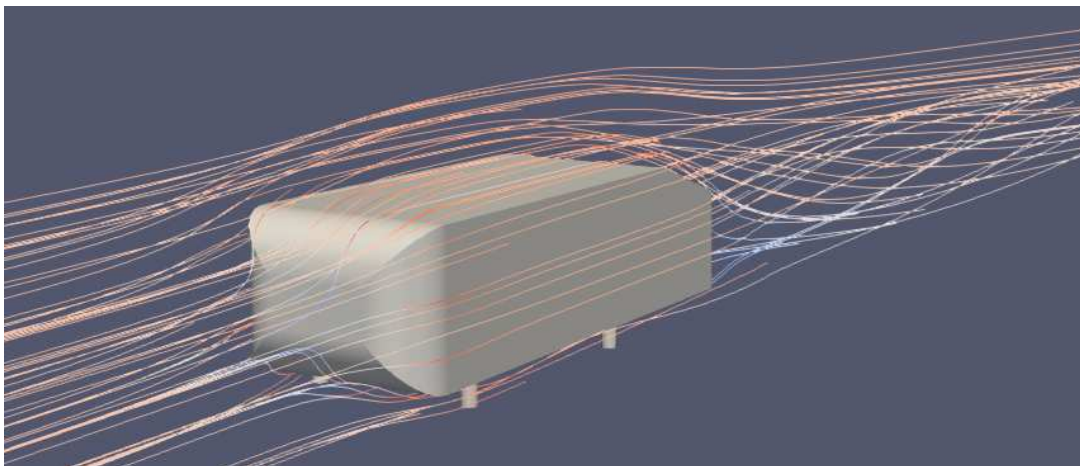




**Northumbria
University**
NEWCASTLE

KB6003 Vehicle Aerodynamics

Tutorial 4



Ahmed Body Solution

Date: 30th October 2018

Contents

TABLE OF FIGURES	2
TABLE OF TABLES	2
1 INTRODUCTION	3
1.1 SIMPLEFOAM.....	3
2 DICTIONARIES FOR SOLUTION	3
2.1 FLUID PROPERTIES (TRANSPORTPROPERTIES).....	4
2.2 TURBULENCE MODEL (TURBULENCEPROPERTIES)	4
2.3 CONTROLDICT	5
2.4 NUMERICAL SCHEMES ('FVSCHEMES').....	6
2.5 SOLUTIONS AND ALGORITHM CONTROL (FVSOLUTION).....	9
2.6 BOUNDARY CONDITIONS IN POLYMESH.....	10
2.7 THE "O_ORG" FOLDER	11
2.8 ESTIMATING THE INITIAL CONDITIONS FOR K AND OMEGA	11
3 PLOTTING RESIDUALS	16
4 RUNNING THE SOLVER (SOLUTION)	18
5 INITIAL POST PROCESSING.....	18
5.1 GENERAL 'U' VISUALISATION.....	18
5.2 TRACE STREAMLINES.....	21
5.3 WALL SHEAR STRESS.....	23

Table of Figures

Figure 2.1: Transport properties.....	4
Figure 2.2: Chosen model.....	4
Figure 3.1: Residual plots.....	16
Figure 5.1: Change the active variable to U.	19
Figure 5.2: Loading the case.....	19
Figure 5.3: Select 'Stream Tracer' from 'Common' toolbar.....	21

Table of Tables

Table 2.1: OpenFOAM case structure (pre-solution).....	3
Table 2.2: Customisation of 'controlDict' for Ahmed body case.	5
Table 2.3: Customisation of 'fvSchemes' for Ahmed body case.....	7
Table 2.4: Customisation of 'fvSolutions' for Ahmed body case.....	9
Table 2.5: Customisation of '0_org' folder dictionaries for Ahmed body case.	12
Table 5.1: Clippings of velocity ('U').	20
Table 5.2: Streamline parameters.	22
Table 5.3: Traced streamlines.....	22
Table 5.4: Wall shear stresses.	24

1 Introduction

OpenFOAM, as described previously, is a powerful utility to solve partial differential equations. It has continuum mechanics, probability, financial and many other applications. It does this via a custom set of solvers. For this tutorial, 'simpleFoam' solver will be used. It is a steady-state solver for incompressible flow with turbulence modelling. A complete list of 'standard' solvers with their description can be seen at <https://www.openfoam.com/documentation/user-guide/standard-solvers.php>.

This tutorial will look into setting up the case for the 'simpleFoam' solver to run (solve) on the previously generated mesh. The subsequent sections will look into the dictionaries that govern the solution, running the solver and some basic post processing.

1.1 simpleFoam







This steady state solver uses the 'simple' (Semi-Implicit Method for Pressure-Linked Equations) to solve the Navier Stokes equations. The algorithm is an iterative process and follows the basic steps described below as per [https://openfoamwiki.net/index.php/OpenFOAM_guide/The SIMPLE algorithm in OpenFOAM](https://openfoamwiki.net/index.php/OpenFOAM_guide/The_SIMPLE_algorithm_in_OpenFOAM).

- 1) *Set the boundary conditions.*
- 2) *Solve the discretized momentum equation to compute the intermediate velocity field.*
- 3) *Compute the mass fluxes at the cells faces.*
- 4) *Solve the pressure equation and apply under-relaxation.*
- 5) *Correct the mass fluxes at the cell faces.*
- 6) *Correct the velocities on the basis of the new pressure field.*
- 7) *Update the boundary conditions.*
- 8) *Repeat till convergence.*

2 Dictionaries for Solution

This sections will look into the dictionaries (controlDict, fvSchemes, fvSolutions, k, p, U, nut, omega, transportProperties and turbulenceProperties) governing the solution. The table 2.1 below shows the location of these dictionaries.

Table 2.1: OpenFOAM case structure (pre-solution).

 case	Folder	Description
	Case	Main case directory
 0_orgs	Constant	Contains the physical properties and geometries for the case. Also contains the mesh when generated.
 constant	polyMesh	Contains the mesh after it is generated.
 polyMesh	triSurface	Contains the geometries (STLs, VTKs, eMesh, etc.)
 triSurface	ExtendedFeatureEdgeMesh	Extracted features.
 system	system	Contains dictionaries used to control and run the case (such as blockMesh, controlDict, snappyHexMeshDict, surfaceFeatureExtractDict, meshQualityDict, etc.)

