NUMERICAL SIMULATION OF STIRRED TANK REACTOR

Internship Report

CFD-FOSSEE Team

Indian Institute of Technology, Bombay

Prepared by

Maharghya Dyuti Das

Department of Chemical Engineering

University of Calcutta

Under the supervision of

Prof. Manaswita Bose

Department of Energy Science and Engineering

INDIAN INSTITUTE OF TECHNOLOGY, BOMBAY

Acknowledgement

The following report was created as a part of FOSSEE summer internship fellowship. I would like to thank FOSSEE, Indian Institute of Technology, Bombay for giving me this opportunity.

I would like to thank my project guide Prof. Manaswita Bose. She has given time for many hour-long video conferencing in spite of her busy schedules. She has given me direction and completely shaped the project work in the present form. The drawbacks of this project work whatsoever is entirely due to me.

I would like to thank my mentor Mr. Ashutosh for the patience and support in explaining OpenFOAM and introducing me to it. It was indeed difficult to understand MRF and implement in OpenFOAM with the mesh generated in freeCAD. However, his never-give-up enthusiasm has helped me in overcoming the intial difficult high tides and sailing through this project in the later phase. I owe a lot to him.

I would like to extend my gratitude to Prof. J. Muralidharan for giving me the inputs during the weekly group meeting. Her constant encouragement has helped me to improve my presentation gradually over the time. I am grateful to her.

I would like to thank Head, Department of Chemical Engineering, University of Calcutta for allowing me to carry out this summer internship in academic institute of IIT Bombay.

Maharghya Dyuti Das

Maharghya Dyuti Das
Department of Chemical Engineering
University of Calcutta
Date: August 7, 2022

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In chemical industry, stirred-tank reactors are frequently used as a vessel where chemical reactants are efficiently mixed to obtain the desired products. The reactants are of different types such that the impeller, their orientation, number and speed will also of wide variation. The study of mixing property, velocity distribution, residence time etc. are of complex nature because of the geometry and the orientation.

Harvey III et al. **?**] have carried out experimental study using laser Doppler velocimetry (LDV). They have carried out a numerical study and have presented the comparison of these two results. Later, Harvey and Rogers **?**] have studied the problem and presented further results. Later on **?**] have modified their computational methodology. Other recent relevant results may be obtained in these references: **?**],**?**],**?**],**?**].

1.1 Aim

1. Here, the fluid flow in a stirred tank reactor is to be solved using OpenFOAM software. A casefile is to be prepared and the accuracy of the solution is to be checked. After doing literature sirvey, it has been found that the work of Harvey III et al.**?**] is appropriate for this purpose. The geometrical and thermophysical data are obtained from this paper to solve the problem

2. After achieving the first objective, the problem on stirrec tank reactor given by FOSSEE is to be solved.

SubFiles/images/Screenshot from 2022-06-22

Figure 1.1: Moving impeller within the tank

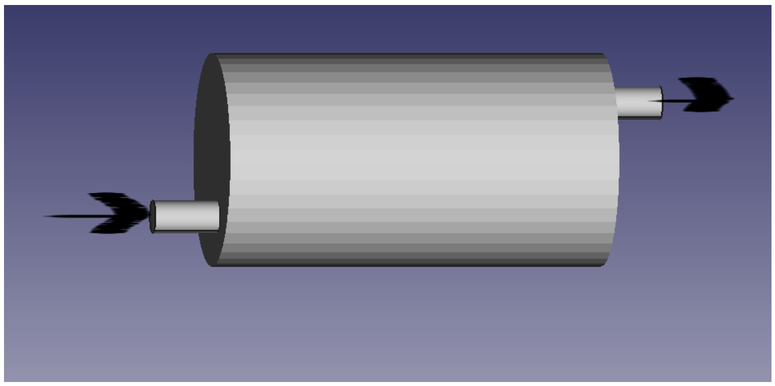


Figure 1.2: Stirred tank reactor with inlet and outlet

1.2 Theory

Here we cannot use stationary meshes which we see commonly used because in this case the impeller is moving. So we need to use the concept of moving meshes where we solve by moving the mesh before the start of each timestep and we solve the unsteady Navier-Stokes equation to get the results. We need to take small time steps (Fig. 1.1) so the process becomes very slow.

This process becomes very expensive. So other alternative was sorted which is single refrence frame (SRF). But this process was also not feasible as here the whole geometry should rotate but thats not the case as we cannot rotate the whole geometry as we have inlet and outlet. So now we have to use Multiple Refrence Frame (MRF) (Fig. 1.2) to solve here we take a one stationary zone and and one or more than one rotating zones.

Incompressible Navier-Stokes Equation:

$$\frac{\partial U}{\partial t} + \nabla.(UU) = -\frac{1}{\rho}\nabla p + \nabla.(\nu \nabla U) \tag{1.1}$$

Hence, acceleration in the relative frame with relative velocity

$$\frac{Du_I}{Dt} = \frac{Du_R}{Dt} + \frac{d\Omega\Omega}{dt} \times r + 2\Omega \times u_R + \Omega \times \Omega \times r \tag{1.2}$$

(refer Appendix A Eq. A.9)

The steady-state Navier-Stokes equation in the relative frame of reference:

$$\nabla.(U_r U_r) = -\nabla p + \nabla.(\nu \nabla U_r) - \underbrace{2\Omega \times u_R}_{coriolis} - \underbrace{\Omega \times \Omega \times r}_{centrifugal} \tag{1.3}$$

(refer Appendix B)

Navier-Stokes equations in the relative frame with absolute velocity

$$\nabla.(U U_r) = -\nabla p + \nabla.(\nu \nabla U) - \underbrace{\Omega \times U}_{source} \tag{1.4}$$

(refer Appendix C, and **?**]

In Multiple Reference Frame (MRF) approach, the global velocity U is taken to solve the Navier-Stokes equation. We now focus on the convection term as the convection term contains relative velocity and global velocity but we have to make it global velocity only.

So now we integrate the Navier-Stokes equation on a control volume, but we will focus on the convection term, $\int_V [\nabla.(UU)]dV$.

Using Gauss divergence theorem:

$$\int_V [\nabla.(UU)]dV = \int_S [U(U.\hat{n})]dS \tag{1.5}$$

Now, splitting the integral over the surface and subscript f is use to denote the value at the face centre.

$$\int_S [U(U.\hat{n})]dS = \sum_{faces} \int_S [U(U.\hat{n})]dS = \sum_{faces} U_f(U_f.\hat{n}_f)A_f \tag{1.6}$$

Moving on to MRF method, we have:

$$\sum_{faces} U_f(U_r.\hat{n}_f)A_f = \sum_{faces} U_f([U_f - \Omega \times r].\hat{n}_f)A_f \tag{1.7}$$

$$\sum_{faces} U_f(U_r.\hat{n}_f)A_f = \underbrace{\sum_{faces} U_f(U_f.\hat{n}_f)A_f}_{Original} - \underbrace{\sum_{faces} U_f((-\Omega \times r).\hat{n}_f)A_f}_{Fluxcorrection} \tag{1.8}$$

(Youtube **?**]

Now in MRF we have new volume flux correction and a source term.

Chapter 2

Problem Statement

The objective was to simulate the 45 degree pitched blade impeller and the case is laminar flow. Then we will have to compare the results of the radial velocities, tangential velocities and axial velocities with the results of Harvey III et al. **?**]. The simulations have been run for three different cases, viz. dimensional case, non-dimensional case and then we have added baffle and run the simulation.

2.1 Schematic Diagram

The schematic diagram of the stirred tank reactor considered by **?**] is shown in Fig. 2.1.

The geometric dimensions are as follows:
Diameter of tank = 14.5cm
Height of tank = 14.5cm
Diameter of shaft = 0.8cm
Baffle thickness = 1.25cm
Radius of impeller blade=2.54cm
Height of impeller blade = 0.1cm
Height of impeller from tank=6.67cm

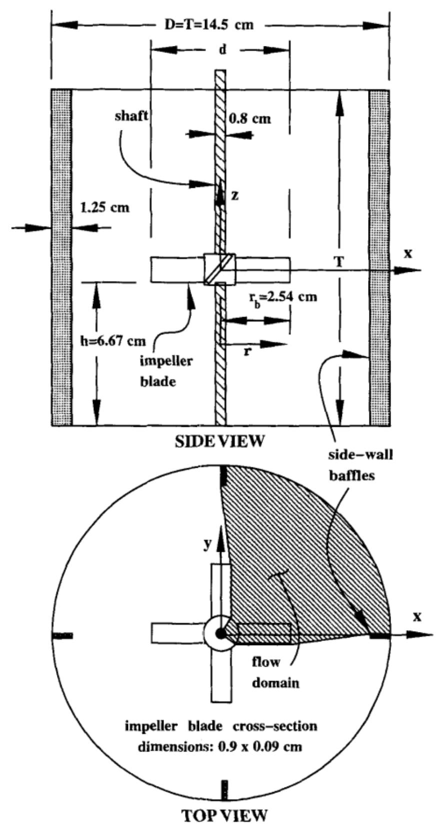


Figure 2.1: Mixing tank geometry with 45°pitched blade impellers **?**]

2.2 Creating an stl File

We are using FreeCAD 0.20 version to create the stl File. First we open the FreeCAD then we select **part** and then we click on the **Create new.** Now we first create the blade; now we click on the **cube** utility (Ref Fig. 2.2).

Figure 2.2: Cube utility

And then we use the **property editor** (Ref. Fig. 2.3) to change the length, height and width according to the given data. Then we save the file with file name blade.

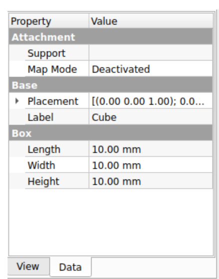


Figure 2.3: Property editor

Finally the blade looks like Fig. 2.4.

Figure 2.4: Blade

Now in the same way we create the shaft (Ref. Fig. 2.5) and the hub (Ref. Fig. 2.6). And now we save the two files.

