Universität Rostock

Traditio et Innovatio

DOCTORAL THESIS

Resolution of Tip Vortices by grid-based, grid-free and coupled methods using CFD

Author:
Mahesh DHONE

Supervisor:
Prof. Dr.-Ing. habil. Nikolai KORNEV

*A thesis submitted in fulfillment of the requirements
for the degree of Doctor of Philosophy*

in the

Chair for Modeling and Simulation
Faculty of Mechanical and Ship Technology

January 24, 2020

https://doi.org/10.18453/rosdok_id00002635

Reviewers

1. Reviewer
   Prof. Dr.-Ing. habil. Nikolai Kornev
   Lehrstuhl für Modellierung und Simulation (LeMoS),
   Universität Rostock
2. Reviewer
   Prof. Dr. rer. nat. habil. Uwe Ritschel
   Lehrstuhl für Windenergietechnik
   Universität Rostock

Date of submission : 14.05.2019

Date of defense : 23.01.2020

Declaration of Authorship

I, Mahesh DHONE, declare that this thesis titled, "Resolution of Tip Vortices by grid-based, grid-free and coupled methods using CFD" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.

Signed: M.D.Dhone

Date: 24.01.2020

"Big whirls have little whirls,
Which feed on their velocity,
And little whirls have lesser whirls,
And so on to viscosity."

Lewis F. Richardson

Acknowledgements

I am using this opportunity to express my gratitude to everyone who supported me throughout my doctoral thesis work. I am thankful for their aspiring guidance and friendly advice during the project work. I am sincerely grateful to them for sharing their truthful and illuminating views on many issues related to the project.

I would like to first thank my parents and especially my beloved wife for the continuous encouragement, love and wholehearted support throughout the duration of my work which cannot be expressed in sufficient words. I am truly blessed. I would like to express my deepest appreciation and thanks to my advisor Professor Dr.-Ing. habil. Nikolai Kornev, you have been a tremendous mentor for me who has continually and persuasively conveyed a spirit of adventure in regard to the thesis work. I wish to thank him for the useful comments, remarks and engagement through the learning process of this thesis. Without his supervision and constant help this dissertation would not have been possible.

I gratefully acknowledge the funding received from the LGF Landesgraduiertenfoerderung, State of Mecklenburg-Vorpommern, Germany towards my doctoral work.

I have to appreciate the help given by Dr.-Ing. Hannes Kröger and M.Sc. Sina Samarbaksh in solving my doubts related to coding and by other team members Dr.-Ing. Johann Turnow , M.Sc. Robert Kasper , Dr.-Ing. Ivan Shevchuk, M.Sc. Andreas Gross and Dr.-Ing. Christian Krüger especially in my project phase presentations which has improved my presentation skills, thanks to their comment and advices. I would also thank to Dr.-Ing. Nawar Abbas to help me out with the data requirement during the initial phase of my work.

UNIVERSITY OF ROSTOCK

Chair for Modeling and Simulation

Faculty of Mechanical and Ship Technology

Abstract

Resolution of Tip Vortices by grid-based, grid-free and coupled methods using CFD

The vortex structure resolution is one of the vital problems of CFD as inherent artificial dissipation effects lead to an unphysical strong decay of the vortices. For example the evolution of the tip vortices of landing aircraft can give rise to the hazard of wake encounter by other successive craft. The overall objective of this work is to improve the resolution of concentrated vortices. This work focuses on grid based, grid free methods and coupled methods to capture the details of vortices especially further downstream after the vortex has rolled up and started to decay. The work focuses on a hybrid method as a coupling of grid based and grid free vortex method. Grid based methods are a very efficient and well developed tool to resolve the turbulent and laminar boundary layers close to the body. On the contrary, the particle based methods have many difficulties with formulation of boundary conditions and smooth representation of thin boundary layer. Therefore there are not much applications of vortex methods for real three dimensional configurations at high Reynolds numbers. On the other side, the grid based techniques have a substantial artificial numerical viscosity in the wake resulting in a non-realistic damping of vortex structures including the tip vortices. A natural way to escape these difficulties and to overcome disadvantages of grid based and particle based techniques is the application of domain decomposition. Close to bodies a grid based method is applied whereas far from the body a grid free one. In this work, we use the domain decomposition procedure originally developed by Cottet 's [112] group to improve the resolution of tip vortices by coupling the vortex method and the Finite Volume Method. The results are validated by means of experimental result of Devenport et al. [2] test case. These methods can be used for a wide spectrum of the fundamental and applied problems in fluid mechanics.

Contents

List of Figures

List of Tables

List of Abbreviations

CFD	Computational Fluid Dynamics
NSE	Navier Stokes Equations
FDM	Finite Difference Method
BDS, FDS, CDS, UDS	Backward, Forward, Central, Upwind Difference Scheme
CFL	Courant–Friedrichs–Lewy
FVM	Finite Volume Method
PISO	Pressure - Implicit with Splitting of Operators
SIMPLE	Semi-Implicit Method for Pressure Linked Equations
DNS	Direct Numerical Simulation
LES	Large Eddy Simulation
RANS	Reynolds averaged Navier Stokes
DES, DDES	Detached Eddy Simulation, Delayed Detached Eddy Simulation
SST	Shear Stress Transport
SA	Spalart Allmaras
CC	Curvature Correction
RSM-LRR	Reynolds Stress Model-Launder-Reece-Rodi
VLES	Very Large Eddy Simulation
AMR	Adaptive Mesh Refinement
AR	Aspect Ratio
AOA	Angle of Attack
VIC	Vortex-In-Cell
FFT	Fast Fourier Transform
FMM	Fast Multipole Method
TSR	Tip Speed Ratio
PIV	Particle Image Velocimetry
VAWT	Vertical Axis Wind Turbine

List of Symbols

t	Time	[s]
u_x, u_y, u_z	velocity	[$m.s^{-1}$]
ρ	density	[$kg.m^{-3}$]
τ	shear stress	[$N.m^{-2}$]
p	pressure	[$kg.m^{-1}.s^{-2}$]
μ	dynamic viscosity	[$kg.m^{-1}.s^{-1}$]
ν	kinematic viscosity	[$m^2.s^{-1}$]
$\overline{u}$	time averaged value of velocity	[$m.s^{-1}$]
$u^{'}$	velocity fluctuations	[$m.s^{-1}$]
ν_t	turbulent viscosity	[$m^2.s^{-1}$]
k	turbulent kinetic energy	[$m^2.s^{-2}$]
ϵ	dissipation rate	[$m.s^{-3}$]
Γ	circulation	[$m^2.s^{-1}$]
c	chord length	[m]
C_p	coefficient of pressure	[-]
C_L	coefficient of lift	[-]
C_D	coefficient of drag	[-]
β	overlapping factor	[-]
γ	vortex element strength	[$m^3.s^{-1}$]
ω	vorticity	[s^{-1}]
Re	Reynolds number	[-]
Fr	Froude number	[-]
K_t	thrust coefficient	[-]
K_q	moment coefficient	[-]
η	efficiency	[-]
n	rotational speed	[$rad.s^{-1}$]
δ_{ij}	Kronecker delta	[-]

CHAPTER 1

Introduction

Contents

1.1 State of the Art and Introduction

The wing tip vortices generated by an aircraft have been a long topic for discussion and development. They are generated inadvertently as a consequence of lifting [3]. It poses an important interest in this regard, since these vortices contain large amount of energy which can influence the aerodynamic performances and can affect the following departing or landing aircraft and thereby raising questions on the capacity of the airport [4] In marine engineering the most important application is the marine propeller hydrodynamics. In case of submerged ship propellers tip vortex cavitation may create noise and lead to erosion of the rudder. The noise prediction is of big importance because it disturbs animals like whales and reduces comfort of the people on ship board [7]. The propeller slipstream including the tip vortices prescribes the flow at the rudder and therewith influences the rudder forces. The lifting foils are used in shipbuilding to damp ship roll motions and for the creation of the dynamic lift on hydrofoil ships. In these cases the trailing tip vortices may influence the inflow of the propeller or foils moving downstream in the wake [7]. Also for Wing-in-Ground (WIG) effect vehicles the wake and tip vortex evolution has a strong influence on the flight stability [5] [6]. Also, another concern is in the field of wind turbine farm, where the vortex wake field from a wind turbine can affect the inlet flow of the successive wind turbine and thereby reduce efficiency.

These wake vortices can be considered into two regions, near wake and far wake, where the near wake region is dominated by the roll-up processes of several vortices, while the far wake region is dominated by the vortex decay. Coherent vortices like the tip vortices trailing from the lifting wings are very long lived in real world i.e they do not decay for a long distance but in numerical world they seem to decay faster due to numerical errors occurring from the discretization of equations on the grid and the errors arising in the turbulence modeling. The influence of tip vortices are directly related to induced drag and for example, it was observed that reducing drag by 10 percent on a large military aircraft saves upto 13 gallons of fuel [8]. Till today many of the developments for tip vortex reduction have been started and undertaken by Whitcomb NASA , most of which include winglets, wingtip sails, raked wing-tips etc [9] [10].

The prediction of tip vortices is a big challenge for Computational Fluid Dynamics as inherent artificial dissipation effects lead to an unphysically strong decay of the vortices. Holzapfel et al [3] in his work have described the analysis of wake vortex evolution and decay in the stably stratified, sheared, turbulent, and convective atmospheric boundary layer and thus lead to a theoretical foundation for the physics of vortex decay. The artificial dissipation or numerical diffusion results from numerical errors due to the discretisation and turbulence modeling. An accurate numerical prediction of the tip vortex evolution is limited by a huge amount of necessary computational resources and imperfection of available mathematical models.

The near-field flow structure of a wing tip vortex behind a sweptback and tapered NACA 0015 wing was also investigated using CFD code at Re of 1.81 x 10^5 which showed a good agreement with the experimental result as described by Samal et.al in his paper [11]. Further studies by Nash'at et. al. have been conducted to analyse the wake in the near field of a NACA0012 wing [12]. Kornev et. al. and Abbas studied CFD performance to predict the near vortex field in the wake of a oscillating wing at distances of one and half chords from the trailing edge [13]. They showed a good agreement with experimental results of Birch and Lee [14].

The evolution of the aircraft wake vortices from the roll-up until vortex decay was studied with LES on a NACA0012 wing and compared with experiments from Chow [15] by Misaka et.al in his paper [16]. They obtained the surrounding flow field by Reynolds-averaged Navier–Stokes simulation sweeping through a ground-fixed computational domain to initialize the wake. After the wake initialization a large-eddy simulation of the vortical wake was performed until vortex decay. Further the simulation in the near field of a NACA0012 half-wing was performed using the NASA's Fully Unstructured Navier-Stokes 3D (FUN3D) code (Anderson and Bonhaus 1994; Anderson et al. 1995; Nielsen 1998) and was studied by Ahmad and Proctor et. al. in their paper [17]. But most of these prior investigations were based on the analysis of the near wake region.

This work focuses on grid based, grid free and coupled methods to capture the details of vortices especially further downstream after the vortex has rolled up and started to decay. This work compares several different approaches to improve the prediction of tip vortices. The validation of the simulation results is performed with the experimental data obtained in a wind tunnel experiment which was conducted by Devenport and colleagues [2]. The experiment is an extensive study of a tip vortex trailing from a NACA0012 wing in a wind tunnel. The data was chosen to be suitable for the validation, since the tip vortex is tracked until 30 chord lengths downstream of the wing. Devenport and colleagues present experimental data of turbulence properties as well as the tangential and axial velocities in the vicinity of the vortex. The quality of the experimental data is superior to many other investigations because they were corrected for the wandering of the tip vortex core caused by instability of wind tunnel flow and possible wing vibrations [7].

According to the measurements of Devenport, the flow in the vicinity of the vortex core is laminar. Utilization of standard Reynolds Averaged Navier Stokes (RANS) turbulence models based on Boussinesq approach leads to an overprediction of the turbulence viscosity within the vortex core which in turn results in its increased decay rate. As a remedy of this disadvantage, improved turbulence models and approaches are validated in this work [7]. Furthermore, the grid free computational vortex method is coupled with the grid simulation to analyse the coherent vortices in the far wake.

1.2 Scope of this work

Insufficient resolution of vortex structures is one of the key problems in Computational Fluid Dynamics (CFD). Different approaches of modeling are discussed in this work for resolution of tip vortices in the wake. A new numerical technique (coupled method) has been developed and validated [7] [18] [19] to track the tip vortices especially in the far wake

The theoretical concepts of CFD are discussed in Chapter 2. The Finite Volume Method is briefly described in Chapter 3 along with different closure models used in our work in Chapter 4. The grid free vortex methods are introduced in Chapter 5. This work focusses on the validation based on the existing wind tunnel measurements of the flow behind the wing, done by Devenport in the case of wing tip vortices [2] and comparison based on the Blade Element Momentum Theory (BEMT) and validation with experiment in the case of rotating body tip vortices. (Refer to Chapter 7 and 8) The latter is also to prove the stability of the developed code of coupled method algorithm for unsteady rotating flows. This is oriented towards understanding the significance of the artificial viscosity or numerical error, which is mainly responsible for the unphysical dissipation of vortex structures in the numerical solution. A strategy of domain decomposition pursued is a novel hybrid approach (coupled method) based on combination of the grid-based finite volume method (FVM) and the grid-free vortex method (CVM). (Refer to Chapter 6) The work concentrates on further developments in the coupled method algorithm in the OpenFOAM framework towards achieving numerical stability, convergence and interface matching. It also studies the influence of different turbulence modeling as well as grid refinement approaches on the numerical solution. Numerous questions occur while achieving converged solutions, which are addressed by proper selection of simulation parameters. These methods can be used for a wide spectrum of the fundamental and applied problems in fluid mechanics with which many industrially important problems can be solved.

CHAPTER 2

Fundamentals of Computational Fluid Dynamics (CFD)

Contents

2.1 Introduction

Since early centuries, many scientists have addressed to solve the fluid flows with the usage of theoretical concepts and experiments. In the mid 20th century, with the introduction of computers and advancements in solving numerical algorithms precisely has lead to the development of the third approach of solving the fluid flows which was called as Computational Fluid Dynamics, commonly abbreviated as CFD [20] [22]. It is a branch of Fluid mechanics which involves mathematical modelling and numerically solving the fluid flow problems. Since then, until now it is being widely used in academics as well as in industry to solve a variety of problems involving fluids. The solution obtained by Computational Fluid Dynamics is comparable to the measurements obtained in a wind tunnel for the same set of input parameters. Since CFD methods involve a computing device instead of big wind tunnels, the same problem could be solved pretty easily on computers [20]. It also has advantages of data portability and data parametrization. In recent times, therefore it has become a very popular tool to deal with the variety of fluid flow problems.

In the early stages, CFD was applied to solve problems of 2-Dimensional flows, nowadays due to development of parallel computing architectures even complex 3-Dimensional problems could be solved in analogy with the real world problems which are also complex and 3-Dimensional in nature. The standard procedure of CFD consists of 4-steps - pre-processing, numerical setup, simulation and post-processing. CFD has become an important tool in the design process in a product development procedure in the industry today. Since construction of the prototypes for testing the product incur huge costs, CFD assists here to reduce the number of viable prototypes by doing some preliminary investigation studies, thereby saving efforts and overall costs. It has also an advantage of extracting the data wherever needed as compared to experiments which give data at limited number of points. It is difficult to say whether CFD will completely replace the testing procedures in future, since prototype testing is an integral part of any design process and is important to estimate behaviour of the product in real world.

Today CFD is used in many fields like automobile, aerospace, manufacturing, mechanical engineering, HVAC systems, civil engineering (flow over high rise building, bridges etc), environmental studies, naval architecture, bio-medical, chemical engineering, electronic devices, military organizations, meteorology and physiological studies [21] [23] [24].

2.2 Governing equations

2.2.1 Continuity equation

Consider a finite volume of fluid V enclosed by a surface S. Let dV be a small infinitesimal volume, dS a small infinitesimal surface patch and $\vec{n}$ as a normal vector of this surface patch. The velocity of the fluid is $\vec{u}$.

The continuity equation states that the total amount of fluid flowing into the volume V will be equal to the amount of fluid flowing out of the volume, provided there are no sources present inside the volume. In simple words, the net flow through the volume V is zero [26]. It represents conservation of mass. We can write it mathematically in the following way :

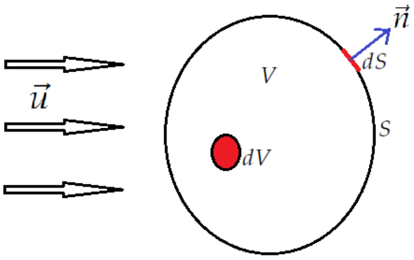


FIGURE 2.1: Fluid flow over a fluid finite volume V

$$\int_S \vec{u} \cdot \vec{n} dS = 0 \tag{2.1}$$

This represents the continuity equation for incompressible flows i.e ρ =const.

The normal vector has the components written as,

$$\vec{n} = \hat{i}cos(nx) + \hat{j}cos(ny) + \hat{k}cos(nz) \tag{2.2}$$

The equation 2.1 could be also represented in the terms of differential form and tensor form respectively as shown below.

$$\frac{\partial u_x}{\partial x} + \frac{\partial u_y}{\partial y} + \frac{\partial u_z}{\partial z} = 0 \tag{2.3}$$

$$\frac{\partial u_i}{\partial x_i} = 0 \tag{2.4}$$

The vector form of this equation is

$$\nabla \cdot \vec{u} = 0 \tag{2.5}$$

2.2.2 Stresses acting in the fluid

Consider a fluid element as shown in the Figure 2.2 in cartesian co-ordinates. τ_{ij} are the shear stresses acting parallel to the surface while p_{ii} are the normal stresses acting perpendicular to the surface. Also by symmetry $\tau_{ij} = \tau_{ji}$. Therefore the symmetric stress matrix can be written as

$$\begin{pmatrix} p_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{xy} & p_{yy} & \tau_{yz} \\ \tau_{xz} & \tau_{yz} & p_{zz} \end{pmatrix} \tag{2.6}$$

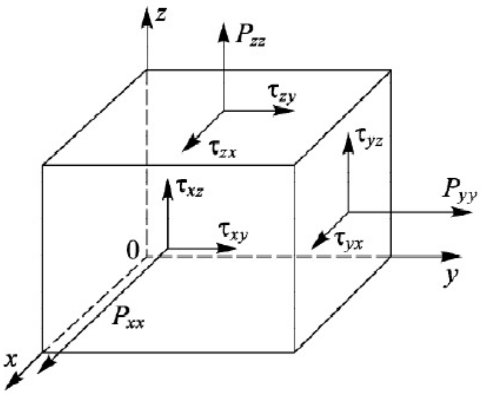


FIGURE 2.2: Stresses acting on a fluid element [26]

The generalised Newton hypothesis expresses this stress matrix in the form of pressure and strain rate.

$$\begin{pmatrix} p_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{xy} & p_{yy} & \tau_{yz} \\ \tau_{xz} & \tau_{yz} & p_{zz} \end{pmatrix} = - \begin{pmatrix} p & 0 & 0 \\ 0 & p & 0 \\ 0 & 0 & p \end{pmatrix} + 2\mu S_{ij} \tag{2.7}$$

where the strain rate tensor is

$$S_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) \tag{2.8}$$

Therefore, the three normal stresses can be written as follows

$$\begin{aligned} p_{xx} &= -p + 2\mu(\tfrac{\partial u_x}{\partial x}) \\ p_{yy} &= -p + 2\mu(\tfrac{\partial u_y}{\partial y}) \\ p_{zz} &= -p + 2\mu(\tfrac{\partial u_z}{\partial z}) \end{aligned} \tag{2.9}$$

And the three shear stresses can be written as follows

$$\begin{aligned} \tau_{xy} &= \mu(\tfrac{\partial u_x}{\partial y} + \tfrac{\partial u_y}{\partial x}) \\ \tau_{xz} &= \mu(\tfrac{\partial u_x}{\partial z} + \tfrac{\partial u_z}{\partial x}) \\ \tau_{yz} &= \mu(\tfrac{\partial u_y}{\partial z} + \tfrac{\partial u_z}{\partial y}) \end{aligned} \tag{2.10}$$

and the pressure in the fluid is

$$-p = \frac{p_{xx} + p_{yy} + p_{zz}}{3} \tag{2.11}$$

2.2.3 Momentum equation

The momentum is derived by the application of Newtons second law on a fluid element i.e the total force acting on the element will be equal to the sum of surface forces and body forces. This equation describes the non-relativistic momentum transport in a fluid element also known as Cauchy Momentum equation [25].

$$\rho \frac{D\vec{u}}{Dt} = \vec{F}_{body} + \vec{F}_{surface} \tag{2.12}$$

It can be written in vector form as

$$\rho \frac{D\vec{u}}{Dt} = -\nabla p + \mu \triangle \vec{u} \tag{2.13}$$

This equation represents conservation of momentum.

2.2.4 Navier Stokes equation NSE

Navier Stokes Equations are named after Georg Gabriel Stokes (1816–1903) and Louis Marie Henri Navier (1785–1836). These equations are build on the work of Leonhard Euler (1707–1783) and Daniel Bernoulli (1700-1782) who equations came to be known as Euler′s inviscid equations [27]. The Navier-Stokes equations came into their final form by the addition of Newtonian viscous terms by Sir George Stokes in 1845 [28] [29].

The Navier Stokes equation is written in the tensor form as:

$$\frac{\partial u_i}{\partial t} + \frac{\partial}{\partial x_j}(u_i u_j) = F_i - \frac{1}{\rho}\frac{\partial p}{\partial x_i} + \nu \frac{\partial}{\partial x_j}\left(\frac{\partial u_i}{\partial x_j}\right) \tag{2.14}$$

In the vector form as :

$$\frac{d\vec{u}}{dt} = \vec{F} - \frac{1}{\rho}\nabla p + \nu \triangle \vec{u} \tag{2.15}$$

Where

$$\vec{u} = \begin{pmatrix} u_x \\ u_y \\ u_z \end{pmatrix} ; \vec{F} = \begin{pmatrix} F_x \\ F_y \\ F_z \end{pmatrix} ; \nabla p = \begin{pmatrix} \frac{\partial p}{\partial x} \\ \frac{\partial p}{\partial y} \\ \frac{\partial p}{\partial z} \end{pmatrix} \tag{2.16}$$

Hence we have 4 unknowns u_x, u_y, u_z and p and we have 3 NSE momentum equations from 2.14 and one continuity equation from 2.1, which leads to a system of four equations in four unknowns. Therefore the system of partial differential equations is closed. We could also solve the energy of the system by addition of the conservation of energy equation i.e the first law of thermodynamics along with the already 3 NSE momentum equations and one continuity equation. In this case, we get a system of five equations with five unknowns as u_x, u_y, u_z, p and T and the system of partial differential equations gets closed.

The term on the left hand side of the equation 2.14 represents the acceleration of the fluid particle in which the first term i.e $\frac{\partial u_i}{\partial t}$ represents local acceleration i.e change of velocity in time while second term i.e $\frac{\partial}{\partial x_j}(u_i u_j)$ represents convective acceleration i.e motion of fluid particles due to velocity gradient in space. The second term is non-linear in nature. On the right hand side of the equation 2.14, the first term represents the external source i.e F_i, while the second term i.e $\frac{\partial p}{\partial x_i}$ represents internal source (fluid flow due to pressure gradient) and the last term i.e $\frac{\partial}{\partial x_j}\frac{\partial u_i}{\partial x_j}$ represents natural diffusion.

2.3 Finite Difference methods

In order to solve the partial differential equation 2.14 inside a computational domain, it is necessary to impose the boundary conditions on the extremities of the domain. For that, the equations have to be mathematically discretized in the respective directions. The numerical process behind this is termed as Finite Difference Method FDM. For example, a continuous process can be analysed in a finite number of discrete intervals [30]. Discretization process leads to the creation of finite difference equations which are converted to a system of algebraic equations and could be easily solved with the help of matrix representations [31]. This provides a noticeable advantage over the conventional analytical methods. For example, to solve a 1D boundary value problem within an interval [a, b], we shall define a finite number of nodes within the interval [a, b] with each node separated by step size h such that $x_i = a + i * h$ where $i = 0, 1, 2, 3, ..., N$ as shown in Figure 2.3 [32]. The differential equation is approximated by finite differences and then solved at each and every node.

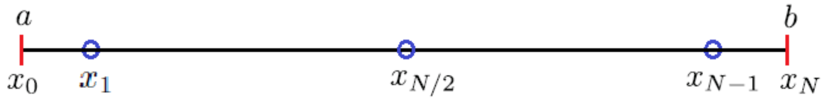


FIGURE 2.3: Discrete nodes for 1D problem within an interval [a, b]

2.3.1 Spatial Differencing schemes

Let us consider $\varphi(x)$ be a function over the x-axis with h = Δx as the step size.

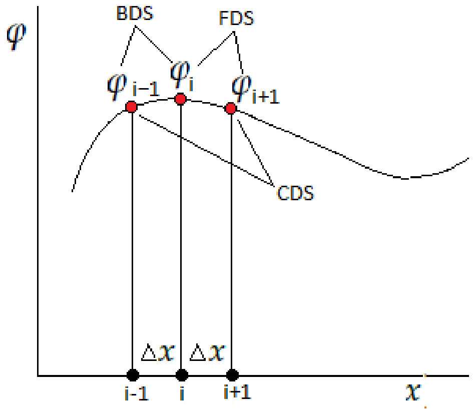


FIGURE 2.4: Application of schemes on the function u at the nodes

where i=1,2,3,....N. The approximation of the derivative $\frac{\partial \varphi}{\partial x}$ can be done with the application of Taylor series for the function φ at the point x_{i-1} [33]

$$\varphi_{i-1} = \varphi_i - h(\frac{\partial \varphi}{\partial x})_i + \frac{1}{2}h^2(\frac{\partial^2 \varphi}{\partial x^2})_i - .... \tag{2.17}$$

From this the Backward Difference Scheme (BDS) is obtained as follows -

$$(\frac{\partial \varphi}{\partial x})_i = \frac{\varphi_i - \varphi_{i-1}}{h} + O(h) \tag{2.18}$$

This scheme is of the first order of accuracy.

The approximation of the derivative $\frac{\partial \varphi}{\partial x}$ can be done with the application of Taylor series for the function φ also at the point x_{i+1} [33]

$$\varphi_{i+1} = \varphi_i + h(\frac{\partial \varphi}{\partial x})_i + \frac{1}{2}h^2(\frac{\partial^2 \varphi}{\partial x^2})_i + ... \tag{2.19}$$

From this the Forward Difference Scheme (FDS) is obtained as follows -

$$(\frac{\partial \varphi}{\partial x})_i = \frac{\varphi_{i+1} - \varphi_i}{h} + O(h) \tag{2.20}$$

This scheme is also of the first order of accuracy.

We get Central Difference Scheme (CDS) by subtracting equation 2.17 from 2.19.

$$(\frac{\partial \varphi}{\partial x})_i = \frac{\varphi_{i+1} - \varphi_{i-1}}{2h} + O(h^2) \tag{2.21}$$

This scheme is of the second order of accuracy. For the second order derivative it can be written as

$$(\frac{\partial^2 \varphi}{\partial x^2})_i = \frac{\varphi_{i+1} - 2\varphi_i + \varphi_{i-1}}{h^2} + O(h^2) \tag{2.22}$$

A hybrid scheme which combines both BDS and FDS is called as the Upwind Difference Scheme (UDS).

$$(\frac{\partial \varphi}{\partial x})_i = \begin{cases} \text{BDS} & \text{if } a > 0 \\ \text{FDS} & \text{if } a < 0 \end{cases} \tag{2.23}$$

This can be expressed as follows [34]

$$(\frac{\partial \varphi}{\partial x})_i = \begin{cases} a\frac{\varphi_i - \varphi_{i-1}}{h} & \text{if } a > 0 \\ a\frac{\varphi_{i+1} - \varphi_i}{h} & \text{if } a < 0 \end{cases} \tag{2.24}$$

In the case of 2D, the function depends on two variables i.e $\varphi(x, y)$ and the step size is Δx in x-direction and Δy in y-direction [33]. For example, an uniform stencil showing five point star with center as (i, j) is shown in the Figure below.

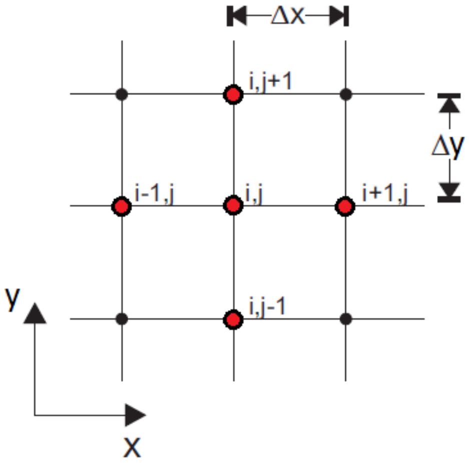


FIGURE 2.5: Uniform stencil showing five point star system

Therefore, the CDS approximation of the first order derivative $\frac{\partial \varphi}{\partial x}$ and $\frac{\partial \varphi}{\partial y}$ at the point x_i, y_j can be expressed as follows

$$(\frac{\partial \varphi}{\partial x})_{ij} = \frac{\varphi_{i+1,j} - \varphi_{i-1,j}}{2\Delta x} \tag{2.25}$$

$$(\frac{\partial \varphi}{\partial y})_{ij} = \frac{\varphi_{i,j+1} - \varphi_{i,j-1}}{2\Delta y} \tag{2.26}$$

While the the CDS approximation of the second order derivative $\frac{\partial^2\varphi}{\partial x^2}$ and $\frac{\partial^2\varphi}{\partial y^2}$ at the point x_i, y_j can be expressed as follows [35]

$$(\frac{\partial^2\varphi}{\partial x^2})_{ij} = \frac{\varphi_{i+1,j} - 2\varphi_{i,j} + \varphi_{i-1,j}}{\Delta x^2} \tag{2.27}$$

$$(\frac{\partial^2\varphi}{\partial y^2})_{ij} = \frac{\varphi_{i,j+1} - 2\varphi_{i,j} + \varphi_{i,j-1}}{\Delta y^2} \tag{2.28}$$

2.3.2 Time Differencing schemes

Suppose that we have an unsteady process described by the following equation [33]

$$\frac{\partial\varphi}{\partial t} = F(\varphi, t) \tag{2.29}$$

The discretization of this equation at time instant n at node i gives the following

$$\varphi_i^{n+1} = \varphi_i^n + \Delta t F_i(\varphi_i^n, t) \tag{2.30}$$

This is an explicit Euler scheme i.e the solution at the time instant $n+1$ depends on the solution at the previous time instant n. An implicit Euler scheme is where the solution at the time instant $n+1$ depends on the solution at the previous time instant n as well as on the solution at the next time instant $n+1$.

$$\varphi_i^{n+1} = \varphi_i^n + \Delta t F_i(\varphi_i^{n+1}, t) \tag{2.31}$$

A Crank-Nicolson scheme is a combination of explicit and implicit scheme [33]

$$\varphi_i^{n+1} = \varphi_i^n + \frac{1}{2}\Delta t(F_i(\varphi_i^n, t) + F_i(\varphi_i^{n+1}, t)) \tag{2.32}$$

2.4 Errors, Stability and Convergence

2.4.1 Error prediction

The solution obtained from numerical computation is approximate. There are always different kinds of errors involved which make the numerical solution diverge from the true solution. The understanding, analysis and control of errors and uncertainty in CFD simulations plays a crucial part in developing the trust and confidence on the final solution [36]. The errors can be classified broadly into three categories - Modeling errors, Numerical errors and User errors. These errors can be further sub-divided into various types as shown in the Figure 2.6.

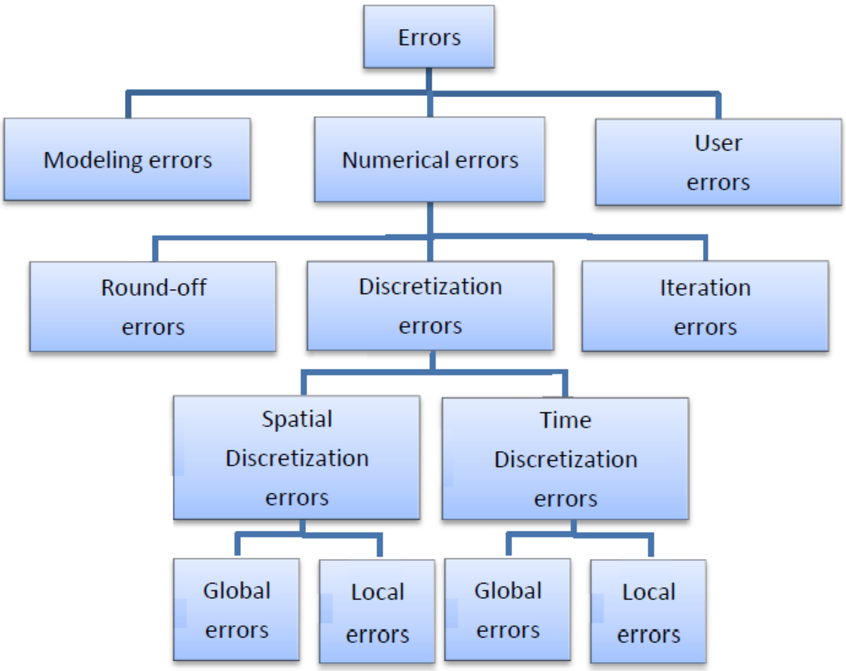


FIGURE 2.6: Classification of errors in CFD

Modeling Errors

In order to solve any CFD flow problem, a mathematical model is required. This model is created in order to represent the exact physical processes occurring within the system. But there always exist an inherent difference between the real flow data and the solution obtained by the mathematical model. This gives rise to an error termed as Modeling error [37].

Various turbulence models have been developed over the past years and are becoming more and more efficient. These turbulence models are being developed to reduce the resolution requirements in time and space and thus save the excessive computational cost involved in directly solving the exact equations i.e Direct Numerical Simulation [39]. They adhere the gap between the real flow and the statistically averaged equations i.e Reynold's Averaged Navier Stokes equations. Some information is also lost due to averaging procedures. Also, the accuracy of the simulation with usage of turbulence model cannot be improved further than the model capabilities itself. Therefore, modeling errors are difficult to estimate and therefore contribute to the uncertainty in the CFD methods [39]. In this work we observe the effect of modeling errors on the solution by application of different modeling approaches like grid-based methods (Curvature correction approach), grid-free methods and coupled methods which are described in later Chapters.

Numerical Errors

The differences in the solution obtained from the discretized equations and the exact equations results into errors commonly known as Numerical errors. These errors can be broadly classified into Round-off errors, Discretization errors and Iteration errors.

Round-off Errors Every computing system is based on the floating point number representation with precisions such as 16-Bit, 32-Bit or 64-Bit to store and process the data. The ability of the digital computers to represent the numbers is restricted due to their size and precision limits [38]. The errors which arise due to this are termed as round-off errors. They are not significant as compared to other errors provided the computing system is of the higher precision. For example, the number 0.06765 is represented by the system as 6.7X10e-2 , a roundoff error of 0.00065 is introduced. These errors may build up, when a certain routine performs a large number of computations [38].

Discretization Errors

Discretization errors result due to the difference between the numerical solution and the exact solution of the model equations. The numerical solution is obtained by representing the partial differential equations onto discrete space and time by usage of different Finite Difference Schemes as discussed in the previous section. The discrete space is termed as grid. Grid density is a major source of discretization errors. This error is of the most concern for any kind of CFD simulation. They have similar effect on the solution as the effect of natural diffusion by the viscosity of the fluid. This similar effect is called as numerical diffusion or artificial diffusion. Discretization errors rely upon how good the flow features are resolved by the grid and the quality of the grid which include various factors like the grid refinement, orthogonality, uniformity, aspect ratios, skewness, smoothness, grid transition zones, grid topology, grid alignment etc [40] [33].

Truncation errors arises due to replacement of the derivative terms in the differential equations like 2.18 by finite differences. The level of truncation error varies depending upon the depth of Taylor series used to obtain the finite differences. Truncation errors are the major source of Discretization errors [39]. Discretization errors can be classified further into spatial discretization errors and time discretization errors depending on the formulation of difference equations in space and time domains respectively.

Discretization error is bound for every grid node in the computational domain [39]. It can be both local as well as global. Local errors attribute to solution errors at a single grid node while Global error attribute to solution error over the entire computational domain [24].

$$Local\ error = \frac{\phi_{exact} - \phi_{numerical}}{\phi_{exact}}$$

$$Global\ error = \frac{||\phi_{exact} - \phi_{numerical}||}{||\phi_{exact}||}$$

Global error is essentially treated as the sum of the local errors. These errors transport, advect and diffuse throughout the grid domain [40]. The aim is always to keep this error below an acceptable limit. This can be controlled in OpenFOAM by the fvSolution dictionary and by the application of different numerical discretization schemes in the fvSchemes dictionary.

Iteration Errors

In numerical simulations the discretized equations are solved in an iterative way to obtain the final solution. An iterative process is applied to sought a steady state behavior in the solution. Ideally, one would require an infinite amount of iterations to obtain a very accurate solution but in reality, iteration process is stopped when the residuals fall below a certain level or some target variables converge, in order to reduce the numerical effort. The errors generated due to this process give rise to the iteration errors in the final solution. Residual norms (relative or absolute) can help to quantify these type of errors [39]. These errors can be controlled by usage of various numerical iterative methods along with their different control parameters in OpenFOAM by the fvSolution dictionery.

User Errors

User errors may occur from the incorrect use of code, poor construction of geometry, use of incorrect boundary conditions, incorrect use of solving parameters etc [39]. It accounts for human errors as well. Errors can also exist in CAD, grid generation software or during data transfer from one software to another [40].

2.4.2 Stability analysis

Numerical stability is a crucial aspect under consideration when solving algorithms. The algorithm for solving partial differential equation is stable if the total deviation of the numerical solution at a particular time instant remains bounded as the step size reduces. In the other words, when the numerical errors propagate and amplify after every step of the algorithm in such a way that they just blow up the calculation, the simulation becomes unstable while if the numerical errors do not amplify, the simulation remains stable [36]. If the errors in the input cause a significant error in the final output, then the algorithm exhibits numerically instability [41]. In order to quantify stability in numerical computations, CFL - Courant–Friedrichs–Lewy condition was introduced by Richard Courant, Kurt Friedrichs, and Hans Lewy which they described in their paper [42]. It is written as follows for the upwind differential scheme when approximated explicitly [43] -

$$C = \frac{||\vec{u}||\Delta t}{\Delta x} \tag{2.33}$$

For velocity as u_{max}

$$C = \frac{u_{max}\Delta t}{\Delta x} \tag{2.34}$$

C is called as CFL number or Courant number. The CFL condition states that perturbations decay when C < 1 in the case of explicit solvers [33]. Physically it means that when a fluid moves across a discrete spatial grid, then the maximum displacement of the fluid particle within the time step Δt does not exceed the cell size Δx. Thus the stability of the algorithm can be controlled effectively by reducing the time step Δt

2.4.3 Convergence evaluation

It is very important for a simulation to converge before we could proceed for interpretation of the results. Monitoring the residuals help us to quantify the errors directly in the solution. During each iteration process, the local imbalance of a conserved variables in each control volume is estimated through residuals. And so, a residual value is present in every cell of a grid. Usually, these local residuals are summed over the entire computational domain and then normalized as [36]-

$$R^{n+1}_{domain} = \sum_{i=1}^{N} |R^{n+1}_{i}|$$

$$R^{n+1}_{normalised} = \frac{R^{n+1}}{R^{0}}$$

As the value of residuals decrease, the errors decrease, therefore the accuracy of the numerical simulation increases. In the CFD simulations, the velocity residuals in the range of 1E-4 are considered to be loosely converged, in the range of 1E-5 are considered to be well converged, and in the range of 1E-6 are considered to be tightly converged. However for the complicated problems, a lower range of velocity residuals are even considered upto the range of 1E-4. For the unsteady calculations, the range of velocity residuals considered is upto 1E-3 to 1E-5 [36]. For example, in one of our simulations, the convergence of the residuals for pressure p and velocities u_x, u_y, u_z is clearly seen in Figure 2.7(left).

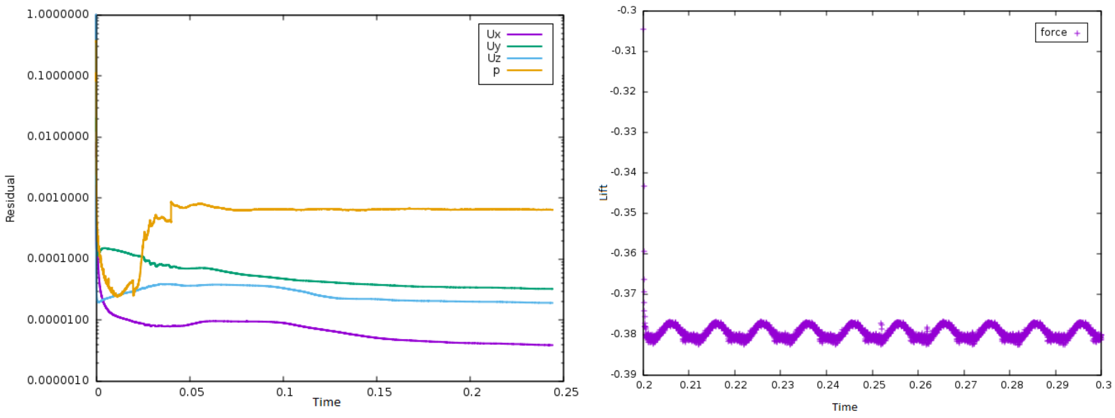


FIGURE 2.7: Monitoring the convergence in CFD

Another way of decide the convergence is to monitor the integrated quantities like lift, drag or instantaneous quantities like velocity at different desired locations in the computational domain over a period of time steps in the case of transient simulations. Whenever the quantities are observed to vary with a small amplitude within certain acceptable limited range of values, the convergence can be concluded. For example, in one of our simulations, the convergence of the lift force w.r.t time is clearly seen in Figure 2.7(right).

2.5 Boundary conditions

In order to solve a CFD problem, the region under consideration is enveloped with a computational domain. Therefore it becomes obvious to specify conditions at the boundaries of the computational domain so that the setup becomes analogous to the physical problem to be solved [44]. Without the application of initial and boundary conditions the system of fluid flow equations cannot be solved numerically. There are mainly three kinds of boundary conditions - Dirichlet, Neumann and Mixed. In Dirichlet boundary condition, the value of the variable is directly specified at the boundary, while in Neumann boundary condition, the gradient normal to the boundary is specified at the boundary. A mixed boundary condition is a combination of both Dirichlet and Neumann boundary conditions. For example, at the surface $S1$ [44] [45]

Dirichlet condition on S_1

$$u = u^*; T = T^* \qquad where \quad u^*, T^* \quad are \quad constants$$

Neumann condition on S_1

$$\frac{\partial u_n}{\partial n} = F_1; \frac{\partial T}{\partial n} = F_2$$

Mixed condition on S_1

$$au^* + b\frac{\partial u_n}{\partial n} = F_3$$

At a single boundary, even different types of boundary conditions can be used for different variables. For example, Dirichlet type for temperature and Neumann type for velocity at the outlet boundary. In OpenFOAM, the boundary conditions fall into 7 different categories which are listed as follows -

- **Basic** - Examples like fixedValue, zeroGradient, calculated etc
- **Constraint** - Examples like cyclic, symmetry, empty etc
- **Inlet** - Examples like freestream, turbulentInlet etc
- **Outlet** - Examples like freestream, totalPressure etc
- **Wall** - Examples like noSlip, movingWallVelocity etc

- **Coupled** - Examples like mappedField, uniformJumpAMI etc
- **Generic** - Examples like codedFixedValue, fixedProfile etc

CHAPTER 3

Grid-based - Finite Volume Method FVM

Contents

The Finite Volume Method is a widely used as grid based method in CFD due to some prominent contributions done by Professor Spalding [50], Patankar [51], Runchal [52] [53] and Gosman [54]. Some properties of the Finite Volume method are as follows [55] [36] -

- It is suitable for wide range of applications for modeling of fluid flows.
- The transported variables are conserved in the computational domain since the flux escaping the face of one control volume is equal the flux entering the same face of next control volume provided that the face is shared between those control volumes.
- The accuracy of the method depends on the errors arising from the discretization schemes which depend further on grid.
- The FVM has an inherent property to handle the discontinuities inside the Control Volumes

- In this method there no need for grid transformation between physical and computational space
- This method works for both structured as well as unstructured meshes.
- The FVM method can be applicable either in explicit form which is robust or in implicit form which is faster.

3.1 Transformation of Navier Stokes equation in FVM

The Navier Stokes equation from the equation 2.14 is

$$\frac{\partial u_i}{\partial t} + \frac{\partial}{\partial x_j}(u_i u_j) = F_i - \frac{1}{\rho}\frac{\partial p}{\partial x_i} + \nu \frac{\partial}{\partial x_j}\left(\frac{\partial u_i}{\partial x_j}\right) \tag{3.1}$$

The integral of this equation is valid in each grid cell or in a finite volume U. i.e

$$\int_U \left[\frac{\partial u_i}{\partial t} + \frac{\partial}{\partial x_j}(u_i u_j)\right] dU = \int_U \left[F_i - \frac{1}{\rho}\frac{\partial p}{\partial x_i} + \nu \frac{\partial}{\partial x_j}\left(\frac{\partial u_i}{\partial x_j}\right)\right] dU \tag{3.2}$$

The Gauss theorem is applied to equation 3.2 term by term [33].