2.1 Fluid Properties (transportProperties)

This dictionary is located in the '<case>/constant' folder and has the properties of the fluid. This should already be set up as a Newtonian fluid with dynamic viscosity (ν) set to 1.5×10^{-5} as shown in figure 2.1 below.

```
/*----- C++ -----*/
|=====|
| \ \ / / | F i e l d | OpenFOAM: The Open Source CFD Toolbox
| \ \ / / | O p e r a t i o n | Version: 4.x
| \ \ / / | A n d | Web: www.OpenFOAM.org
| \ \ / / | M a n i p u l a t i o n |
|=====|
FoamFile
{
    version      2.0;
    format       ascii;
    class        dictionary;
    object       transportProperties;
}
// *****

transportModel Newtonian;

nu              nu [0 2 -1 0 0 0] 1.5e-05;

// *****
```

Figure 2.1: Transport properties.

2.2 Turbulence Model (turbulenceProperties)

This dictionary is located in the '<case>/constant' folder and has the chosen turbulence model. More on this would be covered in subsequent lectures of this module. The model chosen for this simulation is the k-Omega SST model and can be seen in the figure 2.2 below.

```
/*----- C++ -----*/
|=====|
| \ \ / / | F i e l d | OpenFOAM: The Open Source CFD Toolbox
| \ \ / / | O p e r a t i o n | Version: 4.x
| \ \ / / | A n d | Web: www.OpenFOAM.org
| \ \ / / | M a n i p u l a t i o n |
|=====|
FoamFile
{
    version      2.0;
    format       ascii;
    class        dictionary;
    object       turbulenceProperties;
}
// *****

simulationType RAS;

RAS
{
    RASModel      kOmega;

    turbulence     on;

    printCoeffs    on;
}

// *****
```

Figure 2.2: Chosen model.

2.3 controlDict

OpenFOAM sets up databases for its solutions which are governed by 'controlDict'. It controls the time, data input and output of the solution as well as additional libraries loaded at run time. It can be found under '<case>/system/' folder.

Navigate to your Ahmed body 'case/system' and edit the 'controlDict' file to reflect the parameters in the table 2.2 below.

Table 2.2: Customisation of 'controlDict' for Ahmed body case.

Parameter	Value	Description
application	simpleFoam	The solver to be used.
startFrom	latestTime	The start time of the simulation. This could be a subsequent run hence 'latestTime' resumes from the last time step.
startTime	0	Initially start at time.
stopAt	endTime	When to stop the simulation.
endTime	1000	End time for simulation (not real time but internal simulation) when 'stopAt' is given an 'endTime'.
deltaT	1	Time steps for the simulation.
writeControl	timeStep	Data output writing clock (runtime, CPU time, clock time, etc.)
writeInterval	50	Interval for data writing.
purgeWrite	0	Overwrite (cyclic) time directories (value represents number of directories to keep).
writeFormat	ascii	Format of data files.
writePrecision	8	Precision to decimal places
writeCompression	off	To compress with gzip when data is written (on/off). <u>Tip:</u> yes/no, true/false can be used as well.

timeFormat	general	Format for naming time directories. 'general' uses scientific format if exponent is less than -4 or greater than the 'timePrecision' value.
timePrecision	6	Precision to number of decimal places (default is 6)
runTimeModifiable	True	Re-reads the dictionary during the start of each simulation time step. Enabling this allows users to change parameters while during simulation. Eg: Stop simulation after finishing current step (instead of terminating with 'Ctrl+c').
functions { ... }	<leave blank>	Dictionary for functions (such as probes and in-simulation processing).

The complete version of the controlDict file for the Ahmed body case can be found in 'Tutorial3/end.zip/ahmed3' on Github (<https://github.com/NU-Aero-Lab/OpenFOAM-Cases/tree/master/KB6003/Tutorial3>).

2.4 Numerical Schemes ('fvSchemes')

This dictionary sets the numerical schemes for terms used and calculated in the simulation. It is located under '<case>/systems' folder and is further divided by keywords representing terms of particular type such as;

- timeScheme: first and second time derivatives, e.g. $\partial/\partial t, \partial^2/\partial t^2$
- gradSchemes: gradient ∇
- divSchemes: divergence $\nabla \cdot$
- laplacianSchemes: Laplacian ∇^2
- interpolationSchemes: cell to face interpolations of values.
- snGradSchemes: component of gradient normal to a cell face.
- wallDist: distance to wall calculation, where required.

More information can be found on: <https://cfd.direct/openfoam/user-guide/v6-fvschemes/>

Navigate to your Ahmed body case and edit the 'fvSchemes' file to reflect the parameters in the table 2.3 below.

Table 2.3: Customisation of 'fvSchemes' for Ahmed body case.

Parameter	Value	Description
ddtSchemes		Time schemes ($\partial/\partial t$).
default	steadyState	Equates time derivative to zero. Note: Make sure the solver is compatible. i.e. running a transient scheme on a steady state solver (such as 'simpleFoam') would give incorrect results.
gradSchemes		Gradient terms.
default	Gauss linear	This scheme gives more accurate results than Gauss upwind.
grad(U)	cellLimited Gauss linear 1	Overridden to clip each component of the gradient equally.
divSchemes		Divergence terms.
default	none	
div(phi,U)	bounded Gauss linearUpWindV grad(U)	Second order, upwind-biased, bounded, with discretisation of the velocity gradient specified.
div(phi,k)	bounded Gauss upwind	First-order bounded.
div(phi,omega)	bounded Gauss upwind	First-order bounded.
div((nuEff*dev2(T(grad(U)))))	Gauss linear	Second order, unbounded.
laplacianSchemes		Laplacian terms.

default	Gauss linear corrected	Second order, unbounded with correction (see 'snGradSchemes').
interpolationSchemes		Interpolated values of cell centres to face centres.
default	linear	Second order, unbounded.
snGradSchemes		Surface normal gradient terms. Required to evaluate a Laplacian term using Gaussian intergration.
default	corrected	Non-orthogonal correction to maintain second-order accuracy.
wallDist		Calculates distance to walls and normal to walls fields.
method	meshWave	Method for calculating distance to nearest patch for all cells and boundary.
correctWalls	true	Correction distance from near-wall cells to boundary.

The complete version of the controlDict file for the Ahmed body case can be found in 'Tutorial3/end.zip/ahmed3' on Github (<https://github.com/NU-Aero-Lab/OpenFOAM-Cases/tree/master/KB6003/Tutorial3>).

