EG501V Computational Fluid Dynamics (2017/2018)

Practical 1b: CFD modeling of laminar flow using ANSYS Fluent

1 Introduction

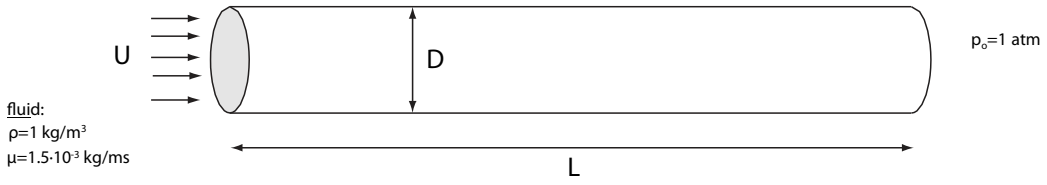
This practical shows how to solve the basic problem of fluid flow through a circular pipe. Two laminar flow scenarios are considered: (1) laminar flow in a pipe and (2) start-up flow between to parallel plates. The **learning outcomes** of this tutorial cover four key elements involving CFD simulation and analysis:

1. Build up model's *Geometry*;
2. *Mesh generation*;
3. Performing the *Fluent analysis*, and
4. *Post-processing* of results.

This tutorial will guide you step-by-step through these four procedures and provide basic skill sets to complete the assignment.

2 Laminar pipe flow

Consider a fluid flowing through a circular pipe of diameter D and length L as sketched below. The pipe diameter is $D = 0.3\,\text{m}$ and the length of the pipe is $L = 6\,\text{m}$. The inlet velocity is constant over the cross-section and is equal to $U = 1\,\text{m/s}$. The fluid density is $\rho = 1\,\text{kg/m}^3$ and the fluid viscosity is $\mu = 1.5 \times 10^{-3}\,\text{kg/ms}$. Note that these properties result in a Reynolds number of $Re = \rho DL/\mu = 200$ which means the flow is laminar. The outlet pressure p_o is equal to 1 atmosphere (standard pressure).



Solve this flow problem using ANSYS Fluent and present the following results:

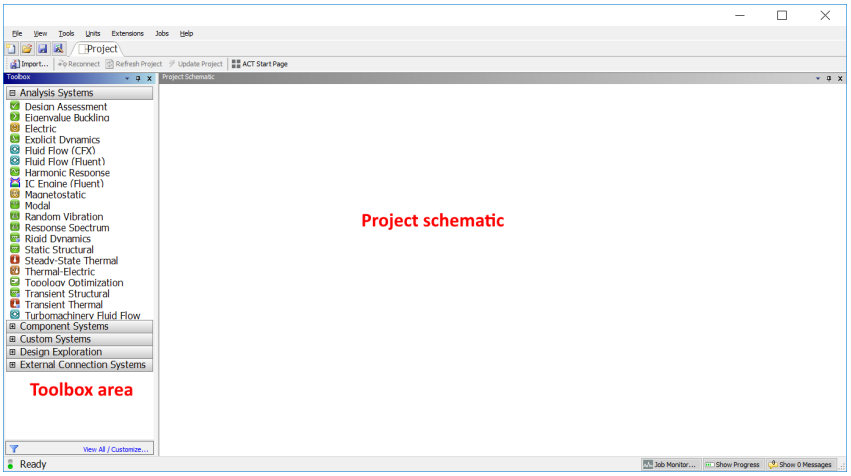
- velocity vectors
- velocity magnitude contours
- velocity profiles
- skin friction coefficient along the pipe wall

Note that the physical problem of flow in a pipe is a 3D problem, however because the flow across any plane passing through the centreline of the pipe is identical we can model the flow in 2Dimendions. Moreover because in this "slice" of dimensions $D \times L$ the flow is symmetrical around the centreline we only need to simulate the flow in one half of the plane of dimensions $r \times L$. Flows which are symmetrical around a centreline are also called *axisymmetric* flows and by modelling it in 2D we save considerable computational resources compared to full 3D simulations.

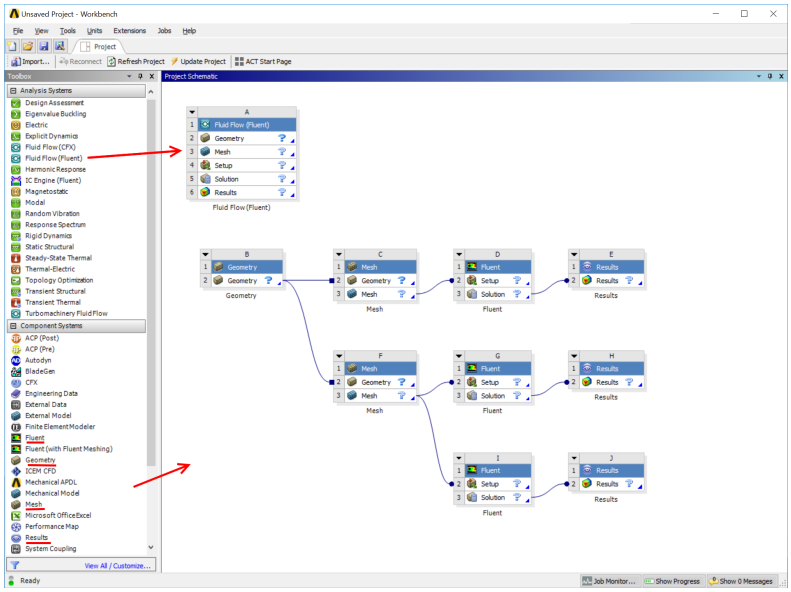
2.1 Starting ANSYS workbench

Prior to opening ANSYS, create a folder (e.g. FluentCFD) on your *Homedrive*. This folder will be used to store the files created during this session, and allow you to reopen them at a later stage. You will need approximately 10 MB of free space for this tutorial, so make sure this space is available on your *Homedrive*.

On the desktop go to the folder **Physical Sciences→Engineering→ANSYS 18.1** and double click the shortcut **Workbench 18.1**, see the figure below need to replace with 18.1.
You should now see a window as in the figure below.



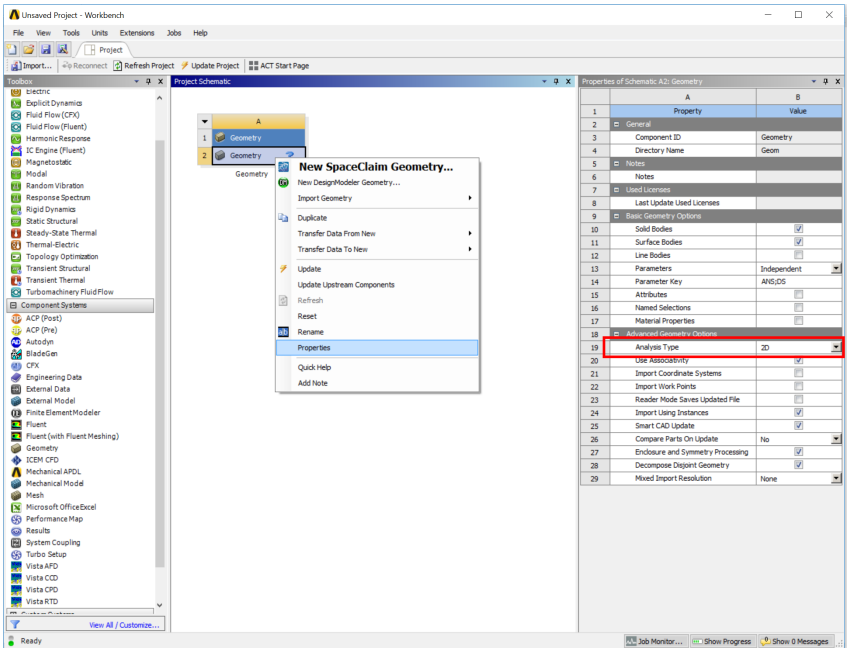
This is the ANSYS workbench which consists primarily of the *Toolbox area* (left column) and the *Project schematic* in the main area. Above these areas you can find the usual toolbar and the menu bar. The main way to work in the workbench is to drag an item from you toolbox to a desired location in the Project schematic. Alternatively you can double-click any item in the *Toolbox area* and it will appear in the *Project schematic*, although you do not have any control over its location this way (it will appear on the next line). The *Analysis Systems* contain all the necessary components to model a specific problem. So in the case of *Fluid Flow (Fluent)* it contains the following components to complete the fluid flow simulation: *Geometry, Mesh, Setup, Simulation, Results*. The components are automatically linked and you would normally work through these components from top to bottom (i.e. you cannot work on the *Mesh* before the *Geometry* is completed etc.). An example of this is object *A* in the project schematic in the figure below. Here we are going to set-up our simulation from the individual components in the *Component Systems* menu and establish the links between components ourselves. This gives us greater flexibility at later stages to link for example the same geometry to different meshes (e.g in the figure below geometry or to link the same mesh to different set-ups. An example is shown in bottom part of the project schematic below where geometry component *B* is linked to mesh components *C* and *F*, and mesh component *F* in turn is linked to Fluent set-ups omponent *G* and *I* (do not create this now!).



2.2 Geometry

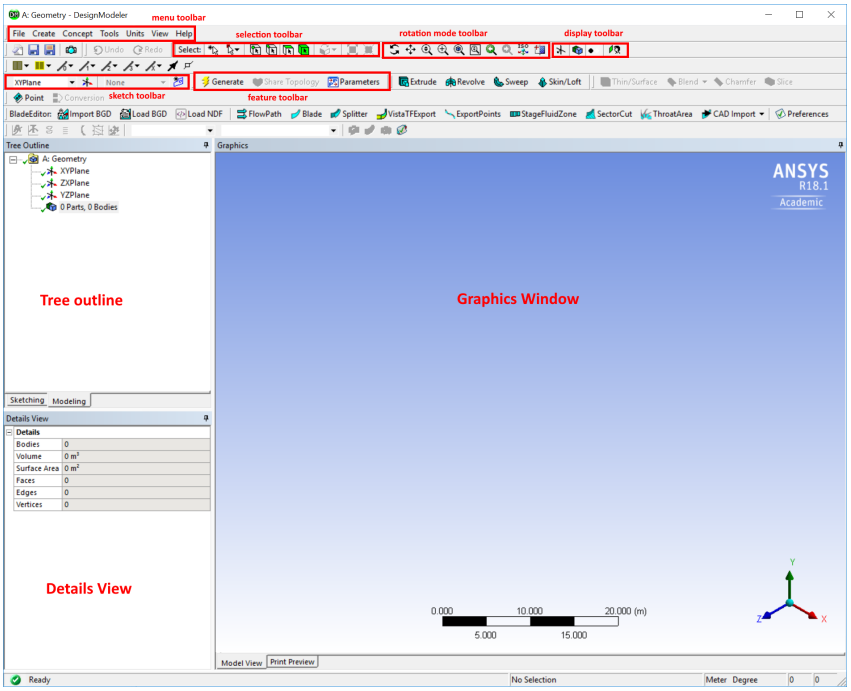
2.2.1 Make a sketch

Now click the boxed + symbol in the **Component Systems** menu and drag the **Geometry** component to the Project schematic. Right click the **Geometry** cell and select **Properties**. Change the value of **Analysis type** from the default value 3D into 2D as indicated below.

