Development of a Direct Numerical Simulation (DNS) Code: Simulating incompressible laminar flows

A DISSERTATION

SUBMITTED FOR THE PARTIAL FULFILMENT OF TRAINING IN EARTH SYSTEM SCIENCES AND CLIMATE (ESSC)

by:
Manmeet Singh

Under the guidance of:
Dr. S. A. Dixit

Centre For Advanced Training
Indian Institute of Tropical Meteorology

December 2014

CERTIFICATE

This is to certify that the work entitled **Development of a Direct Numerical Simulation (DNS) code: simulating incompressible laminar flows** being submitted by Manmeet Singh has been carried out under my supervision, in partial fulfilment of the requirements for the Induction Training at CAT-ESSC, IITM. The matter embodied in this thesis has not been submitted for award of any degree in any University or Institute.

Dr S. A. Dixit

PDTC, IITM Pune

Acknowledgements

I am grateful to my professor and guide Dr Shivsai Ajit Dixit for all the support he has given me in the past one year right from the in house making of setup for FDL laboratory to the starting of the project and to this date.Dr Bipin Kumar was very helpful for my mathematical and computational needs during the project and Ketan Sir has also assisted in numerous ways. Also I would like to put forth my sincere thanks to Dr. R.H. Kriplani and Prof B.N Goswami for all their efforts to make this training progarm a success. Aditya Konduri (Texas A & M University), who helped me walk into this world of DNS has been extremely supportive and I thank him from the bottom of my heart for answering my mails so patiently even after being so busy. Mr Saurabh (IISc) whose comments were and would be very useful in the future code development was very kind in coming and discussing the code in IITM. Dr Vinu Valsala has been very appreciative and helping in the work and I am grateful to him for his guidance. I would also like to give my heartful thanks to all the dear for their support.

Abstract

Direct Numerical Simulation is a method of solving the equations of motion (Navier-Stokes (NS) Equations), without any turbulence model, by directly discretizing the equations alongwith the initial and boundary conditions. In this study a code is developed to solve the NS equations using fractional step method over a 2D rectangular cartesian domain. The present mesh size is 32 x 32 and is solved by finite volume method via finite difference method. The Pressure Poisson Equation is solved by the Successive Over Relaxation method with β =1.2. The present code is for dimensional NS equations for now but will be non dimensionalized in the future. For validating the code the standard test case of lid-driven cavity flow (Re=1000) is chosen. The trends in the results of the present code show reasonable match with the benchmark data to within the present limited spatial resolution. For the initial studies, MATLAB is used whereas the code is written in FORTRAN 90 for further computations. The output from FORTRAN 90 program is written on netcdf files and visualized on MATLAB and Ferret. After validation, the code is tested for the case of laminar jet flow entering the box domain from the bottom boundary and exiting smoothly through the top boundary. Side walls carry no-slip boundary conditions, bottom wall carries no-slip boundary condition in addition to jet velocity profile and the boundary top wall carries zero gradient (in the direction of flow) condition. For the initial conditions,

it is assumed that the fluid is at rest at the start of simulation. The vorticity plots from the simulated flow appear to be qualitatively satisfactory. Trend towards self-similar development is observed only farther in the domain since the domain under present consideration is relatively short (only 6 times the jet diameter).

Contents

1 Introduction **1**

2 Method of Development **8**

2.1 What is DNS? 8

2.2 Navier Stokes Equations 11

2.3 Non Dimensionalization 13

2.4 Discretizing Space Differentials 13

2.5 Discretizing Time Differentials 19

2.6 Algorithm 20

2.6.1 Solving Pressure Poisson Equation 23

2.6.2 Stability Criteria 26

2.7 Boundary Conditions 27

2.8 Visualization Techniques 28

3 Validation **29**

3.1 Validating against Lid Driven Cavity Flow 30

4 Results and Discussion **32**

5 Conclusions and Future Scope **52**

Chapter 1

Introduction

According to the Intergovernmental Panel for Climate Change fifth assessment report (Stocker *et al.*, 2013), clouds and aerosols are providing the largest error in the analysis results of the present day models. It is a well established fact that aerosols are required for the formation of clouds. Any forcing that has links with aerosols is indirectly linked to clouds as well. (Charlson *et al.*, 1992) have shown the impact of anthropogenic aerosols on climate forcing, which is consequently linked to clouds as well. So an in depth study of cloud dynamics is very important.

It has been said until not very long ago that detailed theories of complex natural phenomena like cloud flows have been not developed very well (Turner, 1973, p167). Also it has been stressed (Batchelor, 1954) that the study of free convection that takes place primarily because of buoyancy has not been studied very well. Ogura (1963) has condensed the work done on

cumulus convection into three categories. The first type includes linear perturbation theories by incorporating different static stabilities for upward and downward motions, second comes the theoretical models which include the plume or jet models and the other is the parcel model or bubble theory. The plume model was seen to closely simulate cumulus convection in the Thunderstorm Project. The third type of studies include direct numerical integration of convective motions.

Similarity theory of jets and plumes has been an attractive approach towards developing models for cloud flows (Turner, 1973). In similarity theory, the flow variables scale with appropriate scaling parameters and this enables conversion of governing partial differential equations in to non-linear ordinary differential equations that are fairly easy to solve. Profiles of dimensionless variables become invariant in the streamwise direction as the streamwise coordinate is incorporated in the similarity variable. Hence flows of any size can be studied using the generalized analysis. This property forms the basis of using non dimensionalized jets/plumes for the study of cloud flows. It was seemed appropriate to model convection as a jet/plume as it is also an updraft caused due to the short wave interaction with the land and ocean,

The formation of clouds is caused by the ascent of air which gets moist due to the lapse rate and condensation takes place at the Lifting Condensation Level. The ascent may be caused by buoyancy or orographic lifting. Latent heat release causes the cloud to rise and attain different shapes. This off source buoyancy is the subject of interest for understanding the processes

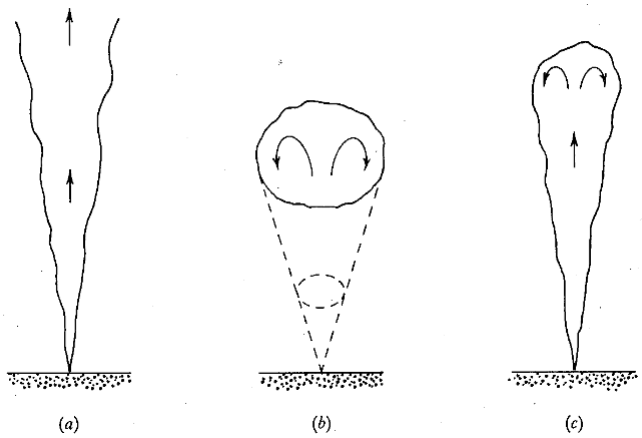

Figure 1.1: Sketches of the various convection phenomena : (a)plume, (b) thermal, (c) starting plume.(Turner (1973))

undergoing in the air.

The experiments on jets, plumes or thermals to understand various aspects of cloud flows were first started at IISc by Prof Narsimha which were followed up by Venkatkrishnan and Agrawal's groups. The schematic of the experiments is shown in figure 1.2. The experiments involve adding thermal energy to the jet/plume in a particular space in the tank of the setup. Heat is provided by the electrical conducting wires and the shear flow (jet/plume) is made conducting by adding HCl in small quantities to the deionized fluid. Usually 5-6 platinum wires of 90 micrometre diameter and 10 millimetres apart are put on a frame. It is important to prevent the release of gas bubbles and electrolysis of water, hence the voltage applied has high frequency (greater than 20 kHz)

Another important reason to study the clouds and associated processes

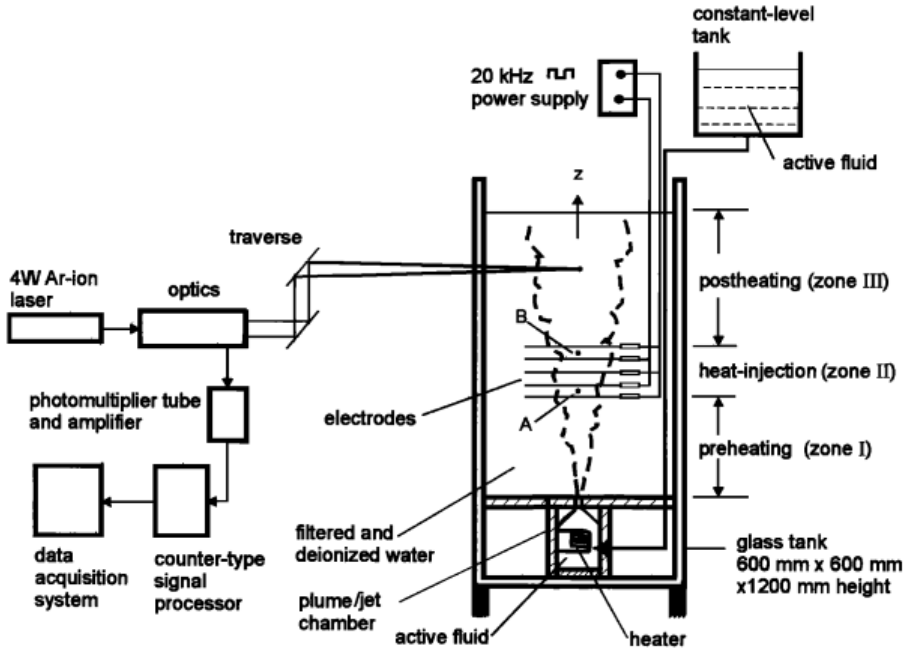

Figure 1.2: Experimental Setup (Venkatakrishnan etal, 1998)

is understanding the entrainment that is a key player in the formation and growth of clouds. Narsimha and Bhat(2008) have explained the dilemma related to the understanding of entrainment in clouds. They have explained that going by Paluch's aerial experiments on cloud thermodynamical properties, entrainment should take place from the cloud top and lateral entrainment should be minimal. Rather, going by the ideas of Turner(1973) on which the ideas of cloud physics models are based, entrainment should take place from the lateral sides. So a numerical Direct Numerical Simulation kind of model can shed some light on the entrainment ideas.

Of particular interest is a transient turbulent plume or a jet that is subjected to volumetric heating; this is considered as an appropriate physical

model for cumulus cloud flows (Diwan et. al. 2011); see Warner (1955) and Stommel (1947) for different proposals for the entrainment mechanisms in real clouds. Bodenschatz et al. (2010) suggest that clouds cannot understood in a meaningful sense unless the turbulence within is understood.

Basu and Narsimha (1999) have demonstrated the use of DNS in studying turbulence in model cloud flows. It has been found that the off-source heat addition in turbulent buoyant plumes leads to a drop in the lateral entrainment into the cloud flow; this is a laboratory analog of the heat release in real cloud flows (Diwan et. al. 2011). The future development on this DNS code can shed useful light on the experimental observation. The importance of improved understanding of basic cloud processes in climate models has been well recognized in the literature (Stephens et al, 1990). DNS is computationally exhaustive since no closure models or parameterizations are used. Previous studies (Prasanth, 2013) have noted the need of adequate resolution in DNS for accurate results.

Although initially planned to solve the three equations(momentum equation, continuity and the thermal energy equation) in 3D by the finite volume method using Chorin's(1968) method of fractional step, but due to limitations in the project duration, a 2D code simulating a jet for only momentum and continuity equations (dimensional,cgs units) has been written in MatLab as well as Fortran 90 . The code developed was then validated for standard test case of lid-driven cavity flow and was visualized in Ferret for a 32 x 32 coarse grid for a Reynolds number of 1000. For the Matlab code, the

simulations were carried on Thinkcentre i5 core processor at IITM as the simulations were computationally time intensive and for Fortran 90 standard intel centrino laptop was found to give decent speed.

As far as results are considered, the jet shows two lobes of vortical structures emerging from the lower boundary which then exit from the top boundary for the 2D case. It was observed from the velocity contours how possibly entrainment could be occurring from the sides with the quiver plots showing near orthogonal vectors to the jet area. This task of code development in future would involve converting this dimensional code to non dimensional form, two dimensional to three dimensional, cartesian to cylindrical coordinate system, laminar inlet to a turbulent one, improving the boundary conditions, incorporating the thermodynamic equation, using the exact profile of jet, making the grids finer and finally parallelizing the code. This thesis is structured as follows: the numerical details are discussed in chapter 2, validation studies in chapter 3, results and discussion in chapter 4 and conclusions and future scope in chapter 5.