More information on 'fvSolution' can be found on <https://cfd.direct/openfoam/user-guide/v6-fvSchemes>.

2.5 Solutions and Algorithm Control (fvSolution)

The equations solvers and the algorithms along with tolerances are controlled via this dictionary. It is located under '<case>/systems' folder. More information on 'fvSolution' can be found on <https://cfd.direct/openfoam/user-guide/v6-fvSolution/>.

Navigate to your Ahmed body case and edit the 'fvSolutions' file to reflect the parameters in the table 2.4 below.

Table 2.4: Customisation of 'fvSolutions' for Ahmed body case.

Parameter	Value				Description
solvers	p	U	k	omega	Solvers used and parameters for each discretised equation
solver	GAMG	smoothSolver			GAMG: generalised geometric-algebraic multi-grid. smoothSolver: solver that uses a smoother.
smoother	GaussSeidel				Option for solvers using smoother.
tolerance	1e-7	1e-8	1e-8	1e-8	Tolerance level at which the solution is deemed of sufficient accuracy.
relTol	0.01	0.1	0.1	0.1	Relative improvements from initial to final solution.
nSweeps	<N/A>	1	1	1	Number of sweeps before residual is recalculated (default 1).
phi					The phi term.
\$p	<blank>				
SIMPLE					The solver algorithm used
nNonOrthogonalCorrectors	0				Used to update the explicit non-orthogonal correction of the Laplacian terms.

consistent	yes					
residualControl	p	U	nuTilda	k	omega	
	1e-4					Simulation deemed as converged when residuals falls below this value.
potentialFlow						Potential flow algorithm controls
nNonOrthogonalCorrectors	10					See ‘SIMPLE’
relaxationFactors						Under-relaxation of field factors.
equations	U	k	omega			
	0.9	0.7	0.7			
cache						Cache-ing
grad(U)						Instant solving speed boost but a slight increase in memory usage.

The complete version of the controlDict file for the Ahmed body case can be found in 'Tutorial3/end.zip/ahmed3' on Github (<https://github.com/NU-Aero-Lab/OpenFOAM-Cases/tree/master/KB6003/Tutorial3>).

2.6 Boundary conditions in Polymesh

At this stage, the solution is almost ready to be run. The few additional steps below are required prior to initiating the solution. This is due to method of mesh generation used i.e. not overwriting the mesh generation steps.

The mesh generated in the previous tutorial is in multiple stages with the final folder containing the generated mesh in full. This needs to be moved to the '<case>/constant/polyMesh'.

Before moving, the 'ground' parameter in the 'boundary' dictionary inside the last time-step '4/polyMesh' needs to be changed from 'patch' to 'slip' as no boundary conditions would be applied to the ground.

Once changed, please enter the following CLI command to move the final mesh and delete the intermediate stages;

```
# Remove existing 'polyMesh' in 'constant' directory
rm -rf constant/polyMesh/

# Copy the desired mesh to the 'polyMesh' in 'constant' directory
# *Please Note: There are spaces before and after '-r' and before 'constant/'
cp -r 4/polyMesh/ constant/
```

```
# Remove intermediate stages (steps) of mesh generation
foamListTimes -rm
```

2.7 The “0_org” Folder

This folder contains the initial conditions for the solution such as the velocity, the interactions with walls, etc. It is located in the main ‘<case>’ folder and contains the ‘k’. ‘nut’. ‘omega’, ‘p’ and ‘U’ dictionaries. It describes the dimensions, fields (internal and boundary) along with their parameters where the said conditions are applied.

Navigate to your Ahmed body case and edit the files under the ‘0_org’ folder to reflect the parameters in the table 2.5 overleaf.

2.8 Estimating the Initial Conditions for k and omega

The initial values for ‘k’ and ‘omega’ can be estimated using calculators such as <https://www.cfd-online.com/Tools/turbulence.php>.

Please navigate to the link above and enter the selected ‘Freestream velocity’ (33 m/s in this case), the ‘Turbulence intensity/level Tu’ as 1% and ‘Eddy viscosity ratio μ_t / μ ’ as 1.

Select the ‘Turbulence variables (k, ϵ , ω) from turbulence intensity (Tu), eddy viscosity ratio (μ_t/μ), freestream velocity (U_∞) and kinematic viscosity (ν)’ option and click ‘Convert’.

This should now present you with values for ‘k’ and ‘omega’ to be included in the ‘k’ and ‘omega’ dictionaries in section 2.7.

Table 2.5: Customisation of '0_org' folder dictionaries for Ahmed body case.

Parameter	Value					Description
Dictionary:	k	nut	omega	p	U	The respective dictionary for k-Omega model.
dimensions	[0 2 -2 0 0 0 0]	[0 2 -1 0 0 0 0]	[0 0 -1 0 0 0 0]	[0 2 -2 0 0 0 0]	[0 1 -1 0 0 0 0]	
internalField	uniform 0.1666	uniform 0	uniform 10300	uniform 0	uniform 30 0 0	The initial values of the internal field.
boundaryField	<See below>					
inlet						Inlet patch.
type	fixedValue	calculated	fixedValue	zeroGradient	fixedValue	Type of the value i.e. fixed, calculated, function, slip, etc.
value	uniform 0.1666	calculated	uniform 10300	<N/A>	uniform 33 0 0	The value, see section 2.8.
outlet						Outlet patch.

type	inletOutlet	calculated	inletOutlet	fixedValue	inletOutlet	Type of the value i.e. fixed, calculated, function, etc.
inletValue	uniform 0.1666	<N/A>	uniform 10300	<N/A>	uniform 0 0 0	The value, see section 2.8.
value	uniform 0.1666	uniform 0	uniform 10300	uniform 0	uniform 33 0 0	The value, see section 2.8.
ground						Ground patch.
type	slip					Slip as no boundary conditions applied.
side.*						Sides patch (both).

type	symmetry					Patch which uses the symmetry plane (slip) condition.
Top						Top patch.
type	symmetry					Patch which uses the symmetry plane (slip) condition.
ahmed1						The main body.
type	kqRWallFunction	nutkWallFunction	omegaWallFunction	zeroGradient	fixedValue	Wall functions applied with fixed and zeroGradient conditions applied.
value	uniform 0.1666	uniform 0	uniform 10300	<N/A>	uniform 0 0 0	Values.

ahmed2						The supports.
type	kqRWallFunction	nutkWallFunction	omegaWallFunction	zeroGradient	fixedValue	Wall functions applied with fixed and zeroGradient conditions applied.
value	uniform 0.1666	uniform 0	uniform 10300	<N/A>	uniform 0 0 0	Values.

