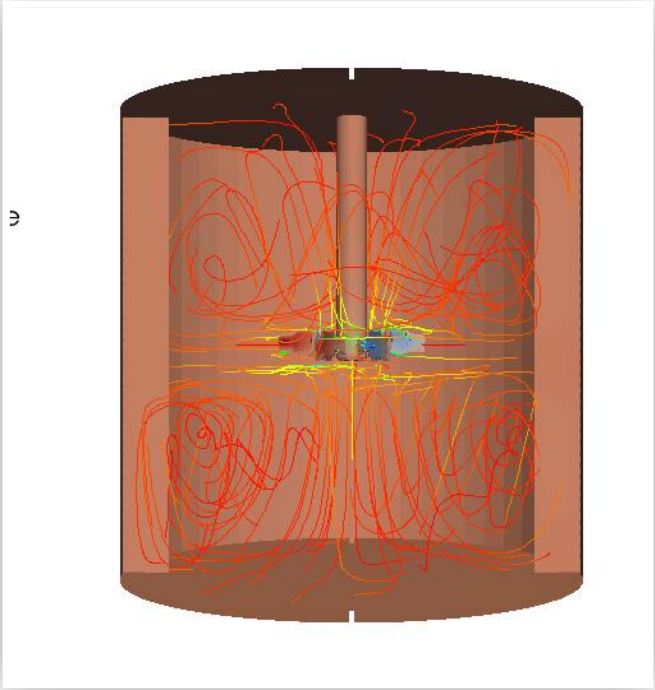
ECMM106 – Computational Engineering

Mini Project

RANS Investigation for a Three Dimensional Rushton Turbine Model

---



Stephen Pavey
570005180

Contents

Acknowledgements

The valuable input of the following parties is acknowledged:

Dr Gavin Tabor:	For advice on changing meshing approach and all input throughout the course which has been used in this report
Alastair Begely:	For instruction in parallel processing

Abstract

This project realises a 3D RANS model of the 6-blade Rushton Turbine using a Multiple Reference Frame approach in the CFD code *OpenFOAM*. Optimising for mesh convergence, turbulence model and considering the differencing schemes used, this demonstrates that the MRF approach in three dimensions can be accurately modelled though validation with both particle imagery and proton emission experimental data from the public sphere.

Key words:

Rushton Turbine, MRF, RANS, SnappyHexMesh

1. Introduction

1.1 The Rushton Turbine

This project is concerned with modelling a Rushton Turbine using the Computational Fluid Dynamics (CFD) code, *OpenFOAM*. The Rushton Turbine is ubiquitous in the chemical engineering industry for mixing applications. It comprises of a cylindrical tank with a vertically mounted impellor in the centre, installed at some clearance level, C, from the base. The impellor is formed by a series of vertical fins mounted radially on a disk. This impellor is spun to introduce turbulence in the flow in order to mix the contents of the fluid. The resulting flow is prevented from rotating around the central shaft by means of a set of baffles at the edge of the tank. Large quantities of turbulent eddies are formed in the flow and a high turbulent energy mixes the culture or constituent phases in the region. A diagram of this design is shown as **figure 1.**

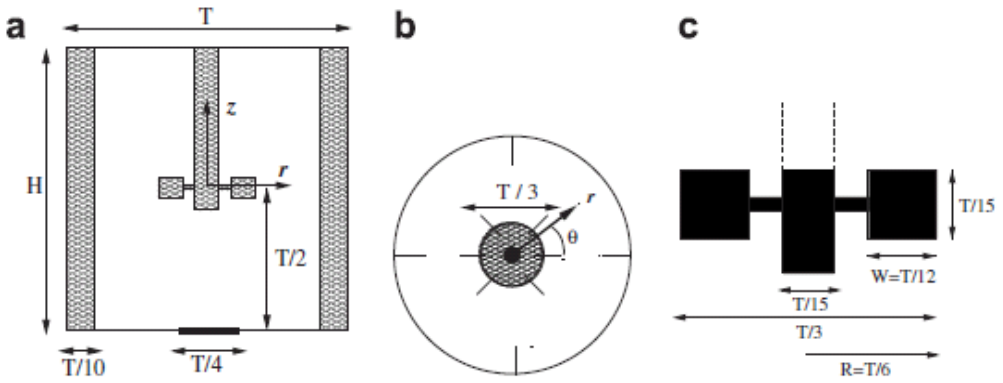


Fig 1. The Rushton Turbine (Chiti, et al. 2011)

1.2 Literature Review Summary

There are several different techniques which have been used to model this type of system. Experimental approaches to assess the flow under laboratory conditions include particulate tracing using reflection of light, or Particle Image Velocimetry (PIV). Other techniques include Laser Doppler Anemometry (LDV) and Proton Emission Particle Tracking (PEPT).

Computation Fluid Dynamics (CFD) approaches include static meshes, where source and sink terms at the boundaries of the fins model the rotation of the turbine, through to dynamic mesh techniques where a region containing the impellor rotates with respect to the outer mesh. The sliding grid approach best models the flow for Ruston turbines, yielding transient effects in three dimensions although approaches which require are less expensive in computer power such as Multiple Reference Frame (MRF) methods. To affect an MRF method, two regions are specified with a fixed interface. For the Rushton Turbine, the code calculates the flow based on a moving reference frame around one region, and a stationary reference frame around the other. The interaction of the flow at the interface is calculated and thus the rotational effects are modelled without requiring mesh movement in a pure sense. This therefore calculates the flow patterns around the features while being able to run a steady state code: *MRFsimpleFoam*.

There is some disagreement in the best turbulence model to be used in this case. In particular more basic models use k-epsilon effectively, although for multiphase flow the LES approach seems to be preferred. Extensive experimental data for validation is available, some of which is

included in this review but no work has been found which uses the *OpenFOAM* CFD code. Modelling multiphase, both gaseous and of other liquids can be modelled as an extension to the dynamic mesh CFD approaches.

The two journal articles used to provide validation data are described below.

1.3 Numerical Simulation of a gas-liquid Rushton stirred reactor – LES and LPT (Arlov, Revstedt and Fuchs 2008)

The Ruston Turbines are often used to mix multiple phases of liquids. In this case an aerated liquid is modelled using both LES on a multiphase region and Lagrangian Particle Tracking (LPT) to model the paths of individual bubbles in the flow. The models used are verified using LDA on the dispersed gas bubbles. The mesh is achieved using a Volume of Solid approach (VOS) to block out cells which contain the impellor.

To build this model the mesh elements must be smaller than the size of the bubbles in the flow. Despite this constraint the dispersal of bubbles was accurately modelled to experimental data and a significant difference was obtained from CFD single phase data showing that this approach can be used well to model the multiphase flow.

1.4 Reynolds number scaling of flow in a Rushton turbine stirred tank. Part I—Mean flow, circular jet and tip vortex scaling (H.S.Yoon 2005)

This Journal describes two methods for modelling the flow in the Rushton Turbine, one experimental and another using CFD. Experimentally this approach uses Particle Image Velocimetry (PIV). The impellor was mounted in a body of fluid with a laser sheet shone through a tube of water before encountering the turbine. Particles seeded into the fluid reflect the light and images are picked up by a CCD. The algorithms used to interpret the results are analogous to the mesh in the CFD as far as integral areas on the images are used to map the flow. The change in light intensity seen in each area, with respect to the change in time between each recorded image can be used to specify the velocity field in one given plane. The disadvantage of this approach is that the baffles are not simulated as the turbine is located in a tube connected to the laser. Nevertheless good quality data is provided on a micro scale near the impellor.

