

# COMPUTATIONAL FLUID DYNAMICS

**By**

**Saamira Yasmin**

**1st year Btech student**

**Chemical Engineering(4yr)**

**IIT Kharagpur**

**Guided by**

**Professor P Sunthar**

**Core Faculty Professor**

**Dept of Chemical Engineering**

**IIT Bombay**

## ACKNOWLEDGEMENT

When I started off with this project, I was completely new to computational fluid dynamics (CFD) and did not know how it worked. I did not have any knowledge of OpenFOAM and the other softwares required for it. I had approached Prof. P Sunthar with the project idea of studying the blood flow in the heart. This topic required high understanding and skills of CFD and is a distant area for me to work on. I started off with the basics of CFD, learning part by part and could finally produce intriguing results which are put down in this project report. This would not have been possible without the constant guidance and supervision of Sunthar Sir. He was always there to hear my doubts and keep my motivation high. I have learnt many things from this work which will surely serve as a strong foundation for my distant interest as mentioned above. I am extremely grateful to learn from Sunthar Sir and look forward to many such projects in the future. I would also like to thank Prachee, who is a Mtech student in the Chemical Engineering Dept. of IIT Bombay, studying under Prof. Sunthar. She was always there to clear my doubts, and we had fruitful discussions regarding the project topic. Lastly, I would like to thank my parents without whom this progress would not have been possible.

## MOTIVATION

I was curious since quite some time to know how blood flows through the coronary arteries of the human heart. This is definitely a difficult topic and research is still going on in this area. This requires high knowledge of computational fluid dynamics and I was very new to the concept then. I wished to learn about the fluid dynamics involved in it and start from the basics. Luckily, I got this opportunity at the end of my first year of Btech, under Prof. Sunthar. He willingly accepted to guide me in this area. I am extremely grateful to Sir for giving me this opportunity. The global pandemic (Covid-19) did not deter our spirits. The learning and work went on for four months i.e. from 1<sup>st</sup> May, 2020 to 31<sup>st</sup> August, 2020. It was indeed a great experience to work with him and I am happy that I could successfully complete my online internship at IIT Bombay. I definitely learned many new and interesting concepts in the area of CFD which I have put down in this report. I strongly believe that the experience gained through this project would help me immensely in the near future to fulfil my aim of knowing how blood flows through the coronary arteries of the human heart, and subsequently contributing to the society by providing some vital inputs to overcome the common health issues related to heart disorder.

## INTRODUCTION TO CFD

Computational fluid dynamics (CFD) is a branch of fluid mechanics that uses numerical analysis and data structures to analyse and solve problems that involve fluid flows. Computers are used to perform the calculations required to simulate the free-stream flow of the fluid, and the interaction of the fluid (liquids and gases) with surfaces defined by boundary conditions.

CFD is applied to a wide range of research and engineering problems, in many fields of science and industries, including aerodynamics and aerospace analysis, weather simulation, natural science and environmental engineering, industrial system design and analysis, biological engineering, fluid flows and heat transfer, and engine and combustion analysis.

## THEORY OF FLUID DYNAMICS

The fundamental basis of almost all CFD problems is the Navier–Stokes equations [1], which define many single-phase (gas or liquid, but not both) fluid flows.

The solver icoFoam [3] is one of the many solvers offered by the OpenFOAM computer software [2], and it uses the PISO algorithm to solve the Navier-Stokes equations for incompressible, laminar flow of Newtonian fluid flows. The density and the viscosity of the fluid are both assumed to be uniform. This algorithm solves

the continuity equation:

$$\nabla \cdot \mathbf{u} = 0 \quad \dots\dots\dots (\text{eqn 1})$$

and the momentum equation:

$$\frac{\partial(\mathbf{u})}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla p + \nabla \cdot (\nu \nabla \mathbf{u}) \quad \dots\dots\dots (\text{eqn 2})$$

where:

$\mathbf{u}$  = fluid velocity  
vector

$p$  = kinematic  
pressure

$\nu$  = kinematic viscosity      and

$\nabla$  = del operator.

The continuity equation (mass conservation) and the momentum conservation equation are the basis of the Navier-Stokes equations for fluid flows.

It should be mentioned here that the OpenFOAM uses the three-dimensional cartesian coordinate system to solve equations involving vectors and tensors.

## AIM OF THE PROJECT

We were interested in simulating and studying the characteristics of a fluid flowing through a pipe using computational fluid dynamics. The focus was purely on understanding incompressible, and Newtonian laminar flows. The different changes in the fluid flow were to be studied in the case of axisymmetric sudden expansion in the pipe. Lastly, we wished to calculate the pressure loss coefficient for the expanded pipe condition for a Reynolds number of 200. This pressure coefficient value was to be tallied with the results published by Oliveira and Pinho, 1997 [4].

## SCOPE

The accuracy of the results was to be calculated based on theoretical equations already known but in case of absence of validating equations, the above-mentioned research paper of Oliveira and Pinho, 1997 was to be used for reference.

## METHODOLOGY

The simulations had been carried out using the finite volume discretization method. The finite volume method is a method for solving partial differential equations like the Navier-Stokes equations in the form of algebraic equations. The physical parameters are approximated at discrete nodes surrounded by finite volumes within the problem domain.

Another important thing to be mentioned here is that instead of a complete circular pipe, a wedge-shaped pipe with an angle of  $1.5^\circ$  had been used. The purpose behind using a wedge-shaped pipe was to reduce the computational time. With the help of a wedge, the number of node points at which numerical computation had to be carried out got reduced.

For all the experimental simulations, axisymmetric fluid flows had been considered.

## TOOLS AND SOFTWARE USED

The software that was used for the simulations was openFOAM v7 (provided by Amazon Web Services EC2 instance [5]). The OpenFOAM is a C++ toolbox for the development of customized numerical solvers, and pre-/post-processing utilities for the solution of continuum mechanics problems, most prominently including computational fluid dynamics.

The OpenFOAM offers many solvers for CFD based on the properties of the fluid. The one used in this research was the icoFoam solver. The icoFoam solver is a transient solver for incompressible and laminar flow of Newtonian fluids.

After the post-processing, the ParaView 5.8.0 [6] was used to generate plots for the fluid flow parameters, to view blockmesh and animations of the fluid flow through the pipe.



## EXPERIMENTAL DETAILS AND RESULTS

The entire work has been divided down into three sections:

**Section I** focuses on fluid flow through a uniform cross-sectional pipe

**Section II** covers fluid flow through an expanded pipe with an expansion ratio of 1:2.6

**Section III** focuses on calculating the pressure loss coefficient

### I. FLUID FLOW THROUGH A STRAIGHT, CYLINDRICAL PIPE

Here we were interested in designing the pipe geometry and understanding the fluid dynamics for an incompressible, Newtonian and laminar fluid flowing through the length of the pipe. The mesh was fixed to be wedge-shaped and the flow to be axisymmetric.

In all the simulations, the value of each of  $\rho$ ,  $U$  and  $D$  (refer notations) was kept equal to 1. Any arbitrary values can be chosen for these parameters. The fluid flow depends solely on the Reynolds number, which is a dimensionless number. Instead of using various parameters to define a system, it is easier to use a single parameter like Reynolds number ( $Re$ ).

The use of dimensionless numbers in engineering and physics allows the important task of data reduction of similar problems. This means that a lot of experimental runs are avoided if data is correlated using appropriate dimensionless parameters.

We know the theoretical relation:  **$Re = \rho v D / \mu$  ..... (Eqn 3)**

where

$\rho$  = density of fluid ( $\text{kg/m}^3$ )

$\mu$  = dynamic viscosity of fluid ( $\text{kg/m.s}$ )

$D$  = diameter of pipe (m)

and

$v$  = velocity of flow (averaged over area of cross-section) ( $\text{m/s}$ ).

Here, Reynold's number is used to illustrate the fluid flow, and it is defined as the ratio of inertia force to the viscous force. It describes the predominance of inertia forces to the viscous forces occurring in the flow systems. In all the simulations, the different  $Re$  values were generated by changing the kinematic viscosity ( $\mu/\rho$ ) accordingly, in the transportProperties file of OpenFOAM.

Before starting off with the simulations, it becomes important to design the geometry of the pipe (in the blockMeshDict file) following which the boundary conditions are to be specified in the 0/ time directory.

The codes that were fed into the program were (codes A1, A2 and A3) which can be referred to in Appendix A.

### **Analysis of the grid/ grid testing and understanding the codes A1, A2 and A3 (refer to Appendix A)**

The interest was in defining a pipe geometry for the fluid flow keeping the length sufficient enough to achieve a developed flow. Computational methods were used to design a 1-D axisymmetric flow through a straight cylindrical pipe with cross-section as that of a sector of a circle, using the three-dimensional cartesian coordinate system. The blockMeshDict (code A1) file clearly states the physical dimensions of the pipe geometry. A single block was used to simulate the flow, keeping  $n_x=100$ ,  $n_r=50$  and  $n_1=1$ . The mesh spacing was kept non-uniform with the ratio 0.1:1:10 along the length of the pipe in order to get more computational mesh points near the developed flow region.

A series of grid testing and analysis was required before coming up with the mesh geometry. The grid testing involved various permutations and combinations of the mesh geometry in order to get a clear idea of the developing length so as to attain a fully developed velocity profile.

The boundary conditions were mentioned in the 0/ time directory in the velocity and pressure files (refer to codes A2 and A3). The centerline was mentioned to be axisymmetric and a no-slip condition was imposed on the walls (zero velocity as walls were stationary). The inlet velocity ( $U$ ) was chosen as 1m/s and a fixed pressure condition of 0.005 N/m<sup>2</sup> was imposed at the outlet face.

There were many other files but only the blockmeshdict file and the boundary conditions (0/U and 0/p) file have been attached in this report, to avoid complexity.

After running the codes A1, A2 and A3 in OpenFoam using the command line and completing the post-processing, the flow results could be viewed in ParaView. The ParaView plots in Figures 1 and 2 give an idea of the velocity magnitudes at different positions of the pipe for the case  $Re=100$  at the initial and last time instances.