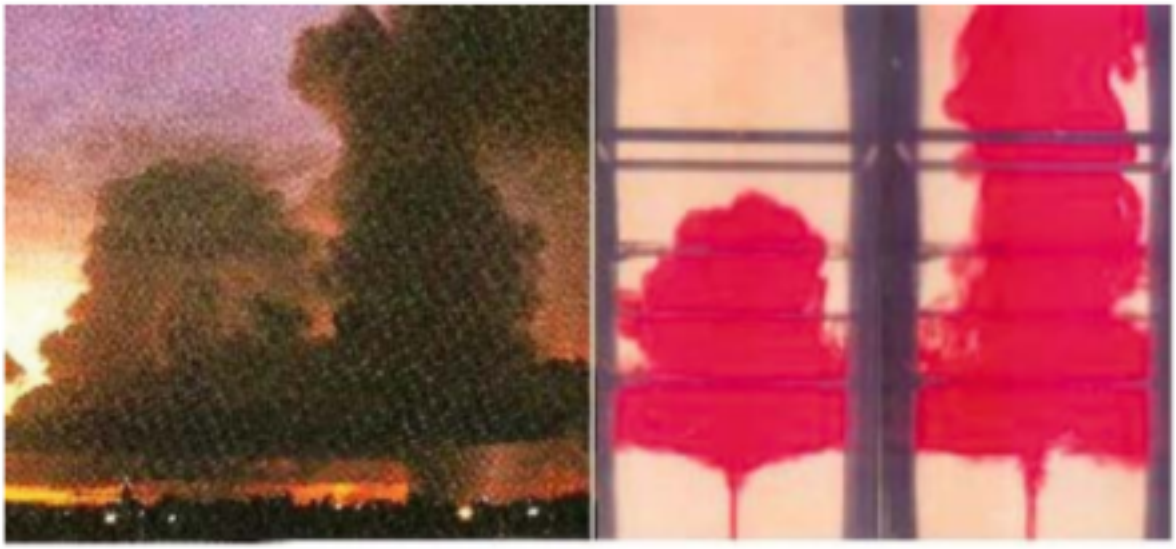

Figure 1.3: (a) Natural clouds. (b) Image from a dye flow visualization of a jet subjected to off-source volumetric heating, neutral with respect to ambient in the lower, denser layer below where the jet spreads out horizontally.(Narsimha and Bhat, 2008)

Chapter 2

Method of Development

2.1 What is DNS?

Direct Numerical Simulation was first performed by Orszag and Patterson(OS) in 1972 on a 32^3 grid at National Centre for Atmospheric Research. It is well established fact that the analytical solutions to the turbulent problems of the Navier Stokes (NS) equations don't exist and hence such equations are solved numerically using what has been known as the Direct Numerical Simulation (DNS), and been an field of research for the past three decades. As DNS involves brute force solutions to the NS equations, they pose a constraints on the solutions of all kinds of flows such as flows at high Reynolds Numbers, because of the computational requirements. The importance of computational resources can be gauged from the fact that the CDC7600 processor which was used by OS in 1972 is five times slower than the 200 MHz

personal computers available these days. Although the computational power has increased, but still performing DNS for complex cases such as global models or high Reynolds Number flows is at high resolution is still an issue. The finest DNS model carried out till date was at University of Texas which had 242 billion grid points. So, rather than resolving every smallest eddy in DNS, Large Eddy Simulation (LES) has also become popular these days in which the eddies having the largest energy scales are resolved whereas smaller scales are parameterized. Another method which has been the favourite of CFD community has been the Reynolds Averaged Navier Stokes (RANS) equations, in which the fields are broken statistically into mean and the turbulent parts based on the statistics, and the terms involving perturbations are then modelled. So RANS as a method to solve Navier Stokes equations falls on the lowest echelon after LES and DNS being the most accurate method of solving the NS equations.

A variety of flows have been understood by using DNS in the past two decades which would, otherwise have been very difficult using laboratory experiments and which may include three dimensional, incompressible, compressible, turbulent, reacting,transient, geophysical and newer domains are getting developed such as biological flows. The smallest length scale that a DNS should resolve is called Kolmogorov Scale η, which is given by

$$\eta = (\nu^3/\epsilon)^{1/4} \tag{2.1}$$

where ν is the kinematic viscosity and ϵ is the rate of kinetic energy (KE) dissipation. But it is not always required to resolve to the Kolmogrov(η) scale but a function of the η. DNS can be done either by Spectral method, finite difference method or by finite volume method. In Spectral method, the field is assumed to be a combination of a number of harmonics and a cutoff frequency depending upon the order of accuracy is chosen to solve the equations. The equations are expanded based on the fourier series expansion and then solved. In finite difference methods (FDM), the partial differential equations are expanded based on the Taylor series expansion. The Finite Volume Method (FVM) is similar to the Finite Difference Method except that the volume integrals in accordance with the divergence theorem are converted to surface integrals and evaluated as flux terms to the surfaces of the volume (Versteeg & Malalasekara, 1995). The finite volume method in conjunction with the finite difference method is used for the analysis in this study. We have converted the volume integrals into the surface integrals and then the final discretizing was done by central finite difference approximation for spatial discretization. For discretizing the time differential, simple Euler's method was used. As the flow is incompressible no shock waves will come and was found to be true in the study.

2.2 Navier Stokes Equations

The set of equations include a set of three equations including Continuity equation, Momentum equation and the Thermodynamic equation which are as shown below:

$$\nabla.\mathbf{u} = 0 \tag{2.2}$$

where $\mathbf{u}$ is the velocity vector.

The continuity equation shown above implies that the flow we are taking is incompressible as $\partial\rho/\partial t = 0$ in the generalized continuity equation gives us the same. This has been observed in the previous numerical analysis (Konduri, 2009), (Basu & Narasimha, 1999),(Prasanth, 2013) and the laboratory experiments of ((Bhat & Narasimha, 1996),(Venkatakrishnan *et al.*, 1998)). Hence the density variation along the horizontal planes is neglected and that in the vertical direction only is considered which is also known as the Boussinesq's approximation.

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\nabla.\mathbf{u} = -\frac{1}{\rho}\nabla p + \nu\nabla^{\mathbf{2}}\mathbf{u} \tag{2.3}$$

where p is the pressure, ρ is the density

The momentum equation shown has the first term as change in momentum, advection and pressure gradient. In the simulations presented in this thesis, the thermodynamic equation has not been used and only a simple laminar jet simulation is presented.

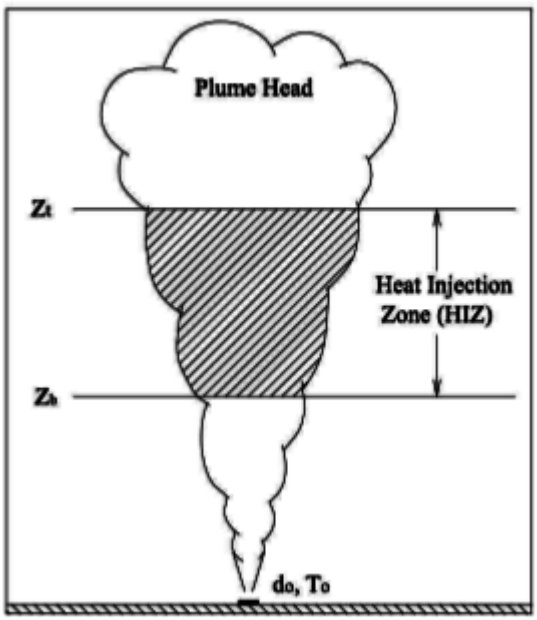

Figure 2.1: Computational domain simulation of the experiment (Konduri(2009))

2.3 Non Dimensionalization

As has been earlier discussed, in the absence of any other theory for turbulent flows such as clouds, similarity theory is used and to simulate large scale flows the fields are non dimensionalized using some reference variables. In the non dimensionalized equations, the characteristic parameters of the flow such as viscosity and density are combined to form non dimensional variables such as Reynolds Number and Froude's Number, so that any scale can be realized . Thus, the non-dimensional variables are formed by dividing the dimensional varible with some reference value of that same variable in the domain and in this study non dimensionalizing of the the navier stokes equations is done by d_0 i.e. the jet inlet diameter as the length scale and by u_0 i.e. the jet inlet velocity, we get the following equations

$$\nabla \cdot \mathbf{u} = 0 \tag{2.4}$$

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\nabla.\mathbf{u} = -\nabla p + \frac{1}{Re}\nabla^{\mathbf{2}}\mathbf{u} \tag{2.5}$$

where Re is the Reynolds number.

2.4 Discretizing Space Differentials

The spatial differentials include advection, diffusion and pressure gradient terms. They are discretized on a staggered grid known as Marker and Cell (MAC) mesh arrangement in which the pressure is taken at the centre and

the velocity vectors are taken at the centre of the edges as shown in the figure below

So the grids for u,v and p are staggered by half the grid spacing. Finite Volume method is used to discretize the terms in the NS equations over the control volume V. So the average velocity in x, y or z direction can be written as

$$\mathbf{u} = \frac{1}{V}\int_V \mathbf{u}(\mathbf{x})dv \tag{2.6}$$

Using the above finite volume approach, the advection(A), diffusion(D) and pressure gradient terms can be written as

$$\mathbf{A} = \nabla.\mathbf{uu} = \frac{1}{V}\int_V \nabla.\mathbf{uu}dv = \frac{1}{V}\oint_S \mathbf{u}(\mathbf{u}.\mathbf{n})ds \tag{2.7}$$

$$\mathbf{D} = \frac{1}{Re}\nabla^{\mathbf{2}}\mathbf{u} = \frac{1}{Re}\frac{1}{V}\int_V \nabla^{\mathbf{2}}\mathbf{u}dv = \frac{1}{Re}\frac{1}{V}\oint_S \nabla\mathbf{u}.\mathbf{n}ds \tag{2.8}$$

and

$$\nabla_{\mathbf{h}}p = \frac{1}{V}\int_V \nabla p dv = \frac{1}{V}\oint_S \nabla p.\mathbf{n}ds \tag{2.9}$$

The final form of the finite volume approach are then discretized using the central finite difference approximation on the MAC mesh with the u,v and p variables being discretized at their respective centres which are displaced by half mesh grid size from each other.

As can be seen in the above figure for the purpose of coding and mathematical treatment, a single grid with pressure as centre and centred at (i,j)

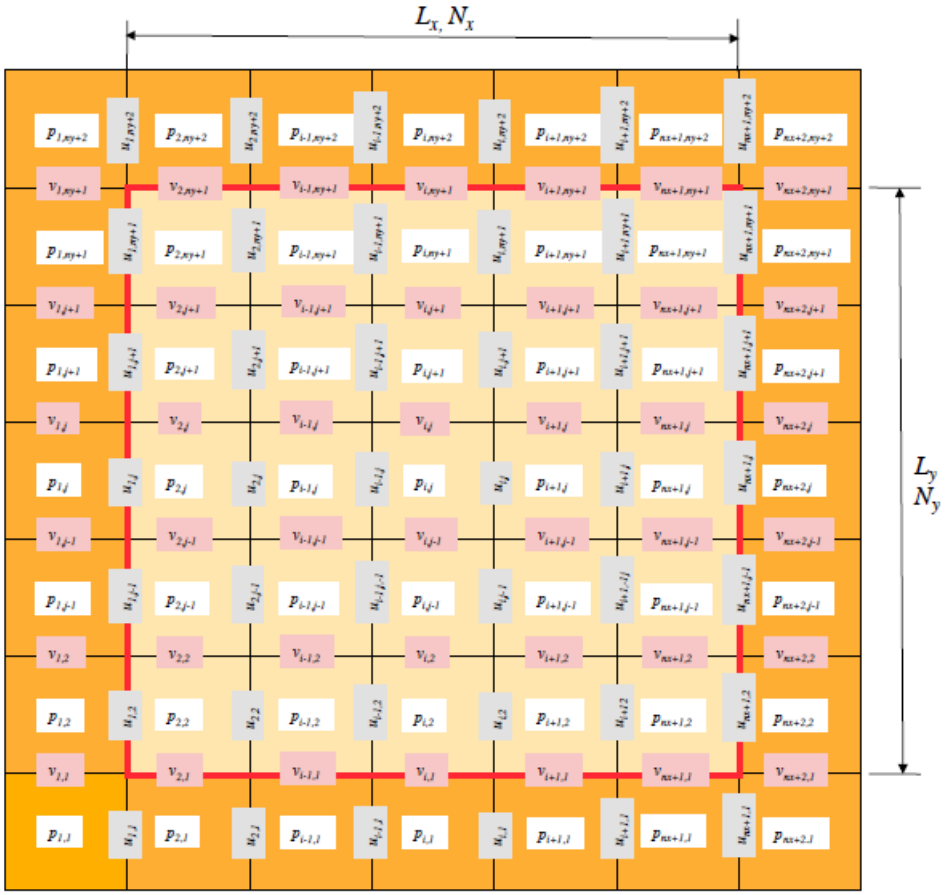

Figure 2.2: MAC Mesh (Tryggvason, 2012)

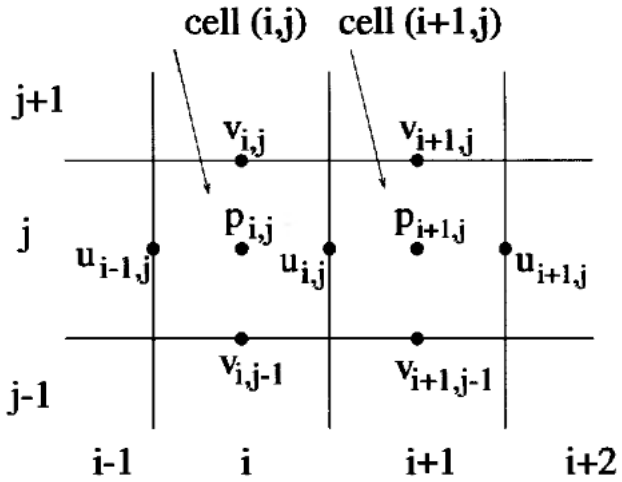

Figure 2.3: Staggered grid with pressure centre,(Gribel, Dornseifer and Neunhoeffer, 1998)

is used as the reference grid. For u velocity vector the centers are taken with reference to the pressure grid, and hence centered at (i+1/2,j). Similarly for v velocity vector, the center of grid is taken as (i,j+1/2) with reference to the pressure grid. Although denoting the u and v velocity centres as (i+1/2,j) and (i,j+1/2) in the mathematical notation, whereas writing from the coding point of view as we cannot store arrays at half index values, we store them at integral index values and rather shift the (x,y) grids of u, v and p accordingly. The discretized advection, diffusion and pressure gradient equations are shown below:

$$A_x = \nabla.uu = \left[\frac{\partial(u.u)}{\partial x}\right]_{i,j} + \left[\frac{\partial(u.v)}{\partial y}\right]_{i,j} \tag{2.10}$$

