

# Large eddy simulation of turbulent flow between counter rotating concentric cylinders

Kameswararao Anupindi

[kanupind@purdue.edu](mailto:kanupind@purdue.edu)

Graduate Student

School of Mechanical Engineering, Purdue University

AAE626 Spring 2011 Project Report

April 29, 2011

## Contents

<b>Nomenclature</b>	<b>2</b>
<b>1 Abstract</b>	<b>2</b>
<b>2 Introduction</b>	<b>2</b>
<b>3 Simulation Method</b>	<b>3</b>
3.1 Governing Equations . . . . .	3
3.2 Geometry & Mesh Details . . . . .	4
3.3 Solver Details . . . . .	4
3.4 Initial Conditions . . . . .	4
<b>4 Results &amp; Discussion</b>	<b>5</b>
<b>5 Conclusions</b>	<b>10</b>
<b>A Appendix</b>	<b>12</b>
A.1 Description of OpenFOAM CFD toolbox . . . . .	12
A.2 Periodic Boundary Conditions in OpenFOAM . . . . .	12
A.3 Coordinate Transformation of Turbulent Statistics . . . . .	13

## List of Figures

1	Mesh used for the Taylor Couette flow simulations . . . . .	5
2	(a)Mean azimuthal velocity profiles (b) Zero azimuthal velocity surface . . . . .	6
3	DNS results of Dong [3] (a) Mean averaged azimuthal velocity (b) Zero azimuthal velocity surface . . . . .	6
4	Instantaneous velocity vectors in a radial-axial plane at different Reynolds numbers . . . . .	7
5	Comparison of profiles of $u'_\theta rms$ , and $\langle u'_r u'_\theta \rangle$ at different $Re$ . . . . .	8
6	Iso surfaces of $\lambda_2$ at $Re = 4000$ colored by radial velocity . . . . .	9
7	Spatio-temporal contours of the azimuthal velocity along a fixed line parallel to z-axis at $Re = 4000$ . . . . .	9
8	Eigen spectrum for velocity components at $Re = 4000$ . . . . .	10
9	Instantaneous velocity vectors in a radial-axial plane at different Reynolds numbers . . . . .	11

## List of Tables

1	Dominant eigen values for radial velocity component ( $u_r$ )	10
2	Dominant eigen values for azimuthal velocity component ( $u_\theta$ )	10
3	Dominant eigen values for axial velocity component ( $u_z$ )	10

## Nomenclature

		$Re_i$	Inner Reynolds number
$\eta$	Radius ratio ( $= R_i/R_o$ )	$Re_o$	Outer Reynolds number
$\langle F \rangle$	Time averaged value/mean of variable $F$	$u'_\theta rms$	r.m.s fluctuation azimuthal velocity
$\nu$	Fluid kinematic viscosity	$u'_r rms$	r.m.s fluctuation radial velocity
$\Omega_i$	Angular velocity of the inner cylinder	$U_i$	Rotation velocity of the inner cylinder
$\Omega_o$	Angular velocity of the outer cylinder	$U_o$	Rotation velocity of the outer cylinder
$\bar{F}$	Filtered component of $F$	CRTC	Counter Rotating Taylor Couette Flow
$d$	Annular gap between the cylinders ( $R_o - R_i$ )	DNS	Direct Numerical Simulation
$f$	Sub-Grid Scale component of $F$	GS	Grid Scale
$f'$	Fluctuation component of variable $F$	LES	Large Eddy Simulation
$L_z$	Axial dimension of the cylinders	r.m.s	Root mean square
$r$	Radial location	RANS	Reynolds Averaged Navier Stokes
$R^*$	Radius of zero azimuthal velocity surface	Re	Reynolds Number
$R_i$	Radius of the inner cylinder	SGS	Sub-Grid Scale
$R_o$	Radius of the outer cylinder		

## 1 Abstract

Flow between differentially rotating cylinders, also known as Counter Rotating Taylor-Couette (CRTC) system exhibit a wide variety of flow states comprising separate laminar and turbulent regions as well as flow states with co-existence of both of them [3]. In this project we focus on simulating incompressible turbulent flow in a CRTC system using large eddy simulation (LES) turbulence model available in OpenFOAM [1] CFD toolbox developed by OpenCFD Ltd. The statistical features of the flow field such as time-averaged mean field and the r.m.s velocity fluctuations are computed for different  $Re$  numbers and shown to be similar to published DNS results reported by Dong [3]. The dynamical features of the flow such as instantaneous velocity field, and instantaneous iso-surfaces of  $\lambda_2$  (the intermediate eigenvalue in Jeong & Hussain [4]) are computed from the simulations. Also, spatio-temporal azimuthal velocity contours close to the inner cylinder show herring-bone line patterns similar to the published DNS results[3]. Dynamic mode decomposition (DMD) [7] of the obtained flow field is conducted and the velocity fields are reconstructed from the computed eigen values. These orthogonal modes could be used further to perform a dynamical system analysis using reduced order modeling techniques. Overall, the results obtained demonstrate the capability of LES to simulate strongly rotating wall bounded flows.

## 2 Introduction

Rotating turbulent flows are ubiquitous in science and engineering. Examples of such flows include atmospheric and ocean flows as well as flows in the wake of ship propellers, jet engines, and wind turbines. Rotating turbulent flows exhibit a number of features that are not present in turbulent flows without rotation. For example, rotating

turbulent flow in a pipe will have azimuthal velocity which makes the flow no longer unidirectional. Also, the effects of centrifugal forces change the turbulence dynamics of the system.

Previously DNS was performed by Dong [3] for this CRTS systems at a range of  $Re$  numbers varying from  $Re = 500$  to  $Re = 4000$ , and a detailed study of the statistical and dynamical features were reported. First LES calculations were performed by Bazilevs, and Akkerman [2] quite recently for a standard Taylor-Couette flow at a  $Re = 8000$ . Bazilevs, and Akkerman [2] used a residual-based variational multiscale (RBVMS) turbulence modeling approach in their LES calculations instead of eddy-viscosity based models as at very high  $Re$  the high flow velocity rotation arrests the energy cascade process and can pose difficulty in capturing the flows accurately. But in the present simulation we focus only on an eddy-viscosity model (Dynamic Smagorinsky model) for all the simulations and the highest  $Re$  considered is only 4000.

