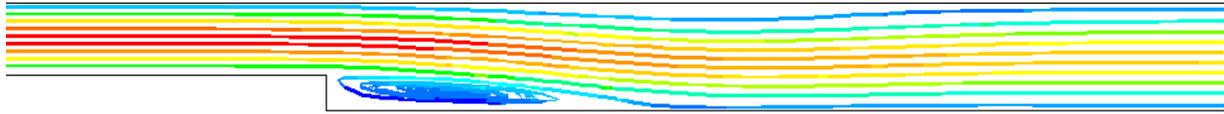


Exercise 2: Laminar flow over a backward-facing step (2D)



Purpose

The aim of the exercise is to simulate a laminar flow over a backward-facing step, give some insight into an influence of the grid density and an order of the spatial discretization on accuracy of the results

Summary

The simulation of the laminar flow through the 2D channel with sudden expansion has to be performed at the Reynolds number $Re_h=230$ (the Reynolds number is based on the step height h and averaged velocity at the inlet, Fig. 1). The computations have to be done on two grids (coarse and fine) using the first- and second-order upwind schemes for discretization of the convective terms in the momentum equations. The numerical results should be compared with provided experimental data. The pressure losses have to be determined.

At the inlet the parabolic velocity profile has to be specified using `DEFINE_PROFILE` function (attached to the instruction). The `DEFINE_PROFILE` has to be compiled in Fluent using 'Interpreted UDF'.

For the final report the following questions have to be answered:

1. What is the cause of pressure losses along the channel length and close to the step?
2. An influence of the order of the discretization scheme on the selected results has to be determined.
3. Is it important to obtain a grid independent solution, or not ?

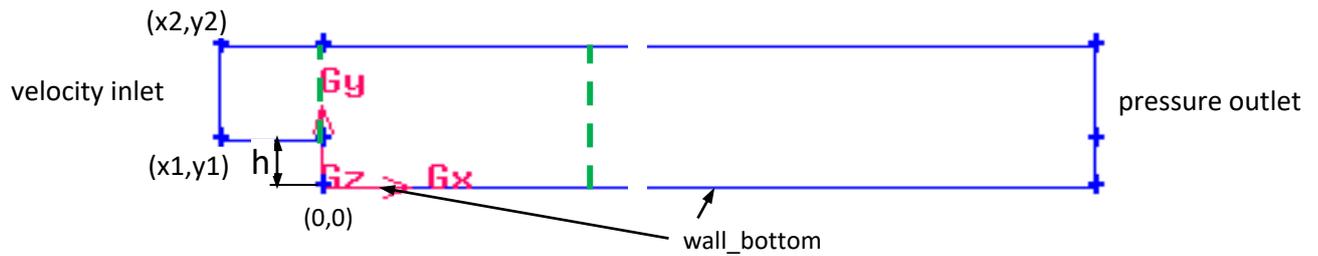


Fig 1. Computational domain and boundary conditions.

Figure 1. Geometry of the computational domain and boundary conditions. The vertical dashed lines (cross sections $x = 0$ and $x = 0.06\text{m}$) show the position at which the numerical results have to be compared with experimental data. Additionally, the x-wall shear stress has to be verified at the bottom wall (determination of the reattachment length).

Fluent program

1. Simulation using 1-st order upwind scheme (basic, coarse mesh)

1. Open the Fluent program. **Double precision solver, 2D, serial**. Read the mesh and display the mesh **Display/Results/Graphics and Animations/Mesh**. The left boundary should appear in blue colour – velocity inlet. The right boundary should be depicted in red – pressure outlet. Walls should be visualized in white.
2. Check a size of the computational domain **General/Mesh/Scale Mesh (Domain Extents – total length 0.70 m height 0.03 m)**. If the domain size is not correct use **Grid Mesh / Scale / Scaling** to rescale the mesh.
3. The flow solver settings:
 - a. **Pressure-based** – recommended for solution of the incompressible flows (**Density-based** recommended for solution of the compressible flows, Mach number > 0.3 – not used now)
 - b. **Steady** – steady flow (**Transient** – means unsteady, time-accurate simulation – not used now)
 - c. Flow is 2-dimensional **2D Space/Planar**
4. Select the appropriate settings in **Models/Viscous**. Note that the Reynolds number is 230.
5. Specify the fluid properties: **Materials**. Select **air** and click **Create/Edit**. Specify the fluid properties like for water: density 1000 kg/m^3 and dynamic viscosity $0.001\text{ kg/(m}\cdot\text{s)}$. (One can add other fluids using **Fluent Database**. But the new fluid has to be later activated using **Cell Zone Conditions/Edit**, confirm **Create/Edit**.)