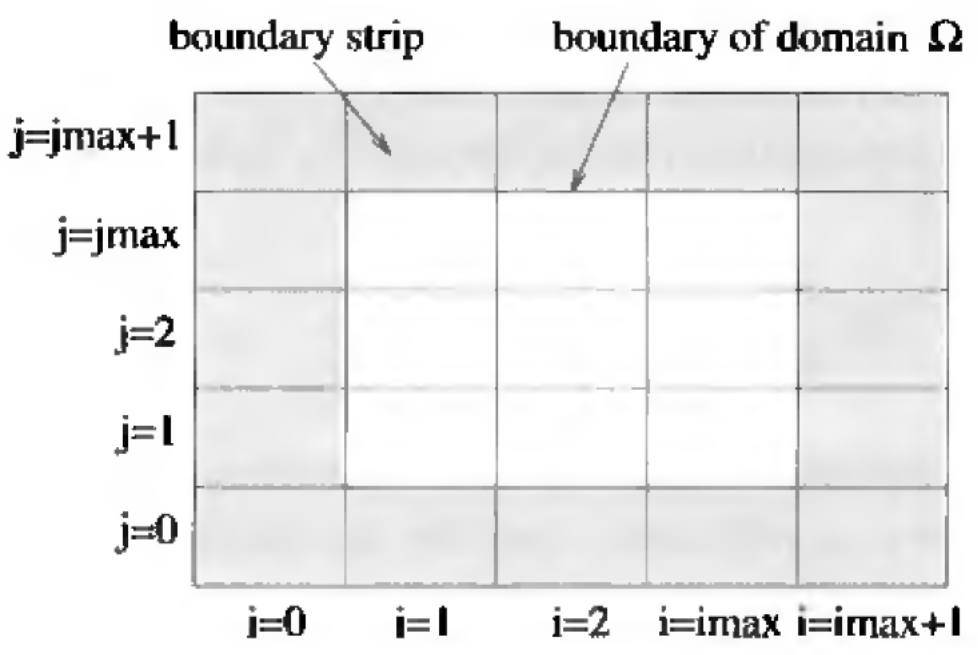

Figure 2.4: Staggered grid with ghost cells around(Gribel, Dornseifer and Neunhoeffer, 1998)

where,

$$\left[\frac{\partial(u.u)}{\partial x}\right]_{i,j} = \frac{1}{\delta x}\left(\left(\frac{u_{i,j}+u_{i+1,j}}{2}\right)^2 - \left(\frac{u_{i-1,j}+u_{i,j}}{2}\right)^2\right) \tag{2.11}$$

and

$$\left[\frac{\partial(u.v)}{\partial y}\right]_{i,j} = \frac{1}{\delta y}\left(\left(\frac{v_{i,j}+v_{i+1,j}}{2}\right).\left(\frac{u_{i,j}+v_{i,j+1}}{2}\right) - \left(\frac{v_{i,j-1}+v_{i+1,j-1}}{2}\right).\left(\frac{u_{i,j-1}+u_{i,j}}{2}\right)\right) \tag{2.12}$$

$$A_y = \nabla.vv = \left[\frac{\partial(u.v)}{\partial x}\right]_{i,j} + \left[\frac{\partial(v.v)}{\partial y}\right]_{i,j} \tag{2.13}$$

where,

$$\left[\frac{\partial(v.v)}{\partial y}\right]_{i,j} = \frac{1}{\delta x}\left(\left(\frac{v_{i,j}+v_{i,j+1}}{2}\right)^2 - \left(\frac{v_{i,j-1}+v_{i,j}}{2}\right)^2\right) \tag{2.14}$$

and

$$\left[\frac{\partial(u.v)}{\partial x}\right]_{i,j} = \frac{1}{\delta x}\left(\left(\frac{u_{i,j}+u_{i,j+1}}{2}\right).\left(\frac{v_{i,j}+v_{i+1,j}}{2}\right) - \left(\frac{u_{i-1,j}+u_{i-1,j+1}}{2}\right).\left(\frac{v_{i-1,j}+v_{i,j}}{2}\right)\right) \tag{2.15}$$

$$D_x = \frac{1}{Re}\nabla^2.u = \frac{1}{Re}\left(\left[\frac{\partial^2 u}{\partial x^2}\right]_{i,j} + \left[\frac{\partial^2 \mathbf{u}}{\partial y^2}\right]_{i,j}\right) \tag{2.16}$$

where,

$$\left[\frac{\partial^2 u}{\partial x^2}\right]_{i,j} = \frac{u_{i+1,j}-2u_{i,j}+u_{i-1,j}}{(\delta x)^2} \tag{2.17}$$

and

$$\left[\frac{\partial^2 u}{\partial y^2}\right]_{i,j} = \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{(\delta y)^2} \tag{2.18}$$

$$D_y = \frac{1}{Re}\nabla^2.v = \frac{1}{Re}\left(\left[\frac{\partial^2 v}{\partial x^2}\right]_{i,j} + \left[\frac{\partial^2 v}{\partial y^2}\right]_{i,j}\right) \tag{2.19}$$

where,

$$\left[\frac{\partial^2 v}{\partial x^2}\right]_{i,j} = \frac{v_{i+1,j} - 2v_{i,j} + v_{i-1,j}}{(\delta x)^2} \tag{2.20}$$

and

$$\left[\frac{\partial^2 v}{\partial y^2}\right]_{i,j} = \frac{v_{i,j+1} - 2v_{i,j} + v_{i,j-1}}{(\delta y)^2} \tag{2.21}$$

$$\left[\frac{\partial p}{\partial x}\right]_{i,j} = \frac{p_{i+1,j} - p_{i,j}}{\delta x} \tag{2.22}$$

$$\left[\frac{\partial p}{\partial y}\right]_{i,j} = \frac{p_{i,j+1} - p_{i,j}}{\delta y} \tag{2.23}$$

where A_x, D_x, A_y and D_y mean advection and diffusion in x and y directions as we are dealing with 2D flows only in this thesis.

2.5 Discretizing Time Differentials

For discretizing time differentials, various good methods are being used these days such as Range Kutta Method, Adam Bashforth's method, Euler's method and various other methods. For the case of simulations performed in this study, simple euler's method was used. The following equations show the use of Euler and Adam Bashforth's method which will be used for further

simulations.

$$\left[\frac{\partial u}{\partial t}\right]_{i,j} = f^n{}_{i,j} \tag{2.24}$$

Euler's Method:

$$\frac{u^{n+1}_{i,j} - u^n{}_{i,j}}{\delta t} = f^n{}_{i,j} \tag{2.25}$$

Adam Bashforth's Method:

$$\frac{u^{n+1}_{i,j} - u^n{}_{i,j}}{\delta t} = 1.5f^{n+1} - 0.5f^n{}_{i,j} \tag{2.26}$$

2.6 Algorithm

If we see the continuity and momentum equation in the Navier Stokes equations, it can be very easily observed that there is no coupling term to include the effect of one in another and vice versa. So to facilitate that, Chorin(1968) introduced the fractional step method or the projection method to solve the Navier Stokes equations as a single system of equations satisfying both the continuity as well as momentum equations. In this method, the NS equation is broken into two parts and a dummy velocity is included. The first step being the calculation of the dummy velocity without the pressure field and the second step being, moving in time to impose the pressure field on the divergence free velocity. This equation is also used to be as an input to the continuity equation to form what is known as Pressure Poisson Equation. The algorithm hence goes like shown below:

- Set $t := 0, n := 0$
- Assign initial values to u, v, p
- While $t < t_{end}$
- Set δt according to the Stability criteria
- Set boundary values for u, v and p
- Compute the u and v dummy velocities
- Compute the right hand side of pressure poisson equation(PPE) i.e. the divergence of dummy velocities
- Solve the Pressure Poisson Equation(PPE)
- Compute the corrected velocities at the next time step after incorporating the pressure term.

$t := t + \delta t$

$n := n + 1$

So, the following equations results after applying the Chorin's Fractional Step method to the solution of Navier Stokes Equations

$$u^{(n+1)} = u^{(n)} + \delta t\left[\frac{1}{Re}\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) - \frac{\partial u.u}{\partial x} - \frac{\partial u.v}{\partial y} + g_x - \frac{\partial p}{\partial x}\right] \tag{2.27}$$

and

$$v^{(n+1)} = v^{(n)} + \delta t\left[\frac{1}{Re}\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right) - \frac{\partial u.v}{\partial x} - \frac{\partial v.v}{\partial y} + g_y - \frac{\partial p}{\partial y}\right] \tag{2.28}$$

can be broken as

$$u^* = u^{(n)} + \delta t\left[\frac{1}{Re}\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) - \frac{\partial u.u}{\partial x} - \frac{\partial u.v}{\partial y} + g_x\right] \tag{2.29}$$

$$v^* = v^{(n)} + \delta t\left[\frac{1}{Re}\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right) - \frac{\partial u.v}{\partial x} - \frac{\partial v.v}{\partial y} + g_y\right] \tag{2.30}$$

and

$$u^{(n+1)} = u^* - \delta t\frac{\partial p}{\partial x} \tag{2.31}$$

and

$$v^{(n+1)} = v^* - \delta t\frac{\partial p}{\partial y} \tag{2.32}$$

Now in the continuity equation

$$\frac{\partial u^{n+1}}{\partial x} + \frac{\partial v}{\partial y} = 0 \tag{2.33}$$

with reference to pressure grid we have

$$\frac{u^{n+1}_{i+1/2,j} - {u^{n+1}}_{i-1/2,j}}{\delta x} + \frac{{v^{n+1}}_{i,j+1/2} - {u^{n+1}}_{i,j-1/2}}{\delta y} = 0 \tag{2.34}$$

Inserting u^{n+1} and v^{n+1} from the pressure corrector step we get the elliptic Pressure Poisson Equation as

$$\frac{{p^{n+1}}_{i+1,j} - 2{p^{n+1}}_{i,j} + {p^{n+1}}_{i-1,j}}{(\delta x)^2} + \frac{{p^{n+1}}_{i,j+1} - 2{p^{n+1}}_{i,j} + {p^{n+1}}_{i,j-1}}{(\delta y)^2} = \tag{2.35}$$

$$\left\{ \frac{{u^*}_{i+1/2,j} - {u^*}_{i-1/2,j}}{\delta x} + \frac{{v^*}_{i,j+1/2} - {v^*}_{i,j-1/2}}{\delta y} \right\}$$

which can be written as

$$\nabla^2 p = \frac{\nabla u^*}{\delta t} \tag{2.36}$$

2.6.1 Solving Pressure Poisson Equation

The Pressure Poisson Equation is a laplacian equation which can be solved using various methods. As the NS equations are elliptic they have a particular set of solutions as against hyperbolic equations. It has been observed in this study as well as in the previous studies that around 75% of the computational resources are consumed by the pressure poisson equation only. As

our flow is incompressible the pressure poisson equation is influenced by the advection and diffusion terms only from the momentum equation. For the Pressure Poisson Equation derived in the above section the following values are required for the same:

$$p_{0,j}, p_{i_{max}+1,j},\ j = 1, ..., j_{max}$$
$$p_{i,0}, p_{i,j_{max}+1},\ i = 1, ..., i_{max}$$

along with the following values of u^* and v^* to compute rhs of PPE

$$u^*_{0,j}, u^*_{i_{max}+1,j},\ j = 1, ..., j_{max}$$
$$v^*_{i,0}, v^*_{i,j_{max}+1},\ i = 1, ..., i_{max}$$

For the equation (2.34) at (1,0) we get

$$\frac{{u^{n+1}}_{1,0} - {u^*}_{1,0}}{\delta t} = \frac{p^{n+1}_{0,j} - p^{n+1}_{1,j}}{\delta x} \tag{2.37}$$

Now inserting ${p^{n+1}}_{0,j}$ from the above equation to the discretized form of pressure poisson equation for i=1, we have,

$$\frac{{p^{n+1}}_{2,j} - {p^{n+1}}_{1,j}}{\delta x} + \frac{{p^{n+1}}_{1,j+1} - 2{p^{n+1}}_{1,j} + {p^{n+1}}_{1,j-1}}{\delta y} = \tag{2.38}$$
$$\left\{ \frac{{u^{*n}}_{1,j} - {u^n+1}_{0,j}}{\delta x} + \frac{{v^{*n}}_{1,j} - {v^{*n}}_{1,j-1}}{\delta y} \right\}$$

Hence, we can say from the above equation that the PPE does not depend on the values of ${u^{*n}}_{0,j}$ as ${u^{*n}}_{0,j}$ does not occur in the combined equation.Hence we can choose any value of ${u^{*n}}_{0,j}$ and the easiest is ${u^{*n}}_{0,j} = {u^{n+1}}_{0,j}$, which

in turn from the first step of pressure corrector scheme gives us ${p^{n+1}}_{0,j} = {p^{n+1}}_{1,j}$ and hence which may be called the neumann boundary condition for pressure.

Now after applying the pressure boundary condition, we may solve the PPE using Successive Over Relaxation method. Hence from (2.38) we have:

$$ {p^{n+1}}_{i,j} = \beta\left(\frac{2}{(\delta x)^2}+\frac{2}{(\delta y)^2}\right)^{-1}\left(\frac{{p^{n+1}}_{i+1,j}+{p^{n+1}}_{i-1,j}}{(\delta x)^2}+\frac{{p^{n+1}}_{i,j+1}+{p^{n+1}}_{i,j-1}}{(\delta y)^2}-\right. \tag{2.39} $$

$$ \left.\left\{\frac{{u^*}_{i+1/2,j}-{u^*}_{i-1/2,j}}{\delta x}+\frac{{v^*}_{i,j+1/2}-{v^*}_{i,j-1/2}}{\delta y}\right\}\right)+(1-\beta){p^{n+1}}_{i,j} $$

Iterating this equation until the error reduces the threshold limit, we get the converged pressure at every grid point.β used in the scheme was 1.7 as it has been well documented that it converges best there. Although SOR method is a useful and easy tool to find the solution of matrices, but still being an explicit method, I was able to get good results only with the dimensional code whereas non dimensional code still remains an issue. Also using the matrix inversion to find out the value of the pressure 2D array. Other methods such as multigrid methods and methods involving parallel computing. Apart from this various packages such as FISHPACK, PETSc, HYPRE, Trilinos etc are also available freely which are optimized to solve the pressure poisson equation.

