Numerical Study of Particle Behaviour in Von Karman Flow

By

Harshit Bhatia – 500027428

M.Tech. Computational Fluid Dynamics

August 2015

Project Advisor: Dr. Eric Climent
PSC Group, IMFT, France

Internal Advisor: Dr. Ugur Guven
Department of Aerospace Engineering,
UPES, India
Acknowledgement

I would like to say sincere gratitude and heartfelt regards to Prof. Eric Climent for his enduring and kind support for my project and discussions related to it.

I am also grateful to Dr. Ugur Guven who provided me with this opportunity which changed my outlook and thought process on research working methodology. Endless support and constant motivation provided by Dr. Guven is reason that I now look future as a never ending pursuit to achieve something better for myself and humankind.

My sincere thanks Dr. Rajesh Yadav, Dr. Gurunadh Velidi, of Department of Aerospace Engineering, UPES, who taught me insights of CFD that helped in developing a numerical understanding that was quite useful for this project.

I would also like to thank my Parents to trust on my decisions throughout life and allowed me to follow my dreams.

Special Thanks to Annaïg Pedrono, IMFT who took time out for giving basic instructions on code Jadim and took care of all the doubts encountered while use of this code.

Other people who supported me in France are Subhendu Rawat, Guiquan WANG.

August 2015

Harshit Bhatia

MTech Computational Fluid Dynamics
# Table of Contents

Acknowledgement ........................................................................................................... i

1. Introduction .................................................................................................................. 1
   1.1 Literature Review ...................................................................................................... 3

2. Problem Statement ....................................................................................................... 10

3. Methodology ............................................................................................................... 11
   3.1 Geometry Description ............................................................................................... 11
   3.2 Mathematical Equations ........................................................................................... 14
   3.3 Code Jadim ................................................................................................................. 16
   3.4 Mesh Selection and Staggered grid ........................................................................... 18
   3.5 Simulation Modelling ............................................................................................... 20

4. Numerical Study on Von Karman Flow ..................................................................... 22
   4.1 Setting up the cases ................................................................................................. 22
   4.2 Critical Reynolds number $Re_C$ and Grid Independence ......................................... 26
   4.3 Post Processing and Results Observations ............................................................... 27

5. Force Coupling Method ............................................................................................. 39
   5.1 Introduction ............................................................................................................... 39
   5.2 Case setup ................................................................................................................. 39
   5.3 Post processing and Results Observations ............................................................... 43

6. Conclusion .................................................................................................................... 47

Appendices ......................................................................................................................... 52
# Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Re</td>
<td>Reynolds Number</td>
</tr>
<tr>
<td>FCM</td>
<td>Force Coupling Method</td>
</tr>
<tr>
<td>PDE</td>
<td>Partial Differential Equation</td>
</tr>
<tr>
<td>NS Equation</td>
<td>Navier Stokes Equation</td>
</tr>
<tr>
<td>LES</td>
<td>Large Eddy Simulation</td>
</tr>
<tr>
<td>RANS</td>
<td>Reynolds Averaged Navier Stokes</td>
</tr>
<tr>
<td>HIT</td>
<td>Homogenous Isentropic Turbulence</td>
</tr>
</tbody>
</table>
List Of Table

Table 1 Values of Viscosity to be used in Jadim................................................................. 23
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Poloidal and Toroidal components of vortex ring.</td>
<td>9</td>
</tr>
<tr>
<td>2</td>
<td>Setup of domain used for simulation, displaying disc position.[4]</td>
<td>12</td>
</tr>
<tr>
<td>3</td>
<td>Dimensions of Domain used for simulation, with sign convention in Cartesian</td>
<td>12</td>
</tr>
<tr>
<td>4</td>
<td>Variables allocation in 2D staggered grid</td>
<td>19</td>
</tr>
<tr>
<td>5</td>
<td>U Velocity - time step</td>
<td>27</td>
</tr>
<tr>
<td>6</td>
<td>V Velocity - Time Step</td>
<td>27</td>
</tr>
<tr>
<td>7</td>
<td>W Velocity - Time step</td>
<td>28</td>
</tr>
<tr>
<td>8</td>
<td>Paraview displaying Iso-contour surface (in white) and Domain colour by velocity magnitude over domain. Opacity 0.2 to display internal Iso-surface</td>
<td>29</td>
</tr>
<tr>
<td>9</td>
<td>Iso-surface of Q dimensionless at 0.0005, Re 100(Left) and Re900(Right)</td>
<td>29</td>
</tr>
<tr>
<td>10</td>
<td>Dissipation Non Dimensional on Log Scale</td>
<td>30</td>
</tr>
<tr>
<td>11</td>
<td>Variation of flux in X direction over grid</td>
<td>31</td>
</tr>
<tr>
<td>12</td>
<td>Poloidal Value in Y direction</td>
<td>32</td>
</tr>
<tr>
<td>13</td>
<td>Toroidal Value in Y direction</td>
<td>32</td>
</tr>
<tr>
<td>14</td>
<td>Poloidal Value in Z direction</td>
<td>33</td>
</tr>
<tr>
<td>15</td>
<td>Toroidal Value in Z direction</td>
<td>33</td>
</tr>
<tr>
<td>16</td>
<td>Ratio of Poloidal/Toroidal value in y</td>
<td>34</td>
</tr>
<tr>
<td>17</td>
<td>Ratio of Poloidal/Toroidal value in Z</td>
<td>34</td>
</tr>
<tr>
<td>18</td>
<td>Location of lines used for making velocity profile</td>
<td>35</td>
</tr>
<tr>
<td>19</td>
<td>Profile plot of mean at top tip of disc</td>
<td>35</td>
</tr>
<tr>
<td>20</td>
<td>Profile plot at bottom tip of disc</td>
<td>36</td>
</tr>
<tr>
<td>21</td>
<td>Clip at X=0.1 and X= -0.1, contour of Velocity magnitude. Re 10</td>
<td>36</td>
</tr>
<tr>
<td>22</td>
<td>Clip at X=0.1 and X= -0.1, contour of Velocity magnitude. Re 800</td>
<td>37</td>
</tr>
<tr>
<td>23</td>
<td>Mid Plane contour of Velocity Magnitude Re 10 at y and z plane</td>
<td>37</td>
</tr>
<tr>
<td>24</td>
<td>Figure 23 Mid Plane contour of Velocity Magnitude. Re 800 at y and z plane</td>
<td>37</td>
</tr>
<tr>
<td>25</td>
<td>Normal velocity in Mid Y plane</td>
<td>38</td>
</tr>
<tr>
<td>26</td>
<td>MSD of particle data collected over various initial positions.</td>
<td>44</td>
</tr>
<tr>
<td>27</td>
<td>Correlation of U velocity</td>
<td>44</td>
</tr>
<tr>
<td>28</td>
<td>Correlation of V velocity</td>
<td>45</td>
</tr>
<tr>
<td>29</td>
<td>Correlation of W velocity</td>
<td>45</td>
</tr>
<tr>
<td>30</td>
<td>PDF of particle</td>
<td>46</td>
</tr>
</tbody>
</table>
Numerical Study of Particle Behaviour in Von Karman Flow

Abstract

Von Karman flow is the flow consisting of counter rotating flow in the domain which can result in interesting observations. One such observation is formation of Vortex ring in the enclosed box that is direct result of interaction of fluid layers in rotating flow. Using this vortex ring and its data, numerical code was developed to find which component of vortex ring dominates the flow, Poloidal or Toroidal. Post processing scripts were created to visualize the flow and its statistics to understand the physics of this kind of flow in the laminar and transition region. Detecting particle behaviour in Von Karman flow was also done with the Force Coupling Model which uses lagrangian method of particle tracking. Data of particle movement was processed to extract more statistical data like Mean Square Displacement, Probability distribution Function and Correlation. This study also confirms the ability of In house code Jadim, to successfully implement FCM in Von Karman flow with some limitations.
1. Introduction

Behaviour of particle in any type of flow affects our daily life and it is so close to human life that sometimes we even fail to appreciate the beauty of natural phenomena occurring around us in every moment. Talk about making tea/coffee by mixing sugar in morning to washing our dirty clothes to pull dust particles off our favourite brand. Talk about industrial system, we can see use of particles in mixture of complex chemical mechanisms, settling of crystal structures, and extensive sedimentation work in petroleum industry, they all involve the particulate flows. Talk from medical industry we see that nasal sprays and micro particles transported in blood are such example. Deposit of sand on river bed, formation of rain clouds, shaping of planets in accretion discs and dispersion of pollen grain in air are few natural occurring phenomenon.

In all these cases we see that interaction of fluids with particles (solid) is really interesting enough to increase the level of understanding and taking this field to study to understand the working of physics behind all these situations. Individual behaviour and response of particle can vary by large factor depending on various parameters like speed which interaction takes place, chemical properties of both particles and fluid, involvement of turbulence factor, size of particle.

What is Von Karman flow?

Any flow involving the swirling factor in it is known commonly as Von Karman Flow. In this problem to be specific; the flow that is being generated by the action of exact counter rotation in the bounded closed domain is called Von Karman Flow. This kind of flow can be very interesting to study and this is of particular interest in enclosed domain as this is the best economical method available to generate homogenous turbulence in laboratory condition for better understanding of turbulence behaviour and fundamental physics related to it. By completing the small scale laboratory model this can be scaled up in man industrial application where counter rotation of disc creates flow turbulence so that energy can be better utilized.
How does Particle behave in general flow?

When particle is very small in size than the smaller vortices or if it has same density as of working fluid the particle will follow the path of fluid purely as a visual tracer. If the density varies, the flow around particle will be laminar locally and so the particle velocity can be calculated and formulating simple model to resolve forces will follow the Stokes law of drag and an additional mass term. However this is possible only for particle falling in specific size range (generally more than 10 times.) as for bigger particles the resolution of forces acting on particle is still a tough call. To solve the governing equation of velocity there are many available model out of which Force Coupling Method is preferred (details in Chapter 5)

What is FCM?

FCM stands for Force Coupling method that is a numerical method developed for solving the flows involving suspension; by using simultaneous solution of fluid governing equations and Lagrangian method of tracking particle. It provides an easy solution to particle equation by using the disturbance magnitude caused by presence of particle in flow.

Roadmap

This numerical study is carried out to simulate experimental work in computational manner. Study takes into consideration details used in laboratory conditions to observe the flow and effects of particles on it. This include the homogenous symmetric flow condition which are hard to find closely in reality where things are moving at much larger scale, constantly changing physical parameters and are full of heterogeneous effects.

This project is completed in two parts, first being the simulation of Von Karman flow, using DNS method for numerical simulation, observing the effect of increasing Reynolds number on it, grid independence study to find the optimum grid required to capture the physics properly and then taking data out of simulation to study the effects and details on factors like energy dissipation, poloidal and toroidal ratio etc.; Second being introduction of particle in the flow to observe its behaviour, coupling effect, path line and extracts various statistics on it using FCM.

Work Flow in this project was
• Literature Review
• Training on code Jadim- running test cases.
• Starting simulation of Von Karman flow.
• Looking for convergence- Using Gnu plot.
• Preparing post processing on Paraview.
• Running cases for grid convergence + preparing of Matlab scripting of Q and Lambda criteria.
• Preparing code for velocity profile in non-dimensional form, at line joining top to top and bottom to bottom of opposite disc.
• Learning about power and energy dissipation + preparing Matlab script for Power and Energy.
• Validation of Energy Dissipation code by using standard flow of Taylor Green Vortex.
• Learning about Poloidal and Toroidal components of flow + Matlab for Poloidal and Toroidal.
• FCM module debugging
• FCM module test cases simulation
• Finding gravity acting on particle, by non-dimensional method.
• Preparing scripts for Mean Square Displacement, Probability distribution function.
• Understanding the concept of Correlation and plotting it by making Fortran Script of Correlation.
• Compiling all the data and making the final plots.

1.1 Literature Review

Although the Von Karman vortex generation can be observed in nature, (atmospheric observation around mountains and sea water movement around islands) for small scale study of this, we need a scaled down model to observe and simulate flow behaviour in the closed domain.

Primarily done in magneto-hydrodynamics field of study but it is also observed and worked upon by other fields of applications involving fluid dynamics also; the
turbulence is often generated by using two counter rotating discs that may or may not enclose within the domain a cylinder or similar object[1]. To get the complete numerical solution of vortex formation dynamics, over the sphere or cylinder involved in flow requires the complete set of force equations involved in flow. This is one of the fields of investigation that which force can act on it. Debate on how the velocity flow field is divided into the smaller internal regimes is old and was first taken up by Batchelor[2] and Stewartson[3]. Work carried out on two infinite discs, in counter rotating configuration, tangential velocity was found to be symmetric about middle plane and was divided in five zones; boundary layer on disc, transitional shear layer at plane and two rotational cores near transitional layer. On other hand Stewartson presented three zones at higher Reynolds number based on distance 'h' between two discs; one boundary layer on each disc and zone of uniform radial flow in domain with zero tangential velocity in between.