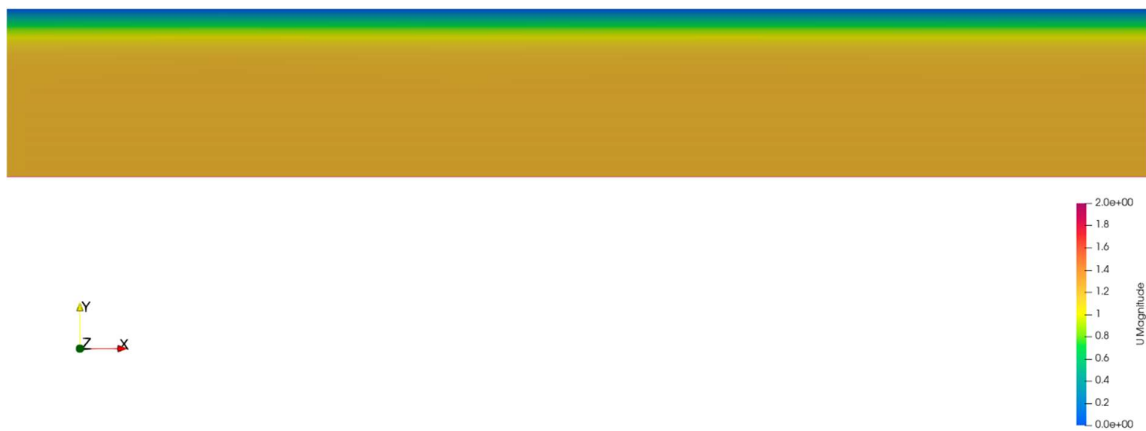


Fig 1: Re 100: visual representation of fluid flow velocity (U magnitude) throughout the length of the pipe at  $t=0.2s$

In fig (1), the left end is the inlet with inlet velocity set to 1m/s and the right end is the outlet.

$t=0.2s$  may be considered as almost  $t=0$  i.e. when the fluid has just started flowing through the pipe. With ParaView, a time animation could be witnessed which gave a clear idea about the fluid flow. Since it is difficult to introduce the animation here, snapshots of the initial and final time steps have only been included in figures (1) and (2). In both fig 1 and fig 2, the origin of the cartesian coordinate system may be assumed to be at the leftmost bottom corner.



Fig 2: Re 100: visual representation of fluid flow velocity (U magnitude) throughout the length of the pipe at the last time step (60s)

It is interesting to see (in figures 1 and 2) that the velocity magnitudes (represented by the colour scale) do not vary with the x-coordinate and differ only along the y-coordinate. This means that there is a velocity gradient only along the radial direction which is justified as the flow is one-dimensional. The achieved colour gradient for the steady state condition (fig 2) throws light on the fact that the velocity is maximum along the centerline ( $y=0$ ) and is minimum or zero at  $y=r$  due to the no slip condition.

Using the results obtained for pressure and velocity, the plots for the centerline velocity and the fluid flow pressure were plotted against the centerline x coordinates in ParaView 5.8.0.

The simulations were performed for Re 100, 250, 500, 650, 800, 1000 and 1200, and the derived results are shown in Figures 3 – 9. The plots for the different Re are corresponding to the last time step i.e. when the flow had become stable with time or at steady state.

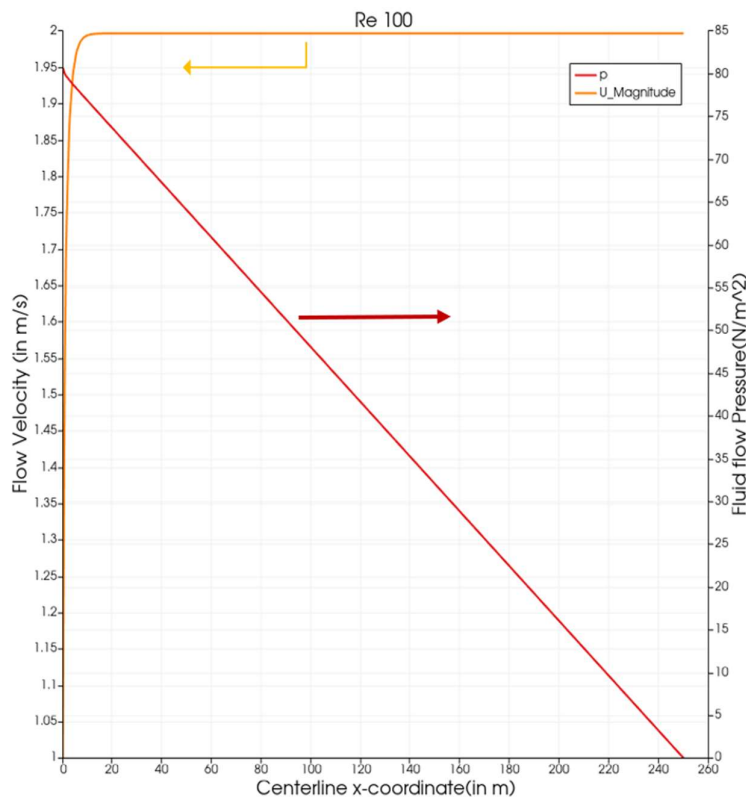


Fig 3: Re 100: plot of fluid velocity(magnitude) and fluid flow pressure at different centerline x-coordinates along the length of the pipe taken at last time step i.e. when flow has stabilised with time

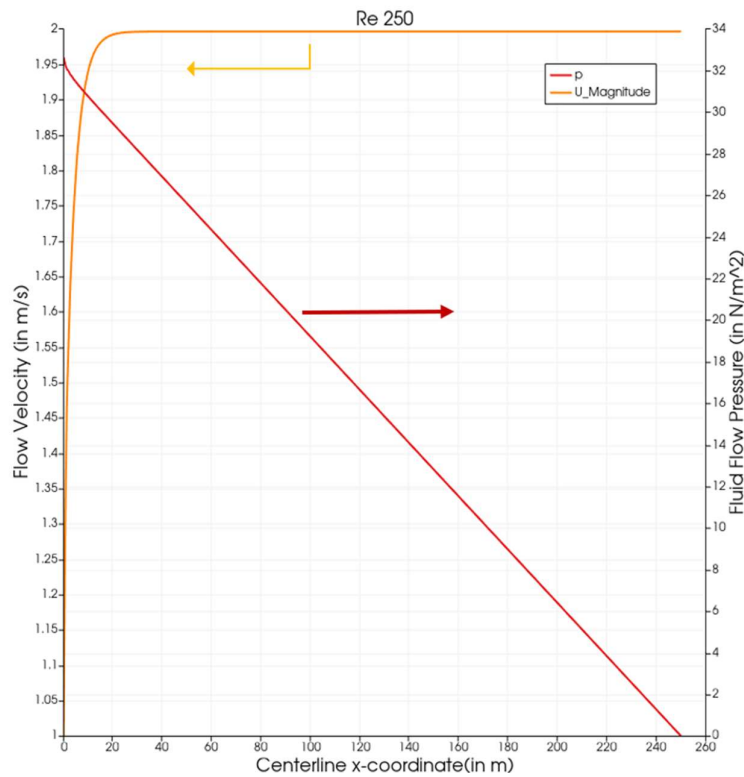


Fig 4: Re 250: plot of fluid velocity(magnitude) and fluid flow pressure at different centerline x-coordinates along the length of the pipe taken at last time step i.e. when flow has stabilised with time

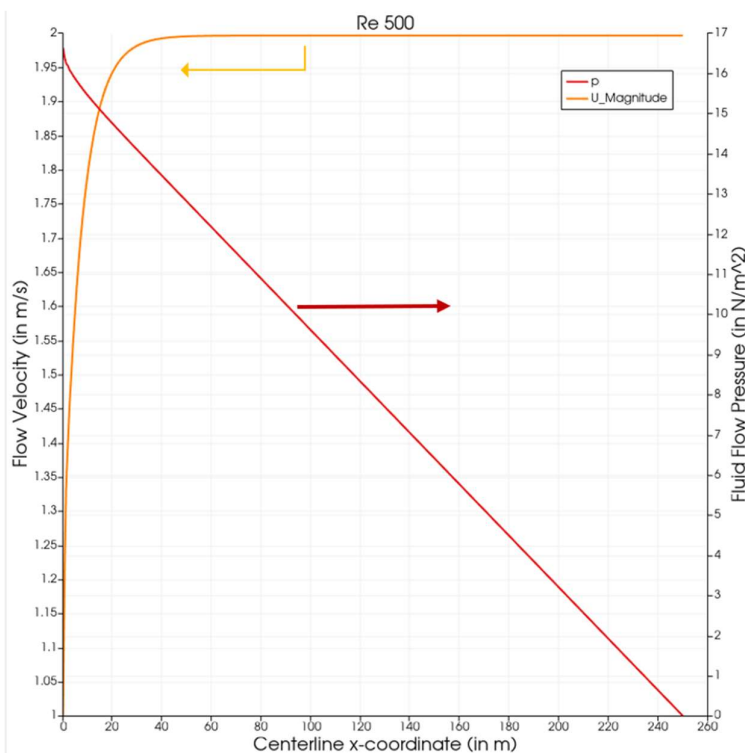


Fig 5: Re 500: plot of fluid velocity(magnitude) and fluid flow pressure at different centerline x-coordinates along the length of the pipe taken at last time step i.e. when flow has stabilised with time

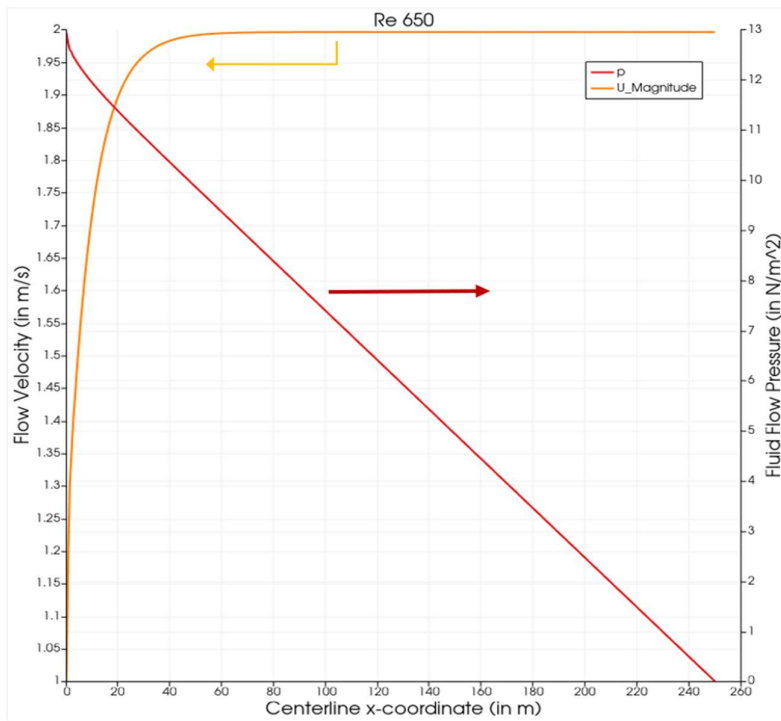


Fig 6: Re 650: Left axis- fluid flow velocity magnitude measured at different centerline positions of the pipe; Right axis- fluid flow pressure measured at different centerline positions of the pipe

