]

Energy t = 5s	0.952	0.965	1.00
Enstrophy t = 3.5s	2.89	3.23	3.46

Chapter 4: Isotropic Turbulence Decay

The aim of this case:

Isotropic turbulence is the turbulent flow whose statistical properties are invariant under all axis rotations. Since physically interesting properties include joint probabilities of field variables at two or more space points, isotropy requires homogeneity as well. Decaying isotropic turbulence has been the subject of many experimental and numerical studies. Due to the vast database available, this flow is a very popular case used for testing the realization of numerical schemes and closure models. In this section, we implemented this classical flow to examine whether the sub-grid stress (SGS) model in FireFOAM has been coded correctly. This test is a validation test rather than a verification, but can be treated as a complementary test to other verification tests performed in the previous sections. The low Reynold number experimental results of Comte-Bellot and Corrsin (CBC) [25] are used to test the numerical results obtained with FireFOAM.

Case initialization:

The computational domain is a box with periodic boundaries of side $L = 9 \times 2\pi$ cm ($\approx 0.566m$) and $\nu = 1.5 \times 10^{-6} m^2/s$ for the kinematic viscosity. The non-dimensional times for these data are: $\frac{x}{M} = 42$(initial conditions), 98 and 171, where M is the characteristic mesh spacing of the CBC wind tunnel and x is the downstream location of this data station. Concerning the average velocity in the CBC experiment, these correspond to dimensional times of $t = 0.00s, 0.28s \ and \ 0.66s$ in our time-evolving simulations. We get the initial velocity field by superimposing Fourier modes. In our case, we assume that the initial density and pressure fields

are constant, note that a more physically realistic initial condition can be obtained by initializing the density and pressure fields using the method by Ristorcelli and Blaisdell [22].

The initial velocity field is given in terms of its Fourier coefficients as

$$\hat{u}(k_1, k_2, k_3) = (\frac{k_2}{k_{12}}a + \frac{k_1}{k_{12}}\frac{k_3}{k}b, \frac{k_2}{k_{12}}\frac{k_3}{k}b - \frac{k_1}{k_{12}}a, -\frac{k_{12}}{k}b) \qquad (4.1)$$

Where k_i are the wavenumbers, $k = \sqrt{k_i k_i}$ is the wavenumber magnitude, and $k_{12} = \sqrt{k_1^2 + k_2^2}$.

This is valid for all wavenumbers other than (0, 0, 0) provided one defines that $\frac{k_1}{k_{12}} = 0$ and $\frac{k_2}{k_{12}} = 1$ for $k_{12} = 0$. The quantities a and b are:

$$a = \sqrt{\frac{2E(k)}{4\pi k^2}}e^{i\varphi_1}cos\varphi_3, b = \sqrt{\frac{2E(k)}{4\pi k^2}}e^{i\varphi_2}cos\varphi_3, \qquad (4.2)$$

Where $\varphi_1, \varphi_2,$ and φ_3 are random numbers uniformly distributed on $[0, 2\pi]$ that are re-sampled for each wavenumber triplet. The energy spectrum function is

$$E(k) = u_{rms}^2 16\sqrt{\frac{2}{\pi}\frac{k^4}{k_0^5}}\exp\left(-\frac{2k^2}{k_0^2}\right), \qquad (4.3)$$

Where k_0 is the most energetic wavenumber. The energy spectrum function is such that $\frac{3u_{rms}^2}{2} = \int_0^\infty E(k)\,dk$.

(a)   (b)
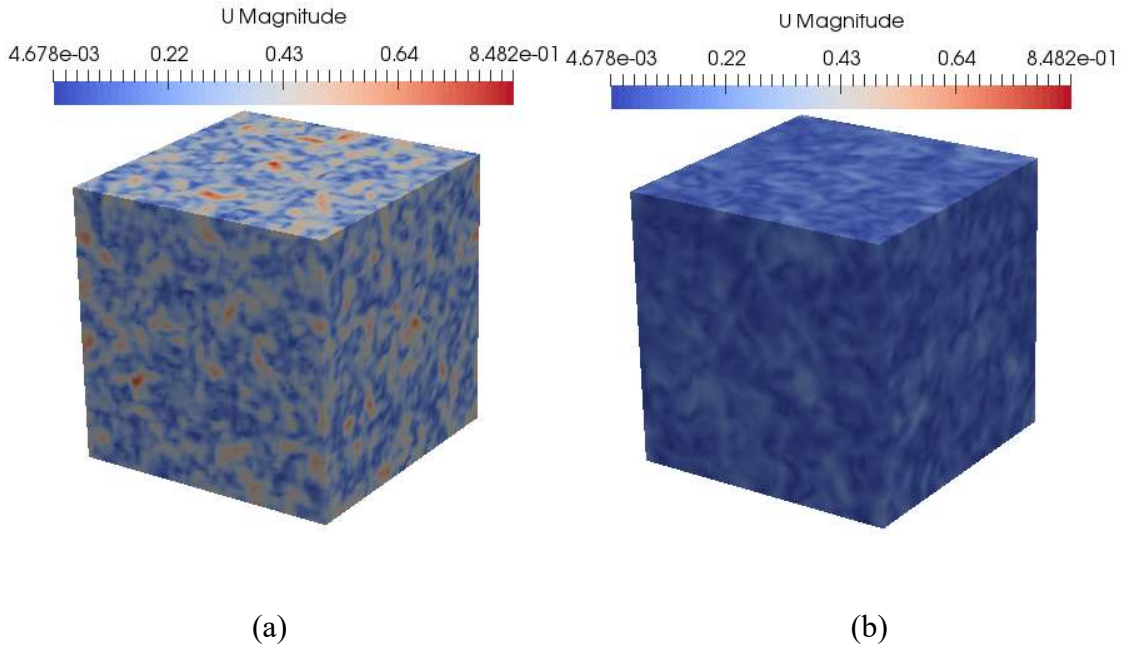
Figure 4.1: Velocity magnitude for the isotropic turbulence field. a) $t = 0s$ b) $t = 2s$

Figure 4.1 shows the initial and final states of the velocity field in the 3D domain. The flow is unforced and the total kinetic energy decays with time due to viscous dissipation.

Now we will define the different turbulent length scales of this flow and calculate the corresponding wave-numbers (k).

Maximum length scale (L): largest eddies in the flow, constrained by the physical boundaries of the flow. L = 0.566m

$$k_L = \frac{2\pi}{L} = 11[1/m] \tag{4.4}$$

Kolmogrov length scale (η_k): smallest eddies in the flow, $\eta_k = 2.9 \times 10^{-4}$m

$$k_\eta = \frac{2\pi}{\eta} = 21{,}666[1/\text{m}] \qquad (4.5)$$

Nyquist limit: smallest eddies resolved by the grid. The characteristic wavenumber, k_c, corresponds to one period and is defined as:

$$k_c = \frac{2\pi}{2\Delta} = 355[1/m] \qquad (4.6)$$

Where Δ is the filter width and is defined as $\Delta = (\Delta_x \Delta_y \Delta_z)^{1/3}$.

The contribution to the kinetic energy k from motions in wavenumber range (k_a, k_b) is

$$k_{(k_a,k_b)} = \int_{k_a}^{k_b} E(k)\, dk \qquad (4.7)$$

Turbulence Model:

Three turbulence models are tested in this case, i.e. the Smagroinsky SGS model, the constant coefficient one-equation model and the dynamic coefficient one-equation model. I will briefly introduce these three models in OpenFOAM.

Smagorinsky SGS model:

The Smagorinsky subgrid scale (SGS) model was developed by Joseph Smagorinsky in the meteorological community in the 1960s [26]. It is based on the eddy viscosity assumption, which postulates a linear relationship between the SGS shear stress and the resolved rate of strain tensor. This model serves as a reference for other SGS models.

The Smagorinsky SGS model is the oldest and best known subgrid scale model. The subgrid

scale stress tensor τ_{ij} is split into an isotropic part $\frac{1}{3}\tau_{kk}\delta_{ij}$ and an anisotropic part $\tau_{ij} - \frac{1}{3}\tau_{kk}\delta_{ij}$.

$$\tau_{ij} = \overline{u_i u_j} - \bar{u}_i \bar{u}_j$$

$$= \frac{1}{3}\tau_{kk}\delta_{ij} + (\tau_{ij} - \frac{1}{3}\tau_{kk}\delta_{ij}) \tag{4.8}$$

In analogy with the molecular viscous stress in laminar flows, the anisotropic part is

approximated by relating it to the resolved rate of strain tensor $\bar{D}_{ij}$.

$$\tau_{ij} - \frac{1}{3}\tau_{kk}\delta_{ij} = -2\nu_{sgs}dev(\bar{D})_{ij} \tag{4.9}$$

where ν_{sgs} is the subgrid scale eddy viscosity, $dev(\bar{D})_{ij}$ is the anisotropic part of $\bar{D}$ and the

resolved-scale strain rate tensor $\bar{D}_{ij}$ is defined as:

$$\bar{D}_{ij} = \frac{1}{2}(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i}) \tag{4.10}$$

In OpenFOAM, the subgrid scale viscosity is computed as

$$\nu_{sgs} = C_k \Delta \sqrt{k_{sgs}} \tag{4.11}$$

Where C_k is a model constant whose default value is 0.094 and Δ is the grid size that defines the

LES filters length scale. The method for calculating Δ is specified in the turbulenceProperties

file. the subgrid scale kinetic energy k_{sgs} is :

$$k_{sgs} = \frac{1}{2}\tau_{kk} = \frac{1}{2}(\overline{u_i u_j} - \bar{u}_k \bar{u}_k) \tag{4.13}$$

The isotropic part can be written into

$$\frac{1}{3}\tau_{kk}\delta_{ij} = \frac{2}{3}k_{sgs}\delta_{ij}$$

(4.14)

Where the SGS kinetic energy k_{sgs} is computed with the assumption of the balance between the subgrid scale energy production and the dissipation (local equilibrium)

$$\bar{D}:\tau \ + C_\epsilon \frac{k_{sgs}^{1.5}}{\Delta} = 0$$

(4.15)

Where the operator : is a double inner product of two second-rank tensors that can be evaluated as the sum of the 9 products of the tensor components. C_ϵ is the dissipation term model coefficient which equals to 1.048. We can compute k_{sgs} by solving Eq.(4.16) as shown below:

$$\bar{D}:\left(\frac{2}{3}k_{sgs}I - 2\nu_{sgs}dev(\bar{D})\ \right) + C_\epsilon \frac{k_{sgs}^{1.5}}{\Delta} = 0$$

$$\Rightarrow \bar{D}:\left(\frac{2}{3}k_{sgs}I - 2C_k\Delta\sqrt{k_{sgs}}dev(\bar{D})\ \right) + C_\epsilon \frac{k_{sgs}^{1.5}}{\Delta} = 0$$

$$\Rightarrow \bar{D}:\sqrt{k_{sgs}}\left(C_\epsilon \frac{k_{sgs}}{\Delta} + \frac{2}{3}tr(\bar{D}) - 2C_k\Delta(dev(\bar{D}) : \bar{D}\ \right) += 0$$

$$\Rightarrow ak_{sgs} + b\sqrt{k_{sgs}} - c = 0$$

$$\Rightarrow k_{sgs} = (\frac{-b + \sqrt{b^2 + 4ac}}{2a})^2$$

(4.16)

Where

$$\begin{cases} a = \dfrac{C_\epsilon}{\Delta} \\ b = \dfrac{2}{3} tr(\overline{D}) \\ c = 2C_k\Delta(dev(\overline{D}):\overline{D}) \end{cases}$$

In the case of incompressible flows, it reduces to

$$b = 0$$

$$c = C_k\Delta|\overline{D}|^2 \tag{4.17}$$

Where

$$|\overline{D}|^2 = \sqrt{2\overline{D}:\overline{D}}$$

By substituting these relations into Eq (4.16), we can get:

$$k_{sgs} = \frac{c}{a} = \frac{C_k\Delta^2|\overline{D}|^2}{C_\epsilon} \tag{4.18}$$

We can get the following expression for the SGS eddy viscosity in the case of incompressible

flows by substituting Eq. (4.18) into Eq.(4.11)

$$\tag{4.19}$$
$$\nu_{sgs} = C_k\sqrt{\frac{C_k}{C_\epsilon}}\Delta^2|\overline{D}|$$

Constant coefficient One-equation eddy-viscosity SGS model in OpenFOAM:

The second category of SGS model is one-equation eddy viscosity models. The main reason to

develop the one-equation SGS model is to overcome the limitation of the assumption of the

balance between the SGS energy production and dissipation adopted in algebraic eddy viscosity

models. The first one-equation eddy viscosity SGS model was theoretically derived by

Yoshizawa and Horiuti [27] based on solving a transport equation for the SGS turbulent kinetic energy k_{sgs} which is:

$$\frac{\partial \rho k_{sgs}}{\partial t} + \frac{\partial \rho k_{sgs} u_i}{\partial x_i} = \frac{\partial}{\partial x_i}\left(\rho(v + v_{sgs})\frac{\partial k_{sgs}}{\partial x_i}\right) + P - \varepsilon_{sgs} \qquad (4.20)$$

ε_{sgs} is the sub-grid eddy dissipation rate which is calculated as:

$$\varepsilon_{sgs} = C_\varepsilon k_{sgs}^{3/2}/\Delta \qquad (4.21)$$

The production term, P is expressed as:

$$P = -\frac{2}{3}\bar{\rho}\left(k_{sgs} + v_{sgs}\frac{\partial \bar{u}_k}{\partial x_k}\right)\frac{\partial \bar{u}_i}{\partial x_i} + 2\bar{\rho}v_{sgs}\bar{D}_{ij}\bar{D}_{ij} \qquad (4.22)$$

The first term is the turbulent kinetic energy variation caused by thermal expansion and the second term is caused by strain rate.

Dynamic coefficient one-equation model in OpenFOAM:

Dynamic coefficient one-equation model can be expressed as the dynamic version of constant coefficient one-equation model [28]. Using constant coefficient coefficient SGS model is difficult to simulate the transition from laminar to turbulent flow. It cannot present the fast transition of SGS kinetic energy during the flow transition from laminar flow to turbulence. This dynamic model uses the grid-scale (GS) level information to evaluate the coefficients of the SGS model. More detailed procedure to calculate C_ε and C_k can be found in [29].