Suggested by study of Nathanael et al[4] at larger sized particles, effect of mean structure of flow is great influencer. As the particle diameter gets closer to integral length of the flow, particle behaviour becomes more uniform in Von Karman flow, with great chances that the particle moves towards the poloidal neutral axis. Factors suggested to affect this were Re, particle density and surface roughness of particle.

To better understand the stability analysis of flow and how it get affected by the aspect ratio of geometric domain is described by C. Nore[5] who carried series of numerical simulations to obtain a three dimensional stability. Analysis performed on aspect ratio between 0.5 to 3, of cylindrical domain showed non axis symmetric modes are dominant in the flow and they are stationary. Study also suggested that the critical azimuthal wavenumber is function of this aspect ratio. Patterns of disturbances analysed show that thresholds of axisymmetric instabilities is always higher than present in non-axisymmetric. Study investigates into dynamics dominated by Kelvin Helmholtz instability and gives rise to stationary vortices that are radial along static periphery. Similar study was carried out by same author with fixed aspect ratio of 1:2 as height to radius ratio[7] talks in detail about loss of stability by increase of Re and goes on to investigate the effect to various azimuthal wave number.

When dealing with the energy injection by the rotating disc the study by O. Cadot[6] was looked into. Study deals with the observation of mean energy injected and
dissipated in order to conclude that the smooth discs are less efficient in injecting energy into the system. This is supported by two experiments carried out by the author one by counter rotating stirrers and other by counter rotating cylinders setting up the Coutte Taylor Flow. Observation of author was that with smooth stirrers the dissipation in fluid domain was much weaker than the Re dependent dissipation in boundary and with inertial discs the dissipation in fluid becomes dominant and is according to Kolmogorov behaviour.

**Literature for Vortex Identification**

To identify the vertex in any of flow is a complex process because the size of vertices differs largely in size and structure. If we look in structures at boundary layer level and in middle of rotating flow of the disc the structures in boundary layer that are so small that they are hard to identify separately. The vortex structures in the middle of rotating flow are very large in size so that dominate over all other smaller size vertices. To solve this problem a method of vortex identification was suggested based on point-wise method of analysis of velocity gradient tensor as suggested in study of vortex identification schemes by Pinaki Chakraborty [11].

This study suggests various methods of vortex identification with new proposed requirements of vortex core. Mathematical relationship were explored and studied to get relationship between different size and criteria of flow kinematics. Study tested this method in canonical turbulent flow and the vertical structures found match the analytical results remarkable. Since then this method has been utilized to get better visualisation.

The problem with characteristics of vertical structures in turbulence in matter of great interest and concern as till date there is no proper methodology to accurately map the structure. With the study of velocity gradient. We can identify velocity in form of filaments and sheets that have been already identified. Current area of interest includes vortex worms in isotropic zone [12] , vortex braids in sheer layers [13] and hair pin vertices in wall turbulence. In ideal fluids the existence of visible difference between irrotational fluid and rotational fluid is the basic adopted definition of vortex filament [14] however in real fluids this visible boundary is not sharp and is more of diffused kind. This diffusion due to viscosity along with strained region of fluid increases the complexity of vortex identification. According
to thus study the local method of vortex identification uses a function that is solved at each point of the domain and then classified each point whether it is inside or outside the vortex according to the predefined criteria. Most of these criteria are dynamic in nature that use velocity gradient tensor. Most popular local criteria that are in use are $Q$[15], $\lambda_2$[16], $\delta$[17], $\lambda$-ci[18]. These methods are successful in detection as well as in rendering of vortex filaments in much better manner (sharply visible), they don’t discriminate vortexes on basis its structures. Study was kept limited to filamentary type vortexes but it is worth to mention that non filament type vortexes also form a very important part of fluid kinematics.

Homogeneous turbulence is the turbulence that has same structure propagating quantitatively in all direction of flow fields. This term also suggest that velocity fluctuations are random in nature but average characteristics are not dependent are independent regarding to flow field condition and position (unaffected by access translation).

Isotropic turbulence is the turbulence that has no directional preference and is completely chaos in manner. Any isotropic turbulence contains the features that have directional preference and the mean velocity of this turbulence has a gradient. The fluctuations does not depend on axis selected and are also not affected by rotation and reflection of axis.

**Q criterion** this criteria originally given by hunt et al 1998, uses a method that detects a region in flow as vortexes by looking for positive second in variant of $\Delta V$ matrix. if there is presence of positive value [$Q>0$] this suggests the region is a vortex. The second important in this is that the pressure in same region is required to be lower than the available ambient pressure.

Mathematically in an incompressible flow $Q$ is the parameter showing more rotation rate than strain rate. Point suggested here is that $Q>0$ is not the assurance of existence of minimum pressure in the same region but it is assumed to be.

**$\lambda_2$ criterion**

This criterion is based on theory that the local minimum pressure does not identify vortices which are dominated by unsteady and viscous flow condition.

This method uses vortex core as a connective region with support of two different Eigen values of positive pressure measurement. If Eigen values of tensor $S^2+\Omega^2$ are
classified as \( \text{lambda1} > \text{lambda2} > \text{lambda3} \) then \( \text{lambda2} \) should be less than zero at each point of vortex core.

Main difference between Q criteria and \( \text{lambda2} \) criteria is that Q criteria looks for abundance of rotational rate over the strain rate by magnitude in all direction but \( \text{lambda2} \) searches for same extra available magnitude difference in an specific plane.

**Delta criterion**

This criteria uses critical point theory given by Chong et al which uses core of vortex as the region where gradient of velocity has complex Eigen values. If \( \text{delta} > 0 \) then gradient of velocity has complex Eigen values and is part of vortex region. it is also observed that Q criteria is more closely restricted then delta criteria.

**Swirling strength criterion**

This criterion is based on delta criterion used with imaginary section of complex conjugate of Eigen value of gradient of velocity that is used to identify velocities. The swirling strength given by \( \text{lambda-ci} \) is actually measurement of local swirl rate inside the vortex. How much the vortex will stretch or compress is determined by this criteria.

Based on new mathematical developments new vortex identification techniques have been generated but the use of these classic criteria are most commonly used. As these are straight forward in mathematical modelling and easy to control in coding environment.

**Correlation**

Talking in statistical sense correlation is simply a relation between two variable or two different data cells. It tells how much the quantity is dependent on other data cells. This can also be used in understanding the physical status of parent data cell and children data cell.

Use correlation plot is that it gives a prediction of how the quantity will be in collaboration. In scientific sense, it provides the dependence of one quantity over other. In this study of one von Carmon flow and particle study is that we find the behaviour of particle in laminar region and how it is affected by the steady nature of flow. The ideal prediction of correlation what is it should fall as a line horizontal line close to zero value. When the correlation is plotted in the two different data sets, it
gives us the relation but when we want to find correlation with the same quantity or same data set, we need to create a relation that treats the wave differently at different time steps. This will give us the better idea of how the quantity will behave in the future. Observations from the graphs are that correlation of velocity in x direction is trending towards zero in a smooth manner. But the lack of data points available the graph does not reaches toward exact zero. While correlation of velocity in y direction and velocity in z direction is not a smooth line but a repetitive pattern that fluctuates around the zero value. This fluctuation was surprising in nature but can be interpreted as,

1) Turbulence is not fully developed yet so there is no straight line observed.

2) The flow is in periodic motion so the value keeps fluctuating suggesting that velocity domination is not a smooth but is a periodic in nature.

3) Correlation is a product of same velocity function at different time steps taken as mean into account. While normalizing this function it is divided by root mean square velocity. This plot is in auto correlation function versus integral time step. In this plot the peak of the graph, the maximum kinetic energy is hold by the flow but as the time increases, the kinetic energy is dissipated and based on the scale of turbulence it should reach as close to zero as possible. In ideal conditions the turbulent flow field if is homogeneous and isotropic (assumption), correlation tensor is the resultant of distance are between two points and not there location in flow field. But unfortunately it is very difficult to observe this type of turbulence in real life. Otherwise it was very simple to get the velocity reading anywhere inside the turbulent flow then substract the laminar component and we get correlation value exactly.

The physical interpretation of this is that the kinetic energy of flow starts from one (hundred per cent energy input) and slowly reaches to zero value that suggest the input energy has been fully dissipated. The general observation of the plot obtained is that the flow in x direction is steadily moving towards turbulence but is not fully turbulent.

There are two regions of vortex ring-
Poloidal- A region around imaginary axis where the fluid rotates in a circular type shape. In fluid dynamics these are created by secondary flow. Shown by red in figure 1.

Torroidal- is region around imaginary axis around which the fluid creates enclosed loop by spinning of fluid. It is combination of many poloidal rings. These are due to azimuthal flow. Shown by blue in figure 1.

Taking ratio in this study gives us opportunity of knowing which kind of flow dominates our simulation. Theoretically from past experience the ratio of mean poloidal value over mean torroidal value should be more than 1 in Von Karman flow.

Figure 1 Poloidal and Toroidal components of vortex ring. [E3]
2. Problem Statement

Particles in general behave differently based on various continues phase properties (working fluid) and dispersed phase properties (particle phase). Predicting particle behaviour in simple conditions with smooth flow taking place is quite simple with available mathematical model, like behaviour of particle in standing water or simple pipe flow. It becomes uncertain when flow is complex, containing vortices and circulation.

To target this tough call numerical study is carried out with DNS. This is most accurate numerical model available which solves the NS equation without any assumption or in other words there is no loss of information while solving the equations. Other commonly used methods, which are computationally complex, and involve assumptions in its mathematical model which results in loss of important information about fluid properties and turbulence, while solving the equations numerically. But these models are faster in iterations (as less complex equations are solved) and so are much cheaper economically. For example RANS and LES are not accurate enough because information related to time is less accurate. Turbulence intensity calculated by DNS will be far more accurate than obtained by use of LES or RANS. Solution to this problem gives more insight about how to target the mathematical model within computational constraints to obtain reasonable particle behaviour. Scope right now is limited only to theoretical study of single particle, in near future, when computational power increases enough to handle DNS at larger scale this problem can be scaled up to industrial level where particle laden flow makes a big part. So it becomes essential to study and understand the basic behaviour of particle in Von-Karman Flow and use our understanding to make industrial processes dealing in the field more efficient by providing with better processing model design with valuable inputs.

Objective-

Objective of this study is to numerically find the dynamic statistics of particle introduced inside Von Karman flow generated by counter rotating discs in enclosed domain, with the Lagrangian particle tracking method. For this FCM model of code CFD Jadim was used.
3. Methodology

3.1 Geometry Description

To generate Von Karman flow we require a closed box with two rotating discs. Setup of the simulation is similar to various experiments carried out till date containing an enclosed domain (generally square in shape), with two disc arrangements on opposite sides to form a cylinder type rotation in between.

The most general type of configuration is shown in figure below, and same was used to carry out the simulations. The setup has one central domain of cube of size 20 cm that is fitted with two discs of 9.5 cm radius on opposite end of X direction (in YZ plane).

Flow domain is internal flow with the turbulence in flow being injected by rotation of discs, one at a time or both discs simultaneously. The size of domain is selected to be of convention 2R (twice the radius of disc) to be large enough to handle the flow circulation without obstructing proper circulation and small enough to handle the experimental and numerical grid restriction to reasonably not going beyond limit of giving in precious time resource.

The domain of cube was chosen to be made of perfect flat wall, to experimentally make the proper visualization for accurate measurements by limiting optical distortion. Numerically choosing a flat cross-section means less scope of error in grid node spacing and less error is induced by discretization of equations.

This setup accounts for best suitable configuration for study of particle behaviour and dynamics in fully developed turbulent condition with the approximation of homogeneous turbulence at the centre part of domain. One of the differences in the simulations conducted numerically was that in experiments most study has been done by using impeller blades so as to impart the momentum to flow, but in this current study the impeller blades are not modelled in the geometry. This can be further talked as the two different cases of study on basis of how the stirring is done:
1) Viscos Stirring: Flat smooth discs with no blades involved
2) Inertial Stirring: Discs with blades to impart the momentum.

Figure 2 Setup of domain used for simulation, displaying disc position.[4]

Figure 3 Dimensions of Domain used for simulation, with sign convention in Cartesian coordinates.[4]

In viscous stirring the flow is driven by the shear force of the flow. When the disc starts rotating, the layer of fluid starts experiencing the effect of shear force. This force is then passed on to next layer of fluid and thus results in movement of fluid without imparting any significant inertia to fluid; but driving force only as viscous effects. When the disc containing blades is used to impart force to fluid, the blade section passes on inertia to fluid section close to blade. This inertia further starts the
movement of fluid to next available volume section. Thus it is so called that if disc
does not have blades the stirring is called as viscous stirring and is blades are present
the n it is called as inertial stirring.