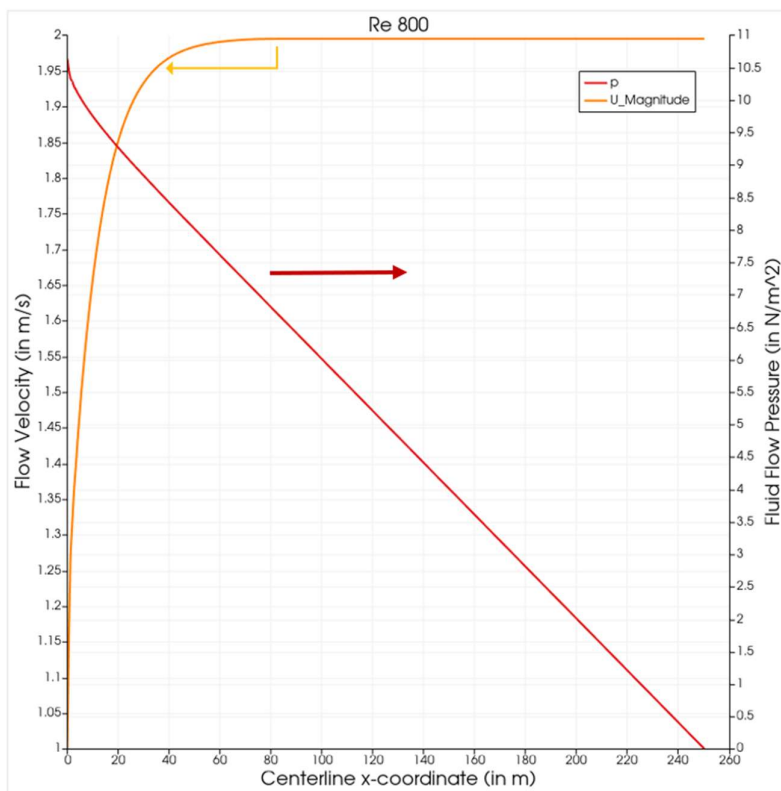


Fig 7: Re 800: Left axis- fluid flow velocity magnitude measured at different centerline positions of the pipe  
Right axis- fluid flow pressure measured at different centerline positions of the pipe

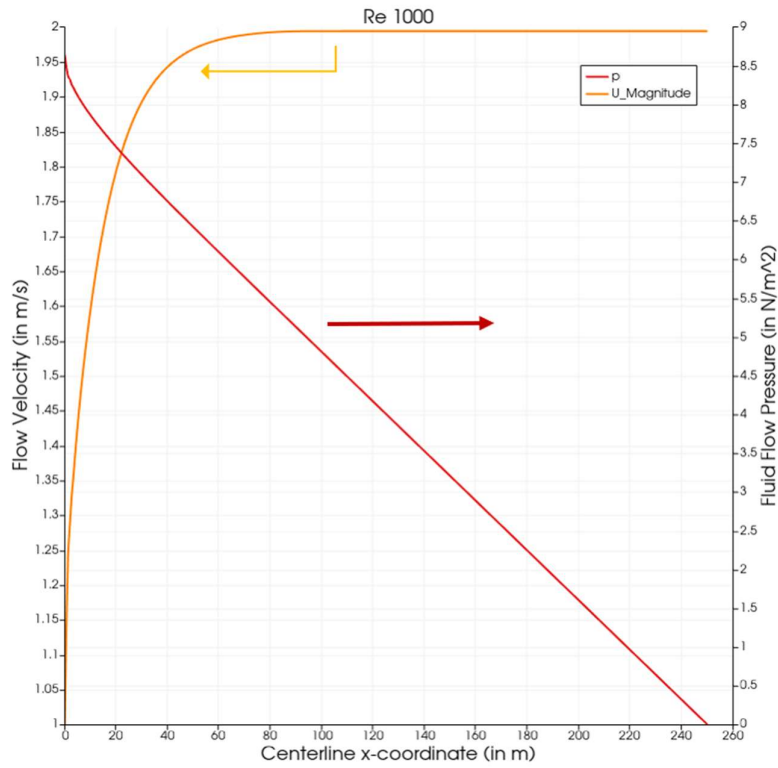


Fig 8: Re 1000: Left axis- fluid flow velocity magnitude measured at different centerline positions of the pipe  
Right axis- fluid flow pressure measured at different centerline positions of the pipe

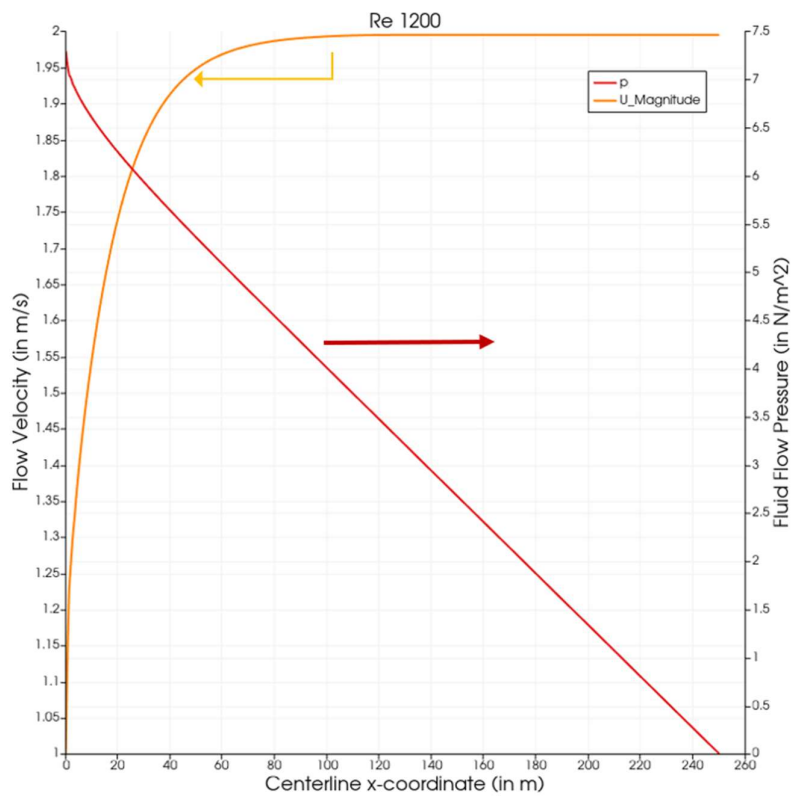


Fig 9: Re 1200: plot of fluid velocity(magnitude) and fluid flow pressure at different centerline x-coordinates along the length of the pipe  
taken at last time step i.e. when the flow has stabilised with time

In all these plots, we find that the velocity profile is that of a parabola. The inlet velocity (at  $x=0$ ) is 1m/s and then the centerline velocity magnitude increases with increasing  $x$ -coordinate and finally becomes stable after attaining the maximum velocity (i.e. typically approaching 2 m/s). The length required to attain this steady developed velocity is known as 'developing length'.

The fluid pressure profile can be seen decreasing drastically. Such kind of behaviour can be understood from the engineering Bernoulli term present in the Bernoulli's equation. This term accounts for the frictional and viscous losses. The pressure drop increases with length.

The entries for Table 1 have been obtained after studying the ParaView generated plots (Figures 3-9). The blockmeshdict file given earlier stated the geometry of the pipe. Re value was varied by changing the kinematic viscosity ( $\nu$ ) values in the transportProperties files using knowledge of Eqn 3.

<b>Table 1: Readings for various flow parameters and geometrical restrictions for different Re under uniform cross-sectional area of pipe</b>							
<b>Re</b>	<b>100</b>	<b>250</b>	<b>500</b>	<b>650</b>	<b>800</b>	<b>1000</b>	<b>1200</b>
<b>Nu(m<sup>2</sup>/s)</b>	0.01	0.004	0.002	0.0015	0.0012	0.001	0.0008
<b>D(m)</b>	1	1	1	1	1	1	1
<b>Pipe Length(m)</b>	250	250	250	250	250	250	250
<b>Inlet U(m/s)</b>	1	1	1	1	1	1	1
<b>Dev Len(L) (m)</b>	8.5	18.5	37	48	58.75	73.25	88
<b>L/D</b>	8.5	18.5	37	48	58.75	73.25	88
<b>L/(D*Re)</b>	0.085	0.074	0.074	0.0738	0.0734	0.0732	0.0733
<b><math>8\pi\mu L v/a</math> (N/m<sup>2</sup>)</b>	76.89	29.48	13.56	9.89	7.61	5.63	4.29
<b>Fully Dev Vel(m/s)</b>	1.99	1.99	1.99	1.99	1.99	1.99	1.99
<b>Delta P(N/m<sup>2</sup>)</b>	77.27	29.63	13.63	9.94	7.65	5.66	4.32
<b>L(99) (m)</b>	6.5	15	29.25	38	46.75	58.5	70

The column 'Dev Len' stands for developing length and is defined as the minimum pipe length required to attain a developed fluid flow or a constant velocity profile. For all the different Re it has been found that the developed velocity comes out to be 1.99m/s (i.e. 99.5% of theoretical developed velocity) which one can refer to in the 'Fully Dev Vel' column. The theoretical developed velocity magnitude comes out to be 2 m/s, considering the shape of the velocity plot to be of parabolic profile. This can be calculated using the equation  **$U_{max}=2*U_{avg}$  (eqn 4)**. Here  $U_{avg}$ = inlet velocity of 1m/s and hence  $U_{max}$  becomes 2m/s. In order to verify the correctness of the developing length results, the empirical relation  **$L/D=0.06 Re$  (eqn 5)** was used. The graph for  $L/D$  v/s  $Re$  along with the empirical relation was plotted using the software gnuplot.