This investigation concluded that the CFD under-predicts the turbulent energy, both for the k-epsilon and the RNGk-epsilon models. The standard k-epsilon approach most closely follows both radial and tangential velocity profiles on the blades where the RNG model does not compare well with the experimental profiles near the blade tips. It is noted that improved performance near the tips of the impellors can be achieved using a finer mesh. Another study is noted here for the quality of data found using the same PIV approach. Although not reviewed in this section the data displayed may be useful at later stages of this project. (H.S.Yoon 2005)

1.5 Project aims

This project aims to realise a model of the Rushton Turbine using the *OpenFOAM* code. It aims to take the given *MRFSimpleFoam* case of the four-blade Rushton Turbine in two dimensions and expand it into three dimensions. This model will then be optimised for mesh convergence, turbulence model and, with respect to differencing schemes. This model will demonstrate that the MRF approach in three dimensions can be accurately modelled in *OpenFOAM*.

2. The 2D Case

2.1 Overview

The starting point for this project is the *MRFSimpleFoam* model: *mixerVessel2D*. This models a four-blade Rushton Turbine in a four-baffle mixing tank. This model was analysed to ensure it matched experimental data. The first models in three dimensions were built using this as a starting point.

2.2 Geometry and Meshing Techniques

The case geometry was constructed using *blockMesh* which creates a series of blocks as defined in the file *blockMeshDict*. The blocks in the domain are specified by eight vertices and the cylindrical geometry was built by specifying the inner and outer edges as arcs. Since the rotor was offset by $1/8^{th}$ turn from the baffles there were two blocks between each impellor blade. Four layers of blocks were specified in order to respect each feature. The baffles and blades are modelled as two dimensional elements with patches between blocks being specified as walls. Where either feature connected with the shaft or wall, the vertices were duplicated in order to topologically define the blades and baffles as hexahedra, while the geometry maps the faces in identical locations. This ensures each adjacent block owns a patch for these fixed boundaries. All other shared patches are automatically removed through *faceMatching*. A sketch of this geometry is shown as **figure 2.**

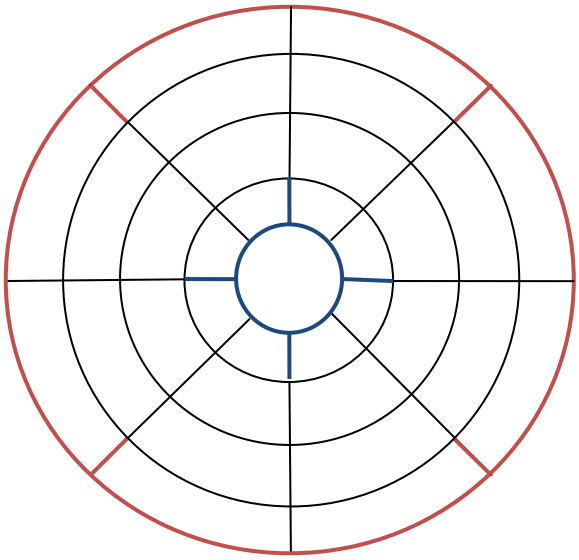


Fig 2. BlockMesh Geometry mixerVessel2D

The walls of the rotor are shown in blue and the baffles and the outer circumferences are shown in red. There are two block layers between these features to ensure the interface between the two reference zones to lies beyond the edges of the baffles and blades.

The 2D model is simulated using *empty* boundary conditions for the front and back faces of the model. This prevents wall effects being introduced at these faces and thus allows the simulation to act as a slice through the turbine. Since each block is defined in the same sequence, it was decided that it would reasonably straight-forward to multiply up the required number of blocks to model the six blades on the impellor shown in **figure 1** with blocks above and below the case.

The simulation was constructed using the k-ε turbulence model and with Gauss-linear differencing schemes. The resulting velocity contours at 500 time steps is shown as **figure 2.**

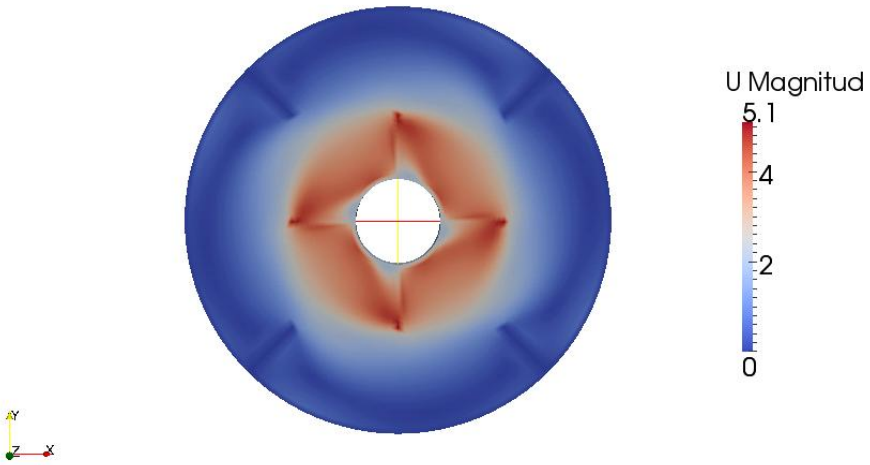


Fig 3. 2D case velocity plot

The rotational effects in the sudo steady-state MRF approach is outlined in the file */constant/MRFZones*. The central 'rotor' blocks are named in the *blockMeshDict* as the first two rings of blocks shown in **figure 2.** It is to these that the rotational reference frame is linked. The axis of rotation is specified along the centre axis of the model and the angular velocity, ω, is then specified. The relevant lines are shown below:

```
rotor
{
origin    origin [0 1 0 0 0 0 0]   (0 0 0);
axis      axis   [0 0 0 0 0 0 0]   (0 0 1);
omega     omega  [0 0 -1 0 0 0 0] 104.72;
}
```

It is noted that any approach to meshing must retain these zones in order to realise an MRF model.

2.3 limitations of 2D model

In order to validate this with other results the geometry was changed to match the case for simulation data available in the public sphere conducted by Arlov and Fuchs (Arlov, Revstedt and Fuchs 2008). The velocity contours and the graphs comparing the datasets are shown as **figure 4.**