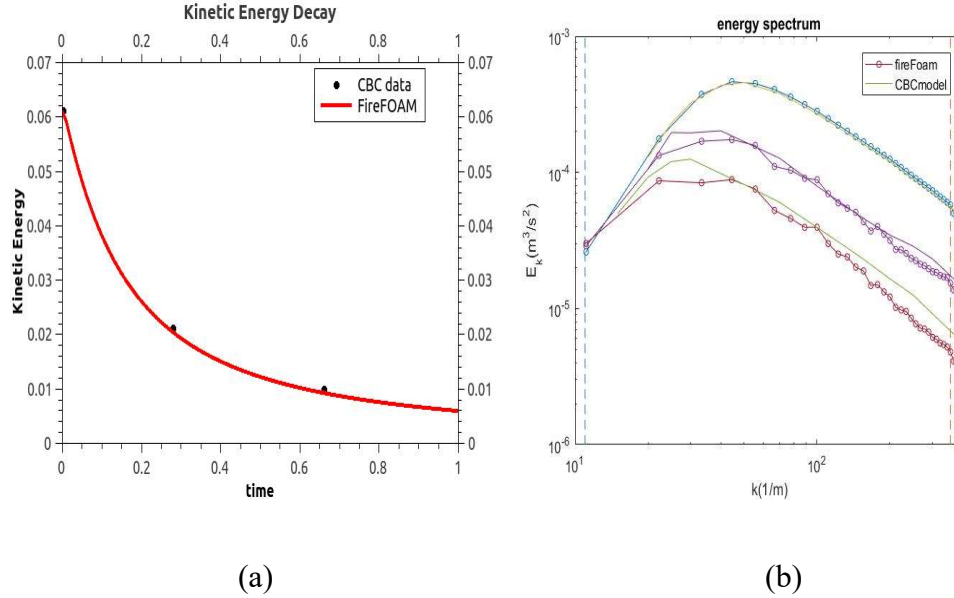(a)   (b)

Figure 4.2: Kinetic Energy comparison between CBC data and FireFOAM using the Smagorinsky turbulence model for a 64^3 grid resolution. a) Kinetic Energy decay over time; b) Partition of kinetic energy versus wavenumber. The first vertical dashed line is the wavenumber for the maximum length scale (k_L) and the second vertical dashed is the wavenumber for the Nyquist limit k_C.

Figure 4.2 (a) shows the decay of kinetic energy for a 64^3 grid resolution. The discrepancy between FireFOAM and CBC is less than 10%.

Figure 4.2 (b) shows the corresponding spectral data. The three block solid lines are the CBC spectral data for points in time corresponding to times $t = 0.0s, 0.28s, 0.66s$. The initial

FireFOAM velocity field is specified to match the CBC kinetic energy up to the grid Nyquist limit. Kinetic energy is under-predicted for low wavenumber containing most of the energy.
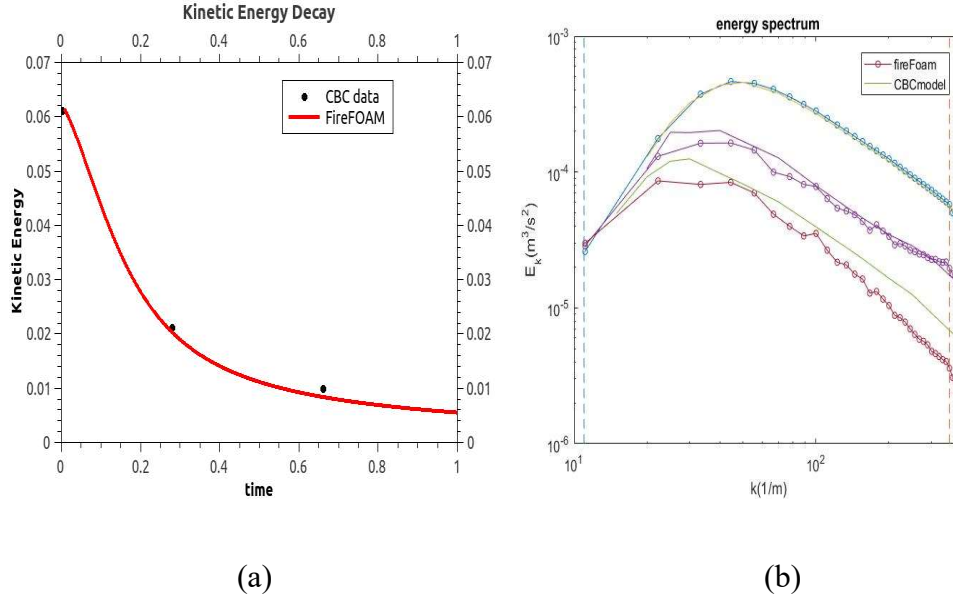

(a)   (b)

Figure 4. 3: Kinetic Energy comparison between CBC data and FireFOAM using the constant coefficient one-equation turbulence model for a 64^3 grid resolution. a) Kinetic Energy decay over time; b) Partition of kinetic energy versus wavenumber. The first vertical dashed line represents the wavenumber for the maximum length scale (k_L) and the second vertical dashed line represents the wavenumber for the Nyquist limit k_C.

Figure 4.3 (a) shows the decay of kinetic energy for a 64^3 grid resolution. The discrepancy between FireFOAM and CBC is less than 15%.

Figure 4.3 (b) shows the corresponding spectral data. The three block solid lines are the CBC spectral data for points in time corresponding to times $t = 0.0s, 0.28s, 0.66s$. Kinetic energy is under-predicted for low wavenumber containing most of the energy. This model is less accurate than the Smagorinsky model.
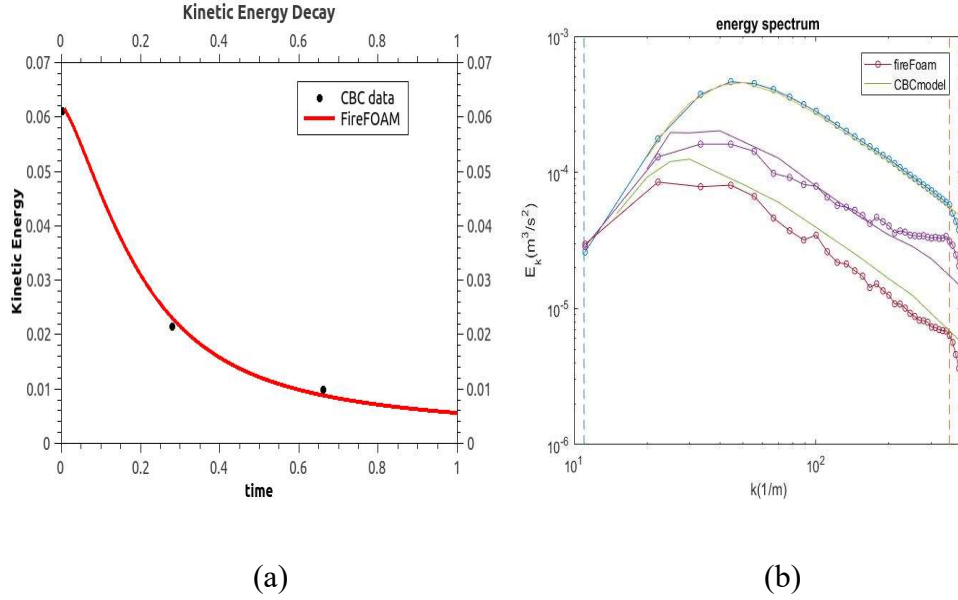
(a)   (b)

Figure 4.4: Kinetic Energy comparison between CBC data and FireFOAM using the dynamic coefficient one-equation turbulence model for a 64^3 grid resolution. a) Kinetic Energy decay over time; b) Partition of kinetic energy versus wavenumber. The first vertical dashed line represents the wavenumber for the maximum length scale (k_L) and the second vertical dashed line represents the wavenumber for the Nyquist limit k_C.

Figure 4.4 (a) shows the decay of kinetic energy for a 64^3 grid resolution. The discrepancy between FireFOAM and CBC is less than 10%.

Figure 4.4 (b) shows the corresponding spectral data. The three block solid lines are the CBC spectral data for points in time corresponding to times $t = 0.0s, 0.28s, 0.66s$. Kinetic energy is under-predicted for low wavenumber containing most of the energy. This model is the most accurate of the three models tested.

Chapter 5: The verification of boundary condition

The aim of this case:

Boundary conditions (BC) are very important in CFD. Almost every CFD problem is defined under the limit of boundary conditions. In most cases, we need to carefully evaluate BC to make the simulation accurate. In all previous cases, we used the periodic boundary conditions which approximates a large(infinite) system by using a small part called a unit cell. This BC is often used in computer simulations and mathematical models. However, to simulate the system in the real world, we often need to use open inflow/outflow BC which have limited accuracy. The intent in this chapter is to try different BCs implemented in FireFOAM in two classic problems, i.e. a Lamb-Oseen vortex case and a buoyant hot bubble case, to test the validity of BCs in these cases and find out an optimal combination.

Lamb-Oseen vortex:

In this case, we focus on the BC for the pressure field and the velocity field, which are the most important ones.

Case Initialization:

The initialization of the Lamb-Oseen vortex has been described in chapter one.

Analytical Solution:

In this open boundary problem, the analytical solution turns into a piecewise function which represents the different stages of the vortex transport in the domain, i.e. the vortex is in the domain, the vortex is leaving and the vortex has already left.

Velocity field:

When $0 < t \le 0.5$s, the maximum velocity of the fields equals:

$$U_{max\ Total} = U_{max} + U_{coflow} \qquad (5.1)$$

Here U_{max} is the maximum velocity of the Taylor-Green vortex as shown in Eq.(2.4). U_{coflow} is the velocity of co-flow which is $0.1 m/s$ in our case. So when the vortex centre is in the domain, the analytical solution of $U_{maxTotal}$ is:

$$U_{maxTotal} = U_{max} + U_{coflow} = U_{max}(t = 0)\left(\frac{2vt}{r_c^2(t = 0)} + 1\right)^{-3/2} + 0.1 \qquad (5.2)$$
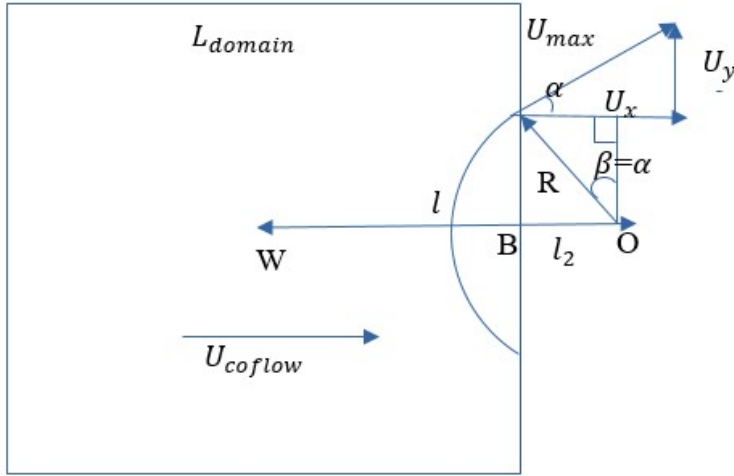

Figure 5.1 Analysis of the velocity field when the vortex partially leaves the domain. U_{max} is the maximum velocity of the vortex, U_x and U_y is the x-component and y-component of U_{max} respectively. α is the angle between U_x and U_{max} and by the geometry relation it also equals to the angle β. R is the radius of the vortex. O is the center of the vortex. W is the center of the

domain. B is the intersection of OW and the domain. l is the length of WO, l_2 is the length of BO. L_{domain} is the length of the domain. Here $L_{domain} = 0.05\text{m}$.

When $0.5\text{s} < \text{t} \leq 0.55\text{s}$, the center of the vortex start leaving the domain. The decomposition of velocity is shown in Figure 5.1. The $U_{maxTotal}$ can be obtained by

$$U_{maxTotal} = \sqrt{\left(U_{coflow} + U_x\right)^2 + U_y^2}$$

(5.3)

Where:

$U_x = U_{max}\cos(\alpha)$

$U_y = U_{max}\sin(\alpha)$

$R = \sqrt{r_c^2(t=0) + 2vt}$

$$\sin(\alpha) = \frac{l_2}{R} = \frac{U_{coflow}t - \frac{L_{domain}}{2}}{R} = \frac{U_{coflow}t - \frac{L_{domain}}{2}}{\sqrt{(U_{coflow} + U_x)^2 + U_y^2}}$$
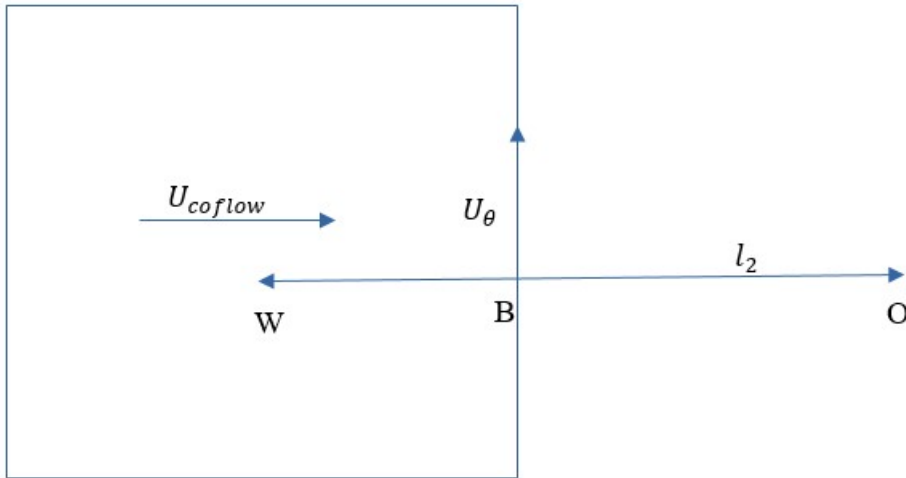
Figure 5.2 Analysis of the velocity field when the vortex completely leaves the domain. U_θ is the tangential velocity of the vortex. W is the centre of the domain. B is the intersection of the domain and OW. l_2 is the length of BO.

When $0.55s < t \leq 2s$, the location where the vortex has the maximum velocity is completely out of the domain, as shown in Figure 5.2. And we have the maximum velocity of the domain:

$$U_{maxTotal} = \sqrt{U_{coflow}^2 + U_\theta^2} \tag{5.4}$$

Where $U_\theta = U_{max}(\frac{l_2}{R})\exp(\frac{R^2 - l_2^2}{2R^2})$

Pressure Field:

We use the minimum pressure p_{min} as the diagnostic to evaluate the boundary condition's influence on the pressure field. The analytical solution of p_{min} for the Lamb-Oseen vortex is:

$$p_{min}(r) = p_\infty - \frac{\rho U_{\theta,max}^2}{2} \exp\left(1 - \frac{r^2}{R^2}\right) \tag{5.5}$$

Where r is the distance between the point and the vortex centre.