In the present study of CRTS system, the inner and outer cylinders are rotating in opposite directions at constant but different angular velocities  $\Omega_i$ , and  $\Omega_o$  respectively. The flow geometry is characterized by the radius ratio,  $\eta = R_i/R_o$ , where  $R_i$  and  $R_o$  are, respectively, the radii of the inner and outer cylinders. The annulus gap width  $d = (R_o - R_i)$  is taken as the length-scale. Two  $Re$  numbers namely, inner and outer  $Re$  numbers which are  $Re_i$ , and  $Re_o$  are defined as  $Re_i = U_i d/\nu$  and  $Re_o = U_o d/\nu$ , where  $\nu$  is the fluid kinematic viscosity, and  $U_i$ ,  $U_o$  are, the tangential velocities on the inner and outer cylinder walls, respectively. In all the simulations performed here,  $Re = Re_i = -Re_o$  and radius ratio was kept at  $\eta = 0.5$

## 3 Simulation Method

### 3.1 Governing Equations

The basic equations for LES were first formulated by Smagorinsky [9] in the context of weather prediction models. As the computational resources were limited to resolve all scales of motion, an alternative turbulence model such as LES was proposed. Based on the theory of Kolmogorov that the smallest scales of motion were uniform and they mainly serve to dissipate energy from the larger scales to the smallest scales through cascade process, it was thought that these uniform smallest scales, or their effect on the larger scales could be modeled while resolving the larger scales. Hence in LES, the largest scales which contain the most of the energy, which are effected strongly by the boundary conditions and do the most of the transporting are calculated directly while the smallest scales are represented by a model.

To separate the large scales of motion from the small some kind of averaging must be done. In LES, this averaging operator is not ensemble average as in RANS, but a filter which is a locally derived weighted average of flow properties over a volume of fluid. Filter width,  $\Delta$ , is one of the properties of the filtering process, which is a characteristic length scale and has the approximate effect that scales larger than  $\Delta$  (resolved or super-Grid Scales (GS)) are retained in the filtered flow field while scales smaller than  $\Delta$  (Sub-Grid Scales (SGS)) are modeled for their effect on the GS.

In LES, any flow  $F$  variable can be composed of a large scale (filtered part  $\bar{F}$ ) and a small scale (residual/SGS part  $f$ ) contribution as follows:

$$F = \bar{F} + f$$

The governing equations are the filtered incompressible Navier-Stokes equations, assuming that the filter commutes with differentiation we can write these equations as follows [6]:

$$\frac{\partial \bar{U}_i}{\partial x_i} = 0 \tag{1}$$

$$\frac{\partial \bar{U}_j}{\partial t} + \bar{U}_i \frac{\partial \bar{U}_j}{\partial x_i} = \nu \frac{\partial^2 \bar{U}_j}{\partial x_i \partial x_i} - \frac{1}{\rho} \frac{\partial \bar{p}}{\partial x_j} - \frac{\partial \tau_{ij}^r}{\partial x_i} \tag{2}$$

The equations 1, 2 look similar to RANS equations. In the above equations the term  $\tau_{ij}^r$  is the anisotropic part of the residual-stress tensor/SGS stress tensor and the isotropic part has been absorbed into the filtered pressure term. Like the RANS equations for  $\langle \mathbf{U} \rangle$ , the filtered equations for  $\bar{\mathbf{U}}$  are unclosed. Closure is achieved by modeling the residual (or SGS) stress tensor  $\tau_{ij}^r$ . In Smagorinsky model the SGS stress tensor  $\tau_{ij}^r$  is computed as follows:

$$\tau_{ij}^r = -2(C_s \Delta)^2 |\bar{S}| \bar{S}_{ij} \tag{3}$$

Where  $C_s$  is a model constant for the Smagorinsky model. But in the present simulations Dynamic Smagorinsky model was used in which the model constant is calculated using averaging over homogeneous directions in the present case (axial, and radial are periodic directions). The constant is evaluated from the resolved flow field using Germano's procedure as follows

$$C = -\frac{1}{2} \frac{\langle L_{ij} M_{ij} \rangle}{\langle M_{kl} M_{kl} \rangle} \quad (4)$$

where

$$M_{ij} = -2\hat{\Delta}^2 |\widehat{S}| \widehat{S}_{ij} + 2\Delta^2 |\widehat{S}| \widehat{S}_{ij} \quad (5)$$

and

$$L_{ij} = \widehat{u_i u_j} - \hat{u}_i \hat{u}_j \quad (6)$$

### 3.2 Geometry & Mesh Details

The geometry of the domain consists of two circular cylinder with  $R_i = 1$ ,  $R_o = 2$  and the length of the domain in axial direction is taken as  $L_z = \pi$ . Both the inner and outer cylinders act as impermeable no-slip walls and the axial direction uses periodic boundary conditions in order to simulate an infinitely long cylinders.

Mesh was generated using commercial meshing software GAMBIT and imported to OpenFOAM using **fluentMeshToFoam** utility. Special care needs to be taken when generating and importing periodic boundaries from GAMBIT to OpenFOAM and the detailed procedure for that is described in Appendix A.2. Mesh was stretched at the inner and outer walls and a fairly uniform mesh was generated in other regions of the flow field. Boundary layer meshing was used for generating stretched mesh in these regions.

Average  $y^+$  of 0.0647 is maintained on the outer wall with minimum and maximum being 0.04 and 0.1, where as average  $y^+$  of 0.11 is maintained on the inner rotating wall with minimum and maximum being 0.065 and 0.17. These values were obtained as a post processing step from OpenFOAM for the  $Re = 4000$  case. But to start with the mesh was generated with a first grid point located at  $5e - 04$  from each wall, i.e.  $\Delta r = 5e - 04$ , and the first grid points in axial and azimuthal directions were set to  $\Delta z = 0.032$ , and  $\Delta \theta = 0.02$ . With the above calculated  $y^+$  and assuming the following ratios  $y^+/\Delta r$ ,  $z^+/\Delta z$ , and  $\theta^+/\Delta \theta$  are equal we can calculate  $z^+$  and  $\theta^+$  as 13 and 8 respectively for the maximum value of  $y^+ = 0.2$  in the domain. With these settings the mesh was generated and it contains a total of 1.7 million cells. The mesh that was used for all calculations is shown in Figure 1, in which clustering of cells near inner and outer walls can be seen.

### 3.3 Solver Details

OpenFOAM 1.7.1 was used for the simulations reported in the present work.  **pisoFoam**  solver which is based on PISO-SIMPLE (Pressure Implicit with Splitting of Operators) algorithm together with Semi Implicit Method for Pressure Linked Equations for handling the pressure-velocity coupling is utilized for doing LES simulations in the present work. More details about the solver in specific can be found in OpenFOAM documentation [1]. All the spatial discretization are second order accurate and time discretization is performed using second order accurate backward differencing scheme. Dynamic Smagorinsky model [5] is used as SGS model.

### 3.4 Initial Conditions

As the transition of the flow from laminar to turbulent regime is not captured by the simulation process we need to artificially trigger this transition process so that the flow develops into a turbulent one. As there are no inlet boundaries in the present simulation, the only way perturbations can be introduced into the flow is by generating an initial perturbation field and let this trigger transition. Schoppa and Hussain [8] discuss about streak instability mechanism in a channel flow using a *sinuous* perturbations. But the perturbations alone cannot trigger transition so they have to be imposed on a base flow field so that after few flow through a turbulent flow field will be generated. Hence, a converged laminar flow field solution was used as the base flow with the sinuous perturbations. Starting with **perturbCylU** utility I modified it so as to take the base flow field and add perturbations. The perturbations that are imposed are as follows. Where the details of various parameters are omitted here but are taken same as discussed in Schoppa and Hussain[8].