2.6.2 Stability Criteria

The stability criteria in the code used are applied by fixing the grid size and computing the value of time step at every iteration to avoid oscillations in the fields. There are three conditions that are very well documented for the stability criteria. They are:

$$\frac{2\delta t}{Re} < \left(\frac{1}{\delta x^2} + \frac{1}{\delta y^2}\right)^{-1} \tag{2.40}$$

$$|u_{max}|\delta t < \delta x \tag{2.41}$$

$$|v_{max}|\delta t < \delta y \tag{2.42}$$

The last two equations are known as Courant-Friedrichs-Levy(CFL) criteria which means that in the time δt no fluid element should travel a distance greater than δx or δy. So to incorporate the above three criteria in the code the following step size control equation was used in the code following Boersma.

$$\delta t = \tau\left(\frac{Re}{2}\left(\frac{1}{\delta x^2 + \delta y^2}\right)^{-1} + \frac{\delta x}{|u_{max}|} + \frac{\delta y}{|v_{max}|}\right) \tag{2.43}$$

Although any value of τ from [0,1] could be used but 0.5 was used in the code.

2.7 Boundary Conditions

For the particular solution of any differential equation, initial and boundary conditions are required. In the case of our laminar jet flow, the initial condition is every term as equal to zero. For the Boundary conditions of 2D rectangular domain, we have four edges to provide the boundary conditions. For the lower edge the boundary condition is no slip everywhere except at the jet inlet wherein, at every time step we will have the v vertical velocity of the jet as verified experimentally as hyperbolic tan profile over the years. Now coming to the two lateral walls, we are providing no slip boundary conditions and for the top edge we had to provide a boundary condition such that the jet exits the domain without any reflection of any wave. So, gradient of u and v in the direction of jet flow is taken to be equal to zero for the flow to exit peacefully. Although for the case of simplicity we have used the no slip and gradient zero boundary conditions at the entry exit and side walls, but to really simulate the cloud flows, we should take the convective boundary condition at the top boundary and traction free boundary condition at the lateral walls. The application of the boundary conditions is as shown:
Boundary Conditions that can be applied at the exit boundary are:
Zero-gradient condition:

$$\frac{\partial \mathbf{u}}{\partial y} = 0 \tag{2.44}$$

Convective Boundary Condition:

$$\frac{\partial \mathbf{u}}{\partial t} = -U\frac{\partial \mathbf{u}}{\partial y} \tag{2.45}$$

where U is the mean velocity at the exit boundary.

Traction Free Boundary Condition:

$$\sigma_{i,j}.n_j = 0 \quad \text{and} \quad \sigma_{i,j} = -P\delta_{i,j} + \nu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) \tag{2.46}$$

No Slip Boundary Condition:

$$\mathbf{u} = 0$$

In addition, a pressure boundary condition needs to be enforced on all the boundaries which is already discussed in the section containing the algorithm.

2.8 Visualization Techniques

The simulations carried out by Direct Numerical Simulations (DNS) generates vast amount of data and to properly interpret that data visualization techniques are required. The flow is visualized by plotting at every time step, the vorticity, streamfunction or the quiver plot of the velocity field. Although a lot of packages are available, MATLAB and Ferret were used for the purpose of this study.

Chapter 3

Validation

As the code was developed in-house, so before moving to the desired particular cases, it is important that the code be validated against well established test cases. The test case should include phenomena very similar to the desired physical phenomena for which the code would have been thought to have been targeted. The targeted simulation was of a heated jet which was not possible in the duration of project, hence the unheated 2D dimensional laminar flow getting generated was needed to be validated. The code has been validated against the standard case of lid driven cavity flow and results found to be matching qualitatively within the range of the established results.

3.1 Validating against Lid Driven Cavity Flow

Although not practically realizable 2D lid driven cavity flows have offered a good understanding of various fluid mechanics phenomena in the past. In the study presented the validation was done against a 2D lid driven cavity flow and hence the comparison is presented between the established results and the results produced by our code.

The 2-D cavity under consideration has left, bottom and right walls stationary and the top wall moves uniformly. The aspect ratio of the cavity is chosen to be 1 i.e. a square cavity of size 1×1. The top wall is given a constant horizontal velocity at every time step; this serves as the no-slip boundary condition at the top wall. On all the other walls the no-slip boundary condition takes the form of zero velocity condition. The analysis is done for Re =1000 by setting $u = 10$ cm/s, $L = 1$ cm and $\mu = 0.01$ poise in the present dimensional code. The resulting data is made dimensionless by using the top plate velocity and the cavity size as the scales. The results may then be compared with those of Prasanth and Erturk having the same Reynolds number of 1000. Figure 3.1 shows that the results of present simulations agree fairly well with those in the literature although the resolution in the present simulation is quite coarse. All the essential trends are well-captured by the present code. It is believed that the match would improve with improvements in the spatial resolution of the grid, switchover to an improved Poisson solver and elaborate, higher-order discretizations for space and time.

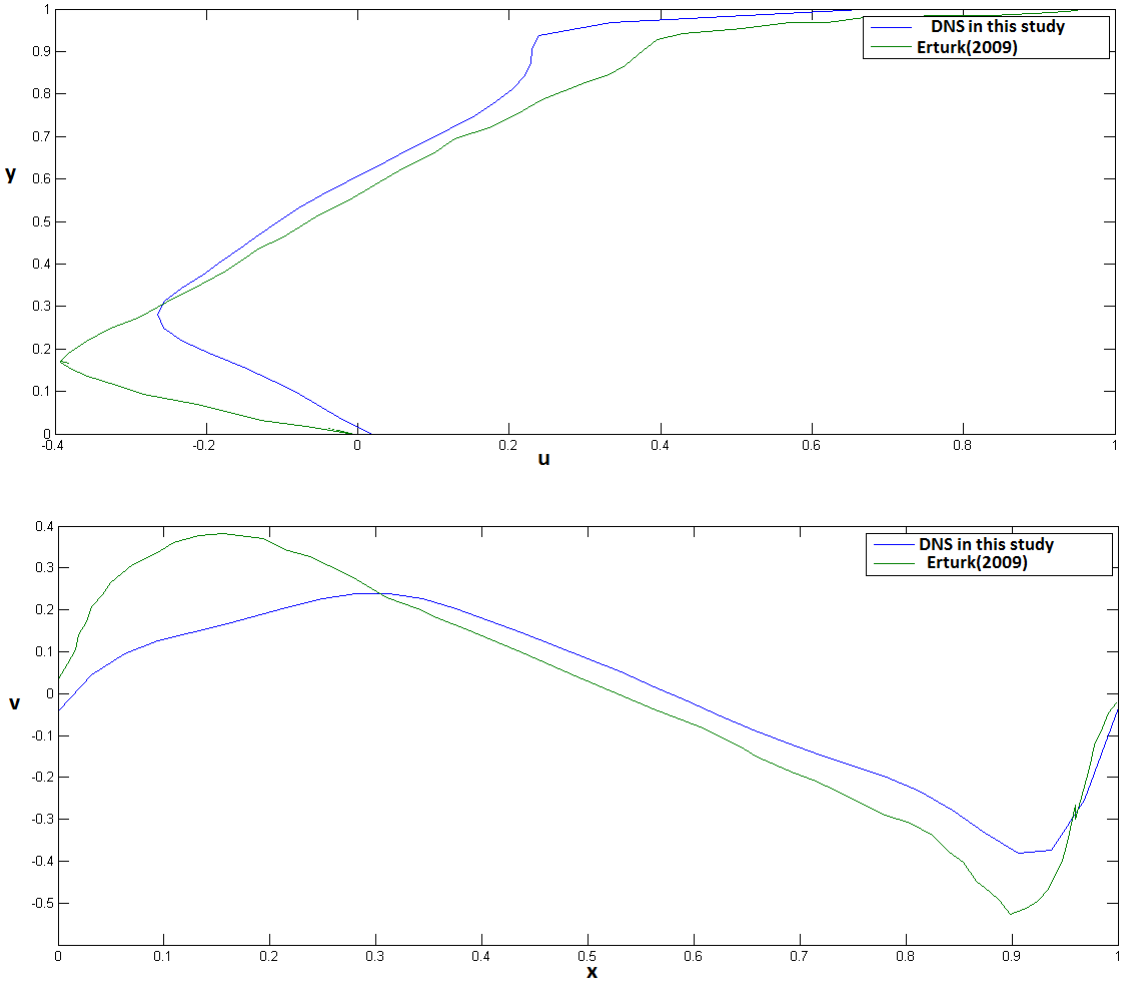

Figure 3.1: Lid-driven cavity flow: Variation of each component of velocity in the direction perpendicular to itself as seen in the central plane.

Chapter 4

Results and Discussion

After testing the code for the 2-D lid driven cavity flow, the 2-D laminar jet flow was simulated. Theory of self-similar 2-D laminar jets (White, 1991) yields the expectations which may serve as a useful check for further validiting the code. Two such expectations are: first, the jet centerline velocity should vary as negative one-third power of the axial distance and second, the jet width should vary as positive two-third power of the axial distance. Figure 4.1 shows the plots of centerline velocity and jet width versus axial distance. It may be observed that the trend towards self-similar behaviour in centerline velocity is observed only farther in the domain which is due to the fact that the length of the domain is just 6 times the diameter of the entering jet so that the jet, although steady, has still not become independent of the initial conditions. The trend in the jet width can be explained based on very limited spatial resolution of the present grid. The jet width

increase is not captured until it becomes more than the grid size. Therefore one has to observe the regions of the curve showing change in the jet width. More simulations will be performed later with the increased domain size and resolution to test these aspects further.

The results for the vorticity in the laminar jet flow are now presented. Vorticity is the curl of the velocity vector field and since the present simulations are 2-D, only one component of the vorticity vector $\omega_z = \partial u/\partial y - \partial v/\partial x$ is relevant. As the code development started of with MATLAB, hence the MATLAB laminar flow vorticity at different time intervals are shown in this section. As the velocity quiver vectors were also plotted along with the vorticity, they were seen to be perpendicular to the laminar flow at the boundary of vorticity, thus indicating some sort of entrainment. The vorticity plots show two lobes coming out from the bottom boundary in accordance with the boundary condition given there. The two lobes rise up in the domain to form independent vortices and this process continues until dynamical stability is achieved. Similar work was done on FORTRAN 90 and plotted on Ferret. In the simulations performed, it was observed that the simulation would blow up if unrealistic values of δt, δx and δy were given. So, as initially the process of using δt was not done by the calculation done by program, hence this blowing up was thought to be the reason of that. But in the last version of the code, the stability criteria is applied and value of δt found out. So, any value of δx and δy should work, but still success has not been achieved completely on that front and Boersma and Griebel's code are prov-

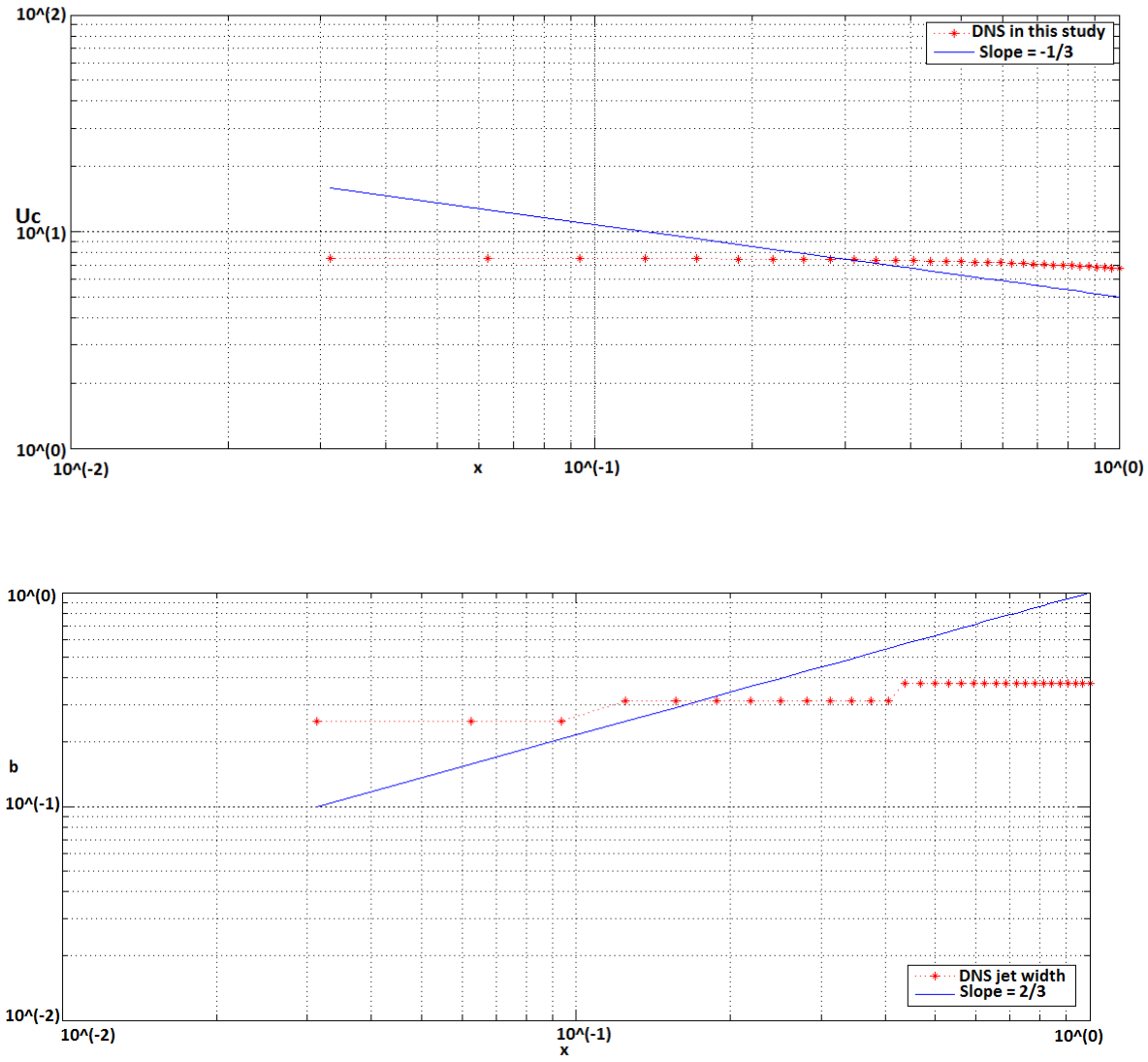

Figure 4.1: Streamwise variation of the centerline velocity and the jet width in a steady-state laminar 2-D jet. Data points show the simulation results and the solid line shows the self-similar expectation from the theory.

ing to be very good directions towards final destination. Another important check that was being supposedly missed until recently, and pointed out in a technical discussion with Mr Saurabh (IISc), was the implementation of the check divergence criteria. This is believed to be one of the lacunae because of which the non-dimensional code is blowing up completely. This may be due to the explicit nature of the SOR technique used for solving the Pressure Poisson Equation. But still the Griebel's C code which has been shown to be running on SOR for the Non-dimensionalization needs to be checked for finding out the exact bug in the program. Thus the final non-dimensionalized code needs a check on SOR and further development of an in-house poisson solver which will cater to the needs of the pressure poisson equation. It is believed that FISHPACk developed in early eighties by NCAR will also be good for the same purpose.