This kind of setup is used for its advantage of ready availability of HIT
(Homogenous Isotropic Turbulence) in the space confinements with limit of small
scale quantities, where a particle can be inserted to study its flow specific behaviour
in homogenous turbulence or fully developed turbulence.

By homogenous turbulence we mean that the properties of turbulent flow are same
everywhere in the domain irrespective of translation. For this kind of flow with
proper time averaging and in periodic domain the Urms, Vrms, Wrms have same
value everywhere (at different location of measurement) but not same with each
other (value of Urms may be different from Vrms) . To get this the measurement has
to be taken away from wall and Reynolds number should be significantly high. In
rotational flow the significance of isentropic turbulence is that the velocity
fluctuations is not affected by rotation any more. These velocity fluctuations are
found to be of similar pattern everywhere. Point to be noted here is that time
averaging is necessary for these observations as instantaneous turbulence can never
be homogenous.

Another point is that by simply changing the rotational behaviour of disc (like one
disc rotating and other at rest or both discs rotating at different speed), different flow
phenomena can be produced over large no of Reynolds Number. If we go into
detailed large scale study, condition becomes anisotropic overall. In this domain if
the continuous supply of flow is not provided by disc rotation the turbulence
developed will stay for some time and then decay but if aim is to target the
homogenous turbulence, continuous supply of energy by disc rotation should be
there.

In this setup the shear layer control is the key to obtain the desired flow conditions;
choice of square domain fixes a specific plane in middle of vessel that is shear layer,
which is obtained if both disc operate on same frequency conditions. This case is not
possible to be developed in the circular or any other shape vessel as the shear layer
transition and shear layer reversal will happen throughout the flow. To compare this
with experiments this setup will act as the effective cylinder with radius of 9.5 cm and length 20 cm.

### 3.2 Mathematical Equations

Governing equation for this problem is Navier Stokes equations for incompressible fluid; where density is treated as a constant variable not evolving over time, so the density variable terms are left out with final equation looking like:-

**Mass Conservation**  \[ \nabla \cdot \mathbf{v} = 0 \]  \text{ Eq. 3.1}  

**Momentum Conservation**  \[ \frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla)\mathbf{v} = -\frac{\partial P}{\partial \rho_f} + \nabla \cdot \mathbf{F} \]  \text{ Eq. 3.2}  

Here in above equations we have $\theta$ as kinematic viscosity, $P$ as pressure field, $\mathbf{v}$ as velocity field $\rho_f$ as density of working fluid. $\mathbf{F}$ contains all additional forces acting on fluid volume example gravity.

In these equations the term accounting for density variation is left out as the case is incompressible with constant density value of fluid. Working fluid for simulation of single phase is taken to be constant 1000 kg/m$^3$ but for multiphase simulations it has been modified as per non dimensional analysis.

The term $(\mathbf{v} \cdot \nabla)\mathbf{v}$ is responsible for inertial properties of fluid and it can be evolved in more particle manner that will be used extensively for our project. This is used in to find Reynolds Number (Re) that is a simple ratio of inertial forces over viscous forces acting on fluid.

In mathematical form it is

\[ Re = \frac{F_{\text{inertial}}}{F_{\text{viscous}}} = \frac{\Omega R^2 \rho}{\mu} \]  \text{ Eq. 3.3}  

Here $\Omega$ represents the angular velocity that is taken to be constant for both of the discs, $R$ stands for Radius of rotating disc, and $\mu$ as dynamic viscosity. The expression is interchangeable with $\theta = \mu/\rho$.

In this project we deal with range of Reynolds number initially to investigate Critical Reynolds number, and for the final stage of particle inclusion we choose $Re$, below the Critical Reynolds number so as to be sure that the simulation is carried out in
laminar condition. This selection of laminar working condition support our cause of study to observe and extract behaviour properties of the fluid, otherwise in turbulent condition tracking properties specific to particle becomes much more difficult as we have to include different mathematical approach to extract useful data.

**FCM equations:**

Mathematical model of FCM can be used with same incompressible condition described for equations 3.1 and 3.2.

Consider in Eq. 3.2, F (force) as function of space and time, f(x,t).

In Standard Multipole expansion [8] method f(x,t) is written as series as

\[
f_i(x, t) = \sum_{n=1}^{N} \left[ \left( F_{i}^{n} + F_{ij}^{n} \frac{\partial}{\partial x_j} + F_{ijk}^{n} \frac{\partial^2}{\partial x_j \partial x_k} + \cdots \right) \delta(x - Y^n) \right]
\]

Eq. 3.4

Here \( \delta(x - Y^n) \) is Dirac delta function for particle location as function of \( Y^n(t) \). \( F_i \), \( F_{ij} \), \( F_{ijk} \) stands for monopole, dipole and quadrupole moments of particle respectively and decides the accuracy of model solution. To obtain solution of this equation, simple Taylor series expansion will be enough as the equation is linear in nature. Using this method does have a problem of existing singularity that exists due to delta function in expansion.

Alternate to this was suggested [9] as replacement of Dirac delta function with Gaussian Function expressed as;

\[
\Delta(x) = (2\pi\sigma^2)^{-3/2} e^{(-|x|^2/2\sigma^2)}
\]

Eq. 3.5

This method is according to original work of Maxey and Patel (2001). Final equation that we get of force is a combination of force monopole and force dipole expressed as

\[
f_i(x, t) = \sum_{n=1}^{N} \left( F_{i}^{(n)} \Delta(x - Y^{(n)}(t)) + G_{ij}^{(n)} \frac{\partial}{\partial x_j} \Delta'(x - Y^{(n)}(t)) \right)
\]

Eq. 3.6

This equation is then solved in both temporal and spatial ways to overall view of momentum distribution in space. This is the force felt by fluid that is freely moving in hydrodynamic forces acting on it. This is then fed into Eq 3.2 and integrated over volume to get the final solution.

FCM uses steepest descent iterative approach [3.6] as it enforce zero strain rate on particle, and FCM also ignores the inertia of particle, thus limiting the choice of
particle density to be as close to fluid density. In this study relative density of particle is selected to be 0.9 and 1.14.

3.3 Code Jadim

3.3.1 Introduction

Jadim is in-house code that was developed by team interface group at IMFT. This code allows describing various physical parameters of fluid flow in best possible manner. This code gives special attention to physical parameters that are related to multi-phase flow. Mathematically this tool solves Navier-Stokes equation in three dimensions for incompressible fluid.

For Von Karman flow this code specially used second order centrally differential scheme for space discretisation and second order Runge Kutta method in temporal discretization part. For pressure solution projection of Poisson’s equation is solved using Fourier solver for single phase flow and iterative ‘sparse’ scheme for multi-phase flow. This code is further capable of solving heat transfer equations volume of fluid method and immersed boundary method but is not related to my project and hence was not used.

3.3.2 Installation and Executable:

To begin with compilation of Jadim, directory is composed by make file comment with all the source files in SRC directory. With every change in source file remake of executable command is necessary to incorporate the changes made in solver part. For example this process was repeated for multiple times when tracking of particle in the flow was being used. Problem faced during project was that, the particle was not moving under the force exerted by fluid. To resolve this champv.f90 file was modified to check, rotation of disc, its speed and force parameter on particle (used as suggested FCM module) and adjusted accordingly and new executable was generated. With this we get executable Jadimles.x but to classify it separately, we use a different name Jadim_paul.x to use it for rotating flow.

3.3.3 Running executable:

When we want to use this executable for running the project. ‘./jadim_paul.x’ is used to run it on single processor. To run it on parallel processors: mpirun –np 4
.jadim_paul.x was used. The resultant file that is most important is ‘rbin-* .out’ which contains important information of velocity, pressure and temperature etc. of flow in binary format. This file is used to restart the project or to post process it to get visual output.

To obtain visual idea of what is being solved and to check if the solution convergence has been reached modification in code was done to make following values being stored in the output.dat file. These values were delta of x velocity (dum), delta of y velocity (dvm), delta of z velocity (dwm), delta of pressure (dp) and delta of x,y,z momentum (duu, dvv, dzz).

Location of probe was inputted as close to centre as possible but it can never be exactly centre numerically. Geometrically centre of the domain lies at (0,0,0) but numerically putting vale as zero will make simulation erroneous. So instead the value inputted in the code was (ni/2,nj/2,nk/2) in champv.f90 field of Jadim source and executable was re made by make command. This value may be in fraction as no of grid node points ni,nj,nk can be any number, so the precaution was taken to convert this numerical value to its nearest whole number for probe point selection.

3.3.4 Converting output to post-process

To post process the result Jadim output file can be converted into input file of tecplot, Paraview, Ensight, Matlab. For this following commands were used according to requirement.

For tech plot:

les2tec VONK range rbin-initial interval rbin-final

Here les2tec calls for file that converts rbin as binary to tecplot readable file; VONK is the name of case file in my project. rbin-initial is the file number that we want to start our visualization and the rbin-final is the last file we want our visualization effect to be; interval is the time step interval selected to save the cases.

Example- les2tec VONK range 0 100 2000; this command reads every rbin file with corresponding number 0 till rbin2000 and reads at every 100 step. Finally we get equal number of techplot readable file with same time interval and step.

For paraview and Matlab same type of commands are executed:
les2par VONK range 1000 100 20000 (reads and converts file form step 1000 to 20000)

les2asc VONK range 1000 100 20000 (reads and converts file form step 1000 to 20000 for matlab).

les2par VONK one 1500 (converts only one file corresponding to step 1500).

For sample output.dat, .asc file and paraview file please refer digital CD provided with this thesis.

3.4 Mesh Selection and Staggered grid

When we talk about using numerical techniques to get solution; first step we deal with is, discretization of partial differentials in the NS equations to make computational simulation work. These discretised sets of equations are then solved over nodes created by various mathematical schemes over the area of interest in computational domain, where these equations can be further solved by use of any of three methods namely finite difference, finite volume and finite element. Discretization is method to convert partial differential equations into computationally efficient simpler algebraic equations by the means of mathematical schemes. Any area of concern is divided into smaller areas in a certain connective pattern that gives spatial information to solver code. This pattern is called grid. The components of grid are nodes (crossing points of gridlines), edge (line joining nodes), face (enclosed by edges), and volume (by stitching grid faces.) and If this pattern is in a logical equal spacing manner then it is called a ‘structured grid’. If any algorithm is followed to randomly arrange elements into domain then it’s called ‘unstructured grid’. Advantage a structured grid holds over other one is; chances of numerical error induced in solution due to node spacing are comparatively less. So here in case setup structured rid was selected; also to be mentioned is that a simple square geometry is best captured by structured mesh method used in Staggered storage system. This results in minimum chances of induced error in the solution due to geometric constrains or mathematical discretization.

While working on storing values of fluid information in computational domain everything would be good approximation for the solver step, if it was on the grid nodes, but unfortunately this is not the way things work. Pressure decoupling is a
classic example as how the non-uniform pressure field is treated as a uniform pressure field as the central discretization always returns a zero value upon implementation [E1].

Solution comes out to be quite useful as accuracy increases with using Staggered storage solution of variables, where the values are stored not at one single place but at different location in the grid. For example in two dimensional allocation of grid, the values of momentum variable may be stored on the sides of line joining the nodes and density might be stored at different location such as centre of the grid. In three dimensions there is much more allocation to be done separately, density is allocated at centre of cube (volume centric) along with internal energy. Velocity components and momentum information is stored as face centric values, which responds to shift by half grid point along the same direction (as matter of uniformity it has to in positive direction), other relative values are places as in edge-centred.

With this staggered grid the differential operator is used in such a way that it returns the value shifted half way up or down the grid point with nearby grid point instead of alternate grid point so decoupling cannot take place. With use of staggered grid computational time can increase (various interpolation of values takes place, with iteration) but it works way better than decoupling scheme.

Use of staggered grid also ensures that the pressure value does not oscillate due to strong coupling between the variables.

Figure 4 Variables allocation in 2D staggered grid
3.5 Simulation Modelling

Numerical modelling of the Von Karman flow has been done by using in-house code of Jadim. Simulations were carried out in Finite Volume Method along with Direct Numerical Simulation model or turbulence capture.

DNS (Direct Numerical Simulation) is a method of numerical simulation in the purest mathematical form, without any assumption of turbulence modelling. The equation is solved for whole range of eddy ranges and is one of the accurate models for turbulence resolution. DNS depends on mesh resolution to take care of spatial scale capturing right from Kolmogorov micro scale eddies to integral scale. This results in requirement of larger grid size with more nodes to capture effects properly, which increases the memory requirements to simulate and store iterative values. Drawback of using this model is it requires large computational time which indirectly results in higher computational costs. Studies suggest the computational cost involved is directly proportional to cube of Reynolds number.

Accuracy of DNS depends on the time step Δt that has to be smaller for more accurate solution. This can be explained by the explicit scheme for time integration, which suggests that in order to get accurate solution time step has to be so small that the movement of small portion of fluid is only in fraction of grid cell size (Δx).