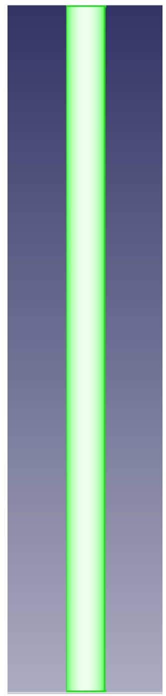


Figure 2.5: Shaft

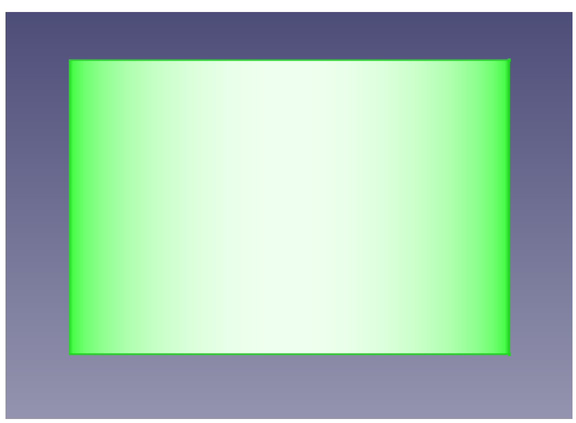


Figure 2.6: Hub

Now we use the **open** to and go to the **Open Document** and go the location where the file is saved and open it and now in the **Combo View** section but now we drag and drop hub into the impeller section which have the shaft.

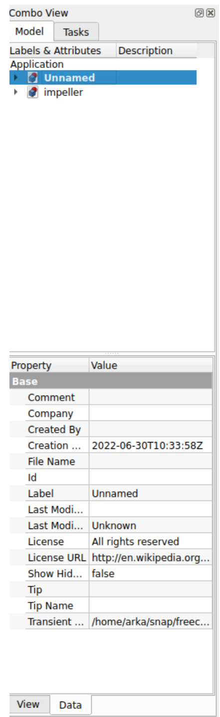


Figure 2.7: Combo view

Now we open the **property panel** (Ref. Fig. 2.8).

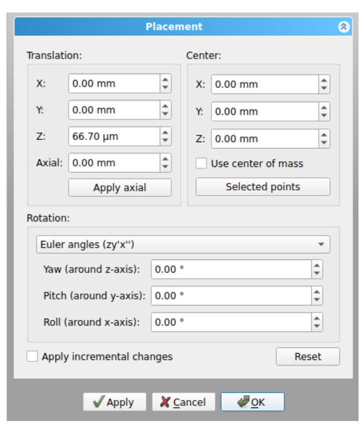


Figure 2.8: property panel

And now by giving the translational z value for the moving the hub to the allocated value.

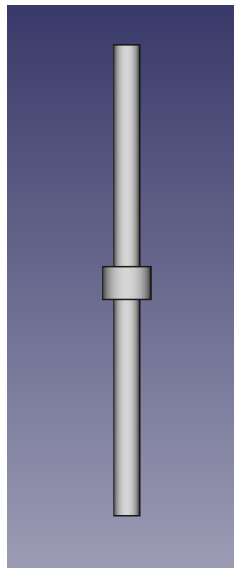


Figure 2.9: Shaft with hub

Now then move forward to add blade first in **Combo view** we add and then similarly we drag and drop down and then we open the **property panel** we add now we add the translational z and then we move to eular angles we set the required roll angles and we similarly add the three remaining blade and set all the data appropiately in **property panel** (ref. Fig. 2.10) all the blades set appropiately.

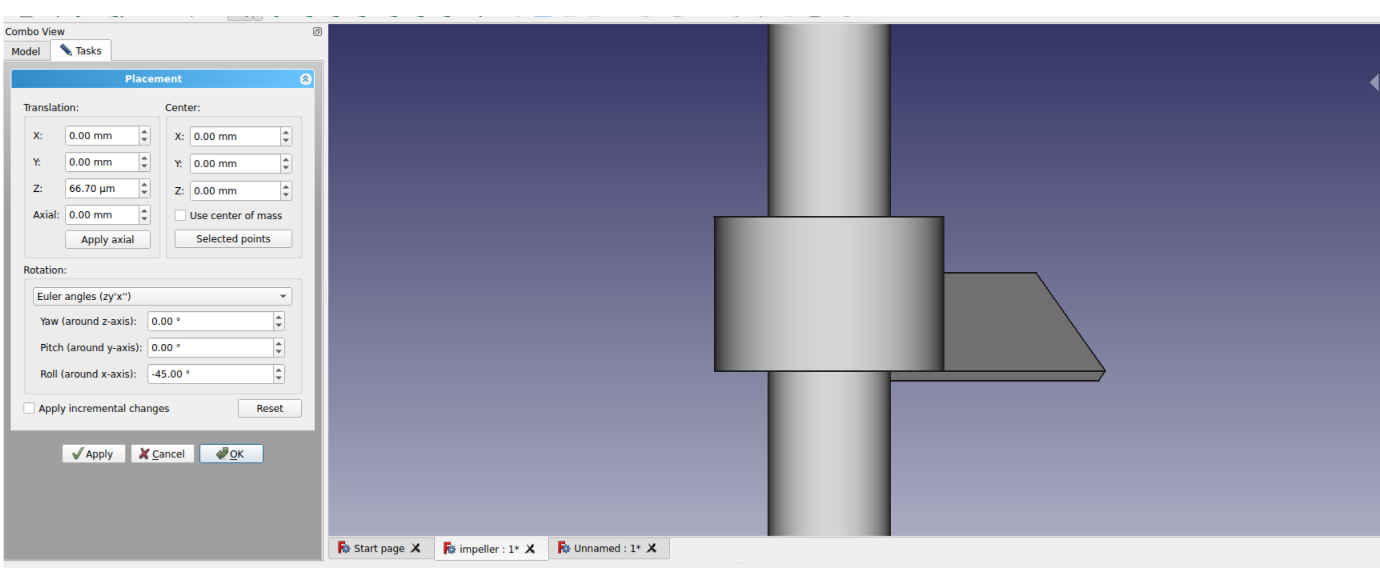


Figure 2.10: property panel for blade

And then we add the tank which we have created **cylinder** utility and radius is set (ref. Fig. 2.11).

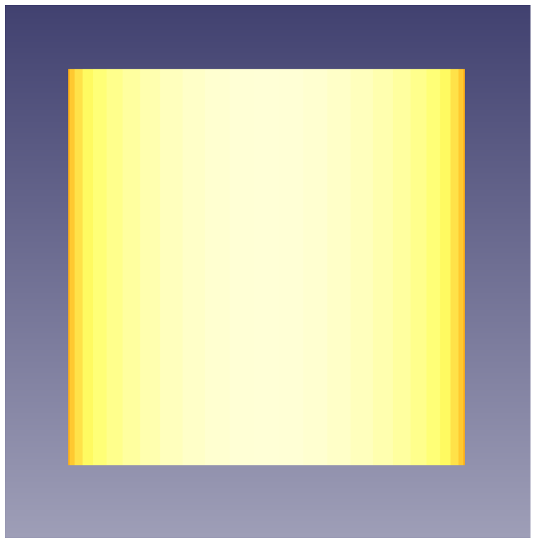


Figure 2.11: tank

And then the now go to the **file→export** and then we add the **files of type** as **Stl mesh** and then we go to the desired location and click the **save** button.

2.3 OpenFoam Setup

Tutorial case

We are using the tutorial case for the motorbike.

File location is /opt/openfoam9/tutorials/incompressible/simpleFoam/motorBike.

2.3.1 Folder Structure

We create the file as **harvey-FOAMdimen** and it is created in the **run** directory. All the files which are there are given in Table 2.1.

0	constant	system
eplison	trisurface	blockMeshDict
k	momentun transport	controlDict
nut	MRFProperties	fvSchemes
p	transportProperties	fvSolution
U	-	meshQualityDict
-	-	snappyHexMeshDict
-	-	surfaceFeaturesDict

Table 2.1: harvey-FOAMdimen directory

2.3.2 Solver

We use OpenFOAM which is a free, open source CFD software which is developed primarily by OpenCFD Ltd. It has many solvers. But in our case we use simpleFoam which is a steady-state solver which is used to solve for incompressible, turbulent flow, using the SIMPLE algorithm which stands for Semi-Implicit Method for Pressure Linked Equations, originally developed by **?**]. Here we have pressure velocity coupling equation where we already know the velocity which we found out by the continuity equation and hence we found out the pressure. **?**]

2.3.3 Laminar Model

For this we check the Reynolds number .

We have,

$$Re = \frac{Characteristic\, velocity \times characteristic\, length}{kinematic\, viscosity}$$

Here, we have characteristic lenght=blade heigth(r_b), characteristic velocity=angular velocity (ω) $\times$ blade length r_b)

Therefore,

$$Re = \frac{\omega \times r_b \times r_b}{v} = \frac{\omega \times r_b^2}{v}$$

By puting the values, we found the Reynolds number

Re = 33

Hence, the flow is laminar.

2.3.4 OpenFOAM Mesh

blockMeshDict

It creates a background mesh of hexahedral cells that fills the entie region within the boundary.

Vertex number : It follows a list of vertex numbers ordered in a manner.

Number of cells : t gives the number of celss in each x, y and z direction.

simpleGrading : It specifies the uniform expansions in the local x, y and z direction with the 3 expansion ratio.

surfaceFeaturesDict

It is basically used to make edgeMesh that can be extracted from trisurface file and also it controls the included angle which specifies that normal of two adjacent angles have less value than that of the included angle.**?**]

snappyHexMesh

snappyHexMesh is used for creating 3D mesh tool. It is situated in system directory.

So now we have **castellatedMesh** on the base mesh is created inside the bounds of *Base Mesh Box*. Basically here first we have created a *Base Mesh Box*. Then we do refinement on one or more specific region on the the specific edge and surface refinements and then spilt in the vicinity of the body surfaces. Here the region is specified i.e. the name of the region itself which is basically the stl surface and the refinement level is also specified in the sub-dictionary i.e. **refinementSurfaces**.

The process of cell splitting over then the process of cell removal begins. Here the region which the cells are retained is basically specified by a location vector within that region. It is specified by the **locationInMesh** keyword in **castellatedMeshcontrols**. We have locted the **locationInMesh** inside the geometry hence the cells are retained inside the geometry region.

Also we have further splitted the cells inside the MRF region. This is done by using the **refinementRegions** subdictionary in the **castellatedMeshControls** were we have used the feature inside ehich refines inside the volume region.