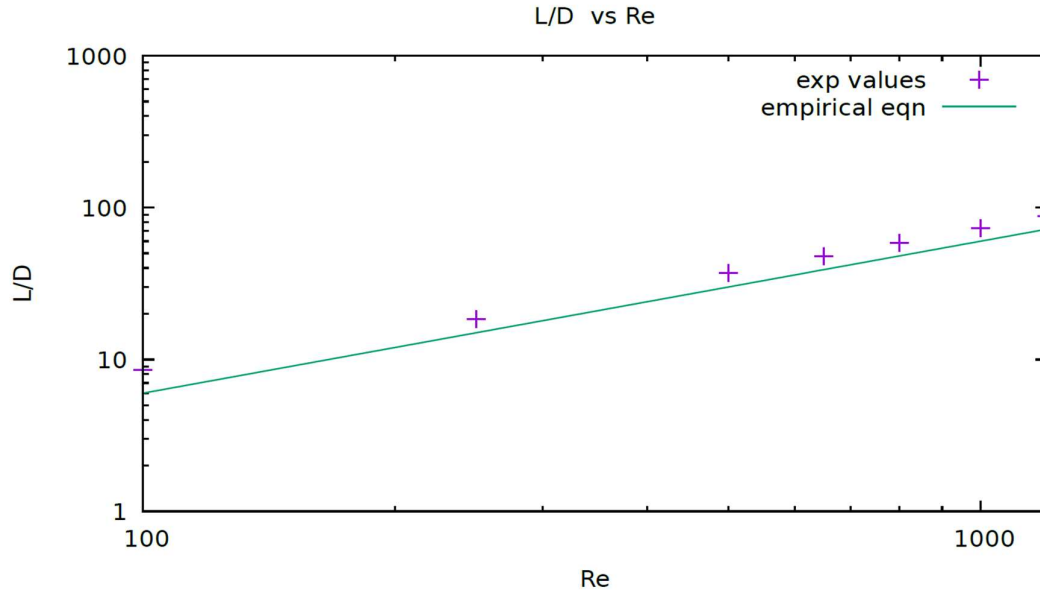


Fig 10: Discrete points plot: y-axis - ratio of developing length (L) to diameter of pipe (D), x-axis - Reynolds number (Re); straight line plot:  $L/D=0.06Re$  (from empirical equation)

The graph in Figure 10 shows a slight deviation of the experimental results from the theoretical relation. In order to get a perfect overlapping of the two graphs, the entries in the column L(99) meaning development length to attain 99% of the theoretical developed velocity (i.e. 2 m/s) was used in place of Dev Len(L). The L(99)/D when plotted against Re gave a perfect fit with the empirical relation (as shown in Figure 11).

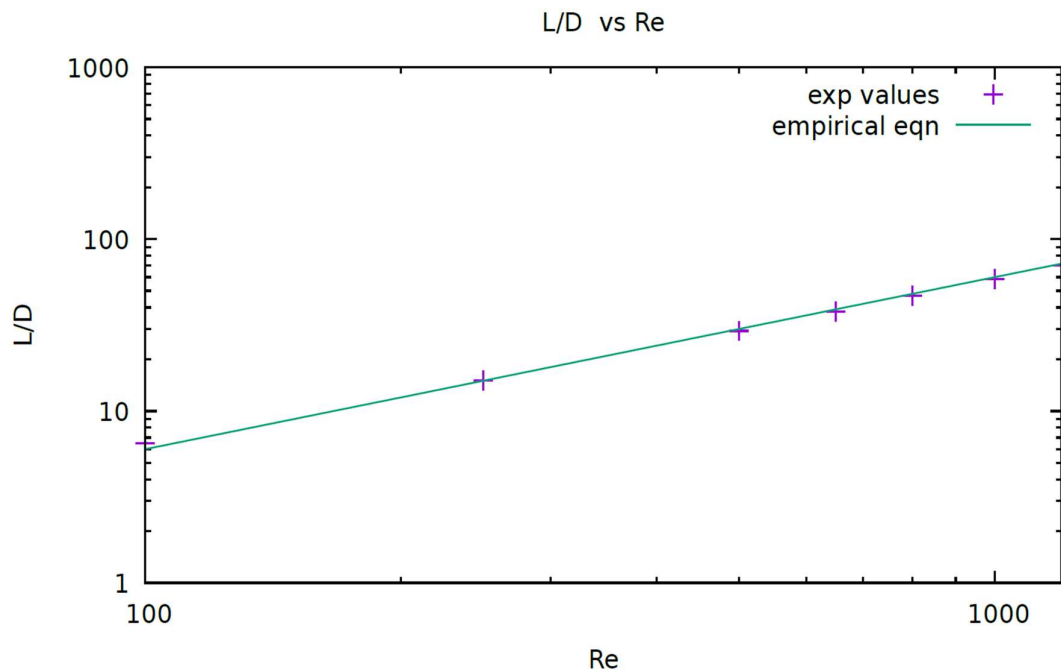


Fig 11: Discrete points plot: y-axis - ratio of developing length to attain 99% of developed velocity (L(99)) to diameter of pipe (D), x-axis - Reynolds number (Re); straight line plot:  $L/D=0.06Re$  (from empirical equation)

Hence, it can be commented that the results from the simulations are 99% correct w.r.t. the theoretical results for developing length.

Another important law that is verified here is the **Hagen Poiseuilles Law** which gives a relation for the drop in pressure in the developed region of the fluid flow. The equation is given by

$$\Delta P = 8 \pi \mu L v / a \quad \text{..... (Hagen Poiseuilles Law)} \quad \text{(eqn 6)}$$

$8 \pi \mu L v / a$  values were calculated for the various Re, and  $\Delta P$  ( $\Delta P$ ) values were noted down from the ParaView plots. The values were very close to each other and thus confirm the high accuracy of the simulations performed and the results obtained. Equation 6 also shows that for a fixed Re, the pressure drop along the flow is directly proportional to the length of the pipe and this relation was verified separately during the grid testing.

For the second part of the project, a sudden expansion in the area of cross-section was introduced at a certain length of the pipe. The expansion ratio was fixed at 1:2.6 for all Re. With this modified geometry, the changes in the physical parameters of fluid flow were noted down.

## II. SUDDEN EXPANSION OF PIPE AND STUDY OF CHANGES IN THE FLUID FLOW

The previous part focussed on a uniform cross-sectional pipe and fluid flow through it. But the pipes used for industrial and drainage purposes may not necessarily be a uniform diameter pipe. They may use pipes of varying cross-section and for this reason it becomes important to understand flows through expanded pipes. Here we deal with a geometry that is made up of pipes with two diameters as the name suggests 'sudden expansion'.

The C++ program codes (codes B1, B2 and B3) given as input in OpenFOAM to obtain the required fluid flow model are given in Appendix B.

The boundary conditions for velocity( $U$ ) and pressure( $p$ ) are an important factor in determining the properties and the profile of the fluid flow through the pipe. These files are located in the 0/ time directory and read as given in Appendix B (refer to codes B2 and B3).

The codes B1, B2 and B3 can be understood from the explanation already given in section I, where codes A1, A2 and A3 are explained. The main difference in this section (II) lies in the fact that here there are three blocks which are patched together at the expansion plane. The condition of an axisymmetric sudden expansion (1:2.6 ratio) has been introduced and greater number of computational mesh points have been placed near the expansion plane for more accurate results. The length of the inlet pipe is kept sufficient enough to achieve developed flow before expansion. Similarly, the outlet pipe (comprising of blocks 2 and 3) has been kept sufficiently long to achieve redeveloped flow.

After feeding in these codes and running the simulation till the endtime (as specified in the controldict file), the post-processing part was done and the results were then extracted in ParaView.

The results were quite interesting and are jotted down here for the Re 10 case, at the initial and final time steps in the form of 2-D schematics as shown in Figures 12 and 13.



Fig 12: Re 10: sudden expansion at 4.5m (in this fig., the length at which one can see the increase in diameter) and picturization of the flow velocity magnitude( $U$ ) at 0.02s

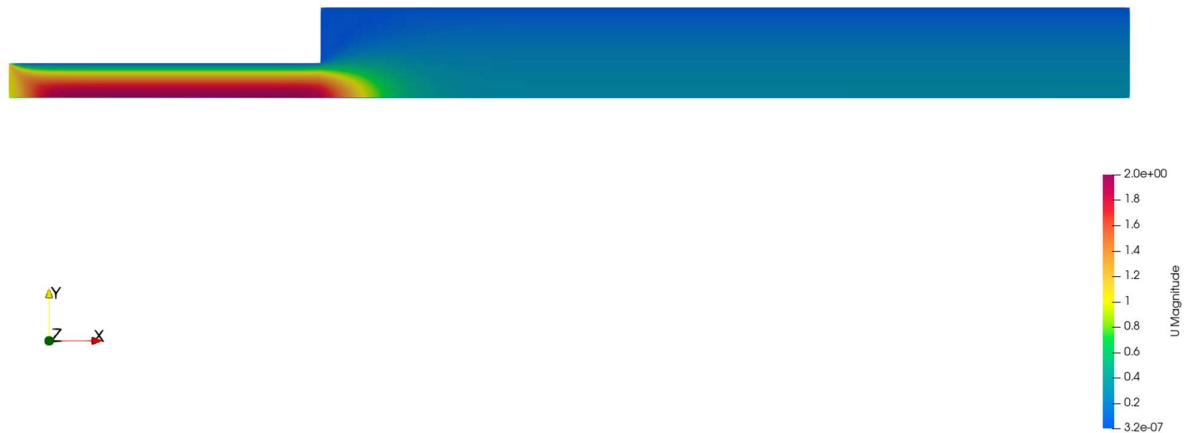


Fig 13: Re 10: Sudden expansion of pipe at 4.5m at last time step (33s) or at steady state

We can infer from figures 12 and 13 that in the case of expansion in the pipe diameters, the flow becomes two-dimensional i.e. varying along the length as well as radially. The decrease in velocity in the downstream region supports the continuity equation (Eqn 7).

In the system/controlDict file the endtime for the Re 10 simulation was mentioned to be 33sec with  $\Delta t$  being  $10^{-4}$ s. The endtime should be sufficient enough to get a steady flow profile (i.e. constant with time) and for the residuals to converge. After the simulation was run, its corresponding time animation fluid flow could be visualised in ParaView (refer to the initial and last time step screenshot given in fig. 12 and fig. 13).

Using ParaView, the centerline velocity and pressure profiles could be viewed from the data plots for various time steps. These plots give great information about the developed lengths, developed velocities, change in profiles due to the sudden expansion of the pipe, etc. The graph in Figure 14 for Re 10 shows the U, p plots v/s the centerline x-coordinate at the last time step (i.e. at 33s).