Convection term modifies as

$$\int_U \left[\frac{\partial}{\partial x_j}(u_i u_j)\right] dU = \int_U \left[\frac{\partial}{\partial x}(u_i u_x) + \frac{\partial}{\partial y}(u_i u_y) + \frac{\partial}{\partial y}(u_i u_y)\right] dU = \int_S u_i\,(\vec{u}\cdot\vec{n})\,dS \tag{3.3}$$

Pressure source term modifies as

$$\frac{\partial p}{\partial x_i} = \vec{e_i}\nabla p\,;\; \int_U \frac{\partial p}{\partial x_i} dU = \int_S (\vec{e_i}\cdot\vec{n})\,p dS \tag{3.4}$$

Diffusion term modifies as

$$\int_U \left[\frac{\partial}{\partial x_j}\left(\frac{\partial u_i}{\partial x_j}\right)\right] dU = \int_U \left[\frac{\partial}{\partial x}\left(\frac{\partial u_i}{\partial x}\right) + \frac{\partial}{\partial y}\left(\frac{\partial u_i}{\partial y}\right) + \frac{\partial}{\partial z}\left(\frac{\partial u_i}{\partial z}\right)\right] dU$$

$$= -\int_S \left[n_x\left(\frac{\partial u_i}{\partial x}\right) + n_y\left(\frac{\partial u_i}{\partial y}\right) + n_z\left(\frac{\partial u_i}{\partial z}\right)\right] dS = \int_S (\vec{n}\nabla)\,u_i dS \tag{3.5}$$

Unsteady term and source term F_i is unchanged. Adding the equations 3.3, 3.4, 3.5 along with the unsteady and source terms, the final equation looks like

$$\frac{\partial}{\partial t}\int\limits_U u_i dU + \int\limits_S u_i\left(\vec{u}\cdot\vec{n}\right)dS = \int\limits_U F_i dU - \frac{1}{\rho}\int\limits_S\left(\vec{e}_i\cdot\vec{n}\right)pdS + \nu\int\limits_S\left(\vec{n}\nabla\right)u_i dS \tag{3.6}$$

Gauss theorem is also applied to the continuity equation as follows

$$\int\limits_U\left(\frac{\partial u_x}{\partial x}+\frac{\partial u_y}{\partial y}+\frac{\partial u_z}{\partial z}\right)dU = \int\limits_S\left(u_x n_x + u_y n_y + u_z n_z\right)dS = \int\limits_S \vec{u}\cdot\vec{n}dS = 0 \tag{3.7}$$

For example we consider a two dimensional case of above equation 3.6 by neglecting the diffusion term

$$\frac{\partial}{\partial t}\int\limits_U u_i dU + \int\limits_S u_i\left(\vec{u}\cdot\vec{n}\right)dS = -\frac{1}{\rho}\int\limits_S\left(\vec{e}_i\cdot\vec{n}\right)pdS \tag{3.8}$$

The above equation can be solved on a staggered grid as shown below in Figure 3.1 where pressure is at the center of volume, ux at center of vertical faces and uy at center of horizontal faces [33]. The volumes displaced in x-direction satisfy x-equation while those displaced in y-direction satisfy y-equation.

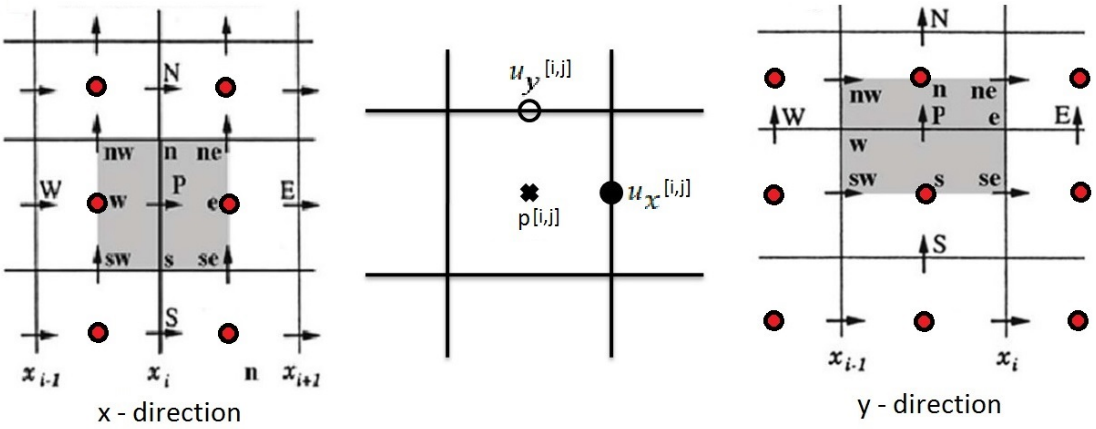


FIGURE 3.1: Staggered grid showing the control volume and location of pressure and velocities on the grid [33]

3.2 Principle of Explicit and Implicit schemes

The principle differences between explicit and implicit schemes is discussed in following table

Explicit scheme	**Implicit scheme**
A upper index ′n′ is used for the time derivative	A upper index ′n+1′ is used for the time derivative
The solution at the time instant ′n+1′ is explicitly expressed through the solution at the time instant ′n′	The solution at the time instant ′n+1′ is implicitly expressed through the solution at the same time instant ′n+1′
First the momentum equations are solved and then the velocity is calculated at the time instant ′n+1′	First the velocity is calculated at the time instant ′n+1′ and then the momentum equations are solved
The application of the explicit scheme is limited by the CFL criteria. The time step Δt should be very small to secure numerical stability	The application of the implicit scheme is not limited by the CFL criteria.

TABLE 3.1: Differences between explicit and implicit schemes

3.3 FVM in OpenFOAM

OpenFOAM stands for Open Source Field Operation and Manipulation. It is an object oriented C++ platform. The main classes of OpenFOAM and their functions are given in the table below which serve the application of the Finite Volume method [55].

C++ Objects	**OpenFOAM C ++ Classes**
Explicit operator	fvc::ddt, fvc::div, fvc::grad, fvc::curl
Implicit operator	fvm::ddt, fvm::d2dt2, fvm::div, fvm::laplacian
Interpolation	surfaceInterpolation<template>
Primitive variables	scalar, vector, tensor
Mesh components	point, face, cell
Finite Volume Mesh	fvMesh

TABLE 3.2: Numerics of Finite Volume Method in OpenFOAM

Consider a momentum equation for a generic scalar ϕ as follows [55] -

$$\underbrace{\frac{\partial(\phi)}{\partial t}}_{\text{unsteady term}} + \underbrace{\nabla\cdot(\vec{u}\phi)}_{\text{convection term}} = \underbrace{P\phi}_{\text{source}} + \underbrace{\nabla\cdot(D^{\phi}\nabla\phi)}_{\text{diffusion term}} \tag{3.9}$$

This is written in OpenFOAM language as follows with reference to the Table 3.2 -

```
// Finite volume discretization of the momentum equation

(
   fvm::ddt(phi)
+  fvm::div(mDot, phi)
-  fvm::laplacian(Dphi,phi)
==
   fvm::Sp(P, phi)
);
```

Here the operators fvc:: and fvm:: helps to build a system of equations which represents the discretized form of the equation 3.9 in each grid cell of the computational domain. The fvm::div and fvm::laplacian returns a fvMatrix containing all the coefficients which are obtained by the finite volume discretization of the convective and laplacian operators respectively. While the fvc::$P\phi$ returns a geometricField in which each grid cell contains the value of the variable $P\phi$ [55].

3.4 Pressure correction methods

3.4.1 Principle

The iterative procedure for implicit scheme gives the equation of the form u = Au + Bp + C. The velocity which satisfy this equation is the solution of linearized Navier Stokes equation. A pressure correction method is used to compute an iterative solution for the complete system of equations. The iterative algorithm consists of the following steps [33].

Step 1 : Intermediate solution is found by pressure from previous iteration

$$u^* = Au^* + Bp^{k-1} + C \tag{3.10}$$

Step 2 : Velocity and pressure is corrected

$$u^{(k)} = u^* + u^{'} \,,\; p^{(k)} = p^* + p^{'} \tag{3.11}$$

Step 3 : Since $u*$ satisfies the equation in Step 1 the equation for the velocity correction is

$$u^{'} = Au^{'} + Bp^{'} \tag{3.12}$$

Step 4 : The velocity at the iteration k is

$$u^k = u^* + Au^{'} + Bp^{'} \tag{3.13}$$

The velocity at kth iteration should satisfy continuity equation

$$\nabla u^{(k+1)} = 0 \tag{3.14}$$

Step 5 : This results into the following

$$\nabla u^{*} = -\nabla B p^{'} - \nabla A u^{'} \tag{3.15}$$

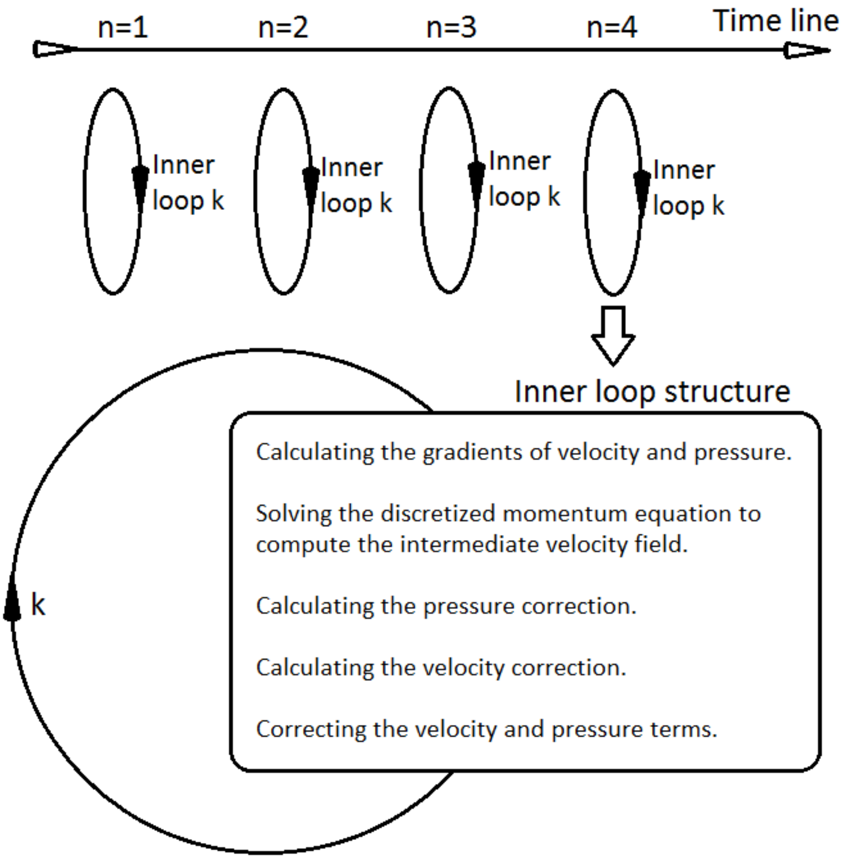
3.4.2 SIMPLE algorithm

SIMPLE method is very famous pressure correction method. SIMPLE stands for Semi-Implicit Method for Pressure Linked Equations. It is a numerical procedure to solve the Navier stokes equations [47]. We have used this algorithm for our steady calculations. The main assumption here is that we neglect the $\nabla A u^{'}$ term in the equation 3.15 and so we have

$$\nabla u^{*} = -\nabla B p^{'} \tag{3.16}$$

This equation is called as the Poisson equation for pressure correction p' [33].

The SIMPLE algorithm is shown in the below Figure 3.4.



Calculating the gradients of velocity and pressure.

Solving the discretized momentum equation to compute the intermediate velocity field.

Calculating the pressure correction.

Calculating the velocity correction.

Correcting the velocity and pressure terms.

FIGURE 3.2: Diagram of the SIMPLE algorithm [33]

Assume that the solution at the time instant 'n' is known and 'k' is the iteration number. In the first step of iteration all parameters are equal to those at previous time instant.

At k = 1

$$u_{ij} = u_{ij}^{n} \ , \ p_{ij} = p_{ij}^{n} \tag{3.17}$$

At each time instant 'n' the inner loop iterations are solved until it reaches certain tolerance limit

$$max|u_{ij}^{k} - u_{ij}^{k-1} = | < \varepsilon_u \ ; \ max|p_{ij}^{k} - p_{ij}^{k-1} = | < \varepsilon_p \tag{3.18}$$

Once the inner loop is converged, the process repeats again with the next time instant.

The SIMPLE algorithm is divided into small steps as follows [33] [47] -

Step 1 : Setting up the boundary conditions.

Step 2 : Calculating the gradients of velocity and pressure.

Step 3 : Solving the discretized momentum equation to compute the intermediate velocity field.

$$u^{*} = Au^{*} + Bp^{k-1} + C$$

Step 4 : Calculating the pressure correction.

$$\nabla Bp^{'} = -\nabla u^{*}$$

Step 5 : Calculating the velocity correction.

$$u^{'} = Bp^{'}$$

Step 6 : Correcting the velocity and pressure terms.

$$u^{k} = u^{*} + u^{'} \ , \ p^{k} = p^{k-1} + p^{'}$$

Step 7 : Repeat the loop until the convergence is attained

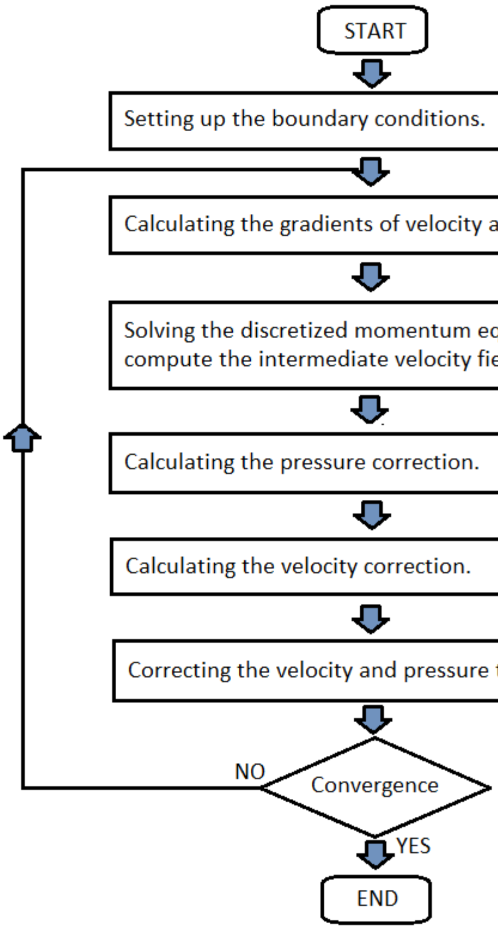

START
Setting up the boundary conditions.
Calculating the gradients of velocity and pressure.
Solving the discretized momentum equation to compute the intermediate velocity field.
Calculating the pressure correction.
Calculating the velocity correction.
Correcting the velocity and pressure terms.
NO
Convergence
YES
END

FIGURE 3.3: Flowchart depicting the SIMPLE algorithm [49]

3.4.3 PISO algorithm

PISO algorithm stands for Pressure - Implicit with Splitting of Operators. It is an extension of the SIMPLE algorithm. Instead of solving all the equations in an iterative manner, PISO splits the operators into one implicit predictor step and two explicit corrector steps [48]. We have used this algorithm for our unsteady calculations. The main assumption in the SIMPLE algorithm was that we neglect the $\nabla A u^{'}$ term in the equation 3.15, but in the PISO algorithm this term is taken into account in two step procedure as follows [33] -

In the first step, the term $\nabla A u^{'}$ is taken as zero and the calculation of pressure correction is done from the Poisson equation

$$\nabla B p^{'} = -\nabla u^{*}$$

After which the velocity correction is calculated from $u^{'} = B p^{'}$

In the second step, the calculation of pressure correction is done from the Poisson equation again within this step

$$\nabla B p^{''} = -\nabla A u^{'}$$

After which the velocity correction is calculated from $u^{''} = Au^{'} + Bp^{''}$

The PISO algorithm is divided into small steps as follows [33] [47] -

Step 1 : Setting up the boundary conditions.

Step 2 : Calculating the gradients of velocity and pressure.

Step 3 : Solving the discretized momentum equation to compute the intermediate velocity field.

$$u^{*} = Au^{*} + Bp^{k-1} + C$$

Step 4 : Calculating the pressure correction $p^{'}$.

$$\nabla B p^{'} = -\nabla u^{*}$$

Step 5 : Calculating the velocity correction $u^{'}$.

$$u^{'} = Bp^{'}$$

Step 6 : Calculating the pressure correction $p^{''}$.

$$\nabla B p^{''} = -\nabla A u^{'}$$

Step 7 : Calculating the velocity correction $u^{''}$.

$$u^{''} = Au^{'} + Bp^{''}$$

Step 8 : Correcting the velocity and pressure terms.

$$u^{k} = u^{*} + u^{'} + u^{''} \;,\; p^{k} = p^{k-1} + p^{'} + p^{''}$$

Step 9 : Repeat the loop until the convergence is attained

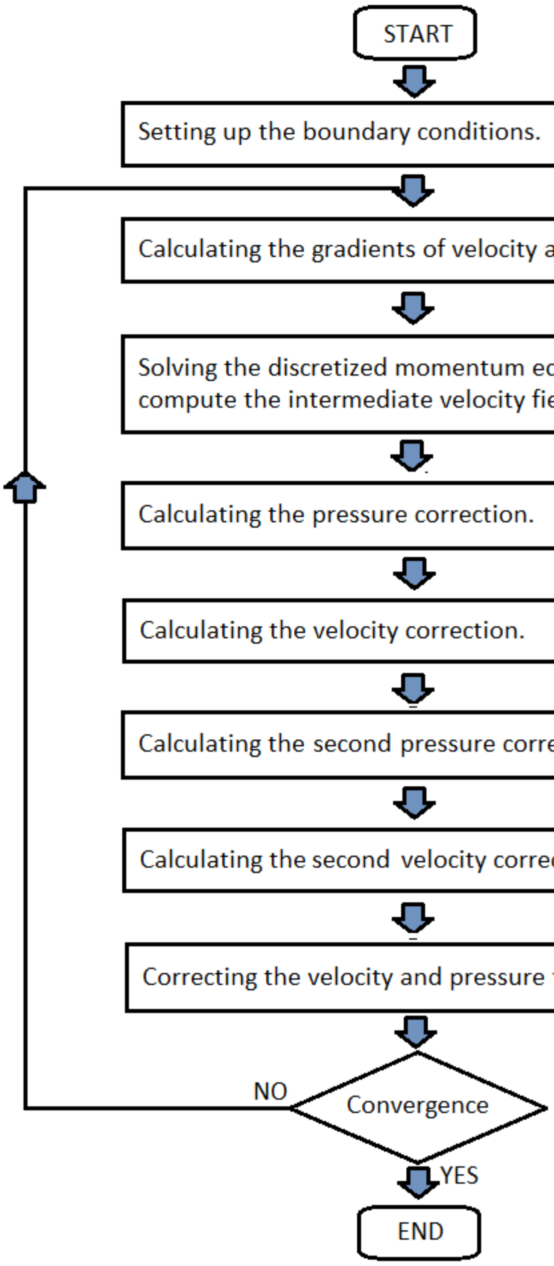

START
Setting up the boundary conditions.
Calculating the gradients of velocity and pressure.
Solving the discretized momentum equation to compute the intermediate velocity field.
Calculating the pressure correction.
Calculating the velocity correction.
Calculating the second pressure correction.
Calculating the second velocity correction.
Correcting the velocity and pressure terms.
NO
Convergence
YES
END

FIGURE 3.4: Flowchart depicting the PISO algorithm

3.4.4 PIMPLE algorithm

The PIMPLE algorithm is a mixture of PISO and SIMPLE algorithms. We have used this algorithm for our unsteady simulations. The PIMPLE algorithm contains a control parameter to control the number of outer correctors to switch the algorithm between PISO and SIMPLE. The iteration loops within each time step depend on the number of outer correctors. It can be controlled to obtain convergence within each time step ′n′ before moving to the next time instant ′n+1′. The rate of convergence can be monitored by the pre-defined absolute tolerance of the solver. The PIMPLE algorithm also incorporates the relaxation factors which could be wisely tuned to obtain a stable and converged solution.

The PIMPLE algorithm contains a control parameter to control the number of inner correctors i.e the number of pressure correction loops within each time step ′n′ before moving to the next time instant ′n+1′. Due to all these properties, the PIMPLE algorithm is more stable and fast especially when dealing with large time steps where we have Courant number consistently soaring above 1 or where the instability prevails in the nature of the solution [73]. The PIMPLE algorithm is shown in the below Figure 3.5 [57].

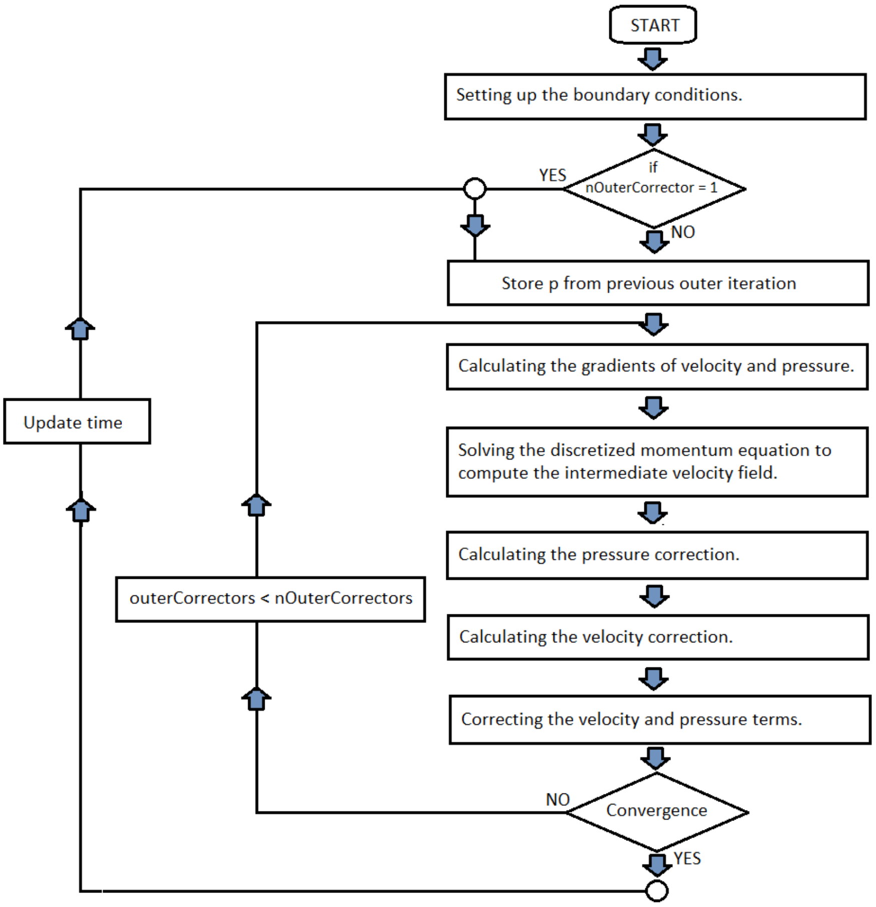

START
Setting up the boundary conditions.
YES
if
nOuterCorrector = 1
NO
Store p from previous outer iteration
Calculating the gradients of velocity and pressure.
Update time
Solving the discretized momentum equation to compute the intermediate velocity field.
Calculating the pressure correction.
outerCorrectors < nOuterCorrectors
Calculating the velocity correction.
Correcting the velocity and pressure terms.
NO
Convergence
YES

FIGURE 3.5: Flowchart depicting the PIMPLE algorithm

CHAPTER 4

Closure Models

Contents

4.1 Fundamentals of Turbulence

The understanding of the turbulent behaviour in flowing fluids is one of the most intriguing, complex and crucial problems in all of classical physics[58]. Most of the fluid flows in nature are turbulent. We see turbulence everywhere from the interior of biological cells, to circulatory and respiratory systems of living creatures, to countless technological devices and household appliances of modern society, in industry, oceans and atmospheres etc. Despite this ubiquity of turbulence, the complexity of turbulence remains to this day the most difficult problem of classical mathematical physics [58].

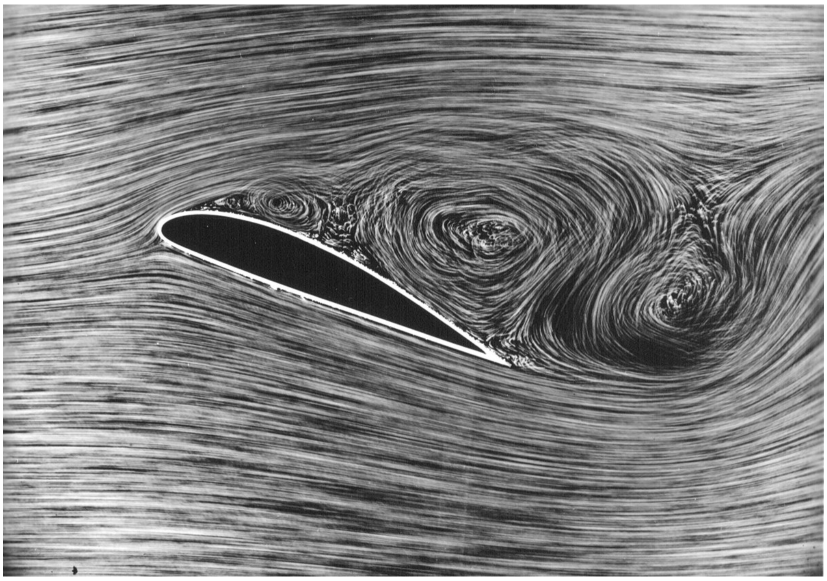


FIGURE 4.1: Photograph of a airfoil in a wind tunnel, showing flow separation over the top surface. Taken by DLR in 1915.

Some properties of a turbulent flow are given below - [58]

- fluctuations of the velocity field [26]
- disorganized motions of the fluid particles
- non-repeatability (i.e. sensitivity to initial conditions)
- enhanced diffusion (mixing) and dissipation (both of which are mediated by viscosity at molecular scales)
- three dimensionality, time dependence and rotational
- intermittent in both space and time.

4.1.1 Vortex cascade

Large vortices of scale L break up into small vortices which break up further into smaller vortices with scale η. The energy is transferred from these large vortices to the small ones. The energy gets dissipated in the small vortices also known as Kolmogorov vortices [26]. The Figure 4.2 shows an example of a vortex cascade

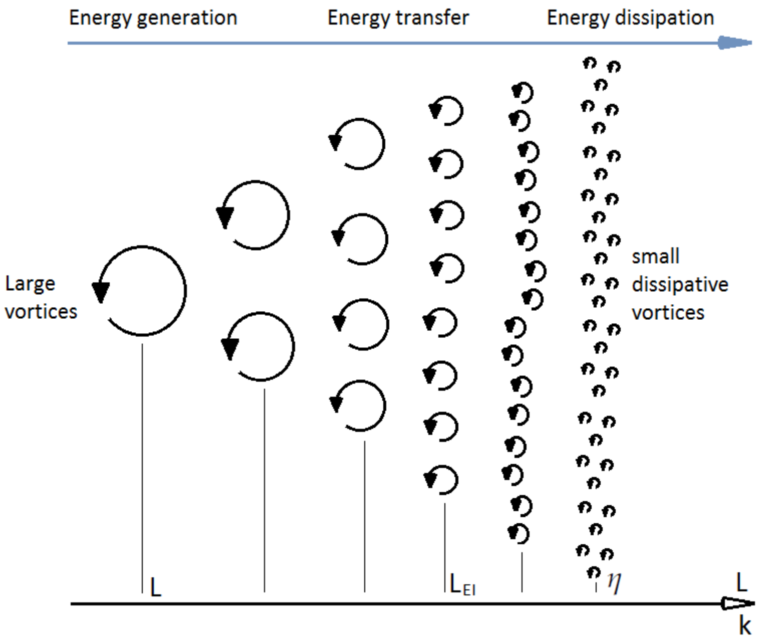


FIGURE 4.2: Example of an vortex cascade [26]

4.1.2 Energy spectrum of the turbulent flow

In the vortex cascade as described in the above section, three different zones are identified as follows [26].

- Energy Containing zone - Here energy is created and large vortices are formed of the scales $L > L_{EI}$ where L_{EI} is a length scale that separates anisotropic large vortices from the isotropic small vortices.
- Inertial zone - Here energy is transferred energy containing large vortices to the small dissipative vortices. The vortex scales are in the range $L_{DI} > L > L_{EI}$
- Dissipative zone - Here the energy of the large vortices dissipates through smaller vortices. The vortex scales are in the range $L < L_{DI}$

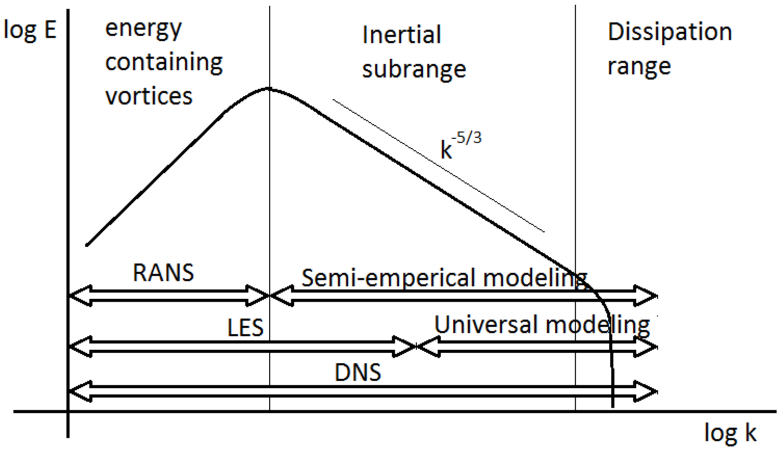


FIGURE 4.3: Energy spectrum of turbulent flow [26]

The Figure 4.3 shows the energy spectrum of the turbulent flow plotted over a wave number k. The three methods of turbulence modeling are classified using the energy spectrum as differentiated in the Table 4.1 below [26]. In this work the RANS and LES methods are being tested for the resolution of tip vortices.

DNS	**LES**	**RANS**
DNS stands for Direct Numerical Simulation	LES stands for Large Eddy Simulation	RANS stands for Reynolds averaged Navier Stokes
DNS models the complete spectrum of scales right from the energy containing large vortices upto the dissipative smaller vortices directly	LES models the energy containing large vortices and some part of inertial zone vortices directly.	RANS models the energy containing large vortices directly.
No further closure models are required	The remaining vortices are modeled by universal models.	The remaining vortices are modeled by semi-empirical models.

TABLE 4.1: Methods of turbulence modeling

4.2 Tip vortices

4.2.1 Turbulence in the vortex core

The tip vortex is rolled up near the tip region of a lifting body such as wing or a turbine blade when it is subjected to a free stream flow. In case of finite wings air spills spanwise over the wing-tips causing wingtip vortices. This tip vortex travels in the wake by convection mechanism. The roll up of the vortices from the surface of a lifting body takes place due to pressure difference between the suction and pressure sides of the body [78] [79]. According to the Lifting Line Theory, in potential flow the total circulation of the tip vortex is equal to the bound circulation over the wing, proportional to the lift. The vortex core is subjected to the following -

- high swirl velocities [81]
- low pressures in the core [80]
- contributions to the induced drag [82]
- axial velocity deficit w.r.t neighbouring flow [83]

Tip vortices are undesirable because they reduce the wing lift and increase the lift-induced drag on the lifting body [84] [85] [86]. Wing tip vortices create a region of downwind behind the wing limited to wing span and upwind beyond it. This downwind reduces the effective angle of attack of a behind finite wing and causes the induced drag. See Figures 4.4 and 4.5.

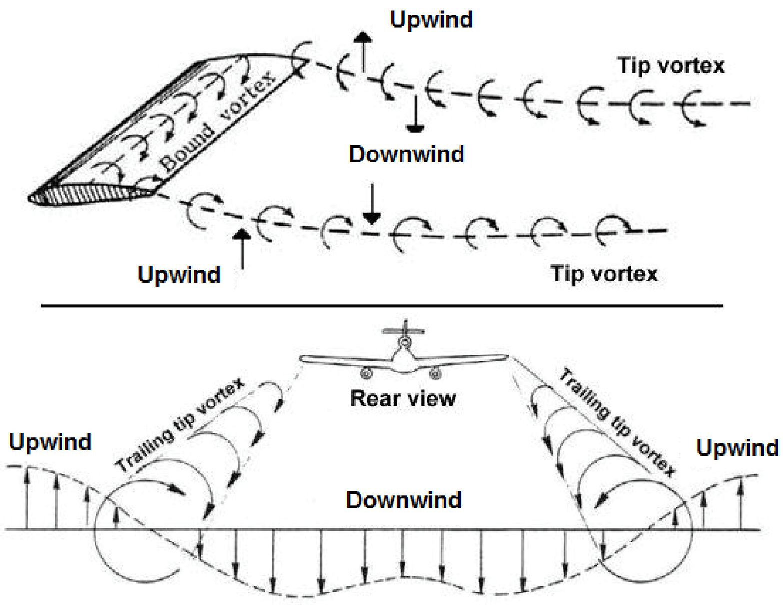


FIGURE 4.4: Upwash-Downwash behind a finite wing [87] [88]

The flow in the tip vortex core exhibits a laminar behavior. This was first proven by Zeman [89] where he solved the RANS using Reynolds Stress model with a analytical turbulent line vortex. He concluded the following - " The angular momentum transfer by turbulence is suppressed by the stabilizing effect of flow rotation ". Also, one more prediction from this study was done which says that the eddy viscosity models tend to predict a fully turbulent vortex core which results in diffusion and core growth [78] [89]. A LES simulation was adopted on an analytical q-type vortex by Ragab et. al. [90], he quoted the following - "If the vortex becomes linearly unstable, disturbances will grow in the form of helical sheets which effectively redistribute axial and angular momentum with the surroundings which results in a reduction of the axial velocity deficit, strengthening of rigid body rotation and a return to a stable laminar core". This argument was further backed by Qin et. al [91] who performed a low-Re DNS simulation for the same [78].

The fact of the existence of laminar core was strongly supported by Devenport et. al. [2] who was the first to provide accurate experimental results in the far wake of a wing tip vortex flow. He concluded that - " the flow in the vortex core is laminar and that velocity fluctuations experienced here are inactive motions produced as the core is buffeted by turbulence from the surrounding wake ". This is the test case which we refer for the validation of our work and we observed the similar effect in our computations too. Also in the near-field experiment performed by Chow et. al [92], the vortex initially was fully turbulent until the roll-up process was complete, thereafter quickly he observed "an extraordinary decay rate of overall turbulence" which is consistent with effect of re-laminarization.

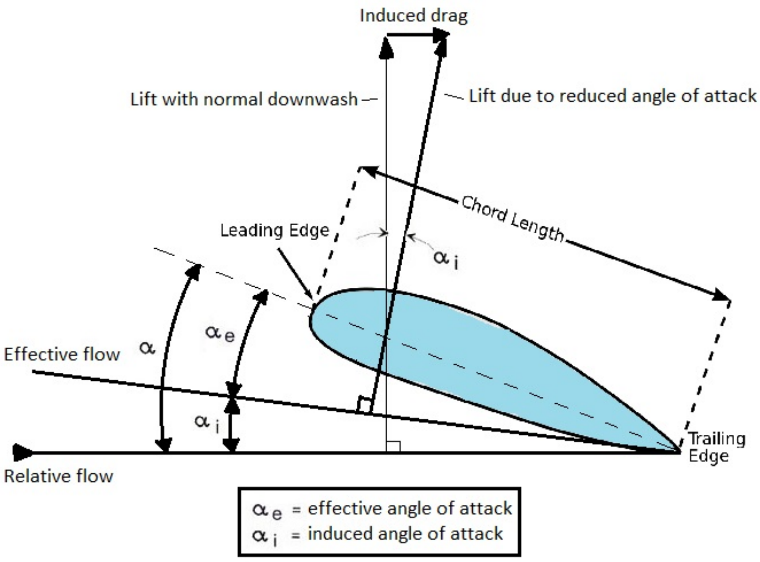

Induced drag
Lift with normal downwash
Lift due to reduced angle of attack
Chord Length
Leading Edge
α_i
α_e
α
Effective flow
α_i
Relative flow
Trailing Edge
α_e = effective angle of attack
α_i = induced angle of attack

FIGURE 4.5: Lift and drag acting on a wing

4.3 RANS

4.3.1 Reynolds averaging

Turbulent vortices generate fluctuations in velocity field [59].

We can represent any stochastic quantity in turbulent flow as sum of its averaged component and its fluctuation component [26]. Thus velocity in all the three directions can be written as -

$$u_x = \overline{u}_x + u_x^{'} \, ; \, u_y = \overline{u}_y + u_y^{'} \, ; \, u_z = \overline{u}_z + u_z^{'} \tag{4.1}$$

This is commonly known as Reynolds decomposition or averaging. Where the averaged velocities are [93]

$$\overline{u}_x = \lim_{T\to\infty} \frac{1}{T}\int_0^T u_x dt \, ; \, \overline{u}_y = \lim_{T\to\infty} \frac{1}{T}\int_0^T u_y dt \, ; \, \overline{u}_z = \lim_{T\to\infty} \frac{1}{T}\int_0^T u_z dt \tag{4.2}$$

4.3.2 Reynolds Averaged Navier Stokes (RANS) Equations

The Navier Stokes Equation is given by [26]

$$\frac{\partial u_i}{\partial t} + \frac{\partial}{\partial x_j} u_i u_j = F_i + \frac{1}{\rho}\frac{\partial \tau_{ji}}{\partial x_j} \tag{4.3}$$

where

$$\tau_{ji} = \rho \nu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - p\delta_{ij} \tag{4.4}$$

By applying the Reynolds Averaging procedure to the equation 4.3 we obtain the Reynolds Averaged Navier Stokes (RANS) Equation as seen equation 4.5 [26].

$$\frac{\partial \overline{u}_i}{\partial t} + \frac{\partial}{\partial x_j}\overline{u}_i\overline{u}_j = \overline{F_i} - \frac{\partial}{\partial x_j}\overline{u_i^{'}u_j^{'}} + \frac{1}{\rho}\frac{\partial \overline{\tau_{ji}}}{\partial x_j} \tag{4.5}$$

Unsteady Reynolds Averaged Navier Stokes (URANS) Equation which includes the fluctuation term $\overline{u_i^{'}u_j^{'}}$ is written as -

$$\rho\frac{\partial \overline{u}_i}{\partial t} + \rho\frac{\partial}{\partial x_j}\overline{u}_i\overline{u}_j = \rho\overline{F_i} + \frac{\partial}{\partial x_j}(\overline{\tau_{ji}} - \rho\overline{u_i^{'}u_j^{'}}) \tag{4.6}$$

As it is observed from the equation 4.3 that NSE equations are for instantaneous quantities while URANS equations 4.6 are for averaged quantities. The term -$\rho\overline{u_i^{'}u_j^{'}}$ is called as Reynolds stress R_{ij}. Considering in all three dimensions we obtain a symmetric Reynolds stress matrix [26]:

$$R_{ij} = \begin{bmatrix} -\rho\overline{u_x^{'}u_x^{'}} & -\rho\overline{u_x^{'}u_y^{'}} & -\rho\overline{u_x^{'}u_z^{'}} \\ -\rho\overline{u_x^{'}u_y^{'}} & -\rho\overline{u_y^{'}u_y^{'}} & -\rho\overline{u_y^{'}u_z^{'}} \\ -\rho\overline{u_x^{'}u_z^{'}} & -\rho\overline{u_y^{'}u_z^{'}} & -\rho\overline{u_z^{'}u_z^{'}} \end{bmatrix} \tag{4.7}$$

These Reynold's stresses are caused due fluctuations in the velocity field. By the procedure of Reynolds averaging, the small vortices are filtered out. This is the main disadvantage of the method making it non-universal [26].

We now have a system of four fluid equations : 3 URANS (Unsteady Reynolds Averaged Navier Stokes) equations from 4.6 in x-y-z directions and 1 continuity equation while we have ten unknowns : three averaged velocity components $\overline{u_x}$ & $\overline{u_y}$ & $\overline{u_z}$, one averaged pressure $\overline{p}$ and six Reynolds stresses R_{ij}. So system of equations is not closed. We need additional equations to express the Reynolds stresses R_{ij} in terms of velocity and pressure terms for the closure of the system of equations [26]. Most of the URANS models depend on Boussinesq Hypothesis which is described in the next section.

4.3.3 Boussinesq Hypothesis

URANS model depend on Boussinesq Hypothesis. Boussinesq Hypothesis is similar to the Newton Hypothesis (discussed in Chapter 1) but is valid for turbulent flows. The Reynolds stress is expressed as :

$$R_{ij} = -\rho\overline{u_i^{'}u_j^{'}} = \rho\nu_t\left(\frac{\partial \overline{u}_i}{\partial x_j} + \frac{\partial \overline{u}_j}{\partial x_i}\right) - \frac{2}{3}\rho\delta_{ij}k \tag{4.8}$$

where ν_t is turbulent viscosity and $k = \frac{1}{2}\overline{u_i' u_i'}$ is turbulent kinetic energy.

4.4 Modeling the turbulence

Several models have been developed till now for modeling the turbulence. These are broadly classified into RANS based models, LES models, DES and other hybrid models and Direct numerical simulations. These are further classified as given in the following diagram 4.7 with some examples

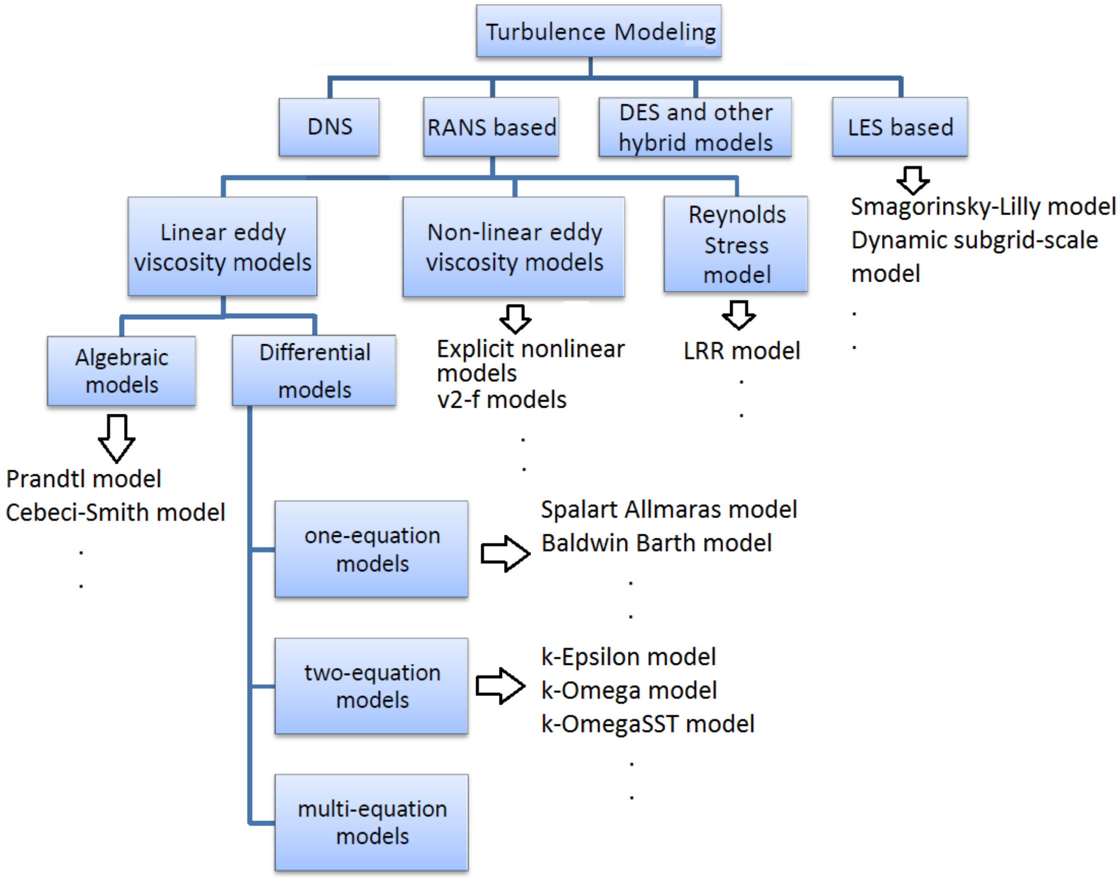


FIGURE 4.6: Classification of mathematical models of turbulence.

In the scope of this work, a variety of RANS based and LES based models are being validated for the tip vortex resolution in the case of flow over a steady wing and flow over a rotating body with respect to different grid refinement approaches. The detailed description of each of these mathematical models is explained in the sections hereafter.

4.4.1 Spalart Allmaras Model

The Spalart Allmaras turbulence model was developed by Spalart and Allmaras in 1994 [60]. It is an one-equation model written in terms of modified kinematic turbulent viscosity v. It stands out to be one of the efficient models in its category [26]. The equation for v is given as follows -

$$\frac{\partial v}{\partial t} + \overline{u}_j \frac{\partial v}{\partial x_j} = C_{b1} \xi v - C_{w1} f_w \left(\frac{v}{d} \right)^2 + \frac{1}{\sigma} \frac{\partial}{\partial x_k} \left[(\nu + v) \frac{\partial v}{\partial x_k} \right] + \frac{C_{b2}}{\sigma} \frac{dv}{dx_k} \frac{dv}{dx_k} \tag{4.9}$$

where the parameters are

C_{b1}	C_{b2}	C_{v1}	σ	C_{w2}	C_{w2}	C_{w3}	k
0.1355	0.622	7.1	2/3	$\frac{C_{b1}}{k^2} + \frac{1+C_{b2}}{\sigma}$	0.3	2.0	0.41

$$f_{v1} = \frac{\chi^3}{\chi^3 + C_{v1}^3} \ ; \ f_{v2} = \frac{\chi}{1 + \chi f_{v1}} \ ; \ f_w = g \left[\frac{1 + C_{w3}^6}{g^6 + C_{w3}^6} \right]^{1/6}$$

$$\chi = v\nu \ ; \ g = r + C_{w2}(r^6 - r) \ ; \ r = \frac{v}{\xi k^2 d^2}$$

$$\xi = S + \frac{v}{k^2 d^2} f_{v2} \ ; \ S = \sqrt{2\Omega_{ij}\Omega_{ij}} \ ; \ \Omega_{ij} = \frac{1}{2} \left(\frac{\partial \overline{u}_i}{\partial x_j} - \frac{\partial \overline{u}_j}{\partial x_i} \right)$$

4.4.2 k-Epsilon Model

The k-Epsilon turbulence model is a class of RANS model in which two additional partial differential equations (beyond those for averaged flow quantities) must be solved (one for the turbulence kinetic energy, k, and one for the dissipation rate, ϵ) to obtain length and time scale information which is needed to construct local (in space) eddy viscosities [58].

k-Epsilon model is one of the the most popular models used for solving practical engineering problems. They produce sufficiently good results, and are much less expensive in terms of computer execution times.

Turbulent Kinetic Energy k

The sum of three diagonal elements of Reynolds stress matrix Ref : 4.7 gives us the turbulent kinetic energy denoted by k [26].

$$k = \frac{1}{2} \overline{u_i' u_i'}$$

Dissipation rate ϵ

Turbulence energy dissipation rate is the rate at which turbulence energy is being converted to thermal energy by viscous effects on small scales. It is denoted by ϵ [26].