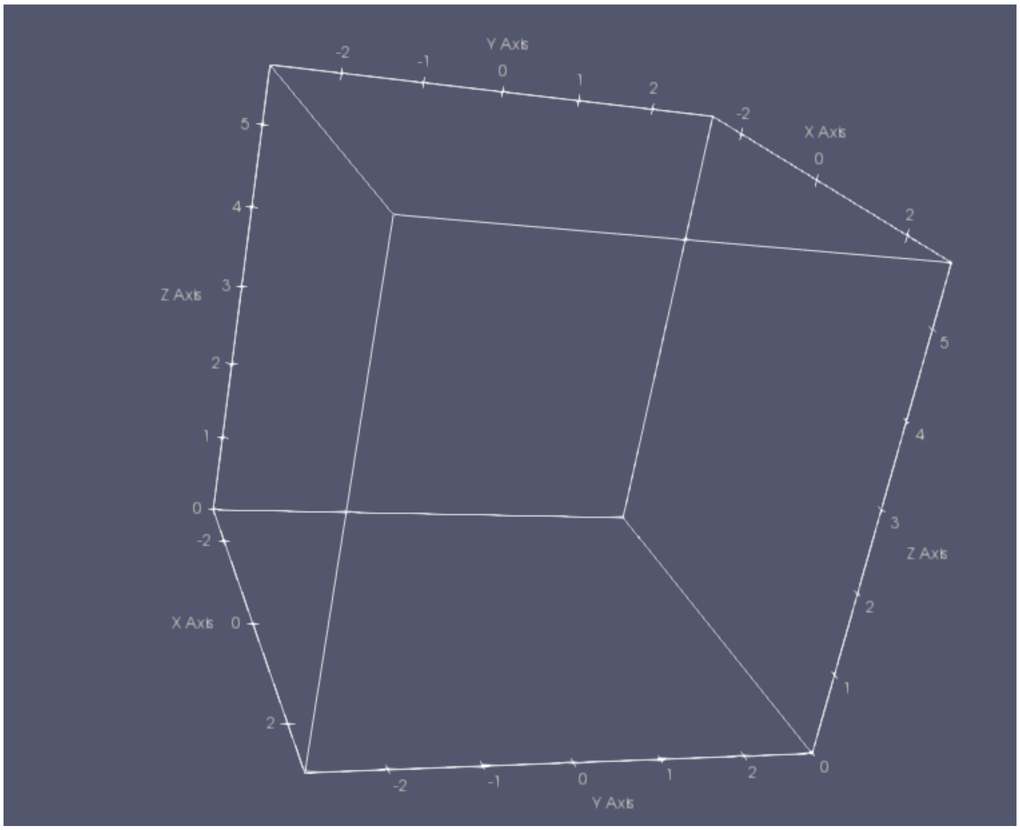
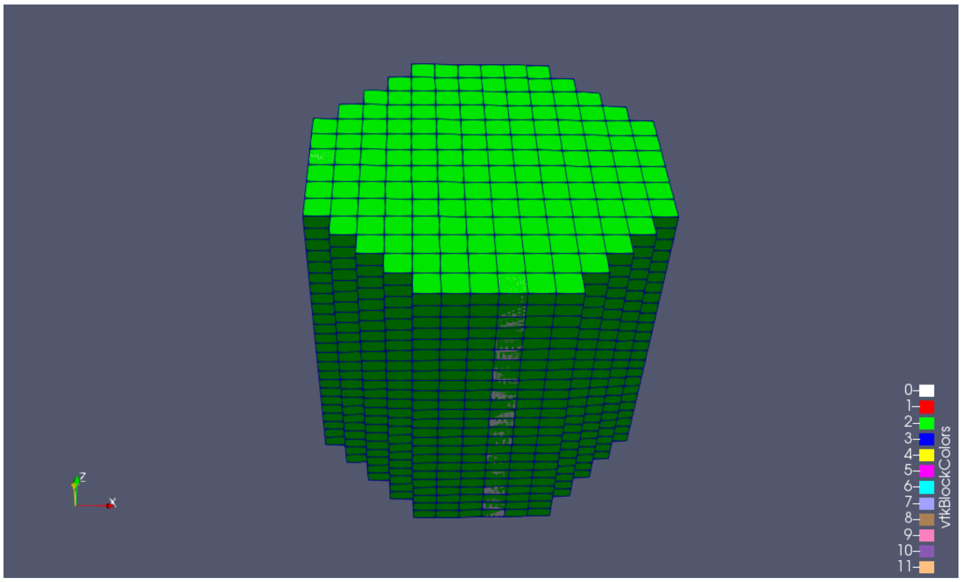


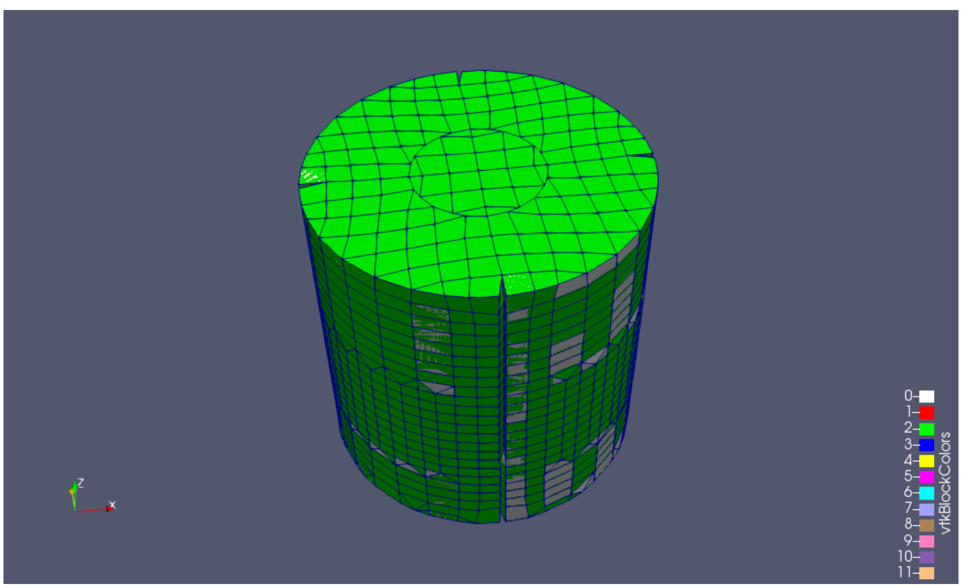
Figure 2.12: Using blockMesh

Now in next we move to snap were basically we smoothen the suface. Here basically the process for mesh alignment with the geometry is done. Here basically we control the settings by using the **snapControls** subdictionary (refer Fig. 2.13). Here only we change the **nSolverIter** where it specifies the number of mesh displacement relaxation iterations.

Also there contains the **addLayers** which we have turned off.

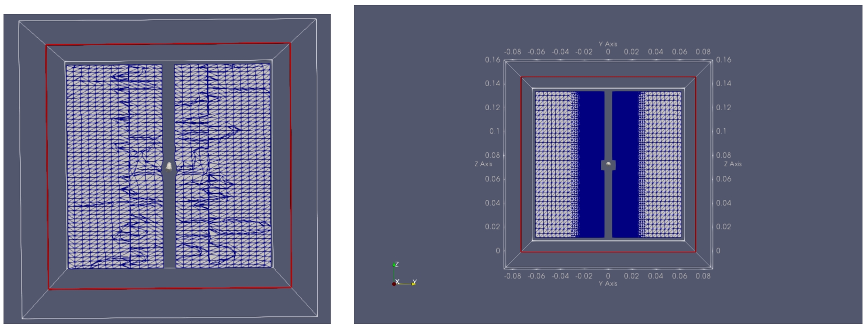


(a) Without snapControl



(b) With snapControl

Figure 2.13: Function of snapControl



(a) (b)

Figure 2.14: (a)slice of internal mesh with less refinement(b)actual slice of internal mesh which we have used

fvSchemes & fvSolution

fvSchemes and fvSolution both are under the system directory. fvSchemes sets such as derivatives of equations, i.e. numerical schemes for term that are calculated in simulation. fvSolution we have preconditioned conjugate gradient solvers(**GAMG**), Smooth solvers(**GaussSeidel**) also number of sweeps(**nSweeps**) is specified and the **tolerance** is also set 1e-07. **relaxationFactors** controls under-relaxation which improves the stability of the computation. Also **SIMPLE** here stands for simle algorithms.**?**]**?**]

2.3.5 controlDict

It is a case control file which is situated in system directory. The controlDict dictionary sets input parameters essential for the creation of the database. The startime is set to 0 s. The endtime is set as 1000 s and the interval is set to 50 s. And $\Delta t = 1$ is also set.

2.3.6 Boundary conditions

The boundary conditions we use noslip boundary condition.

2.3.7 Physical properties

The fluid which we used is silicone oil. All its properties is given in the table 2.2 given below.

Material	Silicone Oil
RPM	100
density	1049 kg/m^3
Dynamic viscosity	211 cP
Kinematic viscosity	0.0002 cSt

Table 2.2: Physical properties

2.4 Post-processing

We now open the terminal by right clicking ang clicking the **open in terminal** option. Then in terminal we run the following command **blockMesh.** Then we run **surfaceFeatures.** Then we run the **snappyHexMesh -overwrite**. Then finally we run **simpleFoam**. Then we run **paraFoam**.

Then we use the **calculator** utility and change the attribute data to **point data** and then we type the formula for conversion of velocity in cartesian coordinates to cylindrical coordinates.

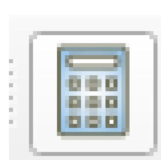


Figure 2.15: Calculator utility

Then wee click on the **cube** utility(Ref Fig. 2.2).The radial velocity $v_r = \frac{xV_x + yV_y}{\sqrt{x^2+y^2}}$ and then the tangential velocity $v_\theta = \frac{xV_y - yV_x}{\sqrt{x^2+y^2}}$.

Then we type the both formula here Fig. 2.16 and the Array name is given u_radial and u_theta.