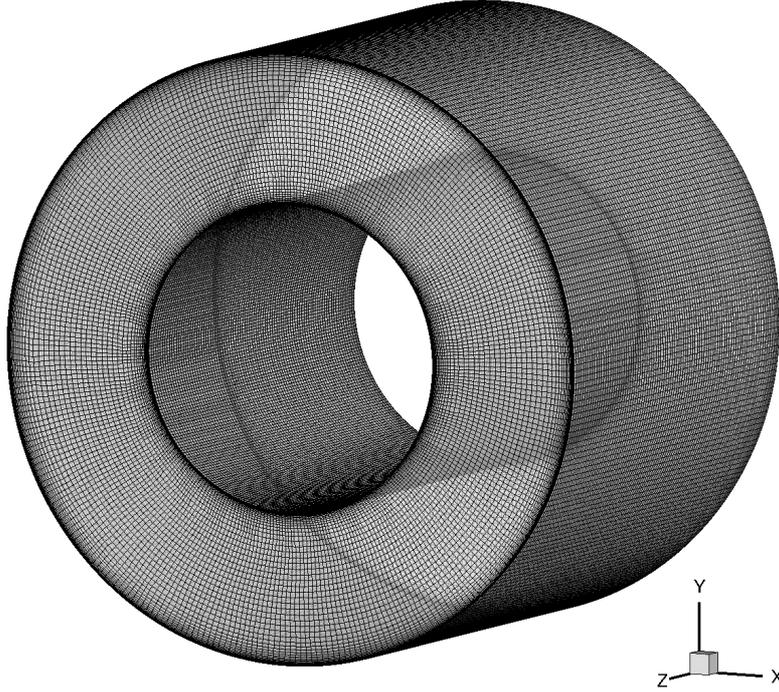


Figure 1: Mesh used for the Taylor Couette flow simulations

$$u'_z = \frac{u_\tau \Delta u^+}{2} \cos(\beta^+ r^+ \theta) \frac{r^+}{30} \exp(-\sigma(r^+)^2 + 0.5) \quad (7)$$

$$u'_\theta = \epsilon \sin(\alpha^+ z^+) r^+ \exp(-\sigma(r^+)^2) \quad (8)$$

## 4 Results & Discussion

Simulations were performed for the CRTC system for  $Re = 500, 1500,$  and  $4000$ . Few specific post processing such as extraction of  $\lambda_2$  contours etc...was performed only for the results of  $Re = 4000$  case. All the cases were run using Dynamic Smagorinsky as SGS model in LES and they have been run upto a  $t = 50$  with a time step of  $\Delta t = 1e - 03$ . Simulations were performed using parallel OpenFOAM solver with Message Passing Interface as the parallelization technique using domain decomposition. A total of 16 processors were utilized for running the solver and each of the simulations have taken a total wall clock time of 100 hrs. Purdue RCAC research cluster **Rossmann** was used for running the code.

In order to validate the results obtained, we first compare the variation of mean azimuthal velocity profiles on  $Z = 0$  plane at different Reynolds numbers as shown in part(a) of Figure 2. The corresponding results from DNS of Dong [3] can be seen in Figure 3. The results obtained from present calculations seem to be matching to that of the DNS results. In the present case the mean profiles are just taken from  $Z = 0$  plane rather than averaged over  $\theta, z$  directions as was done in Dong [3]. This could probably be the reason for not getting a sharp region in the center where all curves match. As the  $Re$  is increased from 500 to 4000 we can see from in part (a) of Figure 2 that mean azimuthal velocity profiles have become steeper with a wider zero azimuthal region in the central portion away from both the walls. Unlike a channel flow the mean velocity profiles are not symmetric and the profile seems to be closer to the inner wall more than it does at the outer wall rendering the asymmetry.

Contours of zero azimuthal velocity surface are shown in part (b) Figure 2 from the present calculations, where as the DNS results are shown in part (b) of Figure 3. Again, as there was no easy way to take the averaging over radial and axial directions in OpenFOAM, I had to just take the  $Z = 0$  plane results here to calculate the zero azimuthal

velocity surface. As the  $Re$  is increased this surface moves outwards from the inner wall as observed in the DNS results also. The zero velocity surface seems to be approaching a asymptotic value as the  $Re$  is increased. From these two plots the mean values seems to be matching well to the DNS results of Dong [3].

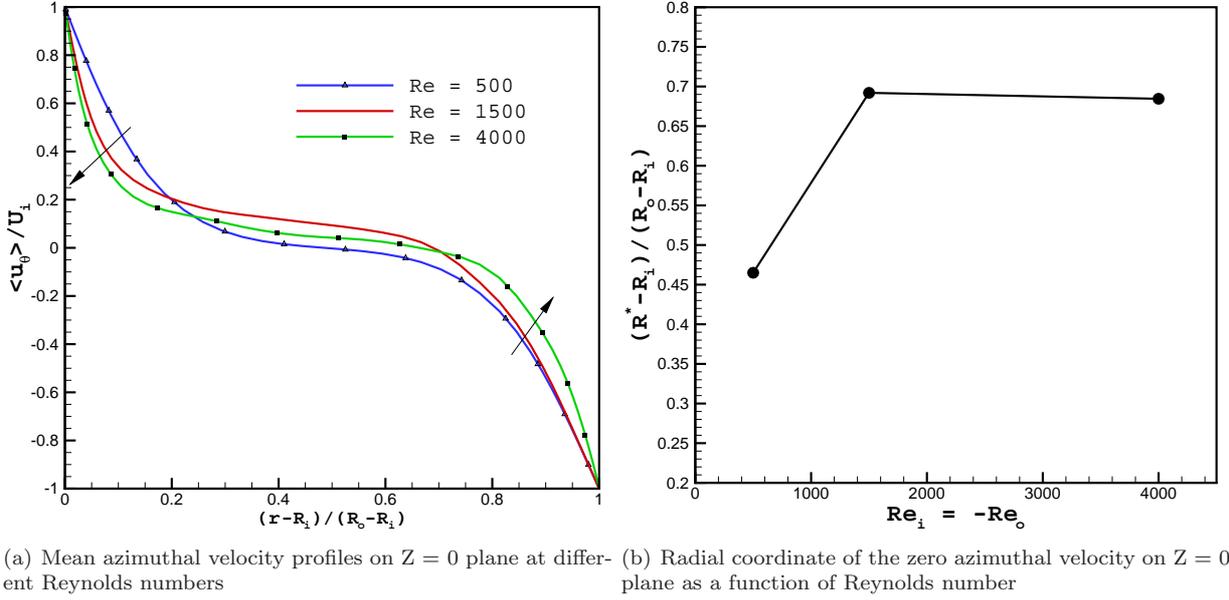


Figure 2: (a) Mean azimuthal velocity profiles (b) Zero azimuthal velocity surface