It is an two-equation model written in terms of Turbulent Kinetic Energy k and Dissipation rate ϵ. The equations are given as follows [26] -

k-equation :

$$\frac{\partial k}{\partial t} + \overline{u}_j \frac{\partial k}{\partial x_j} = \frac{\partial}{\partial x_j}\left(\left(\nu + \frac{\nu_t}{\sigma_k}\right)\frac{\partial k}{\partial x_j}\right) + \tau_{ij}\frac{\partial \overline{u}_i}{\partial x_j} - \varepsilon \tag{4.10}$$

Epsilon-equation :

$$\frac{\partial \varepsilon}{\partial t} + \overline{u}_j \frac{\partial \varepsilon}{\partial x_j} = \frac{\partial}{\partial x_j}\left(\left(\nu + \frac{\nu_t}{\sigma_\varepsilon}\right)\frac{\partial \varepsilon}{\partial x_j}\right) + \frac{C_{\varepsilon 1}\varepsilon}{k}\tau_{ij}\frac{\partial \overline{u}_i}{\partial x_j} - \frac{C_{\varepsilon 2}\varepsilon^2}{k} \tag{4.11}$$

The relation between dissipation rate, turbulent kinetic energy and integral lengths is according to the formula of Kolmogorov and Prandtl is given as

$$\varepsilon = C_D \frac{k^{3/2}}{L}$$

Under these following assumptions

- Generation of turbulent kinetic energy is equal to dissipation of turbulent kinetic energy.
- The turbulence is in equilibrium.
- Turbulent scales are in the inertial range.

Also the turbulent kinematic viscosity ν_t can be expressed in terms of turbulent kinetic energy k and dissipation rate ϵ using dimensionless analysis as

$$\nu_t = C_\mu \sqrt{k} L$$

$$\nu_t = C_\mu \frac{k^2}{\varepsilon} \tag{4.12}$$

where empirical constant $C_\mu = 0.09$

Once we find k and ϵ then the turbulent kinematic viscosity ν_t can be calculated from the equation 4.12 and thereafter the Reynolds stresses can be estimated from the Boussinesq Hypothesis as explained in the section 4.3.3. Finally these six Reynolds stresses can be substituted in the RANS equations. Then we remain with 4 equations and 4 unknowns. Therefore the problem is mathematically closed. This model solves two partial differential equations, therefore it requires less time of computation as compared to RSM model, making it cost efficient [26].

The k-Epsilon Turbulence model is very accurate at large Reynolds numbers. At small Re number this model is not applicable in the regions close to the wall. Therefore it is applied in regions far away from the walls. Near the walls another model called as k-Omega turbulence model is applied.

4.4.3 k-OmegaSST Model

The k-OmegaSST model is also commonly abbreviated as the SST model in short where SST stands for Shear Stress Transport. The SST model is a blend between two models, first is the k-Omega model, which is used in near wall region, and second the k-epsilon model, which is used in regions far from the wall. The SST model is fairly robust and generally is good near solid boundaries. It is also better at capturing the recirculation regions as compared to the other models. The k-Omega model has several advantages, namely that:

- The model is reported to perform better in transitional flows and in the flows with adverse pressure gradients.
- The model is numerically very stable, especially the low-Reynolds model, as it tends to produce converged solutions more rapidly than the k-epsilon model at low Re numbers.
- The low-Re model is more economical and elegant than the low-Re k-epsilon models, that it does not require the calculation of the wall distances, additional source terms and/or damping functions based on the friction velocity.

The "low Reynolds number" designation means that the model can be used throughout boundary layers and beyond. A model that is not "low Reynolds number" requires additional wall functions in order to correctly handle the effect of viscosity near walls. Combined this with the advantages of k-Epsilon model in the regions far away from the wall, make SST model as one of the best turbulence models widely used.

The equations of k-OmegaSST Model are as follows -

$$\frac{\partial k}{\partial t} + u_j \frac{\partial k}{\partial x_j} = P_k - \beta^* k\omega + \frac{\partial}{\partial x_j}\left(\left(\nu + \nu_t \sigma_k\right) \frac{\partial k}{\partial x_j} \right) \tag{4.13}$$

$$\frac{\partial \omega}{\partial t} + u_j \frac{\partial \omega}{\partial x_j} = \alpha S^2 - \beta \omega^2 + \frac{\partial}{\partial x_j}\left(\left(\nu + \nu_t \sigma_\omega\right) \frac{\partial \omega}{\partial x_j} \right) + 2\left(1 - F_1\right) \sigma_{\omega 2} \frac{1}{\omega} \frac{\partial k}{\partial x_i} \frac{\partial \omega}{\partial x_i} \tag{4.14}$$

where the closure terms are as as follows -

$$F_2 = tanh\left[\left[max\left(\frac{2\sqrt{k}}{\beta^* \omega y}, \frac{500\nu}{y^2 \omega} \right) \right]^2 \right]$$

$$P_k = min\left(\tau_{ij}\frac{\partial U_i}{\partial x_j}, 10\beta^* k\omega\right)$$

$$F_1 = tanh\left\{\left[min\left[max\left(\frac{\sqrt{k}}{\beta^*\omega y}, \frac{500\nu}{y^2\omega}\right), \frac{4\sigma_{\omega 2}k}{CD_{k\omega}y^2}\right]\right]^4\right\}$$

$$CD_{k\omega} = max\left(2\rho\sigma_{\omega 2}\frac{1}{\omega}\frac{\partial k}{\partial x_i}\frac{\partial \omega}{\partial x_i}, 10^{-10}\right)$$

$$\phi = \phi_1 F_1 + \phi_2\left(1 - F_1\right)$$

and the parameters are

α_1	α_2	β_1	β_2	β^*	σ_{k1}	σ_{k2}	$\sigma_{\omega 1}$	$\sigma_{\omega 2}$
5/9	0.44	3/40	0.0828	9/100	0.85	1.0	0.5	0.856

4.4.4 Curvature correction models

4.4.4.1 Principle

In computational fluid dynamics it is very important to consider the influences of rotation and curvature on the flows dynamics. Eddy viscosity models have relatively low cost of computing for the turbulent viscosity and are more resilient and stable. These include basic models like Spalart Allmaras, k-OmegaSST etc. One of the major drawback of these models is that they are not able to apprehend the influences of streamline curvature and system rotation. As on the other side, RSM i.e Reynolds Stress models imply explicit curvature production term therefore are much better in considering this effect but this comes at high cost of computation. To overcome this, an approach of introducing curvature correction terms for the eddy viscosity models was proposed by Spalart and Shur [61] [62]. They applied this term to the one equation eddy models SA which showed significant improvement with inclusion of curvature effects for the rotating and curved channel flows. In their study, they performed a validation for 1-Dimensional, 2-Dimensional and 3-Dimensional channel flows with Reynolds number in the range of 10^6 and was compared to the models with and without curvature correction, experimental and DNS data [61].

Another promising approach to take the curvature effects into account was proposed by Smirnov and Menter [63] for the $k-\omega$SST model. The model was validated for 1D turbulent flow for a rotating plane, 2D flow for curved channels and 3D flow for centrifugal compressor etc and were compared with DNS data and experimental predictions [63].

In order to consider the effects of curvature in the eddy viscosity models, an empirical function for curvature correction introduced by Spalart-Shur is as follows [61] -

$$f_{rotation}(r^*, \widetilde{r}) = (1 + c_{r1})\frac{2r^*}{1 + r^*}[1 - c_{r3}tan^{-1}(c_{r2}, \widetilde{r})] - c_{r1} \tag{4.15}$$

This function is used to modify the production term of the original model. The function is then clipped as follows -

$$f_{r1} = max[min(f_{rotation}, 1.25), 0] \tag{4.16}$$

The function f_{r1} is varied in the interval starting from 0 corresponding to no turbulence production up to 1.25 corresponding to the enhanced turbulence production respectively. The non-dimensional quantities defined in the function (4.15) assuming all variables and their derivatives are specified w.r.t reference frame are given as follows -

$$\widetilde{r} = 2\omega_{ik}S_{jk}\left(\frac{DS_{ij}}{Dt} + (\varepsilon_{imn}S_{jn} + \varepsilon_{jmn}S_{in})\Omega_m\right)/D^4 \tag{4.17}$$

$$S_{ij} = 0.5\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) \tag{4.18}$$

$$\omega_{ij} = 0.5\left(\left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i}\right) + 2\varepsilon_{mji}\Omega_m\right) \tag{4.19}$$

$$S^2 = 2S_{ij}S_{ij}, \omega^2 = 2\omega_{ij}\omega_{ij}, D^2 = 0.5(S^2 + \omega^2) \tag{4.20}$$

$$r^* = \frac{S}{\omega} \tag{4.21}$$

The constants c_{r1}, c_{r2} and c_{r3} are defined as -

$$c_{r1} = 1.0, c_{r2} = 2.0, c_{r3} = 1.0 \tag{4.22}$$

The benefits of the curvature models are that they improve the original model, the accuracy predications with curvature are higher than without curvature, they are competitive with RSM model but comparable computing cost is less but they take slightly more time for computation as compared to original models without curvature.

4.4.4.2 Spalart Allmaras Model with curvature correction

The production term of the original one equation Spalart Almaras (SA) [60] model is modified in order to consider the curvature effects [61]. In the original model, the vorticity reaches a local maximum in the core of a vortex causing the eddy viscosity to increase rapidly. This leads to excessive dissipation of the vortex core. To solve this problem the production term is modified by multiplying by a correction factor.

$$\frac{\partial \nu}{\partial t} + u_i \frac{\partial \nu}{\partial x_i} = \frac{1}{\sigma}[\bigtriangledown \cdot ((\nu + \nu) \bigtriangledown \nu) + C_{b2}(\bigtriangledown \nu^2)] + P(\nu) - D(\nu) \tag{4.23}$$

where

$$P(\nu) = C_{b2}\nu \left(\Omega + \frac{\nu}{\kappa^2 d^2} f_{\nu 2} \right) f_{r1} \tag{4.24}$$

4.4.4.3 k-OmegaSST Model with curvature correction

A similar modification is done with the two equation i.e k-OmegaSST model to include the effects of curvature by multiplying the correction factor to the production terms in the original model [64] [63].

$$\frac{\partial(\rho k)}{\partial t} + \frac{\partial(\rho u_j k)}{\partial x_j} = P_k f_{r1} - \beta^* \rho k \omega + \frac{\partial}{\partial x_j}[\mu_{ef} \frac{\partial k}{\partial x_j}] \tag{4.25}$$

$$\frac{\partial(\rho \omega)}{\partial t} + \frac{\partial(\rho u_j \omega)}{\partial x_j} = \alpha \frac{\rho P_k}{\mu_t} f_{r1} - D_\omega + C d_\omega + \frac{\partial}{\partial x_j}[\mu_{ef} \frac{\partial \omega}{\partial x_j}] \tag{4.26}$$

4.4.5 Reynolds Stress Model RSM-LRR

RSM model is used to determine the Reynolds stresses directly from the transport equation instead of obtaining them through eddy viscosity hypothesis. Therefore it does not depend on the Boussinesq approach and therefore takes the anisotropy of stresses into account. This model is one of the best RANS models for calculating the complex interactions within a turbulent flow. Therefore the RSM model offer significant accuracy over the eddy viscosity models. They capture the flow separation regions better in the case of a flow over a wing than the eddy viscosity models. There are computationally expensive as compared to eddy viscosity models but cheaper as compared to DNS or LES models.

The Reynolds Stress Model (RSM) equations can be written as [26]

$$\frac{\partial}{\partial t}(\overline{u_i' u_k'}) + \overline{u_j} \frac{\partial}{\partial x_j}(\overline{u_i' u_k'}) = \frac{\partial}{\partial x_j} D_{jk} + P_{ik} + \Pi_{ik} + \Omega_{ik} - \varepsilon_{ik} \tag{4.27}$$

where the first term on LHS denotes the rate of change of R_{ik} while the second term denotes the transport of R_{ik} by convection.

The first term on RHS denotes diffusion of R_{ik} where :

$$D_{jk} = \nu \frac{\partial \overline{u_i' u_k'}}{\partial x_j} + \frac{1}{\rho} \overline{(\delta_{jk} u_i' + \delta_{ij} u_k') p'} - \overline{u_i' u_j' u_k'} \tag{4.28}$$

While the second term denotes production of R_{ik}

$$P_{ik} = -\overline{u_j' u_k'} \frac{\partial}{\partial x_j}(\overline{u}_i) - \overline{u_j' u_i'} \frac{\partial}{\partial x_j}(\overline{u}_k) \tag{4.29}$$

The third term denotes the transport of R_{ik} due to turbulent pressure-strain interactions and the fourth term denotes the rotation of R_{ik}

The fifth term denotes dissipation of R_{ik}

$$\varepsilon_{ik} = 2\nu \overline{\frac{\partial u_i'}{\partial x_j} \frac{\partial u_k'}{\partial x_j}} \tag{4.30}$$

The six partial differential equations 4.27 solve for the six Reynolds stresses. The production term P_{ik} is closed while the terms like pressure strain correlation Π_{ik} and dissipation ϵ_{ik} are not closed and therefore require closure models. Different closure models are being developed by many people till today like the LRR - Launder-Reece-Rodi model [65], the Hallback-Johanssen model [66], the Speziale-Sarkar-Gatski model [67] etc. Out of which LRR model is famous widely for solving the flows involving separation.

4.4.6 VLES Model (w/o Turbulence)

One of a serious disadvantages of turbulence models which leads to a fast vortex decay is the overprediction of the turbulence intensity in the vicinity of the tip vortex core. To analyse this issue, some simulations are carried out without turbulence modelling, i.e. the unsteady Navier-Stokes equations are solved. This can be classified as an underresolved very large eddy simulation (VLES) without a sub-grid model. In OpenFOAM framework, this type of modeling is named as 'laminar' and is defined as no usage of any turbulence model [73].

4.4.7 Large Eddy Simulation LES

Large Eddy Simulation is one of the widely used method of modeling for a variety of applications in CFD. It was first proposed by Joseph Smagorinsky in 1963 to simulate the atmospheric currents [68] and then the model was applied for turbulent flow channel by Deardorff in 1970 [69]. The difficulty in turbulence modeling is to accurately capture the contributions of all scales of the spectrum [70].

Direct Numerical Simulations require a very fine grid resolution inorder to resolve even the smallest turbulence scales which in many cases need a large computing capacity [70] [71]. A coarser grid however will not be able to resolve the smaller scales i.e smaller than one or two grid cells. The influence of the smaller scales without actually resolving them into the main flow simulation could be done by subsidiary modeling called as subgrid scale modeling. LES therefore stands out to be advantageous as compared to RANS models, since they not only consider the influence of the large scales by resolving them directly onto the grid but also take into account the effect of smaller scales through subgrid scale modeling. The idea therefore behind LES is to bring down the high computational cost which is involved in resolving smaller scales as compared to DNS.

LES filtering

From the Kolmogorov Theory theory of self similarity a conclusion is drawn that the large eddies of the flow depend on the geometry while the smaller eddies are more universal. This makes one to solve the large eddies directly on the grid while the smaller ones by application of sub-grid scale modeling. Mathematically, any field is separated into two parts - first resolved part representing the large eddies in the flow and second the sub-grid part representing the smaller eddies in the flow. This separation is done by using spatial filtering operation [26].

Let G be the filtering kernel and ϕ be any stochastic field function. This function can be written as the sum of filtered part and fluctuation [26].

$$\phi(\vec{x},t) = \tilde{\phi}(\vec{x},t) + \phi^{'}(\vec{x},t)$$

where

$$\tilde{\phi}(\vec{x},t) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} \phi(\vec{x}-\vec{s},t)G(\vec{s})d\vec{s} \tag{4.31}$$

where the filtering function satisfies the following

$$\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} G(\vec{s})d\vec{s} = 1 \tag{4.32}$$

In the equation 4.31 different filter functions could be used. Some examples are Gauss filter, Box filter etc. Generally a box filter is used in Finite Volume Method. This is also termed as simple filter in OpenFoam which we have used in our LES simulations. It can be defined as follows -

Let Δ be the filter width, $G(x)$ be filtering kernel in physical space and $G(k)$ be the filtering kernel in Fourier space.

$$G(x) = \frac{1}{\Delta}H\left(\frac{1}{2}\Delta - |x|\right) \tag{4.33}$$

Where H is the Heaviside function

$$G(k) = \frac{sin(\frac{1}{2}k\Delta)}{\frac{1}{2}k\Delta}$$

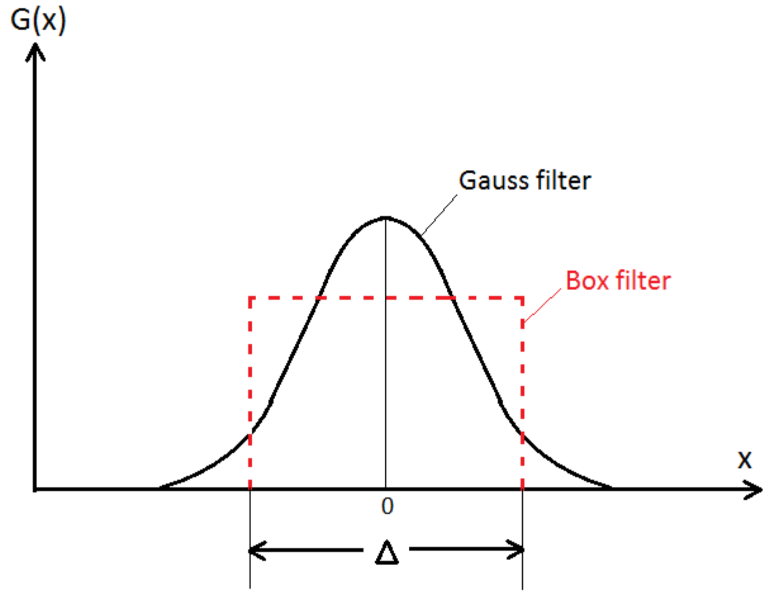


FIGURE 4.7: Different filtering functions of LES [72]

LES equations

The equation for LES are obtained by applying filter operation described in the previous section on the Navier Stokes equation. Therefore the LES equation is written as follows [26] -

$$\frac{\partial(\rho\tilde{u}_i)}{\partial t} + \frac{\partial}{\partial x_j}(\rho\tilde{u}_i\tilde{u}_j) = \frac{\partial}{\partial x_j}[\rho\nu\frac{\partial\tilde{u}_i}{\partial x_j} - \rho\tau_{ij}^{SGS}] - \frac{\partial\tilde{p}}{\partial x_i} + \rho g_i \tag{4.34}$$

where the term τ_{ij}^{SGS} is called as subgrid stress SGS which takes into account the effect of small vortices on large scale motion which is directly resolved on the grid [26].

$$\tau_{ij}^{SGS} = u_i u_j - \tilde{u}_i\tilde{u}_j \tag{4.35}$$

This τ_{ij}^{SGS} gives us a tensor field which cannot be directly computed and therefore requires closure models so that the system of equations is closed. These models are described in the section hereafter which are particular to our application in this work.

4.4.8 Subgrid Stress Model - Dynamic Smagorinsky SGS

The subgrid stress models account for the effect of smaller scales which get filtered out due to filtering operation of LES. Various subgrid models are available in OpenFOAM like Smagorinsky, Dynamic-Smagorinsky, DeardorffDiffStress, kEqn, WALE etc. In this work the Dynamic-Smagorinsky subgrid model is applied. It was essentially developed by Germano et.al. [75] from the original Smagorinsky model of 1963 [77] which is the most applied SGS models [26].

The Smagarinsky model approach is similar as the Boussinesq approach but instead of using turbulent kinematic viscosity ν_t, the subgrid viscosity ν_{SGS} is used here [26] [74].

$$\tau_{ij}^{SGS} - \frac{1}{3}\tau_{kk}^{SGS}\delta_{ij} \approx -2\nu_{SGS}\tilde{S}_{ij} \tag{4.36}$$

The subgrid viscosity ν_{SGS} is expressed by Smagorinsky in the terms of magnitude of strain rate tensor S_{ij} and square of a certain length l_s. This length can found from the mesh size Δ [26].

$$\nu_{SGS} = (C_s\Delta)^2|\tilde{S_{ij}}| \tag{4.37}$$

where

$$|\tilde{S_{ij}}| = \sqrt{2\tilde{S}_{ij}\tilde{S}_{ij}}\ ;\ \tilde{S}_{ij} = \frac{1}{2}\left(\frac{\partial\tilde{u}_i}{\partial x_j} + \frac{\partial\tilde{u}_j}{\partial x_i}\right)$$

here C_s is a constant of Smagorinsky which was first estimated by Lilly by striking a balance between production and dissipation of the turbulent kinetic energy. It was estimated at a value of 0.17. The CFD experience shows that lower values of this constant like 0.065 and 0.1 produce better numerical results when compared to the measurements [26].

The advantages of Smagorinsky model is that it is simple and stable. It also cost low computational power and depicts good accuracy in ideal conditions. There are some disadvantages too, like the laminar flow cannot be modelled, the solution is sensible to the grid, is purely dissipative and damping of pulsation is very strong. The Smagorinsky constant C_s is constant in space and time in Smagorinsky model while it is variable $C_s(x,t)$ in Dynamic Smagorinsky model which was introduced by Germano [26] [75] [76].

He introduced double filtering procedure designated as $\hat{u} = \hat{\hat{u}}$ with filter width set to double length of grid scale i.e 2Δ. Therefore by definition [26]

$$T_{ij} = u_i\widehat{u}_j - \widehat{u}_i\widehat{u}_j = \widehat{\widetilde{u_iu_j}} - \hat{\hat{u}}_i\hat{\hat{u}}_j \tag{4.38}$$

$$\hat{\tau}_{ij}^{SGS} = \widehat{\widetilde{u_i u_j}} - \widehat{\tilde{u}_i \tilde{u}_j} \tag{4.39}$$

A tensor L_{ij} is introduced which corresponds to the difference between equations 4.38 and 4.39

$$L_{ij} = T_{ij} - \hat{\tau}_{ij}^{SGS} = \widehat{\tilde{u}_i \tilde{u}_j} - \hat{\tilde{u}}_i \hat{\tilde{u}}_j \tag{4.40}$$

Using the Lilly procedure the dynamic constant is finally derived as follows -

$$C = max \left\{ \frac{M_{ij} L_{ij}}{2 M_{ij} M_{ij}}, 0 \right\} \geq 0 \tag{4.41}$$

where the subgrid kinematic viscosity is always positive i.e $\nu_{SGS} \geq 0$. The dynamic constant in equation 4.41 is limited to avoid numerical instability occurring due to energy backscattering [26].

CHAPTER 5

Grid-free - Computational Vortex method CVM

Contents

5.1 Vortex properties

The major players in a turbulent flow are the vortices. They are the fluid regions where flow revolves around an axis which can be straight or curved [94] [95]. The vortices are characterized by vorticity which is defined as the curl of the velocity.

$$\vec{\omega} = \nabla \times \vec{u} \tag{5.1}$$

Vorticity is solenoidal $\nabla\vec{\omega} = \nabla(\nabla \times \vec{u}) = 0$. The vortex line is defined as a line which is tangential to the local vorticity vector $\vec{\omega} \times \vec{dl} = 0$ [96] and in 3D, the vortex lines are closed. The velocity induced by a vortex defined as vorticity occupied per unit volume U is computed from the Biot-Savart law as follows [26] -

$$\vec{u}(x,t) = \frac{1}{4\pi} \int_U \frac{\vec{\omega} \times (\vec{x} - \vec{r})}{|\vec{x} - \vec{r}|^3} dU \tag{5.2}$$

The pressure at the vortex core is lowest near its axis and increases as we move outward from the core.

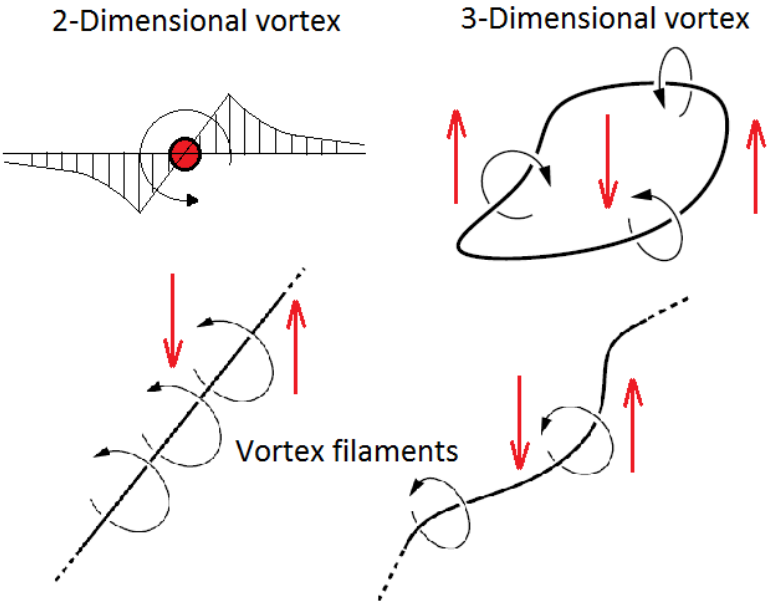


FIGURE 5.1: Vortices in 2-Dimensional and 3-Dimensional space

5.2 Vortex identification

Vortex identification contains different methods which define a particular function which can be calculated at different points in the domain and then classify each point whether being inside or outside of the vortex by setting some threshold on the evaluated values of the function. Most of the vortex identification criteria are Galilean invariant depending on gradient tensor $\nabla \vec{u}$ [97] [98]. Some of the widely used vortex identification criteria are - Lambda2 i.e λ_2 by Jeong and Hussian [99], LambdaCI i.e. λ_{ci} by Zhou et. al. [100], Δ by Chong et. al. [101] and Q by Hunt, Wray and Moin [102]. There are also some non-invariant methods like the Helicity method [103], Swirl parameter method [104], Maximum vorticity method [105], Eigenvector method [106] etc and other global methods like Predictor-corrector method [107], Streamline method [108] [109] etc. Vortex identification is a necessary step for grid-free as well as coupled methods to identify vortices at the interface before proceeding into the grid-free domain. Vortex identification also helps to visualise vortices in the grid based domain. Some of the criteria applied in our work are discussed below -

λ_2 criterion

The λ_2 criterion is one of the most famous criteria used for vortex identification. If $\vec{u}$ is the velocity field, then the gradient tensor for velocity can be written as follows -

$$\nabla\vec{u} = \begin{bmatrix} \frac{\partial u_x}{\partial x} & \frac{\partial u_x}{\partial y} & \frac{\partial u_x}{\partial z} \\ \frac{\partial u_y}{\partial x} & \frac{\partial u_y}{\partial y} & \frac{\partial u_y}{\partial z} \\ \frac{\partial u_z}{\partial x} & \frac{\partial u_z}{\partial y} & \frac{\partial u_z}{\partial z} \end{bmatrix} \tag{5.3}$$

This tensor can be decomposed into two separate tensors - one symmetric S and other anti-symmetric Ω expressed as follows -

$$S = \frac{(\nabla\vec{u}) + (\nabla\vec{u})^T}{2}, \ \Omega = \frac{(\nabla\vec{u}) - (\nabla\vec{u})^T}{2} \tag{5.4}$$

where S denote strain rate tensor and Ω denotes rotation rate tensor.

The tensor $S^2 + \Omega^2$ is of the order 3 and therefore has 3 eigenvalues λ_1, λ_2, λ_3 existing at every point of the domain, which are ordered in a descending order as $\lambda_1 \geq \lambda_2 \geq \lambda_3$. A region inside the vortex core is identified if $\lambda_2 < 0$. Locally, λ_2 signifies the measure of the excess rotation rate relative to the strain rate in a particular plane in an incompressible flow [97].

Q criterion

Another widely used criteria for vortex identification is the Q criteria. It is written as follows -

$$Q = \frac{1}{2}\left(||\Omega||^2 - ||S||^2\right) \tag{5.5}$$

where

$$||\Omega|| = tr\sqrt{\Omega\Omega^T}, \ ||S|| = tr\sqrt{SS^T}$$

Where S and Ω are same as defined in the equation 5.4. A region inside the vortex core is identified if $Q > 0$ i.e region where the second invariant of $\nabla\vec{u}$ is positive. Locally, Q signifies the measure of the excess rotation rate relative to the strain rate in all directions in an incompressible flow [97].

λ_{ci} criterion

The λ_{ci} is also called as Swirling strength. This criteria is used to identify regions of vortex at the interface between grid-based and grid-free domain in our coupled method. This criteria follows the Δ criteria. In the Δ criteria, the region inside the vortex core is identified where the velocity gradient tensor $\nabla\vec{u}$ has complex eigenvalues whereas in the λ_{ci} criteria, the region inside the vortex core is identified where the velocity gradient tensor $\nabla\vec{u}$ has imaginary part of complex eigenvalues [97]. Therefore it is subscripted as 'ci' i.e complex imaginary. Here the velocity gradient tensor is decomposed using real and imaginary eigenvectors in the following manner [110] -

$$\nabla\vec{u} = \begin{bmatrix} \vec{v}_r & \vec{v}_{cr} & \vec{v}_{ci} \end{bmatrix} \begin{bmatrix} \lambda_r & 0 & 0 \\ 0 & \lambda_{cr} & \lambda_{ci} \\ 0 & -\lambda_{ci} & \lambda_{cr} \end{bmatrix} \begin{bmatrix} \vec{v}_r & \vec{v}_{cr} & \vec{v}_{ci} \end{bmatrix}^T \tag{5.6}$$

where the real eigenvalue is λ_r and its corresponding eigenvector is $\vec{v}_r$ while the complex eigenvalues are $\lambda_{cr} \pm \lambda_{ci}$ and its corresponding eigenvectors are $\vec{v}_{cr} \pm \vec{v}_{ci}$. In a co-ordinate system traversed by 3 vectors [$\vec{v}_r$, $\vec{v}_{cr}$, $\vec{v}_{ci}$], the local flow seems to stretch or compress along the real vector axis $\vec{v}_r$ with λ_r as its stretching strength and swirl in the complex plane spanned by complex vectors $\vec{v}_{cr}$ and $\vec{v}_{ci}$ with λ_{ci} as its swirling strength [110]. The λ_{ci} criteria has some important advantages over the above two criteria since it not only identifies the vortex core but also identifies the strength and the local swirling plane [97].

5.3 Vortex Kinematics

The fluid flow can be characterized as Lagrangian or Eulerian depending on the reference system used. Some details of these both are described in the following sections.

5.3.1 Lagrangian characterization of the fluid flow

In Lagrangian characterization of the fluid flow, the properties of the fluid particles are defined with respect to the local frame of reference. It means, we follow the path of the fluid particle as it moves [111]. The identity of a fluid particle can be its initial position say (x_0, y_0, z_0) at time t_0. The kinematic behaviour of the fluid particle i.e. its displacement S, its velocity u and its acceleration a will be then the function of its identity [112] -

$$\vec{S} = f(x_0, y_0, z_0, t) \ , \ \vec{u} = f(x_0, y_0, z_0, t) \ , \ \vec{a} = f(x_0, y_0, z_0, t) \tag{5.7}$$

In Lagrangian characterization of the fluid flow, the mathematical laws can be drafted for each fluid particle [114].

5.3.2 Eulerian characterization of the fluid flow

In Eulerian characterization of the fluid flow, the properties of the fluid particles are defined with respect to the global frame of reference. It means, we concentrate on specific control volume as the flow passes through it [111]. The independent parameter can be the observation point which is fixed in time and space at a certain position say (x, y, z). As the fluid particle passes through this fixed point at time t, its kinematic behaviour can determined [112].

$$\vec{S} = f(x, y, z, t) \ , \ \vec{u} = f(x, y, z, t) \ , \ \vec{a} = f(x, y, z, t) \tag{5.8}$$

The Eulerian and Lagrangian characterization of the fluid flow are related as [113] -

$$\vec{u}(\vec{S}(x_0, y_0, z_0, t), t) = \frac{\partial \vec{S}(x_0, y_0, z_0, t)}{\partial t} \tag{5.9}$$

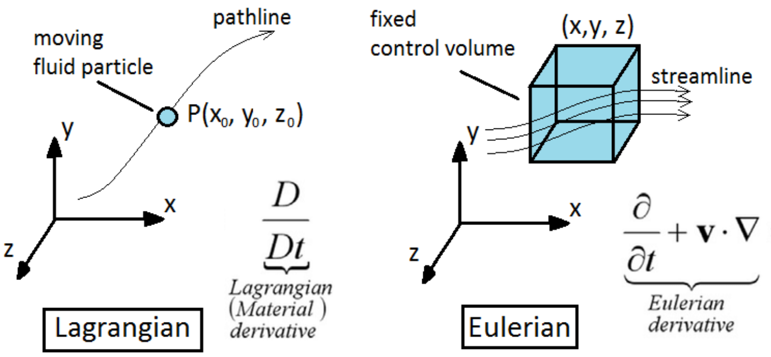


FIGURE 5.2: Lagrangian and Eulerian characterization of the fluid flow [115]

5.4 Vortex Dynamics

5.4.1 Vorticity transport equation

The vorticity transport equation describes the evolution of the vorticity $\vec{\omega} = \nabla \times \vec{u}$ of the fluid particle as it flows. The Navier Stokes equation reads as -

$$\frac{\partial \vec{u}}{\partial t} + (\vec{u} \cdot \nabla)\vec{u} = \vec{F} - \frac{1}{\rho}\nabla p + \nu \triangle \vec{u} \tag{5.10}$$

The vorticity transport equation can be obtained by taking curl of the Navier Stokes equation [116].

$$\nabla \times \frac{\partial \vec{u}}{\partial t} + \nabla \times (\vec{u} \cdot \nabla)\vec{u} = \nabla \times \left(\vec{F} - \frac{1}{\rho}\nabla p\right) + \nabla \times (\nu \triangle \vec{u}) \tag{5.11}$$

By applying the vector identities, the pressure term vanishes in an incompressible flow. While the other terms could be adjusted and derived to get the final version of the vorticity transport equation which can be written as follows [26] -

$$\frac{\partial \vec{\omega}}{\partial t} + (\vec{u} \cdot \nabla)\vec{\omega} = (\vec{\omega} \cdot \nabla)\vec{u} + \nu \Delta \vec{\omega} \tag{5.12}$$

By introducing material derivative it can be re-written as -

$$\frac{D\vec{\omega}}{Dt} = (\vec{\omega} \cdot \nabla)\vec{u} + \nu \Delta \vec{\omega} \tag{5.13}$$

In tensor form, it can be written as [117] -

$$\frac{\partial \omega_i}{\partial t} + u_j \frac{\partial \omega_i}{\partial x_j} = \omega_j \frac{\partial u_i}{\partial x_j} + \nu \frac{\partial^2 \omega_i}{\partial x_j \partial x_j} \tag{5.14}$$

The first term on the left hand side of the equation 5.12 i.e. $\frac{\partial \vec{\omega}}{\partial t}$ represents the unsteady term i.e the rate of change of vorticity with time while the second term i.e. $(\vec{u} \cdot \nabla)\vec{\omega}$ represents the convection of the vorticity from one point to another in the fluid. The first term on the right hand side of the equation 5.12 i.e. $(\vec{\omega} \cdot \nabla)\vec{u}$ represents the stretching of the vorticity due to velocity gradients existing in the flow while the second term i.e. $\nu \Delta \vec{\omega}$ represents the diffusion of vorticity.

Suppose that the flow over an airfoil is with uniform velocity $\vec{u}$ with no vorticity in the upstream. This uniformity of the flow changes due to the vorticity. The source of vorticity is the friction with the solid surface of the wing, where the no-slip condition generates vortices. These vortices get convected downstream and diffused into the main flow due to viscous effects[117].

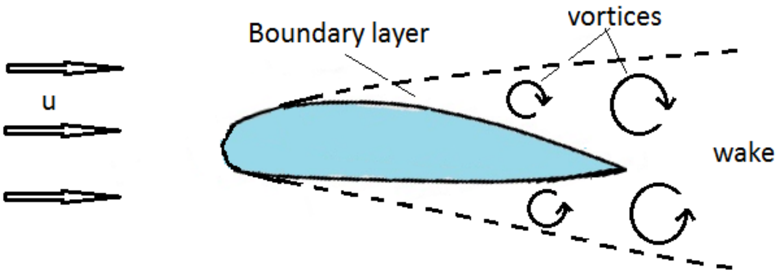


FIGURE 5.3: Vortices generated in the boundary layer close to the wing surface

5.4.2 Vortex amplification

The first term on the right hand side of the equation 5.12 i.e. $(\vec{\omega} \cdot \nabla)\vec{u}$ creates instability and could be critical for strong concentrated vortices at large Reynolds numbers like in our case. Suppose a vortex with its axis along the x-axis i.e. ω_x is positive. If this vortex is located in the region of fluid stretching i.e. where $\frac{\partial u_x}{\partial x}$ is positive, the the vortex stretching term $(\omega_x \frac{\partial u_x}{\partial x})$ is also positive. This leads to an increase in the vorticity ω_x. It means even without any viscous effects, the vorticity could grow to a very large value making the vortex more and more confined with an increase in the energy levels causing instability. But this term is balanced by the diffusion term which makes the vortex thicker. Diffusion counters the amplification [26]. Mechanisms like folding prevents the exponential increase of vorticity and helps to control the energy levels thus satisfying the energy conservation [118] [119].

5.5 Vortex methods

5.5.1 State of the Art

Many vortex method experts still use the traditional vortex method in a pure Lagrangian form ([138], [139], [140] and [141]) while some like Cottet, Winckelmans, Koumotsakos and Garakhani consider the vortex method only with coupling with grid based simulation within the Vortex in Cell (VIC) method. Prof. Mimeau presented a directional splitting scheme of such a method [142] which includes the following steps : [143] [144]

- Calculation of velocity from Poisson equation

$$\Delta\vec{u} = -\nabla \times \vec{\omega}, \tag{5.15}$$

- Fulfilment of the no slip boundary condition through the Brinkman penalization method

$$\frac{\partial\vec{\omega}}{\partial t} = \nabla \times (\lambda\chi(\vec{u}_b - \vec{u})), \tag{5.16}$$

- Amplification and rotation of vorticity

$$\frac{\partial\vec{\omega}}{\partial t} = (\vec{\omega} \cdot \nabla)\vec{u} \tag{5.17}$$

- Diffusion of vorticity

$$\frac{\partial\vec{\omega}}{\partial t} = \frac{1}{Re}\Delta\vec{\omega}, \tag{5.18}$$

- Convection of vortex elements

$$\frac{D\vec{\omega}}{Dt} = 0. \tag{5.19}$$

The grid based finite difference method is applied to first 4 steps whereas only the step (5.19) is treated in a pure Lagrangian way. Below we discuss pros and cons of such a procedure which is now a state of the art in vortex methods.

First, the advantage of this technique is that the computations of velocities are sufficiently accelerated due to application of the Poisson equation (7.1) and Fast Fourier Transformation (FFT) instead of direct summation using the Biot- Savart integral, especially when Fast Multipole Method (FMM) is used only for determination of boundary conditions. Moreover, it was shown that VIC is even faster than FMM [145]. The last conclusion is important for this work. The scalability of the VIC method including remeshing and consideration of boundary conditions is estimated as $O(n \log n)$ [146].

However, the biggest weakness of this approach is the application of uniform $\Delta_x = const, \Delta_y = const, \Delta_z = const$ Cartesian grids. This contradicts to latest tendencies in grid based method, like finite difference or finite volume ones, which are developing towards non-uniform and unstructured grids to handle flows with complex geometries. Prof. Cottet proposed to map the data from uniform grids to non uniform ones when the latter is necessary, for instance, to couple with grid based solvers. As it is known, mapping on unstructured grid is a big challenge from computational point of view and introduces additional errors. If the solution of (7.1) is of the fourth order the approximation should have the same order to keep the accuracy order of the overall procedure.

Second point of concern is the application of grid based methods to the solution of the equation (5.17). It is possible to treat the term $convect$ in a pure Lagrangian way, since the analytical expressions for smoothed velocity distribution $\vec{u}$ is available and the strain rate tensor $\partial u_i / \partial x_j$ can also be calculated analytically. The term $convect$ plays a key role in the vortex dynamics and generation of the turbulence. Due to amplification mechanism this term is responsible for development of singularity $|\vec{\omega}| \to \infty$ within a finite time in inviscid vortex flows. Therefore, a correct treatment of this term in a non- diffusive way should be the key point in numerical simulations in which this term can be a source of instability and lead to breakdown of computations. High artificial viscosity of grid based method helps to damp this instability whereas in pure Lagrangian method with low artificial viscosity it could become a big problem. One possible way to prevent the instability caused by vortex amplification is the incorporation of folding mechanism in some way into the vortex method. Due to the folding the stretched vortex filaments loss stability, create tangles and reduce their kinematic effect due to close approach of elements with opposite vorticity. This mechanism, predicted by Chorin [119], could be a reason why the kinetic energy can be kept constant despite of vorticity singularity. Within this work, in the simulation of the tip vortex evolution we have not met the problem of instability caused by the term $convect$ because the amplification is very weak for such flows. Translation and rotation of vortex elements play a dominating role for tip vortices.

The last step (5.19) is treated in a pure Lagrangian way. However, the grid based remeshing procedure is applied which consists in a redistribution of vortex elements among adjacent grid nodes. In last decade the remeshing became an inevitable procedure in many vortex methods.

As mentioned in [142] the Lagrangian treatment of the convection term $\frac{D\vec{\omega}}{Dt}$, combined with the remeshing operation (i.e., the redistribution of the particles, when required, to compensate for the Lagrangian distortion of the set of particles), leads to a method with accurate treatment of the convection (i.e., negligible dispersion and excellent stability properties). High order remeshing is attained when it is done with Monaghan's interpolation formula M_4' [112] which can be applied only on uniform Cartesian grids. It is claimed that the remeshing is non diffusive (s [145] and [146]).

Despite of the grid introduction the VIC method is classified as Lagrangian or semi-Lagrangian approach, since the vortices, i.e the terms $\partial\vec{\omega}_t + (\vec{\omega} \cdot \nabla)\vec{u}$, are tracked in the Lagrangian way. However, the loss of the most important advantages of pure Lagrangian methods, i.e. grid independence, raises the big question about the efficiency and competitiveness of VIC with respect to common grid based techniques. The dominating part of the vortex method applications is still done for very low and moderate Reynolds numbers in laminar mode. For instance, the flow simulation around a sphere presented in [142] is performed at $Re = 1000$. On contrary, our domain decomposition strategy is applied on Devenport tip vortex with a sufficiently high Reynolds numbers thus capturing the real turbulence behind the wing, since the vortex methods are implemented in the far stream of the flow.

5.5.2 Pure Lagrangian method

Pure Lagrangian method is a vortex method where the flow can be represented in the form of Lagrangian fluid particles carrying individual vorticity. The velocity of the vortex particles is computed from the vorticity and influence of other particles using Biot Savart law as stated in the equation 5.2. Each of these particles will then convect through the flow using its local velocity and its vorticity will get modified due to vortex stretching and viscous effects. This method does not require any grid, therefore is purely a grid-free method [120]. This property of Pure Lagrangian method gives it a huge advantage.

Suppose that there are N vortex particles in the domain, then the calculation of velocity using Biot Savart law for all these particles in a single time step would require $O(N^2)$ calculations. This is not suitable when we have a large number of vortex particles N in the domain. In-order to reduce this computational effort, the excess velocity calculations can be restricted by ignoring the influence of the neighbouring particles beyond a certain threshold distance [120]. This was also shown by Chorin et. al. [121] and used by Schlick et. al. in his molecular simulations [122]. This grid free method was also studied for three-dimensional flows by Leonard et. al [123], Knio and Ghoniem [124], Winckelmans and Leonard [125]. The evolution of a vortex sheet using this method was studied by Lozano et. al. [126].

5.5.3 Vortex-in-cell method

The vortex-in-cell termed as VIC method is one of the well known vortex method. In this method, the velocity is calculated using grid instead of direct calculation as in Pure Langangian Method. VIC method is essentially a particle mesh algorithm. Suppose we have vortex particles in the domain carrying the vorticity $\vec{\omega}$. In the first step the vorticities of these particles is mapped onto a temporary grid. Then the following Poisson equation is solved in order to calculate the streamfunction φ [120] [112]

$$\Delta\varphi = -\vec{\omega} \tag{5.20}$$

Once we have obtained the streamfunction, the velocity is calculated taking the curl of the streamfunction -

$$\vec{u} = \nabla \times \vec{\omega} \tag{5.21}$$

Thereafter, the velocity values from the grid nodes are interpolated back to the particle positions. After this step, the particles move with time step Δt as [127] -

$$\Delta\vec{x}_p = \vec{u}_p \Delta t \tag{5.22}$$

Different methods like Runge Kutta, Euler, Euler corrected etc could be used to achieve this. In this way, the Vortex-in-Cell method works. Solving of the Poisson equation in VIC method is crucial. Different FFT (Fast Fourier Transform) fast solvers or multi-grid methods could be used to accelerate the calculation of Poisson equation [120].

The 2D VIC method was first applied by Christainsen et. al. [128] [129] [130] to solve the two-dimensional Kelvin-Helmholtz instability while the 3D VIC method was first applied by Couet et. al. [131] to study the evolution of vortex rings. The vortex-in-cell method for the viscous diffusion was used by Cottet et. al. [132] [133], by solving for the stretching and velocity on the grid while convection was discretized as Lagrangian [134] [135]. Zawadzki and Aref [136] also studied the phenomena of vortex ring collision using VIC method.

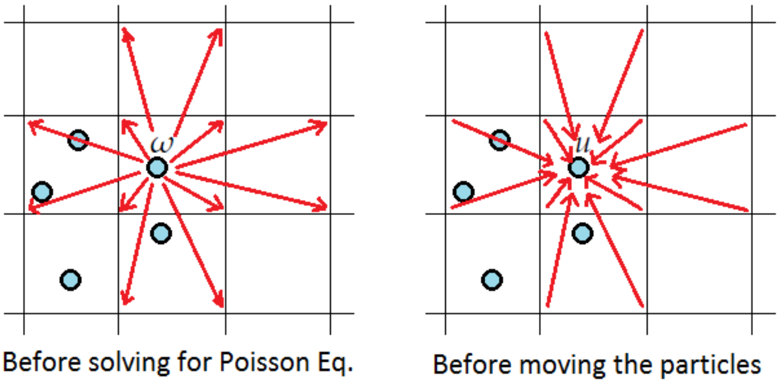


FIGURE 5.4: Interpolations in Vortex-in-cell method [137]

CHAPTER 6

Coupled - Domain Decomposition method

Contents

6.1 State of the Art

The vortex methods discussed in the previous chapter come along with many advantages like -

- Minimum artificial viscosity
- Operator can be calculated analytically
- No constraints with respect to stability (CFL criterion)
- Easy interpretation of results

but there are some disadvantages as well like -

- Difficult to satisfy the Boundary Conditions on wall e.g no-Slip BC
- Diffusion resolution, Big overlapping, Big number of particles

While the grid based methods are well known to precisely incorporate boundary conditions at the wall and accurately solve the near wall flows. But they face a serious disadvantage in the far field due to discretization errors as discussed in Chapter

2. One way to combine the advantages of both the methods and nullify the disadvantages is the use of domain decomposition technique as proposed by Cottet in his book [112]. Close to the wall grid methods are applied while far away from the wall grid free vortex methods are applied.