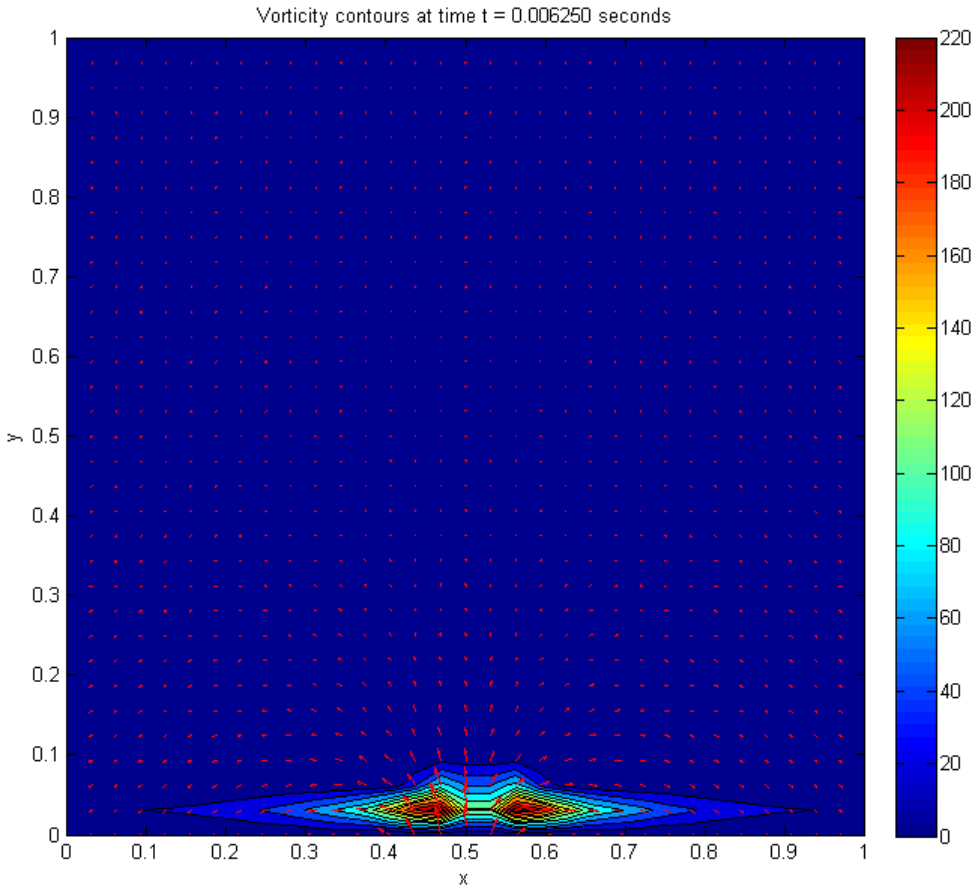

Figure 4.2: MATLAB Simulation for t = 0.00625 seconds(vorticity in the units s^{-1})

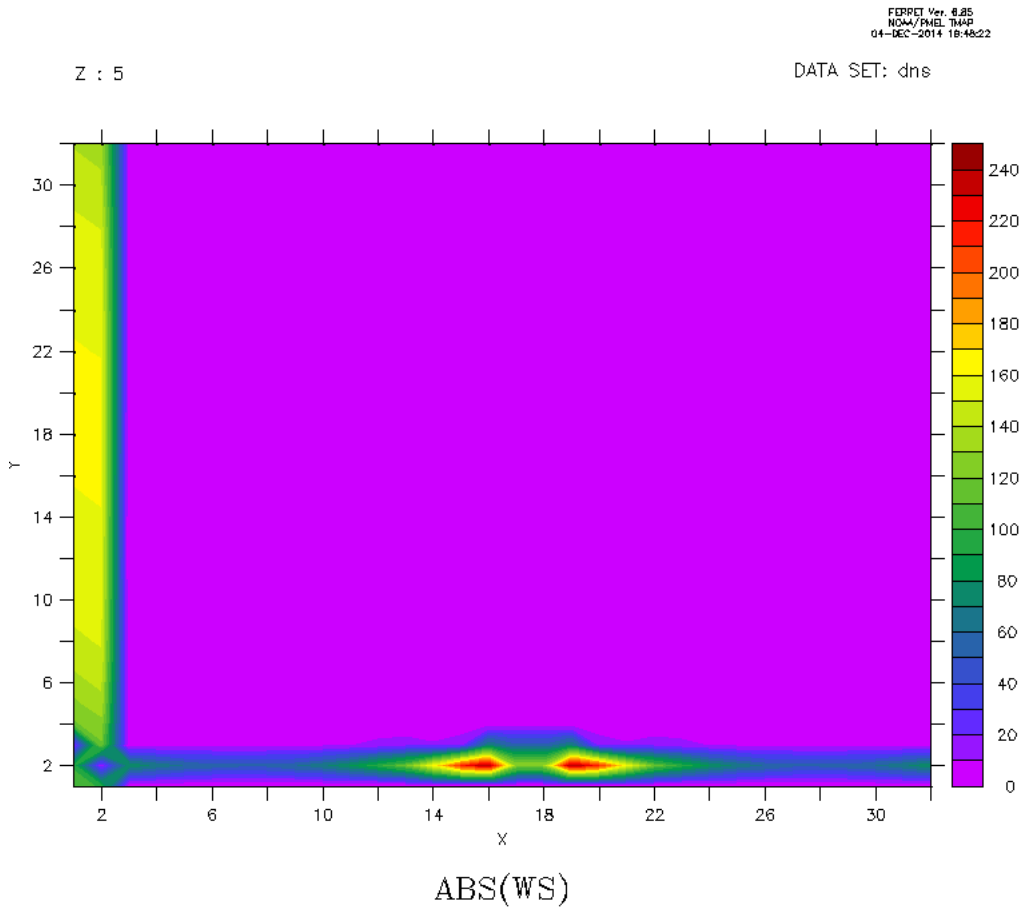

Figure 4.3: FORTRAN 90 Simulation for t = 0.00625 seconds(vorticity in the units s^{-1})

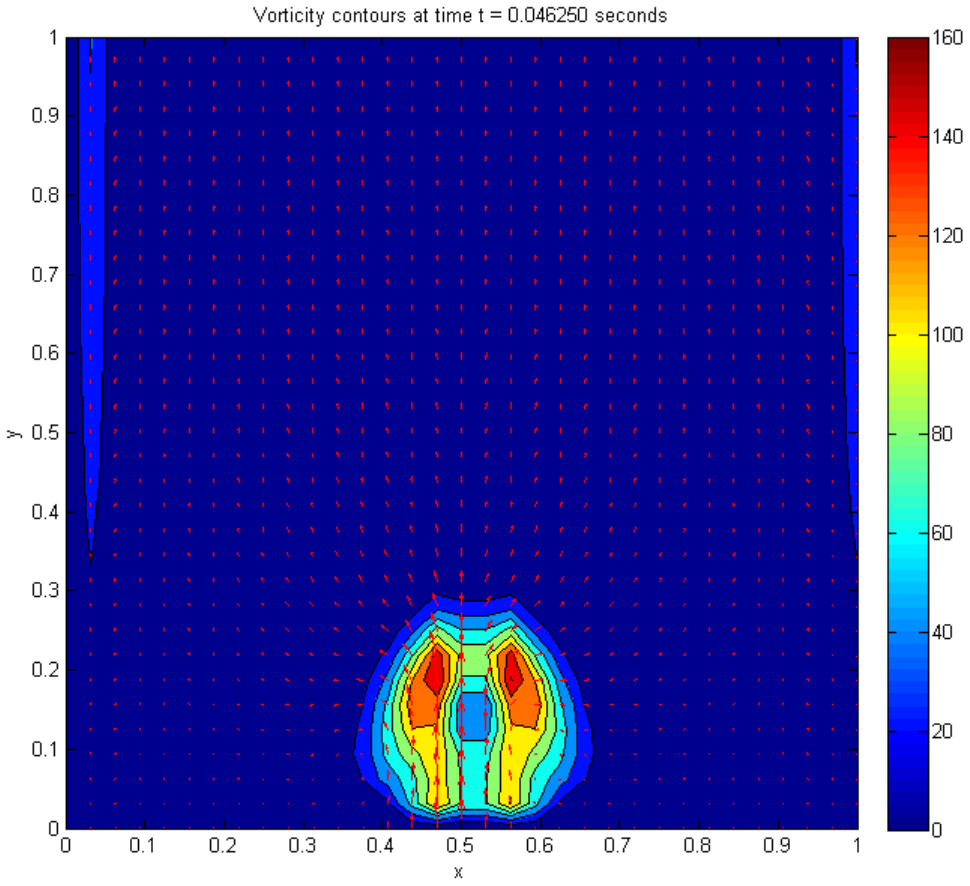

Figure 4.4: MATLAB Simulation for t = 0.04625 seconds(vorticity in the units s^{-1})

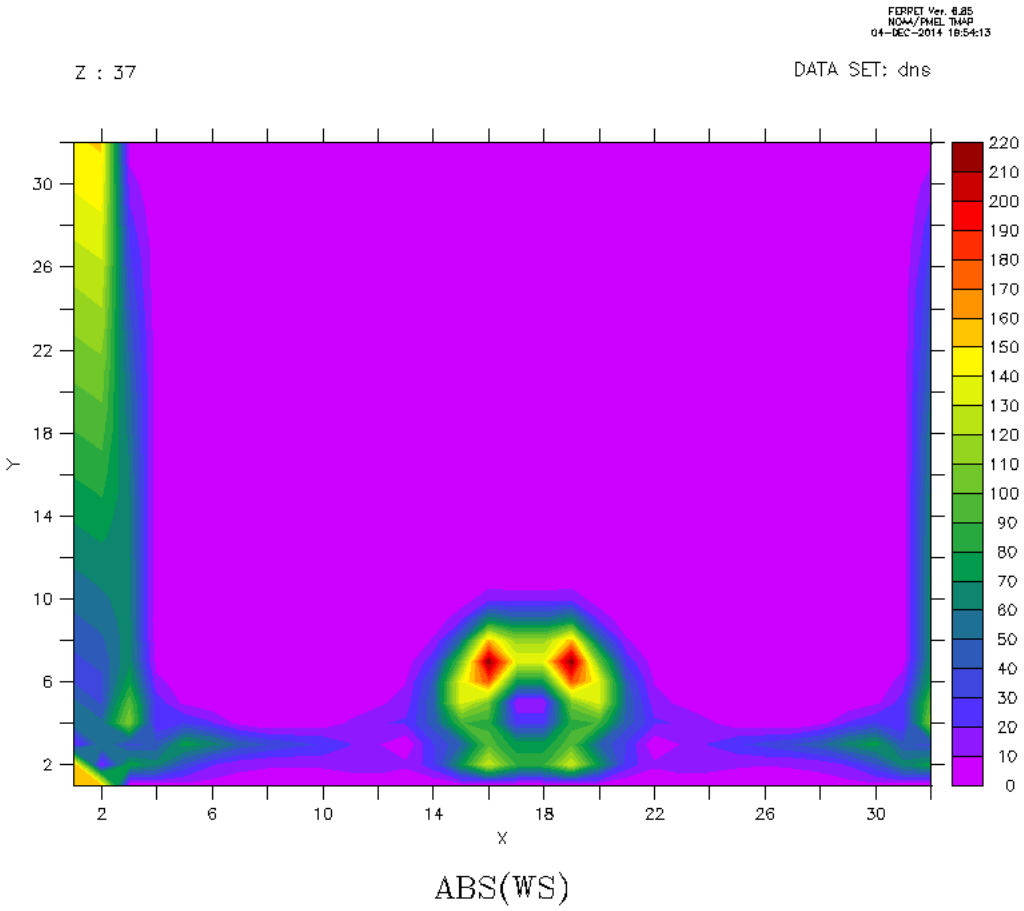

Figure 4.5: FORTRAN 90 Simulation for t = 0.04625 seconds(vorticity in the units s^{-1})