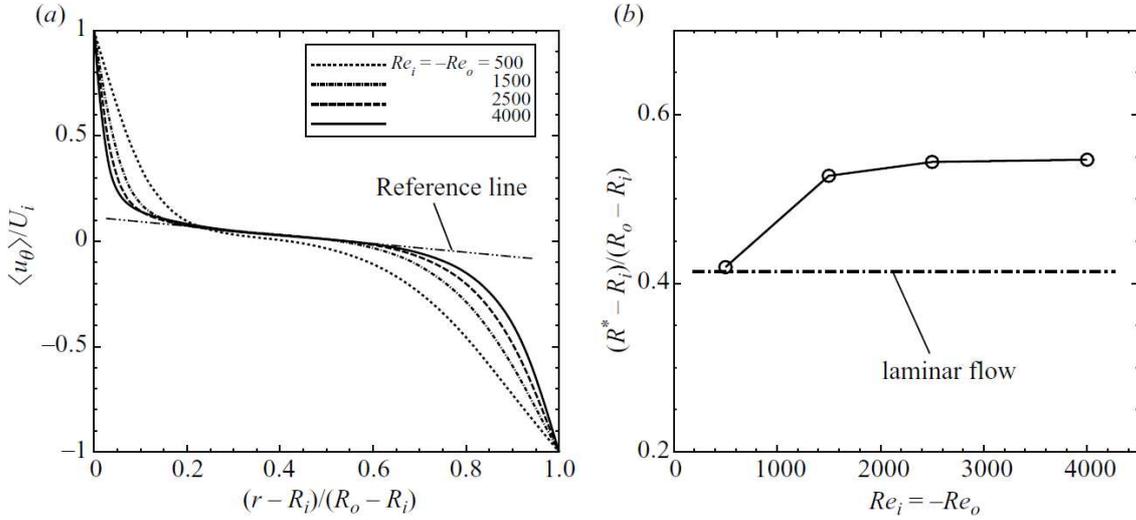


Figure 3: DNS results of Dong [3] (a) Mean averaged azimuthal velocity (b) Zero azimuthal velocity surface

In order to visualize the highly vortical turbulent field, we plot the instantaneous velocity vectors in a radial - axial plane at  $Re = 500, 1500,$  and  $4000$  in Figure 4. We can see from these instantaneous velocity vectors plots that at  $Re = 500$  the flow seems to be still laminar and nice regular vortices are slowly moving out towards the outer wall. Also the symmetry of the flow about  $Z = 0$  can be seen. As the  $Re$  is increased to 1500 and further to 4000 the symmetric, regular pattern has been lost with onset of turbulence and an irregular velocity field which indicates enhanced mixing is observed in parts (b), and (c) of Figure 4

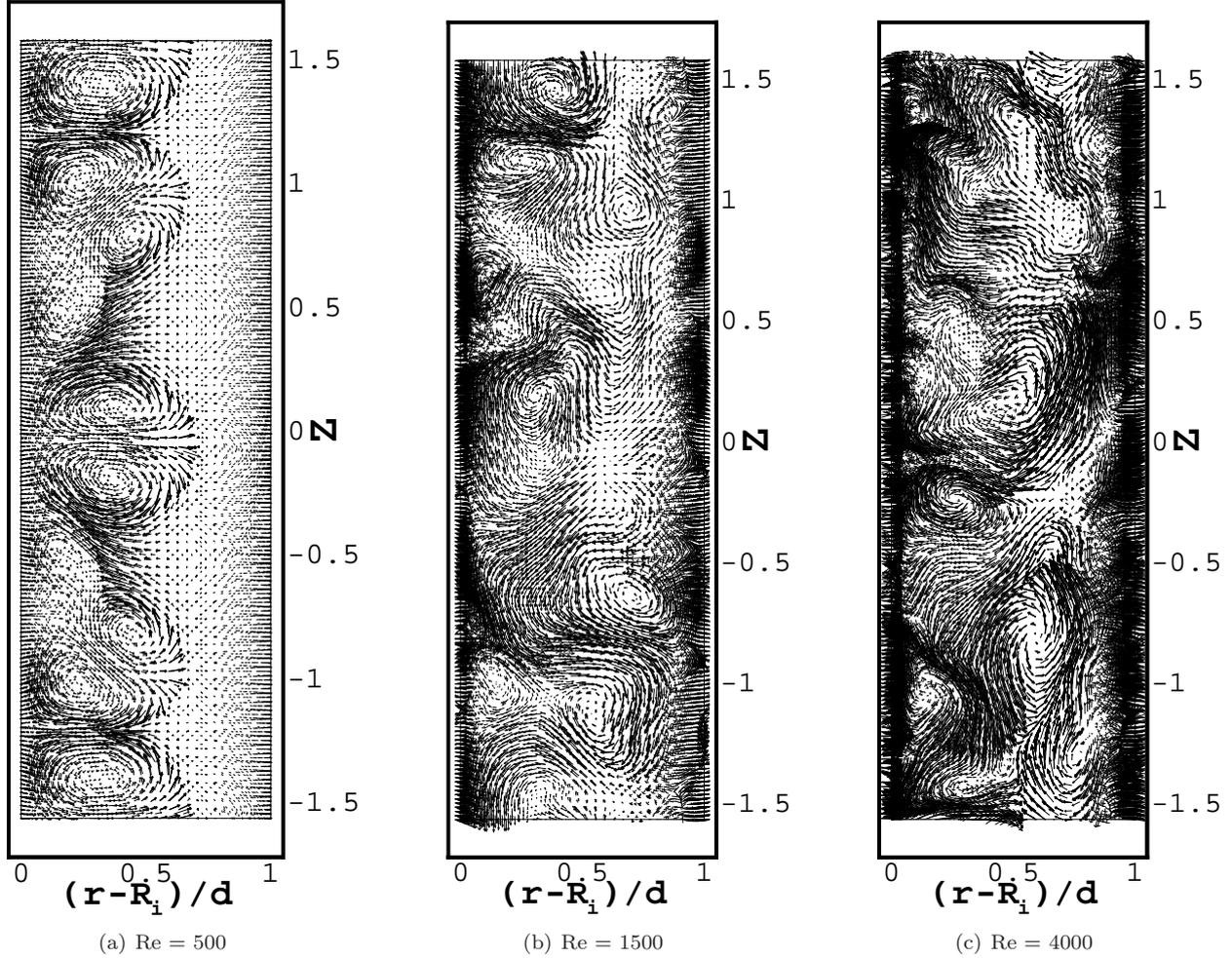
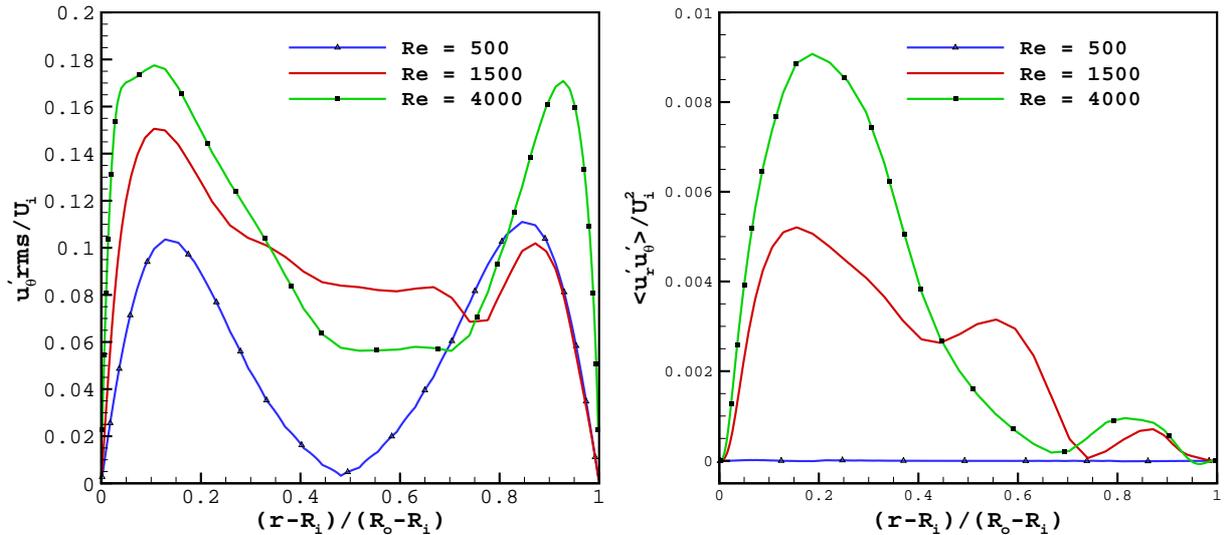