When $0 < t \leq 0.5s$, the minimum pressure can be obtained at the centre of the vortex:

$$p_{min} = p_\infty - \frac{\rho U_{\theta,max}^2}{2} \exp(1) \tag{5.6}$$

where $U_{\theta,max} = U_{max} = U_{max}(t = 0)\left(\frac{2vt}{r_c^2(t = 0)} + 1\right)^{-3/2}$
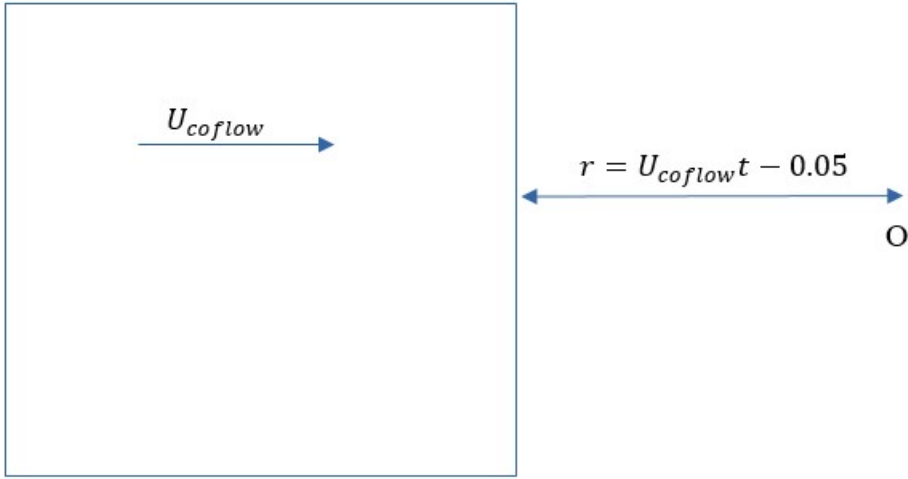
Figure 5.3 Distance between vortex centre and domain.

After the vortex center leaves the domain, $r = U_{coflow}t - 0.05$, as shown in Figure 5.3.

When $0.5s < t \leq 0.55s$:

$$p_{min} = p_\infty - \frac{\rho U_{\theta,max}^2}{2}\exp\left(1 - \frac{r^2}{R^2}\right) = p_\infty - \frac{\rho U_{\theta,max}^2}{2}\exp\left(1 - \frac{\left(U_{coflow}t - 0.05\right)^2}{R^2}\right) \qquad (5.7)$$

where $U_{\theta,max} = U_{max} = U_{max}(t=0)\left(\frac{2vt}{r_c^2(t=0)} + 1\right)^{-3/2}$

When $0.55s < t \leq 2s$:

$$p_{min} = p_\infty - \frac{\rho U_{\theta,max}^2}{2}\exp\left(1 - \frac{r^2}{R^2}\right) = p_\infty - \frac{\rho U_{\theta,max}^2}{2}\exp\left(1 - \frac{\left(U_{coflow}t - 0.05\right)^2}{R^2}\right) \qquad (5.8)$$

where $U_{\theta,max} = U_{max}(\frac{l_2}{R})\exp(\frac{R^2-l_2^2}{2R^2})$
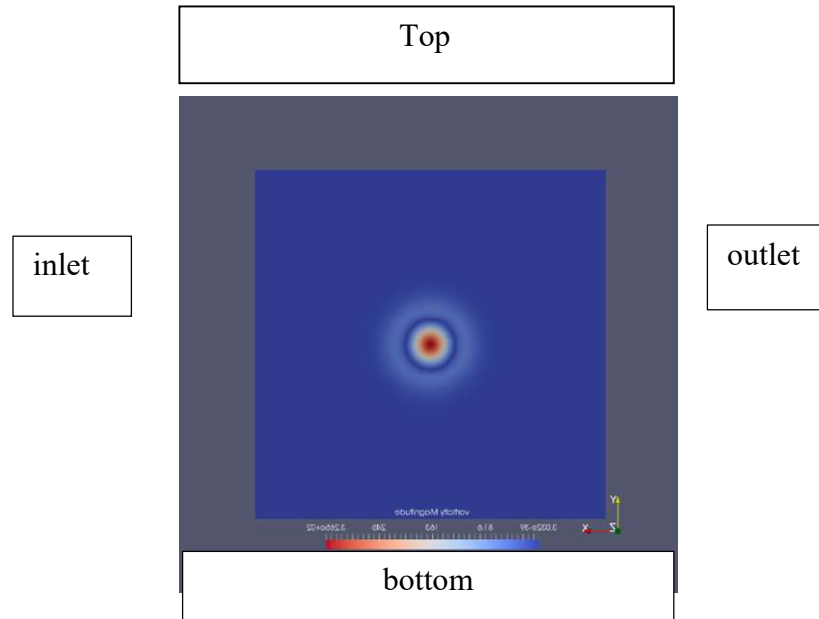
49

Results:

Figure 5.4 Boundary names in this simulation

We tried several combination of standard BCs which are commonly used in FireFOAM.

Case1:

Table (5.1) Combination of BCs used in case1

	inlet	outlet	Top and bottom
U	fixedValue	zeroGradient	zeroGradient
p_{rgh}	fixedFluxPressure	fixedValue	zeroGradient

Here the zeroGradient BC is the one which applies a zero-gradient condition from the patch field to patch faces. The fixedValue BC supplies a fixed value constraint to the patch and is the base class for a number of other BC. The fixedFluxPressure is a common BC used on the pressure field. It sets the pressure gradient to the provided value such that the flux on the boundary is that specified by the velocity boundary condition.
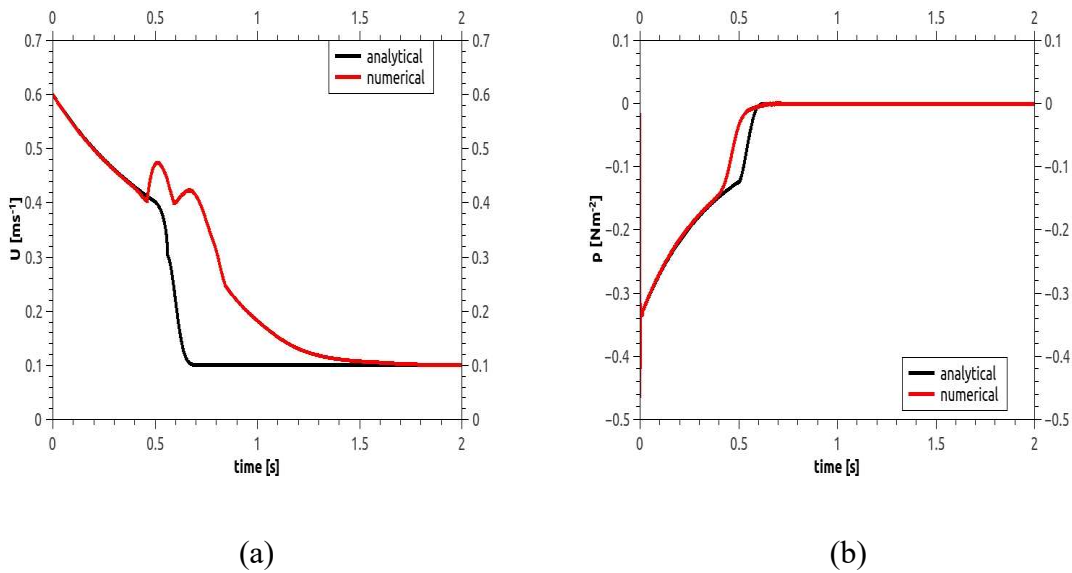

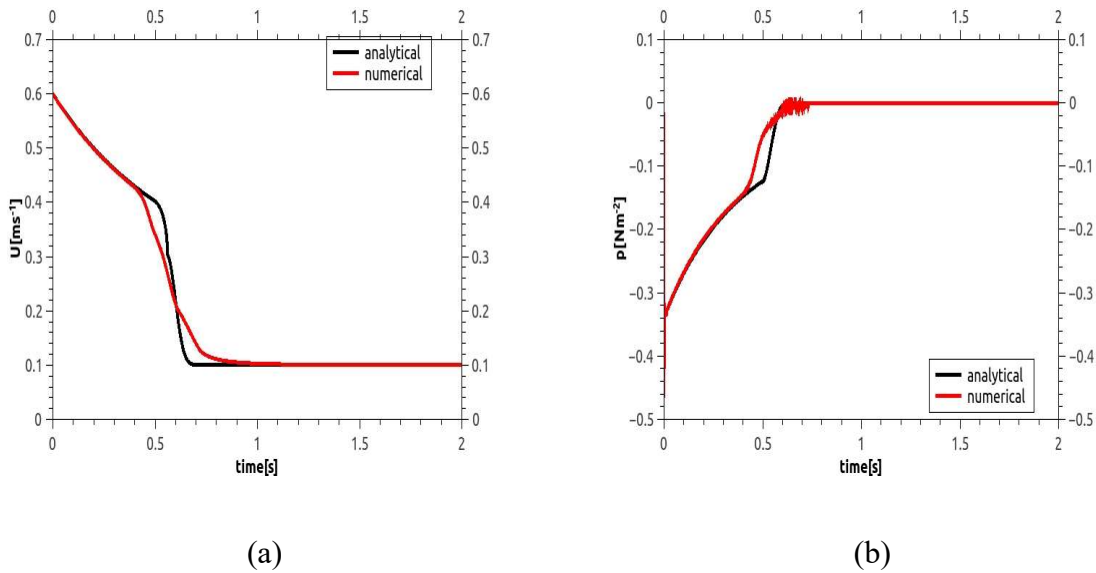(a)   (b)

Figure 5. 5 Case1. a) maximum velocity b) minimum pressure. The red line is the result obtained from FireFOAM. The black line is the analytical solution.

As we can see from Figure 5.5, the error caused by this combination of BCs is large, especially in the velocity field. There are two unphysical oscillation in the solution of velocity which we need to avoid.

Case2

51

Table (5.2) Combination of BCs used in case2

	inlet	Outlet	Top and bottom
U	fixedValue	inletOutlet	zeroGradient
p_{rgh}	fixedFluxPressure	fixedValue	zeroGradient

Compared with case1, the change in case2 is that the outlet BC of the velocity field becomes

inletOutlet. This boundary condition provides a generic outflow condition, with specified inflow

for the case of return flow. When there is no back flow, this BC is the same as the zeroGradient.

(a)   (b)

Figure 5.6 Case2. a) maximum velocity b) minimum pressure. The red line is the result obtained

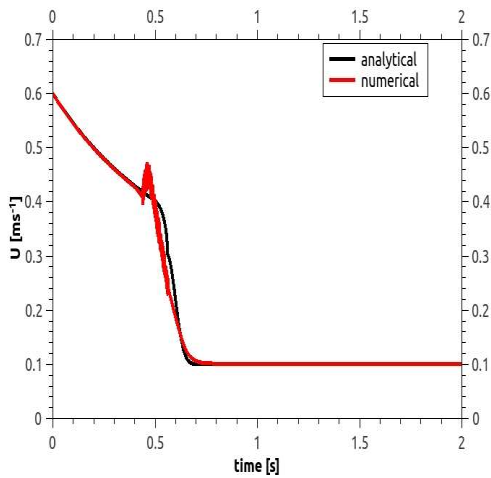from FireFOAM. The black line is the analytical solution.

From Figure 5.6 we can see that the error in the velocity field is significantly reduced. However, there are oscillations in the pressure field at the end of the vortex's leaving.
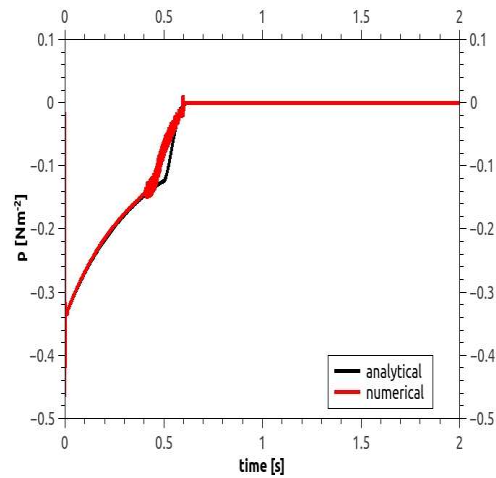
Case3:

Table (5. 3) Combination of BCs used in case3

	Inlet	Outlet	Top and bottom
U	fixedValue	zeroGradient	zeroGradient
p_{rgh}	fixedFluxPressure	totalPressure	zeroGradient

Compared with case1, the change in case3 is that it uses totalPressure BC on the outlet, which provides a fixed value for total pressure instead of a fixed value p_rgh. Both of these are good ways to prevent pressure drifting in the simulation.

(a)   (b)

Figure 5.7 Case3. a) maximum velocity b) minimum pressure. The red line is the result

obtained from FireFOAM. The black line is the analytical solution.

From Figure 5.7 we can see, that although the discrepancy between the numerical and analytical

solution is small in most places, it causes oscillations in result, both in velocity and pressure. In

addition, it caused an unphysical jump in the velocity field. This is what we want to avoid. So
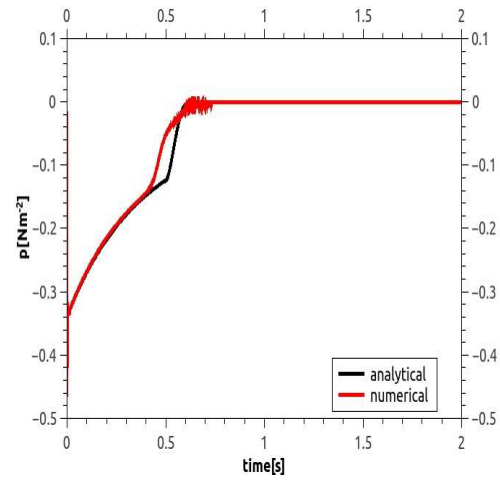
this combination is not satisfactory.