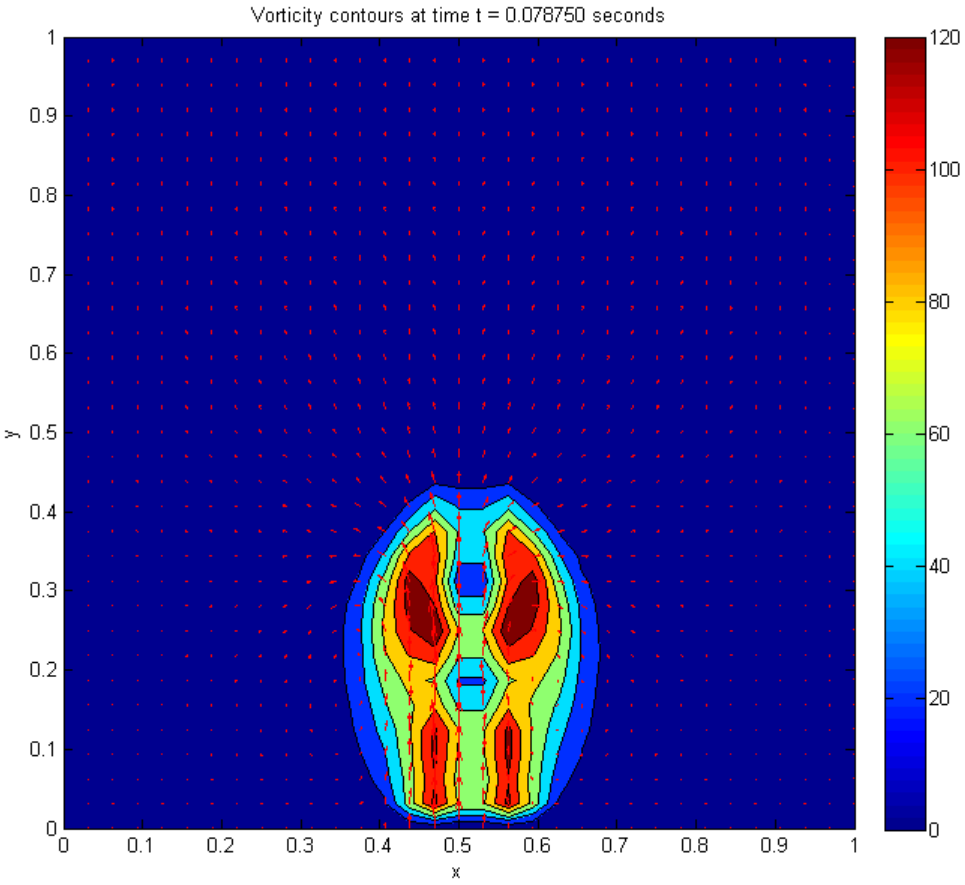

Figure 4.6: MATLAB Simulation for t = 0.07875 seconds(vorticity in the units s^{-1})

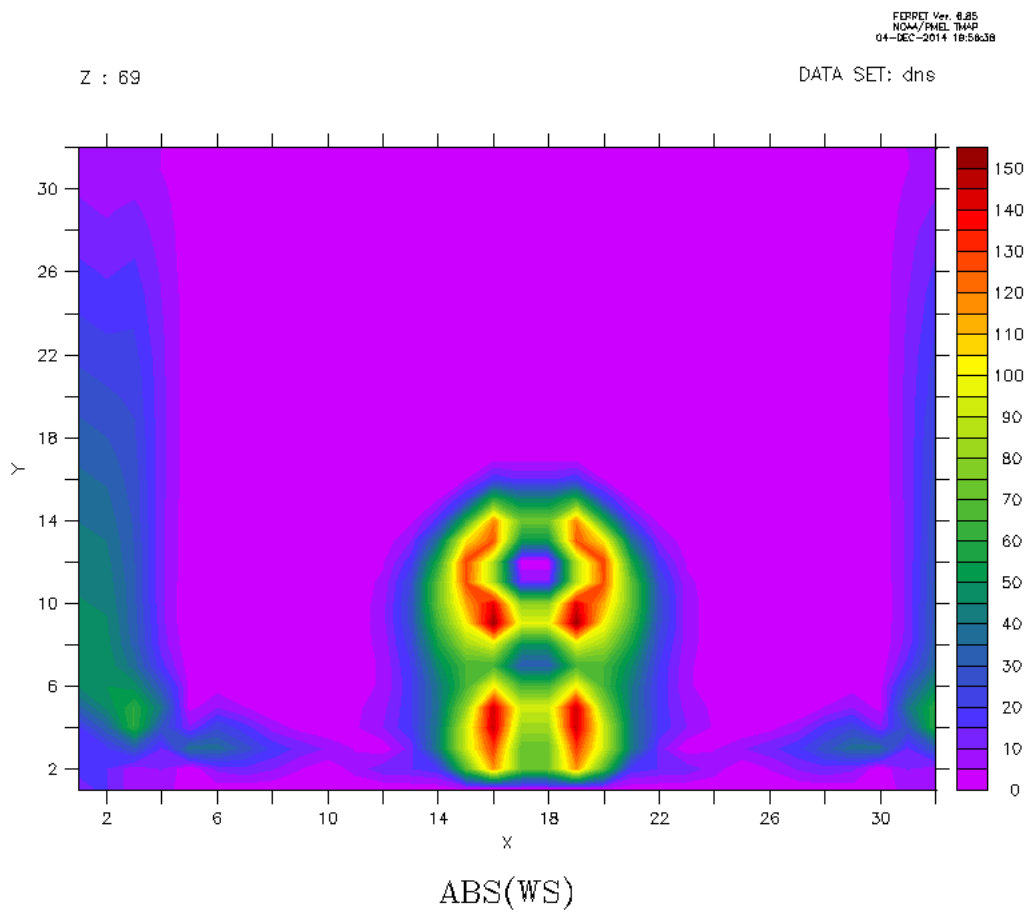

Figure 4.7: FORTRAN 90 Simulation for t = 0.07875 seconds(vorticity in the units s^{-1})

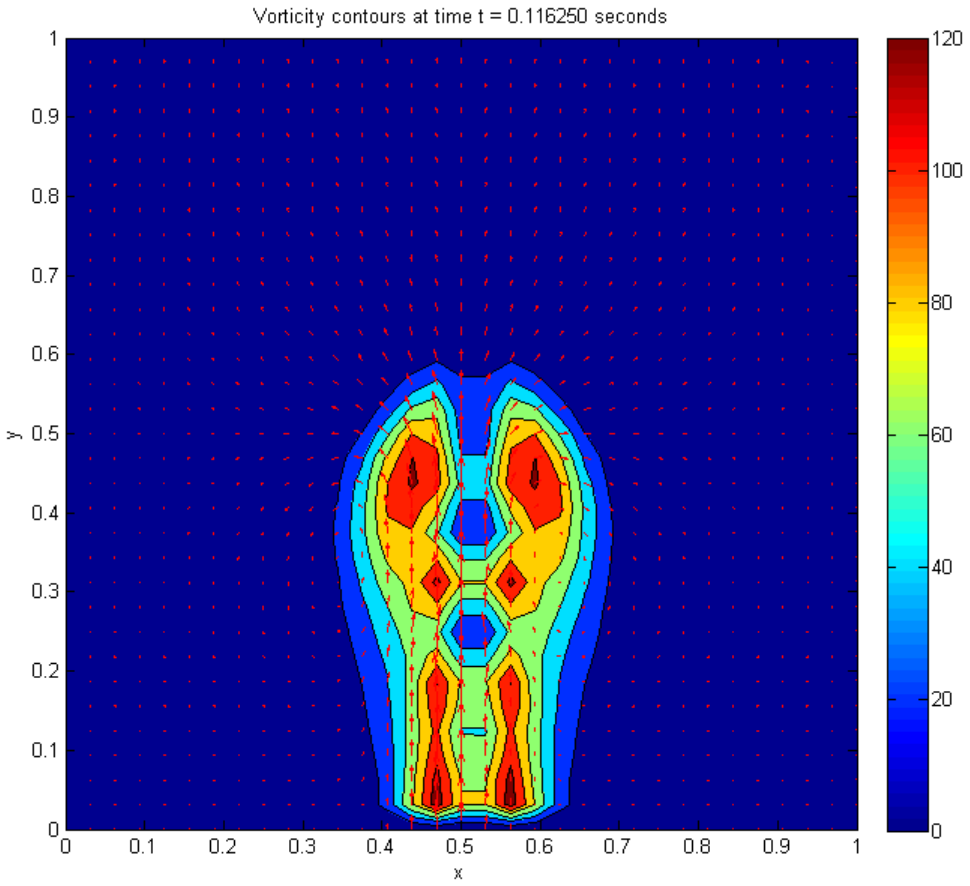

Figure 4.8: MATLAB Simulation for t = 0.11625 seconds(vorticity in the units s^{-1})

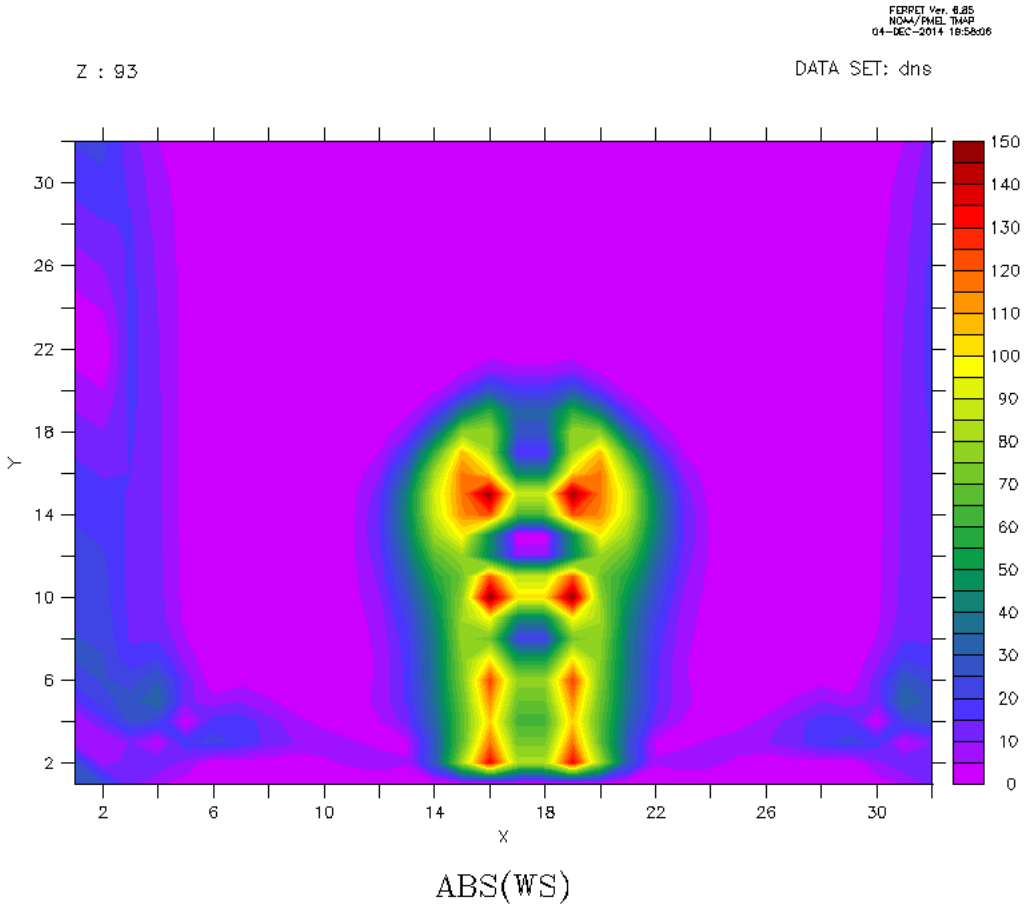

Figure 4.9: FORTRAN 90 Simulation for t = 0.11625 seconds(vorticity in the units s^{-1})

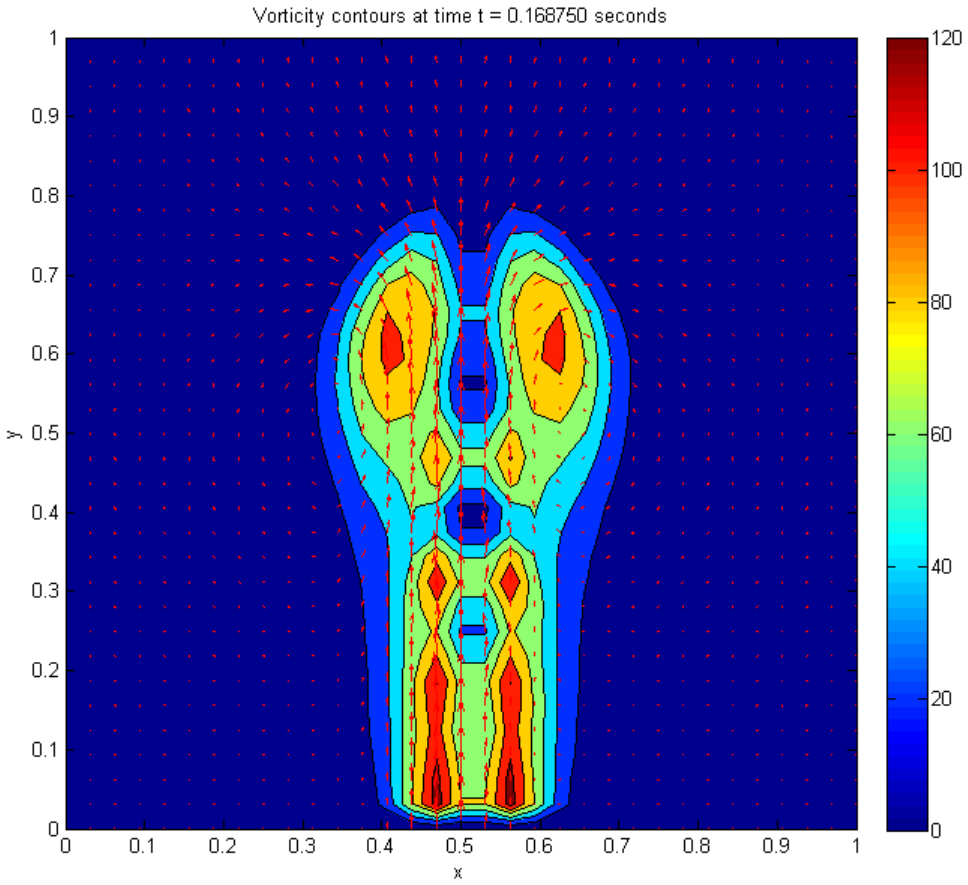

Figure 4.10: MATLAB Simulation for t = 0.16875 seconds(vorticity in the units s^{-1})