Figure 4: Instantaneous velocity vectors in a radial-axial plane at different Reynolds numbers

Next, we look at the r.m.s azimuthal velocity fluctuation for different  $Re$  numbers as plotted in Figure 5(a). From these plots we see two peaks closer to the wall and the fluctuations reaching a minimum value in the central portion of the annulus. As the  $Re$  is increased from 500 to 4000 the peaks move closer to the walls indicating the presence of more near wall vortices and fluctuations. These observations are also similar and match with that made by Dong [3]. Also, the Reynolds stress values  $\langle u'_r u'_\theta \rangle$  are shown plotted in Figure 5(b). From this plot we can see that Reynolds stresses are very small for the case of  $Re = 500$  indicating very little turbulence in this case. As the  $Re$  is increased we see that peak closer to the inner wall moves higher and higher indicating more intense fluctuations and near wall vortices. Where as at the outer wall all the plots seem to closing in a tail like fashion. Through out all  $Re$  we observe a positive Reynolds stress  $\langle u'_r u'_\theta \rangle$ , which can be understood by simply analogy taught in the class. For e.g. a positive radial velocity fluctuation tends to transport fluids of higher azimuthal velocity away from the inner wall that making the product positive; a negative radial velocity fluctuation will tend to transport fluid of lower azimuthal velocity toward the inner wall (i.e. in negative  $r$  direction) hence making the product positive again. The r.m.s azimuthal velocity fluctuation and Reynolds stress in cylindrical coordinates  $\langle u'_r u'_\theta \rangle$  are computed in Tecplot from their Cartesian counterparts written by OpenFOAM through the simulation. A simple transformation of Reynolds stress and other relevant quantities from Cartesian to Cylindrical is included for the sake of completeness in Appendix A. 3.

There are many ways in which the vortical cores/coherent structures in a turbulent field can be extracted/visualized.



(a) R.m.s azimuthal fluctuation velocity on  $Z = 0$  plane along radial line at different  $Re$  (b) Reynolds stress  $\langle u'_r u'_\theta \rangle$  on  $Z = 0$  plane along radial line at different  $Re$

Figure 5: Comparison of profiles of  $u'_{\theta,rms}$ , and  $\langle u'_r u'_\theta \rangle$  at different  $Re$

For e.g. iso-surfaces of vorticity, or helicity, or  $Q$ -criterion,  $\lambda_2$  are few such examples. In the present work we used  $\lambda_2$  the second eigen value in the tensor  $\mathbf{S}\mathbf{S} + \mathbf{\Omega}\mathbf{\Omega}$  where  $\mathbf{S}$  and  $\mathbf{\Omega}$  are the symmetric and anti-symmetric parts of the velocity gradient tensor. Plotting of these  $\lambda_2$  surfaces as discussed in Jeong & Hussain [4] helps us to visualize and explore the structural characteristics of the small-scale vortices in the turbulent CRTC system. The Figure 6 shows numerous small scale vortices extending in general in azimuthal direction. Many azimuthal vortices seem to originating from the inner and outer walls and convecting into the central portion of the annulus. In the present CRTC system the vortices seem to be elongated in azimuthal direction rather than hairpin-like as seen in turbulent channels and flat-plate boundary layers. However, there is a chance for few hairpin-like vortices to form which needs to be further explored.

Next, temporal values of instantaneous azimuthal velocity  $u_\theta$  were collected along a line parallel to the  $z$  - axis at a distance  $r = 0.0033d$  from the inner wall. Using this the spatio-temporal azimuthal velocity contours are plotted on  $z - t$  axes as shown in Figure 7. From this figure we can see herringbone-like pattern for the instantaneous velocity. These kind of patterns were also observed in the presence of near-wall streaks in the standard turbulent Taylor-Couette flow system [3].

Next, I performed a dynamic mode decomposition of the obtained turbulent data. Dynamic mode decomposition (DMD), extracts modes which are orthogonal to each other in a temporal sense purely from a sequence of snapshots as described in Schmid [7]. As the domain under consideration was huge with 1.7 million cells, I took only a slice of the domain  $y = 0$  plane and performed the analysis on it. The eigen values obtained by DMD match with that of the ones obtained by linear stability analysis if the process through which these were generated was a linear one. For a non-linear process like in the present case they would be different but identify the dominant eigen modes. A separate Fortran90 code was used (which was implemented as part of ME611 course project last semester) for this purpose. I had to customize the code so that it will read unstructured mesh data from OpenFOAM, where originally the code was only able to read structured mesh data.