Case4:

Table (5.4) Combination of BCs used in case4

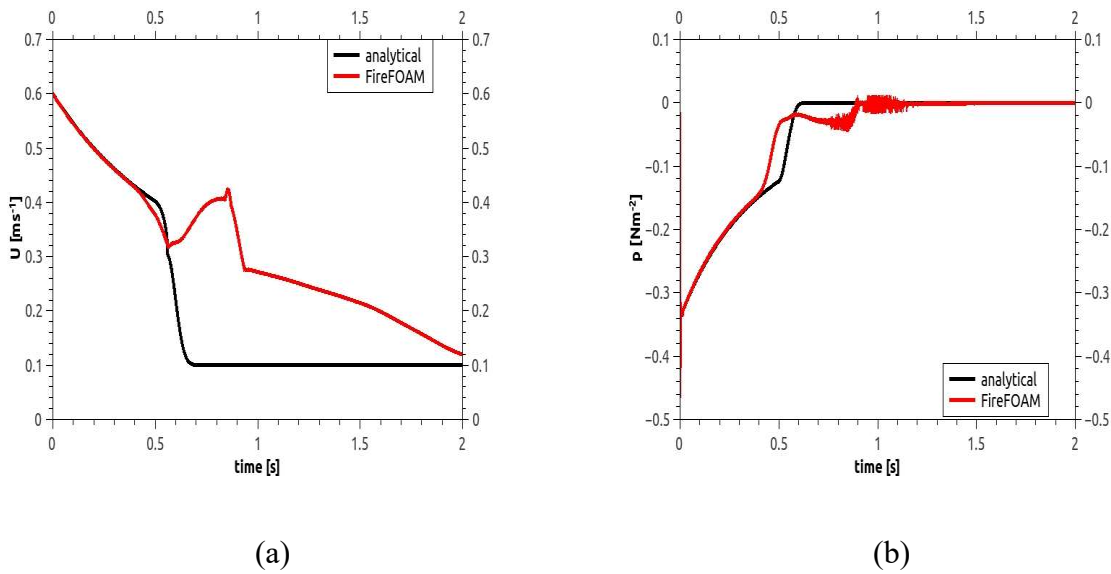|	Inlet	Outlet	Top and bottom
U	fixedValue	inletOutlet	zeroGradient
p_{rgh}	fixedFluxPressure	totalPressure	zeroGradient
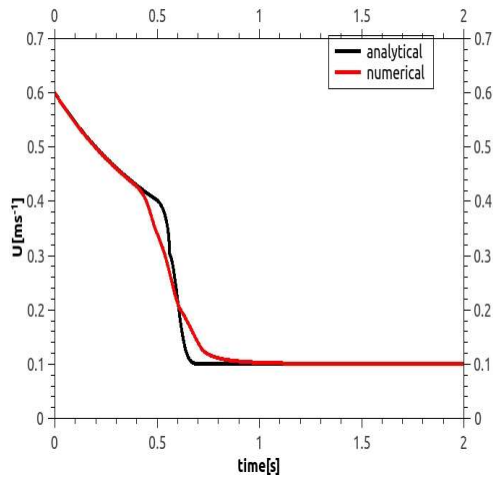
(a) (b)

Figure 5.8 Case4. a) maximum velocity b) minimum pressure. The red line is the result obtained from FireFOAM. The black line is the analytical solution.

From Figure 5.8, we can see this combination of BCs successfully eliminates the oscillation in the velocity field and it has no unphysical jump; it also the limits discrepancy between the numerical and analytical solution. However, the oscillation in pressure field still exists.

Case5:

Table (5.5) Combination of BCs used in case5

	Inlet	outlet	Top and bottom
U	fixedValue	pressureInletOutlet Velocity	zeroGradient

p_{rgh}	fixedFluxPressure	totalPressure	zeroGradient

Here the BC of outlet, i.e. pressureInletOutletVelocity, is different from all the cases shown above. This velocity inlet/outlet boundary condition is applied to patches where the pressure is specified. A zero-gradient condition is applied for outflow (as defined by flux); for inflow, the velocity is obtained from the patch-face normal component of the internal-cell.

(a)   (b)

Figure 5.9 Case5. a) maximum velocity b) minimum pressure. The red line is the result obtained from FireFOAM. The black line is the analytical solution.

From Figure (5.9) we can see that this combination of BCs is worse than the cases above. The error is large both in velocity and pressure field. The oscillation in the pressure field is not restricted.
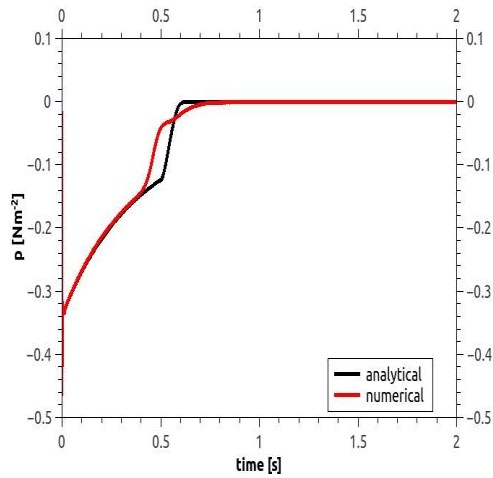
Case6:

Table (5.6) Combination of BCs used in case6

	Inlet	outlet	Top and bottom
U	fixedValue	pressureInletOutlet Velocity	zeroGradient
p_{rgh}	fixedFluxPressure	totalPressure	zeroGradient
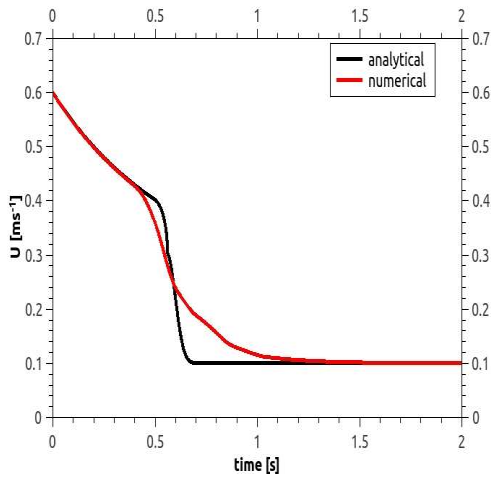
(a)   (b)

Figure 5.10 Case6. a) maximum velocity b) minimum pressure. The red line is the result obtained from FireFOAM. The black line is the analytical solution.

From the Figure 5.10, it seems that the combination of BCs in case5 is the optimal BC combination: there is no oscillation in velocity and pressure. However, the result is still not

satisfying, there is still a significant discrepancy between the analytical solution and the result

obtained from fireFoam. We found that these errors in the results, both the oscillation and big

gap, are caused by the appearance of back flow in this case:

The research on backflow:

To study the influence of backflow, we used two combinations of BCs shown above, one is the

combination that we find optimal and the other is the one which cause oscillations in velocity

and pressure which we used in case3. We only change the co-flow velocity in each case. We set

it to $0.05m/s$ and $0.5m/s$ respectively. Note that when $U_{coflow} = 0.5m/s$, there is no back

flow. Here are the results:

Case A:

In this case, the combination of BCs is what we used in case3, i.e. Table (5.3)

$U_{coflow} = 0.05m/s$
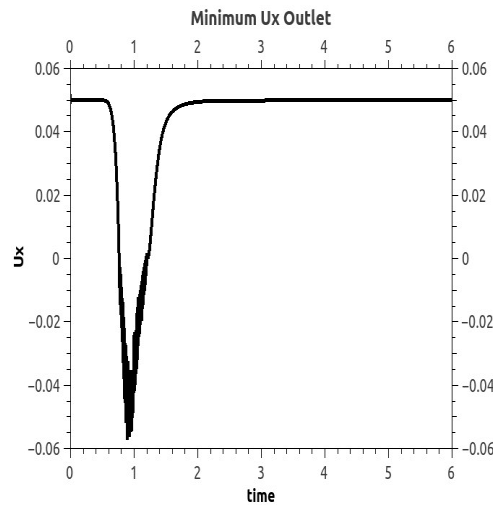
Figure 5.11 The minimum U_x at the outlet (with sign). The negative value means that there exists back flow at the outlet.
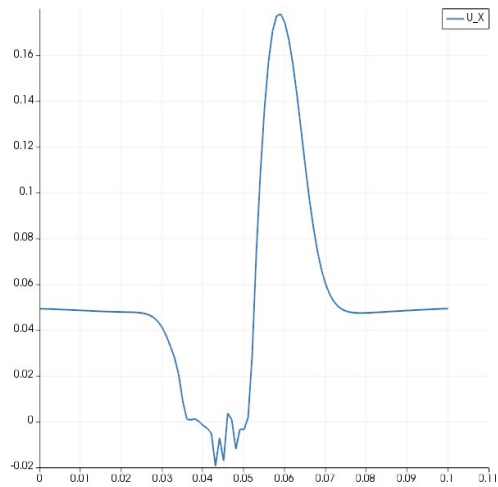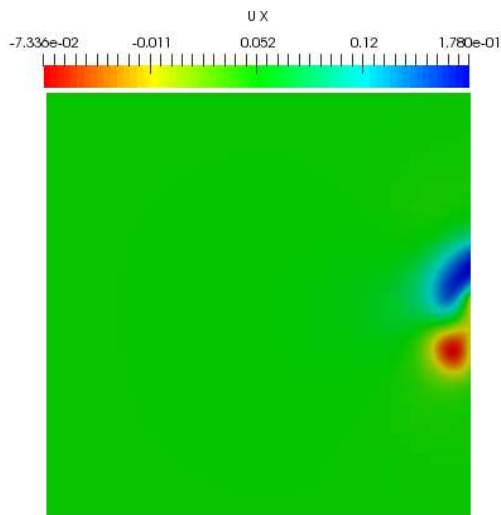
From figure 5.11, we can see the existence of the backflow. The boundary conditions performs badly which cause the oscillation in velocity. We can also see that the oscillation appears with the appearance of backflow. Next we explore the profile of U_x to bring more information on the oscillation.

(ai)

(aii)

(bi)   (bii)
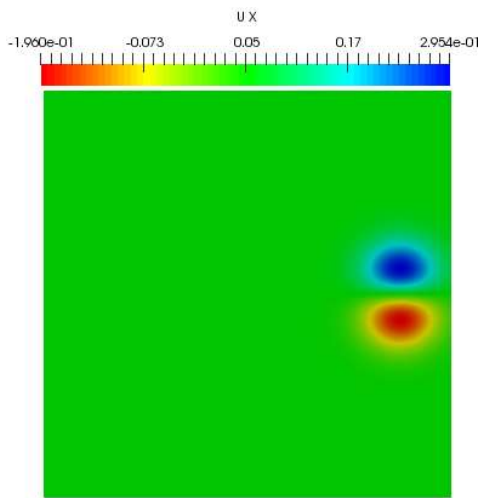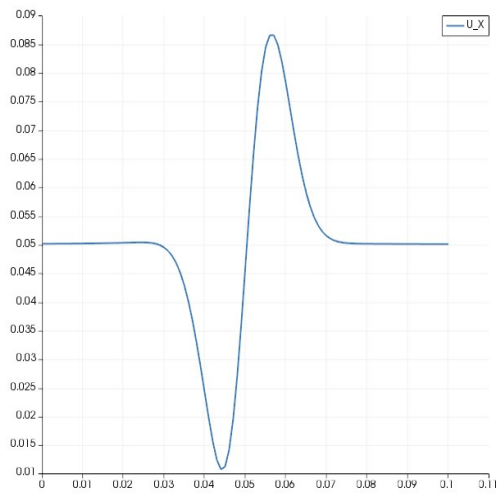
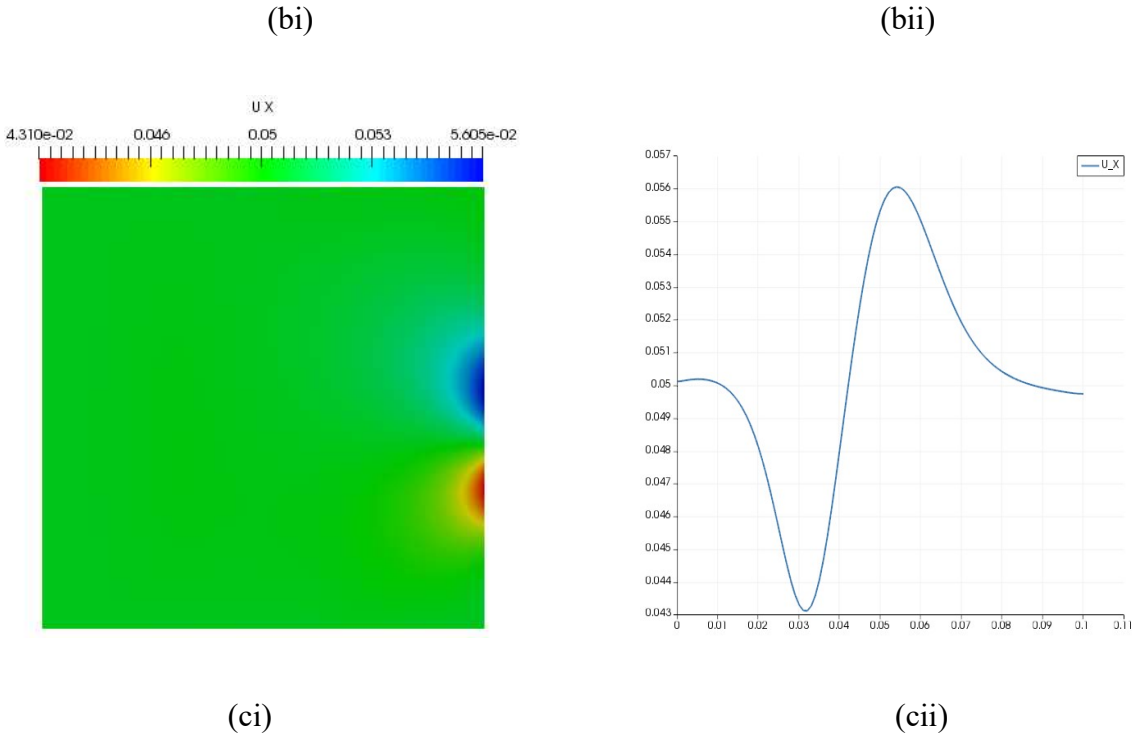(ci)   (cii)

Figure 5.12 Velocity at the outlet when the vortex leaves the domain at different times a) $t =$
$0.8s$ b) $t = 1.0s$ c) $t = 1.2s$ i) the screenshot of the velocity field ii) the profile of steamwise
velocity at the outlet.