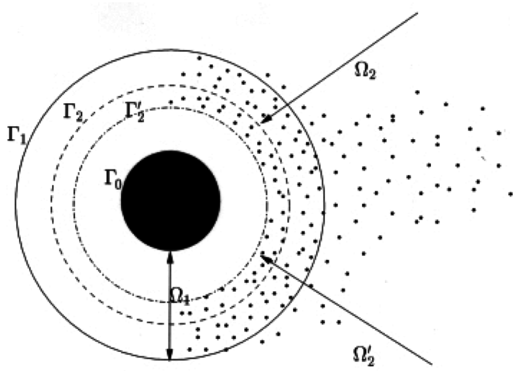


FIGURE 6.1: Domain-decomposition method as illustrated by M. Ould-Salihi et. al. (2000) [147]

M. Ould-Salihi et. al. (2000) [147] and his colleagues discussed in their work on the particle grid domain decomposition by blending finite difference and vortex methods. Figure 6.1 shows two overlapping domains Ω_1 and Ω_2 as presented in the paper of M. Ould-Salihi. Ω_1 is for finite difference while Ω_2 is for vortex method. These domain overlap eachother in $\Omega_1 \cap \Omega_2^{'}$. The boundary condition on Γ_0 is Dirichlet zero while the boundary conditions on Γ_1 and Γ_2 is calculated. The Poisson equation $-\Delta\psi = \vec{\omega}$ is evaluated by finite differences in Ω_1 to give streamfunction and thus velocity. Vorticity is assigned on Γ_1 by the particles of the domain Ω_2 since $\Gamma_1 \subset \Omega_2$. To update the circulations of the particles, M. Ould-Salihi proposed geometrical constraint as $d(\Gamma_2, \Gamma_2^{'}) \geq u_{max}\Delta t + \rho$, where ρ is width of the kernel used in PSE scheme. They proposed the following algorithm for velocity-pressure formulation in the finite difference domain, given that u^n in grid domain and ω^n in grid free domain is known [147]

- using Poincare formula calculate the particle velocities in Γ_2
- differentiate u^n on the grid, interpolate ω on particles in the overlapping region
- In Ω_2 update the particles ω^{n+1}
- Calculate G^n, then $u^{n+1}\nu$ on Γ_1

Another method for computing particle velocities could be through the Biot Savart law. The smooth transfer of vorticity between two domains was proven by computing a flow a over a backward-facing step with $Re = 355$. They observed no fluctuations in the vorticity despite the interface cutting through the many recirculation zones. The fact of smooth transfer of vorticity was also verified by considering a flow past cylinder with $Re = 550$ and 3000. The consistency of interface conditions was also proven by using domain decomposition algorithm based on velocity-pressure formulation at $Re = 1000$. Thus domain decomposition method could be more flexible in 3-dimensional flow [147]. Our domain decomposition technique is a novel approach based on coupling Finite Volume Method domain and Grid free vortex method domain for resolution of wing tip vortices.

Apart from the wing tip vortex, coupled method can also be applied for rotating tip vortices. One of such application was presented during ICVFM 2016 by Winckelmans and his group [148] who applied VIC for calculation of vertical axis wind turbines (VAWT). The vortex wake shed from a blade of turbine interact with blades downstream like in case of rotor craft. Proper resolution of tip vortices is then very important to predict unsteady loadings on blades in wake. Wind Turbines are now considered as a very promising application for vortex methods for which their advantages can play a deciding role to predict blade wake interaction.

6.2 Description of the coupled method

Grid based methods is a very efficient and well developed tool to resolve the turbulent and laminar boundary layers close to the body. On the contrary, the particle based methods have many difficulties with formulation of boundary conditions and smooth representation of thin boundary layer. Therefore there are not much applications of vortex methods for real three dimensional configurations at high Reynolds numbers. From the other side, as shown above, the grid based techniques have a substantial artificial numerical viscosity in the wake resulting in a non-realistic damping of vortex structures including tip vortices. A natural way to escape these difficulties and to overcome disadvantages of grid based and particle based techniques is the application of domain decomposition. Close to bodies a grid based method is applied whereas far from the body a grid free ones. In this work we use the domain decomposition procedure originally developed by Cottet 's group (see [112], sec. 8.3.2 and [147]) to improve the resolution of tip vortices by coupling the

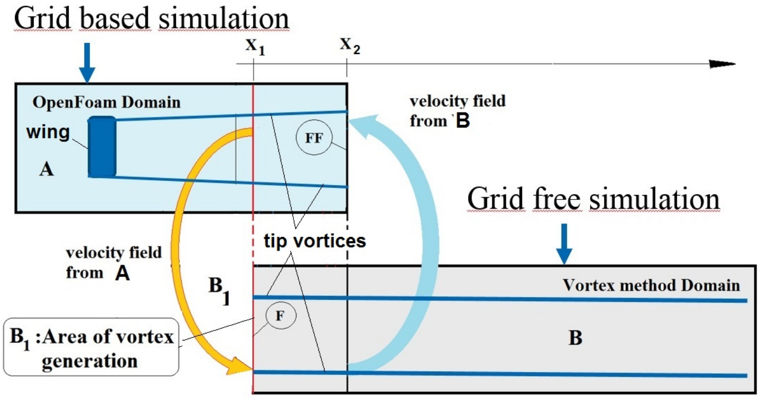


FIGURE 6.2: Illustration of the domain-decomposition method for the tip vortex problem

vortex method and the OpenFOAM code. The domain is decomposed into the upstream (zone A) and downstream (zone B) sub-domains (Fig. 6.2). In the zone A the OpenFOAM grid based simulation is used, whereas the flow in the zone B is handled using the vortex method. The outlet conditions for the A-domain solution are taken from the B-domain by direct calculation of velocities induced by vortex elements located in B. The vorticity is calculated at the interface and flows into the domain B where it is handled by a vortex method. The irrotational and rotational components of velocity $\vec{w}$, induced by the sub-domain A (i.e. induced by the wing and the part of the tip vortex in A) in sub-domain B at any point $\mathbf{x}$ are calculated from the Poincare identity (see formula 4.3.1. and 4.3.2 in [112]):

$$\vec{w}(\vec{x}) = \frac{1}{4\pi}\int_F \frac{\vec{\mu}(\vec{r})\times\vec{r}}{r^3}dS(\vec{r}) + \frac{1}{4\pi}\int_S \frac{\sigma(\vec{r})\vec{r}}{r^3}dS(\vec{r}) + \vec{U} \tag{6.1}$$

where $\vec{\mu} = 2\vec{n}(\vec{r})\times\vec{u}_\tau, \sigma = 2u_n$, $\vec{u}_\tau$ and u_n are respectively the normal and tangential perturbation (excluding $\vec{U}$ contributions) velocities at the interface F at $x = x_1$, $\vec{U}$ is the wing speed, $\vec{n}$ is the normal vector to F and $\vec{r}$ is the radius vector between points at F and the point $\vec{x}$. The Schwartz alternating algorithm is applied to match the solutions in A and B at the interface $x = x_1$. The vortex elements move and change their strength according to trajectory and vorticity transport equations (5.17) and (5.19). The particle strength exchange (PSE) method (see, for instance, [112]) is applied to take the diffusion effect into account (Eq. 5.18). In this form the vortex method runs in a DNS mode. Three following procedures were implemented to perform simulations within the vortex subdomain B:

- Pure Lagrangian Simulation,
- Vortex-in-Cell VIC Method with remeshing at each time instant,
- Vortex-in-Cell VIC Method without remeshing.

The results are obtained using the vortex elements proposed by Winckelmans [149] which induce the velocity at any point $\mathbf{x}_i$ according to formula:

$$\vec{u}_i(\vec{x}_i) = \sum_{j=1}^{M} \frac{\vec{\gamma}_j \times \vec{x}_{ij}}{4\pi\sigma_j^3}\frac{\rho + 5/2}{(\rho+1)^{5/2}}, \tag{6.2}$$

where $\vec{x}_{ij} = \vec{x}_i - \vec{x}_j$ and $\rho = |\vec{x}_{ij}|^2/\sigma_j^2$. σ_j is a size of the vortex element which was taken as $\sigma = \beta\Delta$ where Δ is the size of the grid at the outlet of the A subdomain. The factor $\beta = 2$ secures the overlapping between elements. Remeshing procedure for determination of vorticity in VIC method and for redistribution of vortices is based on the M_4' interpolation formula proposed by Monaghan (see [112], page 229).

6.3 Interface matching

Matching the velocity at the interface between grid and vortex method zones.

The domain decomposition method for pure Lagrangian simulation is extended to introduce an additional procedure for matching solutions in different regions. When the discrete particles are used in the zone B, the most natural condition to set the vortex element strength is

$$\vec{\gamma}_j = \vec{\omega}_{OF}(\vec{x}_j)Vol_j \tag{6.3}$$

where $\vec{\omega}_{OF}(\vec{x}_j)$ is the vorticity of the grid simulation at point $\vec{x}_j$ and $Vol_j = \Delta^3$ is the volume of cell within which the vorticity is replaced by the $j-th$ vortex element. In what follows the index OF stands for quantities of the grid based solution. When the resolution grows $\Delta \to 0$ the vortex elements with the strengths (6.9) induce the velocity $\vec{u}$ which is the same as the grid solution $\vec{u}_{OF}$. At moderate Δ for strong concentrated vortices there is a big discrepancy between $\vec{u}$ and $\vec{u}_{OF}$, namely $\vec{u}_{OF} > \vec{u}$. To match the velocities at the interface an additional adaption is performed at each time instant using the condition of matching between vorticity induced by vortex elements and vorticity of the grid solution

$$\nabla \times \vec{u}_{OF} = \nabla \times \vec{u} \tag{6.4}$$

From experience it is better to use the condition

$$\nabla \times \vec{u}_{OF}(\vec{x}_i) = \nabla \times P[\vec{u}(\vec{x}_i)] = \nabla \times P[\sum_{j=1}^{M} \frac{\vec{\gamma}_j \times \vec{x}_{ij}}{4\pi\sigma_j^3} \frac{\rho + 5/2}{(\rho+1)^{5/2}}], \tag{6.5}$$

where P stands for the projection of velocities $\vec{u}$ onto the grid. The equations with unknowns $\vec{\gamma}_j$ (6.5) can be solved as a system of $3M \times 3M$ linear equations. However, it is a very consuming procedure. That why it is projected (6.5) on x-axis, since the vector $\vec{\gamma}$ has a dominant component along x-axis, and use the iteration process:

$$\alpha_i^{(m)} = \alpha_i^{(m-1)} + \varepsilon(\nabla \times \vec{u}_{OF}(\vec{x}_i) - \nabla \times P[\sum_{j=1}^{M} \frac{\vec{\gamma}_j \times \vec{x}_{ij}}{4\pi\sigma_j^3} \frac{\rho + 5/2}{(\rho+1)^{5/2}}])\vec{i}, \tag{6.6}$$

Once the iteration process is converged, the strengths are updated:

$$\vec{\gamma}_j = \alpha_j^{(m)}\vec{\gamma}_j \tag{6.7}$$

This method, which is very similar to the Beale's iterative method (see [112] pages 208-209), has a very good convergence at moderate overlapping β. At large β the accuracy of approximation sufficiently degrades. This is the reflection of the singular matrix problem when the radial based functions with a large overlapping are used for approximation.

Within framework of the Vortex- in Cell (VIC) method this procedure is not necessary since the formalism is directly based on vorticity. The condition $\vec{\omega}(x = x_1) = \nabla \times \vec{u}_{OF}(\vec{x}_i)$ is imposed on the interface between A and B zones. The grid used is uniform with the size $\Delta_{x,y,z} = \Delta$. Velocity field necessary for inlet conditions at the interface between OpenFoam and vortex method domain was mapped onto the uniform grid. The ordinary differential equations describing the particle motion are integrated using the predictor-corrector or Euler corrected method with the trapezoidal rule. The time step Δt is chosen so that the particles paths close to the interface is around Δ, i.e. $\Delta t = min(\Delta/(U + u_x))$.

6.4 Method algorithm

The coupled method described in the above section is primarily simulated in a series of steps. First of all, a converged field is obtained in the grid domain-A by using grid-based methods in OpenFOAM as described in the Chapter 3 and 4. Then the grid domain is coupled with a grid free domain at the outlet with an interface by introducing a new boundary condition which will be described in details in the next section. The vortices are now identified at the interface x1 as in Figure 6.2. The influence of the upstream conditions is calculated using Poincare identity. Now the vortices are convected in domain B. As they reach a certain distance x2, the boundary condition for OpenFOAM domain is obtained. Again with this boundary condition, a flow is calculated in domain A. This is repeated until the match of solutions at the interface before going to the next time step. This is the loop of Schwarz iteration. The algorithm can be visualised in the following way -

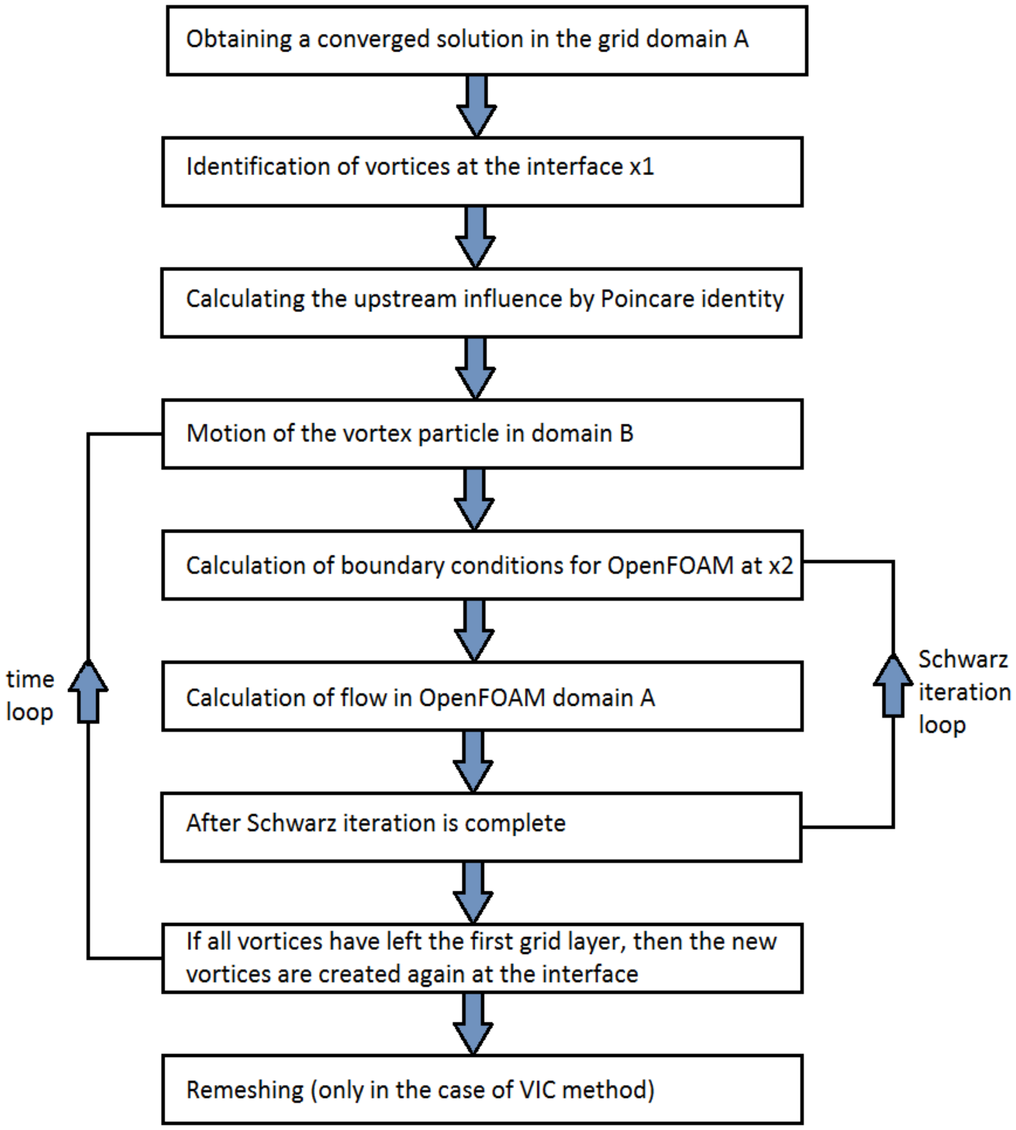

Obtaining a converged solution in the grid domain A
Identification of vortices at the interface x1
Calculating the upstream influence by Poincare identity
Motion of the vortex particle in domain B
Calculation of boundary conditions for OpenFOAM at x2
Calculation of flow in OpenFOAM domain A
After Schwarz iteration is complete
If all vortices have left the first grid layer, then the new vortices are created again at the interface
Remeshing (only in the case of VIC method)
time loop
Schwarz iteration loop

FIGURE 6.3: Flowchart of vortex code

6.5 Description of algorithm in OpenFOAM

The coupled method is in the form of a vortex code which is originally developed by Dr. Kroeger at our faculty. The vortex code is written to suit the OpenFOAM environment i.e. with the use of advanced C++ object oriented programming.

Overview of the OpenFOAM vortex code

The vortex code is implemented as a custom i.e. new boundary condition which switch on's the connection between the grid domain and the grid free domain and enables calculations in the vortex domain as well. The boundary condition should be specified in the following format -

```
// Custom boundary condition for coupling

outlet
{
   type                 schwarzCoupled;
   distanceIntoInterior 0.1;
   setUMean             yes;
   umean                (38.637 0 0);
   runUncoupled         yes;
}
```

Here, the type of boundary condition is *schwarzCoupled*, the parameter *distanceIntoInterior* specifies the location of the interface away from the outlet of the OpenFOAM domain, the parameter *umean* is used to specify the mean velocity of the flow while the parameter *runUncoupled* is used to switch between two types of simulation namely -

- 1-way coupled simulation - Here only the influence of OpenFOAM domain onto vortex domain is considered and the vice versa is switched OFF.
- 2-way coupled simulation - Here the influence of OpenFOAM domain onto vortex domain and the influence of vortex domain back onto OpenFOAM domain is considered.

The vortex code is compiled with the help of two main directories namely *vortoncloud* and *schwarzCoupledFvPatchVectorField* with the support of their respective headers. The compilation is targeted to create a new OpenFOAM library *libschwarzCoupled.so* which contains the information of the custom boundary condition to be applied. Therefore, a pre-processing step for a coupled simulation essentially involves three important steps as follows -

- Existence of a converged and accurate grid solution at the interface.
- Inclusion of custom OpenFOAM library *libschwarzCoupled.so* in the solution control sub-directory.
- Modification of custom boundary to *schwarzCoupled* at the outlet of the grid domain.

The vortex code is constructed within the framework object oriented programming i.e. with the use of class and objects. At first, a class *schwarzCoupledFvPatchVectorField* is inherited from a class-template of OpenFOAM i.e. *fixedValueFvPatchField<vector>*. This inherited class is then developed further to meet the algorithm requirements. Some *protected* members are declared within the inherited class like distanceIntoInterior, UName, setUMean, runUncoupled and umean, which are the variables which help in the definition of the boundary condition. These protected members are accessible in the class that defines them and in classes that inherit from that class. The objects in the inherited class are initialized using many constructors which are a special member function of the class with the class name itself. They will automatically be called when the respective objects are created. An example of a constructor from the class *schwarzCoupledFvPatchVectorField* is shown below -

```
// Construct from patch and internal field(here a vector field)

schwarzCoupledFvPatchVectorField
(
   const fvPatch&;
   const DimensionedField<vector, volMesh>&
);
```

Thereafter, these constructors are declared with various member variables and member functions within the scope of *schwarzCoupledFvPatchVectorField*. Some of the member variables are initialised to user-provided values like *Sortout*, *Poincare limit*, *Ameasure*, λ_{CI}*Threshold* and *Overlapping ratio*. Some of the member functions are also initialised to setup the initial field and update the fields. Some of the member functions are used through pointers to carry out Poincare calculations, vortons creation, storing particle positions, moving the particles and viscosity calculations. All of the above refers to the directory *schwarzCoupledFvPatchVectorField*.

Hereafter, we refer to the directory *vortoncloud*. The *namespace schwarzCoupled* is nested inside the *namespace Foam* which the default OpenFOAM *namespace*. A nested *namespace* has been used to prevent name conflicts in the large code. Various classes are now inherited within the scope of nested *namespace schwarzCoupled* like class *vorton*, class *vortonCloud* and class *interface-FVM-Vortex*. In the class *vorton*, member variables are declared and defined for the properties of vortex particles like their vorticity, volume, radius. This information is collected into an array and is aliased so that it can be referred to with an identifier as *VortonList*. In the class *interface-FVM-Vortex* which inherits from class *vectorField*, member variables are declared and defined for the velocities of Poincare. Also, different member functions are declared to carry out Poincare calculations, update face velocities etc. In the class *vortonCloud* which inherits from class *VortonList* and class *cloud*, member variables are declared and defined for the locations of the particles and member functions are declared for new vortons creation, storing initial positions, moving the particles, viscosity calculations etc. Also here various objects are declared to carry out tasks for calculation of velocity induced due by Poincare i.e. upstream conditions, influence of vortons of the same cloud, influence of vortons of other clouds at all the location of vortons.

Finally in the main directory, first, λ_{ci} is calculated using a member function within the scope of *Foam*. Further in the scope of *schwarzCoupled*, various objects are defined from the class *vorton*, class *interface-FVM-Vortex* and class *vortonCloud*. They are given below

1. Three objects are defined from the class *vorton* to undertake the functions as below

- to calculate the properties of vortons like the vorticity, location, volume and radius
- to write this data of vortons into an array
- to calculate the velocity induction by Biot Savart law

2. Four objects are defined from the class *interface-FVM-Vortex* to undertake the functions as below
    - to find the minimum grid size at the inlet
    - to calculate velocity induction by Poincare identity
    - to update the final velocity field
    - to write debug of all the operations.

3. Six objects are defined from the class *interface-FVM-Vortex* to undertake the functions as below
    - to generate a cloud of existing vortons
    - to find the nearest vorton location from the interface
    - to generate new vortons
    - to do viscosity calculations
    - to store current positions of the vortons
    - to move the vortons with time

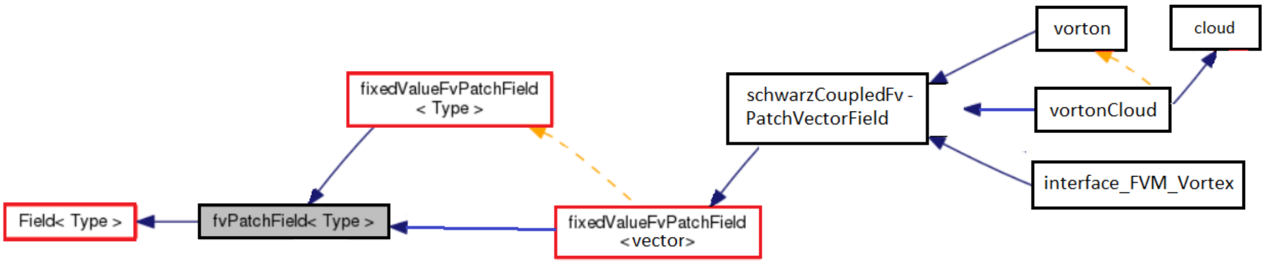

vorton
cloud
fixedValueFvPatchField < Type >
schwarzCoupledFv - PatchVectorField
vortonCloud
interface_FVM_Vortex
Field< Type >
fvPatchField< Type >
fixedValueFvPatchField <vector>

FIGURE 6.4: Class Inheritance : UML diagram

Further Developments in the OpenFOAM vortex code

The vortex code was improved further on 3 very important parameters - stability, numerical convergence and interface matching. The performance of the vortex code is sensitive to the inlet boundary condition. It was estimated during our coupled method simulations, that a physical vortex at the interface aids to attain smooth numerical simulation. For the best result, it is recommended to obtain a converged and validated physical vortex at the interface by prior simulation of various grid based methods. A wise strategy has been adopted to apply the vortex code to get proper validated results. This strategy will be discussed in the Solver Settings section of

Chapter 7. On the other hand, the convergence of the code solver is sensitive to the wise choice of numerical schemes, solver parameters, relaxation factors and residual controls. This will be detailed discussed in the Solver Settings section of Chapter 7.

During computations, a difference in the value of tangential velocity was observed when the new vortex elements are generated at the interface x=x1. This happens due to small errors arising in the process of interface matching iteration. To further optimize the method, a simple scaling algorithm was proposed. Within this algorithm, the strength of the vortex particles is updated based on the ratio between the vortex strength near the interface and the far field. This ratio is then used as a scaling factor during the generation of the discrete vortex particles at the interface.

$$Factor = \frac{Umax_{OF}}{Umax_{VM}} \tag{6.8}$$

$$\vec{\gamma}_j = \vec{\omega}_{OF}(\vec{x}_j) Vol_j * Factor \tag{6.9}$$

This improved the accuracy of results and solved the existed problem. The scaling algorithm is achieved with 3-step procedure in grid free domain. The necessary member variables were declared within the class *vortonCloud* for array definitions of positions of vortons and velocities. In the first step, the maximum tangential velocity is calculated at the interface. This is implemented within the object of *vortonCreation* with the help of a member function which is summarised in Figure 6.5.

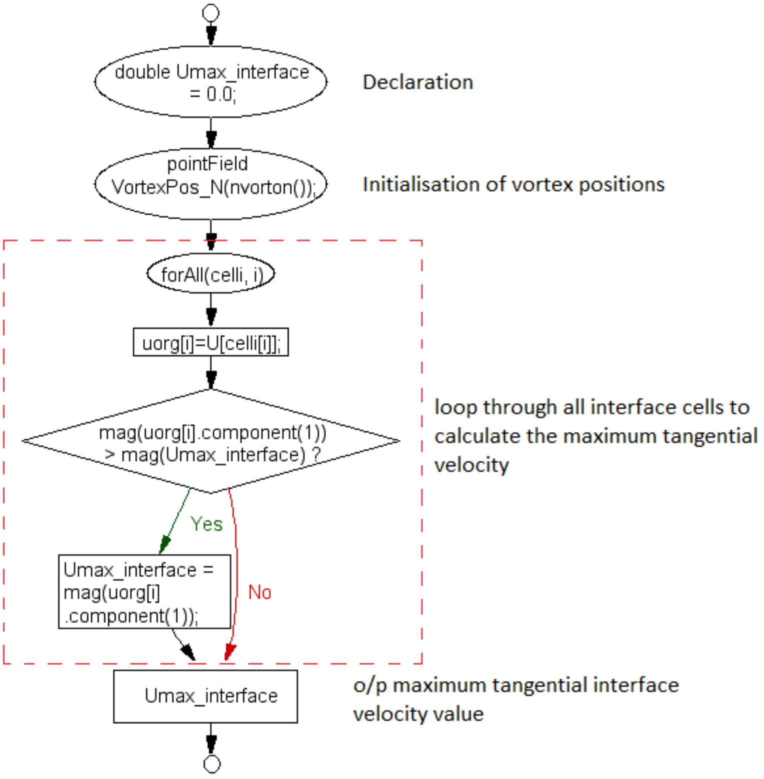

double Umax_interface = 0.0;
Declaration
pointField VortexPos_N(nvorton());
Initialisation of vortex positions
forAll(celli, i)
uorg[i]=U[celli[i]];
mag(uorg[i].component(1)) > mag(Umax_interface) ?
loop through all interface cells to calculate the maximum tangential velocity
Yes
No
Umax_interface = mag(uorg[i] .component(1));
Umax_interface
o/p maximum tangential interface velocity value

FIGURE 6.5: Calculation Diagram of maximum tangential velocity at the interface

In the second step, the maximum tangential velocity at far field is calculated. This is implemented within the object of *vortonCreation* with the help of a member function which is summarised in Figure 6.6. Here, a array captures all the far-field vortices at a specified distance interval away from the interface which is further passed through a sorting process before calculating velocities of vortons. Finally, the maximum tangential velocity at the far-field is obtained.

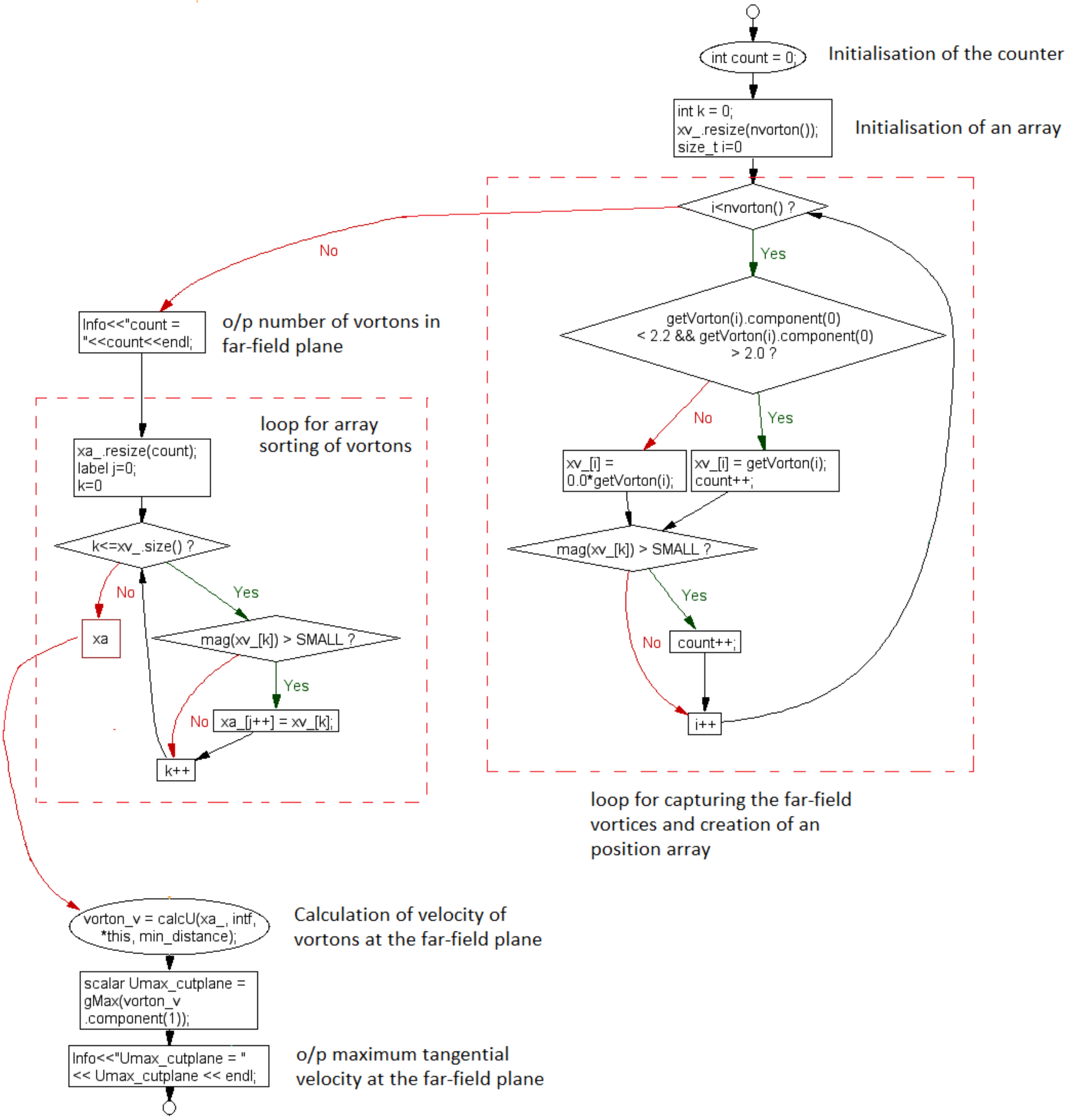

int count = 0;
Initialisation of the counter
int k = 0;
xv_.resize(nvorton());
size_t i=0
Initialisation of an array
i<nvorton() ?
No
Yes
getVorton(i).component(0) < 2.2 && getVorton(i).component(0) > 2.0 ?
No
Yes
xv_[i] = 0.0*getVorton(i);
xv_[i] = getVorton(i); count++;
mag(xv_[k]) > SMALL ?
Yes
No
count++;
i++
loop for capturing the far-field vortices and creation of an position array
Info<<"count = "<<count<<endl;
o/p number of vortons in far-field plane
loop for array sorting of vortons
xa_.resize(count);
label j=0;
k=0
k<=xv_.size() ?
No
Yes
xa
mag(xv_[k]) > SMALL ?
Yes
No
xa_[j++] = xv_[k];
k++
vorton_v = calcU(xa_, intf, *this, min_distance);
Calculation of velocity of vortons at the far-field plane
scalar Umax_cutplane = gMax(vorton_v .component(1));
Info<<"Umax_cutplane = " << Umax_cutplane << endl;
o/p maximum tangential velocity at the far-field plane

FIGURE 6.6: Calculation Diagram of maximum tangential velocity at far-field

In the third step, the interface factor is calculated and applied to update the strength of new vortons while they are generated. This was also implemented within the object of *vortonCreation* with the help of a member function which is summarised in Figure 6.7.

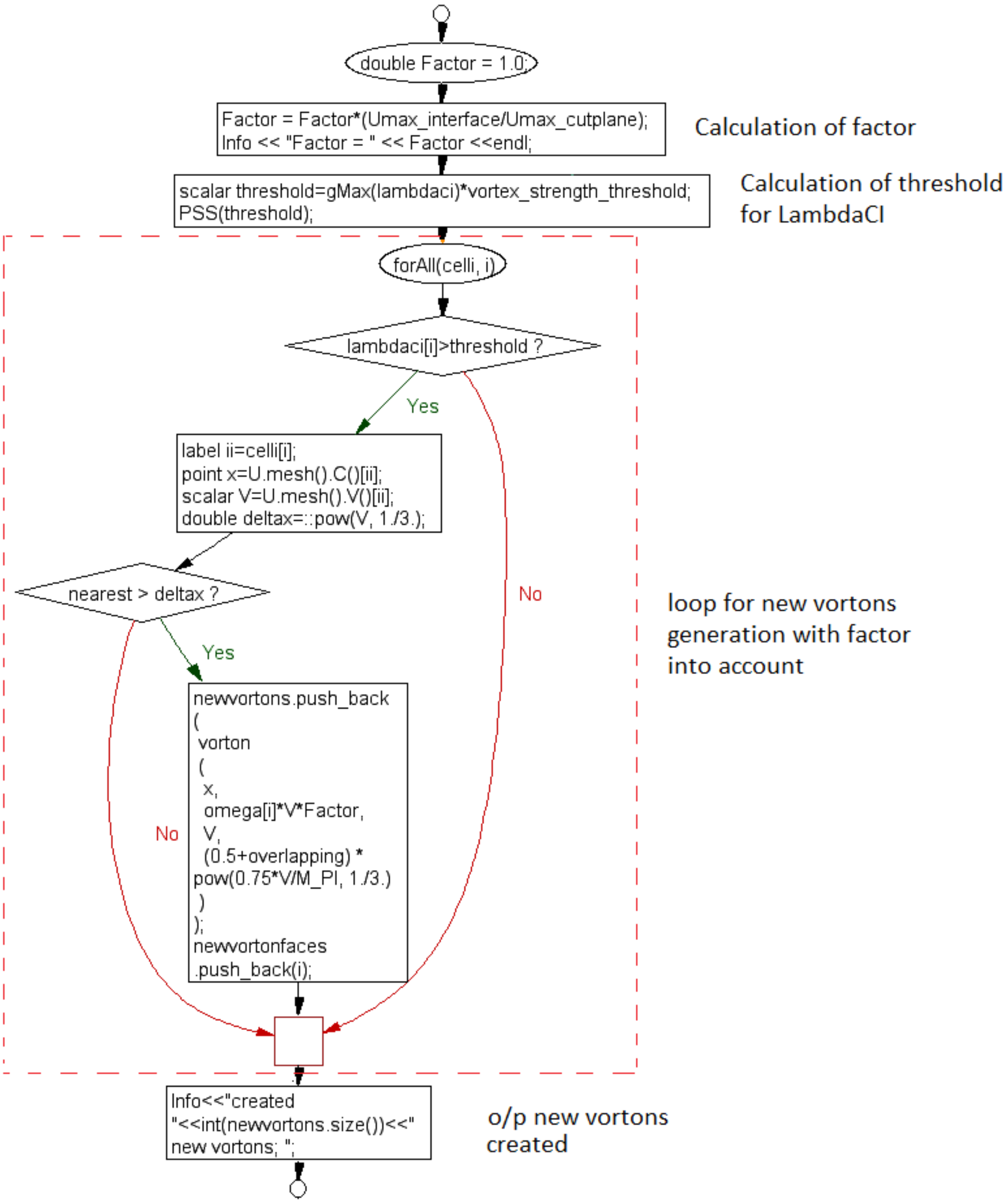

double Factor = 1.0;
Factor = Factor*(Umax_interface/Umax_cutplane);
Info << "Factor = " << Factor <<endl;
Calculation of factor
scalar threshold=gMax(lambdaci)*vortex_strength_threshold;
PSS(threshold);
Calculation of threshold for LambdaCI
forAll(celli, i)
lambdaci[i]>threshold ?
Yes
No
label ii=celli[i];
point x=U.mesh().C()[ii];
scalar V=U.mesh().V()[ii];
double deltax=::pow(V, 1./3.);
nearest > deltax ?
Yes
No
newvortons.push_back
(
vorton
(
x,
omega[i]*V*Factor,
V,
(0.5+overlapping) *
pow(0.75*V/M_PI, 1./3.)
)
);
newvortonfaces
.push_back(i);
loop for new vortons generation with factor into account
Info<<"created "<<int(newvortons.size())<<" new vortons; ";
o/p new vortons created

FIGURE 6.7: Calculation Diagram of interface factor and its influence in new vortons generation

The final results obtained by grid based, grid free and coupled methods will be discussed in the upcoming Chapters 7 and 8.

CHAPTER 7

Validation of methods for Steady Wing Case

Contents

Most of the parts of this Chapter are cited from our publication in International Journal of Ocean Engineering [7].

7.1 Test Case

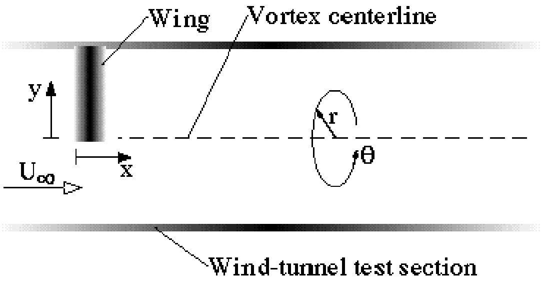


FIGURE 7.1: Schematic view of the wind tunnel test section, from Devenport Experiment [2]

This study targets to validate CFD models using the benchmark test case thoroughly studied in wind tunnel measurements presented in Devenport et al. (1996) [2]. The basic setup of the experiment is shown in Figure 7.1. During the experiment, the evolution of a tip vortex generated by a rectangular wing with the NACA 0012 profile was studied. The wing with a blunt tip has the following dimensions: a span of 0.879 m and a chord length of c = 0.203 m. The wind tunnel has a quadratic test section of 1.83 m width and 7.33 m length. The wing was attached to the wind tunnel wall at half the height of the cross section. Devenport et al. carried out the experiments in the Virginia Tech Stability Wind Tunnel located at the Virginia Polytechnic Institute and State University. The velocity was measured using a four-sensor hotwire probe.

Devenport et al. provide the most extensive data for the Reynolds number $Re = 530000$ (based on the chord length) and for 5° angle of attack. There, the vortex is tracked downstream until 30 times the chord length behind the leading edge of the wing. Experimental data is provided for axial and tangential velocity profiles through the vortex core and for turbulence properties. The target of the simulation approaches used in this work is to predict the evolution of tip vortices until large distances downstream. For this purpose the test case of Devenport et al. is more informative than other experiments at which the tip vortices are tracked only in the near wake of wings.

Besides, another advantage of this wind tunnel data is the proper correction for the vortex wandering motion. This slow side-to-side movement is usually observed for wind tunnel generated tip vortices. Without a proper correction of this effect, the experimental data would suggest an increased vortex decay. The way how the experimental data is corrected is presented in detail in [2]. Feder and Abdel-Maksoud showed in [150] that wandering did not occur within the simulation. Their observation was based on the Hexpress mesh with 6.0 M cells and the SA-DDES turbulence model. As no wandering was observed within the simulations, the corresponding correction is considered not necessary.

7.2 Numerical Setup

7.2.1 Geometrical Setup

The coordinate system is set according to experimental setup [2] and displayed in Figure 7.2. The origin is placed at the leading edge on the wing tip. The x-axis points downstream and the y-axis points along the wing. The plane to which wing is attached is considered as a symmetry plane. This is done to save the computational effort. The computational domain is shown in Figure 7.2. The inlet boundary is located at x/c = -7.4 whereas the outlet at x/c = 46.7. The boundaries with y- and z-normals denote the walls of the wind tunnel's test section. The size of the domain's cross section corresponds to the measuring section of the wind tunnel and the location of the wing in the cross section is identical to the experimental one.

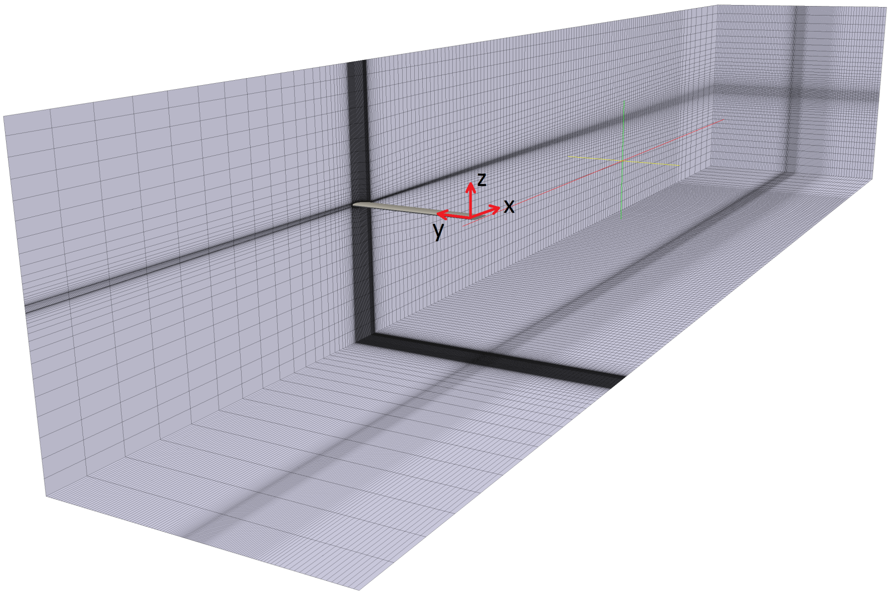


FIGURE 7.2: Computational Domain

7.2.2 Grid Setup

Two broad kinds of grid refinement procedures are adapted in this work - Static grids and Dynamic/Adaptive grids. The static grids are further classified into the High-Re grid without the near wall refinement and the Low-Re grid with the near wall refinement.

7.2.2.1 Static Grids

i. High-Re grid

Three meshes of different resolutions are generated in ANSYS ICEM CFD. The first mesh with 2.7 million of cells is initially coarse in the whole computational domain, while gets a little finer in the regions close to the wing see Figure 7.3. By the use of the special routine of OpenFoam, two boxes are introduced for static refinement within the vortex core. The first box extends from the leading edge to x/c = 10 distance, while the second one extends further to x/c = 20. In all the three directions each cell in the first box is refined with the factor of two, while this factor in the second box is four. This results in intermediate fine 7.2 million mesh grid. Further refining the 7.2 million mesh in both boxes by doubling the cell number in all three directions results in the finest mesh of 42.2 million cells Figure 7.5. The non dimensional position of the first node Y^+ is varied between 5 and 40 on the wing for all grids since the refinement is done only in the zone around the tip vortex. Thus, the inner part of the boundary is not resolved and these grids can be ascribed to the so called high Re number grids. In this the application of wall functions is necessary.