The complete version of the controlDict file for the Ahmed body case can be found in 'Tutorial3/end.zip/ahmed3' on Github (<https://github.com/NU-Aero-Lab/OpenFOAM-Cases/tree/master/KB6003/Tutorial3>).

Finally, the finished dictionaries in '0_org' needs to be copied to the '0' (initial) folder. This can be done via the following CLI commands;

```
# Copy 0_org folder to the 0 folder
cp -r 0_org 0
```

You should now be able to run the simulation using the CLI detailed in section 4. However, it is advisable to add the residuals plot to monitor your simulation as described in section 3.

3 Plotting Residuals

Although not a mandatory step, it is advisable to plot residuals for observation during the simulation. This would let one ascertain if the solution is converging or not. A solution not converging could mean incorrect solution set-up or a low quality mesh. These plots are crucial for longer simulations (spanning hours, days and more) as errors could be spotted and rectified at an earlier stage. The figure 3.1 below shows some examples of residual plots.

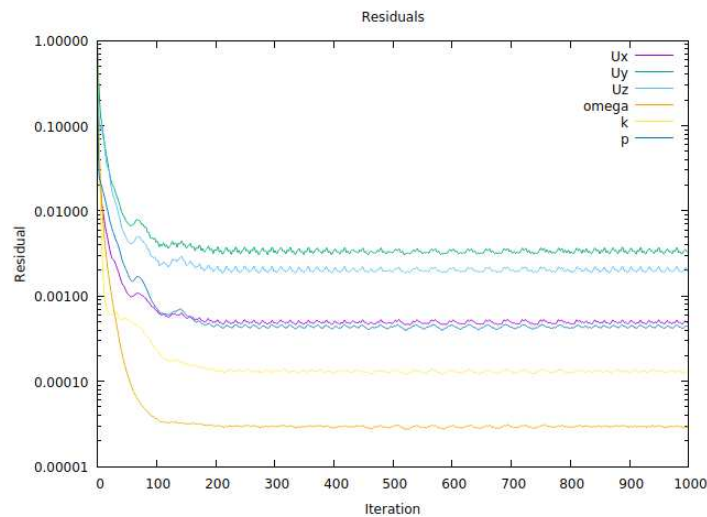


Figure 3.1a: None convergence

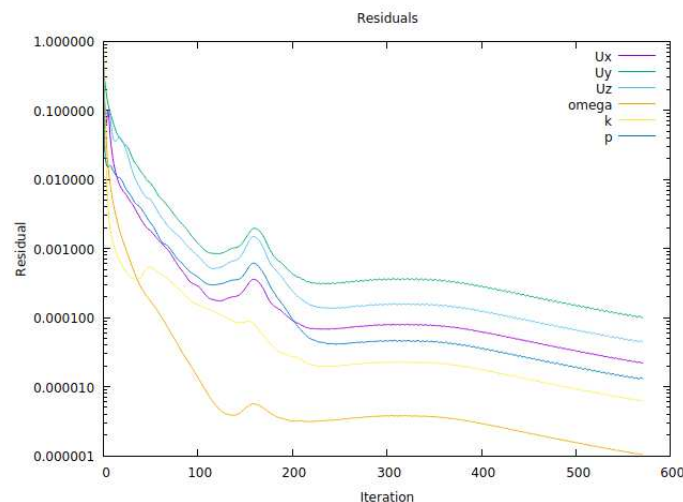


Figure 3.1b: Convergence

Figure 3.1: Residual plots.

In order to plot residuals, a residual script must be created and then initiated during runtime. To create the script, follow the steps below;

- 1) Open a 'Terminal' in your main case directory either by;
 - a. Navigating to the folder in 'File Manager', right clicking on an empty space and selecting 'Open in Terminal'.

- b. Navigate using CLI to the case folder;

```
cd $FOAM_RUN/<case>
```

- 2) Create a text file called 'residuals' in the case folder by typing;

```
gedit residuals .
```

The above CLI tells 'gedit' (the text editor, similar to 'Notepad' in Windows) to create a blank file called 'residuals' in the current ('.') directory.

Tip: alternatively, the text editor could be opened via GUI, then the file saved in the case folder after editing.

- 3) Insert the following code into the blank 'residuals' file.

```
set logscale y
set title "Residuals"
set ylabel 'Residual'
set xlabel 'Iteration'
plot "< cat LOGNAME | grep 'Solving for Ux' | cut -d ' ' -f9 | tr -d ',' title 'Ux' with lines,\
"< cat LOGNAME | grep 'Solving for Uy' | cut -d ' ' -f9 | tr -d ',' title 'Uy' with lines,\
"< cat LOGNAME | grep 'Solving for Uz' | cut -d ' ' -f9 | tr -d ',' title 'Uz' with lines,\
"< cat LOGNAME | grep 'Solving for omega' | cut -d ' ' -f9 | tr -d ',' title 'omega' with lines,\
"< cat LOGNAME | grep 'Solving for k' | cut -d ' ' -f9 | tr -d ',' title 'k' with lines,\
"< cat LOGNAME | grep 'Solving for p' | cut -d ' ' -f9 | tr -d ',' title 'p' with lines
pause PAUSETIME
reread
```

- 4) Change the LOGNAME and PAUSETIME to the logname and the pause time (refreshing time of the graph) of your choosing (such as 'log.simplefoam' and '2').

Note: It is advisable to associate logs with tasks such as solver used (log.simpleFoam), stage (log.SnappyHexMesh), etc. This will make debugging easier.

- 5) Click 'Save' or press 'Ctrl+s' and quit the program.

Note: If the 'Terminal' does not return to normal state, please type 'Ctrl+c' on it. This will terminate the gedit. Please ensure you have saved the file before this.

4 Running the Solver (Solution)

The solution is now ready to be run. It can be done using the following CLI commands;

```
# Run the potential solver to get an initial solution
potentialFoam -initialiseUBCs -noFunctionObjects

# Run the simpleFoam solver and generate 'log.simplefoam' then show the tail of the log file
simpleFoam > log.simplefoam | tail -f log.simplefoam

# Plot residuals (open a new Terminal)
gnuplot -residuals
```

5 Initial Post Processing

This section will look into visualising simulation data in ParaView and where needed, extract data from the finished simulation for further analysis. This tutorial will cover some basic post processing such as general visualisation, tracing streamlines and wall shear stress.