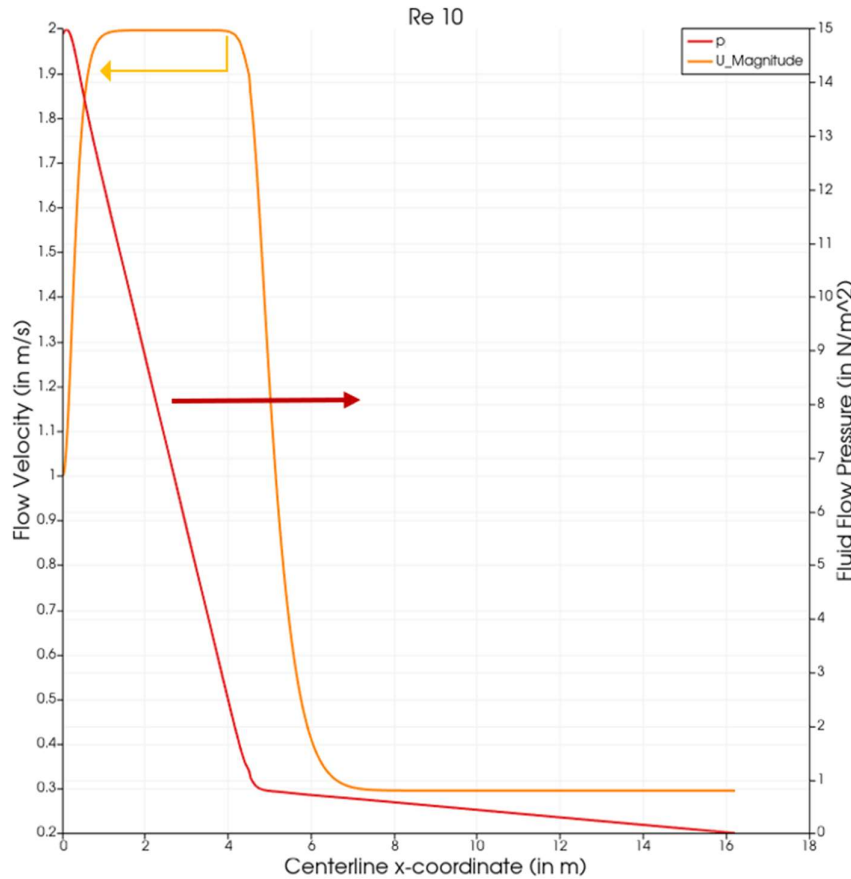


Fig 14: Re 10: Sudden expansion of pipe at 4.5m length  
Variation of fluid flow velocity and pressure at different centerline x-coordinates along the length of the wedge-shaped pipe

The lengths of the first and second pipes (with diameter 1 and 2.6, respectively) were kept such that developed flows were achieved upstream and downstream (w.r.t. expansion region). Using the knowledge gained from the prior experiment with no expansion, and confirmation of the validity of the formula  $L/D=0.06Re$  in our experiments, an estimate for the lengths of the pipes could be made. To witness some part of the developed profiles before and after the expansion, a length of 1.5 times  $L$  was chosen. Accordingly, the end time was adjusted to get a steady flow that remained constant with time.

In a similar fashion, the process was repeated for different Reynolds numbers by changing the kinematic viscosity ( $\nu$ ) in the constant/transportProperties file using the already mentioned formula (refer Eqn 3). Attaching all the files would make the report cumbersome so only the final ParaView graphs have been listed.

The plots for the various  $Re$  that were obtained were (enlisting for  $Re$  50, 100, 250 and 500 in Figures 15-18)):

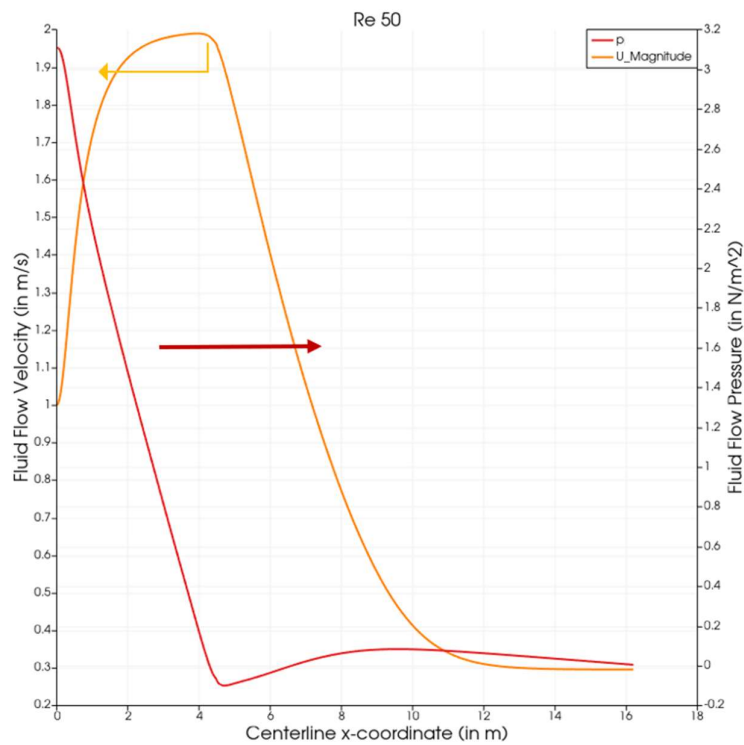


Fig 15: Re 50: Sudden expansion of pipe at 4.5m length  
Variation of fluid flow velocity and pressure at different centerline x-coordinates along the length of the wedge-shaped pipe

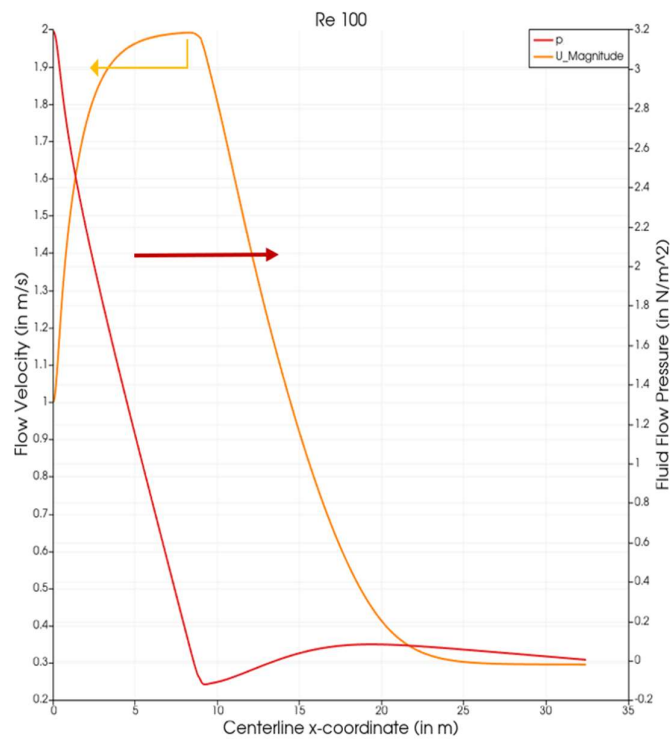


Fig 16: Re 100: Sudden expansion of pipe at 9m length  
Variation of fluid flow velocity and pressure at different centerline x-coordinates along the length of the wedge-shaped pipe

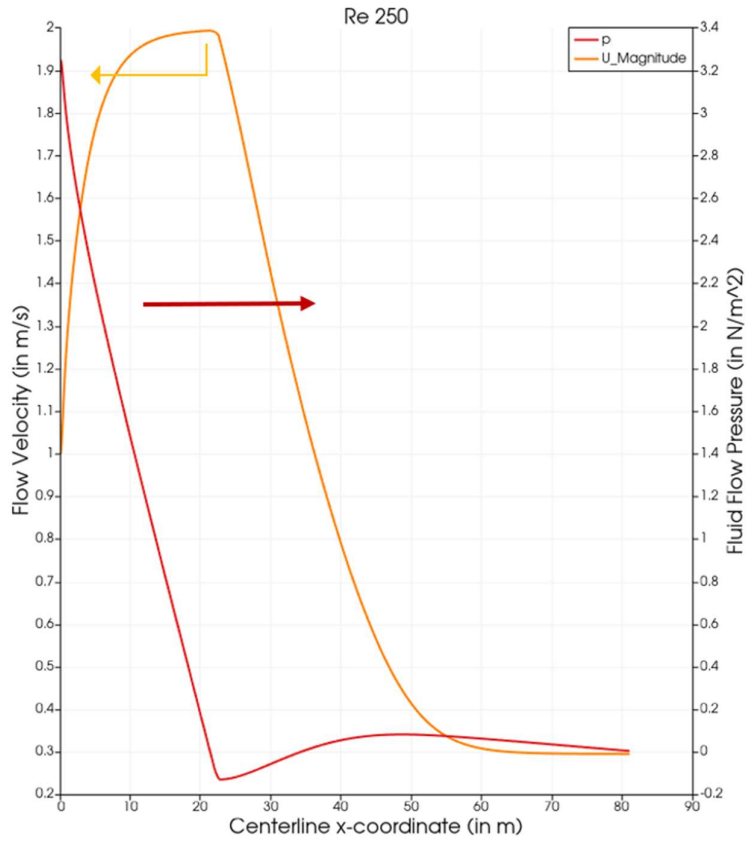


Fig 17: Re 250: sudden expansion in area of cross-section of pipe at  $x=22.5m$  with expansion ratio of 1:2.6 ( $D1:D2$ ); plots depicting variation in left and right axes data with increasing x-coordinate

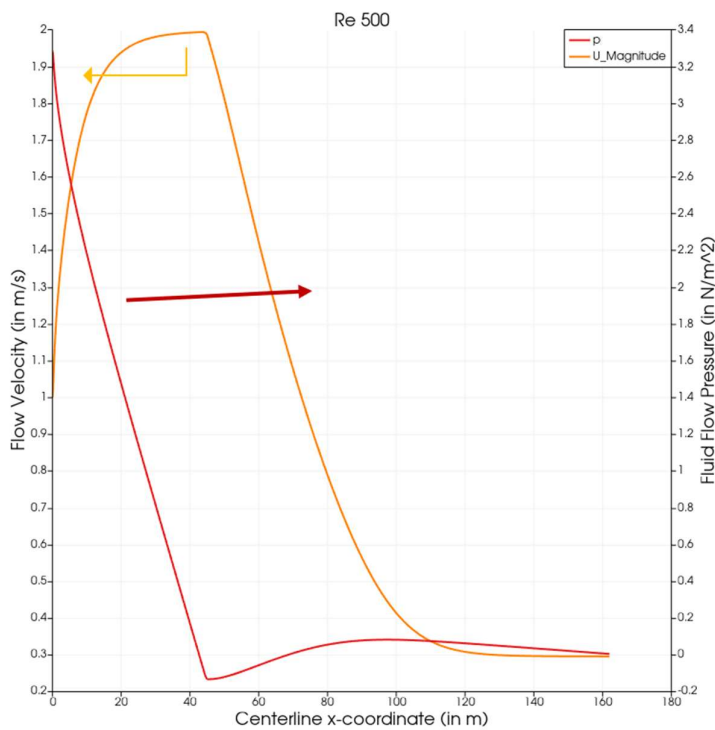


Fig 18: Re 500: sudden expansion in area of cross-section of pipe at  $x=45m$  with expansion ratio of 1:2.6 ( $D1:D2$ ); plots depicting variation in left and right axes data with increasing x-coordinate