A series of 120 snapshots were written for the horizontal slice that was considered with a successive time difference of 0.05. Each of the velocity components  $u_r, u_\theta, u_z$  were processed separately and the eigen spectrum were obtained as described in the paper by Schmid [7]. The details of the process are not repeated here but can be found in the reference paper by Schmid [7]. The eigen spectrum obtained for each of the velocity components is shown in Figure 8. Please note for these plots the scales on  $x$  and  $y$  axes are not same the axes have been stretched so as to visualize the eigen spectrum.

The first three dominant eigen values (positive  $\lambda_r$  values are considered here) are also tabled for each case in

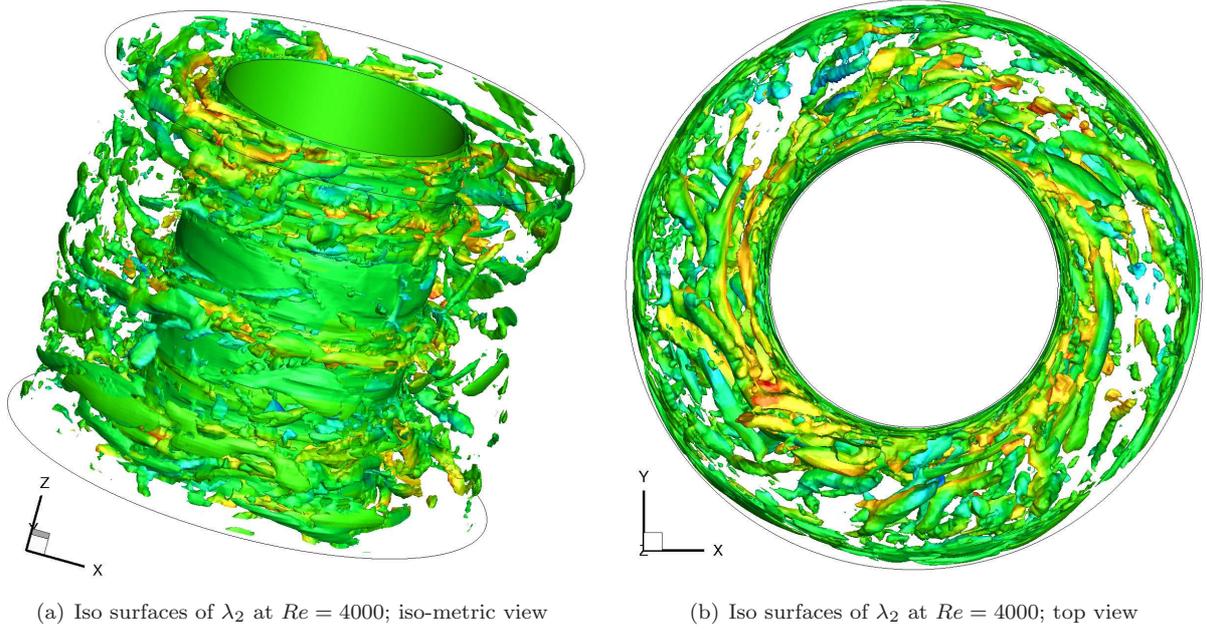


Figure 6: Iso surfaces of  $\lambda_2$  at  $Re = 4000$  colored by radial velocity

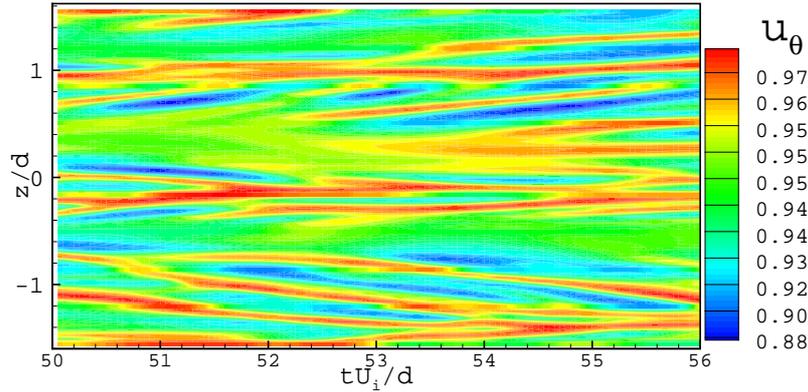


Figure 7: Spatio-temporal contours of the azimuthal velocity along a fixed line parallel to  $z$ -axis at  $Re = 4000$

Tables 1, 2, 3. The eigen-spectrum shown in Figures 8 depicts a whole range of eigen values for each of the velocity component. The spectrum seems to be symmetric about the  $\lambda_i$  axis which is a consequence of processing real valued data. Also, the eigen values of  $u_\theta$  span a wide range of  $\lambda_i$  compared to the other two components of velocity. The size and color of the eigen values spotted in the eigen spectrum correspond to the coherence of the respective eigen mode, which is like projecting the obtained eigen functions on to the proper orthogonal basis as described in Schmid [7]. The computed values of coherence were also tabulated against each eigen value in Tables 1, 2, 3.

Next using these eigen modes and eigen functions, the velocity field is reconstructed from the dominant eigen modes of each of the velocity components. The original instantaneous velocity field is shown plotted in part (a) of Figure 9. The three other vector fields constructed from each of the eigen modes of all velocity components are shown in parts (b), (c), (d) of Figure 9. These results shown in parts (c) -(d) can be used to construct low order modeling of the present CRTC system and can be used for dynamical system analysis. Much of the linear algebra work related to the method is not repeated here but can be found in the paper by Schmid [7].