6. Prepare a text file (with extension *.c) and include (copy and paste) the text fragment shown at the end of the tutorial (section 'Preparation of the UDF'). Specify a proper value of the average velocity 'ver_aver' in the newly created text file. Save the file in the working directory.
7. Read the velocity profile specified in the UDF into the Fluent program. Interpret (compile) the c program: **User-Defined / Functions / Interpreted**.
8. Activate the x-velocity profile at the inlet to the computational domain (use UDF instead of Const), in the **Setup / Boundary Conditions / Edit**. Select a proper Velocity Specification Method.
9. Specify the first order discretization scheme for the convective terms in the pressure-correction equation (continuity) and the momentum equations (upwind 1-st order), **Solution / Methods**.
10. Set the convergence limit to 1e-5 for all equations, **Solution / Monitors / Residual**. The residuals are in fact errors. The final error has to be driven to a sufficiently low level (e.g 1e-5) so we can say that the solution will not change anymore with increasing the number of iterations. The solution of all equations is obtained using the iterative method. The iterative process should be continued until converged solution is obtained (see p. 12 below).
11. Initialize the solution (**Solution / Initialization / Standard Initialization**) selecting the velocity inlet boundary under **Compute from**. Notice that in **Initial Values** the value of X Velocity has been displayed. Compare it with the average velocity specified in the UDF. An error of order 1% is acceptable.
12. Start the calculations **Run Calculation**. Specify an arbitrary number of iterations. Continue the calculations until converged solution is obtained. An information "solution is converged" should appear in the console to ensure that the residuals (errors) already reached the specified error limit. Control the convergence history and save the residuals to file **File / Save Picture**. Once done.
13. Visualize the pressure and velocity contours. Visualize also the velocity vectors and pathlines.
 - a. **Contours** – select an appropriate quantity under **Results/Graphics**. Use **Filled** option to fill the space between isolines using appropriate colour. Do not activate any surfaces under **Surfaces**.
 - b. **Vectors** – One can change **Scale** and **Skip**
 - c. **Pathlines** – select the appropriate surface(s) under **Release from Surfaces**. Now it can be done for the whole interior (2D problem, coarse mesh). One can use **Path Skip** for dense meshes. Set pathlines to **Color by Velocity Magnitude**.
14. Define additional surfaces (line segments) at $x=0\text{m}$, $x=0.06\text{m}$ and $y=0\text{m}$ using **Setting up Domain/Surface/Iso-Surface/Surface of Constant/Mesh/X-Coordinate** option (fig 3). Make the two vertical line segments: $x = 0\text{m}$, $x = 0.06\text{m}$ - specify appropriate values of x under **Iso-Values**. Make the horizontal line segment corresponding to the bottom wall, if necessary. Select the Y-coordinate under **Surface/Iso-Surface/Surface of Constant/Mesh**. Specify $y=0\text{m}$. Show the grid together with the three created line segments (**Mesh/Display**).
15. Compare the x-velocity profiles along : $x = 0\text{m}$ and $x = 0.06\text{m}$ with experimental data (fig. 3) using **Results / Plots / XY Plot**. The experimental data are given in the folder „dane_eksperyment“. Read the experimental profiles using **Load File**.