From the Figure 5.12, we can find that the oscillation occurs when the vortex transport out of the
domain. As shown in 5.12 (b), the oscillation only happens where the $U_x < 0$, i.e. the position
where the back flow happens. Before and after the vortex leaves the domain, there is no
oscillation in velocity.

<div align="center">(a)   (b)</div>
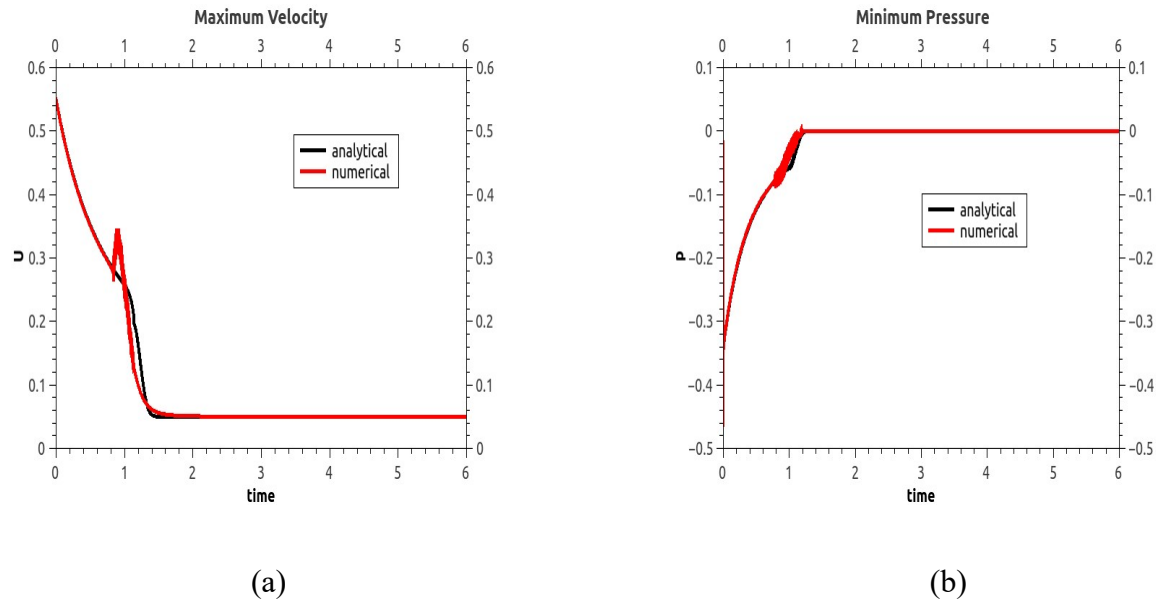
Figure 5.13 CaseA. a) maximum velocity b) minimum pressure. The red line is the result obtained from FireFOAM. The black line is the analytical solution.

Figure 5.13 shows the maximum velocity and minimum pressure of the domain when $U_{coflow} = 0.05m/s$. Similar to the case when $U_{coflow} = 0.1m/s$, there are oscillations in velocity and pressure field.

In the next case, we use the same boundary condition but increase the velocity of co-flow to $U_{coflow} = 0.5m/s$, in which case there is no back flow.
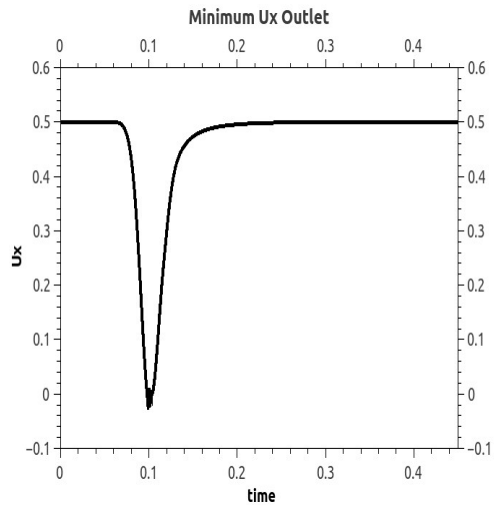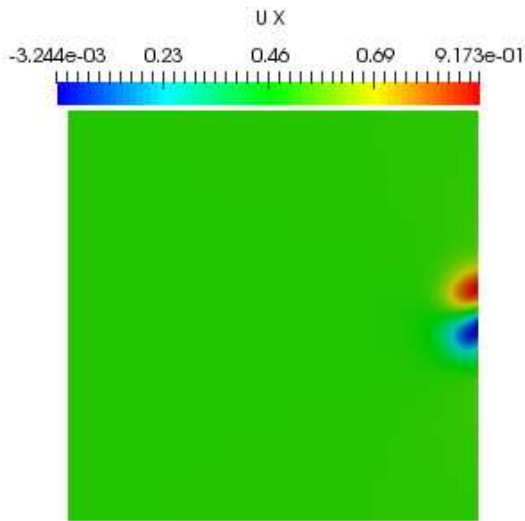
Case B:

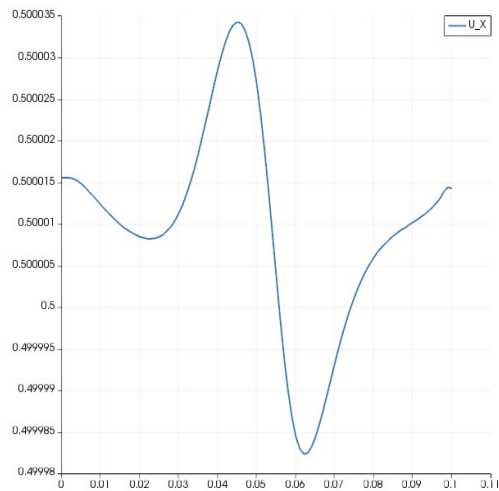Figure 5.14 The minimum U_x at the outlet (with sign). No negative value means there is no
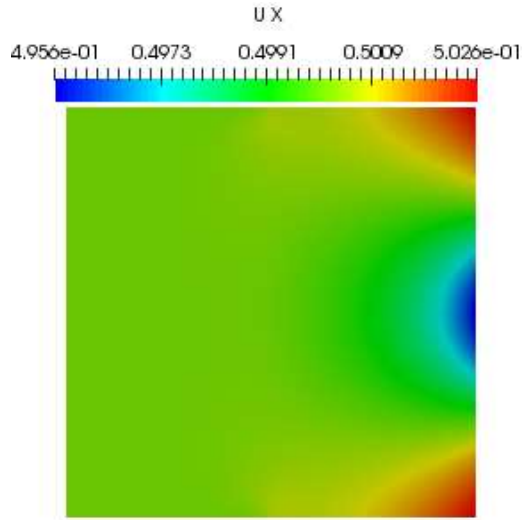
back flow in this case.

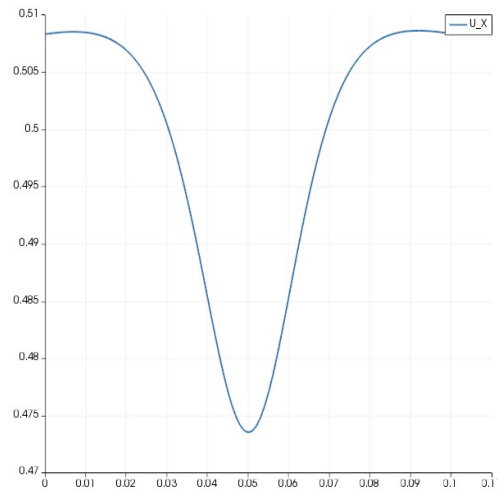(ai)   (aii)

(bi)   (bii)

Figure 5.15, Velocity at the outlet when vortex leaves the domain at different times a) $t = 0.1s$

b) $t = 0.15s$ i) the screenshot of the velocity field ii) the profile of velocity at the outlet.

From Figure 5.15, we can see the velocity develops smoothly, there is no oscillation in this case.

Compared this result with the former case, we can find see that the oscillation is caused by the

back flow. When the back flow happens, the combination that we use cannot restrict the

oscillation

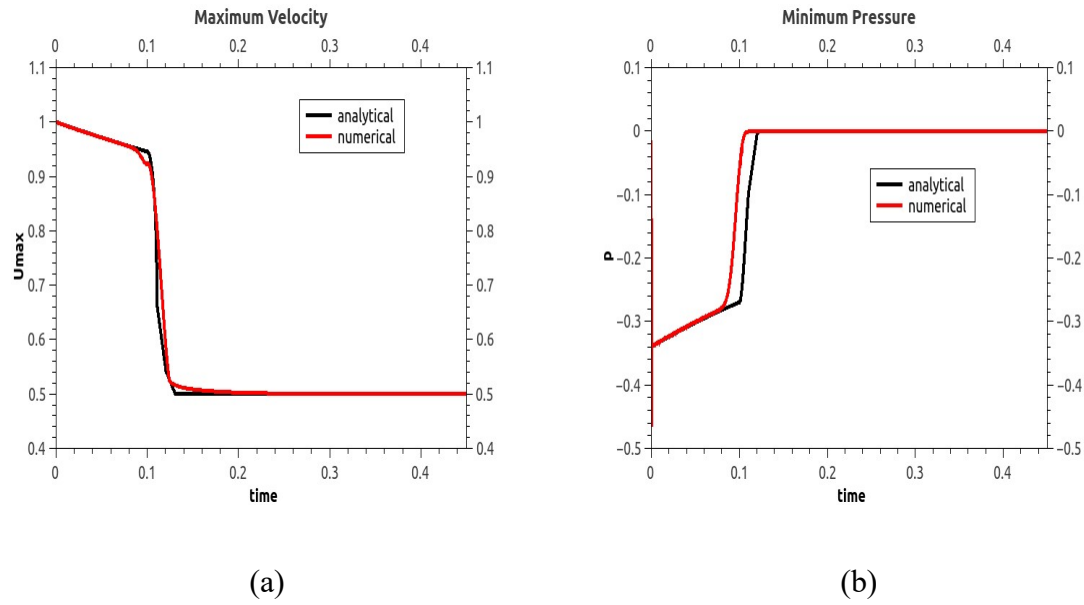(a)   (b)

Figure 5.16 CaseB. a) maximum velocity b) minimum pressure. The red line is the result obtained from FireFOAM. The black line is the analytical solution.

Figure 5.16 shows the maximum of velocity and minimum pressure in this domain. We can see that when there is no back flow, the numerical solution and analytical solution can agree. There is no oscillation and no large discrepancy between the numerical and analytical solution.

In the next two case, we use the same set of co-flow velocity to the combination of BCs which we find optimal above, i.e. Table(5.6).
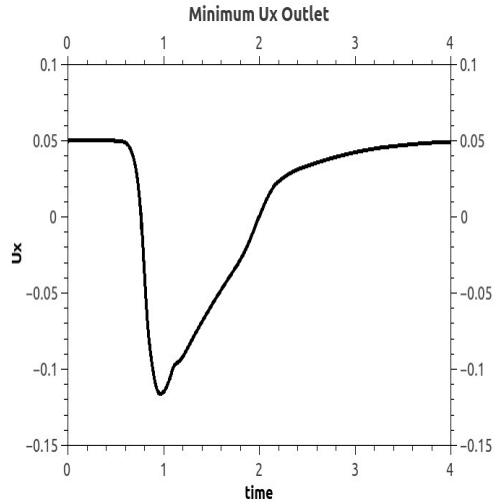
Case C:

$$U_{coflow} = 0.05m/s$$

Figure 5.17 The minimum U_x at the outlet (with sign). The negative value means there exists back flow at the outlet.

Figure 5.17 shows the minimum of U_x at the outlet. We can this combination of BCs restrict the oscillation successfully, but there is highly asymmetric part in the velocity. This combination of BCs break the symmetry of the vortex structure.
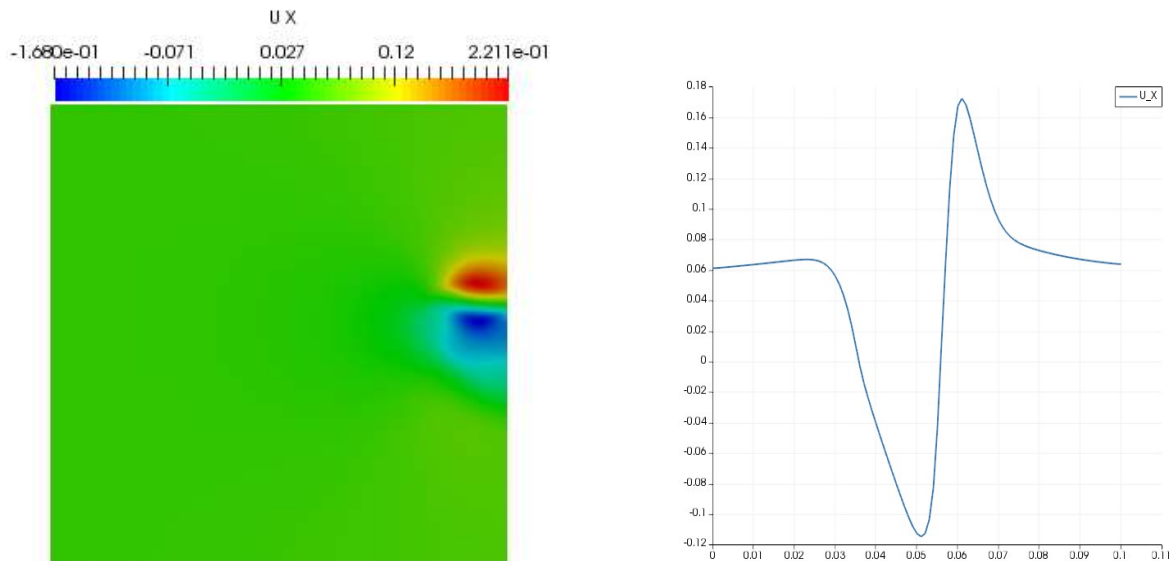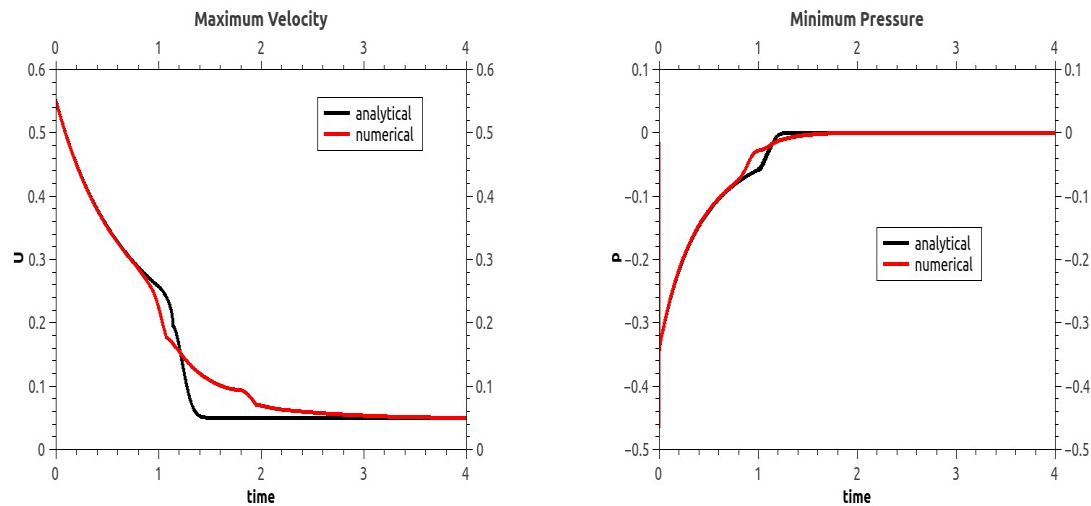
Figure 5.18 Velocity at the outlet when the vortex leaves the domain at $t = 0.1s$ a) the

screenshot of the velocity field b) the profile of velocity at the outlet.