In order to proceed, please open ParaView from the simulated case by typing;

```
ParaFoam -builtin
```

Note: There is a space before '-builtin'.

5.1 General 'U' Visualisation

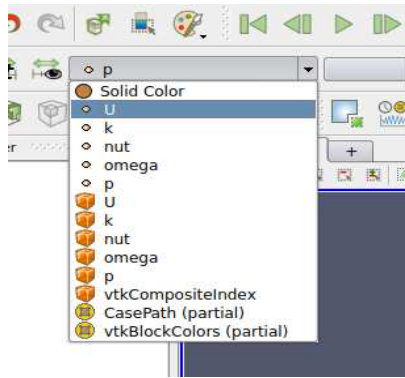
This is a general (basic) view (slice, clip, etc.) of the simulation which would provide information on the intensity of 'U'. It is similar to checking the mesh in tutorial 2 and is outlined in the steps below.

- 1) Make sure the 'internalMesh' and 'Reconstructed Case' is selected from the 'Properties' tab and select 'Apply' (figure 5.2 below).
- 2) Make a 'Slice' or a 'Clip' from the 'Common' toolbar (similar to mesh visualisation in tutorial 2).
- 3) Select the desired normal-to-axis, the co-ordinates and click apply.

Tip: To get a cross section from the middle of the legs, enter 0 or 0.1635 as the 'y' co-ordinate of the clipping.

Tip: For bottom, select 'z-normal' and enter 0.049 as the 'z' co-ordinate with 'Inside Out' selected in clipping.

- 4) Change the active variable to 'U' (velocity) from the 'Active Variable Control' toolbar (figure 5.1).



U - The general velocity.

U - The velocity per mesh cell.

Figure 5.1: Change the active variable to U.

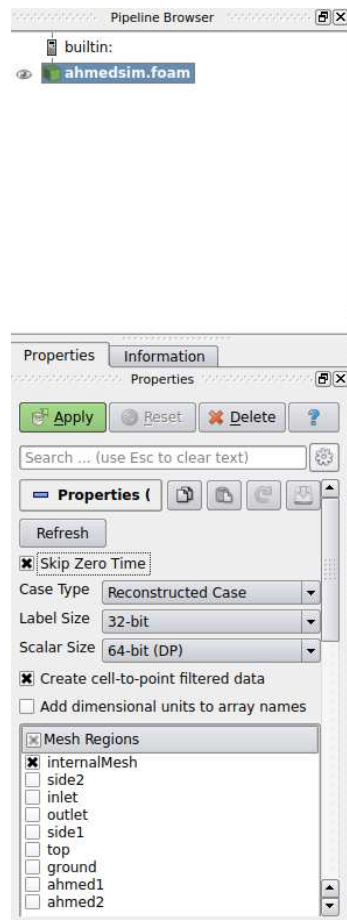


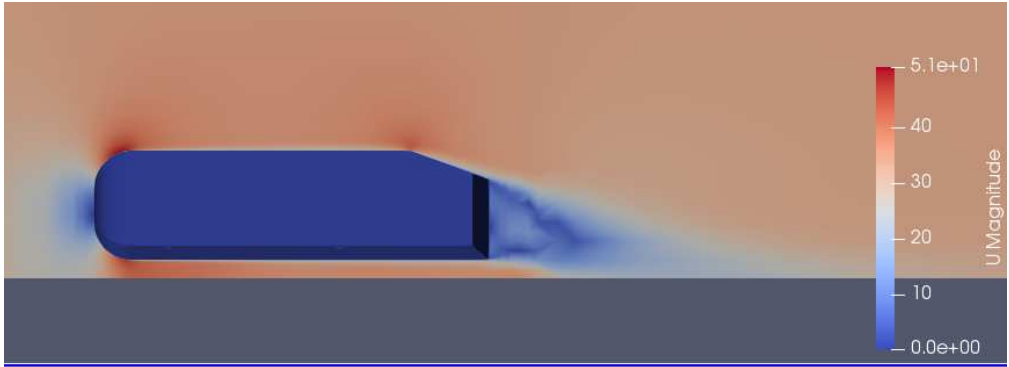
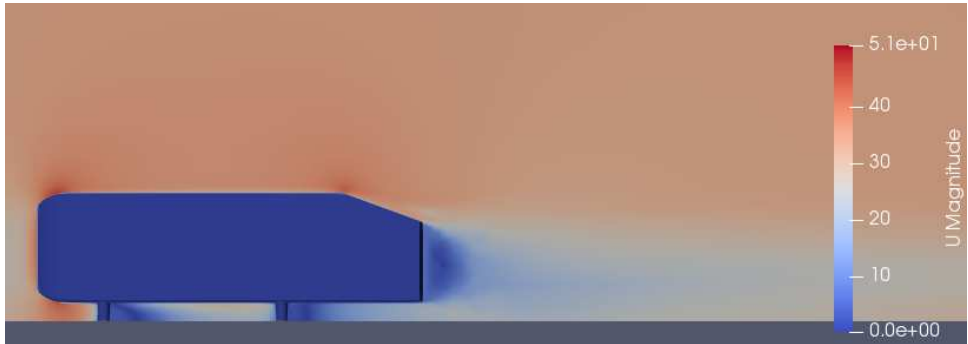
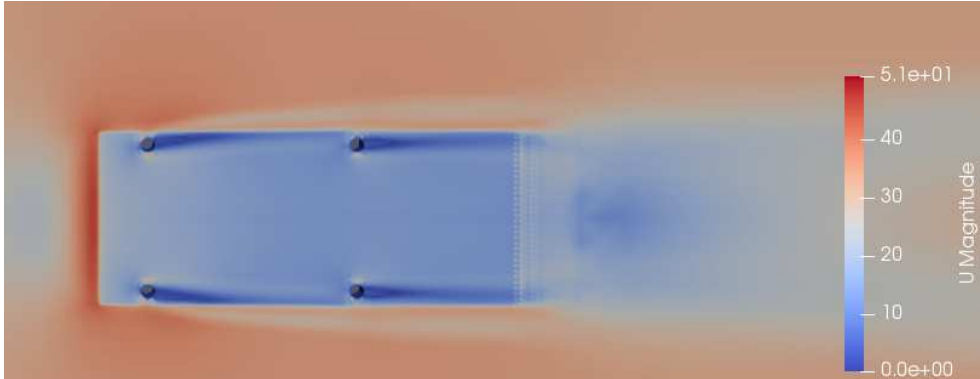
Figure 5.2: Loading the case.