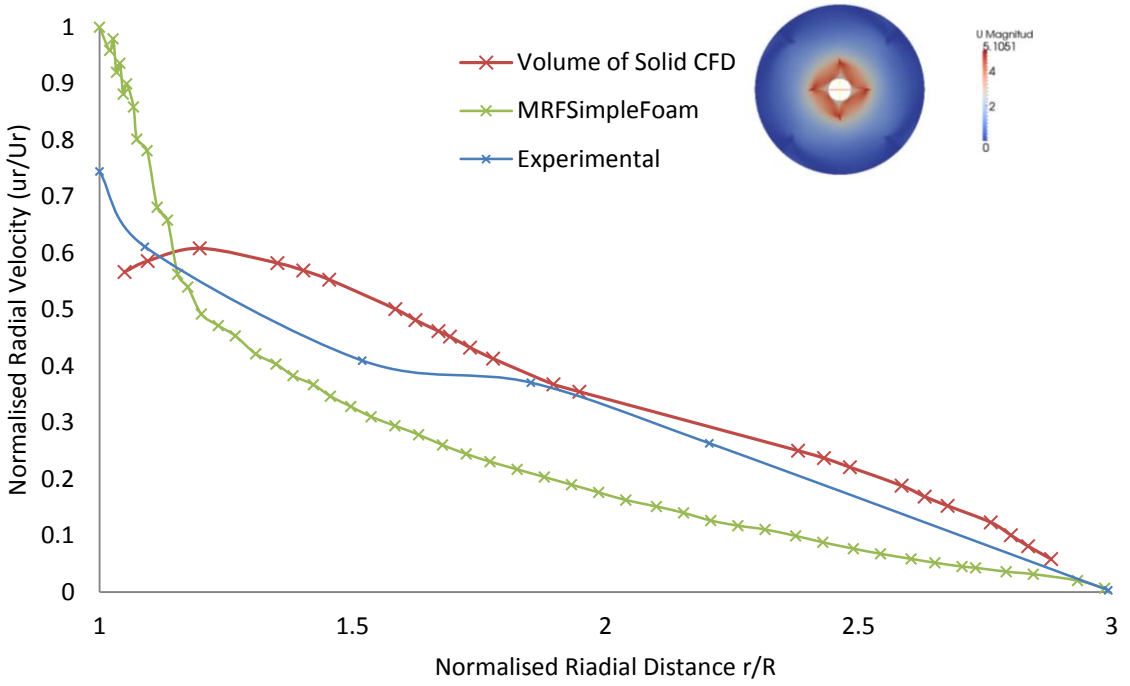


Fig 4 2D Case Validation (Arlov, Revstedt and Fuchs 2008)

In each case the trends are measured from the tip edge to the tank wall midway between two baffles. The trend showing the 2D *MRFSimpleFoam* case file shows how the normalised radial velocity decreased from the speed of the impellor tip to zero at the tank edge. The experimental data sourced from Arlov and Revstedt fits this trend in a similar fashion (Arlov, Revstedt and Fuchs 2008).

By first principles the fluid should share the velocity of the wall at impellor tip so the intial value of 0.75 is considered to be from a measurement error. The Volume of Solid (VOS) approach used in this journal treats the impellor as a black box adding source and sink terms at the interface to the domain to simulate the rotation. This explains why the trend is also lower at the impellor tip. It is also possible that the 2D blades used in the MRF case do not resolve the vortices at the tips.

Generally, all trends show a general decrease with the MRF approach demonstrating a greater initial decrease. Since both this and the VOF methods use the k-ε model it is not the choice of turbulence model that accounts for this difference. Additionally, at the tank wall there is some disagreement with the VOS model showing a more gradual decrease. It is possible that this is a function of the mesh quality in the 2D case. It is most likely that the large mesh elements specified at this point do not adequately resolve the boundary layer.

Overall this demonstrates that this model accurately represents the physics but shows the importance of mesh convergence for later models in the project.

3. Changing the geometry

In order to model the six blades of the standard Rushton Turbine found in other work (Ranade, Perrade, et al. 2001) (Ranade, Perrade, et al. 2001) the *blockMesh* approach used in **section 2** was investigated with a view to expand into three dimensions. Firstly the *blockMeshDict.m4* within this case was assessed in order to see if a simple replication and amendment of entries could be used to construct a three dimensional model.

The file specifies the overall dimensions and then calculates the trigonometric values used to calculate Cartesian co-ordinates in each of the 8 radial directions seen in **figure 2** and a further eight used to specify the points through which the arcs are specified**.** These are then used to calculate the 96 vertices which are required to construct the model. The 80 arcs were then also specified to turn each of the 32 blocks into a segment of the ring.

It was deemed that in order to model the six-bladed impellor to match the validation data, the number of radial directions used to specify the blocks would stay the same since the alignment of the impellor would match the baffles for two of the blades. However the number of blocks would only increase to a minimum of 36 to include the regions above and below the impellor.

Since the meshes achieved through the *blockMesh* approach can be of a higher quality when compared to mesh refinement techniques such as *snappyHexMesh* it was decided to investigate the possibility of meshing using *blockMesh* alone.

3.1 Sequential *blockMesh* development

Rather than adapting the very complex *blockMeshDict.m4* a new *blockMeshDict* was constructed from scratch. Firstly a single quarter-ring was built as shown in **figure 6.** This demonstrated the order in which to specify the vertices and the operation of the *edge arc* entries, which were specified to 3 decimal places. This seemed to be sufficiently accurate at this mesh refinement. Any slightly non-circular geometry resulting from this approach would not affect results since the does not rotate. Thus given that the patches conform internally the blocks do not have to be perfectly circular.

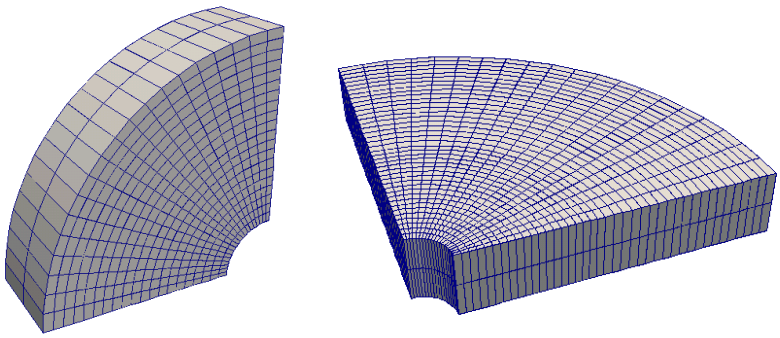


Fig 6. Initial blockMesh constructions

Following this a model was constructed with an inner and outer ring. The model was then run with *MRFSimpleFoam* to see how the block construction affected fluid flow. The left hand contour of **figure 7** showed that the internal faces had not meshed for the first two blocks and the problem was not resolved by building the eight blocks in the right hand image.

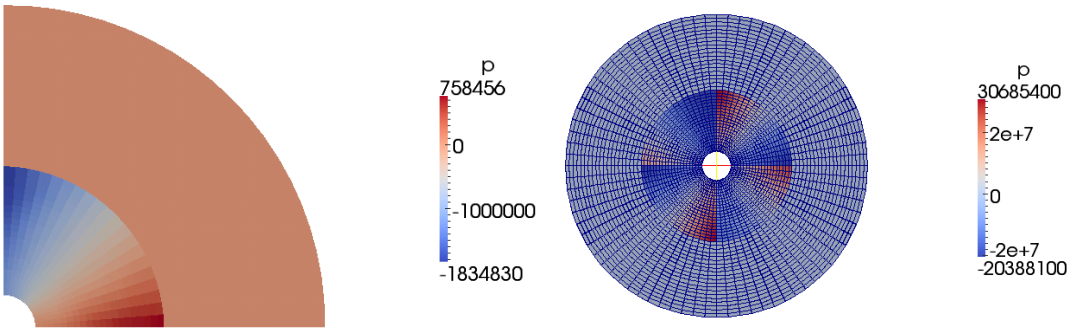


Fig 7. MRF assessment of simple blockMesh geometry

To resolve this issue the *mergePatchPairs* capability was used by specifying in the *blockMeshDict* the internal faces to be merged. However under this approach only two of the internal circumferential faces could be successfully meshed. Despite changing master and slave configurations as advised through forum entries (Dr Hrvoje Jasak 2006), both by swapping which was the master and by naming different numbers of master/slave in each patch definition this issue was not resolved. **Figure 8** shows how the radial faces between the outer rings have been merged with this approach, but that the interface between inner and outer rings could not be realised.

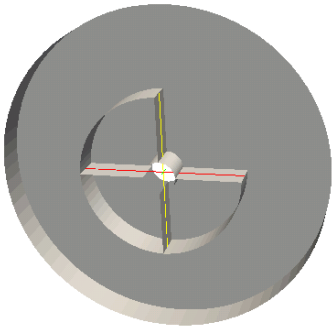


Fig 8 Merging internal faces

It is probable that merging faces which intersect in a t-shape is beyond the capability of *blockMesh* (CFD Online 2011). The intersections of the four faces which require merging in this model exceed this error. To overcome this problem the rings could be meshed separately to be stitched together *mergeMesh* and *stitchMesh* to join them together and assure that to conform the nodes. It was therefore deemed that a new approach was required.