Then we go to **View–>Data Analysis–>Plot over line** there we set point 1 and point 2 all the variables are stored in a csv line which the we use for plotting. We then save the data which we plotted. Then we use the engage digitizer to get the

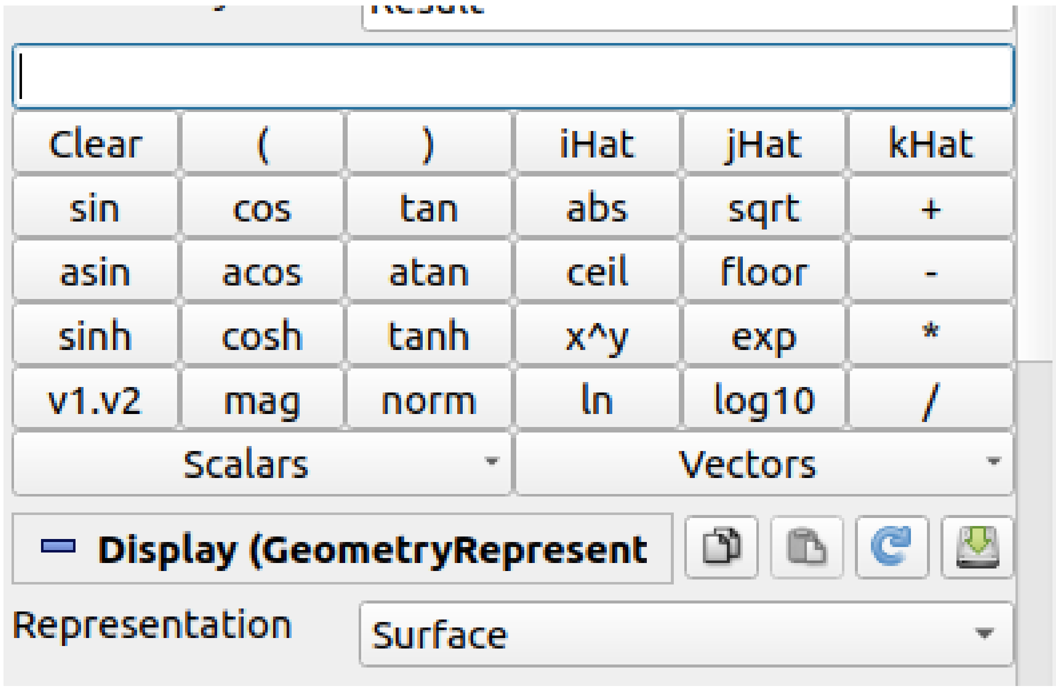


Figure 2.16: typing the two formula here

same corresponding from **?**]. Then we use **gnuplot** to compare the corresponding both data and then we save it as png file.

Chapter 3

Results comparison with ?] data

?] have presented their results in non-dimensional form. The velocities are scaled with the impeller blatye tip velocity and the lengths are scaled based in the imbeller blade height. The comparisons of radial, axial and tangential velocities are presented at four axial locations viz. z/r_b = 1.5433 (well below the impeller, location **c**), z/r_b = 2.5276 (close below the blade, location **h**), z/r_b = 3.9055 (above the impeller, location **m**) and z/r_b = 4.4882 (farthest from impeller, location **p**) (refer Fig. 3.1).

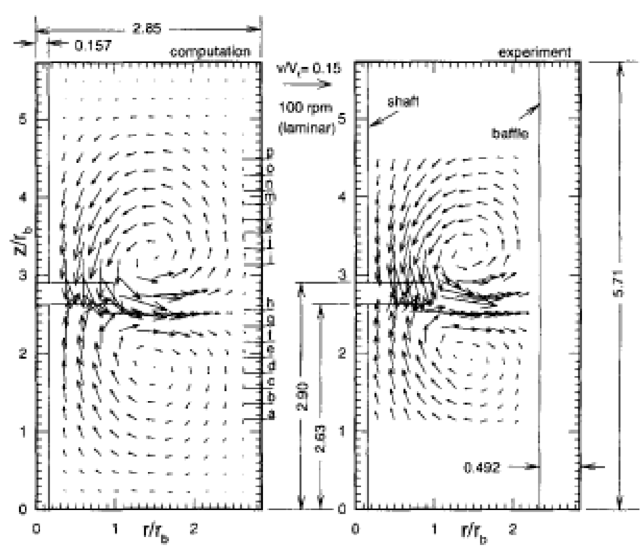
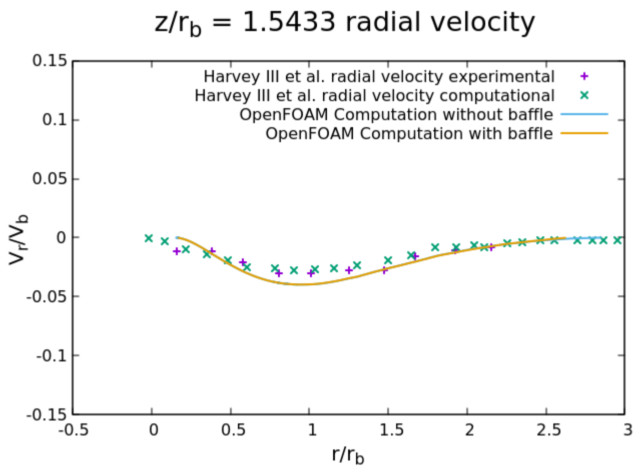


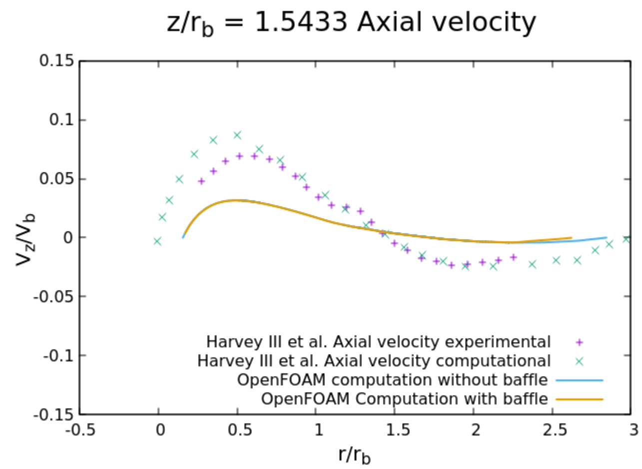
Figure 3.1: Experimental locations of **?**]

3.1 z/r_b = 1.5433 (well below the impeller, location c)

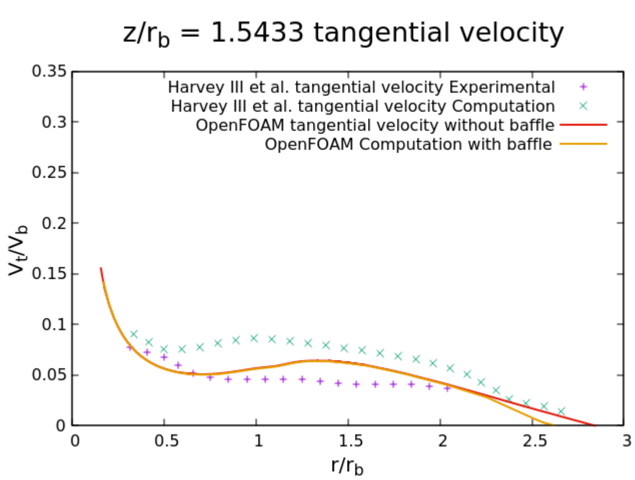
The comparison of radial, axial and tangential velocities are shown in Fig. 3.2 (a), (b) and (c) respectively. It is observed that the results are in fairly good agreement. It is to be noted that near the vessel surface, the velocity for the case with baffle is becoming zero before those of cases for without baffle. this is happening because of the the no-slip boundary condition.



(a) Comparison of radial velocity



(b) Comparison of axial velocity

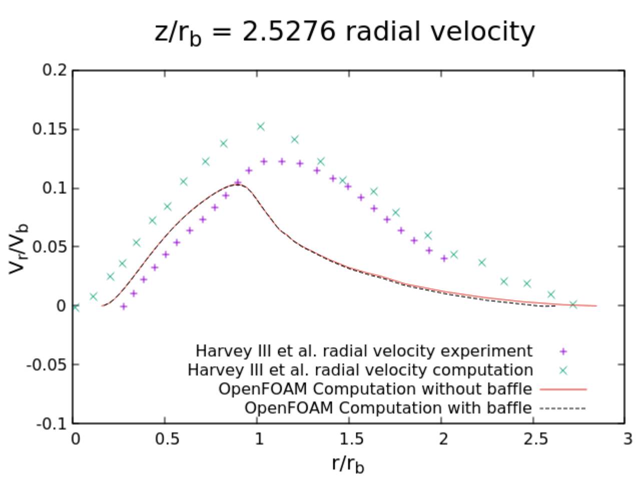


(c) Comparison of tangential velocity

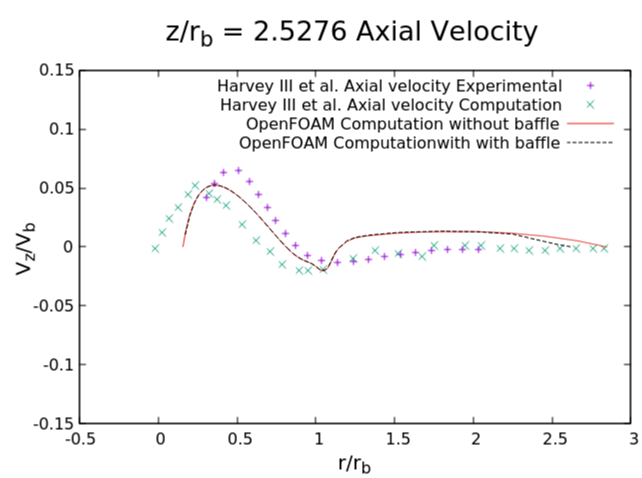
Figure 3.2: Comparison of velocities at location **c**, $z/r_b = 1.5433$

3.2 z/r_b = 2.5276 (just below the impeller, location h)

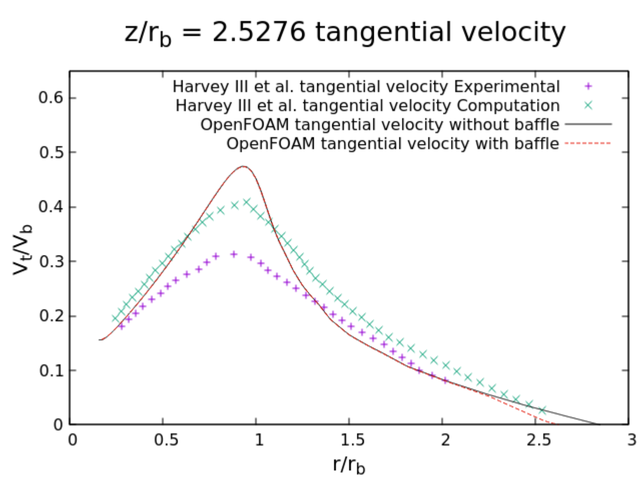
The comparison for those three velocoties at location h is shown in Fig. 3.3 (a), (b) and (c). This location is close below the pitched-impeller where the rotational activity is maximum. It may be noted that the present computation is able to reproduce the results with good amount of closeness.