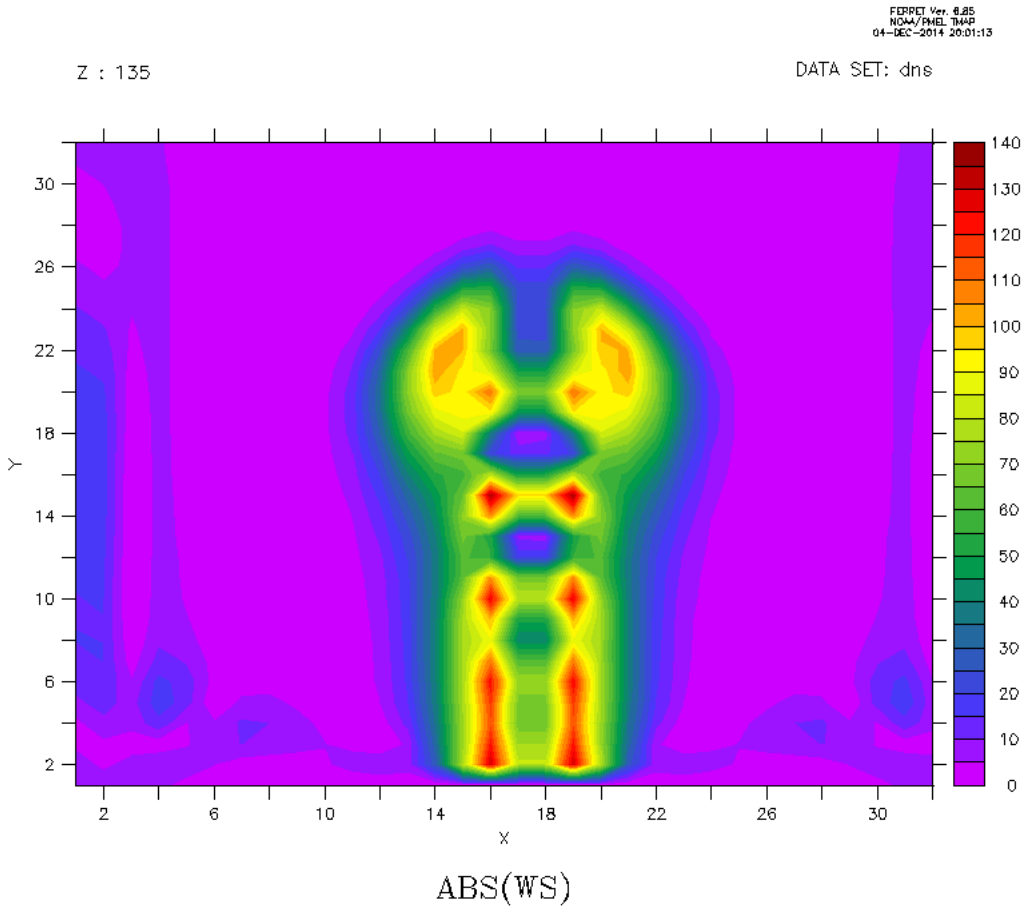

Figure 4.11: FORTRAN 90 Simulation for t = 0.16875 seconds(vorticity in the units s^{-1})

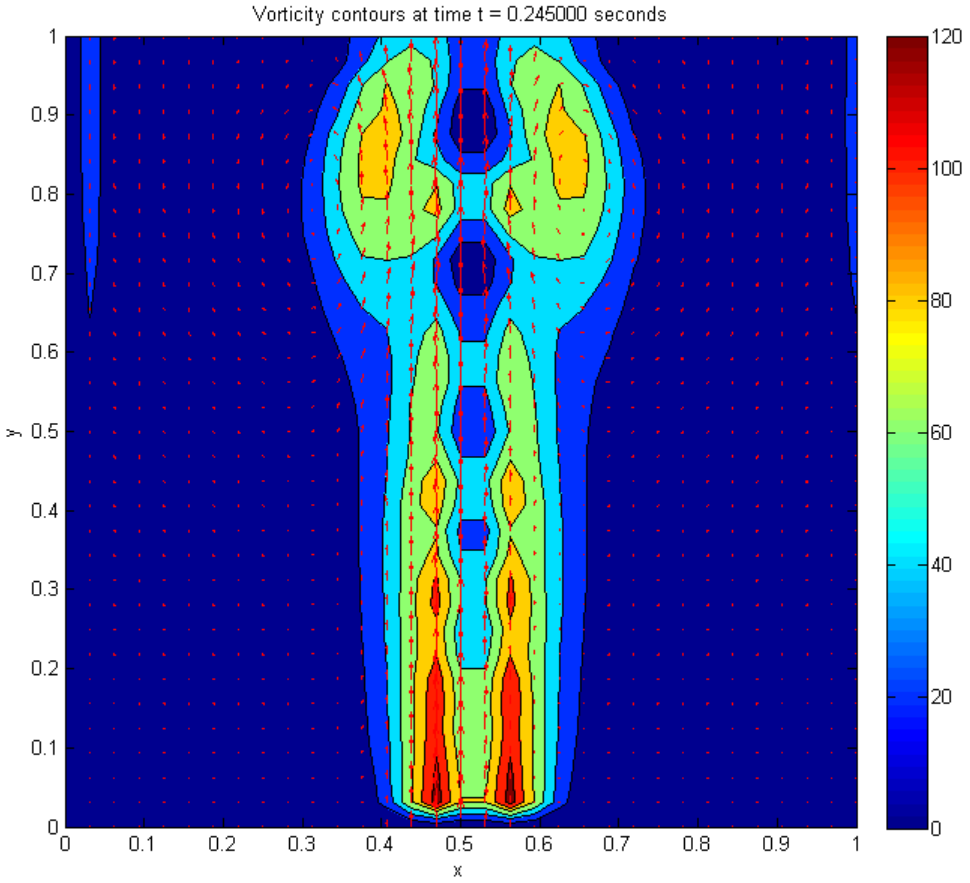

Figure 4.12: MATLAB Simulation for t = 0.245 seconds(vorticity in the units s^{-1})

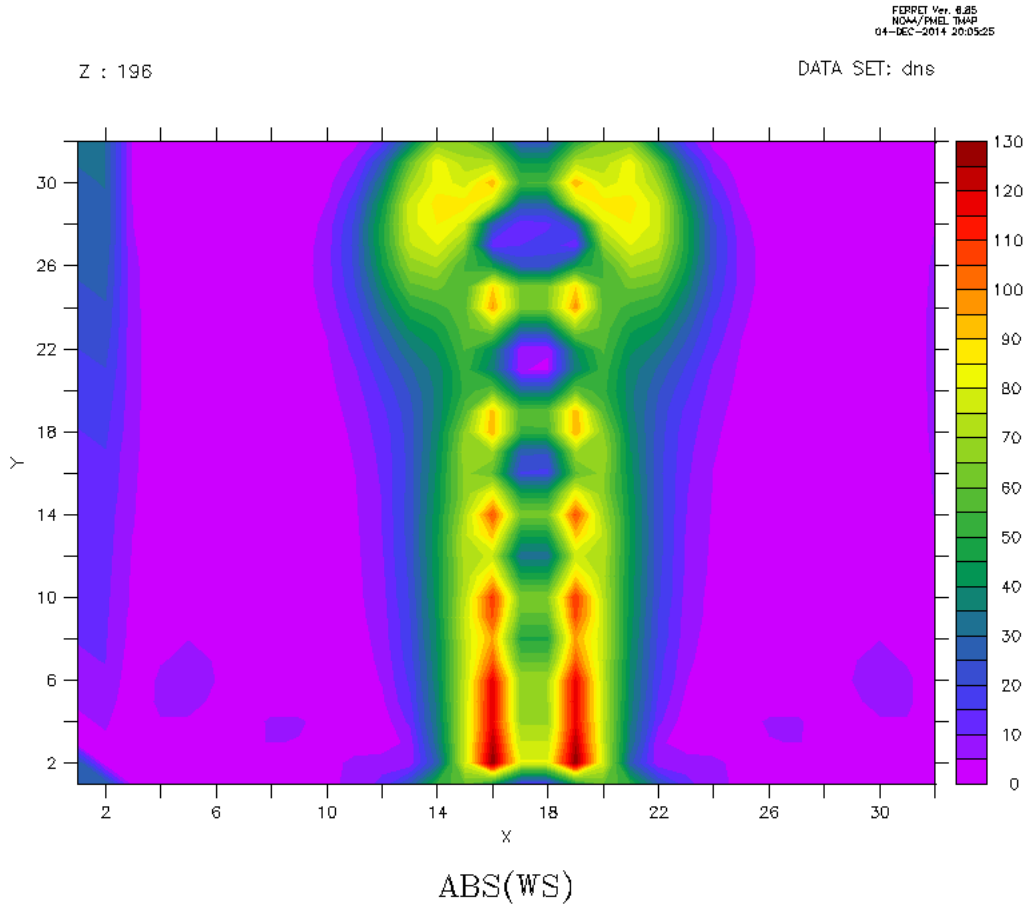

Figure 4.13: FORTRAN 90 Simulation for t = 0.245 seconds(vorticity in the units s^{-1})

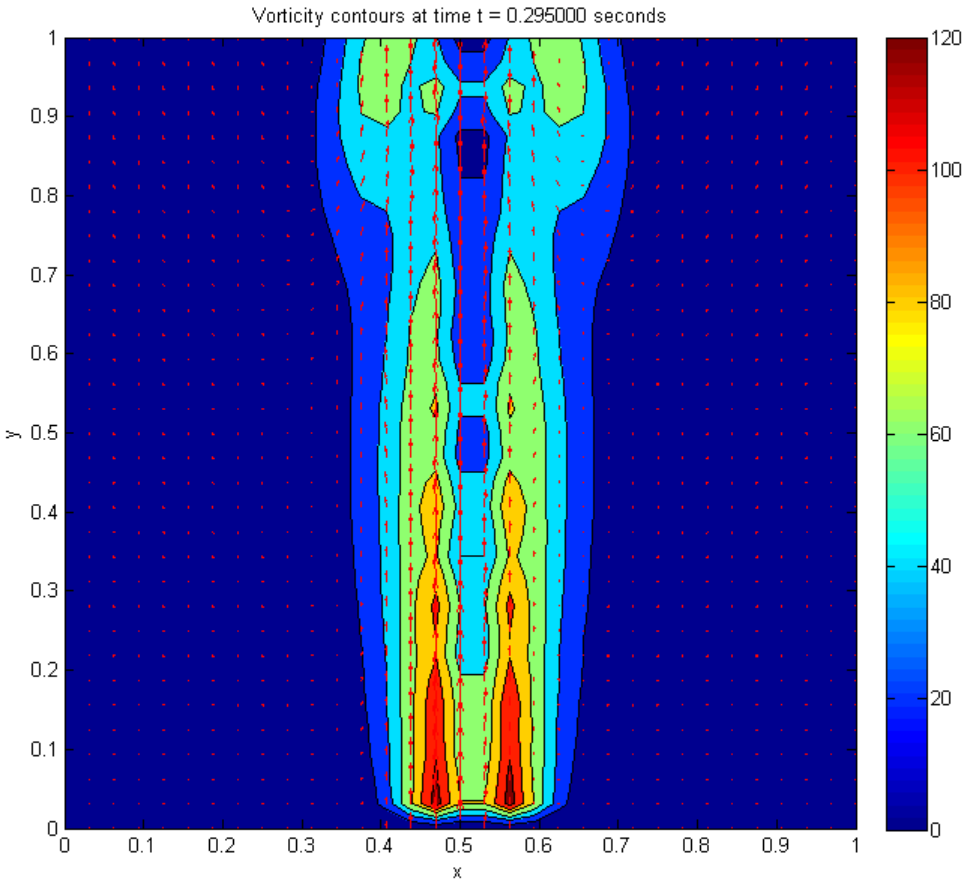

Figure 4.14: MATLAB Simulation for t = 0.295 seconds(vorticity in the units s^{-1})

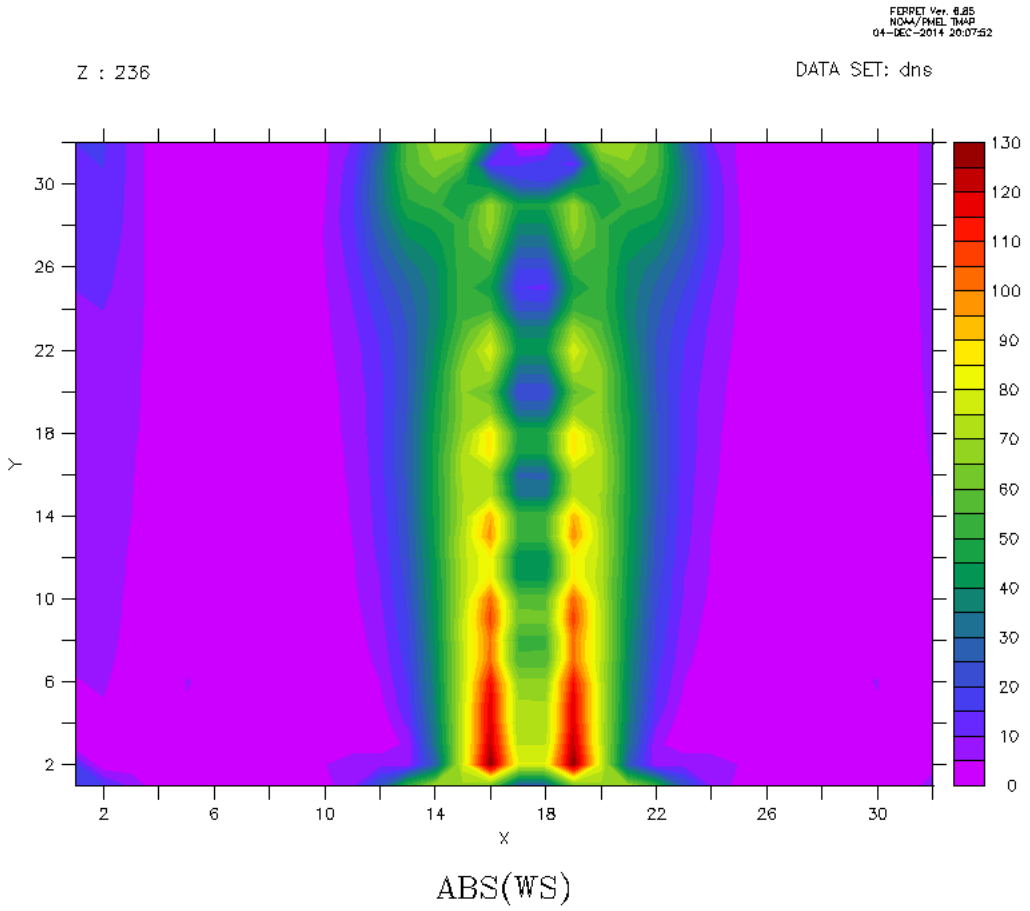

Figure 4.15: FORTRAN 90 Simulation for t = 0.295 seconds(vorticity in the units s^{-1})