The plots (Figures 14-18) were analysed and the derived values were tabulated as given in Table 2:

<b>Table 2: Readings for various flow parameters and geometrical restrictions for different Re under the case of sudden expansion in the pipe</b>					
<b>Re</b>	<b>10</b>	<b>50</b>	<b>100</b>	<b>250</b>	<b>500</b>
<b>L1=1.5*DEV L1 (m)</b>	4.5	4.5	9	22.5	45
<b>L2=1.5*DEVL2(m)</b>	11.7	11.7	23.4	58.5	117
<b>D1(m)</b>	1	1	1	1	1
<b>D2(m)</b>	2.6	2.6	2.6	2.6	2.6
<b>INLET U(m/s)</b>	1	1	1	1	1
<b>P AT OUTLET(N/m<sup>2</sup>)</b>	0.005	0.005	0.005	0.005	0.005
<b>DEV L1(m)</b>	1.1	3.15	6.18	15.14	29.8
<b>DEV L2(m)</b>	2.65	8.57	16.82	41.89	83.95
<b>U MAX_1(m/s)</b>	1.99	1.98	1.98	1.98	1.98
<b>U<sub>expan</sub>(m/s)</b>	1.88	1.95	1.97	1.98	1.98
<b>FINAL DEV U(m/s)</b>	0.29	0.29	0.29	0.29	0.29
<b>DelP across L2(N/m<sup>2</sup>)</b>	1.112	0.079	0.111	0.132	0.137
<b>END TIME=2t</b>	33	170	330	825	1100
<b>DELTA T</b>	0.0001	0.001	0.001	0.005	0.01
<b>CLOCK TIME(s)</b>	7,702	4,838	8,623	7,108	25,421
<b>COURANT MAX</b>	0.02	0.204	0.138	0.389	0.327

Due to the expansion of the pipe, there has been a sudden fall in the centerline velocity (magnitudes) which can be seen in the plots in the expanded region (at  $x = L1$ ). The deformation of the parabolic profile can be seen as it falls to some final developed velocity (column 'FINAL DEV U'). For all the simulations this value comes out to be the same as 0.29 m/s. This 0.29 value matches correctly with  $A_1 U_1 = A_2 U_2$  ..... (Eqn 7) which is a balance equation for the volume flow rates where  $U_1$  and  $U_2$  are the area-averaged velocities. The left-hand side corresponds to the developed flow in the upstream region and the right-hand side to the developed flow in the downstream region. So,  $U_1$  is the column 'U MAX\_1' and  $U_2$  is the column 'FINAL DEV U'.

The pressure profiles in the upstream region are similar to the ones obtained in section I. The significant change that is witnessed here is after the expansion, in the downstream region. The pressure increases to a certain value and then further decreases with increasing x-coordinate. The increase in pressure can be accounted due to the decrease in velocity on expansion and is explained by Bernoulli's equation [1] which is an extension of the Navier Stokes equation (Eqn 1,2).

The column 'COURANT MAX' refers to the maximum courant number and the term can be derived from the CFL condition. The Courant–Friedrichs–Lewy or CFL condition is a condition



for the stability of unstable numerical methods that is important for CFD problems. The CFL condition expresses that the distance that any information travels during the timestep length ( $\Delta t$ ) within the mesh must be lower than the distance between mesh elements ( $\Delta x$ ). In other words, information from a given cell or mesh element must propagate only to its immediate neighbours. On imposing this condition, we get the term Courant number (C):

$$C = \alpha \frac{\Delta t}{\Delta x} \dots\dots\dots \text{(Eqn 8)}$$

Here  $\alpha$  is the magnitude of velocity through the cell and the entire right-hand term is the 'COURANT MAX'. We must ensure that  $C \leq 1$  so that the numerical viscosity does not become negative. This condition was also ensured in the previous part of the experiment and is a must in CFD problems.

The column 'DELTA T' is the timestep length ( $\Delta t$ ) that appears in the Eqn 8. What is to be noticed here is that the  $\Delta t$  value needs to be decreased for lower values of Re. For Re 10, it is in the order of  $10^{-4}$  and for Re 500, it is in the order of  $10^{-2}$ . 'CLOCK TIME' is the total running time taken for the simulation.

After viewing the great impact of pressure drop on the pressure profiles, it becomes important to focus our attention on the pressure loss coefficient.

### III. PRESSURE LOSS COEFFICIENT IN FLUID FLOW

One of the most common problems in fluid mechanics is the estimation of pressure loss. Knowledge of the magnitude of frictional losses is of great importance in determining the power requirements of the pump forcing the fluid through the pipe. These losses need to be calculated accurately as the refining and petrochemical industries need to figure out the proper location of pumps when pumping crude oil and other fluids to large distances.

Whenever the fluid velocity is changed in direction or in magnitude due to obstructions in the flow path, pipe friction, change in the direction of flow or sudden or gradual change in the area of cross-section of the pipe, friction additional to the skin friction is generated in the pipes. Such friction occurs when the normal streamlines are disturbed and vortices are generated.

In the previous part, we dealt with sudden expansion in the geometry of the pipe and this resulted in additional pressure losses which is to be determined in this part of the experiment. Our aim was to determine the pressure loss coefficient for the sudden expansion case for Re 200. The final result was to be verified with that published in the research paper [4] which had performed a similar study, only for Re 200 and Re 50. Hence, work in this section is limited to the case Re 200 only, but would be sufficient enough to give a basic idea about the pressure loss.

Exactly like the previous sudden expansion simulations for various Re, simulation was run for Re 200. After the post-processing, the physical parameters, flow velocity and fluid pressure were plotted in Figure 19 at different centerline positions, throughout the length of the pipe.

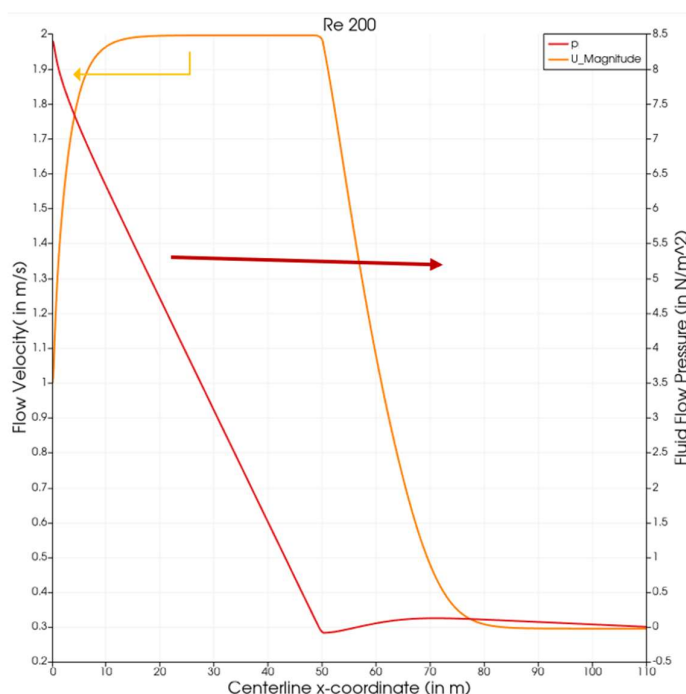


Fig 19: Re 200: sudden expansion at  $x=50\text{m}$ ; ParaView plot depicting fluid flow velocity and pressure at various x-coordinates of the pipe

The change in the nature of the plots due to the sudden expansion has already been discussed in the previous section. Now, we reach to the part where fig (20) and fig (21) were generated by simply following the steps as mentioned in the research paper [4] and reproducing the plots similar to figures 5a and 5b (of the research paper).

In order to understand these plots, it is important to first introduce the quantities: general pressure coefficient ( $C_p$ ), Darcy-friction factor ( $f$ ) and the local irreversible loss-coefficient ( $C_i$ ). Before that it is important to state here that the calculations involved in determining  $C_i$  were based on the assumptions that there was no wall friction and that the pressure was uniform at the expansion plane. These assumptions are definitely not a part of openFOAM and the simulation does not go with these assumptions. OpenFOAM considers the wall friction parameter in the calculation of the Darcy friction factor but we proceeded with our work based on eqn (9).

The theoretical equation to calculate the friction factor is  $f = 64/Re$  ..... (eqn 9)

The other relation that we know is  $f = \Delta C_p / \Delta x$  ..... (eqn 10)

Under ordinary circumstances i.e., when there is no expansion in the pipe, the formula for calculating the pressure coefficient is

$C_p = \Delta p / 0.5 \rho \bar{u}_1^2$  ..... (eqn 11) where  $\bar{u}_1$  is the average velocity magnitude or in other words the inlet velocity

Using equation (11), fig (20) had been generated. The x-coordinate ( $r=0$ ) with value zero corresponds to the expansion plane.  $x=-20$  and  $x=50$  refer to the x-coordinates where the flow has become developed and is measured wrt the expansion plane. With the obtained  $C_p$  values, the plot of  $C_p$  v/s  $x/D1$  had been produced. We notice an almost straight-line curve that is decreasing for the upstream region and a more or less constant curve for the downstream region. These two distinguished curves do not tally with eqns (9) and (10).

Owing to this problem, a modified  $C_p$  with the notation  $\bar{C}_p$  ( $C_p$  bar) had been introduced with a modified set of equations for the left and right pipes. The equation for its calculation reads as  $\bar{C}_p = \Delta p / 0.5 \rho \bar{u}_1^2$  ,  $x \leq 0$

$$\bar{C}_p = \Delta p / 0.5 \rho \bar{u}_2^2 - \frac{(1 - \sigma^2) \bar{C}_{p01}}{\sigma^2} + C_R , x > 0 \quad \text{..... ( eqn 12)}$$

With these two different set of equations for the upstream and downstream pipes, the results come out to be much more accurate and comply with the theoretical relations.

In fig (21), one can see that there is one curve for the experimental values of  $C_p$  bar and two straight lines known as straight line fits. Straight line fitting had been done separately for the left and right parts. Such a straight line fit with the data points of  $C_p$ , gives us the magnitude of the experimentally determined darcy-friction factor ( $f$ ) from the slope of the straight line. Also, the values for  $\bar{C}_{p01}$  and  $\bar{C}_{p02}$  can be used to calculate the irreversible local loss coefficient ( $C_i$ ).  $\bar{C}_{p01}$  and  $\bar{C}_{p02}$  can be obtained by finding out the intercept with the y-axis at  $x=0$  of the straight line fitted to the curve pertaining to  $x \leq 0$  and  $x > 0$ , respectively.

$$C_i = \sigma^2(\bar{C}_{p01} - \bar{C}_{p02}) \dots\dots\dots (\text{Eqn } 13)$$

It must be noted here (as suggested by the paper) that the error in calculating  $C_i$  can be reduced by drawing the straight-line fit graph only for the  $C_p$  points corresponding to the developed flow region.