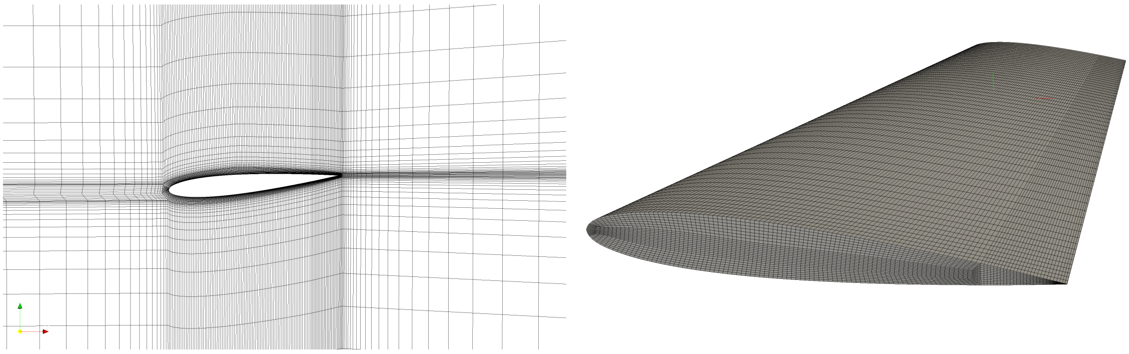


FIGURE 7.3: Left : 2.7 million mesh (coarse), Right : wing mesh

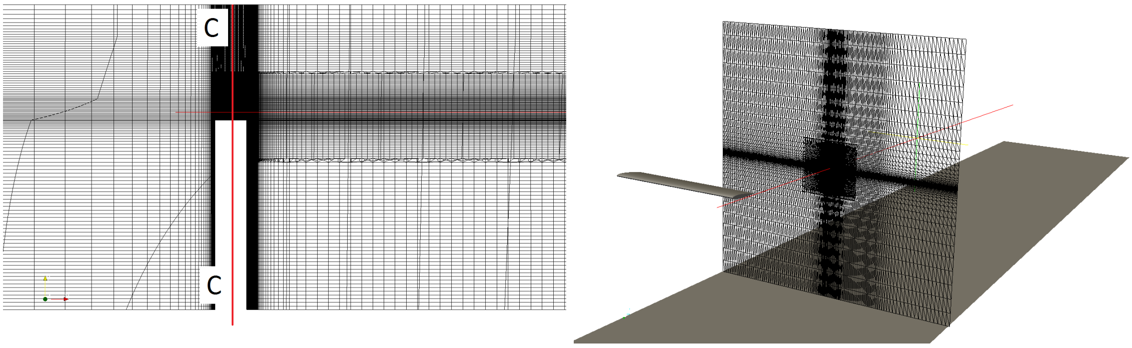


FIGURE 7.4: Left : Cross-section C-C, Right : Static box refinement in the wake

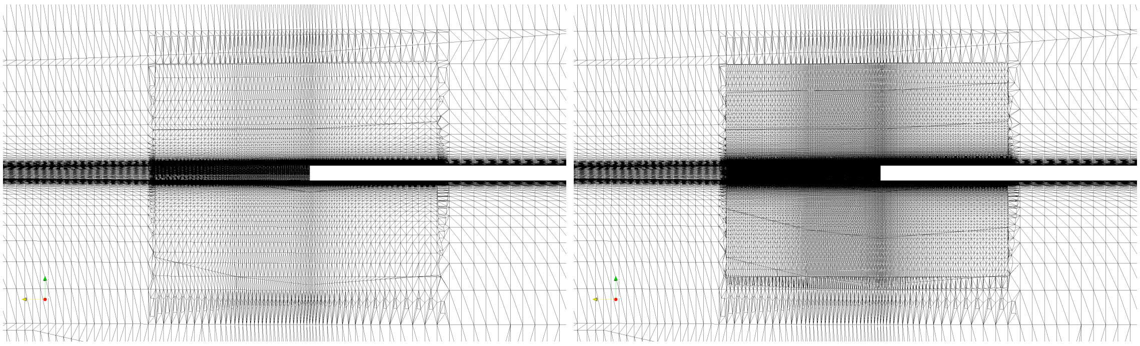


FIGURE 7.5: Meshes with static box refinement at C-C section, Left : 7.2 million mesh, Right : 42.2 million mesh

ii. Low-Re grid

The low-Re grid was generated using SnappyHexMesh. SnappyHexMesh is based on the octree-refinement algorithm. Some major reasons of choosing SnappyHexMesh are that

- the mesh generated is purely hexahedral thereby saving considerable cells with respect to purely tet mesh: up to 40 % less elements, due only to the element shape
- It has an ability of addition of internal and wall layers
- It allows for zonal meshing (vortex core meshing) and the mesh generated is fully compatible to OpenFOAM environment
- The meshing process can be run in parallel mode, therefore high quality meshes can be generated in little time using more computational resources.

The non dimensional position of the first node Y^+ is less than 5 for the low Re-number grid, with the near wall region well resolved using a special routine with the application of geometric progression for the growth of layers in the wall normal direction. (See Figure 7.8) The total number of cells is approximately 7.1 million. The size of the cells around the tip vortex in the measurement section (x/c = 5 to 30) is adjusted such as to obtain about nine cells inside the mean measured core diameter. The figures below show the details of the grid and the near wall refinement.

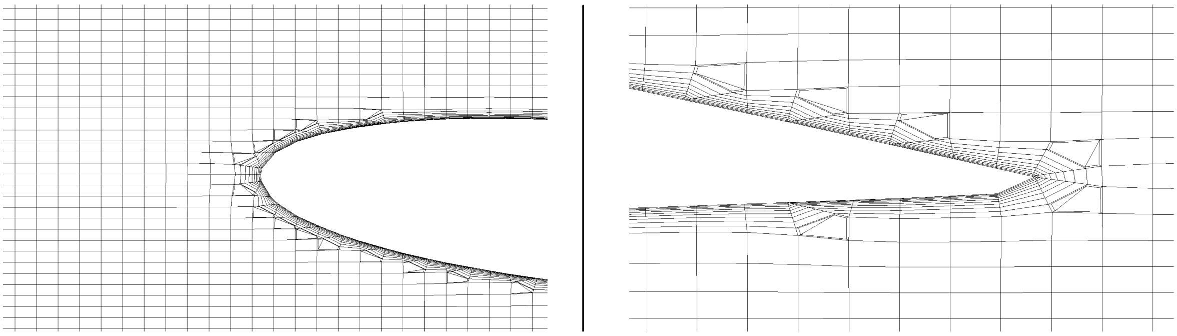


FIGURE 7.6: Near wall refinement

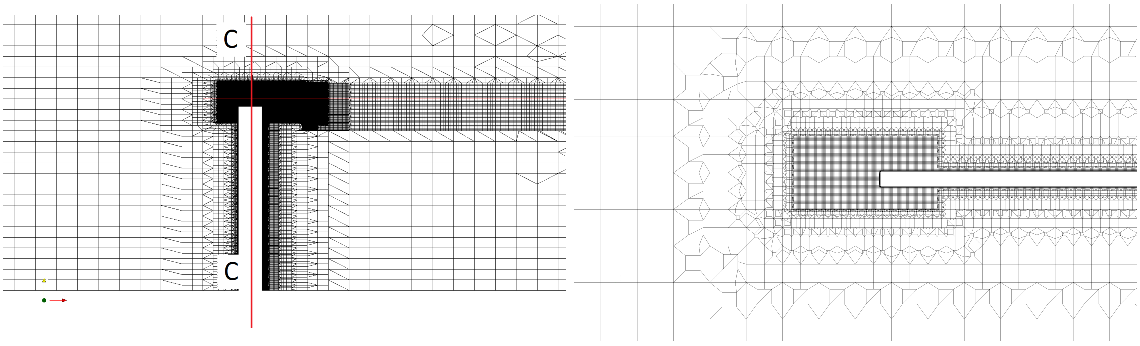


FIGURE 7.7: Left : Cross-section C-C, Right : Zonal refinement at C-C section

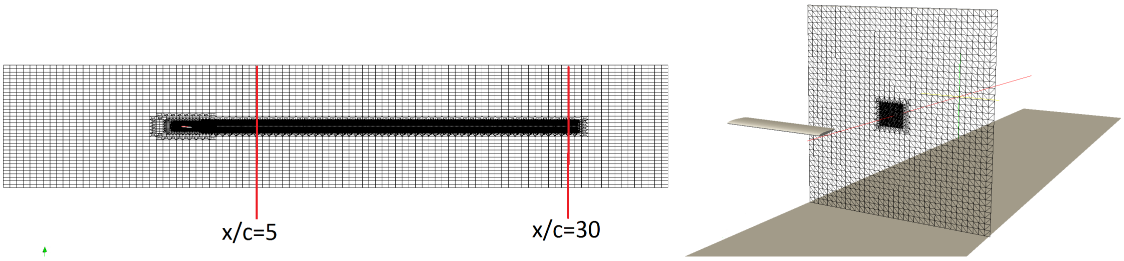


FIGURE 7.8: Left : Zonal refinement in the domain, Right : Zonal refinement in the wake

7.2.2.2 Adaptive Grids

Adaptive Mesh Refinement(AMR) is one measure to analyse the influence of the numerical error due to the domain discretisation. It allows one to refine the grid locally within the vortex core using the Q-criterion. For instance, such a technology developed by [151] is based on the computed numerical error gradients in the flow. Each mesh cell was spilt into two and four subcells in all three directions referring them further as Q1 and Q2 refinement respectively. This method was applied on the meshes described above and adapted at each time step. Buffering of the mesh was also introduced which takes into account the refinement layers of the neighbours of the cells which are set for refinement. The mesh gets refined only in the region of the tip vortex. AMR was applied for simulations with curvature corrected turbulence models and for the w/o turb. model. Figure 7.21 shows a quick look on how the AMR refines the mesh only in the vortex core region.

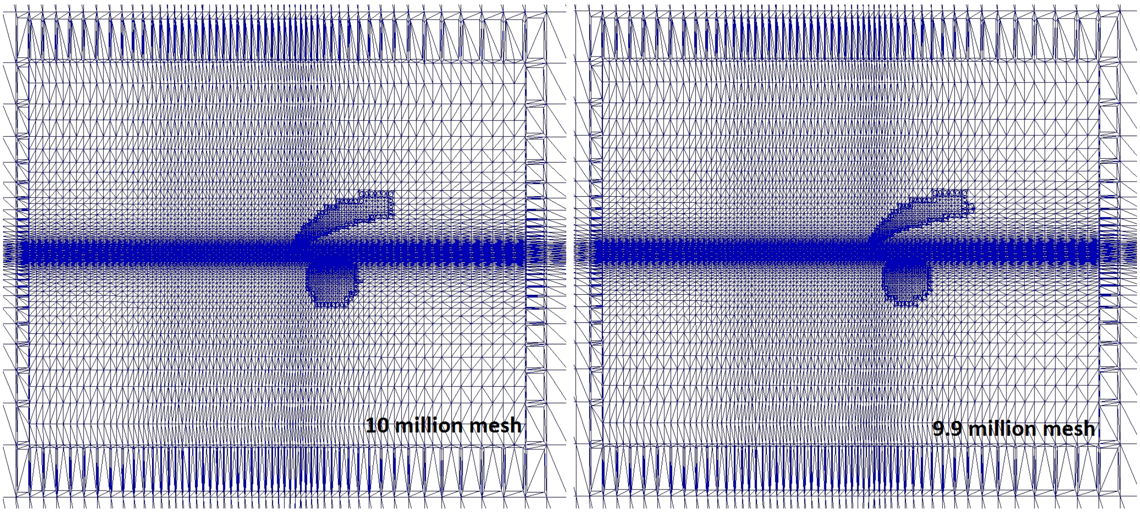


FIGURE 7.9: Grids at x/c = 5 station after application of adaptive mesh refinement. Left is for SA-CC model and Right is for SST-CC model

7.2.3 Solver settings and Boundary conditions

This section gives an overview over the simulation setup presenting the used solvers, discretisation schemes and turbulence models and setup of coupled simulations as well.

The symmetry boundary condition (BC) is used for all quantities within the symmetry plane. The zero gradient BC are enforced at the bottom, top and side faces of the computational domain. At the inlet the velocity has the fixed value u_∞, the turbulent kinetic energy k was set as $k_0 = \frac{3}{2}(IU_\infty)^2$, where I is the turbulence intensity taken as 0.08, the specific dissipation rate ω was specified as $\omega_0 = \frac{k_0^{1/2}}{C_\mu l}$, where $C_\mu = 0.09$ and $l = c$. At the outlet the zero gradient BC are enforced for velocity, k and ω fields. The pressure was zero at the outlet and satisfies the zero gradient BC both on the wing and at the inlet. Velocity fulfils the no slip condition on the wing. Wall functions for k and ω proposed in Menter et al. (2003) [64] are applied on the wing for only the High-Re grid. The Reynolds number used is 5.3 x10^5. Several simulations summarised in Table 7.1 were carried out. The following paragraphs provide further detailed information on the used numerical methods and models. In case of the unsteady simulations, the flow passed the measurement domain Δx = 30c (thirty times the chord length) at least twice (for each case, the simulation was stopped when the mean vortex flow converged).

Several different turbulence models are compared with the same solver settings. The following list gives an overview:

- one-equation turbulence model SpalartAllmaras presented by [60] (termed *SA*),
- one-equation turbulence model SpalartAllmaras with curvature correction presented by Shur et.al [61] (termed *SA-CC*),

- two-equation turbulence model kOmegaSST presented by [64] (termed *SST*),
- two-equation turbulence model kOmegaSST with curvature correction presented by [63] (termed *SST-CC*),
- Launder-Reece-Rodi Reynolds stress turbulence model presented by Launder et.al [65] (termed *LRR*),
- Laminar model i.e. an underresolved very large eddy simulation (VLES) without a sub-grid model. (termed *w/o Turb.*),
- Large Eddy Simulation with Dynamic Smagorinsky subgrid model [68]. (termed *LES*)

For the above RANS simulations following settings were taken.The spatial discretization of the convective term is performed using the filteredLinear scheme implemented in OpenFOAM. This scheme calculates the face values using blending of linear interpolation with a particular amount of upwind, depending on the ratio of the background (in-cell) gradient and face gradient. The amount of upwind is limited to 30%. Laplacian term was discretized using the linear scheme with explicit non-orthogonal correction. Pressure gradient was reconstructed using linear scheme based on the Green- Gauss theorem. The equations for k and ω were discretized in the same manner except the convective term, for which a TVD scheme with Sweby flux limiter was applied.

As mentioned above, one of a serious disadvantages of turbulence models which leads to a fast vortex decay is the overprediction of the turbulence intensity in the vicinity of the tip vortex core. To analyse this issues, some simulations are carried out without turbulence modelling, i. e. the unsteady Navier-Stokes equations are solved.

The simulations were carried out in an unsteady mode using the PISO algorithm running four times for pressure correction loop to reach convergence for the w/o Turb. model while two times for the LES model. Time derivatives discretisation has been done by using the Backward difference scheme. The maximum Courant number was approximately 10 for the finest 42 mio High Re-grid and approximately 12 for the low-Re grid.

Coupled Method Simulations

All the coupled simulations are conducted with pimpleFoam solver, which is a transient solver using the PISO and SIMPLE algorithms. The simulations were carried out in an unsteady mode running the algorithm with five times for pressure correction loop to reach convergence. In each time loop, sub-loops are involved one for solving the grid domain and other for solving the grid free domain. A wise choice of numerical setup must undertaken to conduct proper coupled method simulations, otherwise the stability could be critical. The convection term of the momentum equation is discretized with a linear upwind scheme which is second order, unbounded, and requires discretisation of the velocity gradient to be specified. The relaxation factor is set to 0.4 for all variables. The time step is recommended to be kept low to keep the discrete particles in the grid free part of the domain move slow. In our case, we kept $\Delta t = 10^{-5}$. A high time step, will make the particles traverse a larger distance within each time step, which gives improper solution or even may lead to instability. Also, the spatial interval of generation of new vortoncloud was set to 0.11 times the distance between the interface and outlet. The idea is to keep this value low in such a way that the overlapping occurs between the particles of the successive vorton cloud. The overlapping factor between the vortons was kept at 2. A wise strategy is adopted to run the vortex code to get proper results. It is described as follows - In the first step, switch off the scaling routines in the code and run the code until the few cloud of vortons are generated and moved further in the downstream away from the interface into the grid-free part of the vortex domain. In this first part it is to note that, the velocity of vortons may not match the interface velocity of the grid domain. In the second step, stop the simulation and switch on again but now with the scaling routines switched on, again new vortons will be generated but now their strength will be updated according to the factor which depends on the ratio of interface maximum tangential velocity from the grid based solution and far-field maximum tangential velocity from the grid free part. In the third step, let the scaling effect take place on the complete clouds of vortons as they get generated and move further downstream. Let the earlier clouds pass out of the domain example in case of Devenport case beyond x/c 30. Finally, the result of coupled method simulation is obtained with the complete new updated strength vortons in the full domain, which could be post-processed later. Usually, this process consumes a huge amount of time, since the vortex code works only in a serial mode i.e. on a single processor and is not parallelized yet which results in large computational cost at present. A detailed analysis of the computation costs of the coupled method simulation and parallelization requirements is discussed in Chapter 9.

Influence of interface position - The position of the interface is important while solving a coupled method. When the interface is too close to the wing, the outlet boundary of the grid domain has an effect on the solution close to wing which is not desired. The position of the interface shall vary application to application. By doing some a-priori simulations it could be estimated for a particular case. In our case for the wing, we observed that the interface position > 4c was sufficient enough to shield the effect of the outlet boundary on the near blade field. Since we have a validated result from the Devenport experiment at x/c 5, we chose our interface between grid based and grid free domain at x/c 5.

Mesher	Wall Treatment	Number of cells	Closure Model	Time Modelling
ICEM	High-Re	2.7 mio	SA, SA-CC, SST, SST-CC, LRR, w/o Turb.	steady
		7.2 mio	SA-CC, SST-CC, LRR, w/o Turb.	steady
		42.2 mio	SA-CC, SST-CC, w/o Turb.	steady
			w/o Turb., LES	unsteady $(0.31/10^{-5})$
Snappy-HexMesh	Low-Re	7.1 mio	SST, SST-CC	steady
			w/o Turb.	unsteady $(0.31/10^{-5})$

TABLE 7.1: Summary of simulations: meshing tool, wall treatment on the wing, number of cells in millions, turbulence models and time modelling

7.3 Numerical results and validation

7.3.1 Methodological Investigations

i. Study of the influence of the computational domain size and boundary conditions on the solution of the Poisson equation

In the Vortex in Cell VIC method, the computations of velocities is sufficiently accelerated due to application of the Poisson equation (7.1) and Fast Fourier Transformation (FFT) instead of direct summation using the Biot- Savart integral

$$\Delta\vec{u} = -\nabla \times \vec{\omega} \tag{7.1}$$

However, the solution of the Poisson equation can be influenced by the size of computational domain as well as the boundary conditions used for calculation of velocities. To investigate this influence, an analysis study of Poisson equation was carried out on the Devenport tip vortex in 3D by varying the computational domain size and boundary conditions. The initial vorticity field in the equation (7.1) was calculated based on the converged solution of tip vortex. The boundary conditions for velocity was then varied to Dirichlet ($u = 0$) and Neumann ($\partial u/\partial n = 0$) respectively to see the influence on the solution of (7.1). Computational domains c1 is 9 times larger than the chord length of the wing while c2 is 4.5 times larger than the chord length of the wing. It can be noticed from the Figure 7.10 left, that the influence of boundary conditions on solution of (7.1) is very negligible for a larger computational domain size c1. While from the Figure 7.10 right, when the boundaries of domain start to narrow, the influence of boundary condition is clearly visible with discrepancy by the application of Dirichlet boundary condition and increases much further in farfield. The Neumann boundary condition is observed to be more robust. It concludes that, maintaining the domain size sufficiently larger than the core size of the tip vortex (atleast 5 times the chord length), the influence of domain size as well as the boundary conditions on the solution of (7.1) could be diminished.

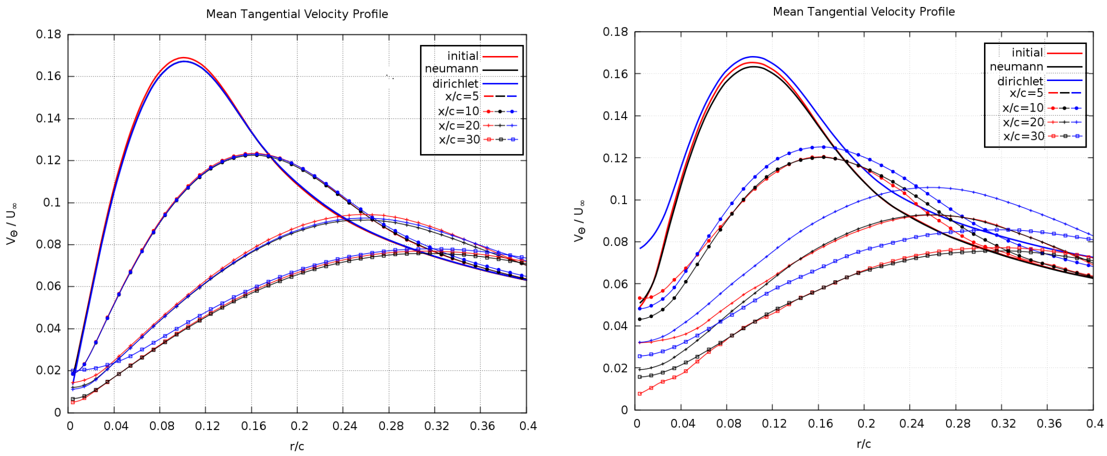


FIGURE 7.10: Influence of the computational domain size and boundary conditions on the solution of the Poisson equation, c1(9cx9c) domain (left) c2(4.5cx4.5c) domain (right)

ii. Accuracy study of the grid based simulation using analytical vortex elements

An analytical 2D vortex was modeled so as to have the similar properties of Devenport tip vortex (See Table 7.2). The input data is taken from Table 2 of the Devenport paper [2]. A Finite Volume method simulation is carried out to observe the diffusion of the vortex core with and without effect of viscosity considerations i.e. pure influence of numerical diffusion. The simulation was carried out for 250 and 1000 time steps. With the grid resolution of $\Delta = 0.001$, the $maxu_\tau / maxu_{\tau 0} = 0.94$ and 0.75 was obtained without the viscosity considerations and $maxu_\tau / maxu_{\tau 0} = 0.93$ and 0.72 was obtained with viscosity considerations at the end of 250 and 1000 time steps respectively. Therefore, in conclusion, the diffusion of tip vortex core due to numerical viscosity can be kept minimum, when the grid resolution in vortex core is finer i.e more than $\sim$ 7 grid cells/vortex core.

$U_\infty c/\nu$	x/c	r/c	Γ/ν	Δ	ν on/off	$maxu_\tau/maxu_{\tau 0}$ 250 time steps	$maxu_\tau/maxu_{\tau 0}$ 1000 time steps
530000	5	0.036	33900	0.001	off	0.94	0.75
530000	5	0.036	33900	0.001	on	0.93	0.72

TABLE 7.2: Results of analytical Devenport vortex

iii. Accuracy study of the grid free vortex method simulation using analytical vortex elements

The ”analytic” vortex tube used as a test case [149] which induces the velocity at any point P according to formula is given as :

$$\vec{V}(p) = \sum_{j=1}^{N} \frac{\vec{\Gamma} \times \vec{r}}{4\pi\sigma^3} \frac{\rho + 5/2}{(\rho+1)^{5/2}} \tag{7.2}$$

where $\vec{r} = \vec{p} - \vec{x}_j$, $\rho = |\vec{r}|^2/\sigma^2$, $\sigma = 0.075$, $\vec{x}_j = \delta_x(\vec{j}-1)\vec{i} + (\vec{j}+\vec{k})/2$, $\Gamma = 0.2/\sigma^2\delta_x$, $\delta_x = L_x/(N-1)$ and $N = 100$, L_x is the length of the vortex tube.

This analytic vortex tube is then computed using Vortex in Cell, Pure Lagrangian methods and Pure grid method and the results are as shown in Table 7.3 below.

Method	Scheme	Interpolation	Δ	$maxu_\tau/maxu_{\tau 0}$
Pure grid	Euler simple Euler simple Euler simple	- - -	0.02 0.01 0.002	0.80 0.82 0.92
Vortex-in-cell	Euler simple Euler simple Euler simple Euler 2nd order	W_1 W_1 M_4' M_4'	0.02 0.01 0.01 0.01	0.60 0.53 0.74 0.67
VIC w/o remeshing	Euler simple Euler 2nd order Euler simple Euler 2nd order	W_1 W_1 M_4' M_4'	0.01 0.01 0.01 0.01	0.76 1.02 0.77 1.07
Pure Lagrangian	Euler 2nd order Euler simple Euler 2nd order	- - -	0.02 0.01 0.01	0.82 0.70 0.85

TABLE 7.3: Results of analytical vortex tube after 250 time steps

It is seen from Table 7.3 that with Pure grid more diffusion of vortex is seen at a coarser resolution but result improves when the resolution is increased. VIC without remeshing gives better results as compared to with remeshing. The results of Pure

Lagrangian and Vortex in cell without remeshing are comparable and are better than the pure grid simulation.

7.3.2 Data assessment procedure

Numerical results are compared with measurements for the vortex core radius r_1, the peak tangential velocity $V_{\theta 1}$ and the axial velocity at the vortex center U_0. They are extracted from simulations without the correction of vortex wandering, which was not documented in simulations although it occurs under real measurement conditions due to instabilities of the wind tunnel flow and small wing oscillations. The vortex center at each x-position is determined by the local minimum of λ_2 criterion in the $x = const.$ plane. The difference in results using any other criterion, for instance Q or λ_{ci}, is negligible. The axial velocity U_0 is evaluated at the center. At each radius r measured from the vortex center, the tangential velocity is averaged in circumferential direction using 32 lines with equal angular spacing around the vortex centre. Figure 7.11 shows four of these (dashed) lines as examples. The coordinate system $\{y', z'\}$ is shifted to the vortex centre. The vortex core is defined as the radius r_1 at which the averaged tangential velocity reaches its maximum value.

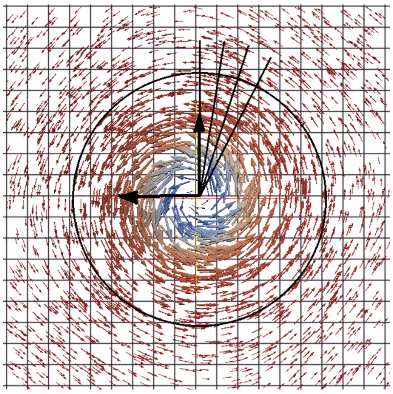


FIGURE 7.11: Cross section at $x/c = 5$, the vortex velocity profiles are extracted from the simulation results on the dashed lines; the black isoline represents $\lambda_2 = 0$, and the arrows indicate the tangential velocity [7]

The vortex core parameters are analysed at the x-positions of the measurements $x/c = [5, 10, \ldots, 30]$ and additionally at $x/c = [1.5, 2, 3, 4]$. Upstream of $x/c = 1.5$ the concentrated tip vortex is hard to identify since the roll up process is not completed. If the core radius exceeds $0.4c$ at large distances from the wing, the vortex core parameters are not evaluated, because the vortex becomes too smooth and dissipates to a weak level. The evaluation of the unsteady simulations starts after 0.3 s. In this time, the flow passes the distance of interest $x/c = 30$ more than two times what is enough to exclude the start-up transitional phase. The unsteady simulations results are averaged in time within the period of 0.1 s, which is large enough to guarantee the convergence of statistical data. For a proper validation, it is important to compare our results not only with measurements but also with other numerical simulations. Unfortunately, we found only one available simulation data for the Devenport et al. test case performed by Wells et. al. (2010) [78], who applied various turbulence models to assess their performance in predicting a tip vortex flow: the one-equation model of Spalart Allmaras (with and without curvature correction) and the Reynolds stress transport model (RSM) LRR proposed by Launder et al. (1975) [65]. A structured mesh with 11.9 M cells and wall functions for the resolution of the boundary layer flow on the wing were utilised. Unfortunately, there is no information available on the grid resolution in the vicinity of the vortex core. Wells determined the tip vortex parameters in a slightly different way than in this study. The vortex center is determined by the maximum helicity. The tangential velocity profile used to determine r_1 and $V_{\theta 1}$ is evaluated only along one line parallel to the z-axis through the vortex core without averaging in circumferential direction. The main conclusion of his work is the superiority of the RSM model in comparison with others. Its application leads to results which match very well the experimental ones at small x/c. Unfortunately, no information is presented downstream of $x/c = 10$, because of strong dissipation of the tip vortex in the far field.

7.3.3 Influence of curvature correction and mesh refinement on vortex core parameters

Figures 7.12, 7.13 and 7.14 illustrates the influence of the mesh resolution and the curvature correction for the ICEM mesh. First of all, it is to note a strong degradation of the concentrated tip vortex in the far wake field. The maximum tangential velocity, and, consequently, the vortex strength, losses about two thirds of its initial value whereas the vortex core is spread up with the factor of four and more. These results underline the importance of the problem considered in the present work. It is obvious, that the modern numerical models are not capable of predicting the tip vortex flows at relatively moderate distances from the wing. The distance of order of $x/c = 10$ and larger are quite typical for marine applications, for instance for the distance between the marine propellers and rudders if the helicoidal character of the propeller tip vortex is taken into account. Therefore, numerical analysis of important practical problems, for instance, of the tip vortex cavitation influence on the rudder erosion becomes impossible. In this work, we are going to clear which models and grids are able to improve the simulation accuracy. Grid independence study showed that convergence is attained on the grid with 7.2 mio cells with and without curvature correction models. For this resolution the influence of the curvature corrections is positive but insufficient for a radical improvement of the modeling accuracy. As seen, the confining effect of both curvature correction models on the vortex radius is big at coarse resolution (2.7M) and becomes negligible at fine resolution. When observed both the models with respect to the eachother, their effects are comparable.

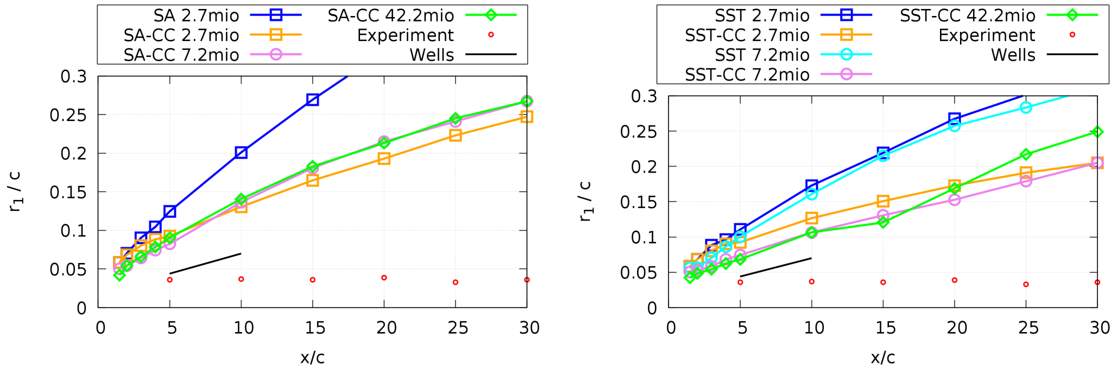


FIGURE 7.12: Influence of curvature correction and mesh refinement on vortex core radius for eddy viscosity models

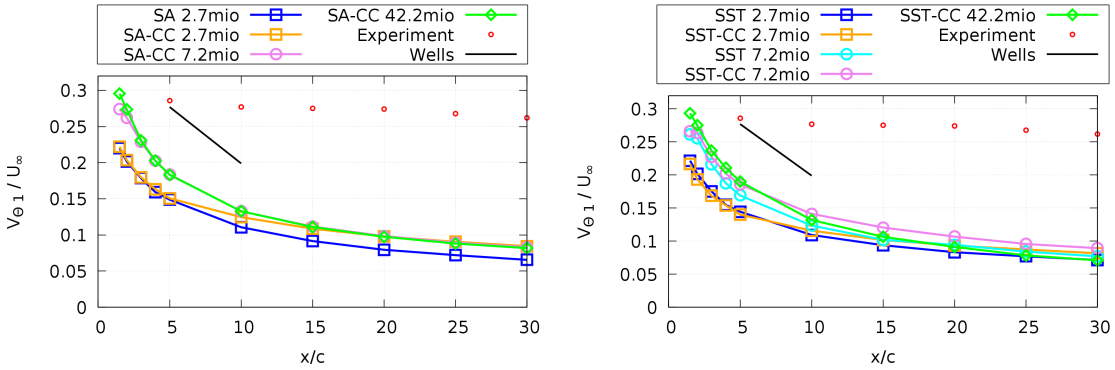


FIGURE 7.13: Influence of curvature correction and mesh refinement on peak tangential velocity for eddy viscosity models

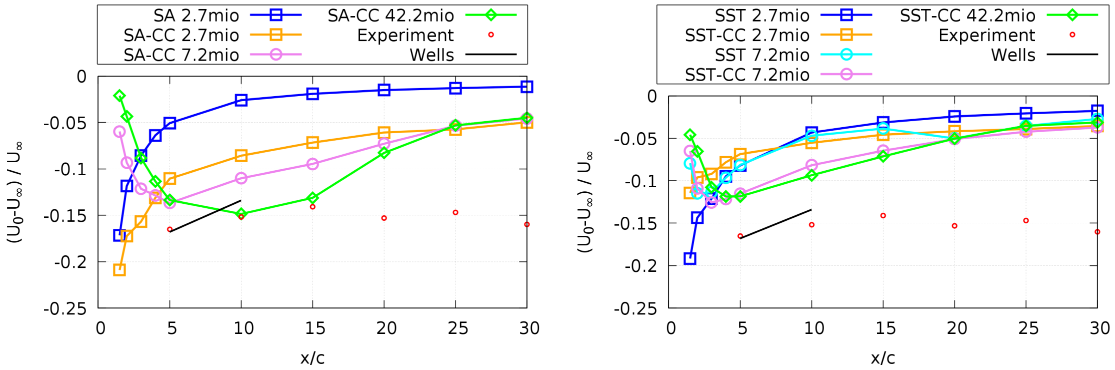


FIGURE 7.14: Influence of curvature correction and mesh refinement on axial velocity deficit for eddy viscosity models

7.3.4 Influence of different closure models and mesh refinement on vortex core parameters

Figures 7.15, 7.16 and 7.17 shows the comparison for the core parameters obtained using different closure models. Among all turbulence models LRR yields the best result. The reason is that the RSM models take the anisotropy of the flow into account and the consideration of the rotation effects is an inherent part of the modeling. Unfortunately, application of the RSM has a strong limitation because of strong numerical instability which was documented in our case at meshes finer than 7.2M. Surprisingly, the most promising results were obtained using the laminar solution which was performed in an unsteady mode since a steady laminar model at high Reynolds number is physically incorrect and results in a large pressure residual in iterations. The present numerical simulations can be classified either as a very large eddy simulation (VLES) without any subgrid models or as an under resolved direct numerical simulation termed here as w/o Turb. model. As seen in Figures 7.15, 7.16 and 7.17, the w/o Turb. model shows the lowest rate of decay followed first by LRR and then by the turbulence models with curvature corrections (CC). The LES model exhibit almost similar behaviour when compared to w/o Turb. model. Strong decay of the vortex strength and the vortex core spreading downstream of $x/c = 20$, is due to coarsening of the grid at $x/c > 20$. Comparing the result obtained without turbulence modelling with the experimental one at $x/c = 30$, the core radius is about 3.8 times larger, the peak tangential velocity is about 42 % smaller for the numerical prediction. Considering the mesh refinements, the accuracy of the core radius and peak tangential velocity prediction can be significantly increased refining the mesh in the vicinity of the tip vortex

Here it should be noted that, as stated by Devenport et al. (1996) [2], the flow on the wing and the tip vortex depend on the tripping of the boundary layer on wing. At $x/c = 10$, for different locations of tripping the radius r_1 is changed between 0.033 and 0.037, whereas without tripping it was 0.038. The tangential velocity is varied between 0.263 and 0.286 being equal to 0.291 without tripping. These differences are around ten percent and can not be used as an explanation of big discrepancy between numerical laminar and turbulent results. The parameter most affected by the turbulence tripping is the axial velocity since its distribution is strongly dependent of the wing boundary layer which is in turn depends on the flow regime on the wing. With tripping the axial velocity deficit $(U_0 - U_\infty)/U_\infty$ is varied between -0.152 and -0.182 whereas it is two times less without tripping -0.086. The latter value is in a good agreement with the w/o Turb. flow unsteady solution See Figure 7.17.

Wells [78] evaluates the simulation results only until $x/c = 10$, as he remarks excessive dissipation of the tip vortex further downstream. This can be seen from all the Figures. Especially the decrease of the peak tangential velocity is very high compared to the results obtained within this study. Nevertheless, the accuracy of all three vortex core parameters at $x/c = 5$ is very good and the result is clearly superior in comparison to the results obtained here. A possible explanation for the decrease between $x/c = 5$ and $x/c = 10$ is a change in the mesh (coarsening) in Wells results. Unfortunately, Wells (2009) [78] provides no detailed information concerning the mesh density in the vicinity of the vortex core which would allow to draw more detailed explanations of the difference between both simulations.

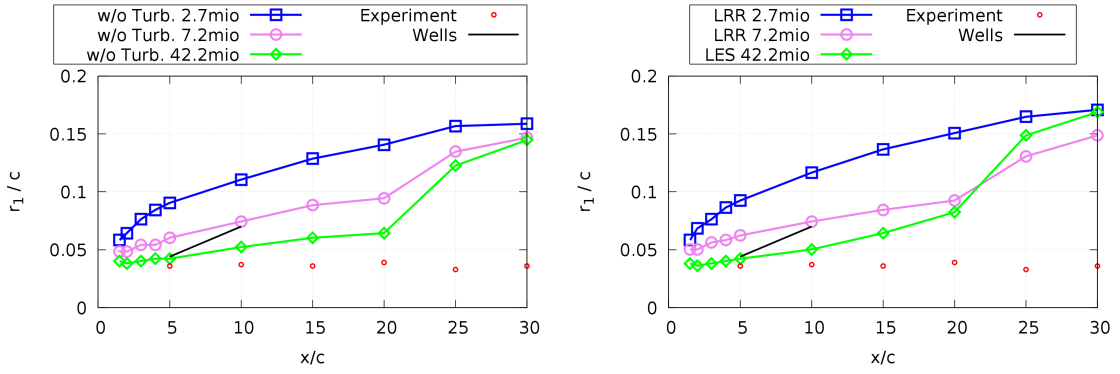


FIGURE 7.15: Influence of different closure models and mesh refinement on vortex core radius

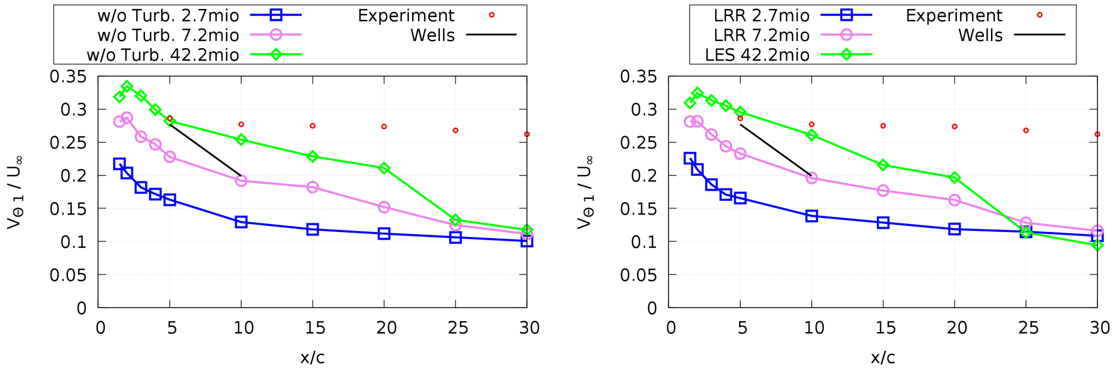


FIGURE 7.16: Influence of different closure models and mesh refinement on peak tangential velocity

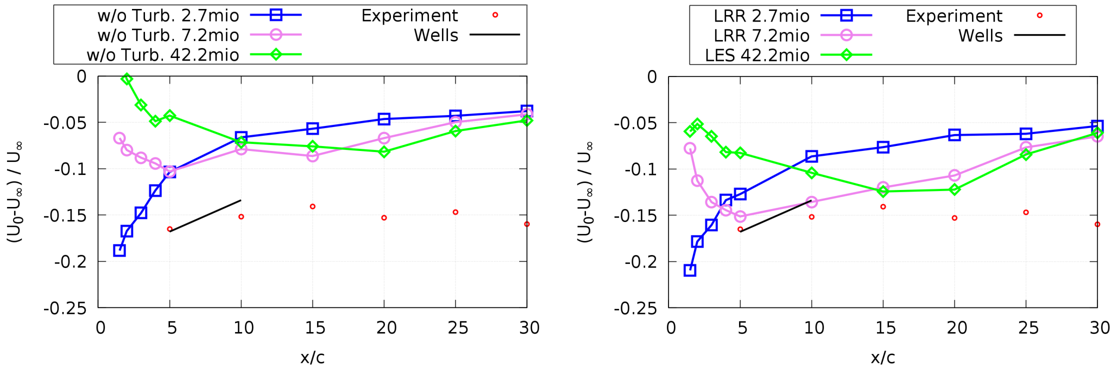


FIGURE 7.17: Influence of different closure models and mesh refinement on axial velocity deficit

Increase of resolution in turbulent simulations (eddy viscosity models) from 7.2 M to 42 M of cells, which reduces the discretization errors and artificial viscosity, shows not much expected improvement in the numerical results. Therefore, superiority of the laminar (w/o Turb.) solution leads to the conclusion that this model is physically more relevant than turbulent models and the flow inside the tip vortex is rather laminar than turbulent although the Reynolds number is high enough to expect the turbulent character of the flow both on the wing and in the wake. This simulation supports the conclusion of Devenport et al. (1996) [2] that "flow in the core is laminar and that velocity fluctuations experienced here are inactive motions produced as the core is buffeted by turbulence from the surrounding wake". The shapes of the profiles do change, however, and at a rate that is not inconsistent with laminar diffusion [2].

7.3.5 Influence of different mesh types on vortex core parameters

Figures 7.18, 7.19 and 7.20 show the influence of different meshes for SST, SST-CC and w/o Turb. models. There is a difference seen on the SnappyHexMesh grid for the SST model as compared to the same from the ICEM grid. This can be explained by different treatment of the boundary layer on the wing. On ICEM grid the inner part of the layer is modelled by wall function. When the boundary layer is well resolved the advantage of the SST approach for boundary layer modelling comes into play. The thickness of the boundary layer, which has a strong impact on the tip vortex core radius, is predicted more accurately by the SST model. See Figures 7.18, 7.19 and 7.20.

For the SnappyHexMesh grid, the size of the mesh cells l in the vicinity of the tip vortex core (in the x-plane) is approximately $l/c = 0.00839$. For the ICEM mesh with 7.2Mcells, the cell size changes in the vicinity of the vortex (in the x-plane), because of the mesh structure. The average cell size is approximately $l/c = 0.01074$, i.e. about 28% larger than for the SnappyHexMesh grid.

Based on the results gained with the w/o Turb. approach, it is clear that the result on the SnappyHexMesh grid is superior compared to the one on the ICEM grid. The core size is smaller, the peak tangential velocity is higher and the axial velocity deficit is also smaller. All this results in a smaller discrepancy with experimental data for each quantity. Comparing the result obtained without turbulence modelling with the experimental one at $x/c = 30$, the core radius is about 1.65 times larger, the peak tangential velocity is about 26 % smaller for the numerical prediction. One reason for the superiority of the SnappyHexMesh grid might be the increased mesh density in the vicinity of the vortex core due to zonal meshing technique upto $x/c = 30$. Another possible reason is the different wall treatment. ICEM grid has large $Y+$ values and requires utilisation of wall functions whereas the boundary layer is resolved on SnappyHexMesh grid.

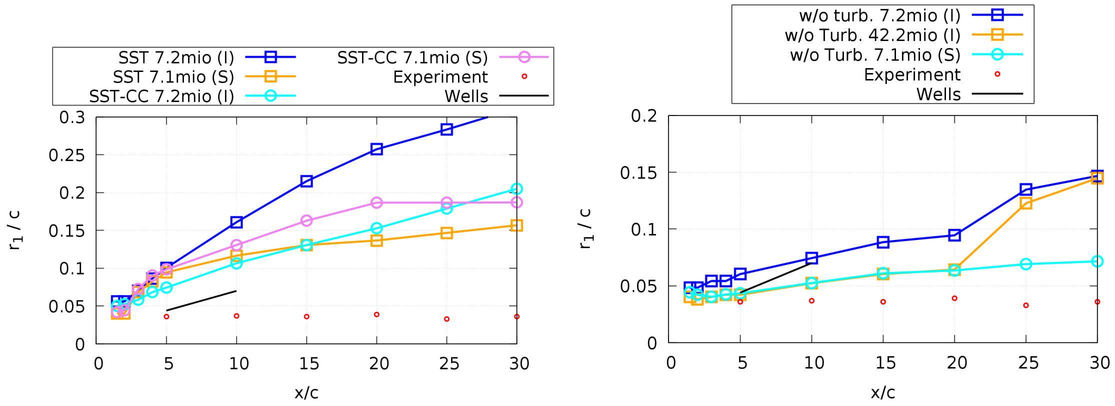


FIGURE 7.18: Influence of different mesh types on vortex core radius

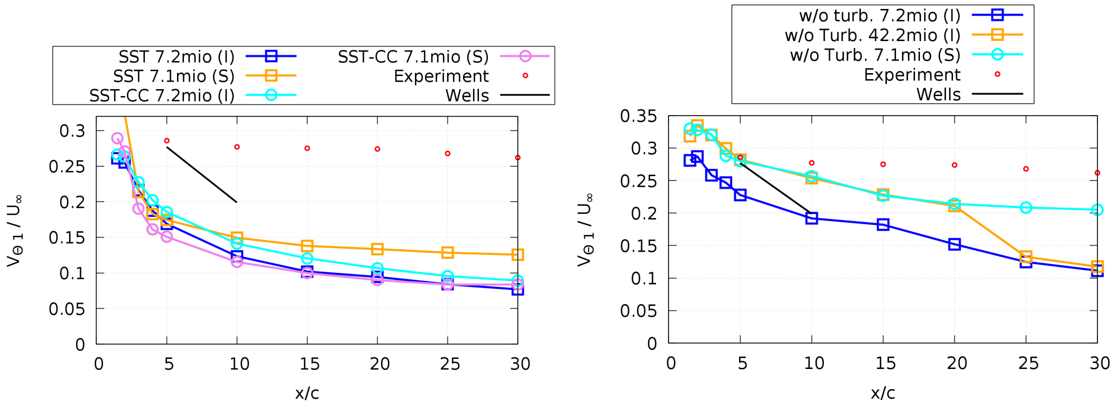


FIGURE 7.19: Influence of different mesh types on peak tangential velocity

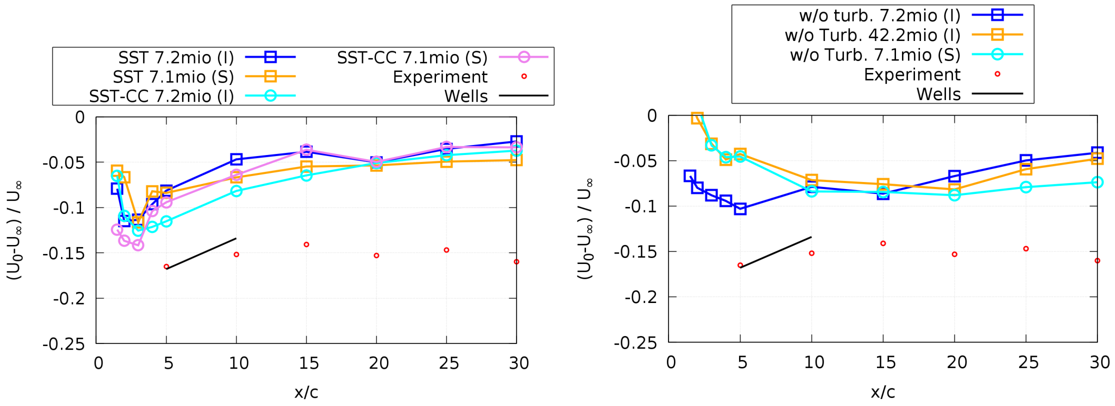


FIGURE 7.20: Influence of different mesh types on axial velocity deficit

7.3.6 Potential of Adaptive Mesh Refinement

Adaptive Mesh Refinement is one of the measure to analyse the influence of the numerical error due to the domain discretisation. This error induces numerical diffusion and leads to an increased dissipation of tip vortices. With many prior investigations, the strategy of implementation of AMR to obtain best results was derived and applied. The algorithm was applied by consideration of structured mesh topology consisting of pure hexahedral cells since the routine splits the hexahedral cells isotropically in all three directions during refinement process. The notion of infrequent refinement was introduced by a factor n which is the total number of refinements required. $n = Totaltime/(deltaT \times refineInterval)$ to achieve stability. With OpenFOAM 3.0.x version there was a effective control kept over the total number of cells exceeding a permissible limit during refinement process. Figure 7.21 shows the comparison between the base grid and the locally refined grid after the application of AMR. Figure 7.23 shows the difference between the ∇u and Q criteria local refinements. Figure 7.22 shows the effect of AMR on velocity iso-surfaces at $x/c = 5$.