(a) Comparison of radial velocity



(b) Comparison of axial velocity

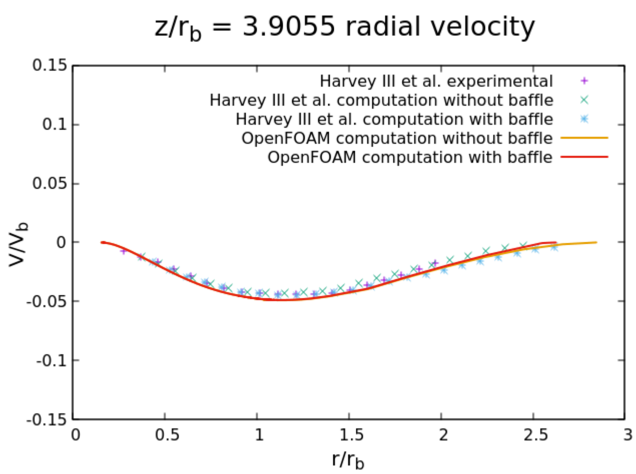


(c) Comparison of tangential velocity

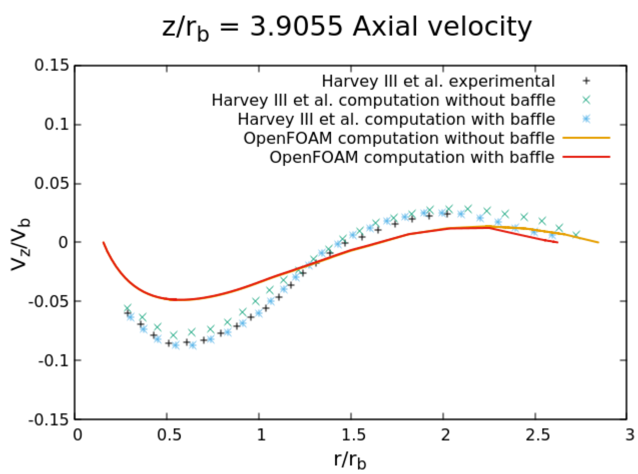
Figure 3.3: Comparison of velocities at location h, $z/r_b = 2.5276$

3.3 z/r_b = 3.9055 (fairly above the impeller, location m)

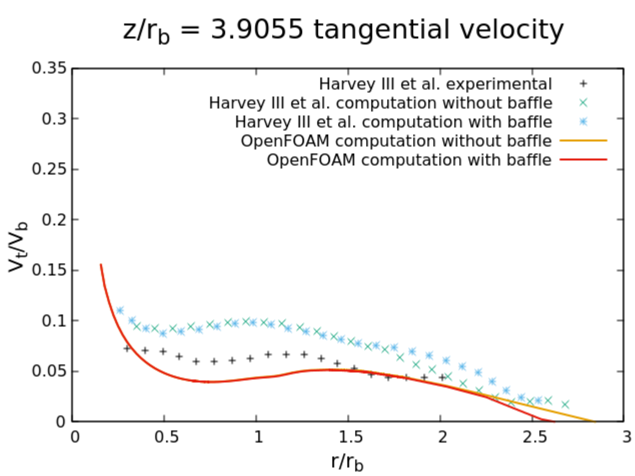
The comparison of velocities at location m is shown in Fig. 3.4 (a), (b) and (c). This location is up above the rotating blade, a little far away. Here, the rotational effect on the fluid is getting dispersed. The results obtained and presented show a good comparison with those of **?**].



(a) Comparison of radial velocity



(b) Comparison of axial velocity

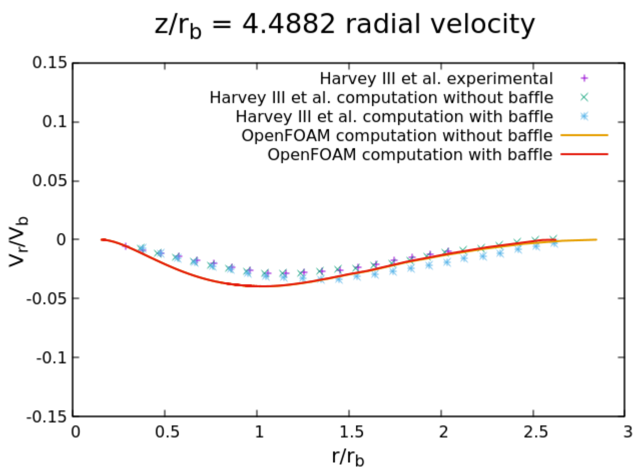


(c) Comparison of tangential velocity

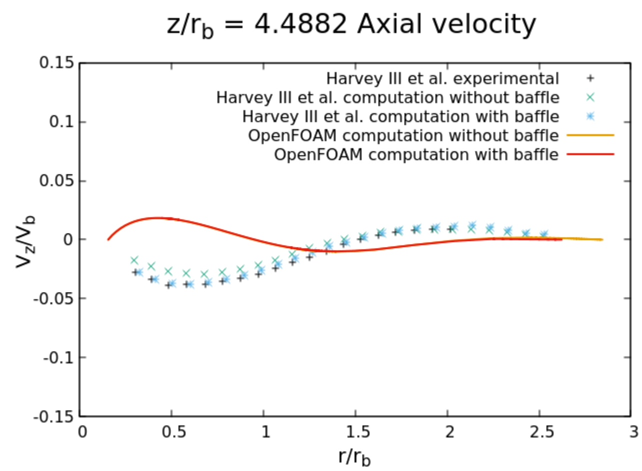
Figure 3.4: Comparison of velocities at location c, $z/r_b = 3.9055$

3.4 z/r_b = 4.4882 (well above the impeller, location p)

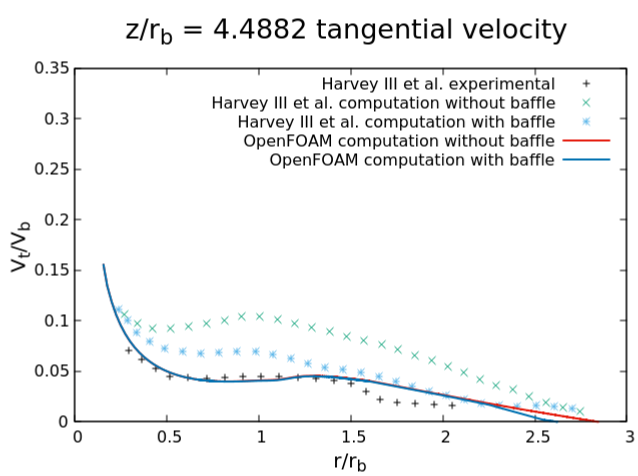
The location p is the last one and the farthest one from the impeller where **?**] have taken their measurements. It is to be noted that the results are in good agreement.



(a) Comparison of radial velocity



(b) Comparison of axial velocity



(c) Comparison of tangential velocity

Figure 3.5: Comparison of velocities at location c, $z/r_b = 4.4882$

Chapter 4

Results of the problem given by FOSSEE

Having done the validation work, we now focus on the stirred tank reactor problem given by FOSSEE.

4.1 Geometry

A two-bladed rushton impeller is shown in Fig. 4.1.

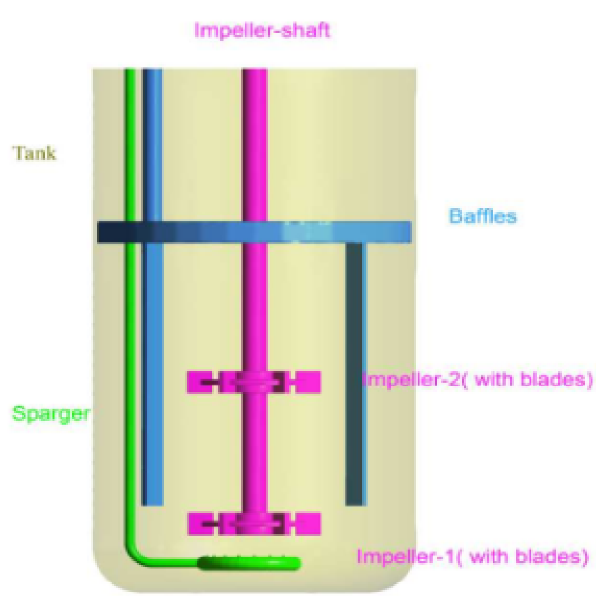


Figure 4.1: Two-bladed Rushton impeller

The geometrical dimensions are given in Fig. 4.2:

Tank Specifications:

Design Parameter	Value
Vessel Diameter	0.13 m
Vessel Height	0.24 m

(a) Vessel Dimension

Stirrer specifications:

Design Parameter	Value
Type	Rushton
No. of Impellers	2
No. of blades	4 (12 in later runs)
Impeller Diameter	0.0500 m
Blade height	0.0100 m
Blade length	0.0125 m
Shaft thickness	0.0100 m