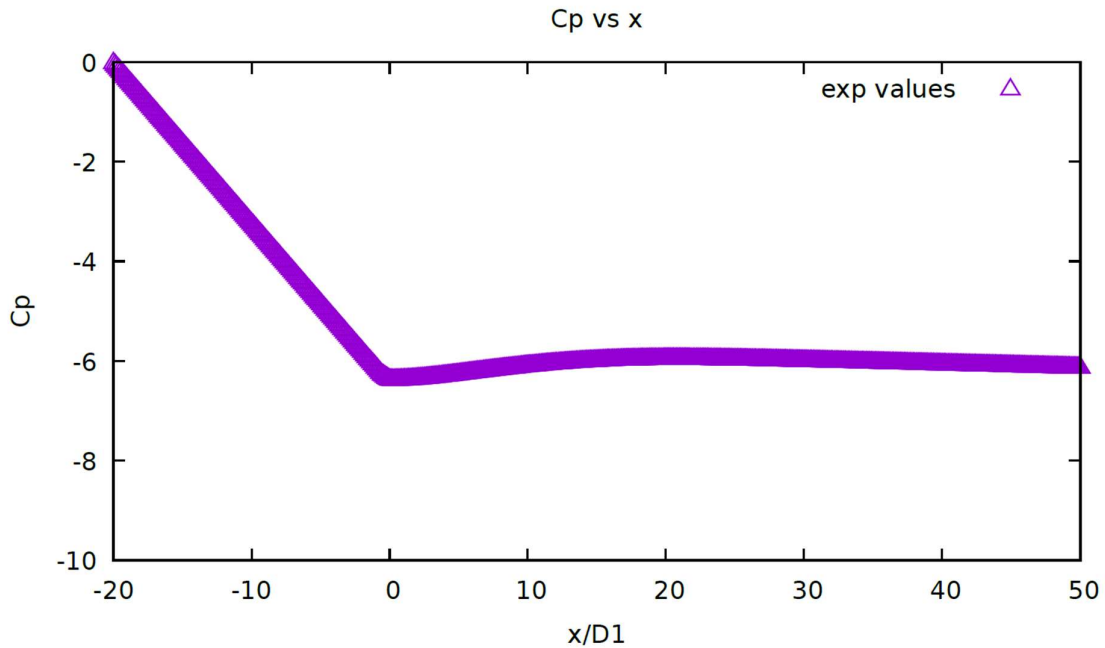


Fig 20: Pressure coefficient v/s  $x/D1$  for Re 200 sudden expansion case

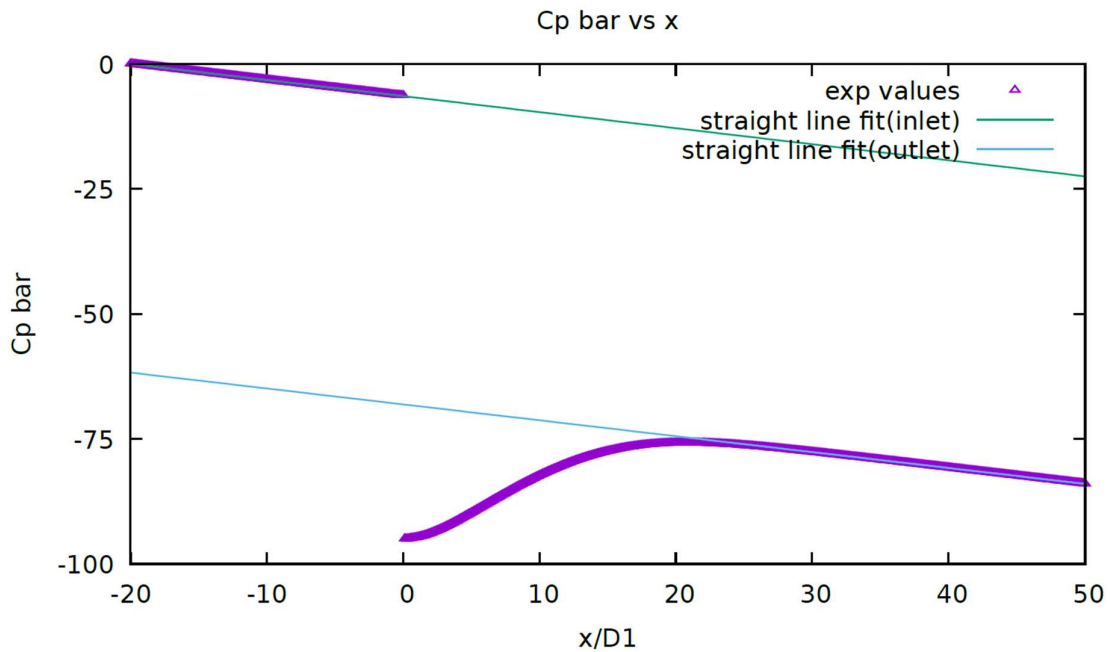


Fig 21: Modified pressure coefficient V/S  $x/D1$  for Re 200 sudden expansion case and straight-line fits

Straight line fit (inlet) corresponds to the straight-line fitting carried out for the  $C_p$  bar values corresponding to the inlet pipe or the upstream pipe and straight line fit (outlet) corresponds to the fitting done for the outlet pipe or the downstream pipe.

From fig (21), the equations for the straight-line fit come out to be (calculated by openFOAM):

Straight line (inlet pipe):  $y = -0.32079x - 6.42488$

Straight line (outlet pipe):  $y = -0.317783x - 64.8434$

Hence,  $f$  (for inlet pipe) = 0.32079

$f$  (for outlet pipe) = 0.317783

and the theoretical Darcy friction factor from eqn(9) = 0.32

Finally, the local loss coefficient ( $C_i$ ) from eqn(13) = 1.27837

This value of  $C_i$  is found to be 1.315, as calculated by the authors of the '97 published paper (can be seen in Table 3 for Re 200 [4]). So, the  $C_i$  value obtained by our simulations differs from the results of the reference paper by 2.785 %. Since the same formulae and conditions had been used to calculate the value, the error can be accounted due to the difference in the software used and the differences in their algorithm accordingly.

With this we come to the end of the project work. The following conclusion section gives a gist of the entire study and sums up the results.

## SUMMARY AND CONCLUSION

The idea of the present summer project was to simulate a laminar pipe flow of an incompressible and Newtonian fluid using the finite volume approach. Special attention was given to the velocity magnitude and pressure at the centerline points of the pipe throughout its length.

We found out the developing lengths for different values of Reynolds numbers and compared the results with the established empirical relation available in the literature. The present results differed by  $\sim 1\%$ . The Hagen Poiseuille law was also verified for the developed region.

Simulations were carried out for a 1:2.6 sudden expansion in the diameter of a pipe and its corresponding graphs were plotted in ParaView. Significant changes were found in the plots for velocity and pressure (kinematic pressure) magnitudes with the expansion of the pipe. There was a deformation in the parabolic profile of the velocity and the developed velocity in the downstream region was disturbed. The velocity curve could be seen falling right from the expansion section and after some redeveloping length, again developed flow was achieved. The fluid pressure curve was typically linear and of decreasing nature for the uniform cross-sectional area pipe flow but was found to differ from the expanded pipe flow. In the sudden expansion case, a jump in the pressure could be seen right after the expansion. Various laws stated in the literature were also confirmed to match with the obtained results.

The local pressure loss coefficient was calculated for the axisymmetric pipe expansion, with  $Re$  being equal to 200. Its value came out to be 1.27837. The relations used to arrive at this value were picked up from the published literature [Ref 4]. The motive of the authors of the reference paper was to reformulate the relations (accounting for pressure drop in a fluid flow) and to make them more accurate. However, in this research work, all calculations for  $C_i$  were done using the uncorrected forms of the equations with the assumptions that there was no wall friction and that the pressure was uniform at the expansion plane. Our aim was to get an approximate of the pressure drop coefficient ( $C_i$ ) in a fluid flow and to tally it with Table 3 of the reference paper [4]. The value of  $C_i$ , derived from our calculations, deviated by 2.785%.

## FUTURE DIRECTION

I strongly believe that I have gained immense knowledge from this project work and I am now comfortable with designing simple fluid flow simulations. This work would serve as a strong foundation and help me to build on it in the near future. At a later stage, I plan to use the understandings of this work in simulating blood flow that supplies the human heart. This would be an interesting area to work upon and I am looking forward to that day.

I must highlight here that this project work was done on a HP workstation and it took several days for the simulations to run. A supercomputer instead would be much appreciated and would have saved tremendous computational time.

## NOTATIONS

1. $Re$	Reynolds number
2. $L$	length
3. $a$ or $A$	area of cross-section of pipe
4. $D$	diameter of pipe
5. $v$	flow velocity magnitude
6. $U$	velocity magnitude (mostly used for inlet velocity)
7. $\bar{u}$	velocity averaged over area of cross-section
8. $p$ or $P$	kinematic pressure
9. $\Delta P$	pressure difference
10. $r$	radius of inlet pipe
11. $R$	radius of outlet pipe
12. $\rho$	density of fluid
13. $\mu$	dynamic viscosity
14. $Nu$	kinematic viscosity
15. $t$	time
16. $x$	x-coordinate along the centerline of pipe
17. $\sigma$	area ratio
18. $C_i$	local irreversible loss (pressure) coefficient
19. $C_p$	general pressure coefficient
20. $C_R$	reversible pressure increase coefficient $C_R = 2(1 - \sigma^2)$ ..... (eqn 14)
21. $f$	Darcy friction factor
22. 1	inlet pipe
23. 2	outlet pipe
24. $C_p$ bar ( $\bar{C}_p$ )	modified $C_p$
25. $\bar{\bar{C}}_{p01}$	y-intercept of straight line fit at $x=0$ for $C_p$ bar values of inlet pipe
26. $\bar{\bar{C}}_{p02}$	y-intercept of straight line fit at $x=0$ for $C_p$ bar values of outlet pipe
27. wrt	with respect to
28. $\nabla$	del operator