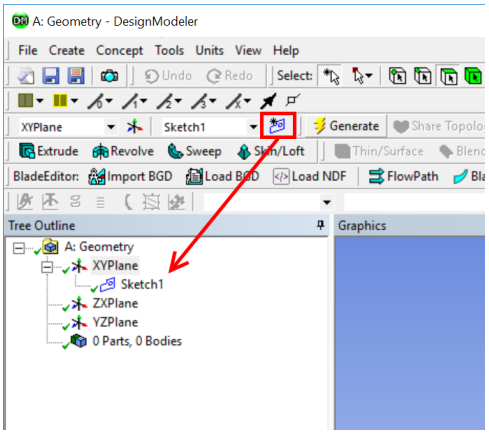


Right-click **Geometry** again and select the second option **New DesignModeler Geometry**. We do not choose the default *SpaceClaim* option to make our geometry, *SpaceClaim* is ANSYS's inbuild CAD software (similar to SolidWorks) which can be used to make very complex geometries, but similar to SolidWorks also requires more time to learn. Because in this course we only use very simple geometries, and because learning another CAD package is not the aim of this course, we are going to use the more basic *DesignModeler*.

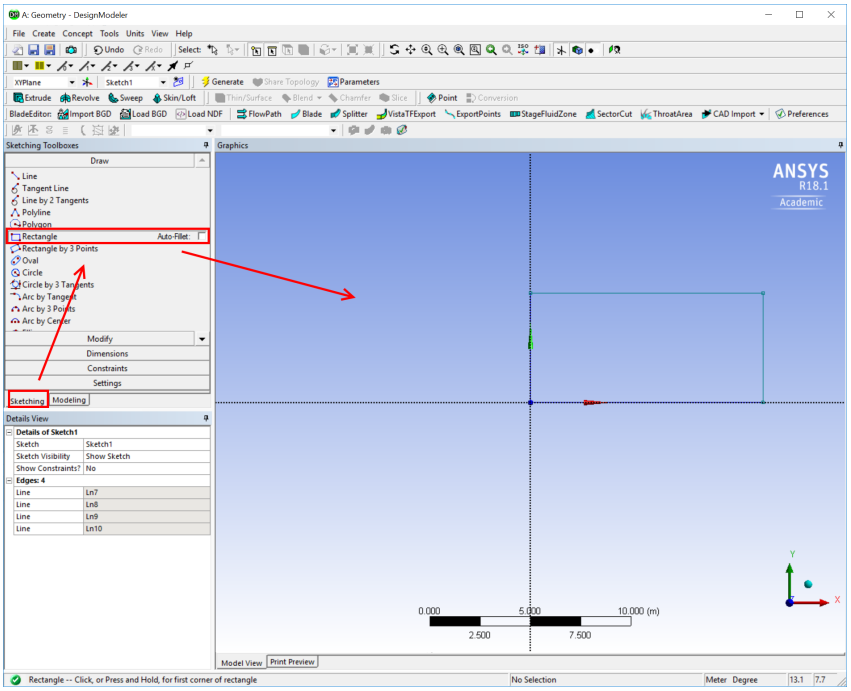
The *DesignModeler* window will now have appeared. The main areas and toolbars that you will be using are displayed in the image below. We will use these expressions (i.e *Tree outline*, *Details View* etc.), to guide you trough the rest of the geometry section.



You will now create a sketch in the XY-plane. First select the **XYplane** in the *Tree Outline* on the left hand side of your window and click the **New Sketch** symbol in the sketch toolbar. **Sketch1** should now appear under the **XYPlane** in your *Tree Outline* as shown below.

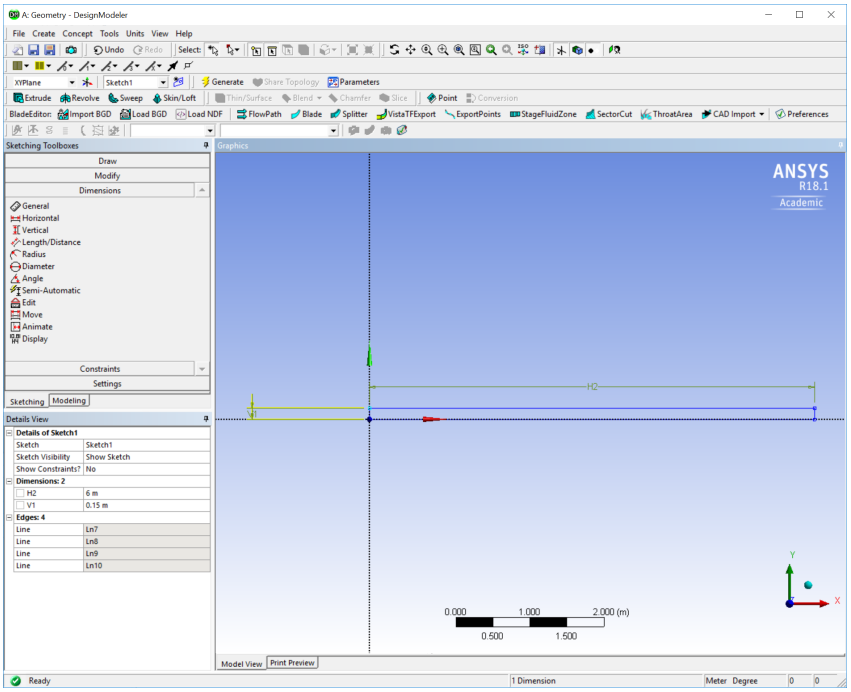


Next, in the *Graphics Window* click on the **Z-axis** in the coordinate system to obtain a 2-dimensional view of the **XYPlane**. Now go to the **Sketching tab** in the *Tree Outline*, select **Rectangle** and draw a rectangle in the *Graphics Window* by clicking first on the origin (a "P" will appear if you hover the mouse over the origin, this P indicates your mouse pointer is at a point of intersection, similarly a "C" will appear of your pointer is on a line - note that the P might be difficult to distinguish in this case since it is overlapped by both axes, the origin and your mouse pointer!). The dimensions of the rectangle (or pipe in this case) do not matter at this stage since we will specify the dimension in the next step.



2.2.2 Setting the dimensions

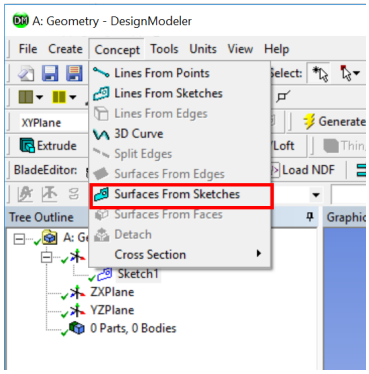
Here we will set the dimensions of the rectangle, which only consist of a horizontal and vertical dimension. Select the **Dimensions** menu in the **Sketching tab** and select **Vertical**. In the *Graphics Window* click once on the upper boundary of your rectangle and once on the lower boundary, then drag "V1" outside the left side of the rectangle and click once more. Note, the label "V1" denotes the "vertical1" where the number 1 indicates the sequence number in your dimensioning (so this would be 2 if you had decided to dimension the horizontal dimensions first). Repeat the procedure for the horizontal dimension by selecting **Horizontal** in the **Dimensions** menu and selecting the left and right boundaries of your rectangle.



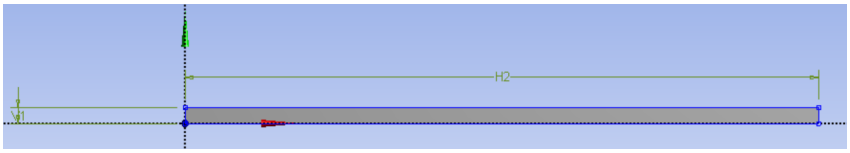
Set the dimensions in the *Details View* table, located in the lower left corner. Remember that the flow is *axisymmetric* therefore we only needed to model a plane of height $R = D/2 = 0.15\,\text{m}$, therefore V1=0.15 m, and length $L = 6\,\text{m}$, therefore H2=6 m,

2.2.3 Creating the surface

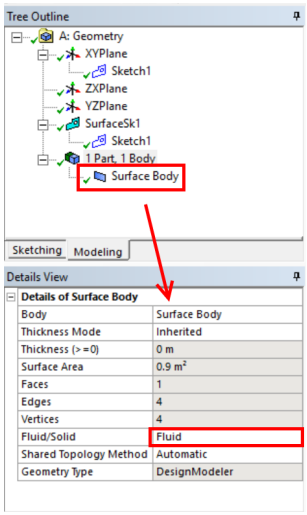
Select the **Sketch1** in the modelling tab of the *Tree Outline*, the four lines of your created rectangle should turn yellow. Now select **Concept** in the *menu bar* and select **Surfaces from Sketches** as shown below.



Click **Apply** in the *Details View* table in the lower left corner and click the Generate button in the *Feature Toolbar*. Your created geometry should now look as follows



Now, we only need to tell the software that the surface we have created is part of the Fluid domain, and not the solid domain which it assumes by default. In the Tree Outline, select the "+" symbol next to *1 Part, 1 Body* and select the *Surface Body* that was created. In the details view below change the domain to *Fluid* as shown below.



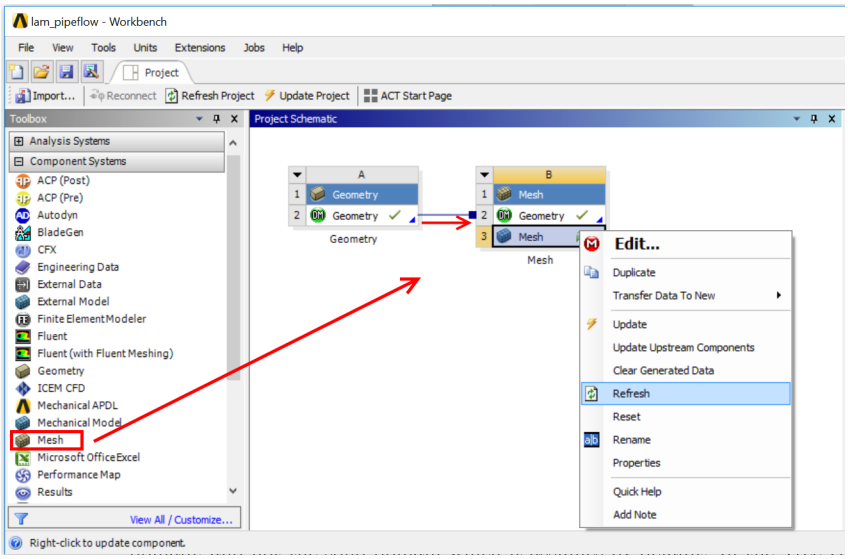
2.2.4 Save the project

Now that you have successfully created your pipe geometry you can close the *DesignModeler* window and return to your *Workbench* window. Note how a ✓ symbol has appeared in the Geometry cell, indicating that you have completed a geometry (if you only draw a single line the ✓ symbol would still have appeared, so it does not necessarily mean you have drawn an valid geometry!).
In the workbench you can save the project by clicking **Save** or **Save As..**, give your project a sensible name, e.g. LamPipeFlow and save it in your working directory on your *Homedrive*. Note that not only the file **LamPipeFlow.wbpj**, is created within your working directory, but also the folder **LamePipeFlow_files**. In order to open your project at a future occasion you will need both the ***.wbpj** file and the accompanying folder with its contents.