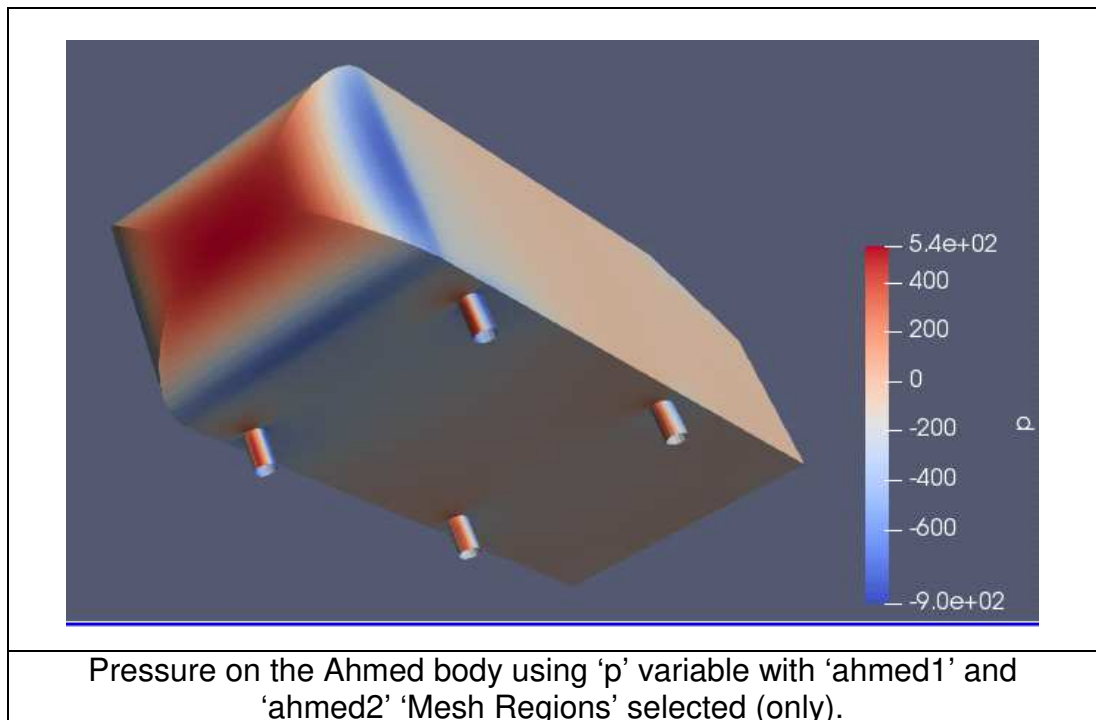
- 5) You can change the representation to 'Surface' or 'Surface with Edges' to suite your view from the 'Representation Toolbar'.
- 6) You can now see the velocity with intensity in different time steps using the 'VCR Controls'.

Tip: You can play the simulation and save the animation as well.

The table 5.1 below shows some of the clippings.

Table 5.1: Clippings of velocity ('U').

 <p>A side-view cross-section of a car model in a fluid flow simulation. The car is dark blue. The background is a color map of velocity magnitude, with a color bar on the right ranging from 0.0e+00 (blue) to 5.1e+01 (red). The flow is from left to right, showing a wake behind the car.</p>	Clipped at y-normal
 <p>A side-view cross-section of a car model, similar to the first image. The car is dark blue. The background is a color map of velocity magnitude, with a color bar on the right ranging from 0.0e+00 (blue) to 5.1e+01 (red). The flow is from left to right, showing a wake behind the car.</p>	Clipped at y-normal and $y = 0.1635$
 <p>A top-down view of a car model in a fluid flow simulation. The car is dark blue. The background is a color map of velocity magnitude, with a color bar on the right ranging from 0.0e+00 (blue) to 5.1e+01 (red). The flow is from left to right, showing a wake behind the car.</p>	Clipped at z-normal and $z = 0.049$ with 'Inside Out' selected.



5.2 Trace Streamlines

This section will outline how to trace streamlines and visualise them with the Ahmed body model loaded. Please follow the following steps;

- 1) Ensure that the case is loaded on ParaView with the 'internalMesh' selected in 'Mesh Regions'.
- 2) Hide (visibility) the loaded case.
- 3) Click 'Open' and select the 'ahmed_body.stl' (whole Ahmed body) file and click apply.

You should now see the full Ahmed body.

- 4) Click on the main loaded case (hidden in step 2) and click on the 'Stream Tracer' in the 'Common' toolbar (figure 5.3).

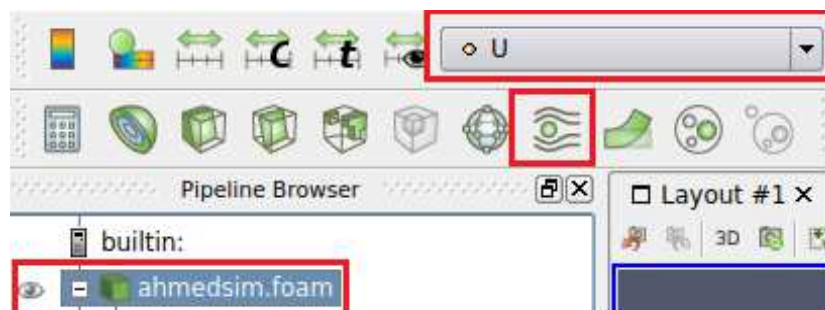


Figure 5.3: Select 'Stream Tracer' from 'Common' toolbar.