Additional models were created with shared vertices to utilise the automatic face matching of *blockMesh*. However as mentioned above triangular topology at the boundaries was required so that each block could own an internal face at the rotor and baffles. When each block has eight unique nodes, new blocks can be simply constructed by adding 8 or 16 to the block and arc definitions. However, when some blocks have 10 associated vertices and others only 8, writing a complex *blockMeshDict* of becomes significantly more difficult and this approach to modelling the geometry was abandoned.

3.3 *SnappyHexMesh* Approach

A more direct approach to model the geometry was to employ the *snappyHexMesh* utility. This allowed for *.stl* surface files to specify regions to be subtracted from the mesh and the mesh quality at the boundaries to be refined.

Firstly a *blockMesh* was defined using the quadrant method described above for the outer ring. The inner cylinder was built using a different approach to overcome the issues with merging faces described in **section 2.** Vertices were defined for a square block and arcs specified to fit the edges to a circle. In this way, with non-repeated vertices, *blockMesh* successfully merged the internal faces by face matching. The rotor and stator were imported as *.stl* files. Despite the care taken to ensure that the origin for the surfaces was shared for these files and the *blockMesh* the output origins did not correspond to the centre of the model and the vertical directions were at right angles as shown in **figure 9**

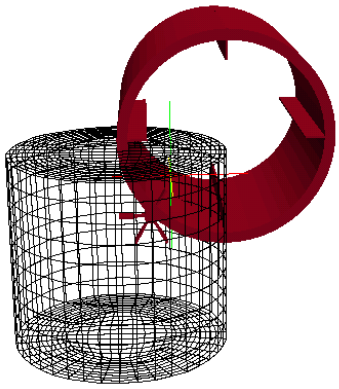


Fig 9. SnappyHexMesh constituent parts initial configuration

The mesh thus required both the *rotateMesh* and *transformPoints –translate* commands to centre the files in the mesh. Two *snappyHexMesh* techniques were employed sequentially and the overall model checked with *MRFSimpleFoam* to ensure that the zones required for the MRF solver had been retained. The axis of rotation in *MRFZones* was also changed accordingly.

The results shown in **figure 10** demonstrate the success of this approach, allowing a refined boundary layer around the wall elements and successfully meshing the internal rotor and the outer baffles while retaining the required cell zones.

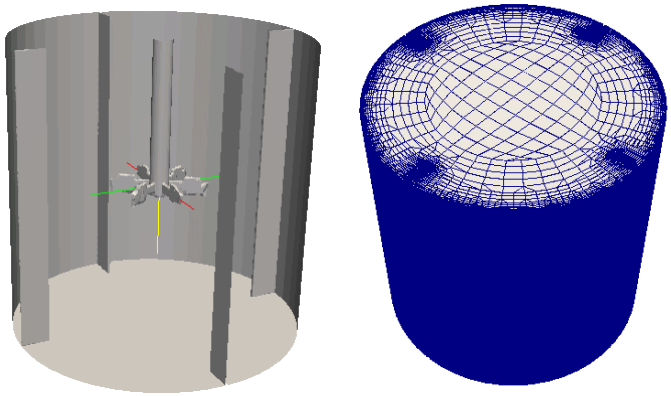


Fig 10 Successful snappyHexMesh layers (5 6)

The boundaries at the rotor and stator were set as *wall*, with velocity specified to match that of the wall. However since the boundaries at the top and bottom of the tank are considered to be moving in relation to one another they could not be modelled as wall surfaces without driving the flow. Therefore these were set to *slip,* ensuring that no flow driving artefacts arose from the simulation at this point. Since nice neither the top or bottom walls are realised the model a 3D slice through a tank. This should not significantly reduce the quality of the results although there may be some limitation to the recirculation as a result.

The model was first run for 500 iterations as in the given 2D case. However this did not converge. It was then run for a further 4500 iterations to assess the required number of time steps. The resultant plot is shown as **figure 11.**

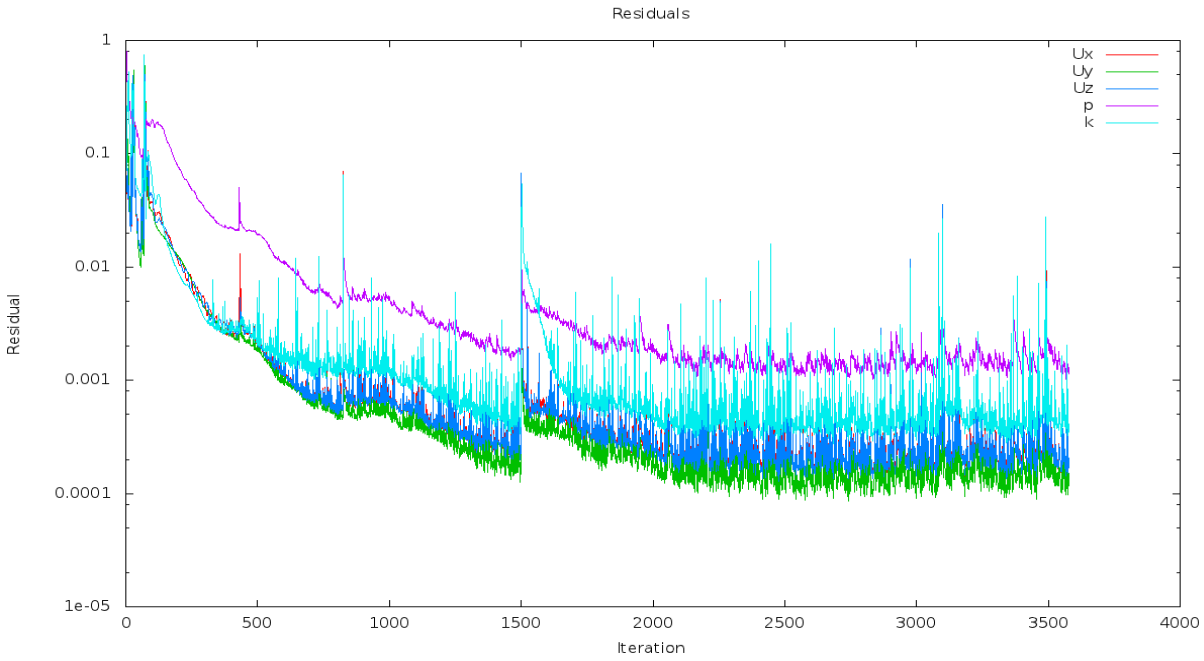


Fig 11. Converged Residuals

Although it could be considered to be converged at 1000 iterations the residual spikes, particularly shown at at 1500 iterations demonstrate how the *MRFSimpleFoam* is beginning to model some transient behaviour despite being a steady-state solver. This is because of the sudo-transient nature of this problem. Although MRF does not involve mesh motion, it simulates the effects of mesh motion by considering each stationary zone from a different, relatively moving reference frame. It was therefore decided to take each simulation to 2000 iterations for convergence.