Factor to decide this is given as: \( \frac{\Delta t}{\Delta x} \leq C_{\text{max}} \). Here C is non-dimensional number commonly known as Courant number, and this condition is called as CFL condition. Value of \( C_{\text{max}} \) depends on lot of factors affecting simulation. In this study times step was calculated by taking \( C_{\text{max}} \) as 1. For details refer Section 4.1, VONK.para.

DNS is most important and powerful tool for fundamental studies of fluids and turbulence but its cost constrains restricts its use in industrial applications. This results in developing other cost effective models by keeping DNS as bench mark that is then used to compare results and accuracy with it.

Turbulence modelling can be broadly classified on basis of closure treatment of NS equation to give information of turbulence.

1) Models that use Boussinesq method of approximation forming base of eddy viscosity model.
2) Models that focus on Reynolds Stresses and solve for them directly.

3) Models not using time averaging method- LES and DNS fall under this category.

Order of accuracy of numerical scheme used to solve NS eq is Runge Kutta 3rd order in Jadim. It is used for temporal discretisation. For solving the spatial discretization Central difference scheme of 2nd order is used.

**Initial and boundary conditions:** In this study the flow starts from rest and so there is no initial flow conditions applied to simulation, however the description of boundary condition is certainly more important one. In the simulations carried out, the wall condition is enforced on all six faces. Walls are defined with no slip boundary condition. This means that wall will have viscous effects dominating on it. No special treatment is given to the boundary layer as we deal with DNS in our study.

This is an enclosed domain study within solid walls so there is no inlet and no outlet condition imposed on any side. There is symmetry in the domain but study is carried out in full dimensions and symmetric plane is not created or used in approach.

For CPU processing algorithm for parallel computing; ‘mpi ’method was used and nothing new was modified in this. For parallelisation of simulation, effect of increasing number of processor on iterative timing was noted. When we increase number of processors for faster computation of cases, speed increases with increase in number of processors up to certain limit. After that the actual time taken to share and pass the information between the processors is much higher then the solver time so overall time required speeding up the calculations increases. With this observation maximum four processor were selected in this study as after using more than four processors iterative timing actually slows down due to additional time required to share data information with each other.
4. Numerical Study on Von Karman Flow

4.1 Setting up the cases

Based on available geometry dimension given the numerical domain was developed by using ‘mesher’ tool of in-house tool Jadim. Through it following files with case name ‘VONK’ was generated .para, .bord, .geom, .limi. These files and other case files have the following use/carry following information:

**VONK.geom** hold coordinates of nodes along with 2 boundary nodes (for 50grid the file contains 52 points information). File stores information in format of Cartesian coordinates, its node count and mask matrix that stores value in i,j as 1 if point described lies inside domain and -1 if it lies outside domain or is used/or holds in obstacle.

**VONK.limi** holds type of boundary details like periodic, wall etc along with no of nodes and the boundedness of geometry. Boundaries are named as 1,2,3,4 corresponding to north, west, south and east respectively. Detailed information about deformable wall surface (like in bubble) or singular axis (like in axisymmetric mode) and VOF required values are entered.

**VONK.bord** has information about geometric faces and types.

Alternatively if we already have a previously generated .mesh file we can edit the desired node values and boundary types and execute generate mesh command to generate new mesh based on given parameters in .mesh file. This .mesh file contains information coordinates system (Cartesian); name of case (vonk); number of cells inside domain (25, 50, 74); bounded domain size (-0.1 to 0.1 in all direction); number of outlets and obstacles if any and info about module of LES or VOF.

After collecting this basic file of geometry and mesh, we use them in our working directory. Following files are necessary along with above generated mesh and geometry files to successfully setup case and execute the simulation part.

**VONK.phys** carries information of physical parameters of working fluid such as density, fluid viscosity and thermal diffusivity. In the same file we can add pressure gradients, and gravity value if required. In our simulations pressure gradients due to external factors and gravity value on working fluid is none as we deal with an enclosed system problem where fluid domain is enclosed by walls on all sides and
no atmospheric effect is modelled in simulation. Thermal diffusivity is also not
needs as we are not dealing with thermodynamic study. The parameters of concern
are density and viscosity. We restrict our self to variation of fluid viscosity and keep
the working fluid density as constant. Density value of working fluid is taken to be
1000 kg/m³ close to that of normal water.

Parameter of viscosity is a careful choice and becomes more important in this study
as the variation of xmu (used in code for dynamic stress viscosity) is the tuned to
select the Reynolds Number of our Cases. As from the equation 3.3; Omega is fixed,
Radius of disc is fixed so; only parameter left for tuning our Re is this. By shuffling
the parameters we can get same equation in form of $\mu = \frac{\Omega R^2 \rho}{Re}$. So now if suppose
we need value of viscosity in order to attain Re 100 we just put related vales on right
hand side and we know all of them so, the final value is entered in code. Simplifying
things further since in present study all other parameters remain same final Eq can be
used as $\mu = \frac{9.025}{Re}$.

To cross verify value every time before proceeding and doing manual calculation use
of this value a table of values was created in Microsoft Excel Sheet. Snapshot of
table below.

**Table 1 Values of Viscosity to be used in Jadim**

<table>
<thead>
<tr>
<th>For Re</th>
<th>use Mu as</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>9.025E-01</td>
</tr>
<tr>
<td>100</td>
<td>9.025E-02</td>
</tr>
<tr>
<td>200</td>
<td>4.513E-02</td>
</tr>
<tr>
<td>250</td>
<td>3.610E-02</td>
</tr>
<tr>
<td>300</td>
<td>3.008E-02</td>
</tr>
<tr>
<td>350</td>
<td>2.579E-02</td>
</tr>
<tr>
<td>400</td>
<td>2.256E-02</td>
</tr>
<tr>
<td>450</td>
<td>2.006E-02</td>
</tr>
<tr>
<td>500</td>
<td>1.805E-02</td>
</tr>
<tr>
<td>550</td>
<td>1.641E-02</td>
</tr>
<tr>
<td>600</td>
<td>1.504E-02</td>
</tr>
<tr>
<td>650</td>
<td>1.388E-02</td>
</tr>
<tr>
<td>700</td>
<td>1.289E-02</td>
</tr>
<tr>
<td>750</td>
<td>1.203E-02</td>
</tr>
<tr>
<td>800</td>
<td>1.128E-02</td>
</tr>
<tr>
<td>850</td>
<td>1.062E-02</td>
</tr>
<tr>
<td>900</td>
<td>1.003E-02</td>
</tr>
</tbody>
</table>
VONK.init contains information of the initial conditions if needed can be set from here. Values that can be entered are initial velocity field, initial swirl, rotational velocity, temperature, shear and variable density factor. This file is untouched and all values remain to zero as we control rotational speed of disc from executable jadim_paul.x and zero value means default settings.

VONK.para is one of the most carefully attended file in setting up the case. It has more detailed numerical parameters to be entered for simulation to take place. It is a long file so I discuss here only parameters that we used for our case. Parameters are set for a 3d numerical setup (i3d), with Maximum number of iterations entered as ‘maxit’. Code also has the specification of axisymmetric case and curvilinear model in geometry section. If we setup new case parameter ‘ijob’ is set to ‘0’ and switched to ‘1’ if it is restarted with some modification or after inserting a particle in multiphase study. Next set of parameters, deal with printing and reporting intervals of output on screen and on file. Short reporting interval will increase the simulation time as computer calculates and reports the value repeatedly, so to speed up the simulation (as this study being 3d, requires large computational time) it is suggested to use a significantly longer reporting interval like writing every 10 iteration or 100 iteration.

This study has been done in pressure solver scheme of Petsc. Petsc is an open source library that is developed to solve the momentum balance equation by using Poisson Equation.

Petsc scheme is mathematical model developed by group of scientists in the Argonne National Laboratory, USA, where it was developed as set of structures and routines that are helpful in solving the large partial differential equations. This scheme allows scalability of solver to massive parallel computing setup basically in MPI format. It is widely used in scientific community to solve matrix of sparse type (where maximum number of elements are zero) and set of PDE’s. Petsc scheme is developed in C, C++ environment but can be used extensively in open source and windows environment. Going deep into study of this Scheme and implementation is beyond
the scope of this thesis however general observation is that this scheme made our solution slower when compared to Fourier solver in pressure equation.

We are dealing with independent study of turbulence so the auto stop criteria of solver is used very minimum value that will not be reached so that the simulation can go on till the time step we want (manual control of simulation). ‘epsip’ deals with the minimum value when reached the simulation stops. Next section deals with the time step variables that are minimum value and maximum value of time step, ‘dtmin’, ‘dtmax’ is used for this purpose. Dtmax value has been calculated manually and carefully selected within error limits to make simulation as smooth as possible without losing accuracy of solution. ‘imp’ parameter in this file goes for solution treatment, selection is 1 for implicit treatment (as required in our case) or if required selecting 0 means tuning to explicit treatment. ‘ista’ is used to define the flow type if has to do nothing(as in this case study), or if it has moving boundary, moving spheres, moving object or any other dynamic change in flow condition. For supporting any type supporting input file is required to be created in order to make this work. ‘iles’ gives us switch in code if we require LES for simulation, if it is used additional parameters are required. We are not used in this report.

Other parameters can be used if required but not used are thermal module, variable density module, statistical model, flow perturbation model, coupling, obstacle, body immersed, rheology, chemistry module.

With entering the required values the executable jadim_paul is used to start the simulation. For faster computation and quick results the parallel computing with ‘mpi’ scheme was used to launch cases in the local parallel processing unit. The system used for study had 12 processors with following specifications. 12 processors were best utilized by observing the best optimized time taken per iteration to be 4 processors per case. This allowed 3 cases to launch simultaneously.

Study was started by using an appropriate grid of 50 nodes and initial set of introduction cases were launched with Re 10, 100 and 200. Basic study of how parameters affect the simulation and flow behaviour was studied based on initial introduction case and material provided by IMFT Jadim support team.
4.2 Critical Reynolds number $Re_C$ and Grid Independence

To start with our single phase numerical study of Von-Karman flow the cases were set at different Reynolds number to find and compare the critical Reynolds no. ($Re_C$) with experimental results; of similar problem. Simulations started from Re 10 were carried out initially with difference of 100 Re and for fine tuning the difference of 50 was used near Re 600. Initial $Re_C$ was observed in range 850-900. This initial study of $Re_C$ was carried out on grid of $50^3$ nodes.

Similar pattern was repeated with grid of $25^3$, $74^3$ and $100^3$.

For any numerical investigation it is necessary to carry simulations in the grid independent environment to ensure that our numerical values are independent of no of nodes used in mesh of domain. This becomes more of importance in non-linear problems, where convergence may not always means a good result. This is important aspect of numerical study as it is directly linked with mathematical value changes as the governing equations are discretised over nodes and solved by iterative process. Mathematical error can be induced at this stage in the solution. To minimize, chance of error getting into our numerical solution Grid Independence/Dependence study is carried out. It is affected by selection of mesh structure (Structured or Unstructured mesh used) and number of nodes/cell count.

Numerically speaking, more nodes mean more accuracy but up to certain limit. Reason for this is ‘Truncation Error’. Truncation error is the difference in the value of exact equation and discretised equation. This error can be controlled by altering the number of grid points used while solving the problem. After this limit, its effect gets stopped and increasing more number of nodes will not affect our solution. If nodes are too less, solution will be computationally fast but physics will not be captured properly; whereas too many nodes means better physics capture but more nodes to iterate upon and thus more computational time.

The objective of this study is to obtain best possible solution with minimum requirement of nodes and thus fast solution timing. The outcome of this study was that Grid of $50^3$ $50^3$ was good enough to be used for further study of Von Karman behaviour and $Re_C$ was close to 900.
4.3 Post Processing and Results Observations

To observe the flow behaviour and the circulation in the flow use of advance visualization tools were used to get the best possible picture of flow. For this combination of software and scripting was used.

4.3.1 To observe flow behaviour during Critical Re study

This was visualised by GNUPLOT. Or this the output file that contains the details of flow data is read and by using following pattern, visualization of flow velocities and other parameters was done.

Following plots from GNU show Velocity profile from Case of Re 800 and 50^3 grid nodes.

![U Velocity - time step](image)

**Figure 5** U Velocity - time step

![V Velocity - Time Step](image)

**Figure 6** V Velocity - Time Step
The output file generated by Jadim in single phase study contains data of, Time Step, X velocity (U), Y velocity (V), Z velocity (W), Pressure (P), X velocity gradient, Y velocity gradient, Z velocity gradient in column 1 to 8 respectively.

This output is used to determine up to which Re the flow remains stable.

4.3.2 Post Processing using Paraview

When Profile read for Paraview was done the file contained information of u,v,w velocity individually but for more complete visualization we require Velocity magnitude, so Calculation of velocity magnitude was created as function in Paraview to do post processing.
Figure 8 Paraview displaying Iso-contour surface (in white) and Domain colour by velocity magnitude over domain. Opacity 0.2 to display internal Iso-surface

Another set of post processing used in Paraview was the visualization of the Q and lambda structures of the dissipation.