2.3 Mesh generation

2.3.1 Launching the meshing window

Drag the **Mesh** component from the *Component Systems* menu to the *Project Schematic*. Establish a link between the geometry and mesh, by dragging the **Geometry** cell from your Geometry component table, to **Geometry** in the Mesh component table. A line will be established between the two and the ✓ symbol will appear next to Geometry in the Mesh component box. Now the Mesh needs to be made aware of the added geometry: right-click the **Mesh** cell and select **Refresh**, after about 5-10s the refreshing procedure is complete and a lightning bolt symbol will appear next to the Mesh.

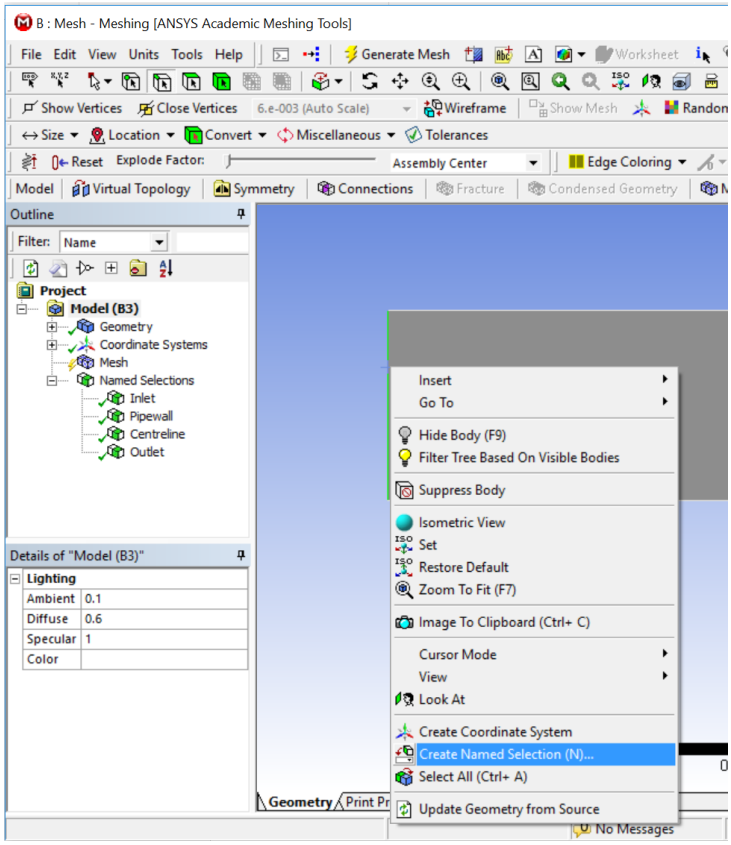


Now right-click **Mesh** and select **Edit** (or double-click **Mesh**), this will open the *Meshing* window. The outline of this window is similar to the *DesignModeller* window which you used to create your geometry, ie. on top you see the various *toolboxes*, in the upper left column you see your project's *Tree Outline*, where you can select different components of your model, in the lower left column you see *Details View* table of the selected component, and the main window contains your *Geometry Window*. Below the geometry you will now also see a *messages* window which displays any error messages that might result from your actions.

2.3.2 Creating named selection

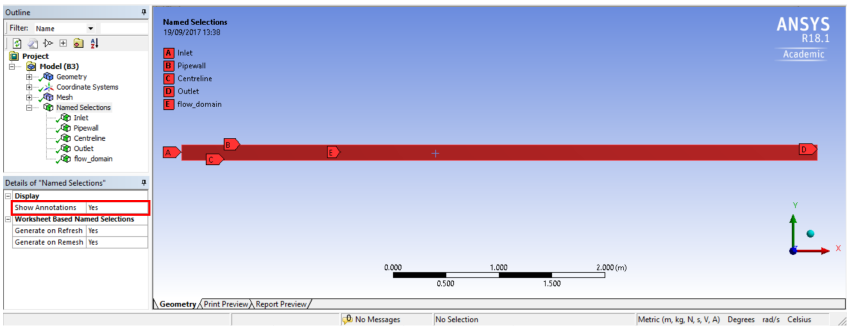
Before we generate the mesh we will first give names to the four edges of our geometry. This will allow us to recognise the edges more easily and assign boundary conditions in the Fluent solver later on.

Click the edge tool symbol in the *selection toolbar* and click on the left edge of your geometry in the *Geometry Window*. Right-click on the green highlighted line and select **Create Named Selection** as shown below. Note that you can zoom in and change the view of your geometry using the various magnifying glass tools in the *rotation mode toolbar* at the top of your window. To restore the view of your entire geometry click the zoom to fit symbol .



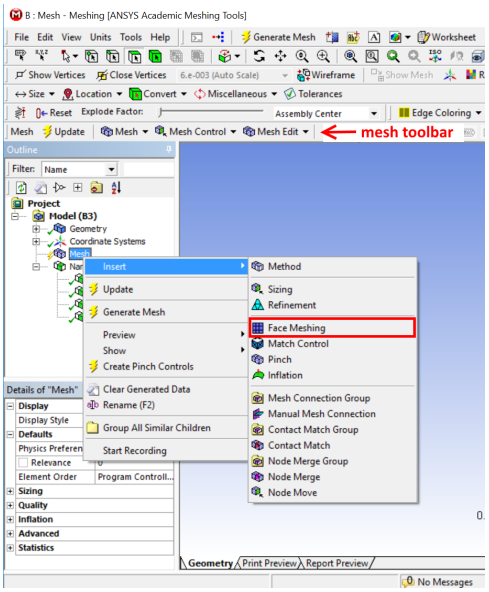
Give the boundary the name **Inlet** in the Selection name window that appears. Repeat the procedure to name the upper (**Pipewall**), right (**Outlet**) and lower (**Centreline**) boundaries. Your four named boundaries should now have appeared under the **Named Selections** item in the *Tree Outline*. Finally, we are also going to name the interior part of our domain: click on the *named selection* in the Project Tree and then choose the face select tool , click on the gemoetry such that it highlights green, now you can rightclick on the surface and rename it to *flow_domain*.

You can display the named selections in your geometry by changing the **Show Annotations** to "yes" in the *Details View* table as shown below.

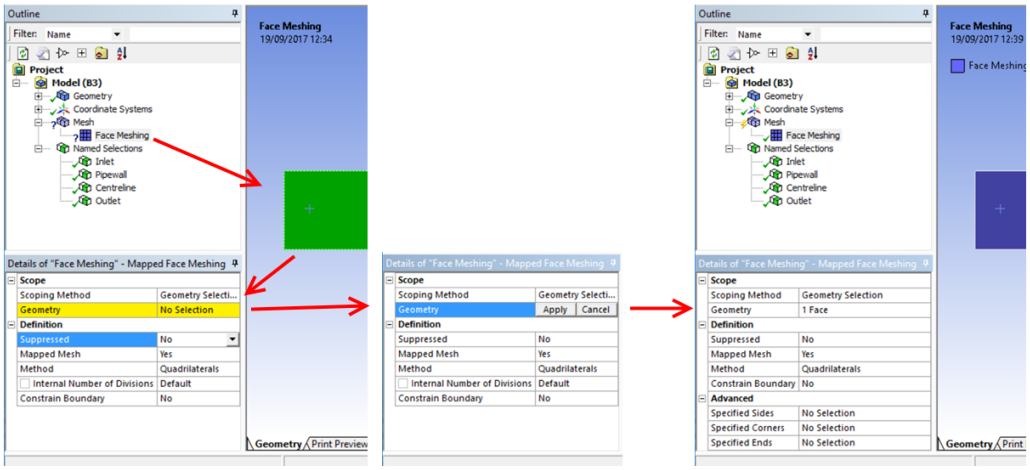


2.3.3 Creating uniform structured Mesh

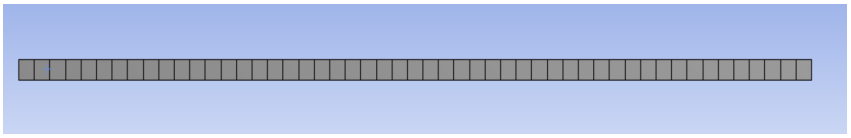
Here we are going to create a *structured* or grid style mesh on a selected face, in this case our pipe. Right-click **Mesh** in the *Tree Outline*, select **Insert** and then select **Face Meshing** from the drop-down menu, as shown below (alternatively, you can select **Face Meshing** from the **Mesh Control** toolbar indicated below).



After clicking face meshing you will see in the *Details View* that the Geometry box is highlighted in yellow and indicates "No selection", this means that we still have indicate to which surface we need to apply the Face Mesh. In our case the Geometry will be the pipe surface, so therefore click on the pipe face in the Geometry Window, the surface should should highlight green. Now go to the *Details View* of the face meshing menu and click on **no selection** in the yellow Geometry cell, then click **Apply**. The Geometry cell should now list **1 Face** and our pipe surface selected face should have turned blue in colour.