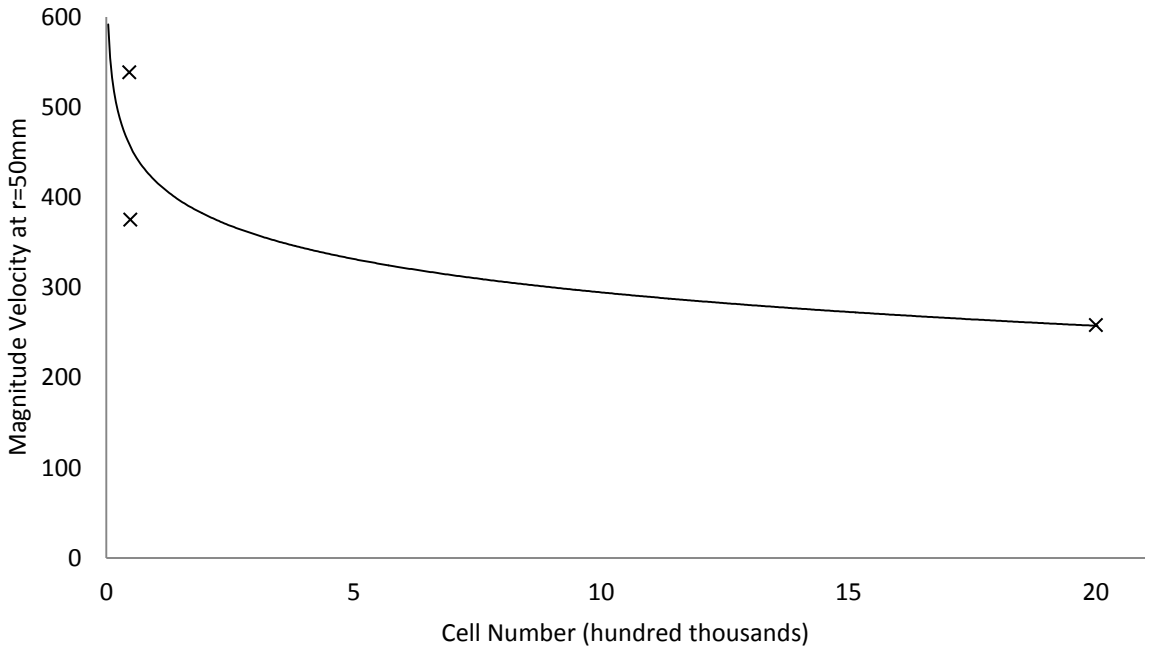
3.4 Mesh Refinement study

In order to ensure that the results comment on the physical situation modelled it was essential to ensure that the mesh density is sufficiently high to resolve the model without using an unnecessarily expensive number of cells.

This was achieved by changing the refinement levels within *snappyHexMeshDict.* This entry in the dictionary specifies the maximum and minimum number of levels between the *blockMesh* geometry and the surface meshed. The initial attempts used (1 2), (3 4), (5 6), and (6 7) level

entries to refine the mesh. However, only the top two levels meshed successfully. The first two introduced severely non-orthoganal cells, identified through *checkMesh* and through running simulations which crashed out between 7 and 47 iterations. It was therefore decided to try different combinations which would allow a greater level of freedom for *snappyHexMesh* to meet its own mesh quality criteria.

Out of the ~27 meshes tried, only the combination (1 3) yielded a mesh which did not crash the *MRFSimpleFoam* solver. Although this is not an ideal number of successful meshes for a convergence study, the principle for convergence are illustrated and the estimated logarithmic convergence shown in **figure 12** was used to select an appropriate mesh for subsequent models. Since the mesh at the walls will share the wall velocity the magnitude of velocity 50 mm radially from the middle of the impellor tip was used to compare the meshes. From this study a mesh of 500,000 cells would give a reasonable result although it cannot be considered to have completely converged with this number of trials. Subsequent trials were therefore completed with the highest mesh density yielded from levels (5 6) to ensure the highest quality results.



The source of the meshing problems is unlikely to lie within the settings for *snappyHexMesh*. Inspection of the mesh surfaces showed how the severely skewed cells are present at a cross section with the impellor. These are highlighted in **figure 13.** Problem cells occur at the baffle edges since these are very thin elements and spurious refined regions are also present. In particular there are heighly skewed elements at the corners of the inner cylinder.

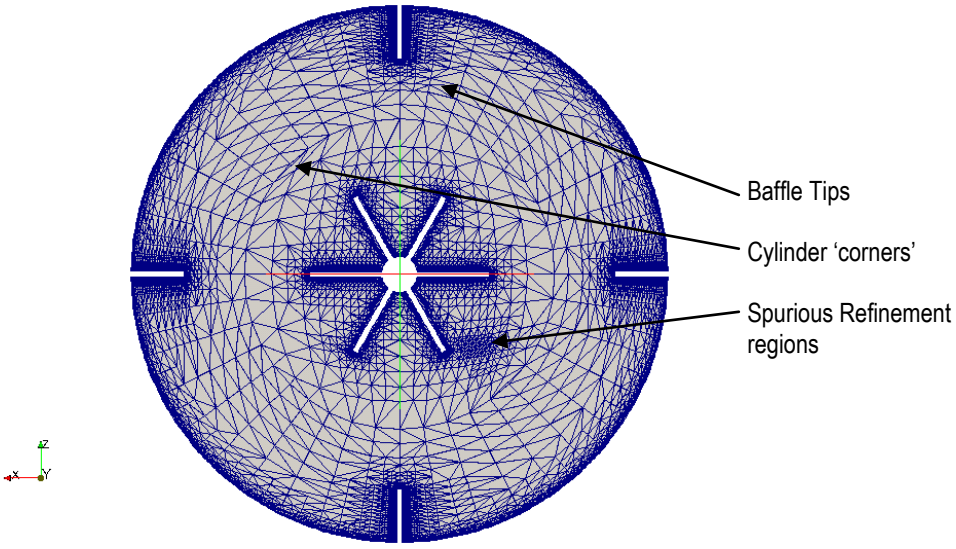


Fig 13 Mesh Errors

Both the spurious regions and the baffle tips were resolved for some meshes so methods for refining the mesh at the cylinder 'corners' were considered. These errors arise since the grading at the curved edges forces these elements to be highly skewed. It would be desirable to seed these edges with a graded mesh however since only non-symmetrical grading along an edge was identified this was not possible. Since it was necessary to use this block approach to creating the cylinder in order to ensure face matching, a different *blockMesh* method was not found and only a few successful meshes could be achieved.

3.5 Parallel Processing

In the time available the high number of mesh operations was achieved by using parallel processing. By dividing the geometry between four cores the time for each mesh and simulation was reduced by a quarter to ~2 hrs. The geometry was divided into four equal segments, one of the top segments is shown as **figure 14.**

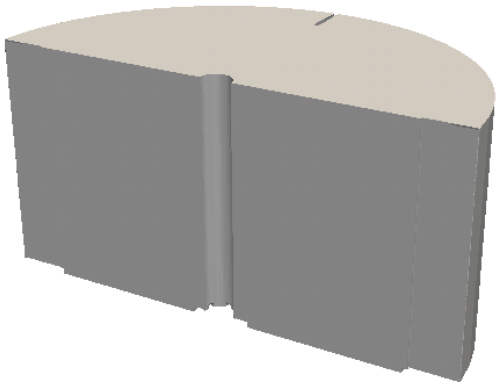


Fig 14 Segment from parallel processing

The required rotations, translations and two *snappyHexMesh* operations were written into a *./Allrun* file, shown below:

```
#!/bin/sh
# Source tutorial run functions
. $WM_PROJECT_DIR/bin/tools/RunFunctions
blockMesh
rotateMesh "(0 0 1)" "(0 1 0)"
transformPoints -translate '(50 45 50)'
cp system/snappyHexMeshDict1 system/snappyHexMeshDict
snappyHexMesh -overwrite
transformPoints -translate '(-35 -42 -36.6879)'
cp system/snappyHexMeshDict2 system/snappyHexMeshDict
snappyHexMesh -overwrite
rm -rf 0
cp -r 0.orig 0
runApplication decomposePar
runParallel MRFSimpleFoam 4
runApplication reconstructPar
```

A further file was used in a directory one level higher than each mesh study to run one after another. This simplified the complex meshing study and allowed all the possible options to be considered. This approach was used throughout the project to simulate all subsequent studies.