Figure 5.18 shows the velocity field and the profile of U_x at the outlet when the vortex is leaving

the domain. We can see the existence of back flow and the highly asymmetric part in the velocity

field.

(a)   (b)

Figure 5.19 Case C. a) maximum velocity b) minimum pressure. The red line is the result

obtained from FireFOAM. The black line is the analytical solution.

Figure 5.19 shows the maximum velocity and minimum pressure of case C. We can see that

there is no oscillation in the pressure field which is the same as the case 6. But the error is

significant in the velocity and pressure fields.
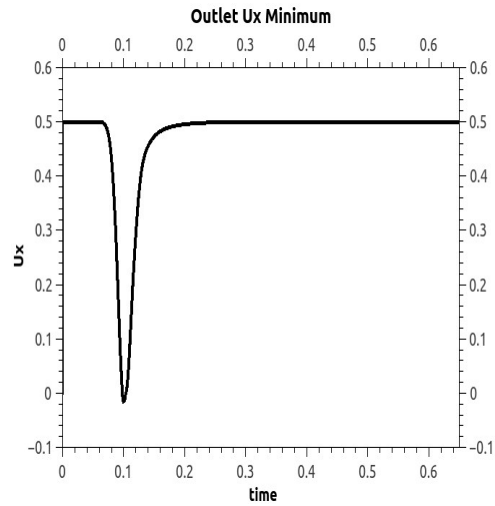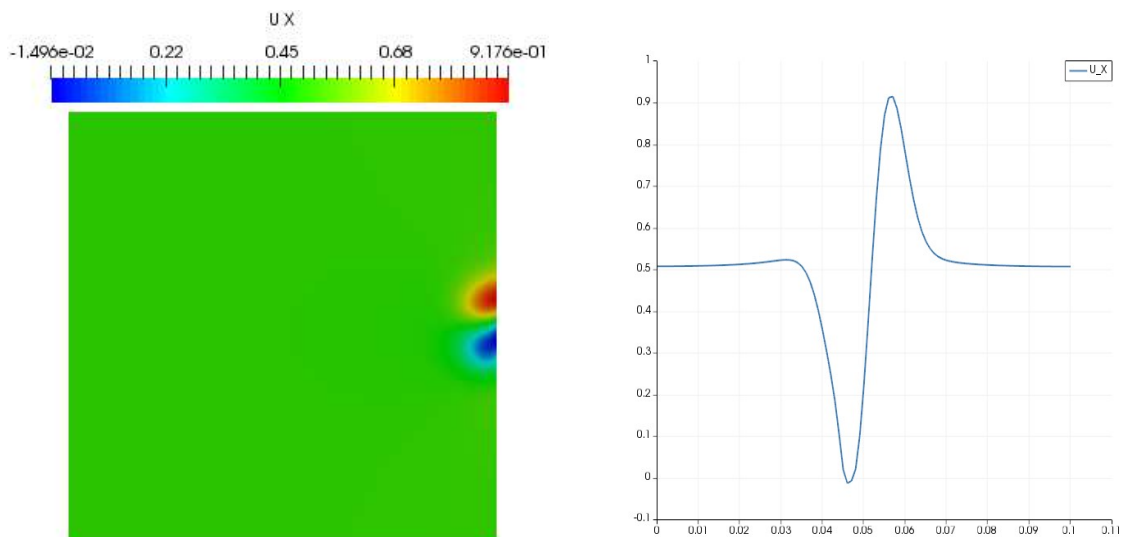
Case D:

$$U_{coflow} = 0.5 m/s$$

Figure 5.20 The minimum U_x at the outlet (with sign). No negative value means there is no back

flow in this case.

Figure 5.21 Velocity at the outlet when the vortex leaves the domain at $t = 0.1s$ a) the

screenshot of the velocity field b) the profile of velocity at the outlet.

From Figure 5.21 we can find, the structure of vortex is much more symmetric compared the

case with back flow. Without the influence of backflow, this combination of BCs can maintain

the symmetry of the vortex when it is leaving the domain.
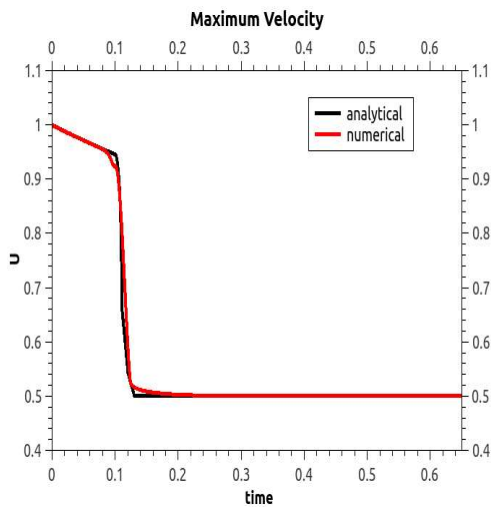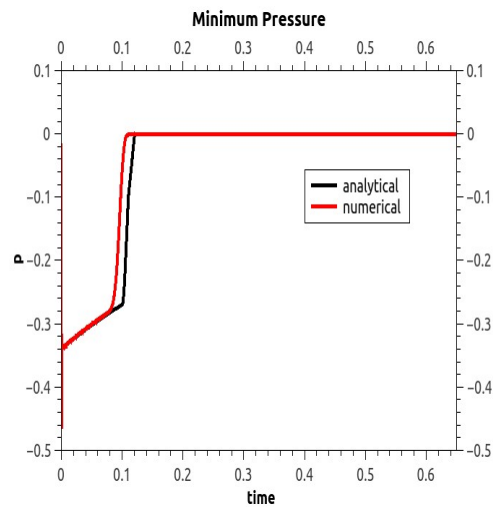
(a)   (b)

Figure 5.22 Case D. a) maximum velocity b) minimum pressure. The red line is the result

obtained from FireFOAM. The black line is the analytical solution.

Figure 5.22 shows the maximum velocity and minimum pressure of the field obtained from case

D. As we can see, the discrepancy between the analytical solution and numerical solution is

much smaller compared to the case in which there exists backflow which indicates that the error comes mainly from the lack of ability of the boundary condition to handle backflow.

Buoyant hot bubble:

In the case above, we have tested the impact of the different choices of BC could cause different influence to velocity field. In this case, we test this impact on the temperature field.

Case 1 of hot bubble:

We set the temperature field as presented in Eq. (5.9):

$$T(x,y) = 294 + 50(\tanh(\frac{(x-x_0)}{\sigma}) - \tanh(\frac{(x-x_1)}{\sigma})(\tanh(\frac{(y-y_0)}{\sigma})$$
$$- \tanh(\frac{(y-y_1)}{\sigma})))$$

(5.9)

where: $x_0 = -0.008$, $y_0 = -0.008$, $x_1 = 0.008$, $y_1 = 0.008$, $\sigma = 0.0015$.

Figure 5.23 shows the initial distribution of temperature field for a hot bubble. The size of hot spot is $0.02m$. The domain of our case is $0.2m \times 0.15m$
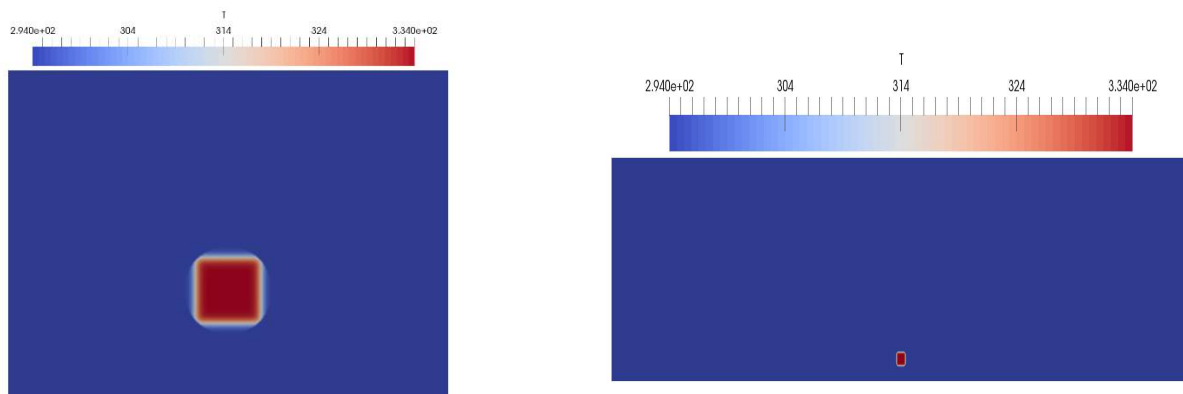
(a)   (b)

Figure 5.23 Initial temperature field for a "hot bubble". a) case with BC's influence b) reference

case without BC's influence
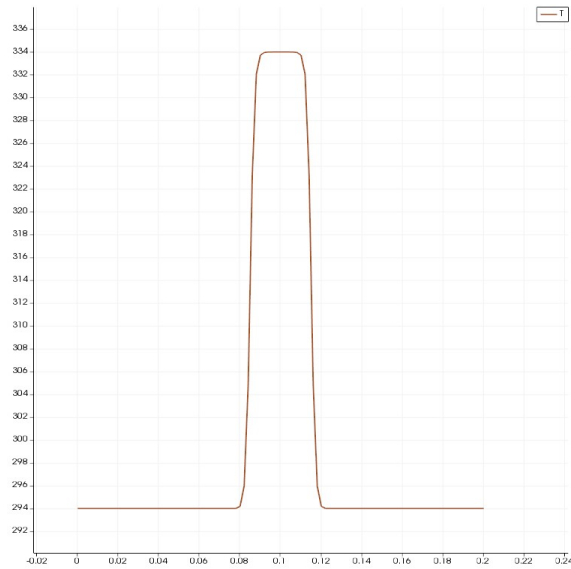
Figure 5.24 Initial temperature over bubble center line.

Since there is no analytical solution to this case, we construct a domain which is large enough

size($2m \times 0.5m$) that can then be used as a reference case.

(a)   (b)

(c)
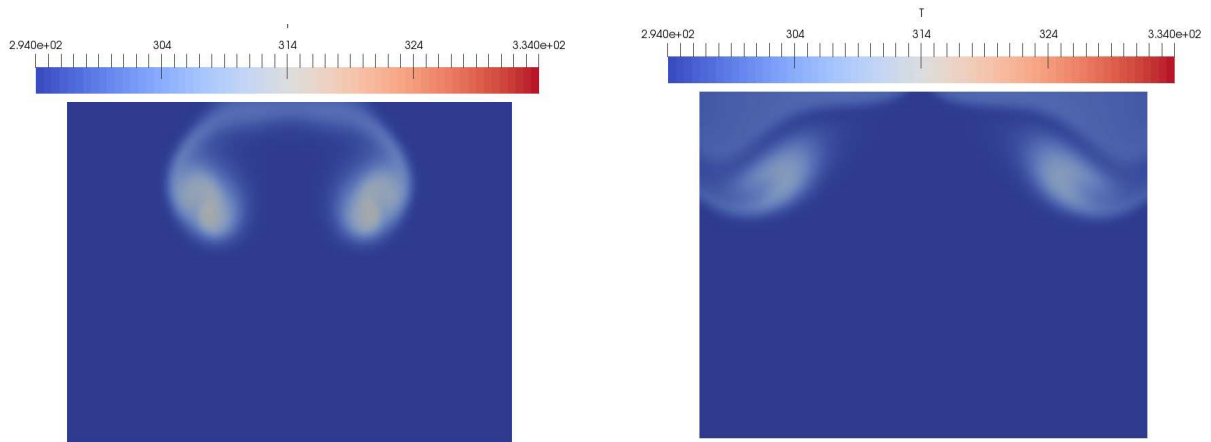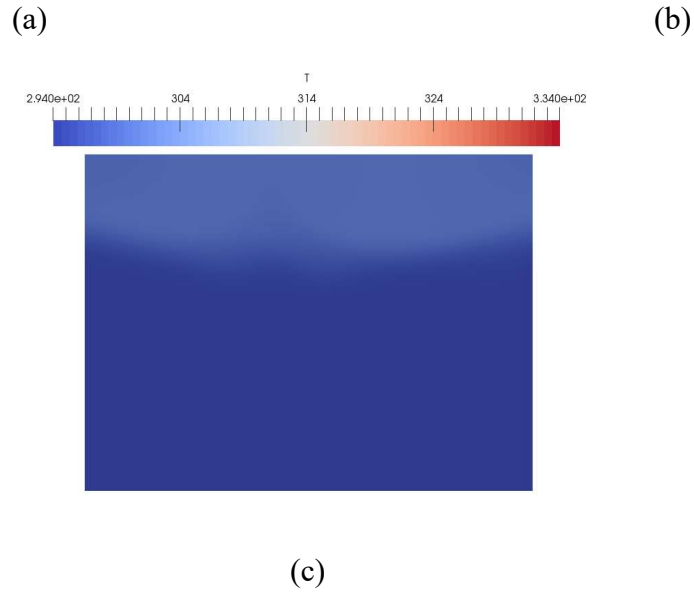
Figure 5. 25 Development of hot bubble (a) $t = 1s$ (b) $t = 1.5s$ (c) $t = 5s$