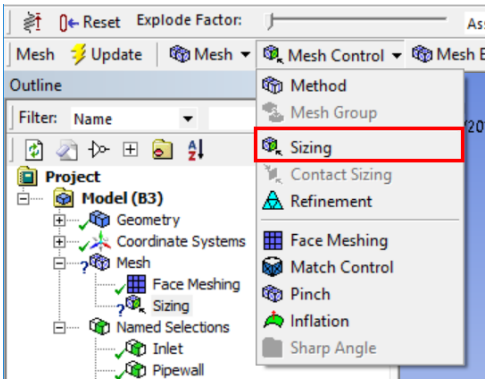


We are almost there. You still need to generate the mesh: right-click **Mesh** in the *Tree Outline*, and select **Generate Mesh**. Great, we have now created our default mesh, which can be seen if you select **Mesh** in the Tree Outline. The mesh should look as follows:

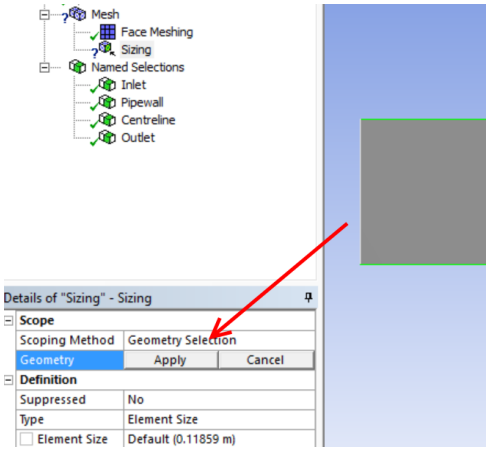


2.3.4 Edge sizing

Our default mesh only consists of 51 elements (this can be seen in the *Details View* of "Mesh" by double-clicking on statistics *Statistics*). This is rather coarse mesh, in particular in the vertical (radial) direction we only have one cell. There we are going to increase the mesh to 600 elements, 8 in the vertical and 75 in the horizontal. We can achieve this by specifying the number of divisions on the edges of our geometry. We will start by specifying the number of divisions at the Pipewall and the Centreline. Click **Mesh Control** in the *mesh toolbar* and select **Sizing** from the drop-down menu:



Now, make sure that you have the edge tool selected, then hold down the Ctrl-key and select the top and bottom edge of your geometry, which should highlight green, then click **Apply** in the *Details View* table as follows:



Change the cells in the Details View table as follows:

Details of "Edge Sizing" - Sizing	
Scope	
Scoping Method	Geometry Selection
Geometry	2 Edges
Definition	
Suppressed	No
Type	Number of Divisions
Number of Divisions	75
Advanced	
Size Function	Uniform
Behavior	Hard
Bias Type	No Bias

Click the update button Update in the *mesh toolbar* to update your mesh. Now repeat the process to apply 8 divisions to the left (inlet) and right (outlet) boundaries of your pipe.

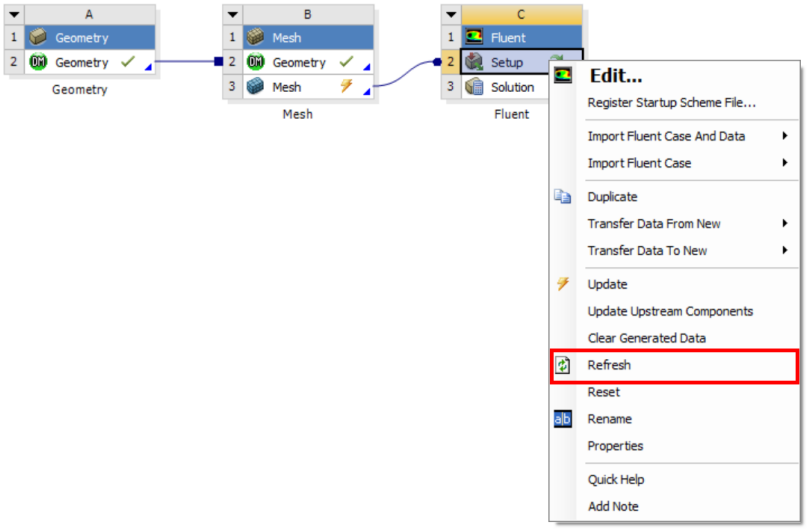
Select **Mesh** in the *Tree Outline* to view your created mesh, which should look as below. Verify in the **Details of Mesh** table that your mesh contains the required 600 elements. (Note that by changing the sizing behaviour from *soft* to *hard* has forced the divisions along an edge to be of equal size. At this stage you can check yourself what effect the behaviour setting has on your mesh).

Save your project and close the *Meshing* window, then in your *Workbench* click the Update Project button and make sure to save your project once more.

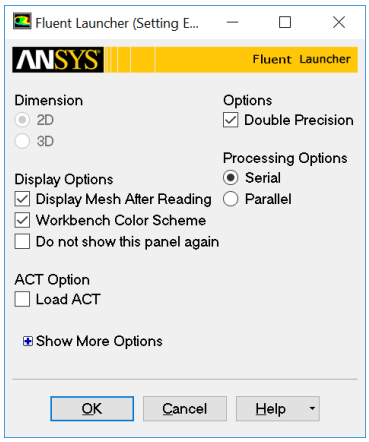
2.4 FLUENT analysis

Now, we are at the third stage, the Fluent Solver. This aspect contains of two main steps: 1. setting-up our mathematical model and 2. specifying how we are going to solve our mathematical model. In the first step we need to specify the governing equations and boundary conditions for our problem and while in the second step we specify which discretization methods we are going to use when solving the equations.
In the *Workbench* drag the **Fluent** component system to the *project schematic* and then drag the **Mesh** cell from the mesh component table to the **Setup** cell of your Fluent component table. This action might cause a symbol to appear next to you previously generated mesh, if so,

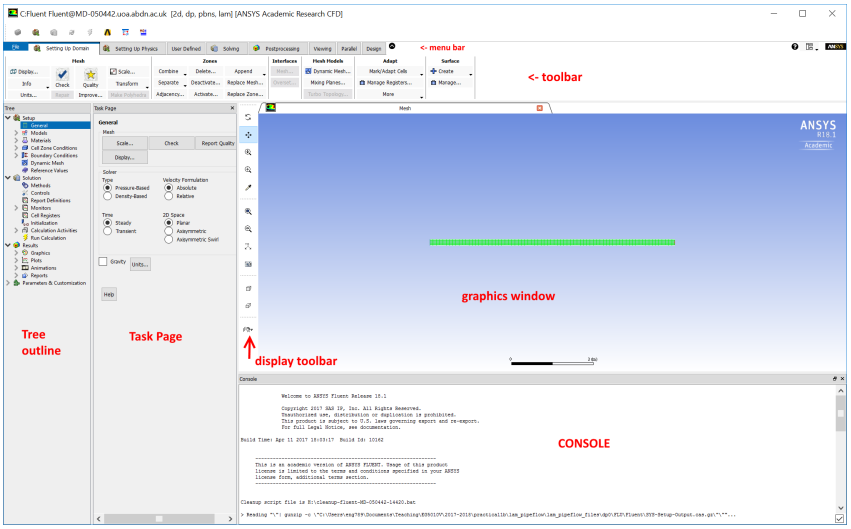
right-click on the mesh and select **Update**. Now, right-click on **Setup** in the fluent component and select **Refresh** from the drop-down menu.



Double-click **Setup** (or right click and press Edit), then change the options to **Double Precision** in the Fluent Launcher window that appears, then select **OK**.



After a while, the Fluent interface will show up, which consists of the areas shown below. As before we will use these terms to guide you through the process.



In the *Project Tree* you can see the two main components discussed above, the first is *Setup* in which we define our mathematical models and *Solution* under which we going to specify how we are going to solve the mathematical model. In the Fluent analysis we will essentially work our way through the items in the *Project Tree* from the top until *run calculation*. The activities can be grouped as follows:

Perform pre-analysis checks:

1. Solving any warning messages.
2. Mesh checking.

Setting up the mathematical model:

3. Select Model and material properties
4. Setting up the boundary conditions.

Specify how to solve the model:

5. Setting up the solution methods and controls.
6. Monitoring solution convergence.

Perform the calculation:

7. Initializing the simulation.
8. Running the calculation

In the following section we will go through these activities. Note that analysis (i.e. *post-processing*) of the results will be addressed separately in Section 2.5.

2.4.1 Checking warning messages

It is good practice to scroll through the Console window and check if there any warning messages displayed. If you have followed all the steps correctly so far there will not be any warning messages displayed. However, if you do see warning messages at this stage than you will have made an error in the geometry or meshing stages. The warning message will give a description of the error which you will first need to correct before proceeding in the Fluent solver.

2.4.2 Check Mesh and Display

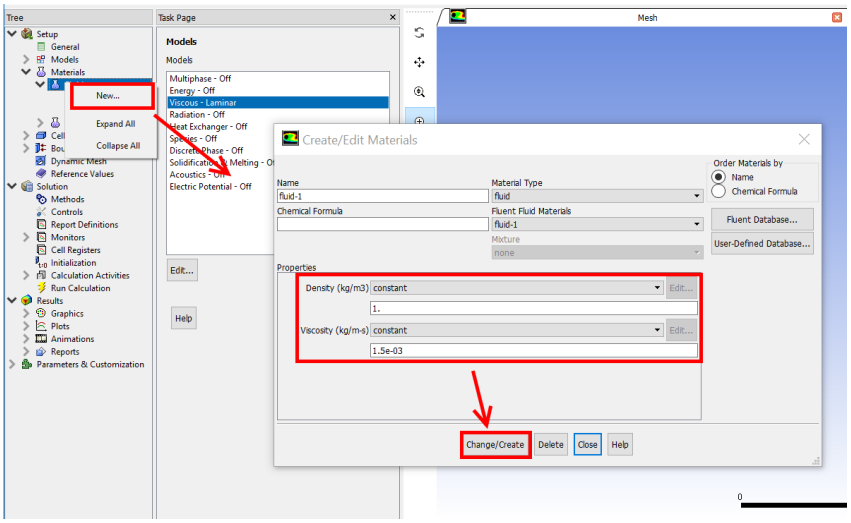
In this step we are going to check whether the mesh is correctly imported into the fluent solver. The first step before we are going to perform this check is telling Fluent that the model is an *axisymmetric*: click **General** in the *Navigation Pane* and then in the *Task Page* select **Axisymmetric** under 2D space. In the Mesh menu select select **Check**, the console should now display the following:

```
Domain Extents:
  x-coordinate: min (m) = 0.000000e+00, max (m) = 6.000000e+00
  y-coordinate: min (m) = 0.000000e+00, max (m) = 1.500000e-01
Volume statistics:
  minimum volume (m3): 8.835729e-05
  maximum volume (m3): 1.325359e-03
    total volume (m3): 4.241150e-01
  minimum 2d volume (m3): 1.500000e-03
  maximum 2d volume (m3): 1.500000e-03
Face area statistics:
  minimum face area (m2): 1.875000e-02
  maximum face area (m2): 8.000000e-02
Checking mesh.........................
Done.
```

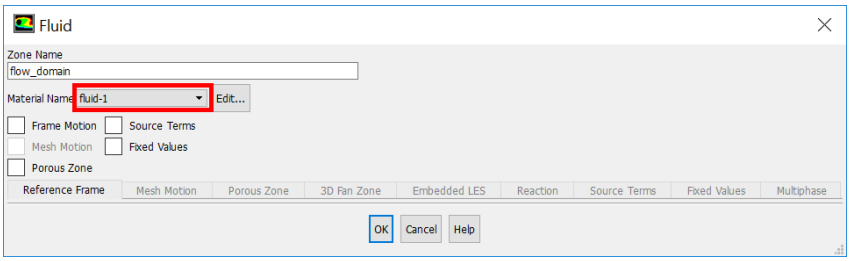
If console displays *Done* after the mesh checking than your mesh is approved. If it indicates errors than you will need to go back to the meshing stage and correct the errors. More information can be obtained from the Mesh Info buttom which you can find in the toolbar. The Size Info will display information on the number of Cells (elements) which we have previously set 600.
Click **Display** in the *Task Page* or in the *Toolbar*. In the Mesh Display dialog box that appears note that under Surfaces our four named boundaries are listed, and the (interior) fluid domain. Select all five surface and click the **Display** button. Note how your mesh is displayed in the Graphics Window. Use the various tools from the Graphics Toolbar to investigate the mesh in more detail, remember that you can restore the full view by clicking  in the display toolbar. **Close** the Mesh Display dialog box.