4. Turbulence Model Study

As discussed in **section 2** the given 2D case uses the k-ε model. However this is not the best approach for such a model. Therefore the k-ε, k-ω and k-ωSST models were compared to optimise the RANS solution to this problem. For each assessment the rotational velocity was arbitrarily chosen as 10 rps

4.1 Standard k-epsilon model

Equations 1 and **2** are the governing equations for the k-epsilon model. (Tabor, Tutorial Sheet 1 2012)

$$k = \frac{2}{3}(U_{ref}T_i)^2 \; m^2 s^{-2}$$

Eqn 1.

$$\epsilon = C_\mu^{\frac{3}{4}} \frac{k^{3/2}}{0.07l} m^2 s^{-3}$$

Eqn 2.

The turbulent intensity T_i is given as a percentage. The length scale, l is the radius of the rotor and the value C_μ is a coefficient with the value of 0.09. For this case the reference velocity is the tangential velocity of the blade tips and the turbulent intensity was chosen as 5%.

The boundary condition, *kqRWallFunction* within the */0, k* file is an initial value for the turbulent energy based on this rule of thumb. These values were solved as shown below:

$$k = \frac{2}{3}(r\omega \times T_i)^2 = \frac{2}{3}(0.15 \times 2 \times \pi \times 10 \times 0.05)^2 = 0.148\, m^2 s^{-2} \text{ (4sf)}$$

Eqn 5.

$$\epsilon = (0.09)^{\frac{3}{4}} \times \frac{(0.148)^{3/2}}{0.07 \times 0.15} = 0.891\, m^2 s^{-3} \text{ (3sf)}$$

Eqn 6.

These were input into all the uniform values for *kqRWallFunction* in *0/k* and all values inside *0/epsilon* files respectively.

4.2 The k-ω model

Using the k-ω model is arguably a better approach than the k-ε method. This function resolves wall velocities far better, although it is more expensive in computer power. (CFD online 2010). This was borne out in the following simulations which required upwards of twice the computational time.

To move from the k-ε model to k-ω, the files were copied into a new directory and the entries for epsilon in the */0* file and the *fvSolution* files were changed to omega. The value the for wall functions was calculated using **equation 7.**

$$\omega = \frac{\epsilon}{k} = \frac{0.148}{0.891} = 0.166\ \text{s}^{-1}\ \text{(3sf)}$$

Eqn 7.

4.3 k- ωSST

The blended technique, k-ωSST was then employed in order to best use the available computational power. This can be achieved since the k-ωSST uses the k-ω model at the walls, best utilising its stability in the critical flow regions which are also the areas of worst mesh quality in this case. This model is blended with the k-ε model away from the wall. The *RASProperties* file was changed for the *komegaSST* case and all wall coefficients kept from the previous trials.

4.4 Differencing schemes

It was found that even for the more stable k-ω and k-ωSST options the simulation was too unstable with the given mesh quality. In order to solve this problem the differencing scheme was changed to upwind in each case to good effect. Changes were made the divergence entries in the *fvSchemes* file in each case. The changed entry is shown below for k-ω:

```
divSchemes
{
    default         none;
    div(phi,U)      Gauss upwind;
    div(phi,k)      Gauss upwind;
    div(phi,omega) Gauss upwind;
    div((nuEff*dev(T(grad(U))))) Gauss linear;
}
```

Upwind differencing maps the change in a quantity to the upwind centre of the adjoining face between cells. This can produce poorer flow patterns than higher order models, although it is known as the most stable. Errors with this simple scheme can be introduced, particularly for

skewed cells where faces are not at 90° to the line between cell centres. Nevertheless, since the system did not run with linear differencing this was the selected option. It was found that the divergence in the last term needed to be left as linear to allow for a complete solution. This change allowed for the models to be run successfully.

4.4 Comparison

In order to compare the turbulence models the velocity profiles on a surface on a radius equally spaced between the tip edges was considered since there are similar velocity plots available from Particle Emission Path Tracking (PEPT) experimental methods. The experimental plots are shown below in **figure 14** and the plots for the three turbulence models below at as **figure 15.**

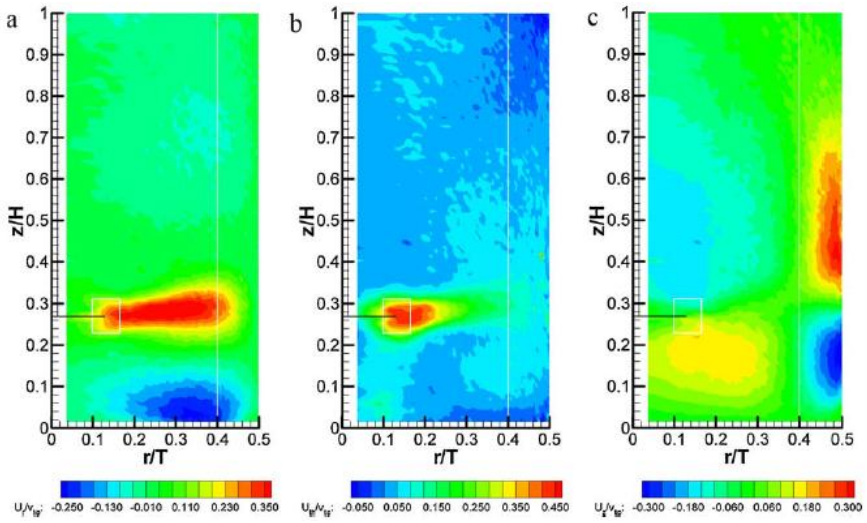


Fig 14 Velocity contours for PEPT methods Ur/Uimpellor, Utheta/Uimpellor, Uz/Uimpellor respectively (Chiti, et al. 2011)

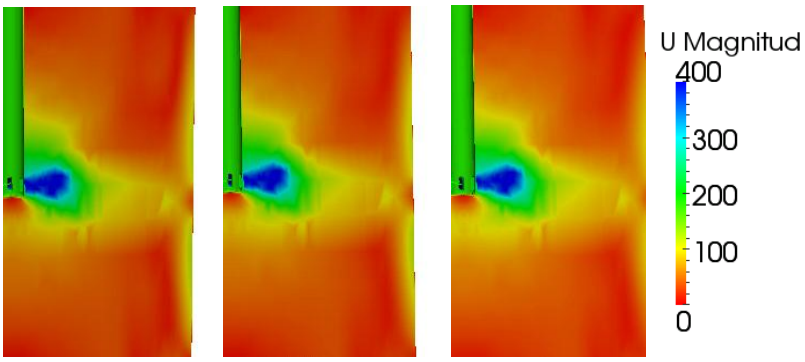


Fig 15. Magnitude of velocity profiles for turbulence models k- ω SST, k- ω and k-ε respectively

There are notable differences between the shapes of the profiles from the MRF model when compared to the experimental data. Firstly there is a high flow region shown in green above the impellor by the shaft which is not seen in the experimental data. That is because the impellor modelled here is a set of radial blades attached directly to the shaft. In the experimental conditions the blades are mounted on a horizontal plate which does not disturb the flow significantly near the shaft and thus this is not a product of the turbulence models.

The other differences can be accounted for by the fact that the three images from the experimental data include separate components of the velocity, the features of which can be identified in all the MRF models.