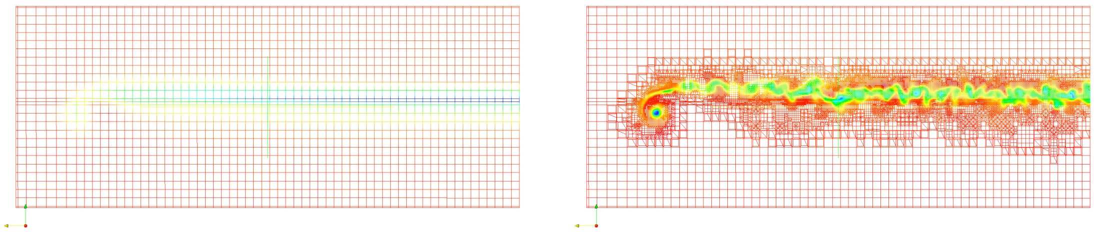


FIGURE 7.21: Local dynamic refinement in the vortex core at $x/c = 5$.
Left : base grid, Right : adaptive grid

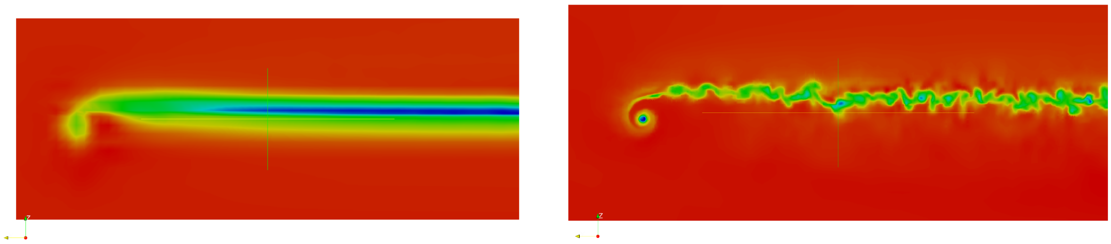


FIGURE 7.22: Velocity iso-surfaces showing the effect of adaptive mesh refinement at $x/c = 5$. Left : with base grid, Right : with adaptive grid

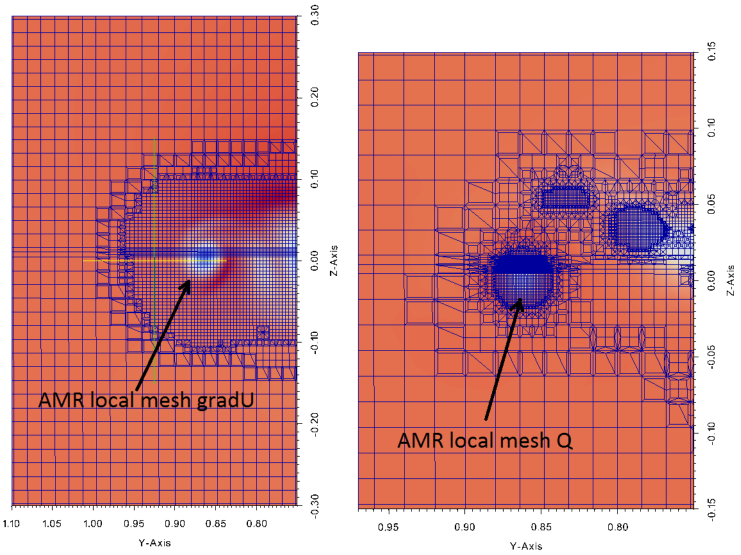


FIGURE 7.23: Mesh distribution using AMR in the vicinity of the vortex core at x/c = 5, Left : criteria ∇u, Right : criteria Q

The results of influence of AMR on Devenport 3D tip vortex are shown in the Figure 7.24 for SST model and in the Figure 7.25 for w/o Turb. model. The adaptive mesh refinement shows its potential by locally refining the vortex core dynamically resulting in improvement in accuracy in both the models. Further the refinement criteria Q was better as compared to ∇u (See Figure 7.24) since it helps in identifying effectively the strong local gradients especially in the case of w/o Turb. model where the tip vortex is pretty unstable consisting of smaller rolled up vortices. The improvement in accuracy w.r.t base mesh in w/o Turb. model was better as compared to the eddy viscosity model SST. Nevertheless, the results are far from the experimental values.

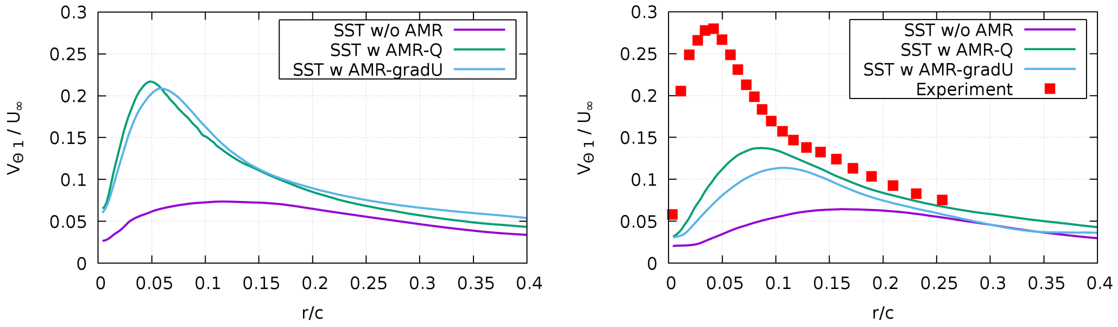


FIGURE 7.24: Influence of adaptive mesh refinement on mean tangential velocity for SST model. Left : at $x/c = 1.5$, Right : at $x/c = 5$

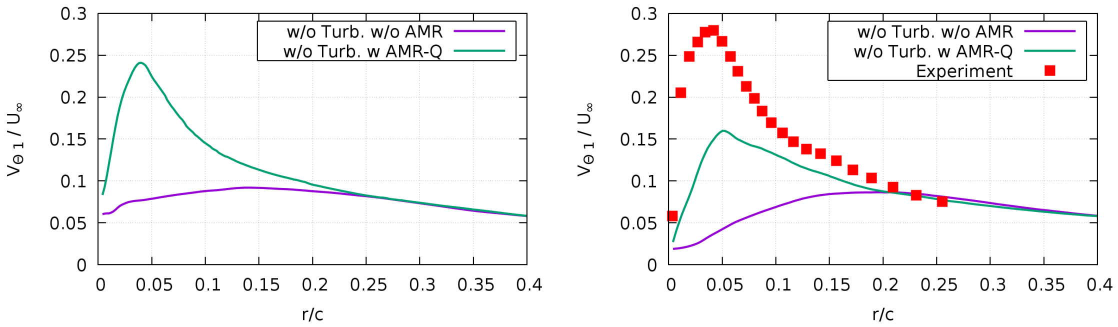


FIGURE 7.25: Influence of adaptive mesh refinement on mean tangential velocity for w/o Turb. model. Left : at $x/c = 1.5$, Right : at $x/c = 5$

A detailed exploration was done in regard to the state of the art in adaptive mesh refinement technology. Jasak et. al. (1996) [153] discussed the technique of AMR based on the numerical error to distribute the error uniformly throughout the domain to improve solution accuracy. He used a test case of cross-flow with a line source. Mustafa et. al. (2000) [154] used error indicator for adaptive refinement of tip vortex for rotorcraft blade analysis w.r.t pressure distributions on the blade surface. Hee Jung Kang et. al. (2002) [155] used local maximum of vorticity for adaptive refinement of tip vortex in the wake until $x/c = 1$, but presented the results only for Cp. Unfortunately no data of effect on velocities in the wake was presented. In similar way, effect of AMR on force coefficients was discussed by Yann Le Moigne et. al. (2004) [156] in his PhD work for simulations of delta wings. Also, Pashias et. al. (2005) [157] had shown qualitative results of the effect on structure of vortex due to AMR for the case of propeller tip vortex simulation. In the similar way, many people like James et. a. (2008) [158], Zurheide et. al. (2010) [159], Nathan et. al. (2011) [160], Phillips et. al. (2011) [161], Haris et. al. (2013) [162], David et. al. (2015) [163], Windt et. al. (2015) [164], Wang et. al. (2015) [165], Neal et. al. (2017) [166] and Jinlan et. al. (2017) & (2018) [167] [168] discussed several strategies of adaptive refinement for vortices and presented results based on AMR effect on force coefficients, Cp distributions, computation times and some other qualitative results. The effect of AMR on wake velocities was unfortunately not documented in these works. Only few notable works of Wissink et. al. (2010) [169], Kamkar et. al. (2011) [170], Kasmai et. al. (2011) [171] and Peron et. al. (2013) [172] discussed on the effect of AMR on wake velocities of tip vortex. Wissink et. al. (2010) [169] applied AMR based on identifying the regions of high vorticity for rotorcraft wake resolution. His analysis was based on NACA0015 wing at 12 deg angle of attack(AOA) and aspect ratio(AR) of 6.6 with Reynolds number as 1.5 million. The results were presented on effect of AMR on tangential velocity at $x/c = 2$ and $x/c = 6$. Kamkar et. al. (2011) [170] applied AMR based on various criteria like Q, λ_2, λ_{ci}. His analysis was based on NACA0015 wing at 12 deg AOA and AR of 6.6 with Reynolds number as 1.5 million. The results were presented on effect of AMR on tangential velocity at $x/c = 1$ and $x/c = 6$. Kasmai et. al. (2011) [171] applied AMR based on identifying features by maximum tangential velocities. His analysis was based on NACA0012 wing at 10 deg AOA and AR of 0.75 with Reynolds number as 4.6 million. The results were presented on effect of AMR on tangential velocity at $x/c = 1.4$. Peron et. al. (2013) [172] applied AMR based on criteria Q. His analysis was based on NACA0015 wing at 12 deg AOA and AR of 6.6 with Reynolds number as 2 million. The results were presented on effect of AMR on tangential velocity at $x/c = 1$ and $x/c = 6$. All the four above compared their results at respective $x/c's$ with experimental data. In all the above studies, AMR showed potential of improving the accuracy w.r.t without AMR but still faced discrepancies w.r.t experimental data. This refers to the same conclusion we have also drawn from our AMR simulations. Figure 7.26 shows the error between experimental data and AMR simulation at different x/c stations referred to various previous works including our present work. It is clear, that with our derived implementation strategy of AMR, our results showed minimum error predictions between AMR result and experimental data when compared to rest of the works in this field.

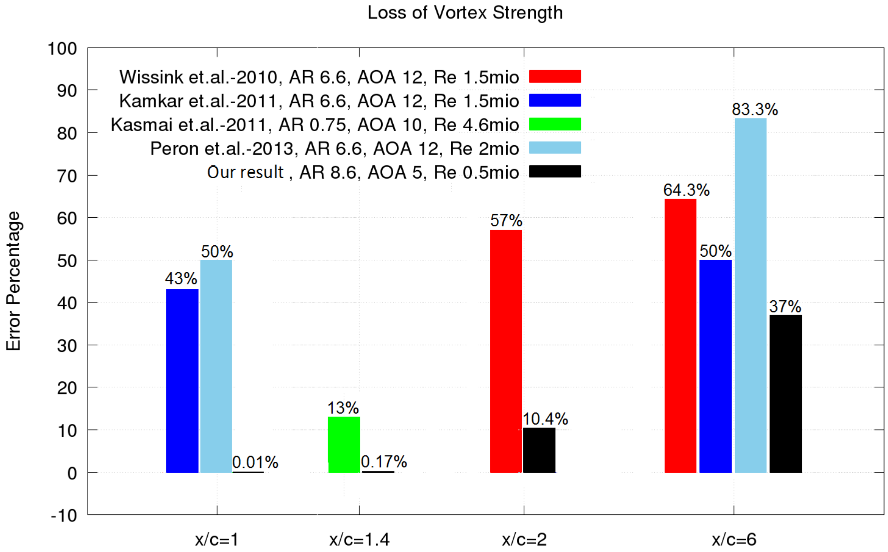


FIGURE 7.26: Error predictions for peak tangential velocity at different x/c stations using adaptive mesh refinement

7.3.7 Validation of different grid methods at x/c 5

Figure 7.27 illustrates the normalized mean tangential velocity profiles for different grid approaches with their respective best solutions at the distance of 5 chords behind the wing. Mean velocity profiles are obtained by averaging the maximum tangential velocities in circumferential direction along radial lines passing through the vortex core which is identified using λ_2 criteria. Among the eddy viscosity models, the SST-CC model is slightly better as compared to SA-CC model w.r.t the peak of the mean tangential velocity profile as well as confinement of vortex radius.

LRR yields better result overall when compared to SA-CC and SST-CC models w.r.t the peak of the mean tangential velocity profile. Confinement in the vortex core is also observed with LRR which occurs due to inclusion of rotation term in the equations of LRR model. The most promising results were obtained by use of dynamic Smagarinsky LES and w/o Turb. model with ICEM grid as well as SnappyHexMesh grid showing the best agreements with the experimental profile at $x/c = 5$.

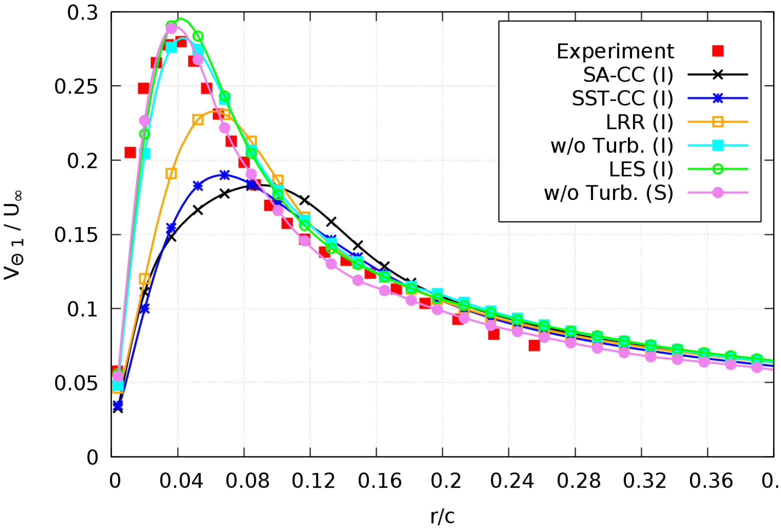


FIGURE 7.27: Influence of different closure models on mean tangential velocity at x/c = 5

7.3.8 Validation of force coefficients

The tangential velocity close to the wing is proportional to the vortex circulation and, therefore, to the lift. As shown in the Table 7.4 the lift coefficient C_L obtained on the ICEM grid is around 0.42 for the turbulent regime and 0.46 for the laminar one. The latter value is equal to this obtained from the panel code Autowing [152]. There is a clear convergence for the SST-CC and SA CC models. The simulations on the SnappyHexmesh grid provide the values comparable to the values obtained from ICEM mesh. We suppose that at small angle of attack of 5 degrees the lift coefficient should be independent on the viscous effects. Therefore the results for laminar flows and turbulent flows should be close each to other. A possible reason for the low lift coefficient on the ICEM grid with turbulent models is the inaccuracy in the turbulent boundary layer modelling using wall functions applied in the buffer zone region between $5 < y^+ < 30$. As a result, the boundary layer thickness is overestimated what results in the overestimation of the displacement effects and reduction of the lift coefficient.

Model	C_L	C_D
SA-CC 7.2mio (I)	0.421	0.0184
SA-CC 42.2mio (I)	0.420	0.0184
SST-CC 7.2mio (I)	0.418	0.0183
SST-CC 42.2mio (I)	0.416	0.0183
SST-CC 7.1mio (S)	0.447	0.0184
LRR 7.2mio (I)	0.418	0.0207
LES 42.2mio (I)	0.441	0.0178
w/o Turb. 42.2mio (I)	0.460	0.0177
w/o Turb. 7.1mio (S)	0.450	0.0176

TABLE 7.4: Lift and drag coefficients for the different closure models

7.3.9 Study of circulation in the vortex core

The Figures 7.28, 7.29 and 7.30 provides a perspective of the roll-up process of the tip vortex at incremental stations behind the wing starting from $x/c = 0.1$ upto $x/c = 5$. Since the lift coefficient for all turbulent models is almost the same, there is no big difference in the circulation of the tip vortex as shown in Figures 7.28, 7.29 and 7.30. The circulation Γ was found using the classical definition $\Gamma_R = \oint V_\theta dC_r$, where C_r is the circle with the radius r/c. Numerical integration along the circle is performed using $L = 360$ points. The velocity V_θ at the points is calculated using a linear interpolation between neighbouring grid cell centers. Increase of L doesn't lead to the increase of accuracy because the distance between the integration points is smaller than grid cell size. The circulation of the laminar solution inside of the tip vortex is higher than this of the turbulent solutions what is in accordance with relations stated above for the tangential velocity. Outside of the vortex core at $r/c > 0.05$ the LES Γ is larger than these of the turbulent calculations but eventually converges until $x/c = 5$, while the laminar Γ is proved to be smaller than these of the turbulent calculations. As seen in Figures, the roll up process is not completed up to $x/c = 5$ and the circulation continues to grow in the range up to $r/c = 0.5$ due to vortex sheet shed from the trailing edge. Contrary to the turbulent solution the laminar solution is very unsteady with creation of vorticity of opposite signs in the vortex sheet. The contribution of vorticity with rotation opposite to this of the tip vortex reduces the growth of the circulation and makes its dependency on r non monotonic. Another reason for the non monotonic change of Γ could be insufficient time of averaging of high unsteady laminar flow.

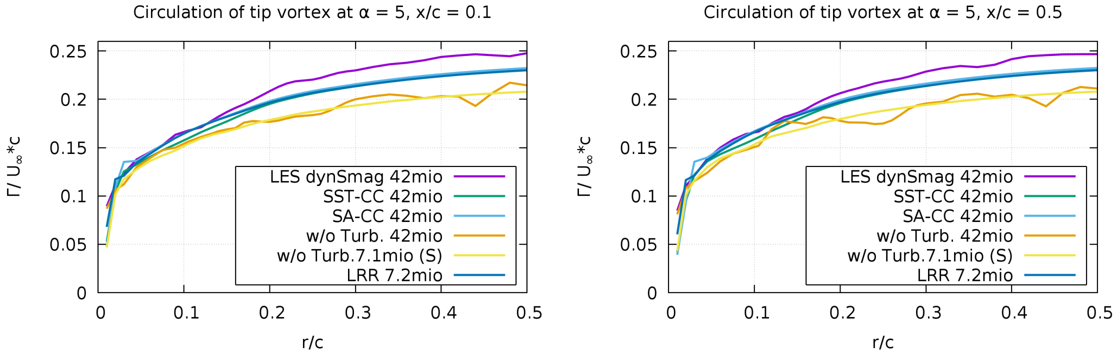


FIGURE 7.28: Circulations of the tip vortex with the radius chord distance at $x/c = 0.1$ and $x/c = 0.5$

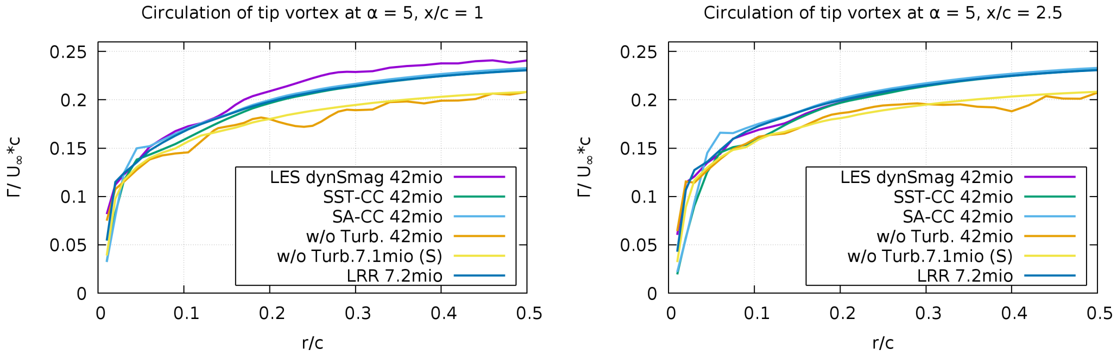


FIGURE 7.29: Circulations of the tip vortex with the radius chord distance at $x/c = 1$ and $x/c = 2.5$

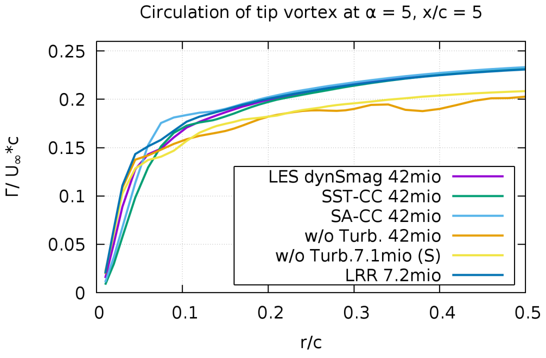


FIGURE 7.30: Circulations of the tip vortex with the radius chord distance at $x/c = 5$

7.3.10 Velocity distributions in boundary layer

The following section compares the velocity distribution profiles in the boundary layer close to the wing surface at different locations or probes as shown in the Figure 7.31.

- **Probe 1 :** Location far from wing tip at $y/b = 0.25$, $x/c = 0.5$
- **Probe 2 :** Location at mid-wing at $y/b = 0.5$, $x/c = 0.5$
- **Probe 3 :** Location near the wing tip at $y/b = 0.75$, $x/c = 0.25$
- **Probe 4 :** Location near the wing tip at $y/b = 0.75$, $x/c = 0.5$
- **Probe 5 :** Location near the wing tip at $y/b = 0.75$, $x/c = 0.75$

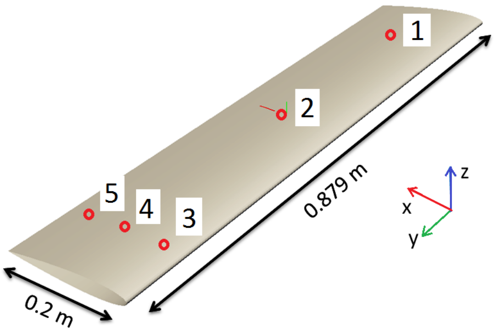


FIGURE 7.31: Locations of the probes on the wing surface

It could be observed that far away from the wing tip even at the mid-wing section the mean velocity grows faster in the boundary layer for LES model as compared to the RANS-SST model, but eventually matches with the SST model as we move away from the wing surface in normal direction. This holds true for both pressure as well as suction sides of the wing. See Figures 7.32 and 7.33.

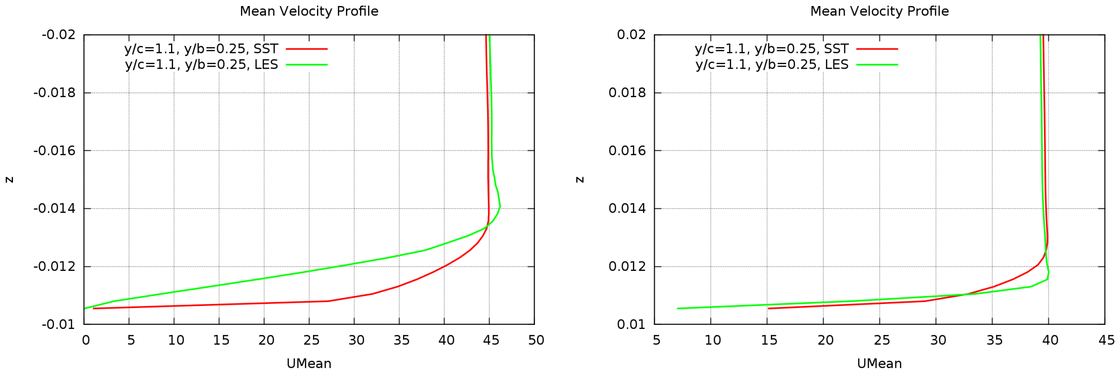


FIGURE 7.32: Velocity profile normal to the wing surface calculated at *Probe*1. Left : suction side, Right : pressure side

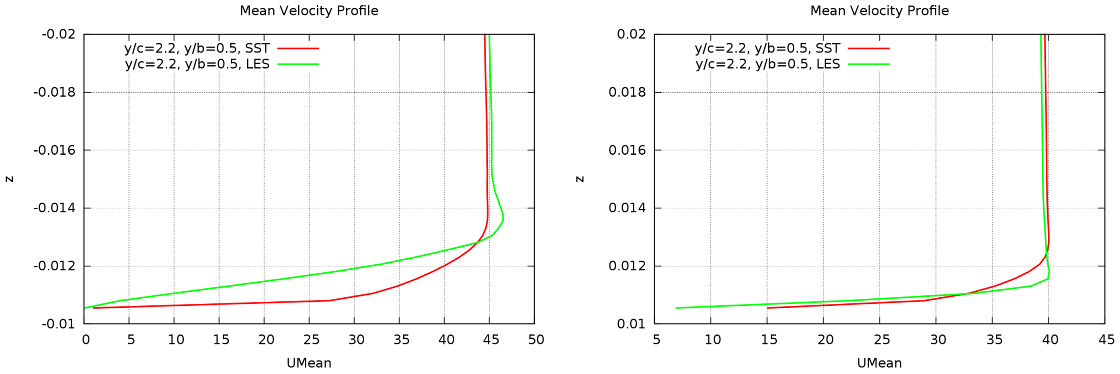


FIGURE 7.33: Velocity profile normal to the wing surface calculated at *Probe*2. Left : suction side, Right : pressure side

On the other hand, near the wing tip it is observed that the mean velocity grows almost at same rate in the boundary layer for both the models, but eventually for LES model gets higher than the SST model as we move away from the wing surface in normal direction. This holds true for both pressure as well as suction sides of the wing. See Figures 7.34, 7.35 and 7.36. As we move from leading edge towards the trailing edge the velocity induction reduces downstream. Also, in all of the Figures, the total velocity induction at the suction side is higher than that at the pressure side, which should be obvious thereby confirming the lifting characteristics of the wing.

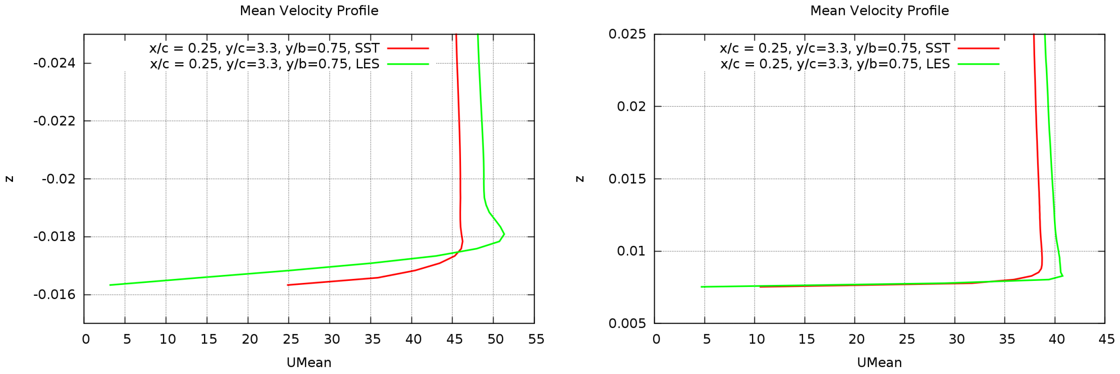


FIGURE 7.34: Velocity profile normal to the wing surface calculated at *Probe*3. Left : suction side, Right : pressure side

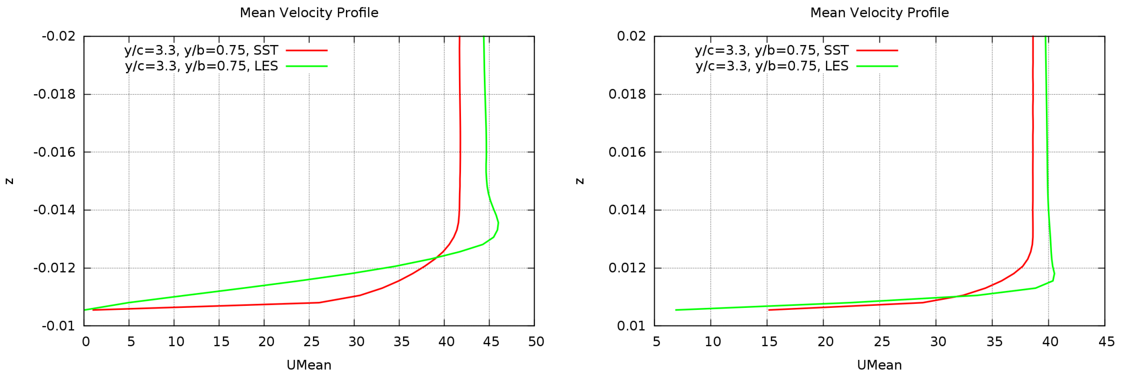


FIGURE 7.35: Velocity profile normal to the wing surface calculated at *Probe*4. Left : suction side, Right : pressure side

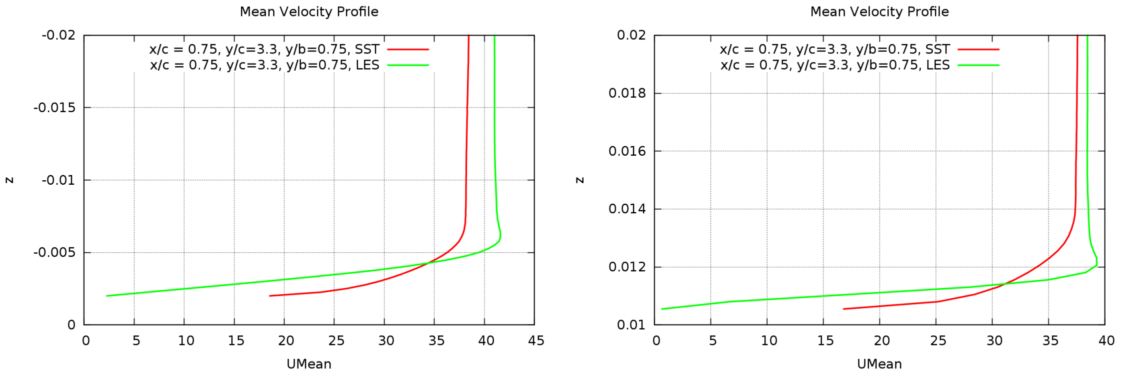


FIGURE 7.36: Velocity profile normal to the wing surface calculated at *Probe*5. Left : suction side, Right : pressure side

7.3.11 Velocity isosurfaces

Figure 7.38 shows the comparison between velocity iso-surfaces for main three different types of modeling - RANS model, LES model and w/o Turb. model. It clearly shows the differences between the models in solving the tip vortex in a qualitative manner.

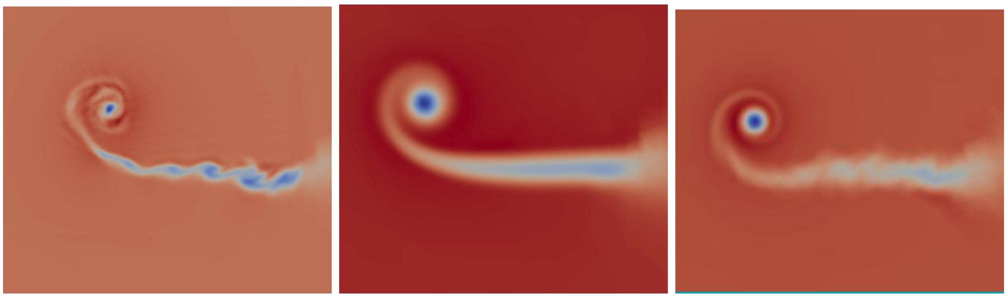


FIGURE 7.37: Velocity iso-surfaces at $x/c = 5$. Left : w/o Turb. model, Middle : SST model Right : LES model

7.3.12 Structure of wing-tip vortex

Figures 7.38 shows the development of the wing tip vortex from the leading edge of the wing and Figure 7.39 shows the structure of the wing tip vortex until downstream $x/c = 30$ alongwith velocity distributions at different stations along the tip vortex. The model referred here is the w/o Turb. model.

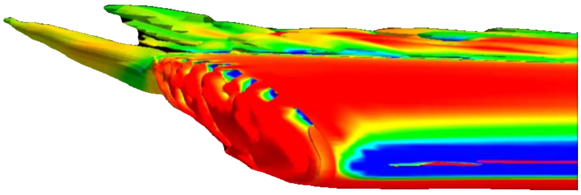


FIGURE 7.38: Development of the wing-tip vortex

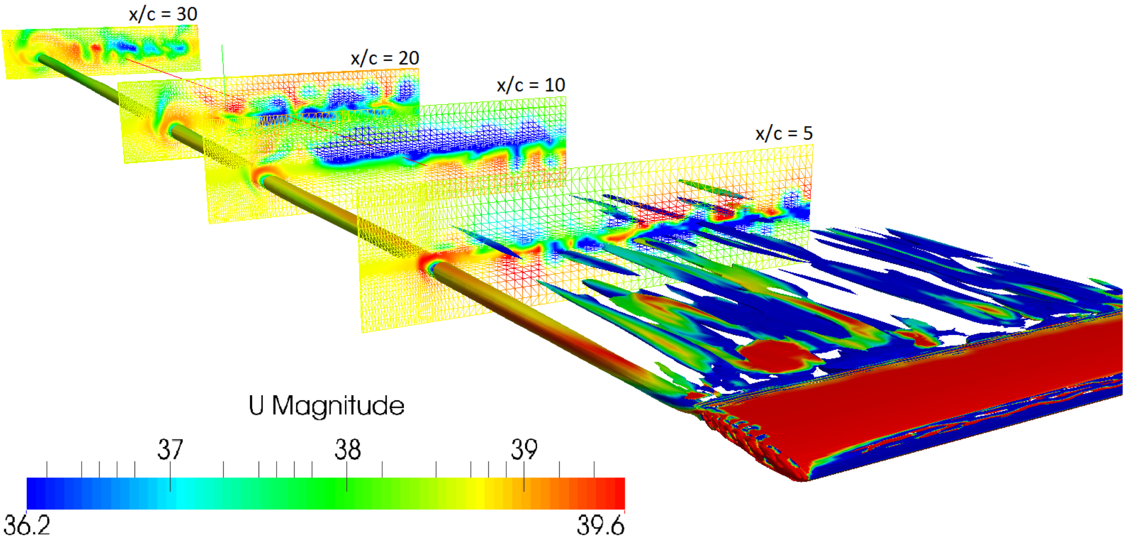


FIGURE 7.39: Structure of the wing-tip vortex

7.3.13 Grid Free VIC Simulations using CVM

The Poisson equation (7.3) is solved with fourth order CDS approximation within the computational domain ranging in the downstream direction from $x/c = 4$ to $x/c = 34$ using FFT method instead of direct summation using the Biot-Savart integral. The cross section of the domain, which corresponds to this in OpenFOAM simulations, is ranged from the symmetry plane of wing $y_0 = 0$ to $y_1/c = 6$ in spanwise direction and from $z_0/c = -3$ to $z_1/c = 3$. The same Monaghan's interpolation formula M_4' was implemented for both the redistribution of vorticity between nodes and for interpolation of velocities from grid nodes to vortex particle positions. This allows one to avoid non physical self induced motion of vortex particles. It was shown in series of special tests that the vorticity is conserved in the remeshing procedure and there is no lost of vorticity at the interface between the vortex method B and OpenFoam A domains.

$$\Delta\vec{u} = -\nabla \times \vec{\omega} \tag{7.3}$$

The solution of the Poisson equation is influenced by the size of computational domain as well as by the boundary conditions. In our methodical investigations in Section 7.3.1 it was confirmed that the zero Neumann conditions ($\partial\mathbf{u}/\partial n = 0$) enforced on the side faces of the domain are more flexible and robust than the zero Dirichlet ones ($\mathbf{u} = 0$). A quite acceptable results were obtained for the computational domain with the domain cross section of $9c \times 9c$ centered around the tip vortex axis at the inlet.

Results are illustrated in Figure 7.40. The VIC method was implemented with and without remeshing. The VIC solution without remeshing is irregular because the VIC technique in its classic form has no difference between the singular and smoothing (mollified) vortices. The total particle vorticity is redistributed among nodes and neither mollification nor smoothing are utilized. However, the averaged results (fitting red line) show that the VIC without remeshing is non diffusive. The tangential velocity around the tip vortex remains almost constant in the wake within the range $0 < x/c < 30$, where c is the wing chord. Although the remeshing makes computations stable, it causes a strong diffusion and can not be accepted (see blue line with remeshing at each 3rd time step). A common way to reduce the diffusive influence of remeshing is the refinement of the remeshing grid. However, a strong limitation for this way is the use of uniform grids which are non- avoidable because the Monaghan's formula is applicable only for uniform grids and determination of velocities from the Poisson equation is commonly based on the Fast Fourier Transformation. The refinement should then be done uniformly in the whole domain including areas without any vortices. Unfortunately, the domain boundaries can not get closer to the tip vortex because zero boundary conditions (either Dirichlet or Neumann) will be not valid.

An efficient remedy for this problem is the application of set of overlapping uniform grids. A very fine grid is placed around the vortex core, the next one is coarser and covers the outer part of the tip vortex, and, finally, the coarse grid is used to represent the large outer area of the computational domain. The zero Neumann boundary conditions are used for the coarsest grid, whereas the boundary conditions for the embedded grid are taken from the solution on the coarser grid. It was proved for two dimensional steady flows and got acceptable results. However, according to our estimations, the grid cell number in this technique goes to a few dozens millions in 3D case. Along with difficulties of remapping of results onto different grids, this makes this technique complex and its competitiveness with respect to common grid based methods raises doubts.

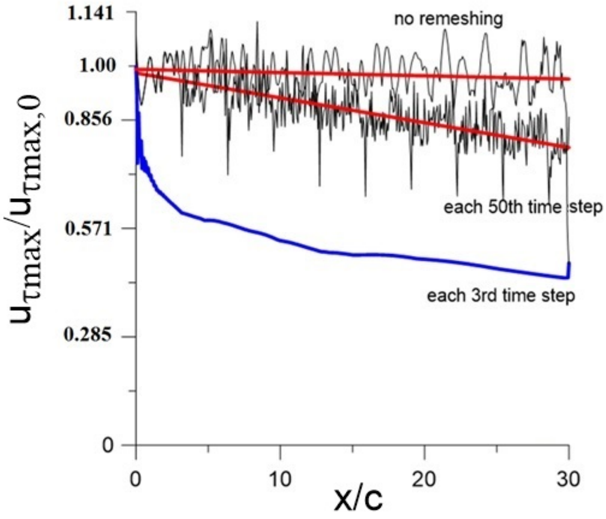


FIGURE 7.40: Maximum tangential velocity around the tip vortex versus the distance from the wing. The velocity is referred to that at the inlet of the vortex method domain. Results are presented for the VIC simulations without remeshing (no remeshing) and with remeshing done at different time steps. Uniform grid with $\Delta/c = 0.06$, where c is the chord.

7.3.14 Grid Free Lagrangian Simulations using CVM

Results obtained using Lagrangian simulation are presented in Figure 7.41 and 7.42. The grid used was uniform with the size $\Delta_{x,y,z} = \Delta$. Velocity field necessary for inlet conditions at the interface between OpenFoam and CVM domain was mapped onto the uniform grid. The ordinary differential equations describing the particle motion are integrated using the predictor- corrector or Euler corrected method with the trapezoidal rule. The time step Δt is chosen so that the particles paths close to the interface within Δt is around Δ, i.e. $\Delta t = min(\Delta/(U_\infty + u_x))$, where u_x is the perturbation velocity induced by vortex and wing. In simulations, presented below, it was around $\Delta t = 1.04.10^{-4}$ s. To reduce the computational time the number of vortex element is limited. When the vortices are identified at the interface F from the OpenFoam solution, only vortices with the vorticity magnitude larger than 10 percent of the maximum vorticity enter into the domain B.

The discrete vortices at the interface F at $x/c = 4$ are determined on the uniform grid with the size Δ using the matching algorithm described above. At $\Delta/c = 2E-2$ the computational domain of the vortex method is occupied by 86500 vortex elements with 61 elements in the tip vortex cross section at $x/c = 4$ (see Figure 7.41). The vortex core parameters presented in Figure 7.42 were obtained by averaging within 0.01 seconds.

The maximum tangential velocity related to that at the interface between the grid based and grid free computational domains is presented. A similar ratio is presented for the vortex core radius. It should be noted, that, despite a discrete representation of vorticity by vortex elements, the distribution of the velocity and radius remain smooth and regular up to the end of the computational domain excepting a very short initial range close to the interface. As seen from Figure 7.42 the radius of the tip vortex is slightly grows by fourteen percent along the whole computational domain at $4 < x/c < 34$. The maximum tangential velocity decreases only by seven percent at the end of the domain.

So far this method provides the best results among all studied approaches for the simulation of the tip vortex strength and tangential velocity till 30 chords behind the wing. Unfortunately, the accuracy of the axial velocity prediction is very weak. It shows a strong potential of the grid free method to radically improve the quality of numerical simulation of the tip vortex flow. However, this method has two substantial restrictions. First, since the method has a sufficiently reduced numerical viscosity, the stability of the pure Lagrangian simulation can be critical for strong concentrated vortices at large Reynolds numbers. Second, improvement of the tip vortex resolution by increase of particle number is limited due to high computational costs when the particle number increasing.

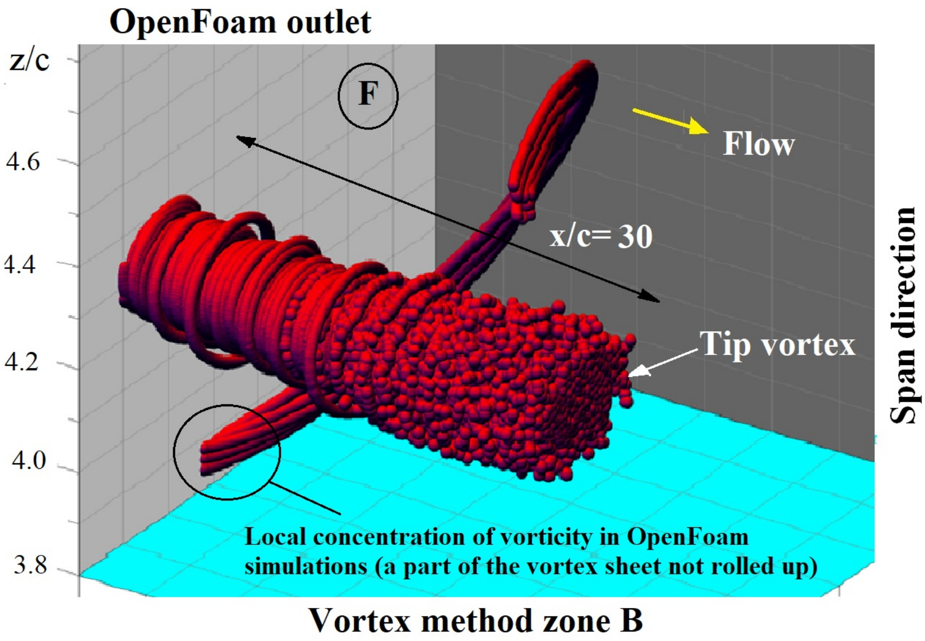

OpenFoam outlet
z/c
4.6
4.4
4.2
4.0
3.8
F
Flow
x/c= 30
Tip vortex
Span direction
Local concentration of vorticity in OpenFoam simulations (a part of the vortex sheet not rolled up)
Vortex method zone B

FIGURE 7.41: Instantaneous distribution of vortex elements in grid free Lagrangian simulation. The radius of bubbles is proportional to the vortex element strength.

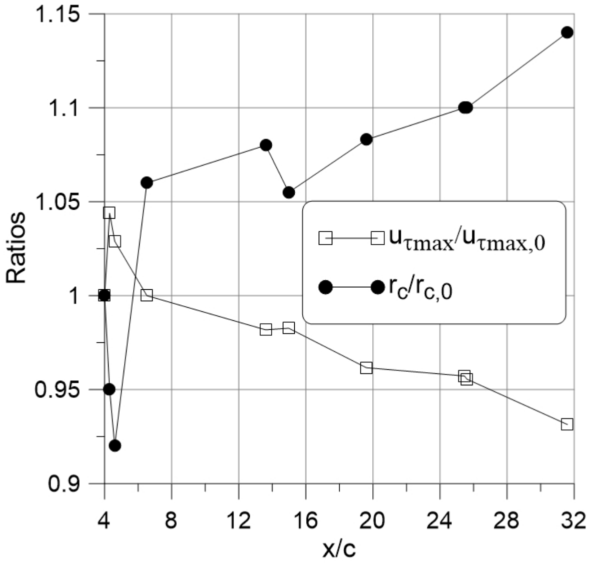


FIGURE 7.42: Results of Pure Lagrangian Simulations. r_c is the radius of the tip vortex, $u_{\tau max}$ is the maximum tangential velocity. The lower index 0 stands for the value at the inlet of the vortex method computational domain.