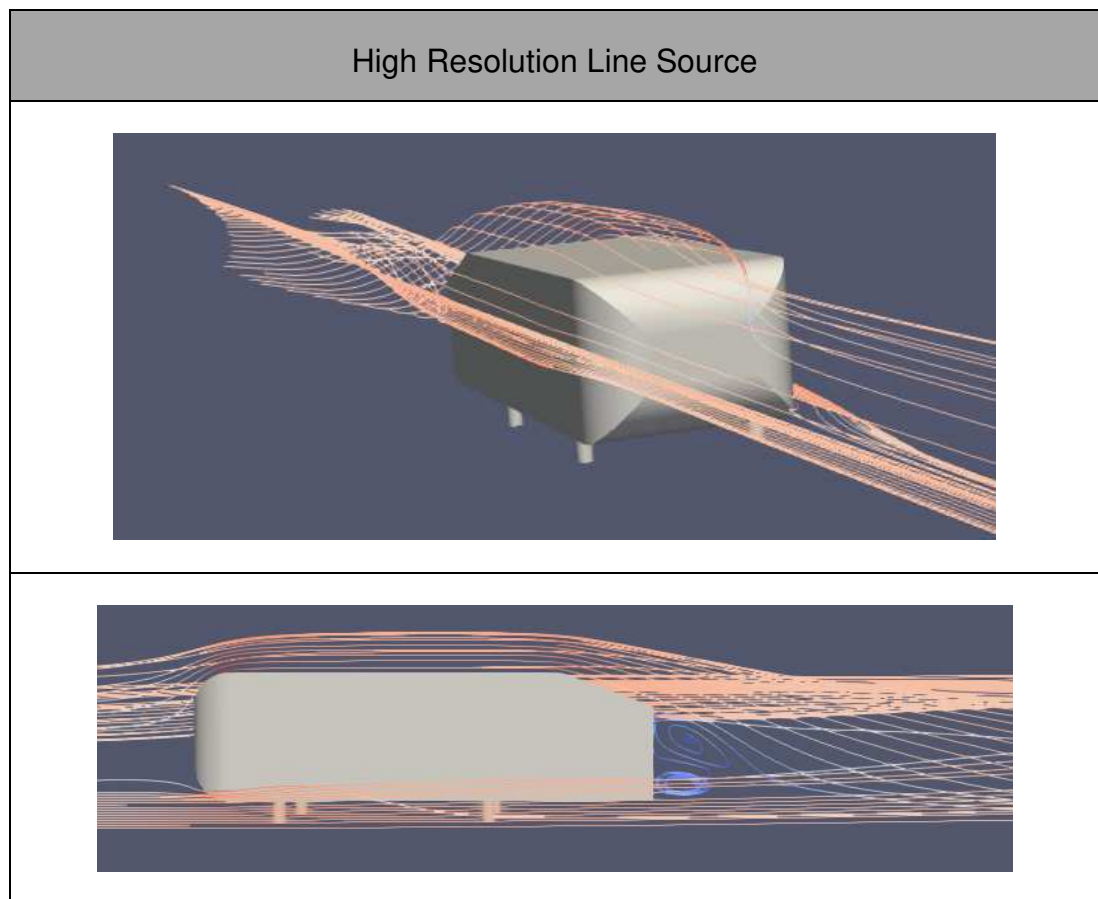
- 5) Please input the parameters for the 'Properties' outlined in table 5.2 below.

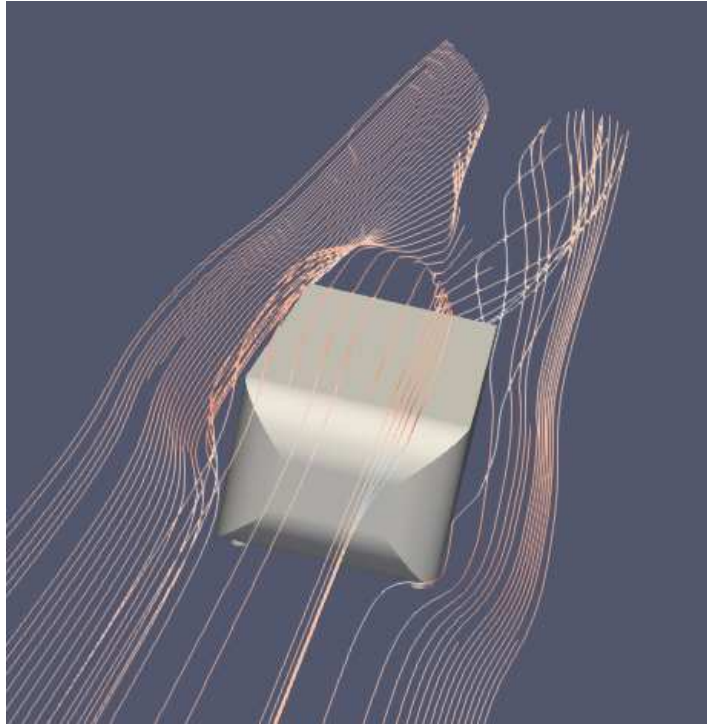
Table 5.2: Streamline parameters.

Parameter	Value
Vectors	U
Seed Type	High Resolution Line Source
Line Parameters	Point1: -2, -0.3, 0 Point2: 5, 0.3, 0.35
Resolution	70
Colouring	U
Alternatively	
Seed Type	Point Source
Center	-2, 0, 0.2
Radius	0.3

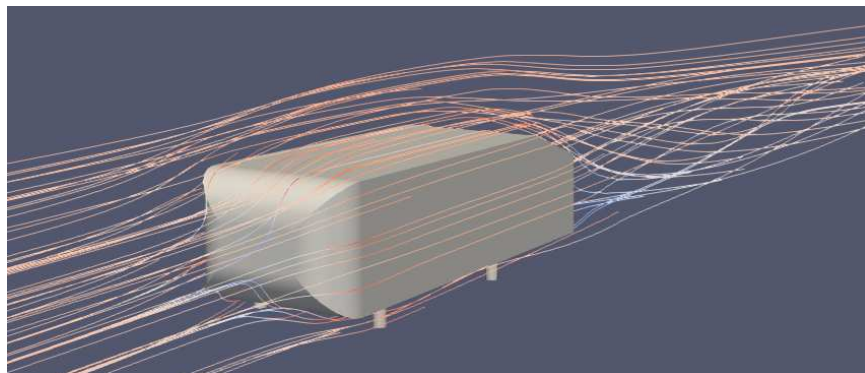
Click 'Apply' and the output should be similar to table 5.3 below.

Table 5.3: Traced streamlines.





Point Source



At this stage, it is advisable to save the ParaView state so that you can return to it at a later stage. To save, click on 'File' and click 'Save State'. Give an appropriate name and click 'OK'.

5.3 Wall Shear Stress

This section will look into extracting the wall shear stresses and imposing them onto the Ahmed body. The wall shear stresses are not saved during simulation. They need to be extracted post-simulation. In order to do this exit ParaView and return to the Terminal with the case path.