The moderately excited region below and to the right of the blade in the tangential velocity image in the experimental data is mirrored by the yellow region in the same area for each of the turbulence models. Additionally the vertical velocity profile at the wall surface shows a stagnated region at the impellor height seen in the third of the experimental images. This has been modelled in each of the turbulence models with the k-ε showing a slightly less well developed flow at this wall area, as expected due to its poor performance at wall surfaces.

The time taken to compute the different models was noted for each and is shown in **table 1.**

Model	Time (s)
k-ε	8752
k-ω	9659
k-ωSST	9874

Table 1 Computational Time

Table 1 shows that the computational time for the k-epsilon model is lower than for either k-ε model. The difference between k-ωSST and k-ω was negligible and may be accounted for by post-processing being carried out on the same computer while running the final model.

Since the k-ε model has been discounted due to its lower performances at wall surfaces it was deemed that the blended performance of the k-ωSST model for negligible computational time difference made it the best choice to model the turbine.

6. Validation of the 3D model

6.1 Velocity Profiles

To determine if the model accurately models the Rushton Turbine the velocity plots were compared to plots achieved with PIV. The experimental data was averaged both in the tangential and radial directions for a given vertical heights. An example of the data is shown in **figure 16.**

The radial velocity is plotted for different Reynolds numbers from the centre of the blades to the free stream region above the impellor. The velocities in this case are averaged both across the radius and for the full range of angles. This gives a value for the averaged velocity for a given height, preventing turbulent effects acting as noise on a plot across arbitrarily chosen length.

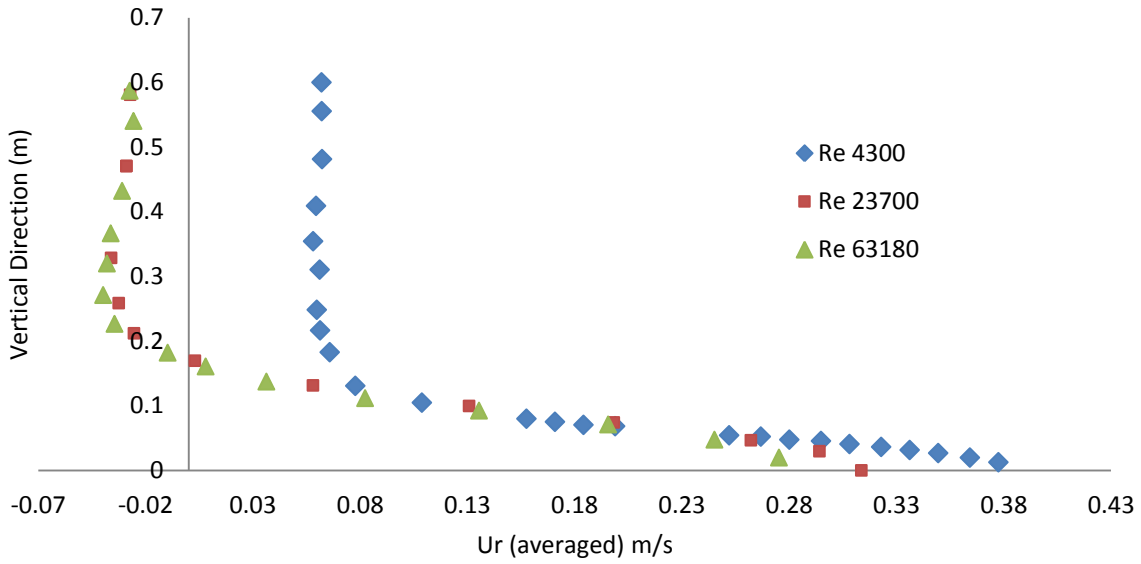


Fig 16 Average radial velocity plot (H.S.Yoon 2005)

The Reynolds number is given by **equation 8** (H.S.Yoon 2005)

$$Re = \frac{ND^2}{v}$$

Eqn 8

This relates the revolutions per second, N to the impellor diameter D and the kinematic viscosity. This was used to simulate the same Reynolds number for the k-ωSST case. The aim was to normalise this data with respect to blade-tip velocity and impellor radius. The results for the MRF model is shown as **figure 17.**

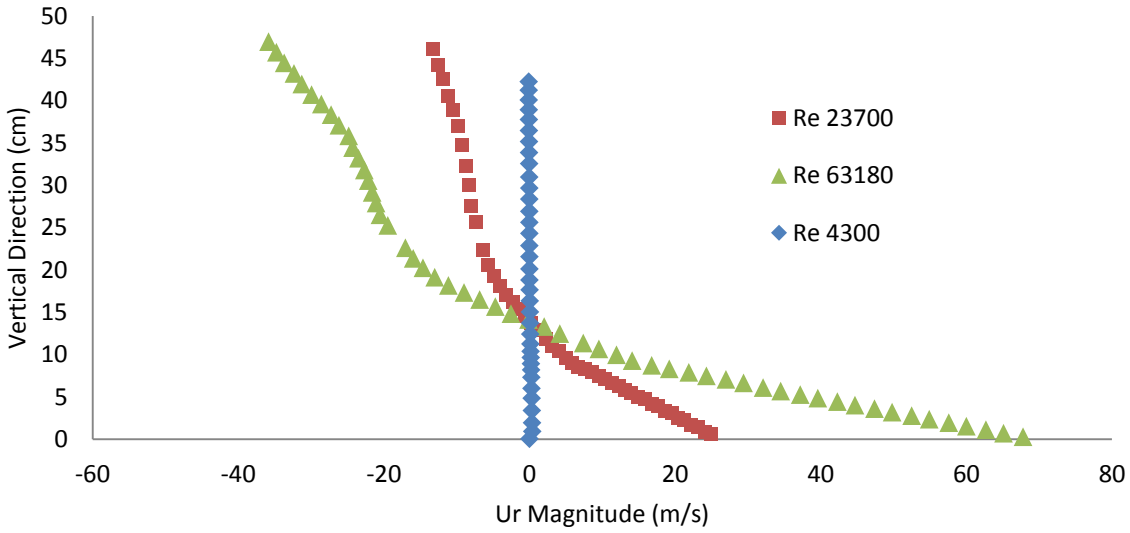


Fig 17 Radial velocities with vertical distance over a range of Reynolds numbers

This data was measured at a distance of 150mm from the tip edge, roughly half way between the baffle and the blade tip. However this did not match the averaged data from the PIV experiment.

This is not surprising since the high turbulence in this part of the flow causes fluctuations in the flow at any chosen location for the vertical plot. Without a known method of achieving this value from O*penFOAM;* the above data is the best fit to the experimental achieved.

In **figure 17** the same broad trends can be seen as in **figure 16** with the highest velocity at the height of the impellor which decays away to negative values at about 1/3 of the vertical direction. In this respect these trends mirror the PIV data well. This indicates that the model has accurately modelled the Rushton Turbine in three dimensions.