7.3.15 Coupled method simulations

Interface matching between solutions of grid based and grid free domains

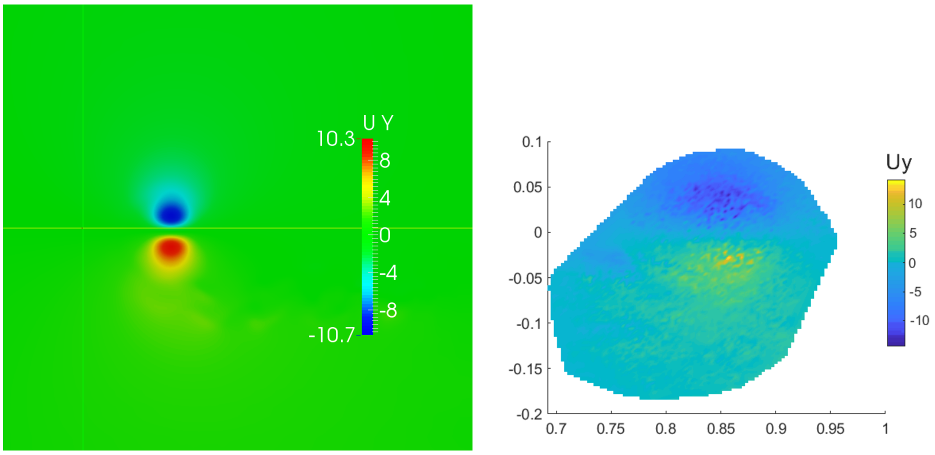


FIGURE 7.43: Interface matching between solutions of grid based and grid free domains at x/c 5 in coupled method simulation

The Figure 7.43 shows on the left the tangential velocity distribution on the interface with grid based solution and on the right the velocity distribution of the vorton cloud after it has traversed a small distance of one grid cell size in downstream from the interface in the grid free part of the domain. The grid free part of the domain is solved in Pure Lagrangian way, thereby has no physical existence of any grid. The Figure on the right is therefore obtained with additional post-process coding with

MATLAB by mapping the properties of the vorton cloud onto an uniform grid density plane and then plotting an iso-surface to visualise the distribution as shown. It is clear on comparing two figures that the tangential velocity of grid based solution and grid free solution match each other at the interface. This matching has been obtained due to the scaling algorithm taking into effect as explained already in the section of Solver Settings.

The coupled method simulation operates as Finite Volume in the grid domain coupled with Pure Lagrangian in the grid free domain. The previous results gained with grid based methods showed that the vortex decay is large as compared to the experimental result with the approach w/o Turb. modeling being the best one. The reason for that is the inherent numerical viscosity due to discretisation on the grid. Results obtained using coupled method are described in the Figures 7.46 and 7.47. The velocity field required for the inlet condition of the interface between the two domain is taken from the already well converged and validated grid based simulations at $x/c = 5$ discussed in previous sections. The interface matching is shown in Figure 7.43. The details of the numerical setup used has already been described in the Section 7.2.3. Figure 7.44 shows the time history for force coefficients C_L and C_D of coupled method simulation depicting that the solution was stable and converged. Figure 7.45 on the left shows the instantaneous positions of the center of vortex elements located within the grid free side of the computational domain while on the right shows the vortex elements are highly correlated with their vorticity vectors aligned along the vorticity vector of main tip vortex. Figure 7.46 shows the instantaneous distribution of vortex elements in grid free domain with the number of vortex elements of approximately ≈ 110000.

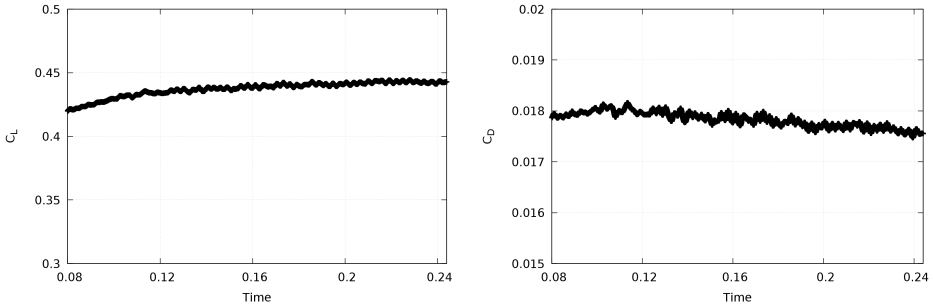


FIGURE 7.44: Time history of force coefficients C_L and C_D for coupled simulation

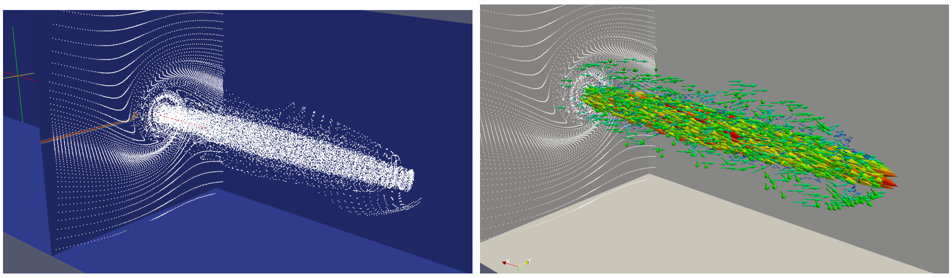


FIGURE 7.45: Left: Instantaneous positions of the center of vortex elements. Right : The co-related vorticity vectors of vortex elements

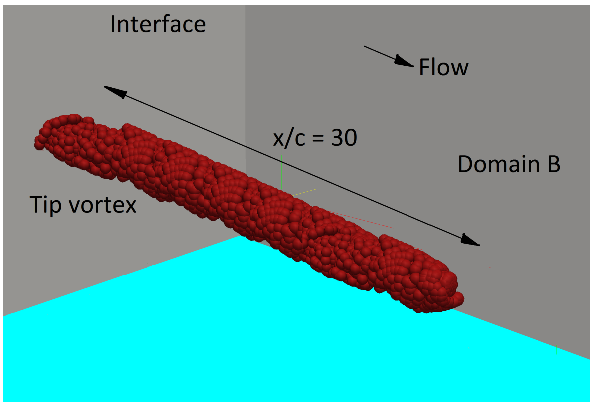


FIGURE 7.46: Instantaneous distribution of vortex elements ≈ 110000 in the grid-free part of the domain in coupled method simulation.

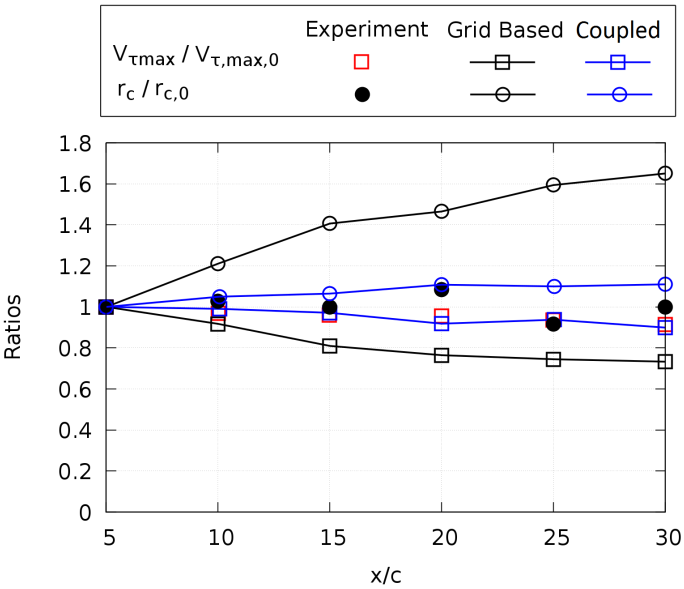


FIGURE 7.47: Results of coupled method simulations. r_c is the radius of the tip vortex, $V_{\tau max}$ is the maximum tangential velocity.

The maximum tangential velocity normalised by the maximum at the interface between the grid based and grid free computational domains is shown in the Figure 7.47. Also in a similar way, the normalised vortex core radius ratio is also shown in the same Figure. The vortex core radius seems to grows upto 1.12 times by the end of the domain in coupled method which agrees closely with the experimental data where it almost remains constant while in the case of pure grid based method grows upto 1.65 times. The maximum tangential velocity decreases by 11 percent at the end of the domain in coupled method which agrees to the experimental data while in case of pure grid based method decreases by 26 percent. This depicts a clear advantage of using a coupled method over the pure grid based method for simulation of tip vortex in the far wake. So far this is the best result obtained among all the approaches studied in this work.

CHAPTER 8

Validation of methods for Rotating Body case

Contents

In the previous Chapter, the domain decomposition method proved to be the best for resolution of the wing tip vortices and preserving the tip vortex strength in the far wake as compared to the Pure grid approaches. This Chapter verifies the same fact for the rotating tip vortices. Two tests are considered within this framework - Openwater propeller and wind turbine. The results obtained from simulation are compared with the BEMT (Blade Element Momentum Theory)for both and with experiment only for wind turbine. There are particularly two aims of this Chapter -

1. To prove the accuracy improvement with coupled method as compared with Pure grid based method in predicting the rate of decay of vortex strength in far wake.

2. To prove the stability of the developed vortex code for unsteady rotating flows

8.1 Openwater Propeller case

8.1.1 Geometrical Setup

The propeller used is a standard propeller. It is type of a conventional propeller with design corresponding to Jess-T404B propeller by Mercury. Following are some geometrical parameters in the Table 8.1 below -

Parameters	Value
Diameter	0.223 m
Pitch ratio, $r/R = 0.7$	1.53
Chord length, $r/R = 0.7$	0.09 m
Area ratio	0.76
Hub ratio	0.24
Number of Blades	4

TABLE 8.1: Geometrical parameters of propeller

The computational domain for the simulation is shown in the Figure 8.1. The propeller is placed with its axis as x-axis. The domain is extended about 2.7D in spanwise direction and 4.5D in longitudinal direction. The propeller is placed at ≈ 0.9D from the inlet while the domain extends about 3.6D in the downstream from the propeller.

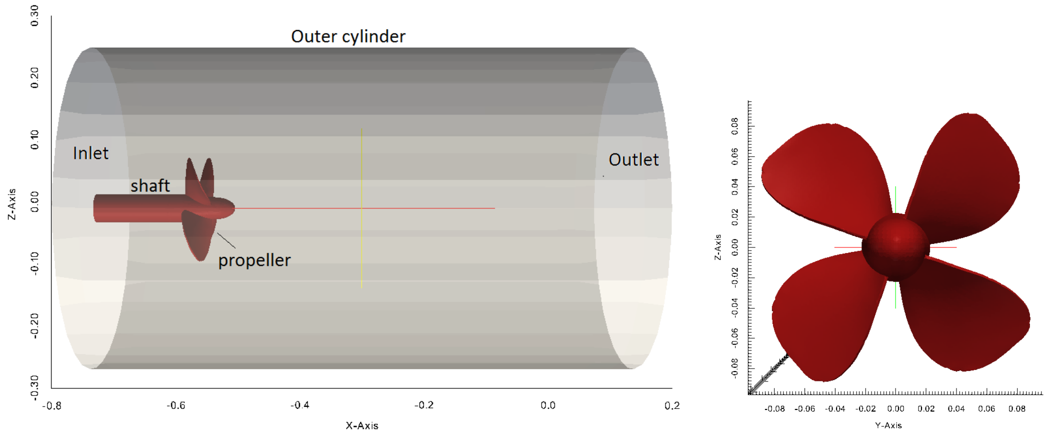


FIGURE 8.1: Computational domain for propeller simulation

8.1.2 Rotating Grids

The grid used for computation of the grid based solution is shown in the Figure 8.2. The background mesh i.e. stator mesh was generated using the blockMesh utility of OpenFOAM while the rotor mesh and the zonal mesh was generated using the SnappyHexMesh. The total number of grid cells was 1M. The average Y^+ lies in the buffer zone.

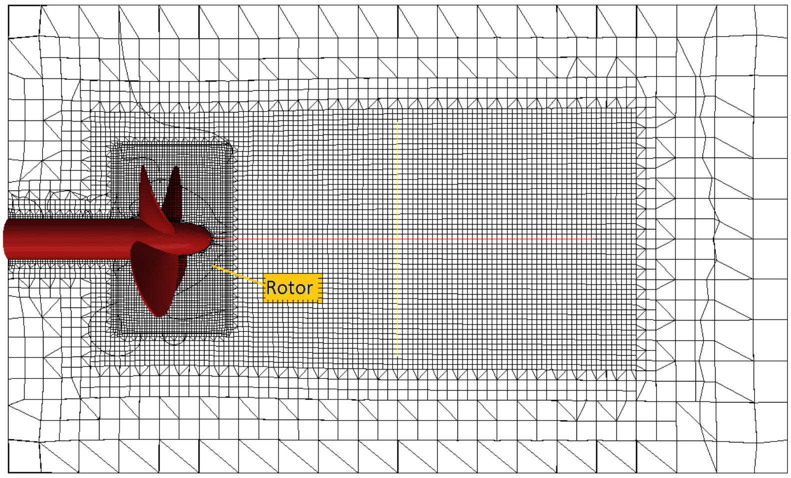


FIGURE 8.2: Computational grid for propeller simulation

8.1.3 Solver settings and Boundary conditions for Coupled Method simulations

The flow was studied at an advance ratio of $J = 0.892$. The propeller rotates with $n = 25/s$ and the inlet flow is at $U_\infty = 5m/s$. The boundary condition used for the propeller is movingWallVelocity where the flux is corrected due to the mesh motion taking into account that the total flux across the moving wall is zero. The simulation was carried out using the kEpsilon model in the grid based part and Pure Lagrangian in the grid free part. As average Y^+ lies in the buffer zone, so wall functions are used for k and ϵ. The interface between the rotor and stator was solved using sliding mesh technique and therefore cyclicAMI boundary condition was imposed. This is a special boundary condition which utilizes the idea of grid to grid interpolation in a conservative way using Galerkin projection and this enables us to simulate across stationary stator and rotating rotor domains [173]. No-slip boundary condition for velocity is imposed on the domain outer cylinder. The zero gradient BC is enforced at the outlet while the inlet is kept at fixed velocity U_∞. The interface between the grid and grid free domain is located at 0.9D downstream of the propeller.

The temporal discretisation was done using Euler method. The convective term of the momentum equation is discretised using linear upwind scheme. The simulation was carried out using the dynamic version of the PIMPLE solver with two outer corrector loops within each time step. The maximum Courant number is 2 in all the simulations. The time step was $\Delta t = 10^{-5}$. In each time step the propeller rotates by 0.0904 degrees. The simulation was carried out for a total time of 0.3 sec with the flow pass atleast twice the domain before the simulation was stopped once the mean vortex flow was converged. All the strategy discussed in section 7.2.3 were applied here in a similar way to obtain proper, stable and converged results for the coupled simulations.

8.1.4 Numerical results and validation

In the following section, we shall compare the results of the coupled method simulations with those of pure grid based simulations. Figure 8.3 shows the vortex structures ($\lambda_2 < 0$) in a pure grid based simulation. It could be seen that the vortex dissipates after a certain distance. This happens due to the numerical viscosity induced by the grid.

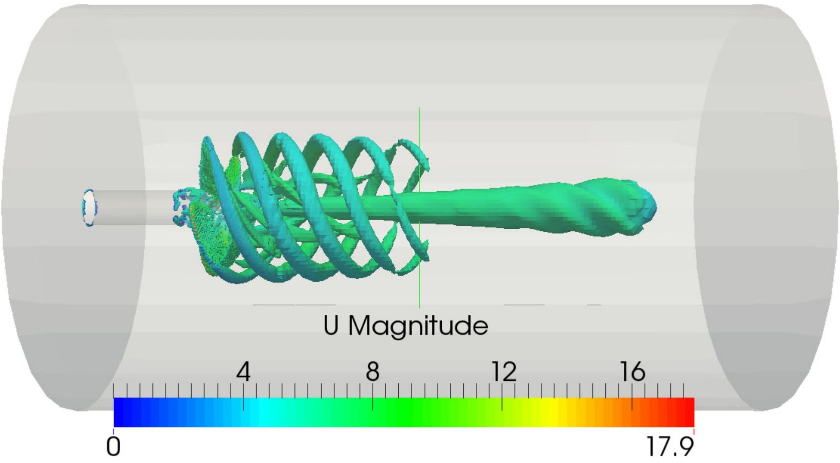


FIGURE 8.3: Propeller tip vortex structures using Pure grid based simulation

Figure 8.4 shows the residuals of coupled simulation while Figure 8.5 shows the variation of thrust and torque of the propeller in time in coupled simulation. It could be clearly seen that the residuals are converged and stable. Figure 8.5, the fluctuations in the values of thrust and torque show a repeated pattern within small amplitude indicating flow stability. The peaks in all the below plots appear due to intermediate pause and restart of the simulation and not to be worried about. Observing the convergence of residuals as well as forces fluctuations in time show that the developed coupled simulation algorithm faces no instability.

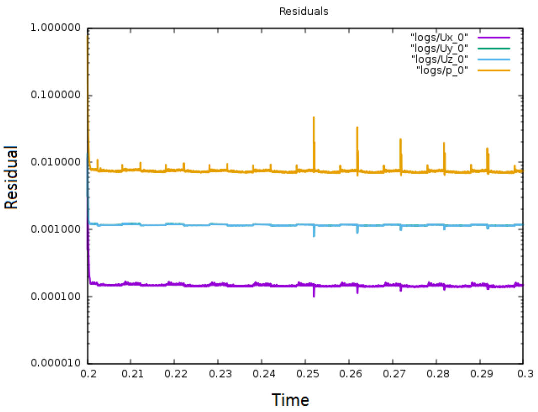


FIGURE 8.4: Convergence of residuals in the coupled simulation. (Propeller case)

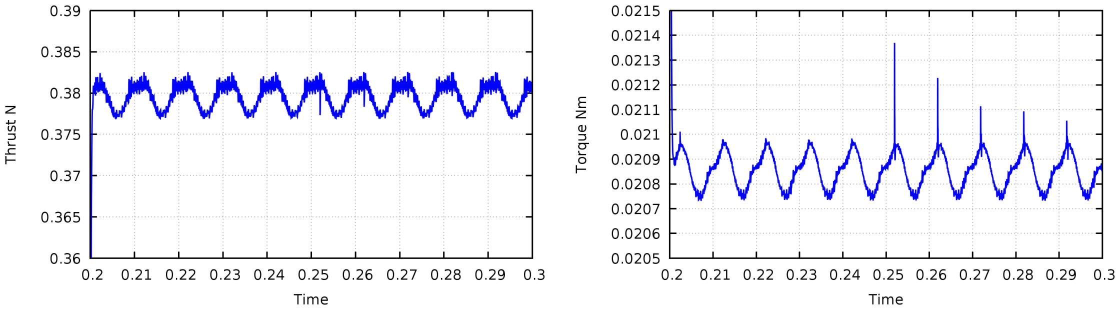


FIGURE 8.5: Convergence of forces and moments in the coupled simulation. (Propeller case) Left : Thrust, Right : Torque.

Below Figures 8.6 and 8.7 show the pressure and velocity distributions on the surface of the propeller for pure grid based, 1-way coupled and 2-way coupled simulation. The interface between the grid based and grid free domain is located downstream at 0.9D from the propeller. Although, it is rare to expect any changes in pressure and velocity distributions on the surface of the propeller due to downstream flow but in simulation like 2-way coupled where the solution of vortex domain B affects the boundary condition of the grid domain A, thereby affecting the solution in the grid domain A, this could be verified. It could be seen from the Figures that the peak values of pressure and velocities agree with each other. The tip speed ratio λ evaluates to 3.53 in the simulation which is very close to the analytical tip speed ratio $\lambda_{analytical} = 3.58$

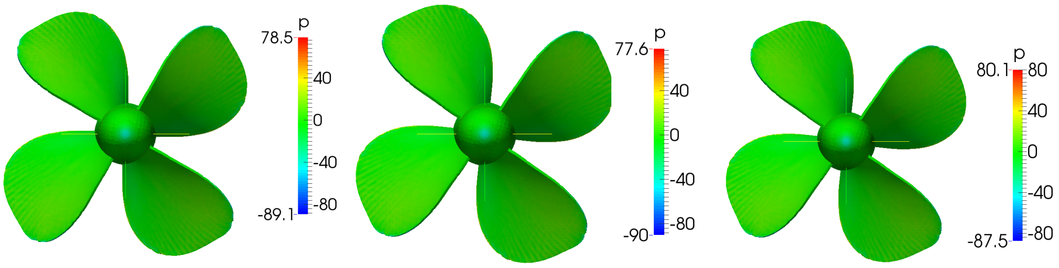


FIGURE 8.6: Pressure distributions on the surface of propeller. Left : Pure grid, Middle : 1-way coupled, Right : 2-way coupled

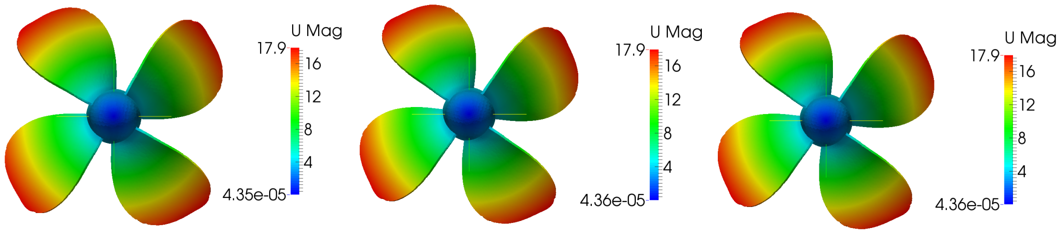


FIGURE 8.7: Velocity distributions on the surface of propeller. Left : Pure grid, Middle : 1-way coupled, Right : 2-way coupled

The results of the coupled method simulation are compared against the BEMT. BEMT is a very popular method used for blade analysis. The method is based on two theories - momentum theory and blade element theory. In momentum theory, a momentum balance is calculated on a rotating annular stream tube passing through the turbine while in blade element theory, the forces generated are examined by the airfoils lift and drag coefficients at various sections along the blade. Finally using these two, a series of equations are sought and solved iteratively and performance parameters are determined by integration along the span of the blade. The results of simulation are compared with the BEMT in the following Table 8.3 -

Simulation	J_s	Thrust N	Torque Nm	K_t	$10 * K_q$	η
Pure grid	0.892	0.376	0.021	0.196	0.49	0.569
1-way coupled	0.892	0.374	0.021	0.195	0.50	0.551
2-way coupled	0.892	0.380	0.022	0.198	0.50	0.559
BEMT	0.892	0.396	0.0185	0.207	0.43	0.67

TABLE 8.2: Comparison of thrust, torque and coefficients obtained by different methods with the BEMT analysis for coupled method simulations. (Propeller case)

It could be seen from the Table that the results of coupled method simulations are in good agreement with the BEMT for the thrust while less in agreement for the torque. BEMT method largely depends upon the number of element sections considered along the blade, the more number of sections the more accurate prediction of forces.

This could be one of the reasons for discrepancy. Figure 8.8 shows the decay of the vortex strength measured in the terms of maximum tangential velocity. This velocity is normalised by the maximum at the interface $V_{\tau max,0}$. It could be seen that the vortex strength decreases rapidly with grid based simulation by 47 % at the end of the domain while in the case of coupled simulation it is by 9 % at the end of the domain. Therefore this proves the accuracy improvement of coupled method as compared to pure grid based method in predicting the rate of decay of vortex strength in far wake. The vorticity vectors in the grid free part of the domain is shown in the Figure 8.8 on the right side. Figure 8.9 shows the instantaneous distribution of vortex elements ≈ 3000 in the grid-free part of the domain in coupled method simulation.

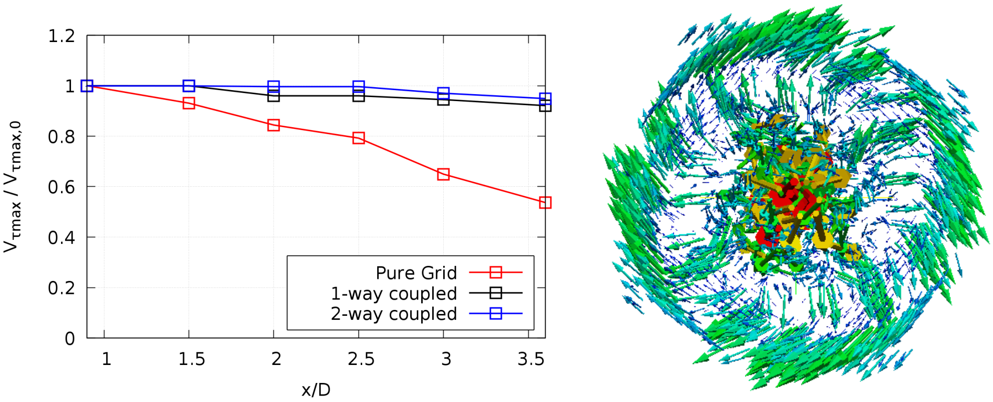


FIGURE 8.8: Left : Results of coupled method simulation of propeller. $V_{\tau max}$ is the maximum tangential velocity. Right : Vorticity vectors in the grid free part of the domain

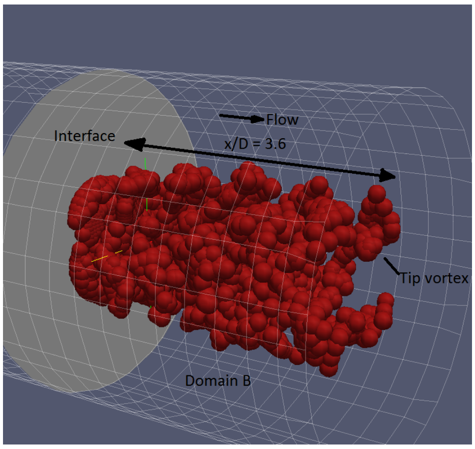


FIGURE 8.9: Instantaneous distribution of vortex elements ≈ 3000 in the grid-free part of the domain in coupled method simulation of propeller

8.2 Rotating Wind Turbine case

8.2.1 Geometrical Setup

In this section, the coupled method simulation for a wind turbine will be discussed. The wind turbine referred is the DOWEC 6MW/NREL 5MW turbine. Both turbines posses the same blade profile and construction just the operational parameters are different due to their respective power ratings. DOWEC stands for Dutch Offshore Wind Energy Converter [174]. It is an offshore wind turbine. The turbine consists of three blades. The full scale turbine has a rotor diameter of 129 m, rated rotor speed of 11.36 rpm at rated wind speed of 10 m/s. It is geometrically complex turbine. Eight different airfoil stack together along the span of the blade to render its geometry. They are from tip to root: NACA-64-418, DU21 A17, DU25 A17, DU30 A17, DU35 A17, DU40 A17 and 2 cylinders. Details of this blade can be found in [174] [175]. We have used the zero pitch version of the turbine.

For simplicity of computation, a scaled model was developed. The necessity for this will be discussed in details in the Computational cost analysis section in the Chapter 9. From the research review, it was noticed that a pure geometrical scaling of such turbine is not effective due to very famous Reynolds scaling effect. Therefore Froude scaling law is being applied to develop the scaled version of the turbine. This approach is also recommended in [176] [177] and by the Wind Energy Department of our University. The scaling factor was ≈288.

The computational domain for the simulation is shown in the Figure 8.10. The wind turbine is placed with its axis as x-axis. The domain is extended about 2.8D in spanwise direction and 4.6D in longitudinal direction. The wind turbine is placed at ≈ 1.4D from the inlet while the domain extends about 3.2D in the downstream from the wind turbine.

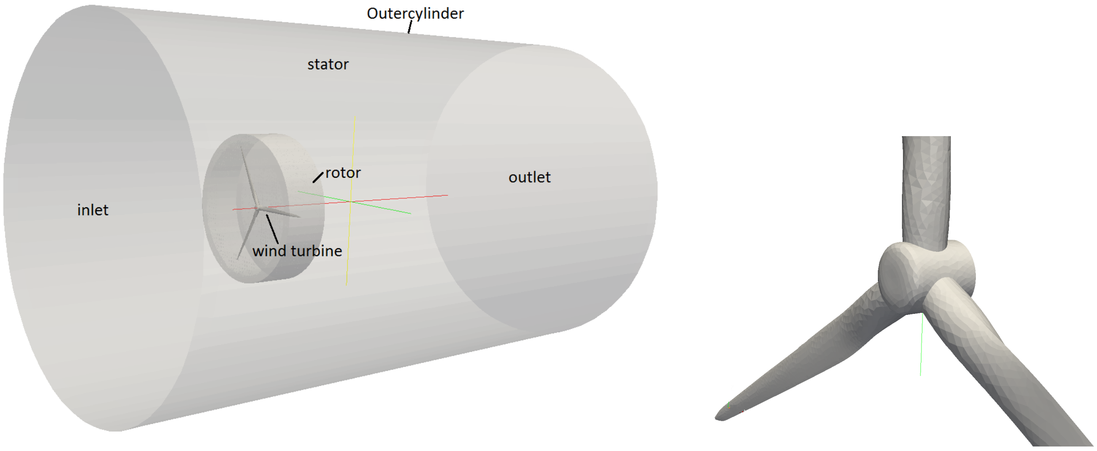


FIGURE 8.10: Computational domain for wind turbine simulation

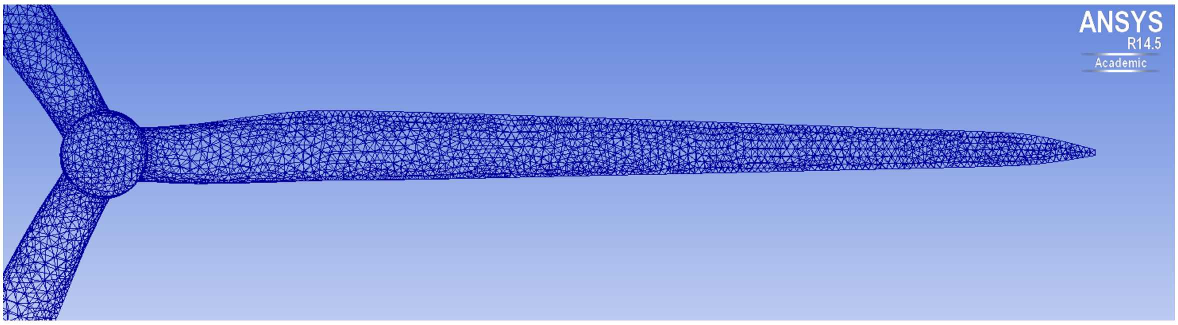


FIGURE 8.11: Blade of the Turbine

8.2.2 Rotating Grids

The grid used for computation of the grid based solution is shown in the Figure 8.12. The background mesh i.e. stator mesh was generated using the blockMesh utility of OpenFOAM while the rotor mesh and the zonal mesh was generated using the SnappyHexMesh. The total number of grid cells was 2.7M.

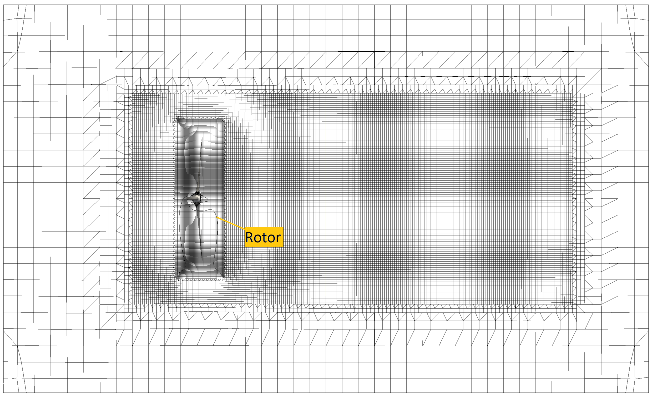


FIGURE 8.12: Computational grid for wind turbine simulation

8.2.3 Solver settings and Boundary conditions for Coupled Method simulations

The following section describes the numerical setup for coupled method simulation. The inlet flow is at $U_\infty = 0.589 m/s$. The turbine rotates at 20.17 rad/s. The boundary condition used for the turbine is movingWallVelocity. The simulation was carried out using the w/o Turb. model in the grid based part of the domain and Pure Lagrangian in the grid free part of the domain. The interface between the rotor and stator was solved using cyclicAMI boundary condition. No-slip boundary condition for velocity is imposed on the domain outer cylinder. The zero gradient BC is enforced at the outlet while the inlet is kept at fixed velocity U_∞. The interface between the grid and grid free domain is located at 0.7D downstream of the wind turbine.

Influence of interface position - The position of the interface is important while solving a coupled method. When the interface is too close to the turbine, the outlet boundary of the grid domain has an effect on the solution near the turbine which is not desired. The position of the interface shall vary application to application. By doing some a-priori simulations it could be estimated for a particular case. In our case, we observed that the interface position > 0.5D was sufficient enough to shield the effect of the outlet boundary on the near blade field.

The temporal discretisation was done using Euler method. The convective term of the momentum equation is discretised using linear upwind scheme. The simulation was carried out using the dynamic version of the PIMPLE solver with two outer corrector loops within each time step. The maximum Courant number is 2 in all the simulations. The time step was $\Delta t = 10^{-5}$. In each time step the turbine rotates by 0.0115 degrees. The simulation was carried out for a total time of 12.4 sec with the flow pass atleast 3.5 times the domain before the simulation was stopped once the mean vortex flow was converged. All the strategy already discussed in the section 7.2.3 was applied to obtain stable and converged result for the coupled simulations.

8.2.4 Numerical results and validation

In the following section, we shall compare the results of the coupled method simulations with those of pure grid based simulations. Figure 8.13 shows the vortex structures ($Q > 0$) in a pure grid based simulation. Due to numerical viscosity, the vortex dissipates in the far field away from the turbine.

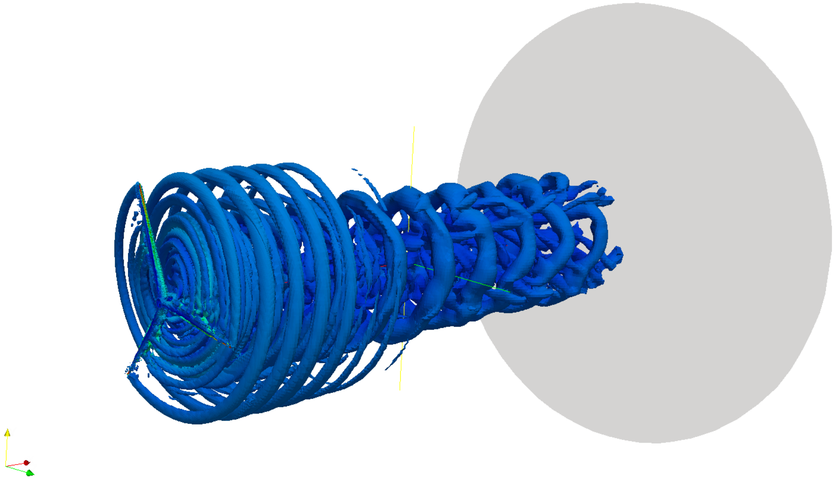


FIGURE 8.13: Wind turbine tip vortex structures using Pure grid based simulation

Figure 8.14 shows the residuals of coupled simulation while Figure 8.15 shows the time history of axial force and torque of the turbine in coupled simulation. It is clear that the residuals are converged and stable. Figure 8.15, the values of axial force and torque fluctuate within small amplitude interval indicating flow stability. From the convergence of residuals and the fluctuations of the the axial force and torque in time it can be concluded that the developed vortex code faces no instability, since in 2-way coupling the vortex domain also has an effect back on the OpenFOAM domain boundary condition, in other words on the solution in the grid domain as well.

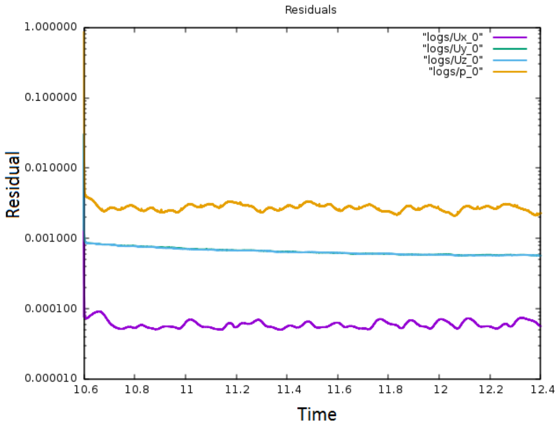


FIGURE 8.14: Convergence of residuals in the coupled simulation. (Wind turbine case)

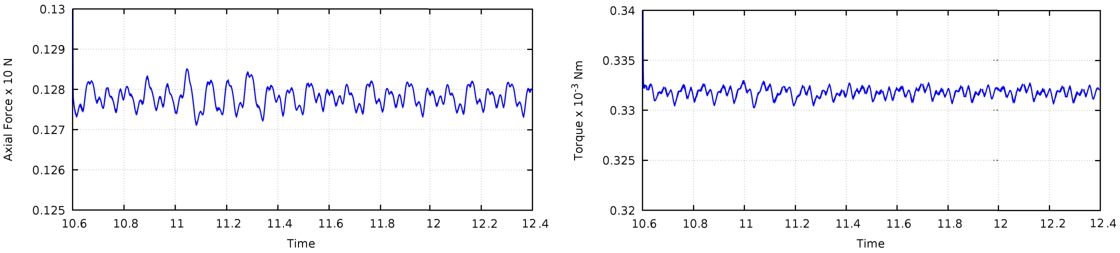


FIGURE 8.15: Convergence of forces and moments in the coupled simulation. (Wind turbine case) Left : Thrust, Right : Torque.

Below Figures 8.16 and 8.17 show the pressure and velocity distributions on the surface of the turbine for pure grid based, 1-way coupled and 2-way coupled simulation. The turbines appear rotated since it is their respective positions at the end of the simulation. Figures show that the peak values of pressure and velocities agree with each other. The tip speed ratio TSR evaluates to 7.58 in the simulation which is very close to the analytical tip speed ratio $TSR_{analytical} = 7.43$

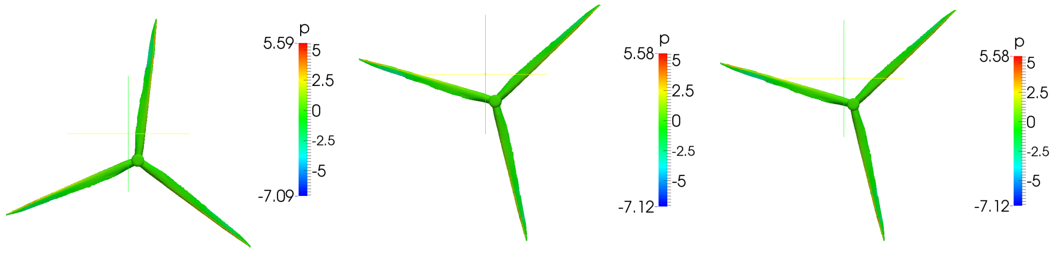


FIGURE 8.16: Pressure distributions on the surface of wind turbine. Left : Pure grid, Middle : 1-way coupled, Right : 2-way coupled

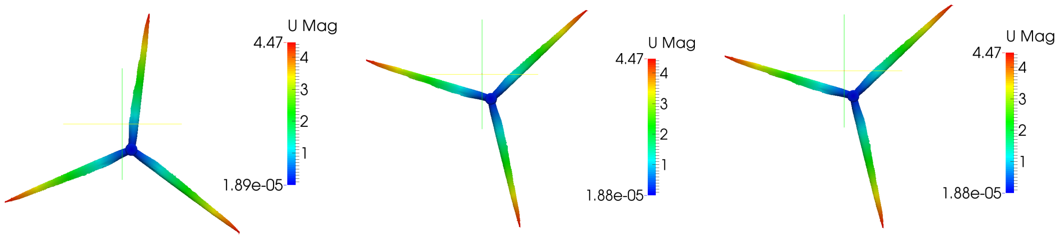


FIGURE 8.17: Velocity distributions on the surface of wind turbine. Left : Pure grid, Middle : 1-way coupled, Right : 2-way coupled

The simulation results of the simulation are validated against the BEMT. Below Table 8.3 shows that the results of coupled method simulations are in good agreement with the BEMT for the axial force and less in agreement for the torque.

Simulation	Axial force ×10 N	Torque $\times 10^{-3}$ Nm
Pure grid	0.1273	0.330
1-way coupled	0.1279	0.331
2-way coupled	0.1280	0.332
BEMT	0.1281	0.2946

TABLE 8.3: Comparison of thrust, torque and coefficients obtained by different methods with the BEMT analysis. (Propeller case)

Analysis of vortex wake downstream of the wind turbine in coupled method simulation

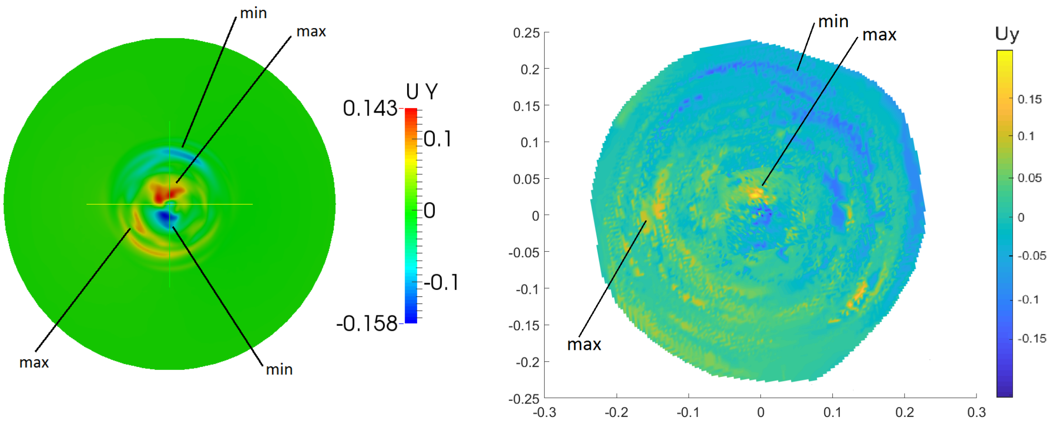


FIGURE 8.18: Detailed Velocity field inside of the vortex wake of the wind turbine in coupled simulation at x = 0.8D. Left : From grid based solution, Right : From grid free solution

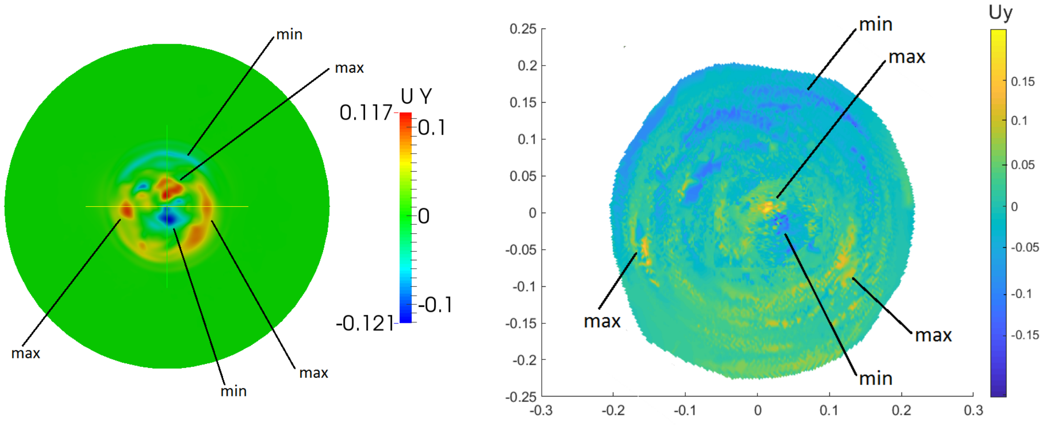


FIGURE 8.19: Detailed Velocity field inside of the vortex wake of the wind turbine in coupled simulation at x = 1.6D. Left : From grid based solution, Right : From grid free solution

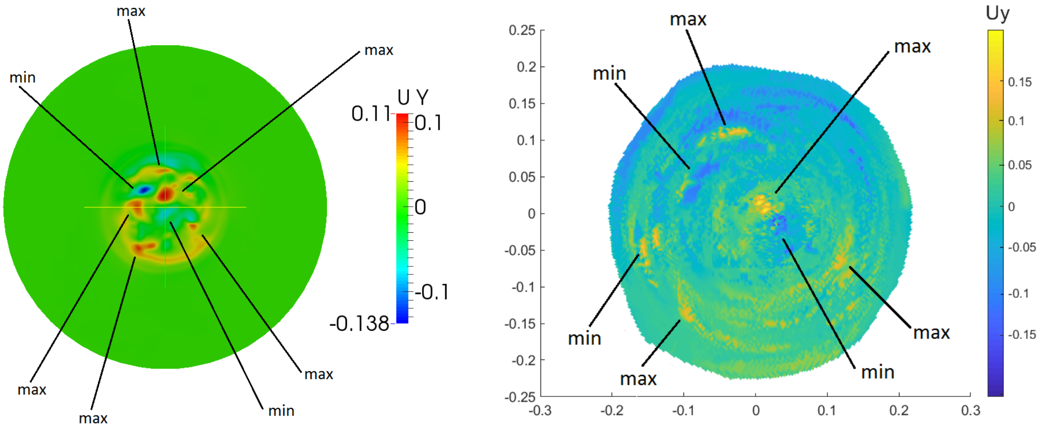


FIGURE 8.20: Detailed Velocity field inside of the vortex wake of the wind turbine in coupled simulation at x = 3.1D. Left : From grid based solution, Right : From grid free solution

Figures 8.18, 8.19 and 8.20 show the distribution of tangential velocity in the wake downstream of the turbine at locations x/D 0.8, 1.6 and 3.1 respectively. The left side Figures show the tangential velocity distributions due to Pure grid based solution while on the right due to coupled method simulation. It should be noted here that the grid free part operates in Pure Lagrangian way, therefore has no physical grid present. The Figure on the right is obtained by a external post-processing done in MATLAB by mapping the tangential velocities of the vorton cloud onto an uniform grid density plane and visualising it by an iso-surface. The right figures show only the middle part of the circular plane of size of the diameter of the rotor. The vortex strength decay faster in grid based solution as compared to coupled simulation. The vortex wake shows the existence of regions of maximum and minimum vortex strength distribution. The coupled method is able to show not same but similar kind of existence of regions of maximum and minimum vortex strength distributions in the wake at 0.8D, 1.6D and 3.1D as compared to the grid solution, only in the case of grid based solution they are of reduced strength due to artificial viscosity effects.

The interface between the grid based and grid free domain is located downstream at 0.7D from the turbine. Figure 8.21 shows the decay of the vortex strength measured in the terms of maximum tangential velocity normalised by the maximum at the interface $V_{\tau max,0}$. It could be seen that the vortex strength decreases strongly with grid based simulation by almost 70 % by the end of the domain while in the case of coupled simulation it is by about just 13 % by the end of the domain.