Note: Ensure the ParaView state was saved.

Type the following CLI command into the Terminal (from case path);

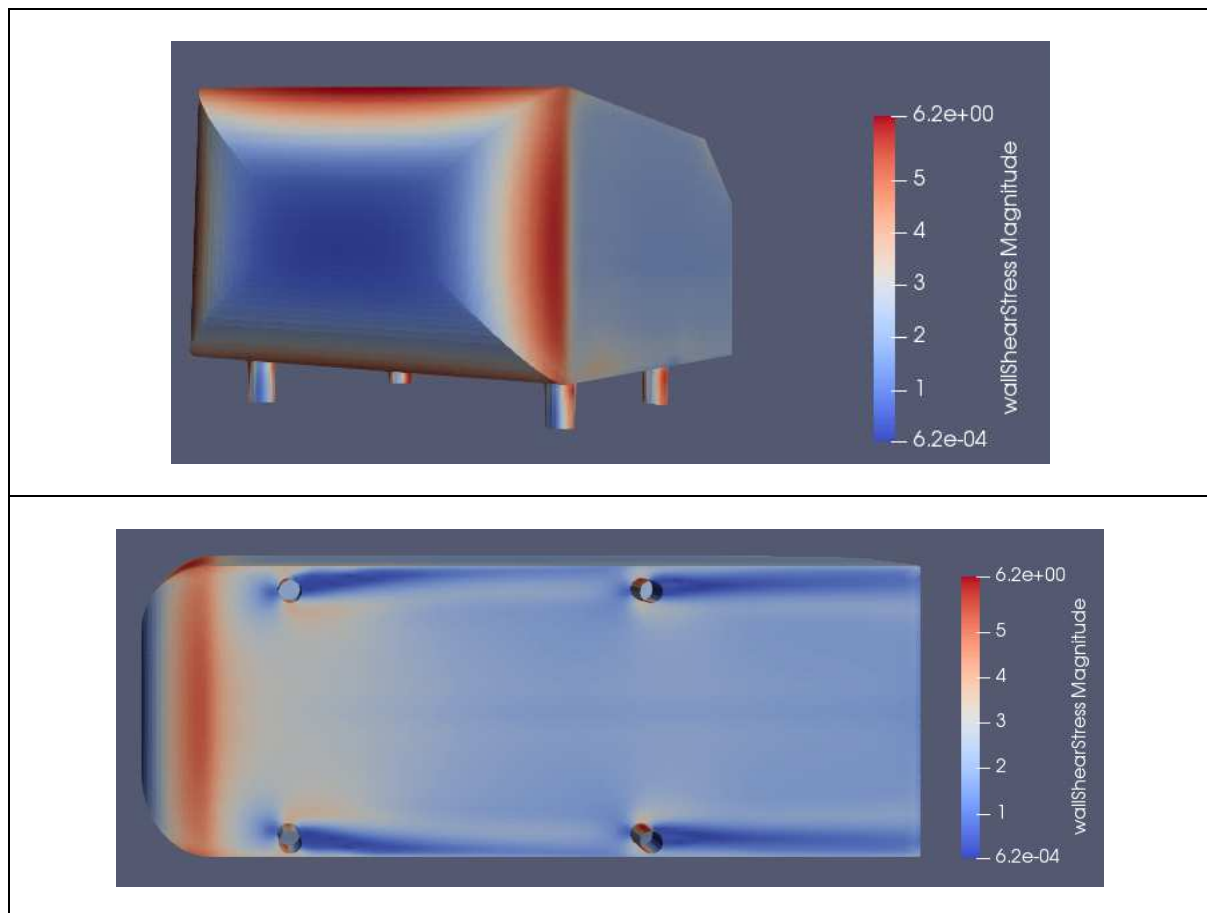
```
# Extract wall shear stress
simpleFoam -postProcess -func wallShearStress -latestTime

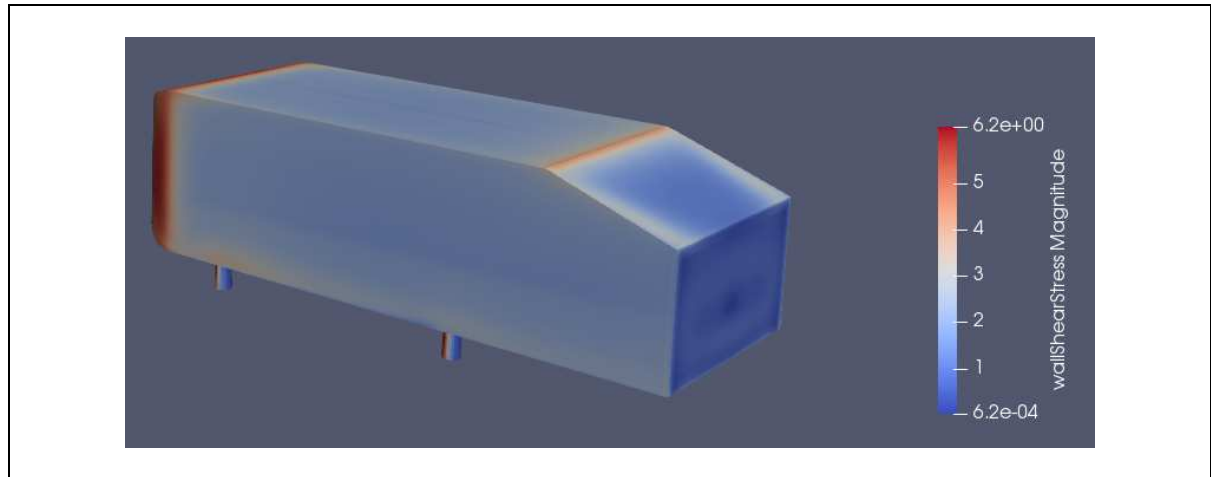
# Open ParaView to visualise results
ParaFoam -builtin
```

- 1) When ParaView is open, unselect the 'internalMesh' and select the 'ahmed1' and 'ahmed2' from 'MeshRegions'.
- 2) Make sure you are on the last time step (from 'VCR Control') and that the 'wallShearStress' is selected from the 'Cell Arrays' and click 'Apply'.
- 3) Select the 'wallShearStress' from the 'Active Variables Toolbar'.

Your Ahmed body should now display the wall shear stresses similar to table 5.4 below.

Table 5.4: Wall shear stresses.





More post processing methods would be looked at in the next tutorial.