2.4.3 Select Model and material properties

In this section we will essentially tell Fluent which are the governing equations that need to be solved. In any fluid flow problem we will solve the *continuity equation* and the *momentum equation*, these you therefore do not have to select in Fluent. However, you do have to tell Fluent which type of the momentum equation you will solve, i.e. whether it is for inviscid flow, laminar flow or for turbulent flow. In our case the flow is laminar which we can specify by selecting under **Models** in the *Navigation Pane*. Note how the *Task Page* will display a list of models, in this case we only have to set the **Viscous** model. Select the **Viscous** model and click **Edit**. The Viscous Model dialog box will show a of the various ways in which we can solve the momentum equation in Fluent, besides the inviscid and laminar models this list contains various types which can be used to solve turbulent flow (lec and the next tutorial). We are dealing with laminar flow in this problem so for now leave the viscous model selected at **Laminar**, click **OK** to exit the dialog box. Ensure that all the other models (Multiphase, Energy, etc.) are switched off, these are not part of our flow problem.
The next step is tell Fluent what type of fluid we are dealing with, because the density and viscosity of the fluid appear in the governing equations. First of we need to create our fluid: click on **Materials** in the *Tree Outline* and then right click on **Fluid** and select **New**. In the dialog box that appears you can enter the density and viscosity of the fluid. Note there is also a button *Fluent database* in this dialog box, this allows you to pick a fluid from a vast database incorporated in Fluent. In this case we are going to manually enter our fluid properties, so change the properties to our working fluid ($\rho = 1.0\,\mathrm{kg/m^3}$ and $\mu = 1.5 \times 10^{-3}\,\mathrm{kg/ms}$). Click **Change/Create** to close the window (note that the new fluid is named by default *fluid-1* - you can change this name if you like). Under *Fluid* in the *Tree Outline* you will now find your newly created fluid named *fluid-1*.



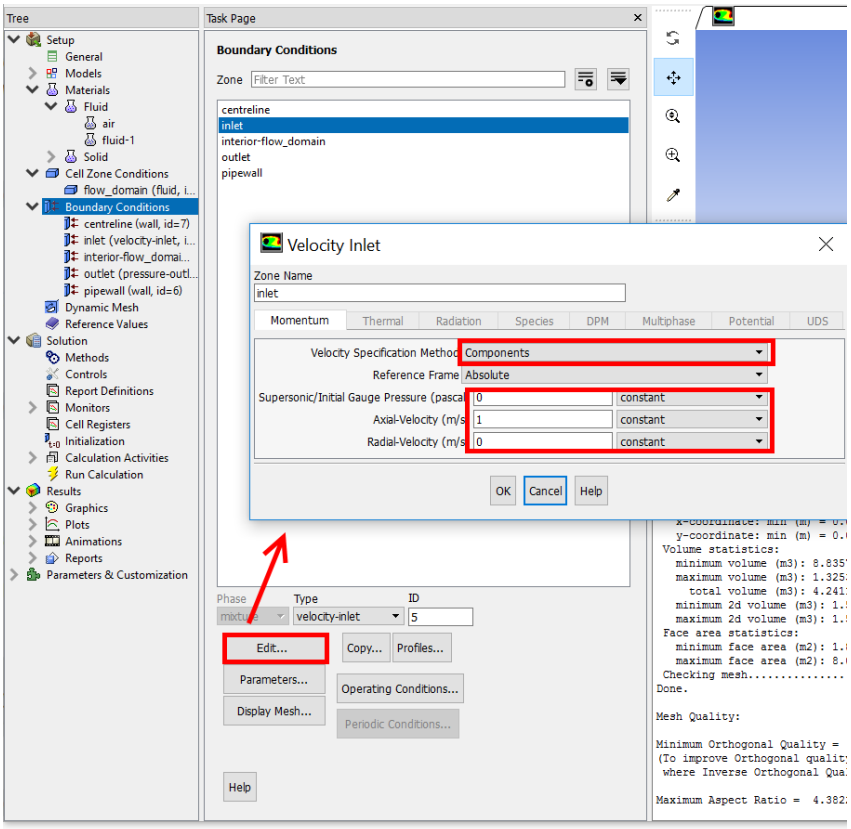
Now we that we have created our fluid we still need to assign it to our fluid domain as follows, under **Cell Zone Conditions** in the *Project Tree* double-click on the **flow_domain**. In the dialog box that appears select our newly created *fluid-1* under **Material Name** and select **OK**.



2.4.4 Boundary Conditions

Here we are going to specify our boundary conditions. We have four boundaries to our domain (inlet, outlet, pipewall and centreline) so we have to set 4 boundary conditions (refer to fig 1 for b.c.'s??):

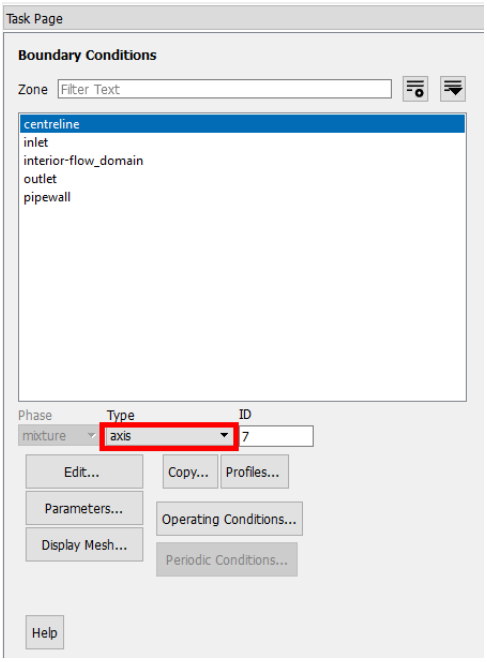
1. *inlet*: double-click **Boundary condition** in the *Project Tree*, select **inlet** in the Task Page. In the bottom of the Task Page you can see under *Type* that Fluent has already correctly guessed that the Type of boundary conditions is *velocity-inlet* boundary condition. There are many more types of boundary conditions, but Fluent has made this guess (correctly) because we have named the boundary Inlet. Now click **Edit**, this will display the velocity inlet dialog box. Set the **Velocity Specification Method** to **Components** and set the **Axial Velocity** to 1 m/s as shown below, then click **OK** to close the velocity inlet dialog box.



2. *outlet*: explain how fluent works with gauge pressure rather than absolute pressure. Before we are going to specify the outlet boundary condition we are first going to set the operating pressure. Select **Operating Conditions**, in the dialog box that appears you will see that the operating pressure is set to 1 atm (101325 Pa), therefore leave this setting as it is and click *OK*. Now select **outlet** in the boundary conditions task page and check if the **Type** is automatically set to **pressure-outlet** (this should be the case, but if it is not, change the type to pressure-outlet!). No further changes are needed since the gauge pressure by default is set to 0 Pascals (you can verify this by clicking Edit).

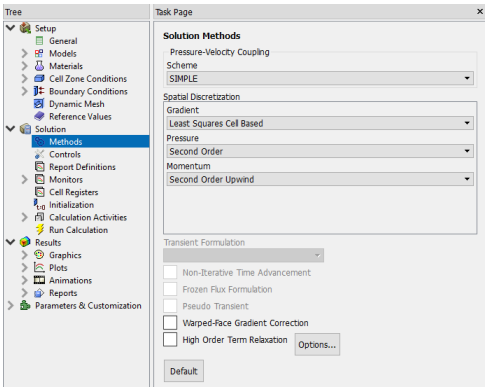
3. *pipewall*: select the **pipewall** in the boundary conditions task page and verify that the **Type** is set to **wall**, if it is not then change it to **wall**. Click *edit* and verify that the *wall motion* is set to a stationary wall (our pipewall is stationary) and the *shear condition* is set to No Slip (which means zero velocity at the wall). Click **OK** to close the dialog box.

4. *centreline*: select the **centreline** from the boundary conditions task pane. You will notice that the **Type** is set to wall by default. This is incorrect guess by Fluent, because we want Fluent to treat it as an axis of symmetry, therefore change the type to **axis**. Click **OK** in any of the dialog boxes that appear.

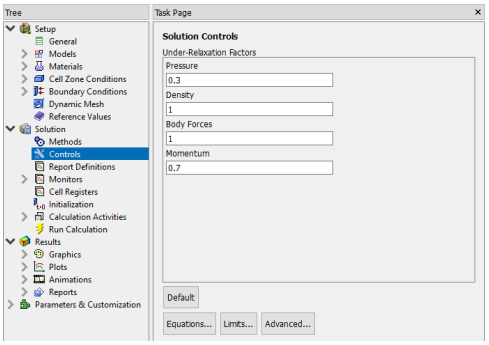


2.4.5 Solution Methods and Controls

The solution methods and controls refer to how the governing equations are discretisized (Lecture Notes 3). Here we will use the 2nd order spatial discretization to solve our momentum equation. Select **Solution Methods** from the *Project Tree* and verify that the settings are as follows. The other settings we are leaving our settings at their default values, however by checking the other options in the drop-down menus you can verify that Fluent contains many solution methods!