## REFERENCES

1. Fox and McDonald's Introduction to Fluid Mechanics, 10th Edition  
- Robert W. Fox, Alan T. McDonald, John W. Mitchell
2. OpenFOAM v7 user guide:  
<https://cfd.direct/openfoam/user-guide-v7/>
3. icoFoam solver (c++ files and details):  
[https://cpp.openfoam.org/v7/dir\\_3d750c9b126c5f578034b73d9637396c.html](https://cpp.openfoam.org/v7/dir_3d750c9b126c5f578034b73d9637396c.html)
4. 1997 published research paper 'Pressure drop coefficient of laminar Newtonian flow in axisymmetric sudden expansions' by the authors PJ Oliveira, FT Pinho:  
<https://web.fe.up.pt/~fpinho/pdfs/ijhffa.pdf>
5. Amazon elastic compute cloud console:  
<https://aws.amazon.com/ec2/>
6. ParaView 5.8.0 tutorial and software download:  
<https://www.ParaView.org/download/>
7. basics of linux command line:  
<https://ubuntu.com/tutorials/command-line-for-beginners#1-overview>

## APPENDIX A

### Code A1:

```

/*-----*- C++ -*-----*\
=====
\\      / F ield           | OpenFOAM: The Open Source CFD Toolbox
\\      / O peration      | Website:  https://openfoam.org
\\      / A nd             | Version:   7
\\      / M anipulation    |
\\*-----*-*/

FoamFile
{
    version 2.0;
    format ascii;
    class dictionary;
    location system;
    object blockMeshDict;
}

    convertToMeters 0.001;

// Edit wedge dimentions here
r 500; // radius of pipe in mm
L 250000; // Length of pipe in mm
th 1.5; // Wedge angle in degree

// Edit discretization here
nx 100; // along x-axis
nr 50; // radial direction
nt 1; // theta direction

wa #calc "degToRad($th/2)"; // half wedge angle in radian
yp #calc "$r*cos($wa)";
zp #calc "$r*sin($wa)";
zn #calc "-1*$zp";

vertices
(
    ( 0.0 0.0 0.0) // vertex 0
    ( 0.0 $yp $zn) // vertex 1
    ( 0.0 $yp $zp) // vertex 2
    ( $L 0.0 0.0) // vertex 3
    ( $L $yp $zn) // vertex 4
    ( $L $yp $zp) // vertex 5
);

blocks
( hex ( 0 1 2 0 3 4 5 3) ( $nr $nt $nx) simpleGrading ( 0.1 1 10)
);

```

```

edges
(
    arc 1 2 ( 0 $r 0)
    arc 4 5 ($L $r 0)
);

boundary
(
    axis
    {
        type empty;
        faces ( (0 3 3 0) );
    }
    inlet
    {
        type patch;
        faces ( ( 0 0 2 1) );
    }
    wall
    {
        type wall;
        faces ( (2 5 4 1) );
    }
    outlet
    {
        type patch;
        faces ( ( 3 4 5 3) );
    }
    front
    {
        type wedge;
        faces ( (0 3 5 2) );
    }
    back
    {
        type wedge;
        faces ( (0 1 4 3) );
    }
);
mergePatchPairs
(
);

```

Code A1: BlockMeshDict file stating geometry of pipe and wall type of each face

**Code A2:**

```

/*-----*-- C++ -----*\
=====
\\      /  F ield      | OpenFOAM: The Open Source CFD Toolbox
\\      /  O peration  | Website:  https://openfoam.org
\\      /  A nd        | Version:  7
\\      /  M anipulation|
\*-----*/
FoamFile
{
    version      2.0;
    format       ascii;
    class        volVectorField;
    object       U;
}
// *****

dimensions      [0 1 -1 0 0 0];

internalField    uniform (0.0 0 0);

boundaryField
{
    axis
    {
        type      empty;
    }

    inlet
    {
        type       fixedValue;
        value       uniform (1 0 0);
    }
    wall
    {
        type       fixedValue;
        value       uniform (0 0 0);
    }
    outlet
    {
        type       pressureInletOutletVelocity;
        value       $internalField;
    }
    front
    {
        type       wedge;
    }
    back
    {
        type       wedge;
    }
}

```

Code A2: Boundary condition for velocity(U) mentioned for each face of the

**Code A3:**

```

/*-----*- C++ -*-----*/
=====
\\      / F ield      | OpenFOAM: The Open Source CFD Toolbox
\\      / O peration  | Website:  https://openfoam.org
\\      / A nd        | Version:  7
\\      / M anipulation|
/*-----*-*/

FoamFile
{
    version      2.0;
    format       ascii;
    class        volScalarField;
    object       p;
}
// *****

dimensions      [0 2 -2 0 0 0 0];

internalField    uniform 0;

boundaryField
{
    axis
    {
        type      empty;
    }

    inlet
    {
        type      fixedFluxPressure;
    }

    wall
    {
        type      zeroGradient;
    }

    outlet
    {
        type      fixedValue;
        value      uniform 0.005;
    }

    front
    {
        type      wedge;
    }

    back
    {
        type      wedge;
    }
}

```

Code A3: Boundary condition of pressure(p) mentioned for each face of the geometry

## APPENDIX B

### Code B1:

```

/*-----* C++ -*-----*\
=====
\\      / F ield      | OpenFOAM: The Open Source CFD Toolbox
\\      / O peration  | Website:  https://openfoam.org
\\      / A nd        | Version:  7
\\      / M anipulation|
\*-----*/

FoamFile
{
    version 2.0;
    format ascii;
    class dictionary;
    location system;
    object blockMeshDict;
}

convertToMeters 0.001;

// Edit wedge dimentions here
r 500; // radius of pipe in mm
la 4500; // Length of 1st pipe in mm
lb 16200; //length of 2nd pipe in mm
th 1.5; // Wedge angle in degree
R 1300; //larger radius

// Edit discretization here
nx 100; // along x-axis
nX 300;
nr 50; // radial direction
nR 80;
nt 1; // theta direction

wa #calc "degToRad($th/2)"; // half wedge angle in radian
yp #calc "$r*cos($wa)";
zp #calc "$r*sin($wa)";
zn #calc "-1*$zp";
yq #calc "$R*cos($wa)";
zq #calc "$R*sin($wa)";
zm #calc "-1*$zq";

vertices
(
    ( 0.0 0.0 0.0) // vertex 0
    ( 0.0 $yp $zn) // vertex 1
    ( 0.0 $yp $zp) // vertex 2
    ( $lb 0.0 0.0) // vertex 3
    ( $lb $yp $zn) // vertex 4
    ( $lb $yp $zp) // vertex 5
    ( $la $yq $zm) // vertex 6
    ( $la $yq $zq) //vertex 7
    ( $la $yp $zn) //vertex 8
    ( $lb $yq $zm) //vertex 9
    ( $lb $yq $zq) //vertex 10
    ( $la $yp $zp) //vertex 11
    ( $la 0.0 0.0) //vertex 12
);

blocks
(
    hex ( 0 1 2 0 12 8 11 12) ( $nr $nt $nx) simpleGrading ( 0.1 1 10)
    hex ( 12 8 11 12 3 4 5 3) ( $nr $nt $nx) simpleGrading (0.1 1 10)
    hex ( 8 6 7 11 4 9 10 5) ( $nR $nt $nx) simpleGrading (0.1 1 10)
);

edges
(
    arc 1 2 ( 0 $r 0)
    arc 4 5 ($lb $r 0)
    arc 6 7 ($la $R 0)
    arc 8 11($la $r 0)
    arc 9 10 ($lb $R 0)
);

```

```

boundary
(
    axis
    {
        type empty;
        faces ( (0 12 12 0) (12 3 3 12) );
    }
    inlet
    {
        type patch;
        faces ( (0 0 2 1) );
    }
    left_wall
    {
        type wall;
        faces ( (2 11 8 1) );
    }
    right_wall
    {
        type wall;
        faces ( (7 10 9 6) );
    }
    outlet
    {
        type patch;
        faces ( (3 4 5 3) (4 9 10 5) );
    }
    front_small
    {
        type wedge;
        faces ( (0 12 11 2) (12 3 5 11) );
    }
    back_small
    {
        type wedge;
        faces ( (0 1 8 12) (12 8 4 3) );
    }
    front_big
    {
        type wedge;
        faces ( (11 5 10 7) );
    }
    back_big
    {
        type wedge;
        faces ( (8 6 9 4) );
    }
    interface_wall
    {
        type wall;
        faces ( (11 7 6 8) );
    }
}

);
mergePatchPairs
(

);

```

Code B1: BlockMeshDict file stating the geometry and the patch type of each wall constituting the pipe

**Code B2:**

```

/*-----* C++ -*-----*/
=====
\\  F i e l d      | OpenFOAM: The Open Source CFD Toolbox
\\  O p e r a t i o n | Website: https://openfoam.org
\\  A n d           | Version: 7
\\  M a n i p u l a t i o n
/*-----*/

FoamFile
{
    version      2.0;
    format       ascii;
    class        volVectorField;
    object       U;
}
// *****

dimensions      [0 1 -1 0 0 0 0];

internalField    uniform (0.0 0 0);

boundaryField
{
    axis
    {
        type      empty;
    }

    inlet
    {
        type      fixedValue;
        value      uniform (1 0 0);
    }

    left_wall
    {
        type      noSlip;
    }

    right_wall
    {
        type      noSlip;
    }

    outlet
    {
        type      pressureInletOutletVelocity;
        value      $internalField;
    }

    front_small
    {
        type      wedge;
    }

    back_small
    {
        type      wedge;
    }

    front_big
    {
        type      wedge;
    }

    back_big
    {
        type      wedge;
    }

    interface_wall
    {
        type      noSlip;
    }
}

```

Code B2: Boundary condition for velocity(U) specified for each face of the geometry



**Code B3:**

```

/*----- C++ -----*/
=====
\\      F ield      | OpenFOAM: The Open Source CFD Toolbox
\\      O peration  | Website:  https://openfoam.org
\\      A nd        | Version:   7
\\      M anipulation
/*-----*/

FoamFile
{
    version      2.0;
    format       ascii;
    class        volScalarField;
    object       p;
}
// *****

dimensions      [0 2 -2 0 0 0];

internalField    uniform 0;

boundaryField
{
    axis
    {
        type      empty;
    }

    inlet
    {
        type      fixedFluxPressure;
    }
    left_wall
    {
        type      zeroGradient;
    }
    right_wall
    {
        type      zeroGradient;
    }
    outlet
    {
        type      fixedValue;
        value      uniform 0.005;
    }

    front_small
    {
        type      wedge;
    }
    back_small
    {
        type      wedge;
    }
    front_big
    {
        type      wedge;
    }
    back_big
    {
        type      wedge;
    }
    interface_wall
    {
        type      zeroGradient;
    }
}

```

Code B3: Boundary condition for pressure(p) mentioned for each face of the geometry