(b) Impeller (Stirrer) Dimension

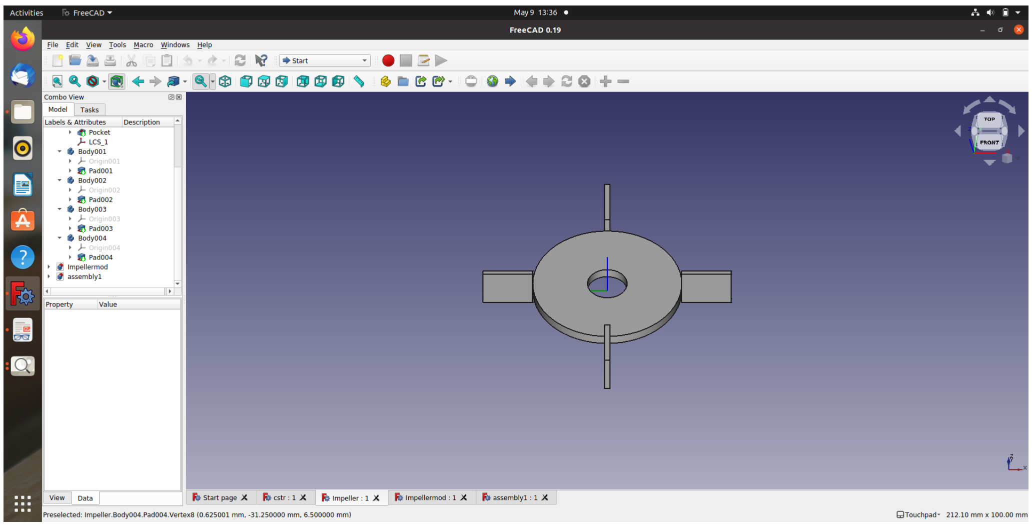
Clearance (Impeller to tank floor)	0.0325 m
Distance between impellers	0.0650 m
Hub thickness	0.0025 m
Hub diameter	0.0375 m

(c) Hub Dimension

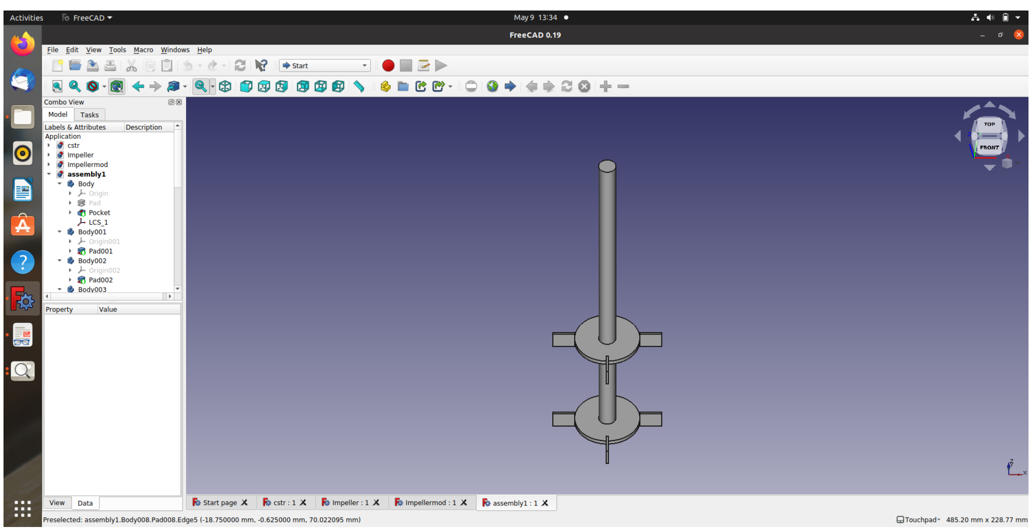
Figure 4.2: Various dimensions of the stirred tank reactor

4.2 Creation of geometry

Geometry has been created in FreeCAD. Fig. 4.1 shows the two-bladed Rushton impeller.



(a) Rushton impeller



(b) Rushton two impellers

Figure 4.3: Rushton impeller geometry in FreeCAD

The vessel with impeller is shown in Fig. 4.4

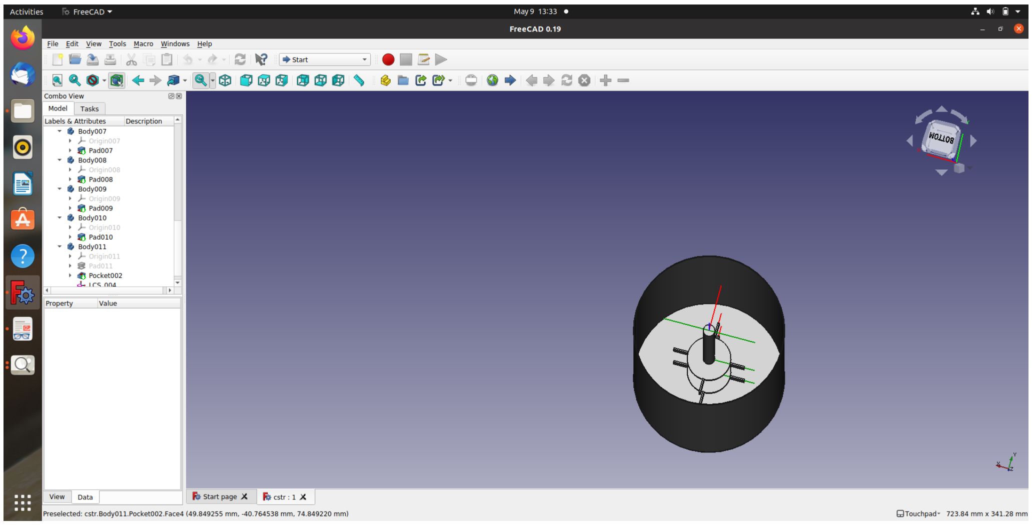
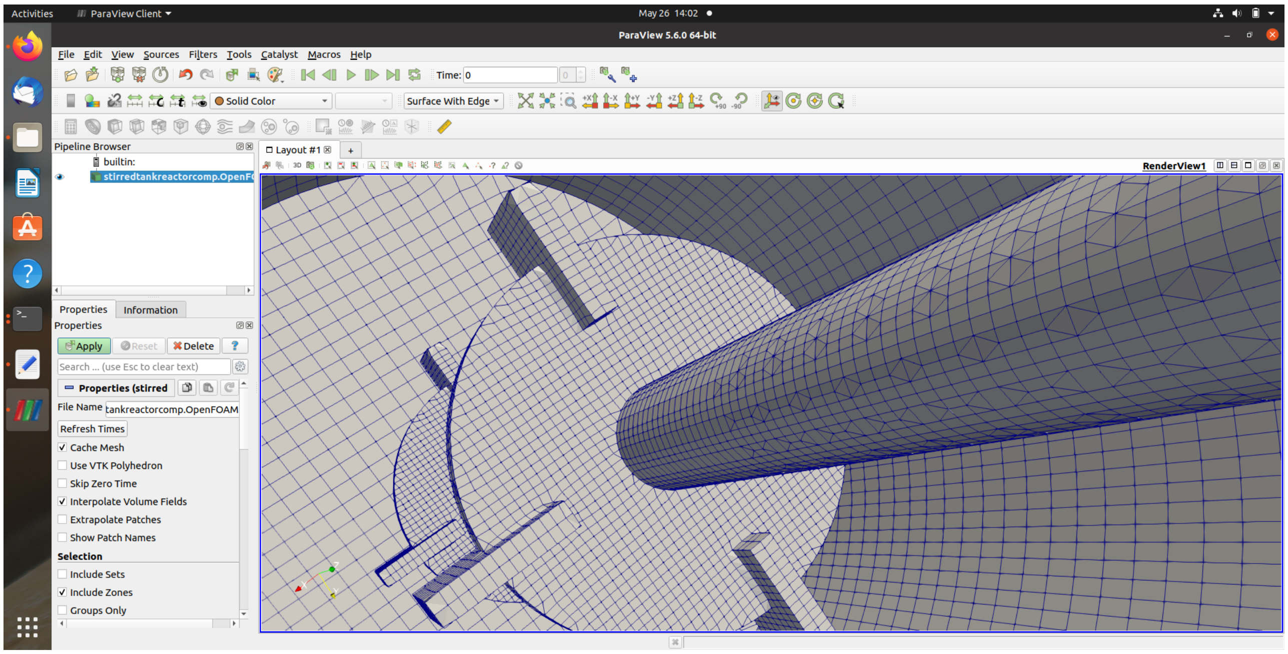


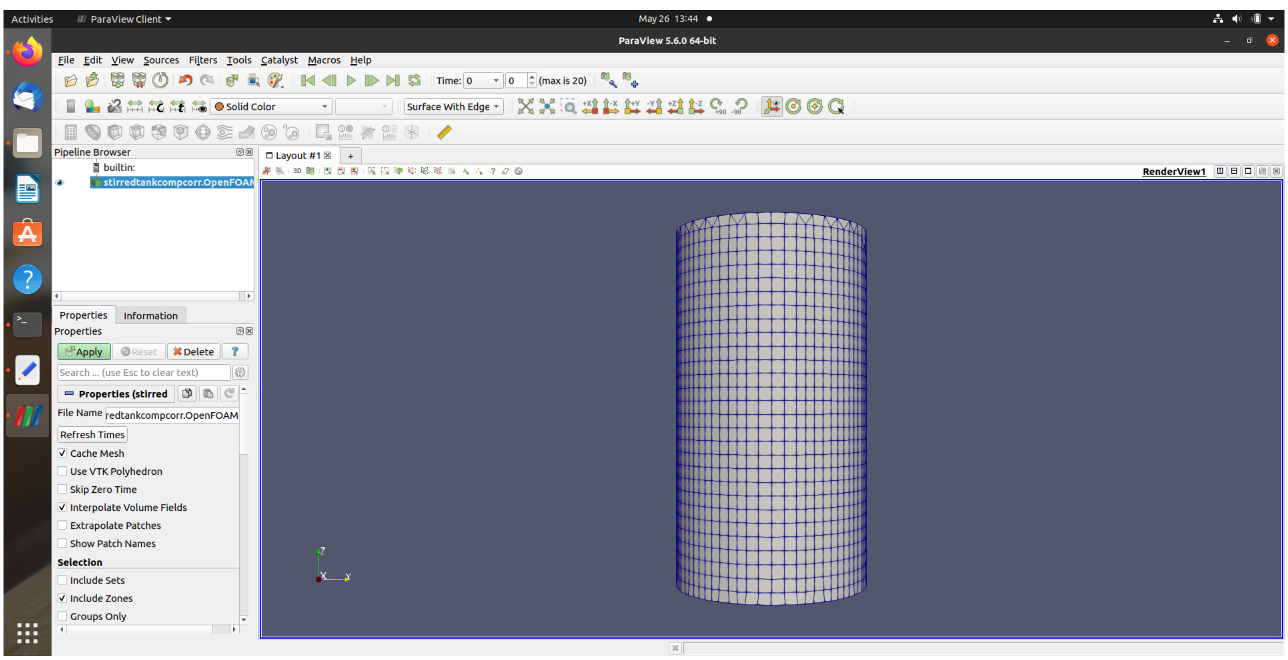
Figure 4.4: Vessel with Rushton impeller

4.3 Meshing

The stl files are imported in OpenFOAM and meshing is done using snappyHexMesh. Figure 4.5 shows the meshing thus done.