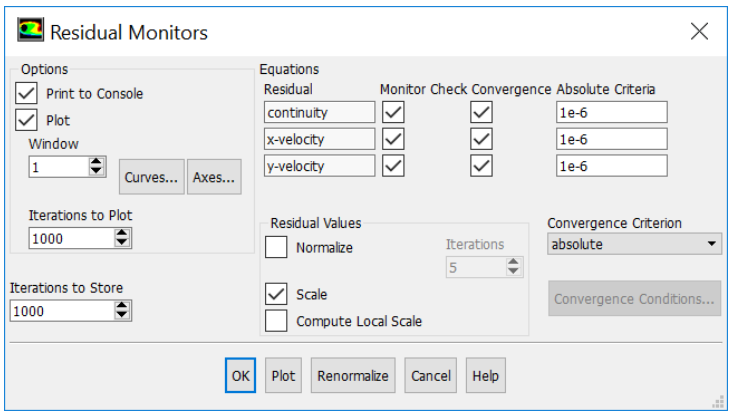


Select **Controls** in the *Project Tree*. In the *Task Page* you will now see a table with **Under-Relaxation factors**. Recall the *Successive Over-Relaxation* method in Lecture Notes 4, where an over-relaxation factor ($\omega > 1$) was introduced to accelerate the iteration process and reach faster convergence. Here *Successive Under-Relaxation* is applied ($\omega < 1$) which reaches convergence slower, but leads to a more stable solution. Finding the right relaxation factors is a largely empirical exercise and based on experience. The default values in Fluent are set to values that are near optimal for a large number of situations, we will therefore adopt the default values:



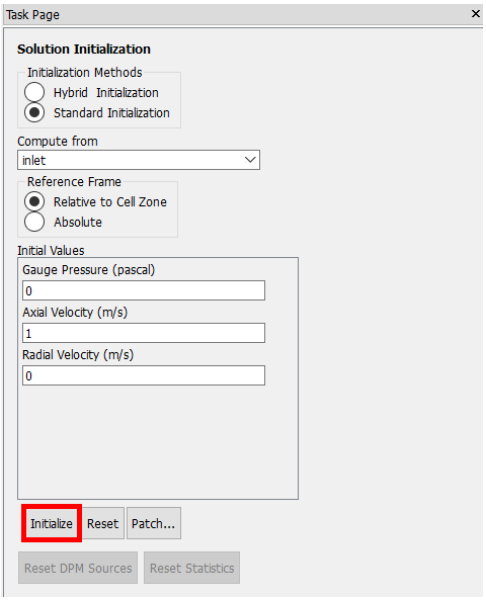
2.4.6 Monitoring solution convergence

add some comment here on residuals - can we refer to lecture notes?
Double-click **Monitors** in the *Project Tree* to show the various Monitor options and then double-click **Residual** which shows the Residual Monitor dialogue box. The monitors provide an overall view of the iteration convergence of the equations that we are solving (here continuity, x-velocity and y-velocity) at all nodes after every iteration. At every iteration the residual (Lecture Notes 4) is compared to the specified **absolute criteria**, if the residual is less then the specified criteria the solution of the equation is considered to be converged. In our case we are going to set all three convergence criteria (**Absolute criteria**) to 1e-6. Ensure that all other setting in the dialog box are set as follows:



2.4.7 Initialisation

Before we can run the simulation we first need to give all the variables at each node an initial value (our *initial condition*, refer to lecture notes??). In our case we only have three variable (pressure, axial and radial velocity) therefore we have three initial conditions to specify. As our initial condition we are going to specify a radial velocity of 1 m/s in every cell in our flow domain. Double-click on **Initialisation** in the project tree, and select **Standard Initialization** in the *Task Page*. Choose **inlet** in the **compute from** drop-down menu. Ensure that the values are defined as below and press **Initialize**.



2.4.8 Run the calculation

Finally it is time to run our calculation! Before we do so however, we want to save a variable for later use in the post-processing stage which by default is not saved by Fluent. Go to **Files** in the *menu bar*, select **Data File Quantities** and in the right column select **Skin Friction Coefficient** and click **OK**.
Now select **Run Calculation** from the *Project Tree*. We now need to set the **Number of Iterations**, which we set here to 100. Remember that we have also specified convergence criteria before, therefore our calculation will finish either when the specified number of iterations are completed, or when all the specified convergence criteria are met. Note that it is good practice to specify a small number of iterations initially (say 100) until you are sure that the solution will converge (this you can see from your residual monitors). This will avoid unnecessarily long computations in case of a non-converging iteration due to errors in the set-up. If you have verified that your simulation is converging after the initial iterations you can simply continue the simulation by pressing **Calculate** again (you can change the number of iterations if you want). While the calculation is running you can switch between the residual monitor and drag force monitor view in the top left corner of your *Graphics Window*.

After about 76 iterations the calculation stops since all the residuals have fallen below the specified convergence criteria of 1e-6. Note that although the console will display that the *solution is converged* as shown below, this does not necessarily mean that the simulation is really converged! Remember that the solution criteria were defined by ourselves, so if we had put there some large numbers the simulation would probably have converged after a few iterations. We will therefore need to make the judgement ourself based on the residual history and the variable history. The converged simulation would include two factors: 1) a small residual and 2) negligible changes of our variable. Here we have set the drag coefficient as our monitor variable and we can see from the graphics window and in the console (fifth column) that our drag coefficient has reached a steady value after approximately 55 iterations.

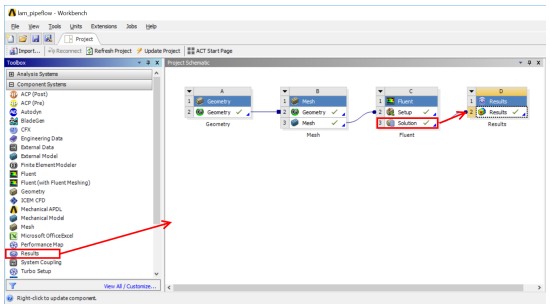
Now that the simulation is completed we can start with *post-processing* of our results, or in other words displaying our results in figures, tables, animations etc. You can see in the *Project Tree* that the Fluent Solver has an inbuilt post-processing features which you can find under **Results** in the *Project Tree*. Although these features are fine to quickly display results some

results while you are testing your set-up, the ease of use and functionalities are rather limited. We will therefore use the external pos-processer called *CFD-Post*.
Therefore, save your project and close the *Fluent Solver*.

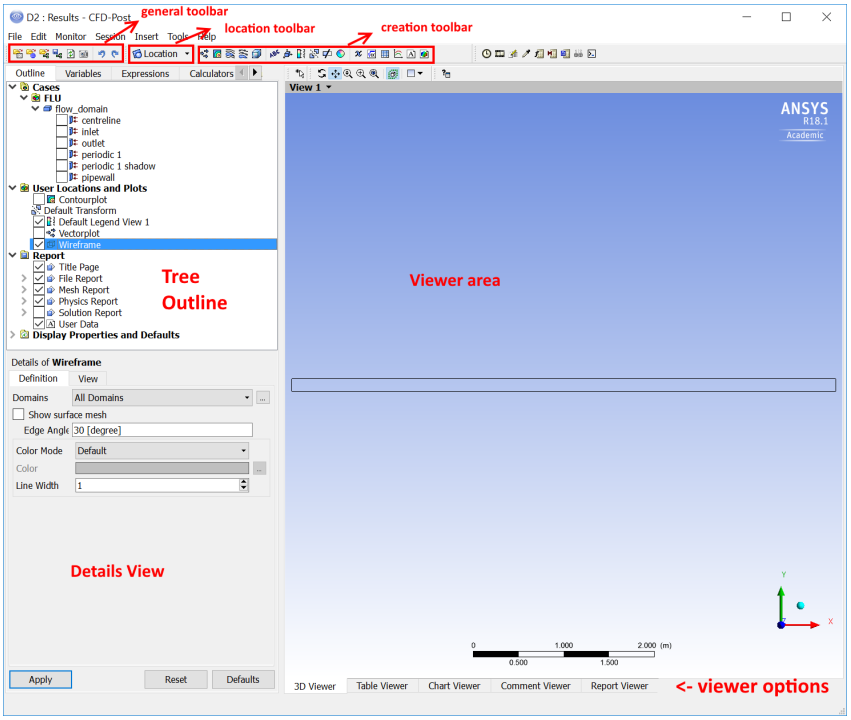
```
   64  2.0918e-05  5.3608e-08  6.9561e-09  8.6336e-02  0:00:02   36
   65  1.7113e-05  3.6552e-08  5.2714e-09  8.6336e-02  0:00:01   35
   66  1.3973e-05  2.4415e-08  3.9509e-09  8.6336e-02  0:00:01   34
iter  continuity  x-velocity  y-velocity  drag-coeff     time/iter
   67  1.1329e-05  1.5966e-08  2.9339e-09  8.6336e-02  0:00:01   33
   68  9.0638e-06  1.0231e-08  2.1594e-09  8.6336e-02  0:00:01   32
   69  7.1517e-06  6.5447e-09  1.5801e-09  8.6336e-02  0:00:01   31
   70  5.5678e-06  4.1932e-09  1.1533e-09  8.6336e-02  0:00:00   30
   71  4.2854e-06  2.6627e-09  8.4451e-10  8.6336e-02  0:00:00   29
   72  3.2645e-06  1.6924e-09  6.2440e-10  8.6336e-02  0:00:00   28
   73  2.4624e-06  1.0888e-09  4.7217e-10  8.6336e-02  0:00:00   27
   74  1.8394e-06  7.1023e-10  3.6162e-10  8.6336e-02  0:00:00   26
   75  1.3612e-06  4.7006e-10  2.7531e-10  8.6336e-02  0:00:00   25
!  76 solution is converged
   76  9.9428e-07  2.9698e-10  2.0777e-10  8.6336e-02  0:00:00   24
```

2.5 Post-processing

In this section we will display to make the the most common type of figures using **CFD-Post**, which are *Vector Maps* (which show magnitude and direction of a variable in a 2D plane), *Contour Maps* (which show magnitude of a variable in a 2D plane) and *LinePlots* (which show the variable as a function of distance).
Drag the **Results** component in the *Component Systems* to the *Project schematic*, and connect the **Solution** cell from the Fluent component table to **Results** cell in the Results component table. Right-click on **Results** and select **update** from the drop-down menu.



Double-click **Results** in the results component box, this will launch *CFD-Post* which is the post-processing program which we will use to plot our results. The interface of *CFD-Post* consists of the *tree outline* and *detailed viewer* on the left, the main window is called the *viewer area*, here you can change the view in the different tabs on the bottom, and the *toolbars* on top, of which the *creation toolbar* contains the main functions for plotting results.

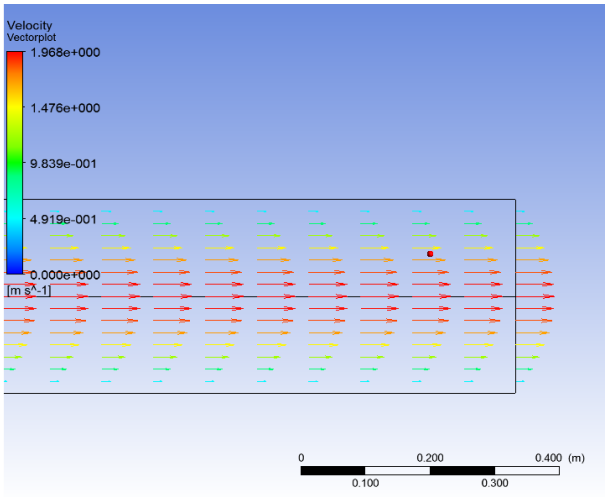


In the following subsection we will give step-by-step procedure to create *velocity vectors*, a *velocity magnitude contour*, *velocity profiles* and to create a plot of the *skin friction coefficient along the pipewall*.