6.2 Visualisations

The turbine modelled introduces useful turbulence into the flow for mixing. This is demonstrated by the 400 stream tracers which track the velocity in **figure 18.**

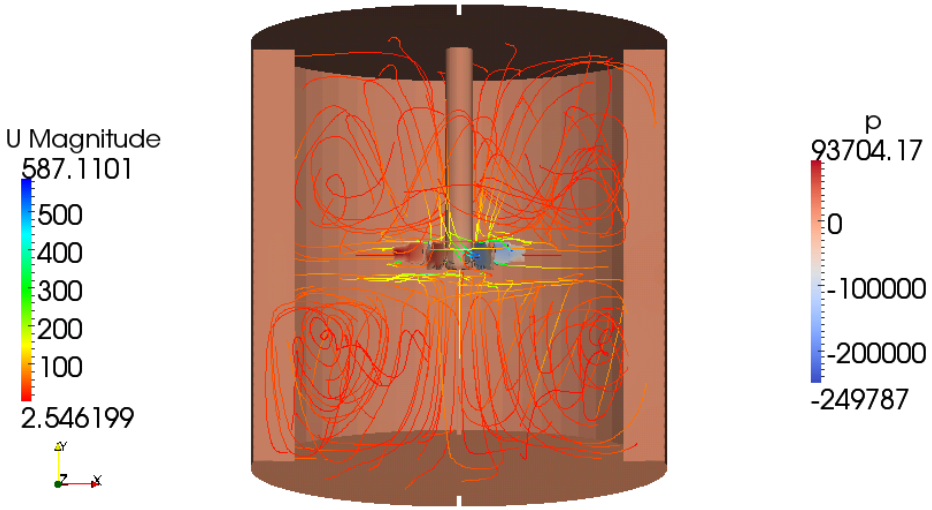


Fig 18 Stream Tracers Re 63180

The driving pressure, mapped here on the impellor blades, is causing turbulence in all the regions of the model.

To track individual streams, two plots are shown below in **figure 19.** These show four randomly generated streamlines from above and from the side. The vertical line in the right hand image shows fluid flow between the top and bottom of the model.

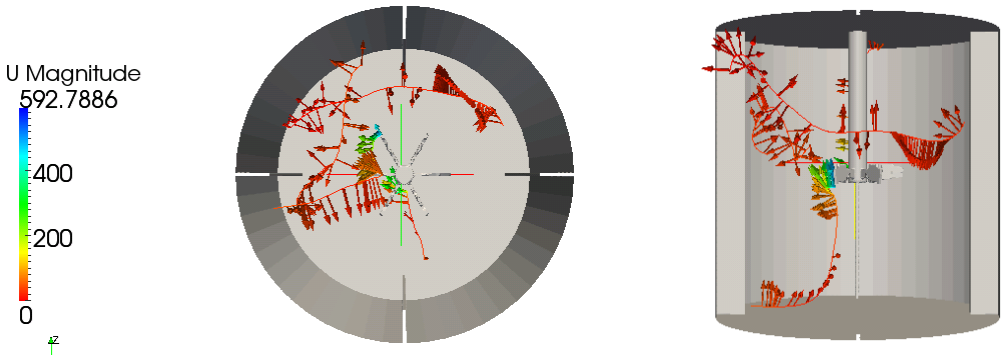


Fig 19 Stream Traces with velocity arrow glyphs

Equally the top view in the left image shows a streamline with rotational effects moving around the impellor. The glyph arrows were added, also graded with velocity contours to demonstrate the direction of the flow. These map the magnitude of the velocity, taking into account all constituent velocities and are thus at right angles to the stream tracer in each case.

7. In Summary

This report has demonstrated that the *MRFSimpleFoam* solver within the CFD code *OpenFOAM* is capable of modelling the Rushton Turbine in three dimensions. Utilising this Multiple Reference Frame technique it has been demonstrated in both two and three dimensions that the model can adequately model the velocity profiles and turbulent flow.

The methods used have indicated that the best technique to mesh this system is to utilise the *snappyHexMesh* utility within the code. This was achieved by using two different meshing operations, one for both the rotor and the stator. It is likely that this could be simplified by using only one mesh operation, including surfaces. This would require less mesh transformation to meet the surface file locations. Since *snappyHexMesh* retains the cell block names with two meshing operations it should perform equally well with a more efficient single method.

The RANS turbulence study showed that either k-ω or k-ωSST are suitable turbulence models providing that upwind differencing is used to ensure model stability. Possible extensions to this study include more extensive validation of the 3D model using velocity averaging techniques and multiphase mixing. Additionaly non-RANS turbulence models such as Large Eddy Simulation could be used for comparison. However the extensive mesh study in this report has precluded these.

It is clear that the Multiple Reference frame Approach is a satisfactory technique for modelling this problem. It allows for full realisation of the geometry from blade to baffle without requiring the expensive moving mesh techniques. However this method is limited to sudo-steady state results since it cannot measure the effects of the blades passing the baffles. Nevertheless it provides experimentally verifiable results and can be used to inform design decisions since a good estimate for the flow patterns can be realised. CFD is a balance of quality of results to computational expense and it is compromise methods such as this which are invaluable to industry.

Bibliography

Arlov, D, J Revstedt, and L Fuchs. "Numerical simulation of a gas–liquid Rushton stirred reactor - LES and LPT." *Computers and Fluids*, 2008: 793-801.

CFD Online. *CFD Online: mergePatchPairs error.* 22 06 2011. http://www.cfd-online.com/Forums/openfoam-meshing-blockmesh/82736-mergepatchpairs-error-please-help-me-out.html (accessed 03 20, 2012).

CFD online. *Use of k-epsilon and k-omega Models.* 28 04 2010. http://www.cfd-online.com/Forums/main/75554-use-k-epsilon-k-omega-models.html (accessed 03 28, 2012).

Chiti, Fabio, Serafim Bakalisa, Waldemar Bujalskia, Mostafa Barigoua, Archie Eagleshamb, and Alvin W. Nienowa. "Using positron emission particle tracking (PEPT) to study the turbulent flow in a baffled vessel agitated by a Rushton Turbine: Improving the data treatment and validation." *Chemical Engineering Research and Design*, 2011: 1947-1960.

Dr Hrvoje Jasak. *BlockMeshMergePatchPairs trouble 2 CFD online.* 19 12 2006. http://www.cfd-online.com/Forums/openfoam-meshing-blockmesh/61837-blockmeshmergepatchpairs-trouble-2-a.html (accessed 03 19, 2012).

H.S.Yoon, D.F. Hill, S. Balachandar, R.J. Adrian, M.Y. Ha. "Reynolds number scaling of flowin a Rushton turbine stirred tank. Part I - Mean Flow, circular jet and tip vortex scaling." *Chemical Engieering Science*, 2005: 3169-3183.

J. Kim, S.J Kline, and J. P. Johnston. "Investigation of a reattaching shear layer: Flow over a backwards facing step." *J. Fluids Engineering*, 1980: 102:302.

Montante, G., and F. Magelli. "Liquid Homogenization Characteristics in Vessels Stirred with Multiple Rushton Turbines Mounted at Different Spacing's. CFD Study and Comparison With Experimental Data." *Chemical Engineering Research and Design*, 2004: 1179-1187.

Ng, K., N. J. Fentiman, K. C. Lee, and M. Yianneskis. "Assessment of sliding mesh CFD predictions and LDA measurements of the flow in a tank stirred by a Rushton Impeller." *Trans IChemE Vol 76*, 1998: 737-747.

Ranade, V. V. "An efficient computational model for stimulating flow in stirred vessels: a case of Rushton Turbine." *Chemical Engieering Science Vol 52*, 1997: 4473-4484.

Ranade, V. V., M. Perrade, N. LE Sauze, C. Xuereb, and J. Bertrand. "Trailing Vorticies of Rushton Turbine: PIV Measurements and CFD Simulations with Snapshot Approach." *Trans IChemE Vol 79*, 2001: 3-12.

Tabor, Dr Gavin. *Tutorial Sheet 1.* Tutorial Sheet, Exeter: Exeter University, 2012.

Yapici, Kerim, Bulent Karasozen, Michael Schafer, and Yusuf Uludag. "Numerical investigation of the effect of the Rushton type turbine design factors on agitated tank flow characteristics." *Chemical Engineering and Processing*, 2008: 1340-1349.