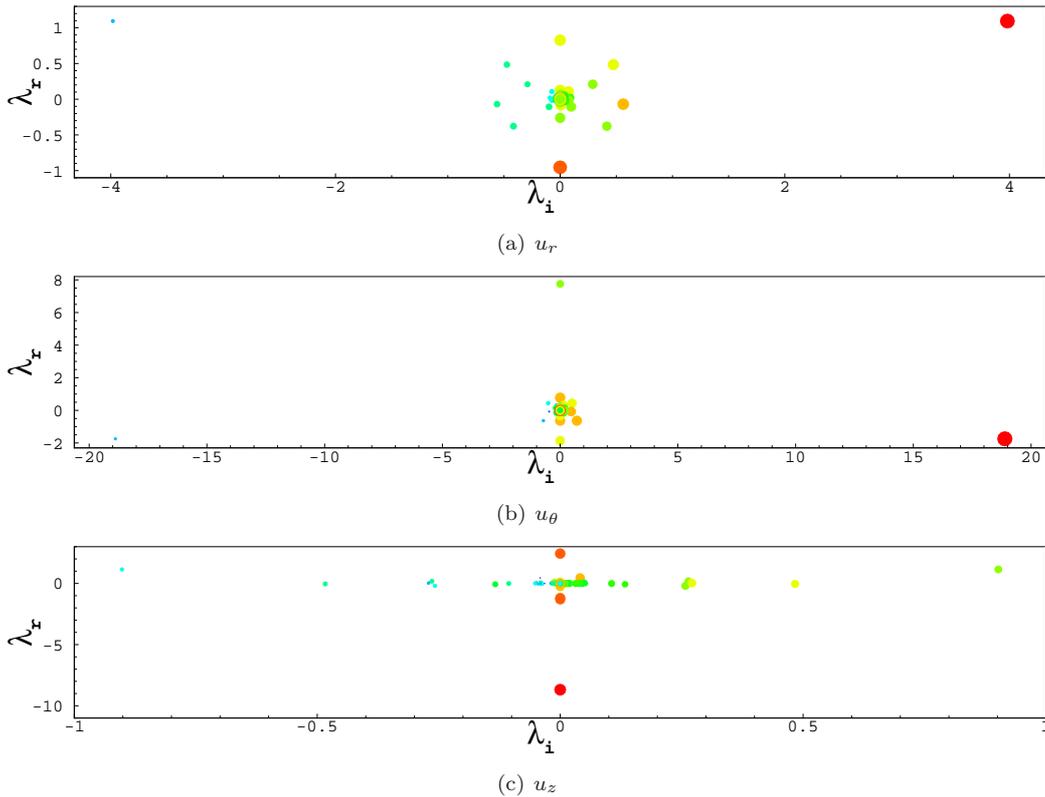


Figure 8: Eigen spectrum for velocity components at  $Re = 4000$

$\lambda_i$	$\lambda_r$	Coherence
3.9818	1.0943	6.347E-003
0.00000	0.8258	4.999E-003
0.47374	0.4838	4.837E-003

Table 1: Dominant eigen values for radial velocity component ( $u_r$ )

$\lambda_i$	$\lambda_r$	Coherence
0.0000	7.7596	4.0212E-003
0.00000	0.7761	5.4063E-003
0.50885	0.4406	4.8136E-003

Table 2: Dominant eigen values for azimuthal velocity component ( $u_\theta$ )

$\lambda_i$	$\lambda_r$	Coherence
0.0000	2.4400	5.7897E-003
0.9017	1.1445	4.3175E-003
0.0041	0.4505	5.2267E-003

Table 3: Dominant eigen values for axial velocity component ( $u_z$ )

## 5 Conclusions

Dynamic Smagorinsky model as the SGS model is used to simulate a CRTIC system at different  $Re$  and the results seem to be matching to the published DNS results. Also, the turbulent statistics and the vortical structures obtained are similar to the DNS results. A DMD on the flow data for the  $Re = 4000$  case reveals that the azimuthal eigen values span a wide range of values than the other two components. The reconstructed velocity fields from the obtained eigen functions can be further used to construct reduced order modeling for dynamical system analysis. The results obtained demonstrate the capability of LES to simulate strongly rotating wall bounded flows.

Overall, the project has given me an opportunity to learn and use OpenFOAM for the first time and quickly adapt to it.

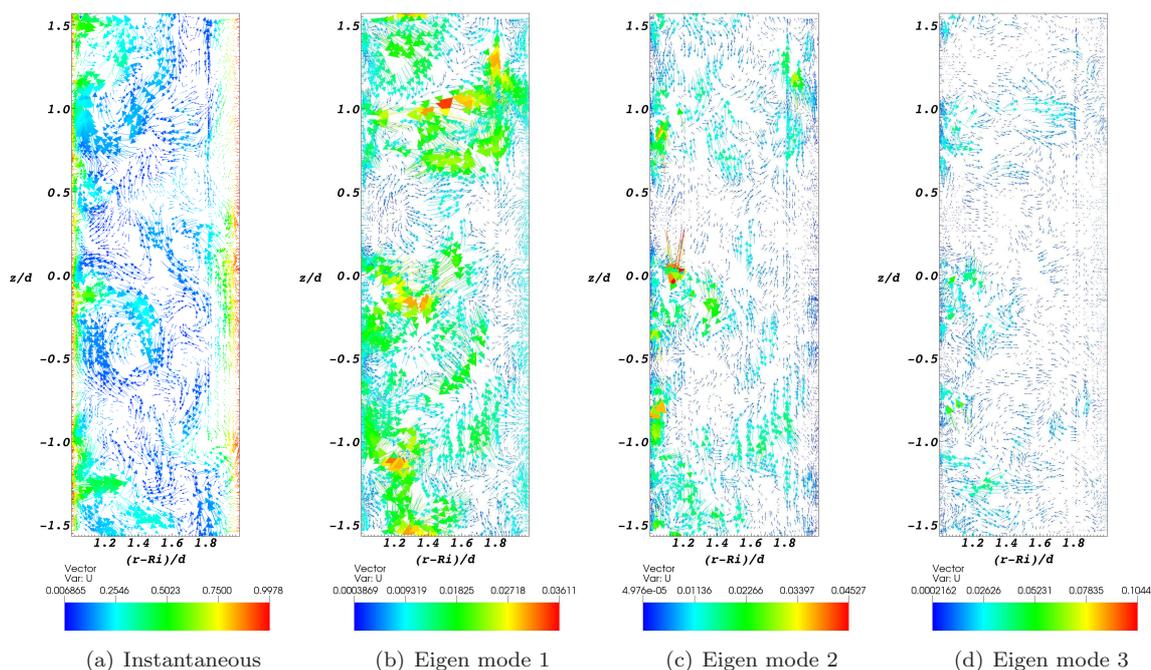


Figure 9: Instantaneous velocity vectors in a radial-axial plane at different Reynolds numbers

## References

- [1] <http://www.openfoam.com/>.
- [2] Y. Bazilevs and I. Akkerman. Large eddy simulation of turbulent taylor-couette flow using isogeometric analysis and the residual based variational multiscale method. *Journal of Computational Physics*, 229:3402–3414, 2010.
- [3] S. Dong. Turbulent flow between counter-rotating concentric cylinders: a direct numerical simulation study. *J. Fluid Mech.*, 615:371–399, 2008.
- [4] J. Jeong and F. Hussain. On the identification of a vortex. *J. Fluid Mech.*, 285:69–64, 1995.
- [5] D. K. Lilly. A proposed modification of the germano subgrid-scale closure method. *Physics of Fluids*, 633, 1992.
- [6] S. B. Pope. *Turbulent Flows*. Cambridge University Press, 2000.
- [7] P. J. Schmid. Dynamic mode decomposition of numerical and experimental data. *J. Fluid Mech.*, 656:5–28, 2010.
- [8] W. Schoppa and F. Hussain. Coherent structure dynamics in near-wall turbulence. *Fluid Dynamics Research*, 26:119–139, 2000.
- [9] J. Smagorinsky. General circulation experiments with primitive equations. *Monthly Weather Review*, 91:99–165, 1963.