2.5.1 velocity vectors

- click on the Z-axis to view the XY Plane
- click on **periodic 1** under surface body in the *tree outline*. Note that periodic 1 is the default name that CFD-Post gives to the interior of the flow
- click the vector icon in the *creation toolbar*, give a suitable name and click **OK**
- select **periodic 1** under location in the *Details View*
- in the **Symbols** tab change the symbol size to 0.2 (otherwise the vectors will appear too large) and click **Apply**

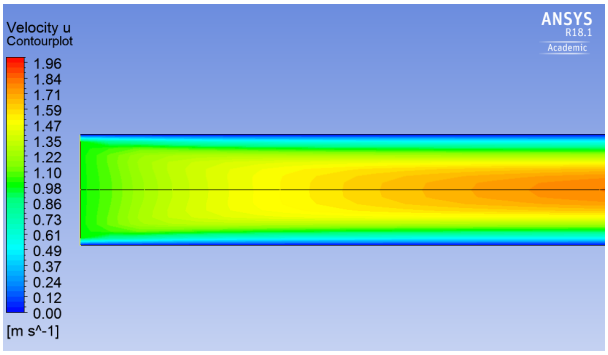
Note that only one half of the pipe cross-sectional area is displayed since we only did simulations for one half of our axisymmetric model. In order to display the bottom half as well double-click **Default Transform** in the *Tree Outline*, untick **Instancing Info from Domain** in the *Details View* and in the bottom (scroll-down) tick **apply reflection** and under Method select the **ZXPlane** and click **Apply**. Your vector plot should now look like below (zoomed in on the inlet). To save a plot press the icon in the general toolbar, set the desired filename and format in the dialog box and press **Save**.



2.5.2 velocity magnitude contours

1. click on the contour icon in the creation toolbar, give an appropriate name and click **OK**
2. in the *Details View* select **periodic 1** under locations
3. change the the Variable from pressure to **velocity u** and click **Apply**
4. to increase the number of contours change **# of Contours** to e.g. 50 and click **Apply** again

The figure below show an example contour plot with 50 contours (zoomed in on the inlet). Note that for this figure the vector plot and WireFrame have been unticked under **User Locations and Plots** in the *Tree Outline*. Note how the velocity is uniform at the pipe inlet (as set by our inlet boundary condition) but that it develops with x-distance showing a high velocity central region in the pipe and low velocity regions close to the pipewall. (Note that you can change the appearance of the legend in the details view of *Default Legend View 1*.)



2.5.3 velocity profiles (velocity as a function of radial position)

Here we are going to plot the velocity profile at the inlet and outlet of our pipe. To make these profiles we first need to specify the lines along which we want to plot our profiles.

1. create a line by clicking Location in the *creation toolbar* and select **Line** and name it **Pipe Inlet**

2. in the *Details View* you can enter the (x,y,z) coordinates for the two points that make up the line, enter (0,0,0) for point 1 and (0,0.15,0) for point 2 and press **Apply**. Set the number of samples to 8, i.e. equal to your mesh size along the inlet and outlet. Note that increasing the amount of samples beyond the mesh size does not lead to more detail in your plots since the results are simply linearly interpolated between the calculation points.

3. repeat the steps above to create the *pipe outlet*, with coordinates (6,0,0) for point 1 and (6,0.15,0) for point 2.

4. check the location of both lines in your view area, you might need to untick the previously generated velocity vectors and contour in the project tree to be able to see them properly.

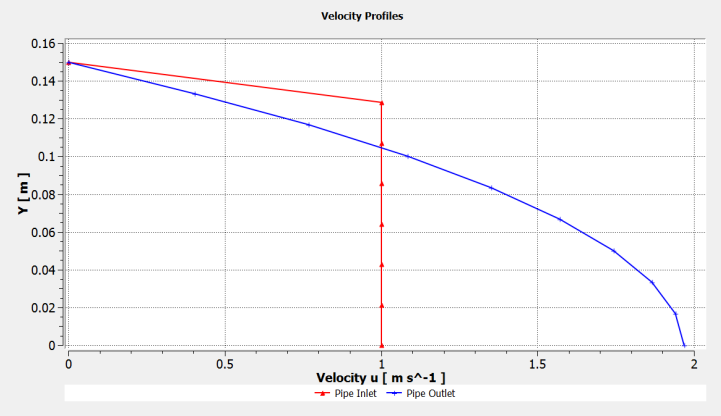
5. select the chart icon from the *creation toolbar*, name it velocity profiles and select **OK**.

6. go to to the **data series** tab in the *Details View* and under location select **pipe inlet**, also under name type pipe inlet.

7. select the **X-axis** tab and select **Velocity u** as variable

8. select the **Y-axis** tab and select **Y** as variable and select **Apply**

Your velocity profile plot should look as below.



The data can be exported to a comma-separated-file (`*.csv`), which can be imported in Excel or Matlab[1] to compare your simulated data to other types of data (e.g. analytical or experimental data). In order to export the data press **Export** in the Details View of your velocity profiles that have just been created (note that the velocity profiles are listed under **Report** in the project tree.

2.5.4 skin friction coefficient along the pipe wall

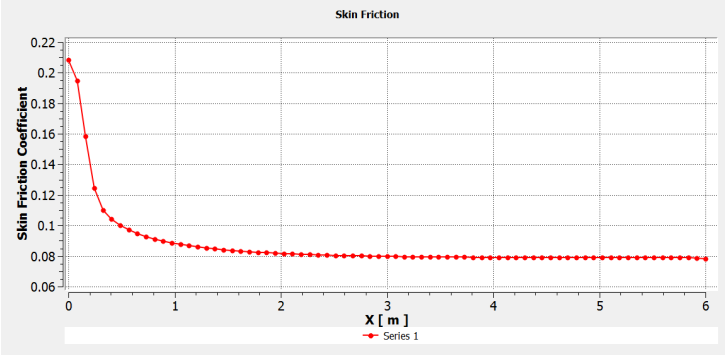
Here we will plot the skin friction coefficient (=normalised wall shear stress) along the pipe wall. The skin friction coefficient in Fluent is (similar to the drag coefficient) calculated with respect to the user-defined reference values for the density and velocity as follows:

$$c_{\mathrm{f}} = \frac{\tau_{\mathrm{w}}}{0.5\rho_{\mathrm{ref}}U_{\mathrm{ref}}^2} \tag{1}$$

We had already set the correct reference values in order to monitor our drag force coefficient and we have already told Fluent to export the skin friction coefficient so we can now plot the skin friction coefficient along the pipe wall as follows:

[1] using the `csvread` command

1. create a line called "pipe wall" starting at (0,0.15) and ending at (6,0.15)
2. select the chart icon from the create toolbar, name it "skin friction coefficient" and press **OK**
3. in the *Details View* select pipe wall under location in the **Data series** tab
4. in the **X-axis** tab select **X** as the variable and under **Y-axis** select **skin friction coefficient** and click **Apply**

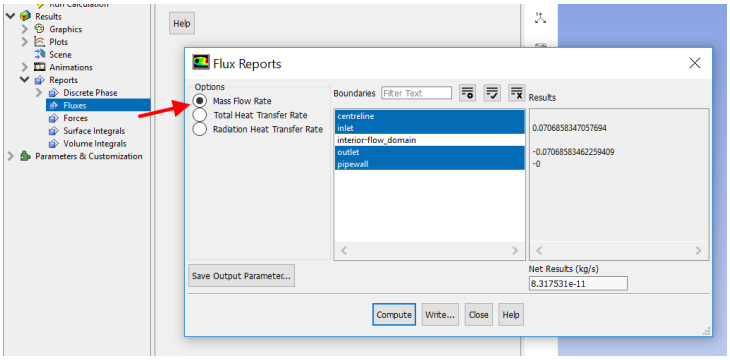


2.6 Verification & Validation

Although we have now produced very nice colorful results it is essential to perform a *verification* and *validation* checks to get more confidence in our results. The verification checks essentially demonstrate whether we have solved our equations correctly, ie whether the numerical procedures are done correctly. However, verification only tells us that we have numerically solved the equations correctly, we could of course still have the wrong set of equations to describe our physical model! Therefore still need to *validate* our model, which can be done by comparing it to analytical or experimental results. Most CFD cases involve complex flow conditions for which analytical results do not exist hence validation is generally performed using experimental data. For our flow problem, laminar flow in a pipe, an exact analytical solution exists so we are going to compare against the analytical solution.

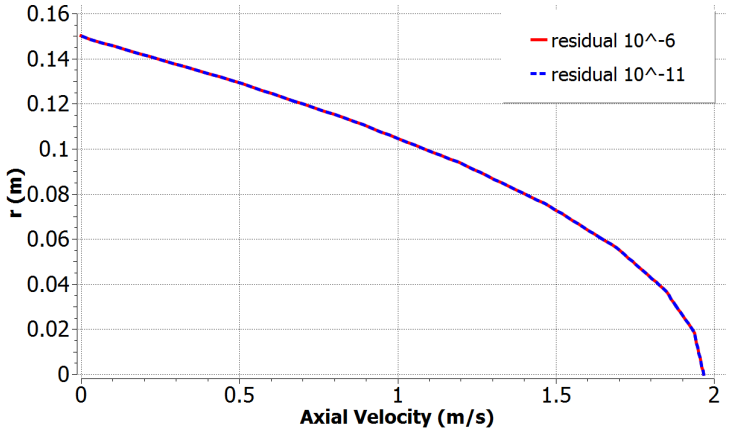
2.7 Verification

- The first (and often overlooked) step involves "common sense" checks, or in other words did we get what we more or less expected to happen. So in this case is the flow from left to right (ans:yes)? Does the velocity profile at the outlet resemble our expectations, i.e. a parabolic shaped profile with larger velocities in the centre of the pipe and smaller towards the wall (ans:yes)? We also need to check whether our boundary conditions are satisfied in our results. In our case the inlet boundary condition was a uniform velocity of 1 m/s and no-slip (u=0 m/s) along the pipewall. We can verify in our results that these boundary conditions satisfied, for example in the velocity contour we made in section 2.5.2.
- Is the conservation of the governing equation satisfied? Or in other words, did we conserve mass (continuity eq), momentum (in x- and y-direction), energy (if we had used the energy equation in a heat transfer problem), etc. In this case we are only going to check whether mass is conserved, since conservation of momentum is a bit more elaborate. Close *CFD-Post* in case you still had it open, then re-open the **Fluent Solver** and run the simulation again. Now in the *Project Tree* go to **Results→Reports** and then double-click on **Fluxes**. Then select the four boundaries of our model "domain" and click **Compute** as follows:

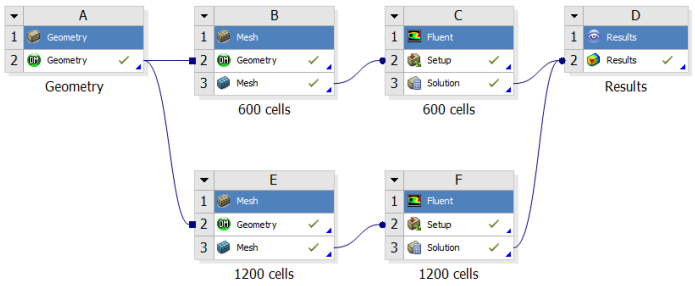


The results window now shows the mass fluxes going across each boundary, with positive values meaning that mass is entering our domain and negative that it is leaving our domain. As expected mass is only entering the inlet and leaving the outlet, there is no mass flux across the pipewall or the centreline. The difference between the mass going in and out of domain is given at the bottom and it shown that this a very small number of $O(10^{-11})$ compared to the flux entering the control volume $O(10^{-2})$. We can therefore conclude that mass is conserved, which gives more confidence that we solved our equations accurately.