(a)   (b)
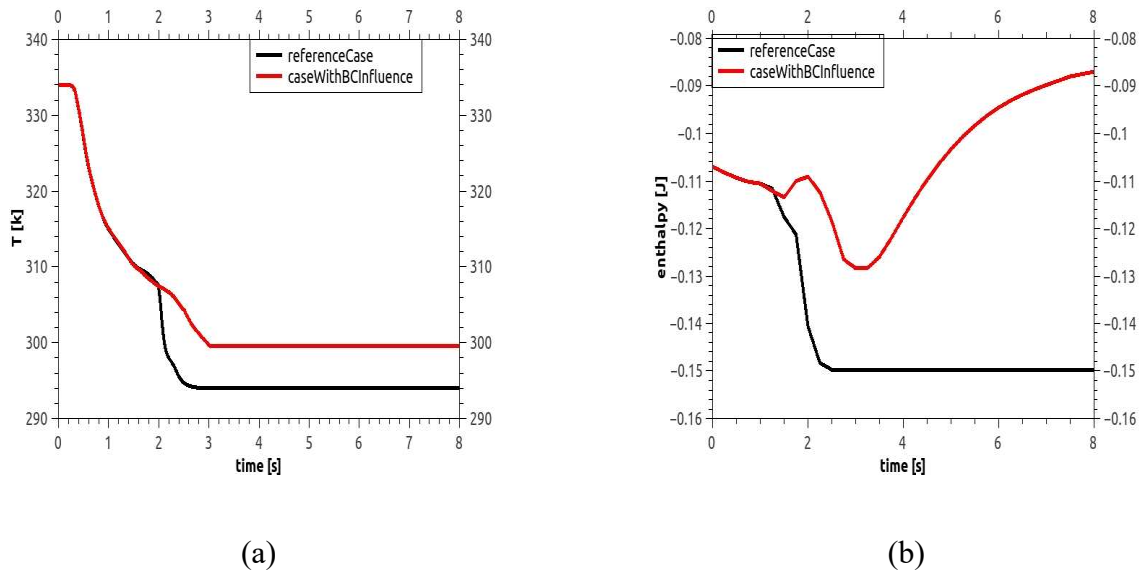
Figure 5.26 Case 1 of hot bubble: a) maximum temperature b) enthalpy integration over the

volume.

From Figure 5.25 and Figure 5.26, we can see that this combination of boundary conditions does not allow the hot bubble to leave the computational domain. The boundary on the top prevents some heat and prevent some heat leaving the domain. Some residual heat remains in this domain after the hot bubble leaves. Some heat even flows back into the domain.

This is because the boundary on the top is not set properly. In this case, the boundary on the top is set to keep the pressure from drifting. However, when the hot bubble leaves the domain, the temperature change will lead to a change in the pressure field, which makes the fixed value of pressure unphysical. One way to fix this problem is to fixed the pressure on the bottom boundary instead of the top boundary, since there is no request for the fixed value of velocity at the bottom, such combination of boundary condition is reasonable, as shown in Table (5.7).

Case 2 of hot bubble:

Table (5.7) Combination of BCs used in case 2 of hot bubble

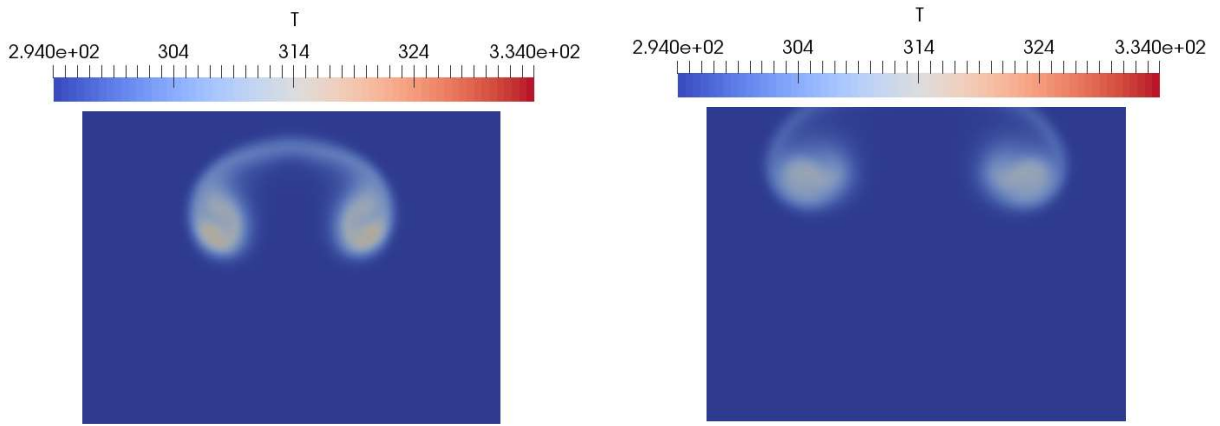|	Inlet	outlet	Top and bottom
U	zeroGradient	zeroGradient	zeroGradient
p_{rgh}	totalPressure	zeroGradient	zeroGradient
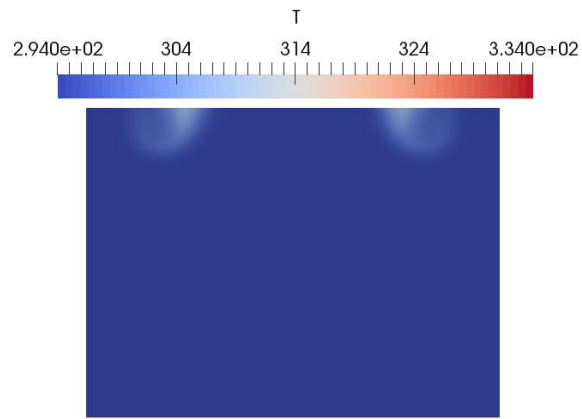
Figure 5.27 Development of hot bubble (a) $t = 1s$ (b) $t = 1.5s$ (c) $t = 2s$
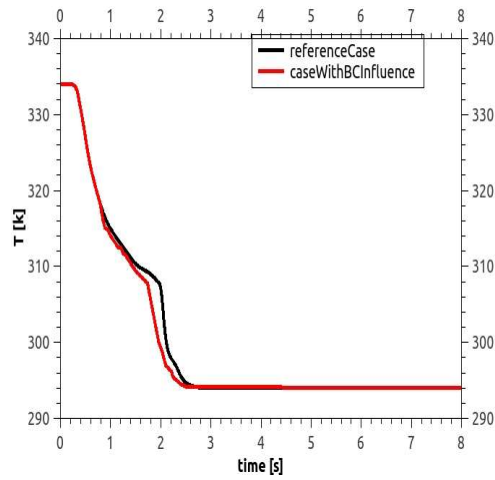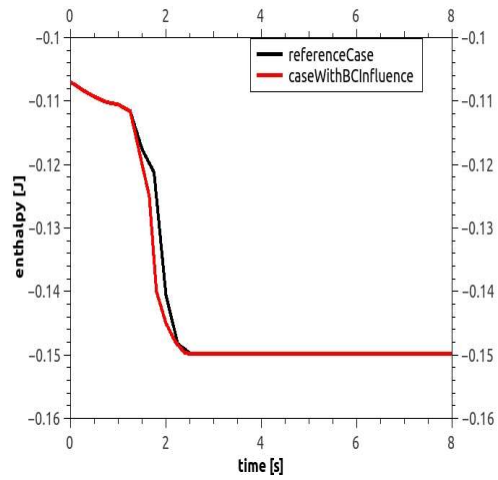
(a)   (b)

Figure 5.28 Case 2 of hot bubble: a) maximum temperature b) enthalpy integration over the

volume.

From Figure 5.27 and Figure 5.28, we can see this combination of boundary condition can make

the hot bubble leave smoothly. The gap between the numerical solution and the reference

solution is small and there is no heat residual that remains in the domain. So this combination of

BCs at the outlet is considered good for the hot bubble buoyancy case.

Chapter 6: McCaffrey's Pool Fire

The aim of this case:

This is a validation case for FireFOAM combustion modeling capability using McCaffrey's pool fire configuration, which is an important classical problem in fire science. Many theories and correlations have been proposed for different aspects of this case [18]; [30]; [31]. These results have been widely used in fire science and engineering application. In addition, the geometry of the flame and fire plume is very simple. So we choose this case as the validation case for fire modeling of FireFOAM.

Case initialization:

The simulation is methane fire on a $0.3m \times 0.3m$ square burner with heat release rate (HRR) at 51.5 kW. In our case, the burner is set at an elevation of 0.1m, which can decrease the influence of floor. Table (6.1) lists the heat release rate (HRR) value of the case. The corresponding non-dimensional heat release rate $\dot{Q}^*$ and characteristic length scale D^* as defined in the following are also listed in Table (6.1):

$$\dot{Q}^* = \frac{\dot{Q}}{\rho_\infty C_p T_\infty \sqrt{g} D^{5/2}} \tag{6.1}$$

$$\dot{D}^* = (\frac{\dot{Q}}{\rho_\infty C_p T_\infty \sqrt{g}})^{2/5} \tag{6.2}$$

where ρ_∞ is the environmental density, C_p is the specific heat under constant pressure, T_∞ is the environmental temperature, g is the acceleration of gravity, D is the length of burner.

The variable $\dot{Q}^*$ is used in many plume correlations as the scaling parameter, and $\dot{D}^*$is often used to evaluate the grid resolution in CFD simulations of fire.

Table (6.1) Case Parameters

	HRR(kW)	$\dot{Q}^*$	$\dot{D}^*$
Case1	51.5	0.6888	0.2918

The computational domain is a $4.8m \times 4.8m \times 4.8m$ cubic box, which is large enough to capture the full plume width and will reduce the influence of the excessive BCs. In addition to blockMesh, another automatic meshing tool called snappyHexMesh is used in this case to refine the mesh. We sets four levels of refinement. The smallest cells are $1.25cm \times 1.25cm \times 1.25cm$, which are uniformly distributed inside a $0.6m \times 1.2m \times 0.6m$ box which includes the burner, such resolution is level-3 refinement in our case. The cell of level-2 refinement is $2.5cm \times 2.5cm \times 2.5cm$ within a $1.2m \times 2.2m \times 1.2m$ box. The level-2 refinement zone is enough to hold the fire plume. The cells in level-1 and level-0 are proportionally increased. The cells of level-0 are $10^3 cm^3$, which are the largest cells of the domain. Such block-structured mesh enables us to concentrate roughly half of the cells into 1.3% volume of the domain.

The BC of the velocity field at the burner corresponds to a fixed mass flow rate. We set the mass flow rate as $1.0323g/s$ which corresponds to the HRR of $51.5kW$. The corresponding BC for the temperature field is the fixed enthalpy flow rate. Convective and diffusive mass and energy fluxes are considered at the inlet. We use the BC which we find optimal for the outflow of heat at

the outlet, i.e. the outlet BC shown in Table (5.7). For the BC at side boundaries, we use typical

entrainment BCs which can fix the pressure. As for the radiation model, we use a simpler

treatment by assuming the radiant fraction as constant equals to 20%, which is the same value as

the value estimated by McCaffrey.

The simulation is run for 20s. Turbulent statistics are collected for the last 12 seconds when the

flow is completely statistically stationary.

Results:

Figure 6.1 presents a screen shot of the temperature field and velocity field after the flame

becomes statistically stationery. $t = 8s$.

(a)   (b)
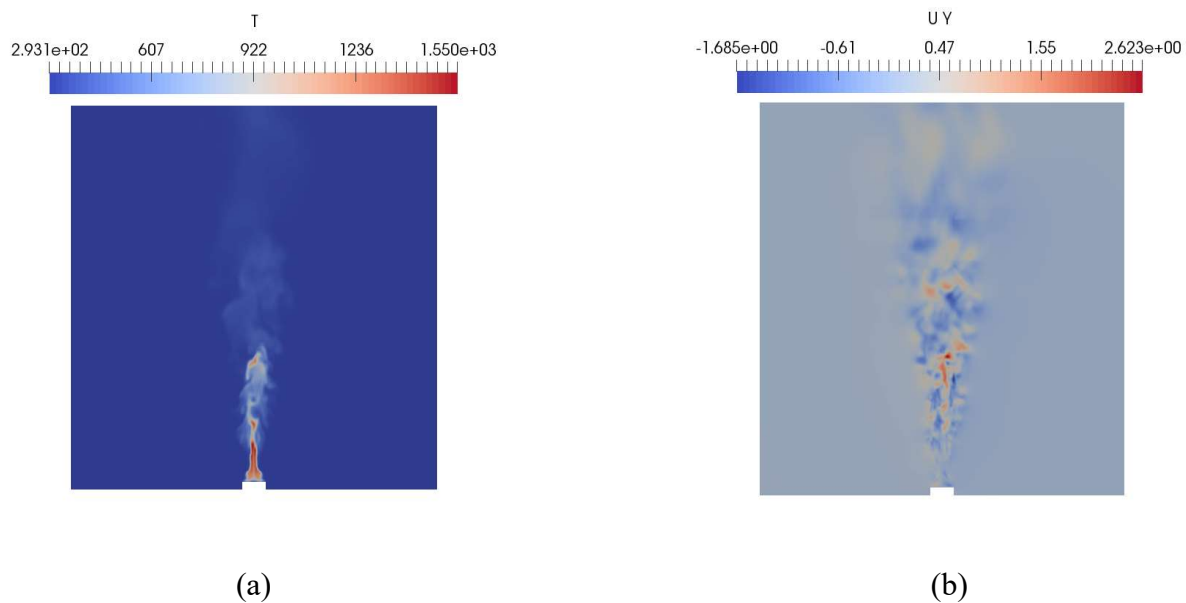
Figure 6.1 Screenshot of field at $t = 8s$. a) Temperature field b) Y component of velocity

Figure 6.2 shows the zoom-in temperature at $t = 8s$. The BC at outlet performs well for the

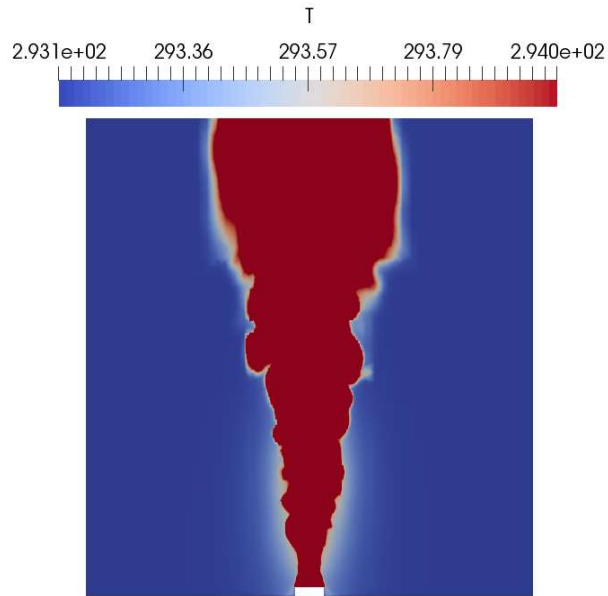outflow of heat. There is no backflow of heat.