Figure 9 Iso-surface of Q dimensionless at 0.0005, Re 100(Left) and Re900(Right)

4.3.3 Post Processing using Matlab scripts

Q criteria- Q criteria and Lambda 2 are common method of visualizing turbulence structure as the vorticity values obtained by other means does not conveys complete information about the same. Initially developed for visualizing turbulence better for large variation in the local and global variation of vorticity this helps in giving better picture over all. Based on Matlab script created to model Q values in dimensionless
method we get the visual picture of how turbulence breaks down as the Re increases (Figure 8).

**Dissipation**- Energy dissipated in internal flow gives the direct relation of how power induced by the rotating disc breaks down in effect of viscous effects of fluid. This power imparted by disc is converted to kinetic energy of fluid which results in increase of internal energy. As the Reynolds number increases in our study Dissipation value (dimensionless) decreases smoothly. This study is limited to Re1000 but it can reduce further down as suggested by the slope of graph in Figure 9. Observation by this graph is that as the Reynolds number decreases the dissipation scale of energy keeps decreasing and may be in further increments of Re(not carried out in this study), is will be constant with no further reduction. Code is appended as A3.

![Dissipation Non Dimensional on Log Scale](image)

**Figure 10** Dissipation Non Dimensional on Log Scale

**Flux**- Flux of any physical quantity is the flow rate per unit of area. In this study the flux generated by the rotating discs is calculated. Since the discs are in Y,Z plane the effect of flux will be observed in the X direction. So, mathematically flux modelling according to our problem is given as

\[
Flux = \iint_U u_x(y, z) \, dy \, dz
\]

Eq. 4.1

Details of code created to find this value is appended in the Appendix A4. Point of observation in using this formula to code is that the disc is having components with
sign negative as well as positive, so to avoid the effect to cancelled out and observe the total flux the modulus of velocity was used.

![Figure 11 Variation of flux in X direction over grid.](image)

Finding poloidal component in the flow was done by modelling following equations in Matlab.

\[
\langle P_y \rangle = \int_{-0.1}^{0.1} \int_{-0.1}^{0.1} \sqrt{u_x^2 + u_y^2} \, dx \, dz
\]

Equation 4.4

\[
\langle P_z \rangle = \int_{-0.1}^{0.1} \int_{-0.1}^{0.1} \sqrt{u_x^2 + u_z^2} \, dx \, dy
\]

Equation 4.3

Code can be found in appendix A4.

To find toroidal value in Y direction,

\[
\langle T_y \rangle = \int_{-0.1}^{0.1} |U_y| \, dx \, dz
\]

Equation 4.4

And in Z direction,

\[
\langle T_z \rangle = \int_{-0.1}^{0.1} |U_z| \, dx \, dy
\]

Equation 4.5
Figure 12 Poloidal Value in Y direction

Figure 13 Toroidal Value in Y direction
Figure 14 Poloidal Value in Z direction

Figure 15 Toroidal Value in Z direction
Figure 16 Ratio of Poloidal/Toroidal value in $y$

Figure 17 Ratio of Poloidal/Toroidal value in $Z$
Profile Plot The velocity plot of azimuthal velocity in non-dimensional form, on line from top of disc and bottom of disc.

Figure 18 Location of lines used for making velocity profile

Figure 19 Profile plot of mean at top tip of disc.
Figure 20 Profile plot at bottom tip of disc.

Figure 21 Clip at X=0.1 and X=-0.1, contour of Velocity magnitude. Re 10
Figure 22 Clip at X=0.1 and X=-0.1, contour of Velocity magnitude. Re 800

Figure 23 Mid Plane contour of Velocity Magnitude Re 10 at y and z plane

Figure 24 Figure 23 Mid Plane contour of Velocity Magnitude. Re 800 at y and z plane
From figure 11, 12, 13, 14, we observe the variation of toroidal and poloidal components in the vortex ring. We see that as the Reynolds no increases the shape of components shifts from smooth parabolic shape to a shape that gives us location of vortex ring. One peak that is visible on Re10 goes in to shape change of much flatter profile, and in to two smaller peaks and thus suggesting the presence of ring. This is observed in both direction Y and Z. Another important observation is that the profile change also follows a smooth pattern till it reaches Re650 after which the shape starts to show the difference in the pattern.

Figure 15,16 show the ratio which suggests that the flow in vortex ring is dominated by the secondary flow or the poloidal component is more stronger than the toroidal component of velocity.

Figure 18,19 shows the velocity profile in azimuthal direction that gives us the effect of boundary layer that is observed till the first two nodes (or upto 1 cm.) from the wall. The variation of the profile suggests that as the Re increases the slope decreases and the profile becomes straighter and less deviated at centre.

Figure 20,21 are the contour of velocity magnitude. From observation we see that as the Re increases the velocity iso lines shifts according to disc rotation and glyph arrows gets closer to the position of vortex ring. At low Reynolds no the velocity distribution is almost in symmetry and as Re is increased the symmetry is lost and elongation or any other type of distortion in the shape is observed. Similar kind of observation can be observed from figure 22,23,24 and 25 which represent the contour of velocity magnitude and its components.
5. Force Coupling Method

5.1 Introduction

FCM is the method developed to track particle by Lagrangian approach that uses the combined effects of Stokeslet effect where particle moved due to its buoyancy effect and Stresslet where particle moves due to its inertial properties. Approach of FCM is to utilize the fluctuations in the velocity field that arise due to presence of particle. These fluctuations occur as the particle is a solid rigid body and no matter how small it is, it will affect the flow at local level.

Finding Gravity value of the simulation through Non dimensional analysis: This was one of most careful area where manual calculations by non-dimensional analysis were carried out to find the gravity value to be fed into the solver part. Gravity value is different from real world as we carry out numerical analysis at different rotational speed of disc, where disc rotation numerically limited to one radian per second. Find Details in appendix.

5.2 Case setup

In addition to Vonk.para parameters added earlier in single phase flow, this additional parameters need to be set for two-phase flow.

--- Lagrangian module --------

itraj - lagrangian particles tracking if 1 (add a *.traj input file)

irtraj - 1 if lagrangian particles tracking restart

ijobstat - 1 if stat on particles restart

We opt for ‘itraj’ as 1 to start the tracking of particle in Lagrangian method and this will further take data inputs from two more files namely VONK.traj and rbin.traj.

‘Rbin.traj’ is the binary file that stores the information about previous location of particle. If earlier simulation was not having particle this file will not exist.

File of main concern in two phase simulation is VONK.traj that contains parameters of particle physics for user to control.
Through this file following parameters can be controlled-

1) Type of coupling between two phases- it can either be 2 way coupling, 2 way coupling with FCM monopole or 2 way coupling with FCM monopole and dipole combined.

2) Number of particles included in simulation – 1 in this study.

3) Radius of Particle- 12 mm

4) Density of particle- 900 kg/m$^3$ and 1140 kg/m$^3$

5) Gravity value acting on particle.

6) Interval count at which the particle trajectory data will be stored – 1, for every iteration in our study.

7) Type of particle- Solid or gas bubble particle (to select Schiller and Nauman’s law) and Mei’s law respectively. Solid in our case

8) Initial position of particle- homogenous repartition, spatial random, initial position read from external file or as in our case particle dropped at specific point.

9) Position at which particle is dropped x,y,z coordinate.

10) Brownian Noise- none in our case.

11) Elliptical particle.

After carefully adding the parameters in these files, last saved data file is taken from single phase flow so that we do not have to start flow development all again and we are sure that particle is dropped in already developed flow.

Ideally this particle movement should go on for infinite time or until the maximum number of iteration is reached but this was not happening in this study. Particle moves and as it hits the wall or disc it did not bounce off and use to stop at the spot of impact. Reason behind this is that the code Jadim does not have rebound force interaction modelled in it. Limited time availability was the reason this was not corrected as the statistics required can be successfully extracted by removing data set close to particle’s impact on wall.
5.2.1 Non Dimensional analysis of gravity value

In this numerical study the physical parameters to be used are different than that in laboratory environment of real life. This is the reason that we cannot use the gravity acting on particle to be real life gravity on earth of 9.8 m/s². The main difference that we observe is of rotational velocity of disc and viscosity of fluid; in experiments and in simulations both are different.

For this non dimensional analysis was done based on Reynolds Number and force acting on particle based on Stokes law, thus the final gravity value used was to be around $2 \times 10^{-2}$.

In this analysis please note that sub-script ‘e’ stands for experimental parameter and sub-script ‘s’ stands for simulation parameter.

Reynolds Number comparison:

$$Re_e = Re_s$$  \hspace{1cm} Eq 5.1

$$\frac{\alpha_e r_e^2}{\phi_e} = \frac{\alpha_s r_s^2}{\phi_s}$$  \hspace{1cm} Eq 5.2

Since both radius are of same length (in simulation radius is chosen to be of same 19 cm in length); so we get, $r_e^2 = r_s^2$ and hence cancelling out these terms we get;

$$\frac{\alpha_e}{\phi_e} = \frac{\alpha_s}{\phi_s}$$  \hspace{1cm} Eq 5.3

We can write the kinematic viscosity $\nu$ as $\mu/\rho$,

$$\frac{\alpha_e}{(\mu/\rho)_e} = \frac{\alpha_s}{(\mu/\rho)_s}$$  \hspace{1cm} Eq 5.4

$$\mu_s = \frac{\alpha_s}{\alpha_e} \times \frac{\rho_s}{\rho_e} * \mu_e$$  \hspace{1cm} Eq 5.5

Now from Stokes law we try to combine the effect of Force between simulation and experiments

According to Stokes law-

$$(\rho_p - \rho_f) \times \left(\frac{4}{3} \pi a^3 g\right) = 6\pi \mu \alpha V_\infty$$  \hspace{1cm} Eq 5.6

Rearranging the terms, we get,
\[ V_\infty = \frac{2}{9} \left( \frac{\rho_p - \rho_f}{\mu} \right) a^2 g \] Eq. 5.7

Now the velocity in non-dimensional form should be equal for experiments and simulation.

\[ \frac{V_{\infty e}}{\Omega_e r_e^2} = \frac{V_{\infty s}}{\Omega_s r_s^2} \] Eq. 5.8

Since radius in simulation and experiments are same, and \( \Omega \) in simulation is 1,

\[ V_{\infty e} = V_{\infty s} * \Omega_e \] Eq. 5.9

\[ \frac{2}{9} \left( \frac{\rho_p - \rho_f}{\mu} \right) a^2 g_e = \frac{2}{9} \left( \frac{\rho_p - \rho_f}{\mu} \right) a_s^2 g_s * \Omega_e \] Eq. 5.10

Particle radius \( a \) in experiment and simulation is kept same, and converting viscosity

\[ \frac{\rho_p - \rho_f}{\mu} g_e = \frac{\rho_p - \rho_f}{\mu} g_s * \Omega_e \] Eq. 5.11

Converting density variables in relative density format and rearranging the terms,

\[ (\rho^* - 1)_e g_e \vartheta_s = (\rho^* - 1)_s g_s \vartheta_e \Omega_e \] Eq. 5.12

\[ g_s = \frac{\vartheta_s}{\Omega_s \vartheta_e} g_e \] Eq. 5.13

Now \( \vartheta_s = \Omega_s \frac{r_s^2}{R_{es}} \) and \( \vartheta_e = \Omega_e \frac{r_e^2}{R_{es}} \) also \( \Omega_s = 1 \)

So,

\[ g_s = \frac{g_e}{\Omega_e^2} \] Eq. 5.14

Taking values as reference from Nathanael et al[4] where the similar study was carried out in laboratory experiments,

Selecting closes Re value available from study, we get Reynolds Number taken as 701 and corresponding viscosity as \( 320 \times 10^{-6} \); solving Reynolds number formula gives the value of angular velocity as \( \Omega_e = 22.432 \).

Substituting this value in equation above we get

\[ g_s = \frac{0.81}{(22.432)^2} \] Eq. 5.15

\[ g_s = 1.945 \times 10^{-2} \] Eq. 5.16
This is the final value that was used in the code for use in FCM module to get similar results. This value when used to act on particle the effect of gravitational pull (force on particle) will be similar to that experienced by particle in the experiment in laboratory environment.

Confirmation by putting values in Force equation:

\[
F = \frac{0.14 \times 1.945 \times 10^{-2} \times (12 \times 10^{-3})^2}{1.42 \times 10^{-5} \times 10^{-1}} = 0.27613
\]

\[
V_0 = \frac{10^{-2}}{700} = 1.42857 \times 10^{-5}
\]

5.3 Post processing and Results Observations

1) MSD- This statistical function stands for Mean Square Displacement gives us the variation of displacement of particle in the domain with reference to the original position.