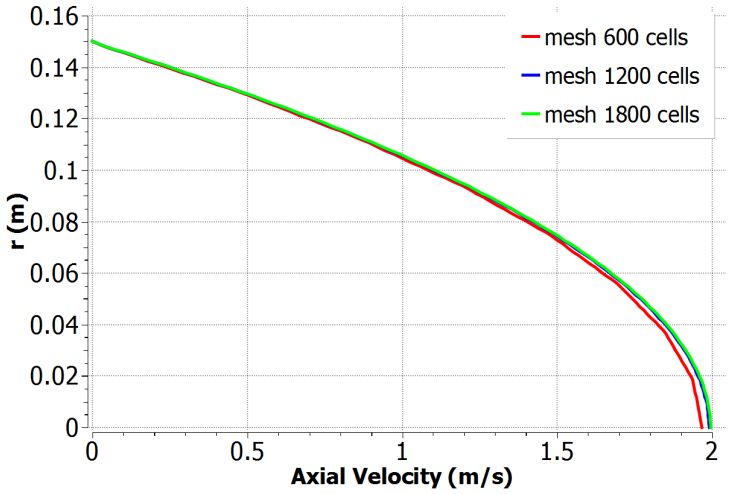
- check linearization error due to number of iterations



- Now we establish that our chosen mesh size is small enough that spatial discretization error has reduced sufficiently such that it does not significantly alter our results. Or in other words, increasing the amount of cells in our domain should not alter our results anymore. We can then say that our solution has become *mesh-independent*.Make a new mesh with double the amount of divisions in the radial direction (16 instead of 8). In order to do this simply make a copy of your mesh in the workbench (right-click the original mesh and select **duplicate**), then drag the original geometry to your new mesh, and open the *Meshing* program to adjust the number of radial divisions and then click **Update** to update your new mesh. Then duplicate the Fluent solver in your workbench and connect this to your new mesh, run the fluent solver again and make sure that the residuals are converged (it will take about 128 iterations, so you need to increase the number of iterations). Then connect the duplicated solution to your previous results component (this will show both solutions in the plots you previously made). Your workbench should look like (note it is good practice to give the components a sensible name):



Repeat the exercise for a mesh of 24 division in radial direction and compare velocity profiles at the outlet in *CFD-Post*:



It is shown that increasing the mesh from 600 cells to 1200 cells has changed the results slightly, in particular the maximum velocity along the centreline has increased. Further increasing the mesh to 1800 cells did not alter the results significantly (green and blue line practically overlap), hence the results have become independent of the mesh size (the spatial discretization error has become so small that we do not notice it). For the remaining calculations we can therefore use the mesh size of 1200 cells. Note that in this case we have checked the mesh sensitivity only for the velocity profile at the outlet, which in our case the main result of interest. Of course, if some other quantity is the variable of primary interest (e.g. the drag force), then typically you would determine the mesh sensitivity based on this variable. For many CFD applications in practice, which may consist of millions of cells, the final mesh choice is often a balanced decision between which error can be accepted and the extra computational time (=cost!) associated with a more accurate mesh.

2.8 Validation (Exercises)

1. Compare your results based on the original mesh and the new mesh with the theoretical velocity profile for fully-developed laminar pipe flow:

$$u(r) = 2U\left(1 - \frac{r^2}{R^2}\right) \tag{2}$$

where r is the radial coordinate, U is the average velocity across the pipe cross-sectional area and R is the pipe radius. You can download a small file with the data here (or we

can ask them to plot it like last year) and save it to your *Homedrive*. The first column in this file is the velocity in m/s and the second column the radial coordinate in m. You can add this data to your plot of the velocity profile in *CFD-Post* as follows: in the *Project Tree* double-click the graph where you previously made your velocity profile, in the *Details View* select the *Data Series* tab and click to a add a new data series. Now under *Data Source* select **File** and then browse for the `*.txt` file which you just saved to your homedrive (in the import dialogue box you may need to change to file type to *text files*), then click **Apply** to load the data. You can change the name of the data-series to something sensible, e.g. *analytical*.
Is our numerical simulation validated?

2. This skin friction they did not understood last year, perhaps remove? Plot the pressure along the axis centreline. Use the following relationship between pressure loss and average velocity to obtain the skin friction coefficient:

$$\Delta P = c_\mathrm{f} \frac{2L\rho U^2}{D} \tag{3}$$

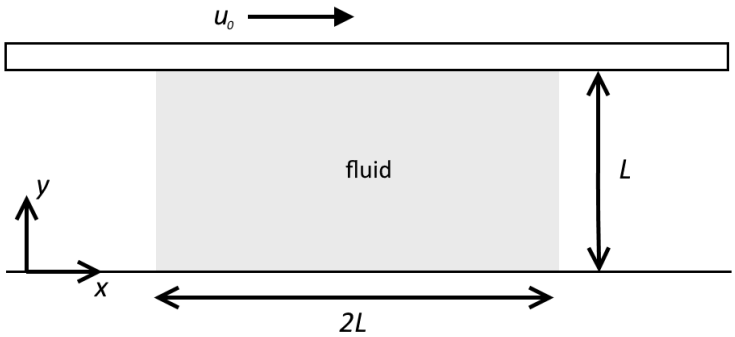
Calculate the skin friction coefficient for the entire pipe length ($x = 0..6\,\mathrm{m}$). Compare the results with the skin friction coefficient for fully developed laminar pipe flow:

$$c_\mathrm{f} = \frac{16}{Re} \tag{4}$$

3. The *entrance region* of the pipe is the region after the inlet where the boundary layer on the side wall is still growing. Once the boundary layer has reached the pipe centreline the velocity profile does not change with x-position anymore, i.e $\partial u/\partial x = 0$. The length of the entrance region, L_e, is often taken as $L_\mathrm{e} \approx 0.06 DRe$. Calculate the skin friction based on the pressure drop in the region ($x = L_\mathrm{e}..6\,\mathrm{m}$), compare your results to the theoretical value for fully developed flow.

3 Start-up flow

In the part we are going to simulate the start-up of a flow between two parallel plates which you have previously considered in tutorial 3 and practical 1a. Initially both plates and the fluid between them are at rest. At time equal zero ($t = 0$) the upper plate starts moving with a constant velocity, u_0, as indicated below. Previously we considered this as 1-dimensional problem by modeling one vertical profile of length L. In Fluent a flow problem need to to have at least 2 dimension therefore we are considering one a "box" of this fluid of dimension $L \times 2L$ as shown below.



Setting up this problem in Fluent is very similar to the laminar pipeflow problem, with the exception we are now dealing with a planar model domain (instead of axisymmetric), it is a

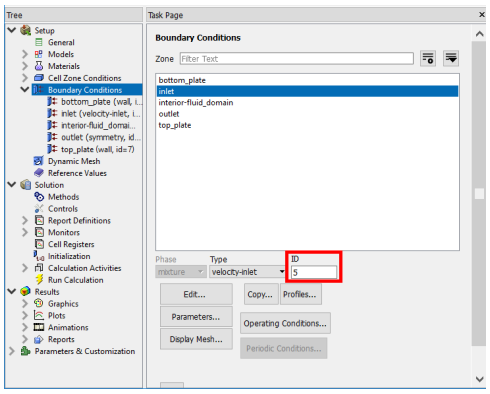
transient simulation (because $\partial../\partial t \neq 0$ in the governing equations), and the inlet and outlet boundary conditions are *periodic* (because we essentially simulate a small element of an infinitely long domain). We are only highlighting the main difference.

1. Create the geometry and mesh for this problem with dimensions as specified in practical 1a. Choose $\Delta x = 1\,\text{mm}$, i.e. one cell in x-direction. Don't forget to name your boundaries in the Meshing.

2. Solver: In the *General* settings in Fluent set the Solver to *Planar* and *Transient*.

3. Fluid: Create a fluid with the specifications that satisfy the required kinematic viscosity.

4. Boundary conditions: set the top and bottom wall boundary conditions as specified in the assignment.
   Click anywhere in the console and hit the enter key on your keyboard, know `>` should have appeared in the console. Type the following in the console `grid/modify-zones/make-periodic` and hit enter. The next line in the console will say `Periodic zone [()]` which is asking you to indicate the Periodic zone. In our case we want our Inlet to be the period zone, you can now type the name of your inlet (which may be *inlet*), or you can use the ID, you can find the ID in the *boundary conditions* task page:



   So in this case I can type either *inlet* (name) or *5* (ID) and then hit enter. The next line will ask for the *shadow zone*, which is the outlet of our domain in this case, so type the appropriate name or ID. In the subsequent lines type `no`, `yes` and `yes`. The console will now indicate that one zone has been deleted and that periodic zones are created. In the boundary conditions *Task Page* you can now see that the inlet boundary condition *type* has changed to *Periodic* and that the outlet boundary condition has disappeared. In the bottom of the boundary conditions *Task Page* you can now see that you can click the *Periodic conditions ...* to set the period boundary condition (which was not selected previously, see figure above). Here we set the periodic boundary condition to specified pressure gradient of $0\,\text{Pa/m}$ (note that we actually have a zero-pressure gradient flow in this case, because the moving top plate is driving the flow, not the pressure!).

5. under *Solution→Methods* tick the *Non-Iterative Time Advancement* box. Leave the *Monitors* at their default values.

6. Initialise your results, and set the Time Step Size to the same timestep used in your Matlab simulation.

3.1 Exercises

1. run a simulation for 0.01 s, 0.1 s and 1 s and compare u_x to your Matlab results obtained in practical 1a. Note that once you have made a graph in *CFD-Post* you can export the data (by clicking Export in the *Details View*) to a comma seperate value file (`*.csv`) or a text file (`*.txt` or `*.dat`) which you can read into matlab (see `csvread` or `dlmread` commands). Alternatively, you can add the Matlab data to a text file and load it into *CFD-Post* following the procedure described in the pipe flow case.

2. any more exercises that we can do? verification exercises? increase mesh and compare? check for mass continuity (=no need because of periodic BC's)

Submit (via MyAberdeen)
For practical 1a (29 Sept) and 1b a *single* document (MsWord document or pdf-file) with the graphs requested, a discussion of the various issues, and answers to all the questions posed. Graphs directly obtained from Fluent need to be exported by clicking the button, do not make direct screen copies of your graphs. Other graphs can be made in Excel, Matlab etc.

Deadline: 11th October 23:59hrs