(a) Two impeller



(b) Vessel of the problem

Figure 4.5: Meshing with snappyHexMesh

4.4 Fluid Property and Reynolds number

The fluid property is given in Table 4.1

Fluid	Water	Remarks
Density(ρ)	1000 kg/m^3	
Kinematic viscosity(ν)	$10^{-6}m^2/s$	
Characteristic length (L)	r_b	
Angular velocity (ω)	ω	
Characteristic velocity (V)	$\omega \times r_b$	
Reynolds number	$\frac{V \times L}{\nu}$	Calculation based on **?**]

Table 4.1: Fluid Properties and Reynolds number

The Reynolds number thus calculated are given in Table 4.2.

RPM	Reynolds number	Remarks
50	818	Calculation based on Table 4.1
75	1227	
100	1636	
120	1963	

Table 4.2: RPM and Reynolds number

4.5 Results and Discussion

The velocity vector in the longitudinal cross-section is shown in Fig. 4.6. Higher range of velocity is observed near the blades because of the rotation of blades.

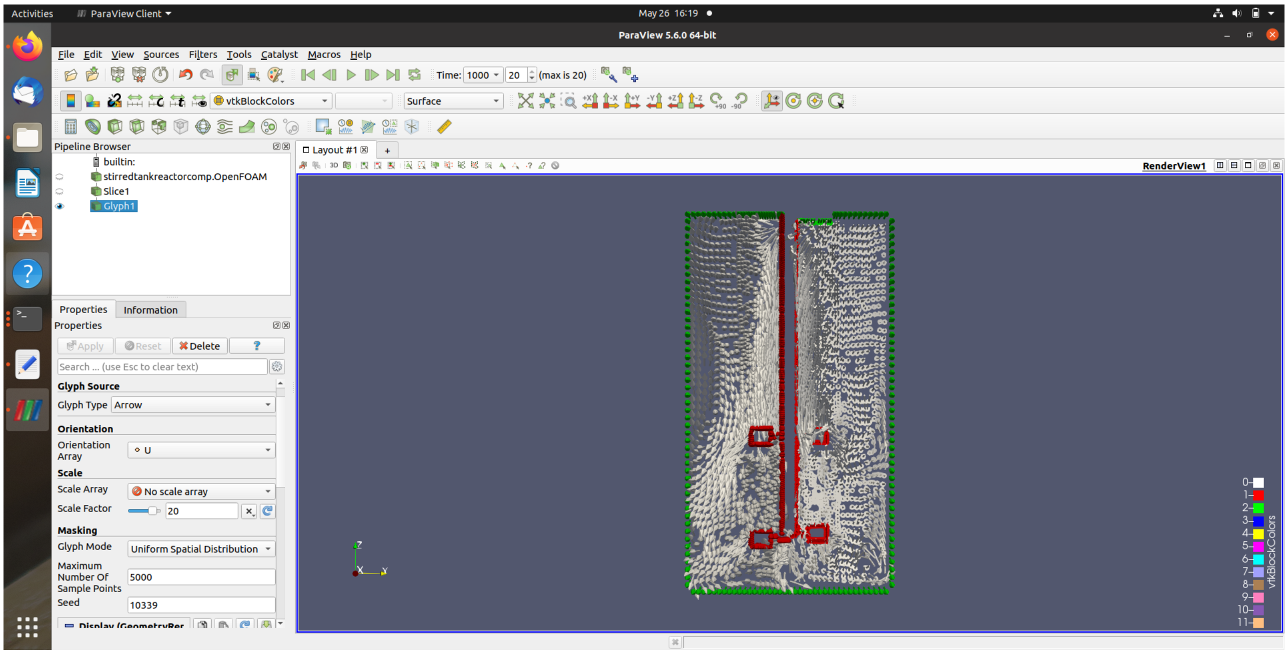


Figure 4.6: Velocity vector distribution in longitudinal cross-section

Comparison of radial velocity is shown in Fig. 4.7.

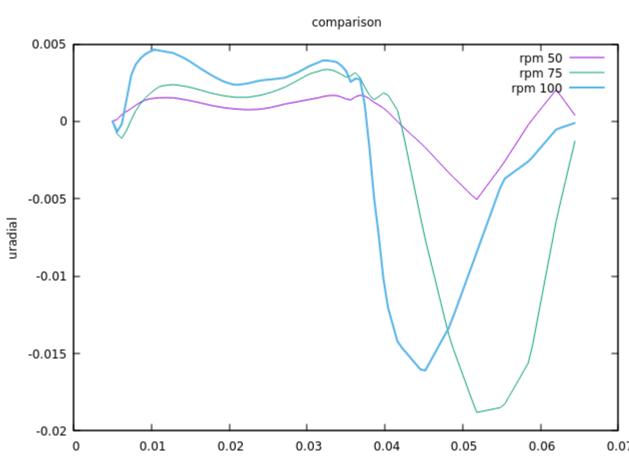


Figure 4.7: Comparison of radial velocity

Comparison of pressur is shown in Fig. 4.8. It is observed that pressure decrease with increase of rotation because of the Bernoulli's theorem.

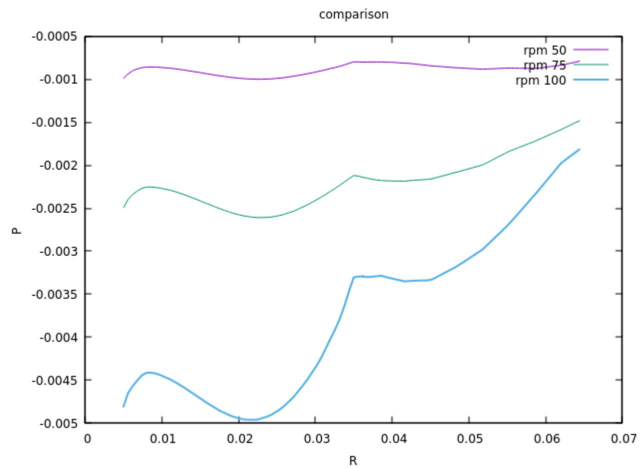


Figure 4.8: Comparison of pressure for different RPM

Chapter 5

Conclusion

The complex fluid flow distribution in a stirred tank reactor has been modelled and solved using OpenFOAM. The problem of 45° pitched blade impeller of **?**] has been solved, compared and presented. The geometry has been created in FreeCAD and imported in OpenFOAM. The MRF has been used. Meshing has been carried out by snappyHexMesh. Solver used is the steady-state simpleFoam. Results show that a good agreement has been obtained.

Later on a two-impeller Rushton type stirred tank reactor has been modelled and solved for different RPM. The results are presented for velocity and pressure.

Appendix A

Acceleration in relative frame of reference

Here XYZ is inertial frame reference and xyz moving frame refrence and P is the object.

Figure A.1

From triangle law of vector addition

$$\overrightarrow{r} = \overrightarrow{r'} + \overrightarrow{R}$$

$$\overrightarrow{r'} = \overrightarrow{r} - \overrightarrow{R}$$

$$\tfrac{d\overrightarrow{r}}{dt} = \tfrac{d\overrightarrow{r'}}{dt} - \tfrac{dR}{dt}$$

But now the moving refrence frame is rotating,

Figure A.2

$$\overrightarrow{r} = \overrightarrow{r}_x\hat{\imath} + \overrightarrow{r}_y\hat{\jmath} + \overrightarrow{r}_z\hat{k}$$

Now differentiating,

$$\tfrac{d\overrightarrow{r}}{dt} = \underbrace{\frac{d\overrightarrow{r}_x}{dt}\hat{\imath} + \frac{d\overrightarrow{r}_y}{dt}\hat{\jmath} + \frac{d\overrightarrow{r}_z}{dt}\hat{k}}_{\frac{d\overrightarrow{r}}{dt}} + \underbrace{\overrightarrow{r}_x\frac{d\hat{\imath}}{dt} + \overrightarrow{r}_y\frac{d\hat{\jmath}}{dt} + \overrightarrow{r}_z\frac{d\hat{k}}{dt}}_{\Omega\times\overrightarrow{r}}$$

Hence,

$$\frac{d\overrightarrow{r}}{dt} = \frac{d\overrightarrow{r}}{dt} - \Omega \times \overrightarrow{r}$$

$$[\frac{dr}{dt}]_I = [\frac{dr}{dt}]_R + \Omega \times r \tag{A.1}$$

$$u_I = u_R + \Omega \times r \tag{A.2}$$

Then we define the acceleration:

$$[\frac{du_I}{dt}]_I = [\frac{du_I}{dt}]_R + \Omega \times r \tag{A.3}$$

$$[\frac{du_I}{dt}]_I = [\frac{d[u_R + \Omega \times r]}{dt}]_R + \Omega \times [u_R + \Omega xr] \tag{A.4}$$

$$[\frac{du_I}{dt}]_I = [\frac{du_R}{dt}]_R + \frac{d\Omega}{dt} \times r + \Omega \times [\frac{dr}{dt}]_R + \Omega \times u_R + \Omega \times \Omega \times r \tag{A.5}$$

$$[\frac{du_I}{dt}]_I = [\frac{du_R}{dt}]_R + \frac{d\Omega}{dt} \times r + \Omega \times u_R + \Omega \times u_R + \Omega \times \Omega \times r \tag{A.6}$$

$$[\frac{du_I}{dt}]_I = [\frac{du_R}{dt}]_R + \frac{d\Omega}{dt} \times r + 2\Omega \times u_R + \Omega \times \Omega \times r \tag{A.7}$$