Mathematically it is represented as

\[
MSD(t) = < (x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 >
\]

Script was created in MATLAB and particle position was imported by the output file of Jadim. As a result we get a MSD function with respect to the time step or iterative position with time.

This is shown in following Figure 25.
Auto Correlation- Auto correlation in this study is the correlation between a variable that is constantly varying with respect to time, between its latest and previous values. It is also known as lagged correlation. It is used to find the repeating pattern in the data set. So in this study plot of correlation shows the repetition of the normalised mean velocity. As can be seen in the Figure 25

Mathematically correlation is expressed as 
\[ r(\tau) = \frac{\langle u(t)u(t+\tau) \rangle}{\langle u_{rms,particle} \rangle} \]

Figure 26 MSD of particle data collected over various initial positions.

Figure 27 Correlation of U velocity
This can also be expressed in relation with turbulence. If this plot is straight as it reaches higher time steps, it displays that the turbulence is fully developed and the particle behaviour is completely random. But in our study the fluctuation around zero value displays the flow is far from turbulent. This is true as we limit our self to Reynolds number in laminar region for particle simulation.

PDF- This stands for Probability Distribution function which in simple terms can be expressed as probability of finding particle at specific differential range. PDF of mean velocity normalized with rms velocity is shown in figure below.
Figure 30 PDF of particle

\[
\frac{V_i - \langle V_i \rangle}{V_{\text{rms}}}
\]
6. Conclusion

Conclusion:

Study of particle behaviour was completed with monopole particle. Physical behaviour of particle was successfully predicted based on analysis of particle trajectory data. MSD and PDF of particle position and Correlation of self-particle velocity show how the particle behaviour in laminar Von Karman flow looks like.

Formation of doughnut shaped vortex ring was studied with plots and calculation of Flux in X direction and Poloidal and Toroidal values in Y and Z direction. Flux plot in X direction shows how the position of the vortex rings vary in X direction with varying Reynolds number. Ratio of poloidal/toroidal value in Y direction remains almost stable with very limited variation till Reynolds number 650 and after this it goes for constant increment. Talking about same ratio in Z direction it goes for constant decrement till Reynolds number 650 and after this ratio increases rapidly. This suggests a stable vortex ring presence (at Lower Re) in the domain that starts its dynamic behaviour above Re650.

Based on mean axial velocity plot at location of top and bottom of disc we conclude variation of Reynolds number shifts the profile line away from disc. For starting 1 cm from disc, profile at different Re overlaps, this signifies the effect of boundary layer effect. After this slope of profile reduces with increase in Reynolds number and profile becomes more and more straight (variation on 15 nodes or till next 8 cm) till centre of domain. For next 5 cm of central region in domain shows no significant variation in mean velocity with increase in Re.

On particle behaviour from plot of PDF we conclude that particle shows behaviour in agreement with the similar experiments carried out in this field. However the plots obtained are not exactly symmetrical as there is data loss in this study, due to particle hitting the wall and simulation stops. To take into consideration other dynamic statistic we have to remove some portion of particle data from few time steps previous to particle reaching wall. Plot of correlation gives us a wavy kind of picture that suggests that the particle is following the circular motion of the fluid and it keeps its correlation symmetric around zero value. This means movement of particle
is similar around its original position. This plot also suggests that the particle has not correlated to the flow completely but partially and that the flow turbulence is still not developed. This is a verified fact as we do the simulation in the laminar region and has not taken this particle study into range of unsteady flow or turbulent flow regimes.

Limited time frame of this study makes it limited to only one particle dynamics but further work can be carried out on this study to observe these behaviours in unsteady and turbulent conditions. And also this study is limited to one particle study only so we cannot conclude on the behaviour of particle on turbophoresis.

Based on this study we can conclude that this study successfully predicts the particle behaviour in Von Karman Flow condition along with FCM module, within its numerical constrains and that the code Jadim is capable of predicting particle behaviour in various flow regimes.

**Future Work:**

This project is done in time constrains and the study has been done without modification in the original code. The original code was tested and found capable of simulating Von Karman Flow and effect of particle presence in the flow. However the problem faced during simulation that may affect the accuracy of data collected is that the particle repulsive force, rebound effect is absent from the code and so the actually physical effect of particle behaviour (how does particle rebounds and behave after being hit by wall, disc or any other particle if present). There is no repulsive force taken into account in Jadim for particle. Modelling this force in near future will enable longer simulation and thus longer particle trajectory which in turn will give us much smooth statistics.

Work can be carried out in modification of Petsc scheme as it is a slow solver. Computational time will be saved by doing this. This study uses the Petsc scheme that is a slow solver scheme; if some mathematical modification in this scheme can be done it will help in future studies.

Particle statistics can be studied and observed by modifying lot of physical properties like, different values of relative densities, particle diameter, dipole and
quadpole particle, and by varying the number of particles more than one. This study can lead to further investigation of how if there are more no of particles present is increased, statistical and dynamics change can be observed. This study is limited to observation of behaviour in steady flow, but next study can be taken up to compare the behaviour in unsteady flow.
References


[E3] "Toroidal coord" by DaveBurke - Own work. Licensed under CC BY 2.5 via Wikimedia Commons - https://commons.wikimedia.org/wiki/File:Toroidal_coord.png#/media/File:Toroidal_coord.png
Appendices

A1 Autocorrelation programme In Fortran 90

PROGRAM rij
INTEGER, PARAMETER::P=264324, M=132162
REAL(10)::sum,out(M),data(P), Var, P2
INTEGER::k,j
P2= P;
open(unit= 3,file="upartdata.dat")
do k=1,P
read(3,*): data (s)
end do
!
! Compute the mean velocity u
Summation=0
do s=1,P
summation = sum + data(s)
end do
u=summation/P
summation = 0
! compute the variance urms=SQRT(Variance) = Ruu(0)
do s=1,P
summation = summation + (data(s)-u)*(data(s)-u)
end do
VAR = summation/(P-1)
do s=1,M
out(s) = 0
do j=s+1,P
out(s) = out(s) + (data(j)-u)*(data(j-s)-u);
end do
out(s) = out(s)/(VAR*P)
end do
open (unit = 8, file = "CORRX.txt")
do s=1,M
write (8,*),out(s)
end do
close(8)

!!!!For V velocity
open (unit = 3, file = "vpartdata.dat")
do k=1,P
read(3,*) data(k)
end do
!
Compute the mean velocity \( u \)
sum=0
do k=1,P
  sum = sum + data(k)
end do
u = sum/P
sum = 0
!
Compute the variance \( \text{urms} = \sqrt{\text{VARiance}} = \text{Ruu}(0) \)
do k=1,P
  sum = sum + (data(k)-u)*(data(k)-u)
end do
VAR = sum/(P-1)
do k=1,M
  out(k) = 0
  do j=k+1,P
    out(k) = out(k) + (data(j)-u)*(data(j-k)-u);
  end do
  out(k) = out(k)/(VAR*P)
end do
!
For \( Z \) velocity
open (unit = 4, file = "wpartdata.dat")
do k=1,P
  read(4,*) data(k)
end do
!
Compute the mean velocity \( u \)
sum=0
do k=1,P
  sum = sum + data(k)
end do
u = sum/P
sum = 0
!
Compute the variance \( \text{urms} = \sqrt{\text{VARiance}} = \text{Ruu}(0) \)
do k=1,P
  sum = sum + (data(k)-u)*(data(k)-u)
end do
VAR = sum/(P-1)
do k=1,M
  out(k) = 0
  do j=k+1,P
    out(k) = out(k) + (data(j)-u)*(data(j-k)-u);
  end do
end do

!!!
END PROGRAM
A2 Matlab code for Vortex identification by Q criterion and lambda 2 criterion

clear
cle
steps=input('Enter the time steps: ')
range=input('Enter the length of time step: ')
start=input('Enter the start time step: ')
nu=input('Kinetic Viscous : ')

for t=1:1:steps
  %read data
  %n is *.asc are there we want to treat ,t is the time step account. Define a 4-d array to store all datas in the calculation U(t,i,j,k)
  %time strp is t, i is mesh count along x, j is for y direction, and k for z.
  % Input the kx, ky kz, to know mesh count in each direction
  %t1=clock; % Calculate starting time
  a=start+(t-1)*range;
  % get the length of 'a'
  b=num2str(a);
  d=b;
  for c=length(b):1:8
    if c<8
      d=sprintf('%s%s','0',d);
    end
  end
  name=sprintf('%s%s%s','res_',d,'.asc');
  % lecture_asc has been modified
  lecture_asc;
  % Calculate the gradient of velocity in three directions, matlab
  % gradient, if it is not boundary, Central interpolation, if it is boundary, then interpolate by Forward scheme or backward
  xx(1:ni)=X(1:ni,1,1);
  for k=1:nk
    for j=1:nj
      ux(:,j,k)=gradient(U(:,j,k),xx);
      vx(:,j,k)=gradient(V(:,j,k),xx);
      wx(:,j,k)=gradient(W(:,j,k),xx);
    end
  end
  yy(1:nj)=Y(1,1:nj,1);
  for k=1:nk
    for i=1:ni
      uy(i,:,k)=gradient(U(i,:,k),yy);
      vy(i,:,k)=gradient(V(i,:,k),yy);
      wy(i,:,k)=gradient(W(i,:,k),yy);
    end
  end


end
zz(1:nk)=Z(1,1,1:nk);
for i=1:ni
    for j=1:nj
        uuz=squeeze(U(i,j,:))';
        uuz(i,j,:)=gradient(uuz,zz);
        vuz=squeeze(V(i,j,:))';
        vz(i,j,:)=gradient(vuz,zz);
        wuz=squeeze(W(i,j,:))';
        wz(i,j,:)=gradient(wuz,zz);
    end
end
%% Vortex identification
cptl2=0;
cptQ=0;
cptDelta=0;
% Get rotational tensor and strain tensor using velocity gradient tensor
for i=1:ni
    for j=1:nj
        for k=1:nk
            gru=[ux(i,j,k) uy(i,j,k) uz(i,j,k) ; vx(i,j,k) vy(i,j,k) vz(i,j,k) ; wx(i,j,k) wy(i,j,k) wz(i,j,k)];
            S=(gru+gru')/2;
            A=(gru-gru')/2;
            T=S*S+A*A;
            % Lamda2 criterion is to get the second biggest eignvalue of T tensor.
            lamda=sort(eig(T));
            lamda2(i,j,k)=lamda(2); %lamda2 criterion
            % Q criterion is to get the average of three eignvalues of T tensor.
            Q_criterion(i,j,k)=-(1/2*(lamda(1)+lamda(2)+lamda(3))); %Q criterion,
            ADRIAN 2005
            % Delta criterion, (Q criterion is restrictive than Delta criterion)
            Delta(i,j,k)=(Q_criterion(i,j,k)/3).^3+(det(gru)/2).^2;
            if lamda2(i,j,k)<0
                cptl2=cptl2+1 ;
            end
            if Q_criterion(i,j,k)<0
                cptQ=cptQ+1 ;
            end
            vort(i,j,k)=sqrt(2*(trace(A*A')));
        end
    end
end
rotx=wy-vz;
roty=uz-wx;
rotz=vx-uy;

for i=1:1:ni
    for j=1:1:nj
        for k=1:1:nk
            U_total(i,j,k,t)=U(i,j,k);
            V_total(i,j,k,t)=V(i,j,k);
            W_total(i,j,k,t)=W(i,j,k);
%P_total(i,j,k,t)=P(i,j,k);
        end
    end
end

%Set different matrixes to store values in a 4d matrix, the fourth
%variable is time.
lamda2_total(:,:,,:,t)=lamda2(:,:,);
Q_total(:,:,,:,t)=Q_criterion(:,:,);
%lamdaci_total(:,:,,:,t)=lamdaci(:,:,);
%rotx_total(:,:,,:,t)=rotx(:,:,);
%roty_total(:,:,,:,t)=roty(:,:,);
%rotz_total(:,:,,:,t)=rotz(:,:,);
end

%%
for t=1:1:steps %Get the non-dimension values of Q_criterion
    Q_total_max(t)=max(max(max(Q_total(:,:,,:,t))));
    Q_total_min(t)=min(min(min(Q_total(:,:,,:,t))));
    Delta_Q(t)=Q_total_max(t)-Q_total_min(t);
    Q_total_dimless(:,:,,:,t)=Q_total(:,:,,:,t)/Q_total_max(t);
end
for j=1:1:nj %Get velocity fluctuation
    umean(j)=mean(mean(mean(U_total(:,j,:,:,:),4),3),1);
    up_total(:,j,:,:,:)=U_total(:,j,:,:,:)-umean(j);
    vmean(j)=mean(mean(mean(V_total(:,j,:,:,:,:),4),3),1);
    vp_total(:,j,:,:,:)=V_total(:,j,:,:,:)-vmean(j);
    wmean(j)=mean(mean(mean(W_total(:,j,:,:,:,:),4),3),1);
    wp_total(:,j,:,:,:)=W_total(:,j,:,:,:)-wmean(j);
end