Figure 6.2 Zoom-in temperature field at $t = 8s$.

Figure 6.3 shows the HRR with respect to time. HRR presents turbulent characteristics. It has the

oscillation which averages around $51.5 kw$, which is the same as the value we expect.
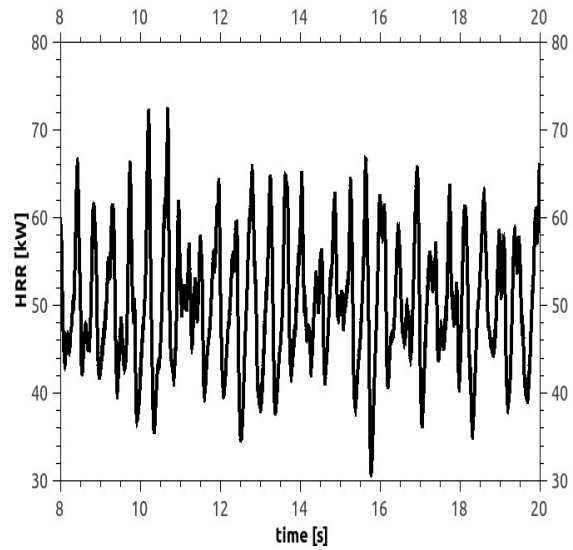
Figure 6.3 Heat release rate with respect to time

Figure 6.4 shows the maximum temperature of the field with respect to time, which corresponds to the flame temperature with 20% radiation.
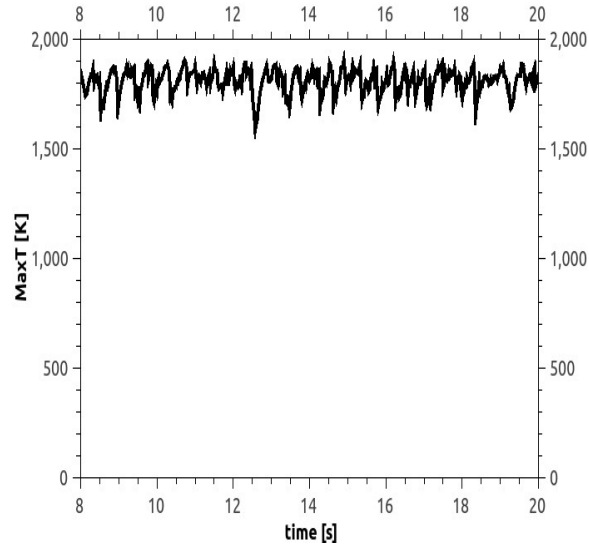

Figure 6.4 Maximum temperature with respect to time

Figure 6.5 shows the minimum temperature of the field with respect to time. Generally the minimum temperature is correct which equals to the environment temperature, i.e. $T = 293k$. There are some unphysical peaks which is caused by numerical error.
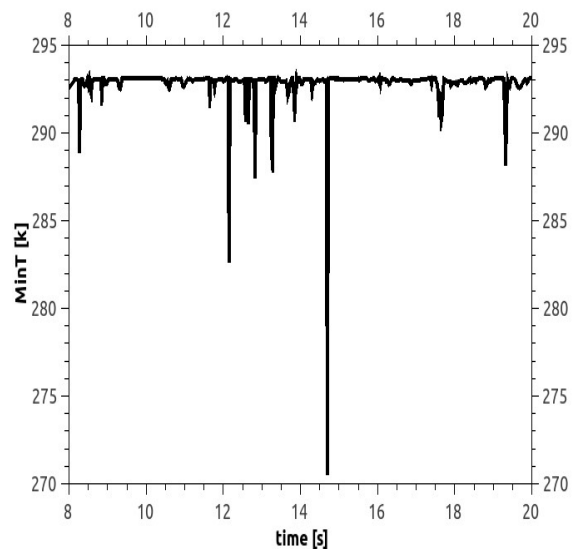
Figure 6.5 Minimum temperature with respect to time

Figure 6.6 presents the mean temperature along the center line. The distance to burner is scaled by $\dot{Q}^{2/5}$, which is the same as the scaling factor proposed by McCaffrey. We get the classic -5/3 slope in the plume zone.

Figure 6.6 Centerline mean temperature rise and decay.

Figure 6.7 presents the mean vertical velocity over center line. It is scaled by $\dot{Q}^{1/5}$. The distance to burner is scaled by $\dot{Q}^{2/5}$. The slope of scaled velocity shows the expected -1/3 slope in the plume zone.

Figure 6.7 Centerline velocity

Chapter 7: Conclusion and furture work:

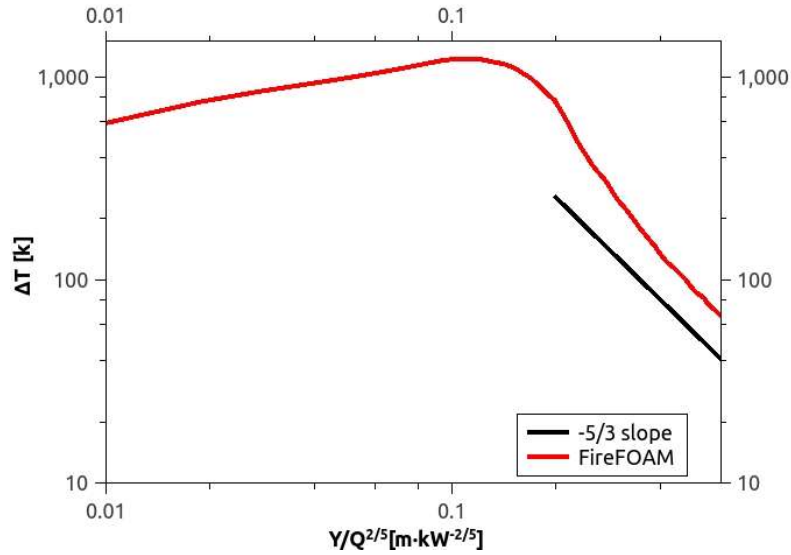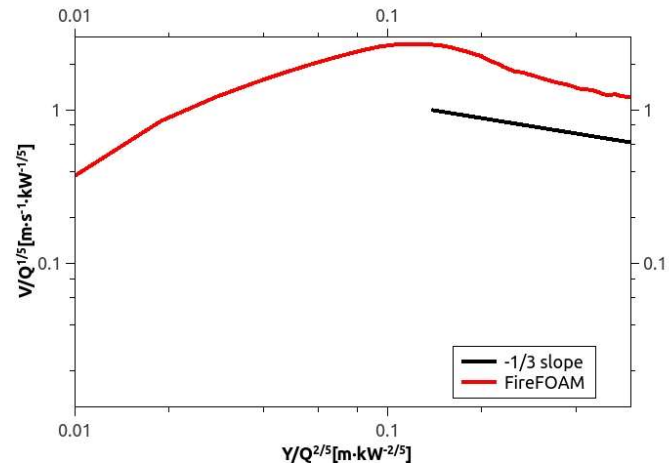This research considers several benchmark cases propelled for the verification and validation of FireFOAM. Several temporal (Euler, backward) and spatial schemes (second order) have been verified through grid convergence through the study of the Lamb-Oseen vortex with co-flow case. The order of accuracy of these schemes in FireFOAM as expected. The conservation of kinetic energy has been verified through the simulation of the Taylor-Green vortex case with different temporal schemes (Euler, backward and Crank-Nicholson). The accuracy of enstrophy growth is also verified in this case. The performance of several SGS turbulence models (Smagrinsky, constant coefficient one-equation, dynamic coefficient one-equation) have been evaluated through the implementation of decaying isotropoic turbulence case. The kinetic energy evolution agrees well with the experimental data of CBC (Comte-Bellot and Corrsin). The dynamic coefficient one-equation model is found to perform best compared to the three models. Several combinations of outlet BCs have been examined through the study of the Lamb-Oseen vortex case and buoyant hot bubble case. The optimal combination of outlet boundary condition is found to be the pressureInletOutletVelocity for the velocity field and the totalPressure for the pressure field. The limitation of current boundary conditions is their treatment of back flow is pointed out. The use of a fixed pressure at outlet is found to be incorrect in the hot bubble case. Some residual heat remain after the hot bubble completely leaves. A change it to zeroGradient can fix this problem. The eddy-dissipation combustion model has been validated through the study of McCaffrey's pool fire case. The classical -5/3 slope for the temperature rise and -1/3 slope for the velocity decay are observed in the plume zone.

There are still several issues in the McCaffrey's case. It is necessary to simulate this case under different HRR to see if the scaling law agrees well with all cases shown in MaCaffrey's original paper. More turbulent properties need to be extracted, such as the normalized velocity fluctuation intensity, to evaluate the combustion model in more detail. In addition to the cases which have been developed, we need to consider several more cases to verify more aspect of FireFOAM, such as the pyrolysis model and the radiation model.

Bibliography

[1] FM Global. FireFOAM. Available from: http://www.fmglobal.com/research-and-resources/research-   and-testing/theoretical-computational-and-experimental-research/open-source-fire-modeling

[2] OpenFOAM. Available from: http://www.openfoam.com/

[3]Jasak, Hrvoje, Aleksandar Jemcov, and Zeljko Tukovic. "OpenFOAM: A C++ library for complex physics   simulations." International workshop on coupled methods in numerical dynamics. Vol. 1000. IUC Dubrovnik, Croatia, 2007.

[4] Direct, C. F. D. "The Architects of OpenFOAM." OpenFOAM User Guide (2015).

[5] OpenFOAM. Wikipedia. Available from: https://en.wikipedia.org/wiki/OpenFOAM.

[6] OpenCFD Ltd. OpenFOAM. Available from: http://www.openfoam.com/.

[7] Jasak, Hrvoje. "OpenFOAM: open source CFD in research and industry." International Journal of Naval Architecture and Ocean Engineering 1.2 (2009): 89-94.

[8] Chen, L. F., et al. "Numerical investigation of wave–structure interaction using OpenFOAM." Ocean Engineering 88 (2014): 91-109.

[9] Ferziger, Joel H., Milovan Peric, and Anthony Leonard. "Computational methods for fluid dynamics." (1997): 80-84.

[10] Issa, Raad I. "Solution of the implicitly discretised fluid flow equations by operator-splitting." Journal of computational physics 62.1 (1986): 40-65.

[11] Oberkampf, William L., Timothy G. Trucano, and Charles Hirsch. "Verification, validation, and predictive capability in computational engineering and physics." Applied Mechanics Reviews 57.5 (2004): 345-384.

[12] Boehm, Barry. "Software risk management." ESEC'89 (1989): 1-19.

[13]] Schlesinger, Stewart, et al. "Terminology for model credibility." Simulation 32.3 (1979): 103-104.

[14] Dowson, Mark. "The Ariane 5 software failure." ACM SIGSOFT Software Engineering Notes 22.2 (1997): 84.

[15] P.G. Saffman. Vortex dynamics. Cambridge Umiversity Press, 1992.

[16] M.E. Brachet, D.I. Meiron, S.A. Orszag, B.G. Nickel, R.H. Morf, and U. Frisch. Small-scale structure of the taylor–green vortex. Journal of Fluid Mechanics, 130:411–452, 1983.

[17] G. Comte-Bellot and S. Corrsin. Simple eularian time correlation of full- and narrow-band velocity signals in grid-generated, 'isotropic' turbulence. Journal of Fluid Mechanics, 48:273–337, 1971.

[18] McCaffrey, Bernard J. Purely buoyant diffusion flames: some experimental results. 1979.

[19] P.G. Saffman. Vortex dynamics. Cambridge Umiversity Press, 1992.

[20] A.E. Green and G.I. Taylor. Mechanism of the production of small eddies from larger ones. In Proc. Royal Soc. A, volume 158, pages 499–521, 1937

[21] Beck, Andrea D., and Gregor J. Gassner. "Numerical simulation of the Taylor-Green vortex at Re= 1600 with the Discontinuous Galerkin Spectral Element method for well-resolved and underresolved scenarios." 1st International Workshop on High-Order CFD Methods at the 50th AIAA Aerospace Sciences Meeting, Nashville, TN. 2012.

[22] E. Johnsen, J. Larsson, A.V. Bhagatwala, W.H. Cabot, P. Moin, B.J. Olson, P.S. Rawat, S.K. Shankar, B. Sjö¨green, H.C. Yee, et al. Assessment of highresolution methods for numerical simulations of compressible turbulence with shock waves. Journal of Computational Physics, 229(4):1213–1237, 2010.

[23] M.E. Brachet, D.I. Meiron, S.A. Orszag, B.G. Nickel, R.H. Morf, and U. Frisch. Small-scale structure of the taylor–green vortex. Journal of Fluid Mechanics, 130:411–452, 1983.

[24] G.K. Batchelor. An introduction to fluid dynamics. Cambridge University Press, 1967.

[25] G. Comte-Bellot and S. Corrsin. Simple eularian time correlation of full- and narrow-band velocity signals in grid-generated, 'isotropic' turbulence. Journal of Fluid Mechanics, 48:273–337, 1971

[26] Smagorinsky, Joseph. "General circulation experiments with the primitive equations: I. The basic experiment." Monthly weather review 91.3 (1963): 99-164.

[27] Yoshizawa, Akira. "Statistical theory for compressible turbulent shear flows, with the application to subgrid modeling." The Physics of fluids 29.7 (1986): 2152-2164.

[28] C. Fureby, G. Tabor, H.G. Weller, and A.D. Gosman. Differential subgrid stress models in large eddy simulations. Phys. Fluids, 9 (11):3578–3580, 1997.

[29] C. Meneveau. Turbulence: Subgrid-scale modeling. Scholarpedia, 5:9489, 2010

[30] Cox, Geoffrey. Combustion fundamentals of fire. London etc: Acad. press, 1995.

[31] Delichatsios, Michael A. "Air entrainment into buoyant jet flames and pool fires." Combustion and Flame 70.1 (1987): 33-46.