Also we know,

$$\frac{Du}{Dt} = \frac{\partial u}{\partial t} + u.\nabla u \tag{A.8}$$

Hence Navier-Stokes equations in the relative frame with relative velocity

$$\frac{Du_I}{Dt} = \frac{Du_R}{Dt} + \frac{d\Omega}{dt} \times r + 2\Omega \times u_R + \Omega \times \Omega \times r \tag{A.9}$$

Appendix B

Introduction

Acceleration in relative frame of reference

Acceleration in the relative frame with relative velocity

$$\frac{Du_I}{Dt} = \frac{Du_R}{Dt} + \frac{d\Omega}{dt} \times r + 2\Omega \times u_R + \Omega \times \Omega \times r \tag{B.1}$$

$$\frac{Du_I}{Dt} = \frac{\partial u_R}{\partial t} + u_R.\nabla u_R + \frac{d\Omega}{dt} \times r + 2\Omega \times u_R + \Omega \times \Omega \times r \tag{B.2}$$

$$\frac{Du_I}{Dt} = \frac{\partial u_R}{\partial t} + \nabla(u_R \otimes u_R) + \frac{d\Omega}{dt} \times r + 2\Omega \times u_R + \Omega \times \Omega \times r \tag{B.3}$$

We know, NS equation in inertial frame of reference is:

$$\frac{Du}{Dt} = -\nabla(\frac{p}{\rho}) + \nu\nabla.\nabla(u) \tag{B.4}$$

In relative frame of reference,

$$\nabla(u_R \otimes u_R) + 2\Omega \times u_R + \Omega \times \Omega \times r = -\nabla(\frac{p}{\rho}) + \nu\nabla.\nabla(u_R) \tag{B.5}$$

Hence,

$$\nabla.(U_r U_r) = -\nabla p + \nabla.(\nu\nabla U_r) - \underbrace{2\Omega \times u_R}_{coriolis} - \underbrace{\Omega \times \Omega \times r}_{centrifugal} \tag{B.6}$$

Appendix C

Navier-Stokes equations in the relative frame with absolute velocity

$$\nabla.(u_R \otimes u_R) = \nabla.(u_R \otimes [u_I + \Omega \times r]) \tag{C.1}$$

$$\nabla.(u_R \otimes u_R) = \nabla.(u_R \otimes u_I) + \nabla.u_R(\Omega \times r) - u_R.\nabla(\Omega \times r) \tag{C.2}$$

$$\nabla.(u_R \otimes u_R) = \nabla.(u_R \otimes u_I) - \Omega \times u_R \tag{C.3}$$

Now,

$$\nabla.(u_R \otimes u_R) + 2\Omega \times u_R + \Omega \times \Omega \times r = \nabla.(u_R \otimes u_I) - \Omega \times u_R + 2\Omega \times u_R + \Omega \times \Omega \times r \tag{C.4}$$

$$\nabla.(u_R \otimes u_R) + 2\Omega \times u_R + \Omega \times \Omega \times r = \nabla.(u_R \otimes u_I) + \Omega \times u_R + \Omega \times \Omega \times r \tag{C.5}$$

$$\nabla.(u_R \otimes u_R) + 2\Omega \times u_R + \Omega \times \Omega \times r = \nabla.(u_R \otimes u_I) + \Omega \times (u_R + \Omega \times r) \tag{C.6}$$

$$\nabla.(u_R \otimes u_R) + 2\Omega \times u_R + \Omega \times \Omega \times r = \nabla.(u_R \otimes u_I) + \Omega \times u_I \tag{C.7}$$

So, the Navier-Stokes equation may be modified as:

$$\nabla.(u_R \otimes u_I) + \Omega \times u_I = -\nabla(\frac{p}{\rho}) + \nu\nabla.\nabla(u_I) \tag{C.8}$$

Hence,

$$\nabla.(UU_r) = -\nabla p + \nabla.(\nu\nabla U) - \underbrace{\Omega \times U}_{source} \tag{C.9}$$

?]

Appendix D

Cartesian coordinates to Cylinder coordinates transformation

Here XYZ is coordinate axis.

Figure D.1

$$\vec{r} = x\hat{e}_x + y\hat{e}_y + z\hat{e}_z \tag{D.1}$$

$$x = |r|\, cos\theta, y = |r|\, sin\theta, z = z \tag{D.2}$$

$$\rho = |r| \tag{D.3}$$

$$\vec{r} = \rho cos\theta\hat{e}_x + \rho sin\theta\hat{e}_y + z\hat{e}_z \tag{D.4}$$

$$\vec{e_r} = \frac{\partial r}{\partial\, |r|} = cos\theta\hat{e}_x + sin\theta\hat{e}_y \tag{D.5}$$

$$\vec{e_\theta} = -\,|r|\, sin\theta\hat{e}_x + |r|\, cos\theta\hat{e}_y \tag{D.6}$$

$$\hat{e}_z = \hat{e}_z \tag{D.7}$$

$$\hat{e_\theta} = \frac{1}{|r|}\vec{e_\theta} = -sin\theta\hat{e}_x + cos\theta\hat{e}_y \quad \text{(D.8)}$$

$$sin\theta = \frac{y}{\sqrt{x^2+y^2}}; cos\theta = \frac{x}{\sqrt{x^2+y^2}} \quad \text{(D.9)}$$

Now

$$\vec{r} = x\hat{e}_x + y\hat{e}_y + z\hat{e}_z \quad \text{(D.10)}$$

$$\frac{d\vec{r}}{dt} = v_x\hat{e}_x + v_y\hat{e}_y + v_z\hat{e}_z \quad \text{(D.11)}$$

$$v_x\hat{e}_x + v_y\hat{e}_y = v_r\hat{e}_r + v_\theta\hat{e}_\theta = v_r(cos\theta\hat{e}_x + sin\theta\hat{e}_y) + v_\theta(-sin\theta\hat{e}_x + cos\theta\hat{e}_y) \quad \text{(D.12)}$$

$$v_x = v_r cos\theta - v_\theta sin\theta \quad \text{(D.13)}$$

$$v_y = v_r sin\theta + v_\theta cos\theta \quad \text{(D.14)}$$

Multiplying eqn. D.13 by cosϑ eqn. D.14 by sinϑ and then adding:

$$v_x cos\theta = v_r cos^2\theta - v_\theta sin\theta cos\theta \quad \text{(D.15)}$$

$$v_y sin\theta = v_r sin^2\theta + v_\theta sin\theta cos\theta \quad \text{(D.16)}$$

$$v_r = v_x cos\theta + v_y sin\theta = \frac{v_x x}{\sqrt{x^2+y^2}} + \frac{v_y y}{\sqrt{x^2+y^2}} \quad \text{(D.17)}$$

Hence, we get

$$v_r = \frac{xv_x + yv_y}{\sqrt{x^2+y^2}} \quad \text{(D.18)}$$

Now similarly multiplying eqn. D.13 by sinϑ, eqn. D.14 by cosϑ and subtracting and finally we get:

$$v_x sin\theta + v_y cos\theta = -v_\theta \tag{D.19}$$

And hence, we get

$$v_\theta = \frac{xv_y - yv_x}{\sqrt{x^2 + y^2}} \tag{D.20}$$

?]

Appendix E

Codes used in blockMesh, snappyHexMesh

blockMesh

```
convertToMeters 1;
vertices
(
(-4 -4 0)
(4 -4 0)
(4 4 0)
(-4 4 0)
(-4 -4 7)
(4 -4 7)
(4 4 7)
(-4 4 7)
);
blocks
(
hex (0 1 2 3 4 5 6 7) (40 40 60) simpleGrading (1 1 1)
);
edges
(
);
```

boundary
(
allBoundary
{
type patch;
faces
(
(0 4 7 3)
(2 6 5 1)
(1 5 4 0)
(0 3 2 1)
(4 5 6 7)
);
}
top
{
type symmetry;
faces
(
(3 7 6 2)
);
}

snappyHexMesh

castellatedMesh true;
snap true;
addLayers false;
geometry
{
impeller
{
type triSurfaceMesh;
file "impeller.stl";
}

vessel
{
type triSurfaceMesh;
file "vessel.stl"; } //- Used to define MRF zone and refine mesh round the impeller
MRF
{
type searchableCylinder;
point1 (0.0 0 0);
point2 (0.0 0 5.7);
radius 1.1;
}
}
castellatedMeshControls
{
maxLocalCells 100000;
maxGlobalCells 2000000;
minRefinementCells 0;
nCellsBetweenLevels 2;
maxLoadUnbalance 0.10;
features
(
{
file "vessel.eMesh";
level 1;
}
{
file "impeller.eMesh";
level 3;
}
);
resolveFeatureAngle 30;
refinementSurfaces
{

```
MRF
{
level (3 3);
cellZone cellMRFzone;
faceZone faceMRFzone;
cellZoneInside inside;
}
impeller
{
level (3 3);
}
vessel
{
level (1 1);
}
}
refinementRegions
{
MRF
{
mode inside;
levels ((1E15 0));
}
}
locationInMesh (1.2 1.2 1.5); // Inside point
allowFreeStandingZoneFaces true;
}
// Settings for the snapping.
snapControls
{
nSmoothPatch 3;
tolerance 1.0;
nSolveIter 450;
nRelaxIter 10;
```

```
nFeatureSnapIter 10;
implicitFeatureSnap true;
explicitFeatureSnap false;
multiRegionFeatureSnap true;
}
addLayersControls
{
relativeSizes true;
layers
{
}
expansionRatio 1.0;
finalLayerThickness 0.3;
minThickness 0.25;
nGrow 0;
featureAngle 30;
nRelaxIter 5;
nSmoothSurfaceNormals 1;
nSmoothNormals 3;
nSmoothThickness 10;
maxFaceThicknessRatio 0.5;
maxThicknessToMedialRatio 0.3;
minMedianAxisAngle 90;
nBufferCellsNoExtrude 0;
nLayerIter 50;
nRelaxedIter 20;
}
meshQualityControls
{
#include "meshQualityDict"
relaxed
{
maxNonOrtho 75;
}
```

```
}
writeFlags
(
);
mergeTolerance 1E-6;
```