% Calculate viscous scale, set coordinate positions as y+
Umean_gradient=abs(gradient(umean,yy));
U_tao=(nu*(Umean_gradient(1)+Umean_gradient(nj))/2)^0.5;
delt_nu=nu/U_tao;
Re_tao=U_tao*(max(Y(1,:,1))+min(Y(1,:,1)))/2/nu;
X_plus=xx./delt_nu;
Y_plus=yy./delt_nu;
Z_plus=zz./delt_nu;
X_plus_mesh=X./delt_nu;
Y_plus_mesh=Y./delt_nu;
Z_plus_mesh=Z./delt_nu;

%%% Calculate energy bulget
%1. Production term, shear flow p=-<uv>\frac{dU}{dY}
uv_p_total=up_total.*vp_total;
Production=-mean(mean(mean(uv_p_total,1),3),4).*gradient(umean,yy);

%2. Dissipation term in Turbulent Kinetic Energy Reynolds Equation
Dissipation=nu<\frac{d\omega}{dxj} \frac{d\omega}{dxj}>,
for t=1:1:steps
  for k=1:nk
    for j=1:nj
      up_x_total(:,j,k,t)=gradient(up_total(:,j,k,t),xx);
      vp_x_total(:,j,k,t)=gradient(vp_total(:,j,k,t),xx);
      wp_x_total(:,j,k,t)=gradient(wp_total(:,j,k,t),xx);
    end
  end
end
for k=1:nk
  for i=1:ni
    up_y_total(i,:,k,t)=gradient(up_total(i,:,k,t),yy);
    vp_y_total(i,:,k,t)=gradient(vp_total(i,:,k,t),yy);
    wp_y_total(i,:,k,t)=gradient(wp_total(i,:,k,t),yy);
  end
end
for i=1:ni
  for j=1:nj
    upz=squeeze(up_total(i,j,:,t))';
    up_z_total(i,j,:,t)=gradient(uuz,zz);
    vpz=squeeze(vp_total(i,j,:,t))';
    vp_z_total(i,j,:,t)=gradient(vvz,zz);
    wpz=squeeze(wp_total(i,j,:,t))';
    wp_z_total(i,j,:,t)=gradient(wwz,zz);
  end
end
end
A3. Energy dissipation code in Matlab

% strain rate and dissipation
%

for t=1:1:steps
    for k=1:nk
        for j=1:nj
            dux(:,j,k,t)=gradient(U_total(:,j,k,t),xx);
            dvx(:,j,k,t)=gradient(V_total(:,j,k,t),xx);
            dwx(:,j,k,t)=gradient(W_total(:,j,k,t),xx);
        end
    end
    for k=1:nk
        for i=1:ni
            duy(i,:,k,t)=gradient(U_total(i,:,k,t),yy);
            dvy(i,:,k,t)=gradient(V_total(i,:,k,t),yy);
            dwy(i,:,k,t)=gradient(W_total(i,:,k,t),yy);
        end
    end
    for i=1:ni
        for j=1:nj
            %                 for k = 1:nk
            %                    AAA(k)=U_total(i,j,k,t);
            %                    BBB(k)=V_total(i,j,k,t);
            %                    CCC(k)=W_total(i,j,k,t);
            %                end
            duz(i,j,:,t)=gradient(AAA,zz);
            dvz(i,j,:,t)=gradient(BBB,zz);
            dwz(i,j,:,t)=gradient(CCC,zz);
        end
    end
    end
    for k=1:nk
        % for k = 1:nk
        AAA(k)=U_total(i,j,k,t);
        BBB(k)=V_total(i,j,k,t);
        CCC(k)=W_total(i,j,k,t);
        end
        duz(i,j,:,t)=gradient(AAA,zz);
        dvz(i,j,:,t)=gradient(BBB,zz);
        dwz(i,j,:,t)=gradient(CCC,zz);
        end
        test1=squeeze(U_total(i,j,:,t))';
        test2=squeeze(V_total(i,j,:,t))';
        test3=squeeze(W_total(i,j,:,t))';
        duz(i,j,:,t)=gradient(test1,zz);
        dvz(i,j,:,t)=gradient(test2,zz);
        dwz(i,j,:,t)=gradient(test3,zz);
        end
        % can be done in this way
        dvz(i,j,:,t)=gradient(V_total(i,j,:,t),zz);
        dwz(i,j,:,t)=gradient(W_total(i,j,:,t),zz);
    end
end
end
% now combining the terms for strain rate

\[ t \]
\[ \text{end} \]

\[
\begin{align*}
\sigma_{11} &= 0.5*(\Delta u_x + \Delta u_x) \\
\sigma_{12} &= 0.5*(\Delta u_y + \Delta v_x) \\
\sigma_{13} &= 0.5*(\Delta u_z + \Delta w_x) \\
\sigma_{21} &= 0.5*(\Delta v_x + \Delta u_y) \\
\sigma_{22} &= 0.5*(\Delta v_y + \Delta v_y) \\
\sigma_{23} &= 0.5*(\Delta v_z + \Delta w_y) \\
\sigma_{31} &= 0.5*(\Delta w_x + \Delta u_z) \\
\sigma_{32} &= 0.5*(\Delta w_y + \Delta v_z) \\
\sigma_{33} &= 0.5*(\Delta w_z + \Delta w_z)
\end{align*}
\]

\[
\begin{align*}
\text{Diss}11 &= 2*nu*(\sigma_{11}.*\sigma_{11} + \sigma_{12}.*\sigma_{12} + \sigma_{13}.*\sigma_{13}) \\
\text{Diss}22 &= 2*nu*(\sigma_{21}.*\sigma_{21} + \sigma_{22}.*\sigma_{22} + \sigma_{23}.*\sigma_{23}) \\
\text{Diss}33 &= 2*nu*(\sigma_{31}.*\sigma_{31} + \sigma_{32}.*\sigma_{32} + \sigma_{33}.*\sigma_{33})
\end{align*}
\]

\[
\text{Diss}_t = \text{squeeze(mean((Diss}11+\text{Diss}22+\text{Diss}33),4))}
\]

\[
\begin{align*}
\text{itest} &= \text{trapz}(zz,\text{Diss}_t,3) \\
\text{itest}_2 &= \text{trapz}(yy,\text{itest},2) \\
\text{Dissipation} &= \text{trapz}(xx,\text{itest}_2,1) \\
\text{Diss_dimles} &= \text{Dissipation}/((0.095^2))
\end{align*}
\]
A4. Poloidal and Toroidal calculation using Matlab

% Harshit Bhatia 06/04/2015
% toroidal calculations Version 2

% time averaging removed, absolute values used, so no mean required in % script

% Ratio= Poloidal Values/Toroidal Values

% Converting Velocities to Absolute values
U_total_pos=abs(U_total);
V_total_pos=abs(V_total);
W_total_pos=abs(W_total);

% Resultant_V, required for poloidal calculations
Resultant_XY= sqrt(U_total.*U_total+V_total.*V_total);
Resultant_XZ= sqrt(U_total.*U_total+W_total.*W_total);

% Flux Calculation
for i=1:1:ni
    uxf=squeeze(U_total_pos(i,:,:));
    flux(i)=trapz(zz,trapz(yy,uxf,1),2)/hdeno;
end

% Toroidal Calculation
for j=1:1:nj
    vxf=squeeze(V_total_pos(:,j,:));
    Tor_y(j)=trapz(xx,trapz(zz,vxf,2),1)/hdeno;
end

for k=1:1:nk
    wxf=squeeze(W_total_pos(:,:,k));
    Tor_z(k)=trapz(yy,trapz(xx,wxf,1),2)/hdeno;
end

% Poloidal Calculation
for j=1:1:nj
    Res_XY_sqz= squeeze(Resultant_XY(:,j,:));
    Pol_y(j)=(trapz(xx,trapz(zz,Res_XY_sqz,1),2))/hdeno;
end
for k=1:1:nk
    Res_XZ_sqz= squeeze(Resultant_XZ(:,:,k));
    Pol_z(k)=(trapz(xx,trapz(yy,Res_XZ_sqz,1),2))/hdeno;
end

figure (1)
pplot(flux,'b-o')
title('Flux in X dir')
saveas(gcf,'flux in x dir','.jpg')
close(figure(1))

figure (2)
pplot(Tor_y,'b-*')
title('Toroidal value in Y dir')
saveas(gcf,'Toroidal in y dir','.jpg')
close(figure(2))

figure (3)
pplot(Tor_z,'b+-')
title('Toroidal value in Z dir')
saveas(gcf,'Toroidal in z dir','.jpg')
close(figure(3))

figure (4)
pplot(Pol_y,'r-x')
title('Poloidal value in Y dir')
saveas(gcf,'Poloidal in y dir','.jpg')
close(figure(4))

figure (5)
pplot(Pol_z,'r-s')
title('Poloidal value in Z dir')
saveas(gcf,'Poloidal in z dir','.jpg')
close(figure(5))

Ratio_y=(Pol_y/Tor_y)
Ratio_z=(Pol_z/Tor_z)
A5. Power and Torque calculation in Matlab script

% For Power calculation
%

r=0.095
center=[0,0]
t3=[yy(50),zz(50)];
t1=[yy(1),zz(1)];
% t2=meshgrid(zz,yy)

yloc_a=Y(1,:,:)-center(1);
yloc=squeeze(yloc_a);
zloc_a=Z(1,:,:)-center(1);
zloc=squeeze(zloc_a);

mu= nu*1000
% calculations for disc omega matrix
for j=1:nj
    for k=1:nk
        if sqrt(yloc(j,k).*yloc(j,k)+zloc(j,k).*zloc(j,k))<=r
            om(j,k)=1;
        else
            om(j,k)=0;
        end
    end
end

Tyx= mu * dvx;
% Tyy= mu * dvy;
% Tyz= mu * dvz;
Tao_y_a= Tyx;
Tao_y_mean= mean(Tao_y_a,4);

Tao_y_wall1=Tao_y_mean(1,:,:);  % calculations at 1st layer of wall
Tao_y_wall1_2d=squeeze(Tao_y_wall1);  % calculations at 1st layer of wall

Tao_y_wall2=Tao_y_mean(50,:,:);  % calculations at last layer of wall
Tao_y_wall2_2d=squeeze(Tao_y_wall2);  % calculations at last layer of wall

Tzx= mu * dwx;
% Tzy= mu * dwy;
% Tzz= mu * dwz;
Tao_z_a= Tzx;
Tao_z_mean= mean(Tao_z_a,4);
Tao_z_wall1 = Tao_z_mean(1,:,:);
% calculations at 1st layer of wall
Tao_z_wall1_2d = squeeze(Tao_z_wall1);
% calculations at 1st layer of wall

Tao_z_wall2 = Tao_z_mean(50,:,:);
% calculations at last layer of wall
Tao_z_wall2_2d = squeeze(Tao_z_wall2);
% calculations at last layer of wall

Prod1_wall1_ty = Tao_y_wall1_2d.*zloc*(-1.*om);
% calculations at 1st layer of wall
Prod1_wall2_ty = Tao_y_wall2_2d.*zloc*(-1.*om);
% calculations at 1st layer of wall
Prod2_wall1_tz = Tao_z_wall1_2d.*yloc.*om;
% calculations at last layer of wall
Prod2_wall2_tz = Tao_z_wall2_2d.*yloc.*om;
% calculations at last layer of wall

Power1 = Prod1_wall1_ty + Prod2_wall1_tz;
Power2 = Prod1_wall2_ty + Prod2_wall2_tz;
Power = Power1 + Power2;

ptest_2 = trapz(yy,ptest_3,1);
ptest_4 = trapz(yy,ptest_3,1);
ptest_2 = % calculations at 1st layer of wall
ptest_4 = % calculations at last layer of wall
A6. Post Processing script in Matlab to cut plane and get contours along with 2d glyphs

% fich=input('Nom du fichier ascii à lire?ut1,'s');
fich=name
fid=fopen(fich);

% Lecture file dimension in direction x,y,z respectively
ni=fscanf(fid,'%i',1);
jf=fscanf(fid,'%i',1);
nk=fscanf(fid,'%i',1);

% Lecture file number
nbvar=fscanf(fid,'%i',1);

% no of physical time steps
temps=fscanf(fid,'%f',1);

% Reading the no of variables
for l=1:nbvar,

    % name of variable
    nomvar=fscanf(fid,'%s',1)

    % declaring array of no of elements as ni*nj*nk
    [var,compt]=fscanf(fid,'%e',ni*nj*nk);

    % reshaping the variable matrix
    for k=1:nk
        var=reshape(var,nj*nk,ni);
        var2(1:ni,1:nj,k)=var(k:nk:nj*nk,1:ni)';
    end