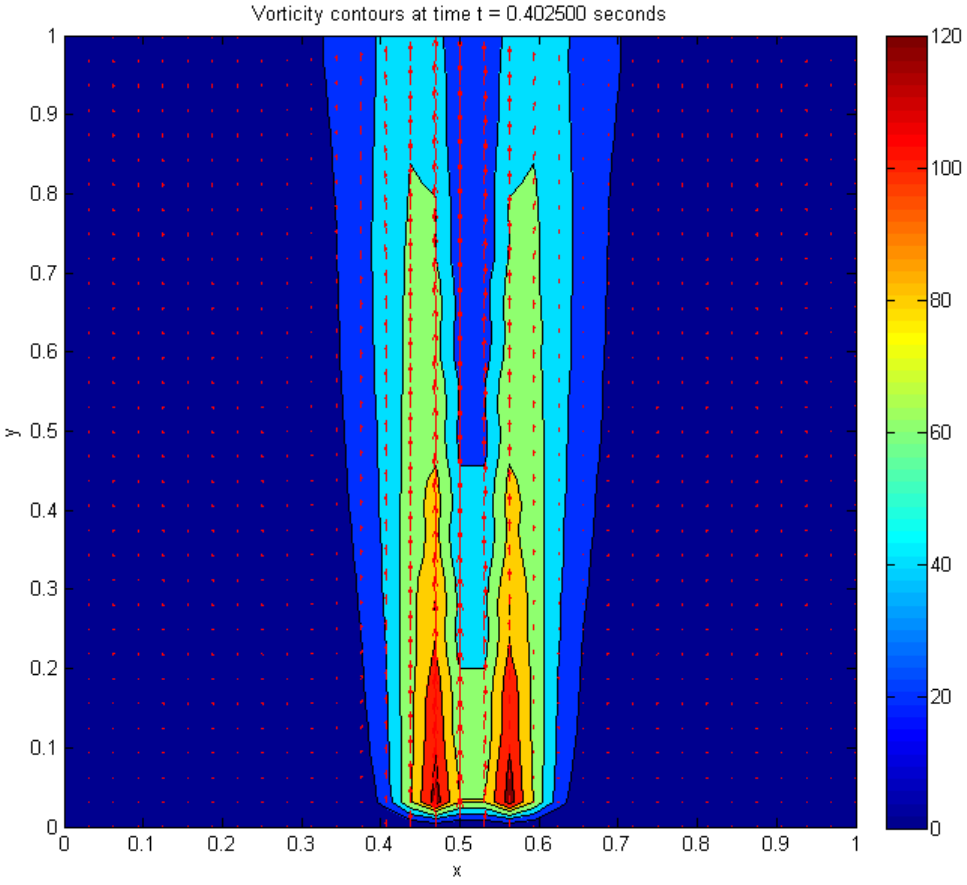

Figure 4.16: MATLAB Simulation for t = 0.4025 seconds(vorticity in the units s^{-1})

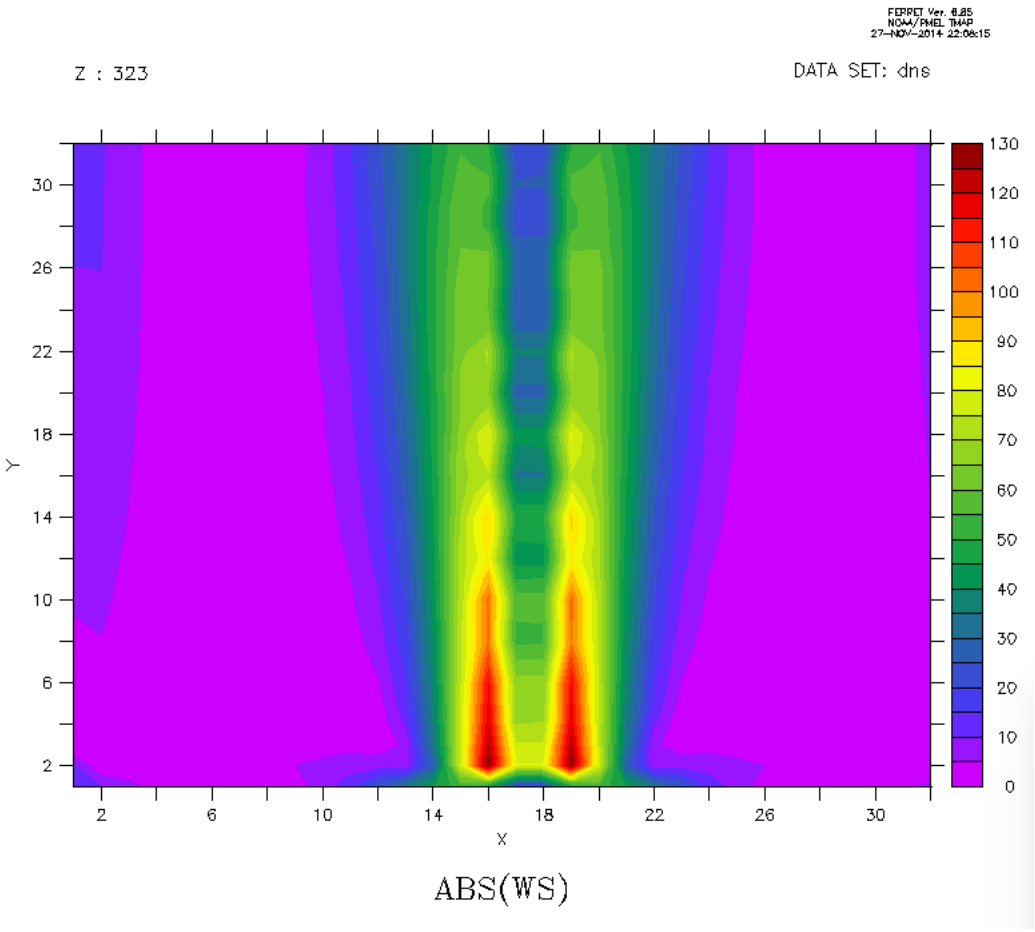

Figure 4.17: FORTRAN 90 Simulation for t = 0.4025 seconds(vorticity in the units s^{-1})

Chapter 5

Conclusions and Future Scope

In the work described in this dissertation, we have developed a 2D incompressible Navier-Stokes solver. This dimensional code has been validated for one test case and in future is expected to study various cloud processes in the atmosphere and the laboratory. Also as the code is run for a constant temperature profile at this stage, it is a prerequisite to include the thermodynamic equation before moving to more complex simulations. The following work needs to be done on this code in future to become a full cloud model:

- The pressure poisson equation solver needs to be made implicit.
- Better schemes for spatial discretization such as hybrid central and upwind finite difference can be used.
- The code needs to include the thermodynamic and scalar(water vapour) equations for being a complete model.

- The grid in the centre of the domain where jet would be entering is to be made fine and on the sides should be fine, so a mapping function for the same would be used.
- The inlet of the jet as carried out in the laboratory is to be implemented as a experimentally proven hyperbolic tan profile of a jet.
- The code would be converted from 2D to 3D
- The coordinate system also needs to be changed from cartesian to cylindrical.
- Finally the code needs to be made parallel for incorporating all the above requirements to be computationally efficient.

Bibliography

Agrawal, A., Boersma, B. & Prasad, A. K. 2004 Direct numerical simulation of a turbulent axisymmetric jet with buoyancy induced acceleration. *Flow, Turbulence and Combustion* .

Akselvoll, K. & Moin, P. 1996 Large-eddy simulation of turbulent confined coannular jets. *Journal of Fluid Mechanics* .

Anand, R., Boersma, B. J. & Agrawal, A. 2009 Detection of turbulent/non-turbulent interface for an axisymmetric turbulent jet: evaluation of known criteria and proposal of a new criterion. *Experiments in Fluids* .

Babu, P. C. & Mahesh, K. 2004 Upstream entrainment in numerical simulations of spatially evolving round jets. *Physics of Fluids* .

Basu, A. J. & Narasimha, R. 1999 Direct numerical simulation of turbulent flows with cloud-like of-source heating. *Journal of Fluid Mechanics* .

Batchelor, G. K. 1954 Heat convection and buoyancy effects in fluids. *Quarterly Journal of the Royal Meteorological Society* **80**.

Beebe, N. H. F. 1992 A summary of fortran. *University of Utah, online resource* .

Bhat, G. S. & Narasimha, R. 1996 A volurnetrically heated jet: large-eddy structure and entrainment characteristics. *Journal of Fluid Mechanics* .

Bhat, G. S., Narasimha, R. & Arakeri, V. 1989 A new method of producing local enhancement of buoyancy in liquid flows. *Experiments in fluids* .

Bodenschatz, E., Malinowski, P., Shaw, R. & Stratmann, F. 2010 Can we understand clouds without turbulence? *Science* **327**.

Boersma, B. J. 1999 Direct numerical simulation of a turbulent reacting jet. *Center for Turbulence Research Annual Research Briefs* .

Boersma, B. J., Brethouwer, G. & Nieuwstadt, F. T. M. 1998 A numerical investigation on the effect of the inflow conditions on the self-similar region of a round jet. *Physics of Fluids* .

Charlson, R. J., Schwartz, S., Hales, J., Cess, R., Coakley, Hansen, J. & Hofmann, D. 1992 Climate forcing by anthropogenic aerosols. *Science* **255**.

Chorin, A. 1992 *A Mathematical Introduction to Fluid Mechanics*. Springer-Verlag Publishing Company.

Claeyssen, J. R., Platte, R. B. & Bravo, E. 1999 Simulation in primitive variables of incompressible flow with pressure-neumann condition. *International Journal for Numerical Methods in Fluids* .

Diwan, S. S., Narasimha, R., Bhat, G. S. & Sreenivas, K. R. 2011 Laboratory studies of anomalous entrainment in cumulus cloud flows. *In Journal of Physics: Conference Series* **318**.

Erturk, E. 2009 Laboratory studies of anomalous entrainment in cumulus cloud flows. *International journal for numerical methods in fluids* **60**.

Griebel, M., Dornseifer, T. & Neunhoeffer, T. 1998 *Numerical Simulation in Fluid Dynamics: A Practical Introduction*. SIAM Monographs on Mathematical Modeling and Computation.

Houze, R. A. 1994 *Cloud Dynamics*. Elsevier Science.

Jonker, H. J., Heus, T. & Sullivan, P. P. 2008 A refined view of vertical mass transport by cumulus convection. *Geophysical Research Letters* **35**.

Konduri, A. 2009 Direct numerical simulations of starting-plume cloud-flows. *Masters thesis, Jawaharlal Nehru Centre for Advanced Scientific Research, Bangalore* .

Moin, P. & Mahesh, K. 1998 Direct numerical simulation a tool in turbulence research. *Annual Review of Fluid Mechanics* **30**.

Narasimha, R. & Bhat, G. 2008 Recent experimental and computational studies related to the fluid dynamics of clouds. *IUTAM Symposium on Computational Physics and New Perspectives in Turbulence* .

Ogura, Y. 1963 The evolution of a moist convective element in a shallow, conditionally unstable atmosphere: A numerical calculation. *Journal of Atmospheric Sciences* **20**.

Prasanth, P. 2013 Direct numerical simulation of transient cumulus cloud flow. *Masters thesis, Jawaharlal Nehru Centre for Advanced Scientific Research, Bangalore* .

Sani, R. L., Shen, J., Pironneau, O. & Gresho, P. M. 2006 Pressure boundary condition for the time-dependent incompressible navier-stokes equations. *International Journal for Numerical Methods in Fluids* .

Stephens, L., Tsay, C., Stackhouse, W. & Flatau, J. 1990 The relevance of the microphysical and radiative properties of cirrus clouds to climate and climatic feedback. *Journal of the atmospheric sciences* **47**.

Stocker, T. H., Qin, Dahe, Plattner, Kasper, G., Tignor, Melinda, Simon, A. K., Boschung, Judith, Nauels, Alexander, Xia, Yu, Vincent, B., Midgley, M, P. *et al.* 2013 Climate change 2013. the physical science basis. working group i contribution to the

fifth assessment report of the intergovernmental panel on climate change-abstract for decision-makers. *Tech. Rep.*. Intergovernmental Panel on Climate Change-IPCC(Switzerland).

Stommel, H. 1947 Entrainment of air into a cumulus cloud. *Journal of Meteorology* .

Tryggvason, G., Scardovelli, R. & Zaleski, S. 2011 *Direct Numerical Simulations of Gas-Liquid Multiphase Flows*. Cambridge University Press.

Turner, J. S. 1973 *Buoyancy effects in fluids*. Cambridge University Press.

Venkatakrishnan, L., Bhat, G., Prabhu, A. & Narasimha, R. 1998 Visualization studies of cloud-like flows. *Current Science* .

Versteeg, H. K. & Malalasekara, W. 1995 *An introduction to Computational Fluid Dynamics*. Prentice Hall.

Warner, J. 1955 he water content of cumuliform cloud. *Tellus* .

White, F. M. 1991 *Viscous fluid flow*. McGraw-Hill New York.