Validation of coupled method simulation with the experimental data

Most of the experimental work done on wind turbine consists of flow field measurements on the turbine itself or very near field of the turbine. It was difficult to find any experimental study in the far wake of the wind turbine. In our research survey we could find two sets of experimental data - one from the Mexnext project (EU project) [180] and other from the Wing Energy Institute, Technical University Munich TUM

[178] [179]. The TUM work is based on a G1 Model type wind turbine which is a research wind turbine while the Mexnext project is based on a MEXICO model wind turbine which is a scaled down turbine with design characteristics of the DOWEC 6MW/NREL 5MW wind turbine (same wind turbine which is in our case). The major difference between these two experiments is that the MEXICO model experiment provides the flow field measurements in the near wake field upto 1.13D while the TUM experiment provides flow field measurements in the far wake field upto 6D. The MEXICO model is a uniform inlet flow model while the TUM model requires a turbulent inlet field. A simulation work is being submitted and is under review by Wang et. al. in July 2018 [178] [179] from the group of TUM who conducted SAS and LES simulations using precursor domain (59 M cells) to generate the turbulent flow inlet with the main flow domain (39M cells) to solve the G1 model turbine and validated their simulation results with experiment in the far wake.

We have referred to the MEXICO model for our validation instead of TUM G1 model. The following paragraph shall give a clear justification about the necessity for the choice of MEXICO model data.

For the coupled simulation, it is necessary to have a grid based solution in agreement with the experimental data at the interface between the grid domain and grid free domain. One of the major drawback of the coupled method computation is that it works in a serial mode i.e. on a single processor. For instance, in the case of our wind turbine 2.7M, the coupled simulation took on average 1.5 times more clock-Time for each time step than the pure grid solution at first few time steps. As the number of particles increases in the domain with the generation of new vortons, the time of computation per time step also increases. The reason is due to the calculation of induced velocity by Biot Savart Law. The more number of particles, the more computations of Biot Savart Law are necessary in each time step proportional to N^2, where N is number of particles. This increases the total time of computation. For example, in our case of wind turbine simulation with 2.7M cells, it took approximately 1.5 months for a coupled simulation to complete. Also, as the vortex code works in a serial mode, in each time step, sub-loops solve for the grid domain and the grid free domain. In the case of TUM G1 model, it means it would result in solving of a very fine domain ($\approx$ 25M cells) in each time step on a single processor, which is almost practically very difficult. Also due to very dense mesh, the number of particles will be high at even first time step and would increase to millions as the vorton cloud travels further downstream. It is practically not possible to solve such a complex problem on a single processor as of now due to limitations of computational power. Parallelization of the vortex code could be a solution to solve such kind of complex problems. Implementing parallelization is itself a complicated task and requires huge investigation. This is not covered in the scope of this present work. It is a part of the future work.

In this regard, we have validated the rate of decay of vortex strength with the experimental data of the MEXICO turbine model See Figure 8.21 Left. The experiment was conducted on a 3 blade wind turbine with zero pitch of 4.5 m diameter at the Low-speed Facility of the DNW in Netherlands [180]. The flow field measurements were done using the PIV technique. The inlet uniform flow was at 10 m/s. The tip vortex strength decay in experiment was measured in the near field only which has been plotted in the Figure 8.21 [180]. It is common practice in field of wind energy to compare the results of scaled wind turbines which have similar TSR i.e. Tip Speed Ratio and tip blade profile (Similarity criteria). The TSR and tip blade profile of MEXICO turbine and our turbine match together, which makes us the basis to compare our simulation results with the measurements of the MEXICO model.

Parameter	MEXICO Model	Our Turbine
TSR	7.64	7.58
Tip Blade Profile	NACA 64-418	NACA 64-418

TABLE 8.4: Comparison between MEXICO Model and our wind turbine

It could be observed from the Figure 8.21 Left, that the coupled method simulation very well predicts the rate of decay of vortex strength when compared to experimental rate of decay of the MEXICO model.

Also, the Figure 8.21 Right, shows the rate of decay of vortex strength in the experiment of TUM G1 Model turbine for a moderate turbulent case (Turbulence intensity of 6%), there the velocity measured in downstream of the turbine is as follows - at 1.4D $V_t/U_0 = 0.6863$, at 1.7D $V_t/U_0 = 0.6586$, at 2D $V_t/U_0 = 0.6415$ and at 3D $V_t/U_0 = 0.6239$. The vortex strength decays by ≈10 % at x=3D [178] [179] in the experiment of TUM G1 Model turbine. Although the TUM case is not compatible to our case, yet it just gives an idea about the dissipation behaviour of tip vortex in far wake which is comparable to the the dissipation behaviour of tip vortex which we obtain with our coupled simulation.

Figure 8.23 shows the instantaneous distribution of vortex elements ≈ 15400 in the grid-free part of the domain in coupled method simulation while Figure 8.23 shows the instantaneous distribution of vorticity vectors.

It is clear in the case of unsteady rotating flows, that the coupled method simulation shows an accuracy improvement when compared to pure grid simulation in predicting the rate of decay of vortex strength in far wake. And also the developed coupled method algorithm runs stable enough to calculate these kind of rotating flows.

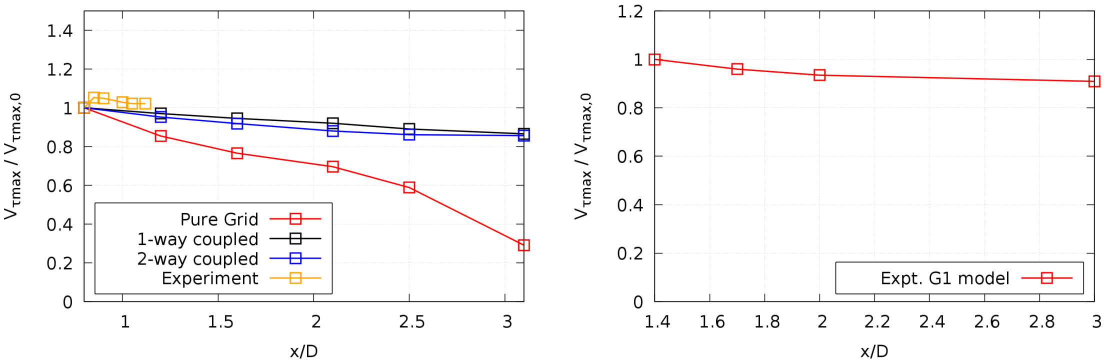


FIGURE 8.21: Left : Results of coupled method simulation of wind turbine and validation w.r.t experimental data of MEXICO turbine. $V_{\tau max}$ is the maximum tangential velocity. Right : Experimental decay of vortex strength for TUM G1 model turbine

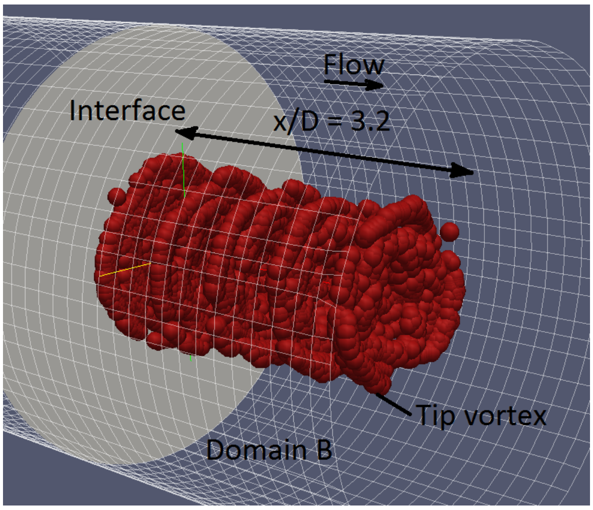


FIGURE 8.22: Instantaneous distribution of vortex elements ≈ 15400 in the grid-free part of the domain in coupled method simulation of wind turbine

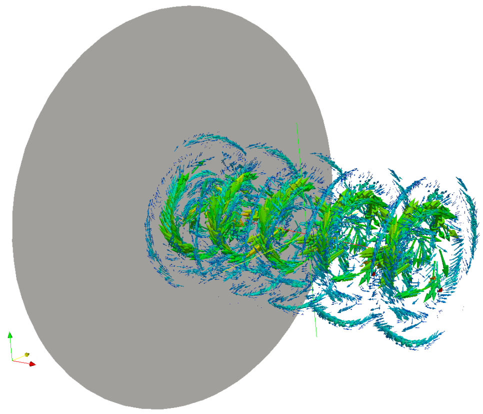


FIGURE 8.23: Instantaneous distribution of vorticity vectors in the grid-free part of the domain in coupled method simulation of wind turbine

CHAPTER 9

Conclusions

Contents

9.1 Conclusions

A proper prediction of the tip vortex evolution is a big challenge for computational fluid mechanics despite of many efforts to solve this classical problem having a big practical importance in aerodynamics and hydrodynamics. In numerical simulations the tip vortex rapidly degrades loosing its strength and spreads up already at distances of a few dozens of the wing chord. As a result, numerical analysis of important practical problems, for instance, of the tip vortex cavitation influence on the rudder erosion with an acceptable accuracy becomes impossible. In this work, we tried to find an efficient way to improve the simulation accuracy using different numerical methods, grids and turbulence models. It was shown that the application of curvature correction (CC) to standard two-equation turbulence models of Spalart Allmaras and SST improved results at a rate which depends on the grid. For the coarse grid the improvement is substantial. For fine grids with large Y^+ and an approximate treatment of the boundary layers using wall functions the influence of the RANS models is negligible. On the contrary, the grids with well resolved boundary layer and small Y^+ show a strong sensitivity to RANS model. The SST model with curvature correction (according to Smirnov and Menter (2009)) has a clear advantage over the corrected Spalart Allmaras model. Also accuracy improvement was observed with the w/o Turb. model for low Re grid. In conjunction with adaptive mesh refinement (AMR) the eddy viscosity models as well as w-o Turb. model showed a significant improvement for fine grid resolution and the lowest error predictions against experimental data as compared to previous works in this field. The highest accuracy among all RANS turbulence models is attained with the

LRR Reynolds stress transport models (RSM) which takes the anisotropy of the turbulent wake into account and the rotation effects are an inherent part of modelling in this technique. The LES model exhibit similar behaviour as of w/o Turb. model. The most promising results among grid based methods were obtained using the approach without turbulence modelling, which can be classified as an underresolved very large eddy simulation without a subgrid model.

The superiority of this solution leads to the conclusion that the used turbulence models do not reproduce the flow relaminarization in the vicinity of the tip vortex core which is the reason for the very slow tip vortex decay. This simulation supports the conclusion of Devenport et al. (1996) that the ”flow in the core is laminar and that velocity fluctuations experienced here are inactive motions produced as the core is buffeted by turbulence from the surrounding wake”.

To further reduce the artificial vortex diffusion, a pure Lagrangian grid free vortex method is applied to simulate the tip vortex dynamics in the far wake. Grid free simulation is coupled with the grid based one in the near wake, where the vortex diffusion can be considered as small. The flow on the wing and in the near wake is treated using grid based method, whereas the remaining flow part is calculated using the grid free approach. The coupled method provides the most promising results showing the lowest rate of the vortex strength decay and vortex core spreading. However, this method has two substantial restrictions. Firstly, since the method has a sufficiently reduced numerical viscosity, the stability of the pure Lagrangian simulation can be critical for strong concentrated vortices at large Reynolds numbers. Second, improvement of the tip vortex resolution by increase of particle number is limited due to high computational costs when the particle number increasing.

Also the validation of the coupled method was carried out for the rotating flows i.e. propeller test case and wind turbine test case. The accuracy improvement with the coupled method over the grid based method for resolution of rotating tip vortices in predicting the rate of decay of vortex strength was justified. Also it proves the stability of the developed algorithm for unsteady rotating flows.

In conclusion, the use of coupled method seems to be very promising way for a radical improvement of the accuracy of tip vortex dynamics prediction.

9.2 Computational Cost analysis and Outline for Future Work

This section describes the computational cost analysis of the new developed coupled method. The coupled method in the present version works only in a serial mode i.e. on a single processor. We compare the computational time consumed by the coupled method simulations in terms of time consumed by the pure grid simulations in the following Table 9.1.

Time interval	Wing case	Wind Turbine Turbine case
First few time steps	average 2.2 times	average 1.5 times

TABLE 9.1: Comparison between computational cost between coupled simulation in comparison with the grid based simulation

In the Devenport case, the coupled method simulation took on average 2.2 times more clockTime for each time step than the pure grid solution at first few time steps. In the case of our wind turbine, the coupled simulation took on average 1.5 times more clockTime for each time step than the pure grid solution at first few time steps. As the number of particles increases in the domain with the generation of new vortons, the time of computation per time step also increases. The reason is due to the calculation of induced velocity by Biot Savart Law. The more number of particles, the more computations of Biot Savart Law are necessary in each time step proportional to N^2, where N is number of particles. This increases the total time of computation. On the other hand, in each time step of coupled simulation, subloops solve for the grid domain and the grid free domain. It means that a single processor solves for a grid solution and corresponding grid free particle solution in a single time step. This even adds up to the total time of computation. For instance, the coupled simulation in the case of wind turbine (2.7M) took approximately 1.5 months to complete while that with the Devenport case (7.1M) took approximately 4 to 4.5 months to complete. This is currently a serious disadvantage of this code. This drawback must be overcome in the future. This elaborates the second stated restriction of the Pure Lagrangian simulation, that the improvement of the tip vortex resolution by increase of particle number is limited due to high computational costs when the particle number increasing.

In the case of TUM G1 model Turbine, which is a complex wind turbine with turbulent inflow conditions. It would result in solving of very fine domain ($\approx$ 25M cells) in each time step on a single processor, which is almost practically very difficult. Also due to very dense mesh, the number of particles will be high at even first time step and would increase to millions as the vorton cloud travels further downstream. It is practically not possible to solve such a complex problem on a single processor as of now due to limitations of computational power.

Parallelization of the vortex code could be a solution to solve such kind of complex problems in future. Implementing parallelization is itself a complicated task and requires huge investigation. This is not covered in the scope of this present work. It is a part of the future work. Efficient parallelization of the code could make it applicable to solve complex flows like the TUM G1 Model turbine, within sufficiently limited time of computation. Complex problems like flow over a atmospheric boundary layer wind turbine, effect of wind turbine wake on the successive wind turbine in a wind farm etc could then be investigated in the future.

Bibliography

[1] Cottet, G. H., and Koumoutsakos, P. D., 2000, Vortex methods: theory and practice. Cambridge university press.

[2] Devenport, W. J., Rife, M. C., Liapis, S. I., and Follin, G. J., 1996, The Structure and Development of a Wing-Tip Vortex, Journal of Fluid Mechanics, Vol. 312, pp. 67106.

[3] Holzapfel F., 2005, Aircraft Wake Vortex Evolution and Prediction Habilitation, Faculty of Mechanical Engineering of the Technical University Munich.

[4] Misaka, T., Holzapfel, F., Gerz, T., 2015, Large-Eddy Simulation of Aircraft Wake Evolution from Roll-Up Until Vortex Decay, AIAA Paper 53-9.

[5] Kornev, N., Matveev, K., 2003. Complex numerical modeling of dynamics and crashes of wing-in-ground vehicles. AIAA Journal 600.

[6] Rozhdestvensky, K. V., 2006. Wing-in-ground effect vehicles. Progress in Aerospace Sciences 42 (3), 211-283.

[7] D.-F. Feder, M. Dhone, N. Kornev, M. Abdel-Maksoud, 2017, Comparison of Different Approaches Tracking a Wing-tip Vortex, Journal of Ocean Engineering., (doi: https://doi.org/10.1016/j.oceaneng.2017.09.036).

[8] Thomas, A. S., 1985, Aircraft Drag Reduction Technology – A Summary, Advisory Group for Aerospace Research and Development (AGARD),Report 723, Belgium.

[9] Whitcomb, R., 1976, A Design Approach and Selected Wind-Tunnel Results at High Subsonic Speeds for Wing Tip mounted Winglets, NASA TND-8260.

[10] Whitcomb, R., 1977, Methods for reducing subsonic drag due to lift," Special course on concepts for drag reduction, AGARD, Vol.2 pp.1-11, France.

[11] Samal, S. K., Shah, D. A., M, B., Sahoo, A., 2013. Computational prediction of tip-vortex of a swept wing. International Journal of Innovative Research in Science, Engineering and Technology 2.

[12] Nash'at N.A., Proctor F.H., Perry R.B., 2013, Numerical Simulation of the aircraft wake vortex flowfield, 5th AIAA Atmospheric and Space Environments Conference, American Institute of Aeronautics and Astronautics.

[13] Kornev, N., Abbas, N., 2016. Numerical simulation of the tip vortex behind a wing oscillated with a small amplitude. Journal of Aircraft (doi: http://dx.doi.org/10.2514/1.C033945).

[14] Birch, D., Lee, T., 2005. Tip vortex behind a wing oscillated with small amplitude. Journal of Aircraft 42 (5), 1200-1208.

[15] Chow, J. S., Zilliac, G., Bradshaw, P., 1994. Turbulence measurements in the near-field of a wingtip vortex. ASME-PUBLICATIONS-FED 203, 61-61.

[16] Misaka, T., Holzapfel, F., Gerz, T., 2013. Wake evolution of high-lift configuration from roll-up to vortex decay. AIAA Paper 362 (2013), 11.

[17] Nash'at N. Ahmad, Fred H. Proctor, R. Brad Perry, Numerical Simulation of the Aircraft Wake Vortex Flowfield, NASA Langley Research Center, Hampton, Virginia, 23681.

[18] Dhone M., Kornev N. and Abbas N., 2016. Resolution of Tip Vortices by Grid Methods and Coupling Vortex Method with Grid Based Simulation. ICVFM 2016 Rostock, Germany, pp. 53-55.

[19] Dhone M. and Kornev N. ,2018. Tracking of tip vortices arising from Marine structures. Proc. 21st Numerical Towing Tank Symposium, NuTTS 18, NUMECA Inc., Cortona, Italy.

[20] Anderson, J., 1995, Computational Fluid Dynamics. McGraw-Hill Education, ISBN 9781259025969.

[21] Bin Xia, Da-Wen Sun, 2002, CApplications of computational fluid dynamics (CFD) in the food industry: a review. Computers and Electronics in Agriculture, Vol 34, pp. 5-24.

[22] Kevin W. Linfield, Robert G. Mudry, 2008, Pro and Cons of CFD and Physical Flow Modeling, Airflow Sciences Corporation.

[23] Krutartha Sudhir Jathar, Vivek V.Kulkarni, 2015, An Elementary Study of Computational Fluid Dynamics For Various Engineering Applications – A Review, International Research Journal of Engineering and Technology (IRJET), Vol. 02 Issue 08, p-ISSN: 2395-0072.

[24] Dmitri Kuzmin, Introduction to Computational Fluid Dynamics - Lecture Notes, Institute of Applied Mathematics, University of Dortmund.

[25] Acheson, D. J., 1990, Elementary Fluid Dynamics, Oxford University Press, p. 205, ISBN 0-19-859679-0.

[26] Kornev N., 2013, Mathematical Modeling of Turbulent Flows, Rostock.

[27] Simon Schneiderbauer, Michael Krieger, 2014, What do the Navier–Stokes equations mean?, European Journal of Physics 35 015020. Simon Schneiderbauer1 and Michael Krieger2

[28] White, Frank, 1991, Viscous Fluid Flow, 3rd Edition, McGraw-Hill Mechanical Engineering, ISBN-10: 0072402318.

[29] Stokes, George, 1851, On the Effect of the Internal Friction of Fluids on the Motion of Pendulums, Transactions of the Cambridge Philosophical Society, 9: 8–106.

[30] C. B. Dolicanin, V. B. Nikolic, D. C. Dolicanin, 2010, Application of Finite Difference Method to Study of the Phenomenon in the Theory of Thin Plates, Journal of Applied Mathematics and Mechanics, Vol. 2, 29-43

[31] Vidar Thomee, 2001, From finite differences to finite elements A short history of numerical analysis of partial differential equations, Journal of Computational and Applied Mathematics, Vol. 128, 1–54

[32] P. K. Pandey, 2017, A finite difference method for the numerical solving general third order boundary-value problem with an internal boundary condition, Russian Mathematics, Vol. 61, No. 12, pp. 29–38, ISSN 1066-369X

[33] Kornev N., 2013, Computational Methods of Heat and Mass Transfer, Rostock.

[34] Patankar, S. V. (1980). Numerical Heat Transfer and Fluid Flow. Taylor and Francis. ISBN 978-0-89116-522-4.

[35] Thorsten W. Becker, Boris J. P. Kaus, 2016, Numerical Modeling of Earth Systems - An introduction to computational methods with focus on solid Earth applications of continuum mechanics.

[36] D. Feszty, T. Jakubik, Lecture on Computational Fluid Dynamics,Szechenyi University, Audi Department of Vehicle Engineering.

[37] Francois Fraysse, 2012, Numerical Error Prediction and its application in CFD using tau-estimation, University of Madrid.

[38] Steven Chapra, Applied Numerical Methods with MATLAB for Engineers and Scientists, McGraw Hill publication, ISBN 978-0-07-340110-2.

[39] 2015, An ANSYS 16.2.3 Documentation

[40] John Slater, 2008, Tutorial on CFD Verification and Validation, NPARC Alliance Verification and Validation NASA.

[41] Macura Wiktor K., Numerical Stability. From MathWorld - A Wolfram Web Resource.

[42] Courant, R., Friedrichs, K. and Lewy, H. Math. Ann., 1928, Mathematische Annalen, Volume 100, Issue 1, pp 32–74, ISSN 1432-1807.

[43] Hirsch, C., 1990, Numerical Computation of Internal and External Flows. John Wiley and Sons. ISBN 978-0-471-92452-4.

[44] Maria Lukacova, 2003, Numerical Modelling in Computational Fluid Dynamics - Lecture Notes, Technical University Hamburg-Harburg.

[45] Versteeg and Malalasekera, 1995, An introduction to Computational Fluid Dynamics: The Finite Volume Method, Longman Scientific and Technical.

[46] M. R. Spiegel, S. Lipschutz, D. Spellman, 2009, Vector Analysis, Schaum's Outlines (2nd ed.). USA: McGraw Hill. ISBN 978-0-07-161545-7.

[47] Patankar, S. V., 1980, Numerical Heat Transfer and Fluid Flow. Taylor and Francis. ISBN 978-0-89116-522-4.

[48] R. I. Issa, 1985, Solution of the Implicitly Discretized Fluid Flow Equations by Operator-Splitting, Journal of Computational Physics, 62, pp 40-65.

[49] Alonzo-Garcia, Alejandro and Gutierrez-Torres, Claudia and Alfredo, Jimenez-Bernal, 2016, Computational Fluid Dynamics in Turbulent Flow Applications, 10.5772/63831.

[50] Launder BE, Spalding DB, 1972, Mathematical models of Turbulence, Academic Press, Massachusetts.

[51] Patankar SV, 1967, Heat and mass transfer in turbulent boundary layers, Ph.D. Thesis, Imperial College, University of London.

[52] Runchal AK, Wolfshtein M, 1969, Numerical integration procedure for the steady state Navier Stokes equations, Journal of Mechanical Engineering Science II 5:445-453.

[53] Runchal AK, 1969, Transport processes in steady two-dimensional separated flows. Ph.D. Thesis, Imperial College of Science and Technology, UK.

[54] Gosman AD, Pun WM, Runchal AK, Spalding DB, Wolfshtein M, 1969, Heat and mass transfer in recirculating flows, Academic Press, London.

[55] F. Moukalled, L. Mangani, M. Darwish, The Finite Volume Method in Computational Fluid Dynamics, Fluid Mechanics and Its Applications 113, ISSN 2215-0056.

[56] OpenFOAM v5, 2017, User Guide: 4.5 Solution and algorithm control.

[57] E.Robertson, V.Choudhury, S.Bhushan, D.K.Walters, 2015, Validation of OpenFOAM numerical methods and turbulence models for incompressible bluff body flows, Computers and Fluids, Vol. 123, Pages 122-145.

[58] J. M. McDonough, 2007, Introductory lectures on Turbulence, Departments of Mechanical Engineering and Mathematics, University of Kentucky.

[59] Lectures on Basics of Turbulent flows, Chapter 7, Massachusetts Institute of Technology.

[60] Spalart, P. R. and Allmaras, S. R., 1994, A One-Equation Turbulence Model for Aerodynamic Flows. Vol. 439. American Institute of Aeronautics and Astronautics.

[61] Shur, M. L., Strelets, M. K., Travin, A. K., Spalart, P. R., 2000. Turbulence modeling in rotating and curved channels: Assessing the spalart-shur correction. Journ. AIAA 38 (5), 784-792.

[62] Spalart, P. R. and Shur, M., 1997, On the Sensitization of Turbulence Models to Rotation and Curvature, Aerospace Science and Technology 1(5), 297-302.

[63] Menter, F., Smirnov, P., 2009. Sensitization of the sst turbulence model to rotation and curvature by applying the spalart shur correction term. Journal of Turbomachinery 131(4).

[64] Menter, F., Ferreira, J. C., Esch, T., Konno, B., Germany, A. C., 2003. The sst turbulence model with improved wall treatment for heat transfer predictions in gas turbines. Proceedings of the international gas turbine congress, 2-7.

[65] Launder, B. E., Reece, G. J. and Rodi, W., 1975, Progress in the Development of a Reynolds-Stress Turbulent Closure., Journal of Fluid Mechanics, Vol. 68(3), pp. 537-566.

[66] Johansson, Arne V and Hallback, Magnus, 1994, Modelling of rapid pressure strain in Reynolds-stress closures. Journal of Fluid Mechanics. 269, 143–168.

[67] Speziale, C.G., Sarkar, S., Gatski, T.B., 1991, Modeling the Pressure-Strain Correlation of Turbulence: an Invariant Dynamical Systems Approach, Journal of Fluid Mechanics, Vol. 227, pp. 245-272.

[68] Smagorinsky, Joseph, 1963, General Circulation Experiments with the Primitive Equations, Monthly Weather Review. 91 (3): 99–164.

[69] Deardorff, James, 1970, A numerical study of three-dimensional turbulent channel flow at large Reynolds numbers, Journal of Fluid Mechanics. 41 (2): 453–480.

[70] Andre Bakker, 2006, Lecture on Applied Computational Dynamics - Large Eddy Simulation, Fluent Inc.

[71] J. Frohlich, W. Rodi, 2002,
Introduction to Large Eddy Simulation of Turbulent Flows, Institute for Hydromechanics, University of Karlsruhe.

[72] Frohlich J. , Rodi W. , Launder B. E. and Sandham N. D., 2002, Closure strategies for turbulent and transitional flows-Introduction to Large Eddy Simulation of Turbulent Flows. Cambridge University Press.

[73] Christopher J. Greenshields, 2018, OpenFOAM User Guide, Version 6.

[74] Pope S. B., 2000, Turbulent flows. Cambridge university Press.

[75] Germano, M., Piomelli, U., Moin, P. Cabot, W. H., 1991, A dynamic subgrid-scale eddy viscosity model. Phys. Fluids A. 3, 1760-1765.

[76] Germano M., 1992, Turbulence: the filtering approach. Journal of Fluid Mechanics, 238:325–336.

[77] Smagorinsky, J., 1963, General Circulation Experiments with the Primitive Equations, Month. Weath. Rev. Vol. 93, pp. 99-165.

[78] Wells, J., 2009. Effects of turbulence modeling on RANS simulations of tip vortices. Master's thesis. Virginia Polytechnic Institute and State University.

[79] Wells, J., Salem-Said, A., Ragab, S.A., 2010. Effects of turbulence modeling on RANS simulations of tip vortices. In: 48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition. Orlando, Florida.

[80] Jeong, J. and Hussain, F., 1995, On the Identification of a Vortex, Journal of Fluid Mechanics, Vol. 285, pp. 69-94.

[81] Page, R.D., Clawson, K.L., Garodz, L.J., Rudis, R.P., 1991, Report on Tower Fly-by Testing, Proceedings of the FAA Wake Vortices Conference, Washington, D.C.

[82] Martin, P.B., Pugilese, G.J., Leishman, J.G., 2001, High Resolution Trailing Vortex Measurements in the Wake of a Hovering Rotor, Proceedings of the 53rd Annual Forum of the American Helicopter Society.

[83] Moore, D.W. and Saffman, P.G., 1973, Axial Flow in Laminar Trailing Vortices, Proceedings of the Royal Society, London, A., Vol. 333.

[84] Shyy, W., and Liu, H., 1996, Flapping Wings and Aerodynamic Lift: The Role of Leading-Edge Vortices, AIAA Journal, Vol. 45, No. 12, pp. 2817-2819.

[85] Green, S. I., 1995, Fluid Vortices: Fluid Mechanics and Its Applications, Vol. 1, Kluwer Academic Publishers, Netherlands, Chap. 10.

[86] Sebastian Gomez, Lindsay N. Gilkey, Bryan E. Kaiser, Svetlana V. Poroseva, 2014, Computational Analysis of a Tip Vortex Structure Shed from a Bio-Inspired Blade, AIAA paper.

[87] Gavrilovic Nikola N, Bosko P. Rasuo, George S. Dulikravich, Vladimir B. Parezanovic, 2015, Commercial Aircraft Performance Improvement Using Winglet, FME Transaction Vol. 43.

[88] Setyo Hariyadi S. P., Sutardi, and Wawan Aries Widodo, 2016, Numerical study of aerodynamic analysis on wing airfoil NACA 43018 with the addition of forward and rearward wingtip fence, AIP Conference Proceedings 1778, 030011.

[89] Zeman, O., 1995, The Persistence of Trailing Vortices: A Modeling Study, Center for Turbulence Research, In: Physics of Fluids, American Institute of Physics.

[90] Ragab, S. and Sreedhar, M., 1995, Numerical Simulations of Vortices with Axial Velocity Deficits, Physics of Fluids, Vol. 7.

[91] Qin, J.H., 1998, Numerical Simulations of a Turbulent Axial Vortex, Ph.D. Thesis, Department of Aerospace Engineering, Purdue University.

[92] Chow, J.S., Zilliac, G.G., and Bradshaw, P., 1997, Mean and Turbulence Measurements in the Near Field of a Wingtip Vortex, AIAA Journal, Vol. 35, No. 10.

[93] J. Ferziger, 1999, Computational Methods for Fluid Dynamics, Springer Verlag, ISBN 3-540-65373-2.

[94] Ting, L., 1991, Viscous Vortical Flows. Lecture notes in physics. Springer-Verlag. ISBN 978-3-540-53713-7.

[95] Kida, Shigeo, 2001, Life, Structure and Dynamical Role of Vortical Motion in Turbulence, IUTAMim Symposium on Tubes, Sheets and Singularities in Fluid Dynamics, Zakopane, Poland.

[96] Kundu P and Cohen I., 2015, Fluid Mechanics, Academic Press. ISBN 9780124059351.

[97] Pinaki Chakraboty, S. Balachandar, Ronal J. Adrian, 2005, On the relationships between local vortex identification schemes, Journal of Fluid Mechanics, Vol. 535, pp.189-214.

[98] P. Chakraboty, S. Balachandar, Ronal J. Adrian, 2007, Kinematics of Local Vortex Identification Criteria, Journal of Visualisation, Vol. 10, No. 2, pp. 137-140.

[99] Jeong J., Hussain F., 1995, On the identification of a vortex, Journal of Fluid Mechanics, Vol. 285, pp. 69-94.

[100] Zhou J., Adrian J., Balachandar S., Kendall M., 1999, Mechanisms for generating coherent packets of hairpin vortices, Journal of Fluid Mechanics, Vol. 387, pp. 353-396.

[101] Chong S., Perry E., Cantwell J., 1990, A general classification of three dimensional flow fields, Physics of Fluids, 2, 765-777.

[102] Hunt R., Wray A., Moin P., 1988, Eddies, stream and convergence zones in turbulent flows, Center for Turbulence Research Report CTR-S88, pp. 193-208.

[103] Y. Levy, D. Degani, and A. Seginer. Graphical Visualization of Vortical Flows by Means of Helicity. AIAA J., 28(8):1347-1352, August 1990.

[104] C. H. Berdahl and D. S. Thompson, 1993, Eduction of Swirling Structure Using the Velocity Gradient Tensor. AIAA J., 31(1):97-103.

[105] R. C. Strawn, D. N. Kenwright, and J. Ahmad, 1999, Computer Visualization of Vortex Wake Systems. AIAA J., 37(4):511-512.

[106] D. Sujudi and R. Haimes, 1995, Identification of Swirling Flow in 3D Vector Fields. In AIAA 12th Computational Fluid Dynamics Conference, Paper 95-1715.

[107] D. C. Banks and B. A. Singer, 1995, A Predictor-Corrector Technique for Visualizing Unsteady Flow. IEEE Trans. on Visualization and Computer Graphics, 1(2):151-163.

[108] I. A. Sadarjoen, F. H. Post, B. Ma, D. C. Banks, and H.-G. Pagendarm, 1998, Selective Visualization of Vortices in Hydrodynamic Flows. In IEEE Visualization '98, pages 419-422.

[109] J. Ming, M. Raghu and T. David, 2005, Detection and Visualization of Vortices, The Visual handbook.

[110] Vivianne Holmen, 2012, Methods for Vortex Identification, Lund University

[111] Batchelor G.K., 1973, An introduction to fluid dynamics, Cambridge University Press, ISBN 978-0-521-09817-5.

[112] Cottet, G.H., Koumoutsakos P.D., 2000, Vortex Methods: Theory and Practice, Cambridge University Press.

[113] Lamb H., 1994, Hydrodynamics 6th edition, Cambridge University Press, ISBN 978-0-521-45868-9.

[114] Andre Bakker, 2006, Lecture on Applied Computational Dynamics - Conservation Equations, Fluent Inc.

[115] M.S. Shadlooa, G. Oger, D. Le Touze, 2016, Smoothed particle hydrodynamics method for fluid flows, towards industrial applications-Motivations, current state, and challenges, Journal of Computer and Fluids, Vol. 136, pp. 11-34.

[116] Dick Yue, 2005, 2.20 Marine Hydrodynamics (13.021), Massachusetts Institute of Technology: MIT OpenCourseWare.

[117] Christopher Brennen, 2004, Basic Fluid Dynamics Book, California Institute of Technology.

[118] Chorin, A., 1982, The evolution of a turbulent vortex. Comm. in Mathematical Physics, Vol. 83, 517-535.

[119] Chorin, A., 1990, Constrained random walks and vortex filaments in turbulence theory. Comm. in Mathematical Physics, Vol. 132, 519-536.

[120] Mark J. Stock, 2007, Summary of Vortex Methods Literature.

[121] A. J. Chorin, 1981, Estimates of intermittency, spectra, and blow-up in developed turbulence, Communications on Pure and Applied Mathematics 34, 853–866.

[122] T. Schlick, 2002, Molecular Modeling and Simulation: An interdisciplinary guide, Springer, New York.

[123] Leonard A., 1985, Computing three-dimensional incompressible flows with vortex elements, Annual Revision Fluid Mechanics, 17, 523-559.

[124] Knio M., Ghoniem F., 1990, Numerical study of a three-dimensional vortex method, Journal of Computational Physics, 86, 75-106.

[125] Winckelmans G., Leonard A., 1993, Contributions to vortex particle methods for computation of three dimensional incompressible unsteady flows, Journal of Computational Physics, 109, 247-273.

[126] A. Lozano, A. Garcia-Olivares, C. Dopazo, 1998, The instability growth leading to a liquid sheet breakup, Physics of Fluids 10 (9), 2188–2197.

[127] Henryk Kudela, Pawel Regucki, 2002, The vortex-in-cell method for the study of three-dimensional vortex structures, Tubes, Sheets and Singularities in Fluid Dynamics, 49-54.

[128] J. P. Christiansen, 1973, Numerical simulation of hydrodynamics by the method of point vortices, J. Comput. Phys. 13, 363–379.

[129] J. P. Christiansen, N. J. Zabusky, 1973, Instability, coalescence and fission of finite-area vortex structures, J. Fluid Mech. 61, 219–243.

[130] J. P. Christiansen, 1970, Vortex — a two-dimensional hydrodynamics simulation code, UKAEA Culham Lab. Rep. CLM-R106, UKAEA Culham.

[131] B. Couet, O. Buneman, A. Leonard, Simulation of three-dimensional incompressible flows with a vortex-in-cell method, J. Comput. Phys. 39 (2) (1981) 305–328.

[132] G.H. Cottet, B. Michaux, S. Ossia, G. VanderLinden, 2002, A comparison of spectral and vortex methods in three-dimensional incompressible flows, J. Comput. Phys. 175, 702–712.

[133] G.H. Cottet, 2000, 3D vortex methods: achievements and challenges, Selected papers of the First International Conference on Vortex Methods, Kobe Japan, pp. 123-134.

[134] R. Yokota, T.K. Sheel, S. Obi, 2007, Calculation of isotropic turbulence using a pure Lagrangian vortex method, Journal of Computational Physics 226, 1589–1606.

[135] Rio Yokota, Shinnosuke Obi, 2007, Pure Lagrangian vortex methods for the simulation od decaying isotropic turbulence, Proceedings of Turbulence and Shear Flow Phenomena, pp. 365-370.

[136] I. Zawadski, H. Aref, 1991, Mixing during vortex ring collision, Phys. Fluids A 3(5), 1405–1410.

[137] Gretar Tryggvason, 2013, Vortex Methods - Computational Fluid Dynamics Course, University of Notre Dame.

[138] Kuji, k., and Fukuda, K., 2016, Numerical Investigation on Unsteady Motion and Deformation of Vortical Structures on an Accelerated / Decelerated Airfoil using a Grid-Free Vortex Method. Proc. Int. Conf. Vortex Flows and Vortex Models, Rostock, Germany, pp. 63-64.

[139] Aparinov., A. A., and Setukha. A. V., 2016, Solution of Aerodynamic Problems of Moving Objects with Vortex Method. Proc. Int. Conf. Vortex Flows and Vortex Models, Rostock, Germany, pp. 19-21.

[140] Dergachev, S. A., and Shcheglov, G.A., 2016, The Vortex Loops Based Algorithm for 3D Flow Simulation Around Bluff Bodies. Int. Conf. Vortex Flows and Vortex Models, Rostock, Germany, pp. 36-37.

[141] Dynnikov, Ya. A., Dynnikova, G. Ya., and Guvernyuk, S. V. Modelling of Pulsating Jets. Int. Conf. Vortex Flows and Vortex Models 2016, Rostock, Germany, pp. 38-39.

[142] Mimeau, C., Mortazavi, I., and Cottet, G-H., 2016, A vortex penalization method for low and moderate Reynolds number flows. Proc. Int. Conf. Vortex Flows and Vortex Models, Rostock, Germany, pp. 75.

[143] C. Mimeau , F. Gallizio, I. Mortazavi and G.-H. Cottet, 2015, Vortex penalization method for bluff body flows. Intern. Journal for Numerical Methods in Fluids, 79, pp. 53-83.

[144] C. Mimeau , I. Mortazavi and G.-H. Cottet, 2014, Passive Flow Control Around a Semi-Circular Cylinder Using Porous Coatings. Intern. Journal of Flow Control, 6(1), pp. 43-60.

[145] Cottet, G.-H., and Poncet, P., 2003, Advances in direct numerical simulations of 3D wall-bounded flows by Vortex-in-Cell methods. Journal of Computational Physics, Vol. 193, Issue 1, 136-158.

[146] El Ossmani, M., and Poncet, P., 2010, Efficiency of multiscale hybrid grid-particle vortex methods, Multiscale modeling and simulations, Vol. 8, No. 5, pp. 1671-1690.

[147] Ould-Salihi, M.L., Cottet, G.-H., and El Hamraoui, M., 2000, Blending Finite difference and vortex methods, SIAM Journal of scientific computing, Vol. 22, No. 5, pp. 1655-1674.

[148] Winckelmans G., Chatelain, P., Marichal, Y., Parmentier, P., Duponcheel, M., Caprace, D-G., and Gillis, T. Recent Developments in Vortex Particle-Mesh (VPM) Methods and Applications. Proc. Int. Conf. Vortex Flows and Vortex Models 2016, Rostock, Germany, pp. 16.

[149] Winckelmans, G. and Leonard, A., 1993, Contributions to vortex particle methods for the computation of three dimensional incompressible unsteady flows, *J. Comput. Phys.*, 109, 247–273.

[150] Feder, D.-F., Abdel-Maksoud, M., 2016, Tracking a tip vortex with adaptive vorticity confinement and hybrid RANS-LES. Open J. Fluid Mech. 6 (4).

[151] Jasak, H., Tukovic, Z., 2010, Dynamic mesh handling in openfoam applied to fluidstructure interaction simulations. In: Proceedings of the V European Conference Computational Fluid Dynamics, pp. 14–17. Lisbon, Portugal.

[152] Kornev, N., 2016, Autowing Code Based on the Unsteady Nonlinear Vortex Lattice Method. http://www.lemos.uni-rostock.de/downloads/cfd/.

[153] H. Jasak, 1996, Error Analysis and Estimation for the Finite Volume Method with Application to Fluid Flows - PhD Thesis.

[154] D. Mustafa, M. Shephard, 2000, Adaptive CFD analysis for rotorcraft aerodynamics, Journal of Computational Methods in Applied Mechanics.

[155] Hee Jung Kang, 2002, Unstructured Mesh Navier Stokes Calculations of the Flow Field of a Helicopter Rotor in Hover, Journal of the American Helicopter Society.

[156] Yann Le Moigne, 2004, AMR and simulations of delta wings, AIAA, ECOMAS.

[157] C. Pashias, 2005, Propeller tip vortex simulation using adaptive grid refinement based on flow feature identification - PhD Thesis, University of Southampton.

[158] M. James, L. David, 2008, Am Output-based Adaptive and Higher-Order Method for a Rotor in Hover, 26th AIAA Applied Aerodynamics Conference.

[159] T. Zurheide, G. Huppertz, E. Fares, M. Meinke, W. Schroeder, 2010, Interaction of Wing-Tip vortices and Jets in the Extended Wake, Summary of Flow Modulation and Fluid-Structure Interaction Findings. Notes on Numerical Fluid Mechanics and Multidisciplinary Design, vol 109. Springer, ISBN 978-3-642-04087-0.

[160] H. Nathan, 2011, Tip Vortex Field Resolution using an Adaptive Dual-Mesh Computational Paradigm, AIAA.

[161] A. B. Phillips, S. R. Turnock, 2011, Application of the VORTFIND algorithm for the identification of the vortical flow features around complex 3D geometries, International Journal For Numerical Methods in Fluids.

[162] M. J. Harris, 2013, Flow Feature Aligned Mesh Generation and Adaptation - PhD Thesis, University of Sheffield.

[163] L. David, U. Jasim, L. Terry, 2015, Visualisation and Quantification of Rotor Tip Vortices in Helicopter flows, AIAA.

[164] J. Windt and J. Bosschers, 2015, Influence of local and Adaptive Mesh Refinement on the tip vortex characteristics of a wing and propeller, VI International Conference on Computational Methods in Marine Engineering MARINE.

[165] B. Wang, Zhihui Liu, 2015, Simulations of tip vortex cavitation flows with nonlinear k-eps model, Journal of Physics.

[166] M. Neal, 2017, Navier Stokes simulation of UH-60A rotor wake interaction using Adaptive Mesh Refinement, 73rd Annual Forum and Technology Display, USA.

[167] G. Jinlan, 2017, A high order element based adaptive mesh refinement strategy for three dimensional unstructured grid, International Journal of Numerical Methods Fluids.

[168] G. Jinlan, Xin Yuan, Xinrong Su, 2018, Adaptive mesh refinement method based investigation of the interaction between shock wave, boundary layer and tip vortex in a transonic compressor, Journal of Aerospace.

[169] A. Wissink, S. Kamkar, T. Pulliam, J. Sitaraman, and V. Sankaran, 2010, Cartesian Adaptive Mesh Refinement for Rotorcraft Wake Resolution, 28th AIAA Applied Aerodynamics Conference 4554.

[170] S. Kamkar, 2011, Feature driven Cartesian adaptive mesh refinement for vortex-dominated flows, Journal of Computational Physics.

[171] N. Kasmai, D. Thompson, E. Luke, M. Jankun-Kelly, R. Machiraju, 2011, Feature-based adaptive mesh refinement for wingtip vortices, International Journal for Numerical methods in fluids.

[172] S. Peron, C. Benoit, 2013, Automatic off-body overset adaptive Cartesian mesh method based on an octree approach, Journal of Computational Physics, Vol. 232.

[173] Farrell P. E. and Maddison J. R., 2011, Conservative interpolation between volume meshes by local Galerkin projection. Computer Methods in Applied Mechanics and Engineering, 200(1):89–100.

[174] Kooijman, H., Lindenburg, C., Winkelaar, D., and Van der Hooft, E., 2003. DOWEC 6 MWPRE-DESIGN - Aeroelastic modelling of the DOWEC 6 MW pre-design in PHATAS. Tech. Rep. DOWEC-F1W2-HJK-01-046/9, DOWEC Project Consortium.

[175] Lindenburg, C., 2002. Aeroelastic Modelling of the LMH64-5 Blade. Tech. Rep. DOWEC-02-KL-083/0, DOWEC Project Consortium.

[176] Bredmose, H., Larsen, S. E., Matha, D., Rettenmeier, A., Marino, E., and Saettran, L., 2012. D2.4: Collation of offshore windwave dynamics. Tech. rep., MARINET.

[177] Paul Schunemann, Timo Zwisele, Frank Adam, Uwe Ritschel, 2018, Development of a scaled rotor blade for tank tests of floating wind turbine systems, Proceedings of the ASME 2018 37th International Conference on Ocean, Offshore and Arctic Engineering OMAE2018.

[178] Jiangang Wang, Chengyu Wang, Filippo Campagnolo, and Carlo L. Bottasso, 2018, A Large-Eddy Simulation Approach for Wind Turbine Wakes and its Verification with Wind Tunnel Measurements, Manuscript under review for journal Wind Energ. Sci.

[179] Jiangang Wang, Chengyu Wang, Filippo Campagnolo, and Carlo L. Bottasso, 2018, Scale adaptive of Wind Turbines and its Verification with respect to Wind Tunnel Measurements, Manuscript under review for journal Wind Energ. Sci.

[180] J.G. Schepers, K. Boorsma, T. Cho,S. Gomez-Iradi, P. Schaffarczyk, W.Z. Shen, T. Lutz, B. Stoevesandt, S. Schreck, D. Micallef, R. Pereira, T. Sant, H.A. Madsen, N. Sørensen, 2012, Final report of IEA Task 29, Mexnext (Phase 1): Analysis of Mexico wind tunnel measurements, ECN-E–12-004.