    % variable concerned
    switch (nomvar)
    case ('XP')
        X=var2;
    case ('YP')
        Y=var2;
    case ('ZP')
        Z=var2;
    case ('UP')
        U=var2;
    case ('VP')
        V=var2;
    case ('WP')
        W=var2;
    case ('PP')
        P=var2;
    case ('T2')
        TAU=var2;
    case ('T3')
    end
TAU3 = var2;
case ('TI')
    TI = var2;
case ('LC')
    LC = var2;
case ('VX')
    VortX = var2;
case ('VY')
    VortY = var2;
case ('VZ')
    VortZ = var2;
case ('TP')
    Tempe = var2;
case ('E1')
    A = var2;
case ('E2')
    B = var2;
otherwise
    'name not valid'
end
end
clear var var2;
fclose(fid);
y1(:,:,1) = Y(1,:,:);
z1(:,:,1) = Z(1,:,:);
v1(:,:,1) = V(1,:,:);
w1(:,:,1) = W(1,:,:);
norm1(:,:,1) = sqrt(v1.^2+w1.^2);
yend(:,:,1) = Y(end,:,:);
zend(:,:,1) = Z(end,:,:);
vend(:,:,1) = V(end,:,:);
wend(:,:,1) = W(end,:,:);
normend(:,:,1) = sqrt(vend.^2+wend.^2);
ycentxy(:,:,1) = Y(:,:,end/2);
xcentxy(:,:,1) = X(:,:,end/2);
vcentxy(:,:,1) = V(:,:,end/2);
ucentxy(:,:,1) = U(:,:,end/2);
normcentxy(:,:,1) = sqrt(vcentxy.^2+ucentxy.^2);
zcentxz(:,:,1) = Z(:,end/2,:);
xcentxz(:,:,1) = X(:,end/2,:);
wcentxz(:,:,1) = W(:,end/2,:);
vcentxz(:,:,1) = V(:,end/2,:);
ucentxz(:,:,1) = U(:,end/2,:);
pcentxz(:,:,1) = P(:,end/2,:);
normcentxz(:,:,1) = sqrt(wcentxz.^2+ucentxz.^2);
```matlab
figure(1)
set(gca, 'FontSize', 14, 'fontName','Arial');
hold on
contourf(y1,z1,norm1,25)
quiver(y1(1:2:end-1,1:2:end-1),z1(1:2:end-1,1:2:end-1),v1(1:2:end-1,1:2:end-1),w1(1:2:end-1,1:2:end-1),1.5,'w')
colorbar
xlabel('Z')
ylabel('Y')
axis equal
axis tight
axis([-0.1 0.1 -0.1 0.1]);
set(gca, 'YTick',-0.1:0.1:0.1, 'YTickLabel',{'-0.1' '0' '0.1'})
set(gca, 'XTick',-0.1:0.1:0.1, 'XTickLabel',{'-0.1' '0' '0.1'})
box('on')
saveas(gcf,'velocity_magnitude_face_disc1','jpg')
close(figure(1))

figure(2)
set(gca, 'FontSize', 14, 'fontName','Arial');
hold on
contourf(yend,zend,normend,25)
quiver(yend(1:2:end-1,1:2:end-1),zend(1:2:end-1,1:2:end-1),vend(1:2:end-1,1:2:end-1),wend(1:2:end-1,1:2:end-1),1.5,'w')
colorbar
xlabel('Z')
ylabel('Y')
axis equal
axis tight
axis([-0.1 0.1 -0.1 0.1]);
set(gca, 'YTick',-0.1:0.1:0.1, 'YTickLabel',{'-0.1' '0' '0.1'})
set(gca, 'XTick',-0.1:0.1:0.1, 'XTickLabel',{'-0.1' '0' '0.1'})
box('on')
saveas(gcf,'velocity_magnitude_face_disc2','jpg')
close(figure(2))
```
```matlab
figure(3)

set(gca, 'FontSize', 14, 'fontName','Arial');
hold on

contourf(xcentxy,ycentxy,normcentxy,25)
quiver(xcentxy(1:2:end-1,1:2:end-1),ycentxy(1:2:end-1,1:2:end-1),ucentxy(1:2:end-1,1:2:end-1),vcentxy(1:2:end-1,1:2:end-1),1.5,'w')
colorbar
xlabel('X')
ylabel('Y')

axis equal
axis tight
axis([-0.1 0.1 -0.1 0.1]);
set(gca, 'YTick',-0.1:0.1:0.1, 'YTickLabel',{'-0.1' '0' '0.1'})
set(gca, 'XTick',-0.1:0.1:0.1, 'XTickLabel',{'-0.1' '0' '0.1'})
box('on')
saveas(gcf,'velocity_magnitude_middle_plane_XY','jpg')
close(figure(3))

figure(4)

set(gca, 'FontSize', 14, 'fontName','Arial'); %taille et type police
hold on

contourf(xcentxz,zcentxz,normcentxz,25)
quiver(xcentxz(1:2:end-1,1:2:end-1),zcentxz(1:2:end-1,1:2:end-1),ucentxz(1:2:end-1,1:2:end-1),wcentxz(1:2:end-1,1:2:end-1),1.5,'w')
colorbar
xlabel('X')
ylabel('Z')

axis equal
axis tight
axis([-0.1 0.1 -0.1 0.1]);
set(gca, 'YTick',-0.1:0.1:0.1, 'YTickLabel',{'-0.1' '0' '0.1'})
set(gca, 'XTick',-0.1:0.1:0.1, 'XTickLabel',{'-0.1' '0' '0.1'})
box('on')
saveas(gcf,'velocity_magnitude_middle_plane_XZ','jpg')
close(figure(4))
```
figure(5)

set(gca, 'FontSize', 14, 'fontName', 'Arial');
hold on

contourf(xcentxz, zcentxz, vcentxz, 25)
quiver(xcentxz(1:2:end-1, 1:2:end-1), zcentxz(1:2:end-1, 1:2:end-1), ucentxz(1:2:end-1, 1:2:end-1), wcentxz(1:2:end-1, 1:2:end-1), 1.5, 'k')
colorbar
xlabel('X')
ylabel('Z')

axis equal
axis tight
axis([-0.1 0.1 -0.1 0.1]);
set(gca, 'YTick', -0.1:0.1:0.1, 'YTickLabel', {'-0.1' '0' '0.1'})
set(gca, 'XTick', -0.1:0.1:0.1, 'XTickLabel', {'-0.1' '0' '0.1'})
box('on')
saveas(gcf, 'normal_velocity_middle_plane_XZ', 'jpg')
close(figure(5))

figure(6)

set(gca, 'FontSize', 14, 'fontName', 'Arial');
hold on

contourf(xcentxz, zcentxz, pcentxz, 25)
quiver(xcentxz(1:2:end-1, 1:2:end-1), zcentxz(1:2:end-1, 1:2:end-1), ucentxz(1:2:end-1, 1:2:end-1), wcentxz(1:2:end-1, 1:2:end-1), 1.5, 'w')
colorbar
xlabel('X')
ylabel('Z')

axis equal
axis tight
axis([-0.1 0.1 -0.1 0.1]);
set(gca, 'YTick', -0.1:0.1:0.1, 'YTickLabel', {'-0.1' '0' '0.1'})
set(gca, 'XTick', -0.1:0.1:0.1, 'XTickLabel', {'-0.1' '0' '0.1'})
box('on')
saveas(gcf, 'pressure_middle_plane_XZ', 'jpg')
close(figure(6))

% to save the data set for further post processing

save ('Dissipation', 'Dissipation', 'Diss_dimles', 'ptest_2')
save ('Profile', 'prof1', 'prof2', 'prof3')
save ('Ratio', 'Ratio_y', 'Ratio_z')
save ('Flux', 'flux')
save ('P_T', 'Pol_z', 'Pol_y', 'Tor_z', 'Tor_y')
A7. Velocity Profile Plot

% Profile of RE_W=(vitesse_Z/0.095)

figure(1)
prof1 = squeeze(W_total(:,49,25,:))
plot(xx,mean(prof1,2),'b- >')
title('Profile on Top line')
saveas(gcf,'Profile on Top line','jpg')
close(figure(1))

figure (2)
prof2 = squeeze(W_total(:,2,25,:))
plot(xx,mean(prof2,2),'b-*')
title('Profile on Bottom line')
saveas(gcf,'Profile on Bottom line','jpg')
close(figure(2))

figure (3)
prof3 = squeeze(W_total(:,25,25,:))
plot(xx,mean(prof3,2),'b-x')
title('Profile on Middle line')
saveas(gcf,'Profile on Middle line','jpg')
close(figure(3))
clear all
% close all
%cd /home/hbhatia/Bureau/post_p

%% 1st data set

s='traj';
urms_bar=0;vrms_bar=0;wrms_bar=0;umean_bar=0;vmean_bar=0;wmean_bar=0
Nsamples=0
for VAR=1:1:1
    s1=num2str([VAR]);
    name=num2str([s1 s]);
    A=load (name);
    upart=squeeze(A(:,6));
    % vpart=squeeze(A(:,7));
    % wpart=squeeze(A(:,8));
    dt=5E-3;
    time_step=squeeze(A(:,2))*dt;
    %for i = 1:length(upart)
    urms=sqrt(mean((upart -mean(upart)).*(upart -mean(upart))));
    % vrms=sqrt(mean((vpart -mean(vpart)).*(vpart -mean(vpart))));
    % wrms=sqrt(mean((wpart -mean(wpart)).*(wpart -mean(wpart))));
    % umrs=sqrt(mean(upart.^2));
    % vmrs=sqrt(mean(vpart.^2));
    % wmrs=sqrt(mean(wpart.^2));
    %end
meanu = mean(upart);

% meanv = mean(vpart);
% meanw = mean(wpart);

urms_bar = urms_bar + urms;
% vrms_bar = vrms_bar + vrms;
% wrms_bar = wrms_bar + wrms;

umean_bar = umean_bar + meanu;
% vmean_bar = vmean_bar + meanv;
% wmean_bar = wmean_bar + meanw;

Nsamples = Nsamples + 1
end

urms_bar = urms_bar / Nsamples;
% vrms_bar = vrms_bar / Nsamples;
% wrms_bar = wrms_bar / Nsamples;
umean_bar = umean_bar / Nsamples;
% vmean_bar = vmean_bar / Nsamples;
% wmean_bar = wmean_bar / Nsamples;

for VAR = 1:1:1
    s1 = num2str([VAR]);
    name = num2str([s1 s]);
    A = load(name);
    upart = squeeze(A(:,6));
    % vpart = squeeze(A(:,7));
    % wpart = squeeze(A(:,8));
    dt = 5E-3;
    time_step = squeeze(A(:,2)) * dt;
    for j = 1:length(upart)
x(j)=(upart(j)-umean_bar)/urms_bar;
% y(j)=(vpart(j)-vmean_bar)/vrms_bar;
% z(j)=(wpart(j)-wmean_bar)/wrms_bar;
end
end

Pdfx(length(x))=zeros;
% Pdfy(length(x))=zeros;
% Pdfz(length(x))=zeros;
for i =1:length(upart)
    for j = 1:length(upart)
        tolx = abs(x(i)) - abs(x(j));
        % toly = abs(y(i)) - abs(y(j));
        % tolz = abs(z(i)) - abs(z(j));
        if (tolx <= 0.0001)
            Pdfx(i)=Pdfx(i)+1;
        end
        % if (toly <= 0.0001)
        %     Pdfy(i)=Pdfy(i)+1;
        % end
        % if (tolz <= 0.0001)
        %     Pdfz(i)=Pdfz(i)+1;
        % end
    end
end

% for i =length(upart)/2:length(upart)
%     for j = length(upart)/2:length(upart)
% tol = x(i) - x(j);
% if (tol <= 0.0001)
% Ppdf(i)=Ppdf(i)+1;
% end
% end
% end
% figure (1)
plot (x,Pdfx/(max(Pdfx)))
hold
% plot (y,Pdfy/(max(Pdfy)))
% plot (z,Pdfz/(max(Pdfz)))
%
% figure (2)
% plot (x,Pdfx/(max(Pdfx)))
%
% figure (3)
% plot (y,Pdfy/(max(Pdfy)))
% figure (4)
% plot (z,Pdfz/(max(Pdfz)))
A9 Mean Square Displacement

load re1g.traj

time_step=squeeze(re1g(:,2));

xpart=squeeze(re1g(:,3));
ypart=squeeze(re1g(:,4));
zpart=squeeze(re1g(:,5));
upart=squeeze(re1g(:,6));
vpart=squeeze(re1g(:,7));
wpart=squeeze(re1g(:,8));

x0=xpart(1)
y0=ypart(1)
z0=zpart(1)

% try1
% for t=1:95000
% msd(t)= mean(((xpart(t)-x0)^2)+((ypart(t)-y0)^2)+((zpart(t)-z0)^2));
% end
% plot (msd)

% try2
for t=2:95000
    msd(1)= (xpart(t)-xpart(t-1))^2;
end
plot (msd)

up_rms=rms(upart)