# A Appendix

## A.1 Description of OpenFOAM CFD toolbox

OpenFOAM(Field Operation And Manipulation) is an object oriented code open source CFD software written in C++, released by OpenCFD Ltd. Over the years many users have contributed to the software with new ideas and extensions and this makes OpenFOAM a rich functionality filled toolbox. OpenFOAM consists of a C++ library, which is used to create applications. OpenFOAM is not a monolithic executable like many CFD softwares rather it is a set of various libraries which can be tailored to build specific applications. These applications consists of *solvers* and *utilities*. The main advantage of OpenFOAM is the ease with which one can create and customize solvers. The solver applications are written with OpenFOAM classes, which simplifies the syntax to resemble the partial differential equations that is being solved. To be able to make this happen the programming language needs to have object oriented properties such as inheritance, template classes, virtual functions, and operator overloading. For e.g. C++, Fortran90 are object oriented. To give an example of the capability of such a top-level objected oriented code, let us consider the momentum conservation equation

$$\frac{\partial \rho \mathbf{U}}{\partial t} + \nabla \cdot (\rho \mathbf{U} \mathbf{U}) - \nabla \cdot (\mu \nabla \mathbf{U}) = -\nabla p \quad (9)$$

Equation 9 can be implemented in OpenFOAM in a natural way as follows, as we can see this is pretty simple, and enables users to easily code up transport equations and build customized solvers.

```
solve
(
    fvm::ddt(rho, U)
+ fvm::div(phi, U)
- fvm::laplacian(mu, U)
==
- fvc::grad(p)
);
```

The solvers and utilities are controlled through the use of *dictionaries*. These are text files where specifications of the applications are accessed and controlled. For e.g. specifications such as discretisation method, start and end times, divergence evaluation scheme, turbulence model to be used, pressure corrector settings, and many other parameters are all controlled and accessed through these sets of dictionaries. OpenFOAM supports OpenMPI parallelization using domain decomposition technique. **decomposePar** utility decomposes the domain into as many processors as specified in the *decomposeParDict*.

A folder by name *constant* contains various dictionaries for specifying the mesh and other transport properties of the flow being solved. Where as the folder *system* contains the dictionaries related to solver settings and discretisation schemes etc... The folder by name *0* contains the initial conditions for all the primitive variables that are solved as well as the various boundary conditions on the zones/patches.

After the simulations are performed the utilities **reconstructPar** can be used to put together the decomposed domain into one and this can be latter used with **foamToTecplot360** utility for writing the output data in Tecplot360 format. Various other utilities also exist to convert the data to ones favorite postprocessor.

## A.2 Periodic Boundary Conditions in OpenFOAM

Here are steps to be followed for applying periodic boundary conditions in OpenFOAM when the mesh was generated in GAMBIT are described.

1. While meshing in GAMBIT, link the periodic faces and mesh one of them, the other one will get meshed automatically using links. Leave the *Reverse Orientation* option ON while linking the face zones. This is important as the face normals get flipped.

2. Save the two periodic face zones as two different patches in GAMBIT and export the volume mesh in FLUENT 5/6 format.
3. Using **fluentMeshToFoam** utility convert the exported mesh to OpenFOAM readable format.
4. Generate a **createPatchDict** dictionary in the system/ folder and combine the separate periodic faces zones into one patch and apply **cyclic** boundary condition on them. For e.g. to combine *periodic* and *shadow* which are two different linked periodic face zones into one cyclic patch *periodicShadow* the code would look like:

```
{
  name periodicShadow;
  type cyclic;
  constructFrom patches;
  patches (periodic shadow);
  set f0;
}
```

5. run **createPatch** utility which will combine the two periodic patches into one which can be used by OpenFOAM.
6. Move the newly created **polyMesh** folder inside a folder for e.g. **0.001** to constant/ folder  
This process should set up periodic zones properly for OpenFOAM1.7.1 to work.

### A.3 Coordinate Transformation of Turbulent Statistics

OpenFOAM solves for velocity components in Cartesian coordinates, but the literature on CRTC uses turbulence statistics in Cylindrical coordinates because the domain is a cylindrical annulus. Hence in order to convert various turbulent statistics from Cartesian to Cylindrical the following equations were used whose derivations are included here for the sake of completeness.

Velocity components can be transformed to Cylindrical coordinates using the following equations.

$$U_r = +U \cos \theta + V \sin \theta \quad (10)$$

$$U_\theta = -U \sin \theta + V \cos \theta \quad (11)$$

Where  $U, V$  are the velocity components in Cartesian coordinates and  $U_r, U_\theta$  are the radial and circumferential velocity components in Cylindrical coordinates. Decomposing the velocity vector  $\mathbf{U} = (U\hat{i}, V\hat{j}, W\hat{k})$  into the sum of a mean value  $\langle \mathbf{U} \rangle$  and a fluctuation component  $\mathbf{u}' = (u'\hat{i}, v'\hat{j}, w'\hat{k}) = (u'_r\hat{e}_r, u'_\theta\hat{e}_\theta, w'\hat{e}_z)$  we have

$$\mathbf{u}' = \mathbf{U} - \langle \mathbf{U} \rangle \quad (12)$$

For e.g. the Reynolds stress in Cylindrical coordinates  $\langle u'_r u'_\theta \rangle$  can be obtained from the Cartesian Reynolds stress components using the following expression.

$$\langle u'_r u'_\theta \rangle = \frac{(\langle v'v' \rangle - \langle u'u' \rangle)}{2} \sin 2\theta + \langle u'v' \rangle \cos 2\theta \quad (13)$$

The above equation can be derived by following the definition of  $\langle u'_r u'_\theta \rangle$  and substituting the respective quantities from Equations 10, 11 as follows.

$$\begin{aligned} \langle u'_r u'_\theta \rangle &= \langle (+u' \cos \theta + v' \sin \theta)(-u' \sin \theta + v' \cos \theta) \rangle \\ &= \langle -u'u' \sin \theta \cos \theta + u'v' \cos^2 \theta - v'u' \sin^2 \theta + v'v' \sin \theta \cos \theta \rangle \\ \langle u'_r u'_\theta \rangle &= \frac{(\langle v'v' \rangle - \langle u'u' \rangle)}{2} \sin 2\theta + \langle u'v' \rangle \cos 2\theta \end{aligned}$$

Similarly we can derive root mean square fluctuation of radial and azimuthal velocities as follows.

$$u'_r rms = \sqrt{\langle u'u' \rangle \cos^2 \theta + \langle v'v' \rangle \sin^2 \theta + \langle u'v' \rangle \sin 2\theta} \quad (14)$$

$$u'_\theta rms = \sqrt{\langle u'u' \rangle \sin^2 \theta + \langle v'v' \rangle \cos^2 \theta - \langle u'v' \rangle \sin 2\theta} \quad (15)$$

These equations can be easily coded up in Tecplot for visualization purposes and the same is done in the present work.