Think about how the variables should be assigned to the x and y axes (see fig 3). Deselect the **Position on x axis** . Assign the x-velocity component to the horizontal axis and y-distance to the vertical axis. In order to show the results along the y-direction (two vertical lines at $x=0$ and

x=0.06m) specify X = 0 and Y=1 under **Plot Direction**. In order to show the results along the x-direction (bottom wall) specify X = 1 and Y=0 under **Plot Direction**.

One can use „Write to File” option to store the data in the text format for their later comparison with the other results/experiments.

16. Determine the reattachment length (fig. 4) by plotting the x-wall shear stress along the bottom wall. The reattachment length corresponds to the x distance at which the shear stress changes sign from negative to positive. **Results/Plots/XY Plot/Wall Fluxes/X Wall Shear Stress**. Select iso-surface of $y=0$, position on x axis and **Plot Direction** to X=1 Y=0

The experimental results for the reattachment length are given in the file re-vs-reattachment-point_woda_cm.txt (the reattachment length is now only for Re=230).

17. Save the Fluent cas and dat files in the working directory **File/Write/Case&Data**.

II. Simulation using the 2-order upwind scheme

Change the discretization scheme for all equations to the second order upwind **Solution/Methods**. Make the computations, compare the results with experiment and store the numerical results („Write to File”) under **Results/ Plots / XY Plot** and save the new cas and dat files.

III. Simulation on the fine grid using the 1-order upwind scheme

First, verify a size of the computational domain using e.g. **Mesh/Scale Mesh**. Use **Adapt/Mark/Adapt Cells/Region** with property specified coordinates of the rectangle embedding the computational domain **under Input Coordinates** and push the **Adapt** button. Make a denser mesh and visualize the mesh.

Specify the 1-st order discretization scheme. Make the computations, compare the results with experiment and store the numerical results („Write to File”) under **Results / Plots / XYPlot** and save the cas and dat files.

IV. Simulation on the fine grid using the 2-order upwind scheme

Change the discretization scheme to the 2-order upwind scheme. Switch **Pressure-Velocity Coupling\Scheme** to **Coupled**. Make the computations. Compare all the current and previous results

and experimental data altogether at selected distances $x=0$ and $x=0.06\text{m}$. Make the graphical files **(File / Save Picture)** for the final report.

Preparation of the UDF (User Defined Function)

The UDF program is show below. It allows to specify the parabolic velocity profile (fig. 2) at the inlet to the computational domain. One have to provide the averaged velocity under *****Specify Here***** (highlighted in green) and save the file. Use dot as decimal mark.

The averaged velocity vel_aver [m/s] is computed knowing the Reynolds number

$$Re = \frac{u_{aver} h \rho}{\mu} = 230$$

Make a copy of the code given below (using Notatnik). Read and compile this code in Fluent using **User-Defined / Functions / Interpreted** function.

```
#include <udf.h>
DEFINE_PROFILE(xvel, t, position)
{
    face_t f;
    float x[ND_ND];
    float vel,vel_aver,vel_max,y1,y2,y,ycent,h;

    /***** SPECIFY HERE *****/
    h=0.01; /* step height*/
    y1= h;
    y2=y1+2.0*h;
    vel_aver= . . . . . ;
    /***** */
    vel_max=1.5*vel_aver;
    ycent = 0.5*(y1+y2);
    begin_f_loop(f,t)
    {
        F_CENTROID(x,f,t);
        y=x[1];
        vel = vel_max*( 1.0 - SQR((y-ycent)/(y1-ycent)) );
        F_PROFILE(f,t,position)=vel;
    }
    end_f_loop(f,t)
}
```

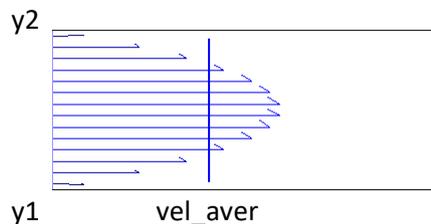


Fig. 2 Velocity profile at the inlet

In general the Reynolds number is defined as: $Re = V \cdot L / \nu = V \cdot L \cdot \rho / \mu$

V – velocity magnitude [m/s]

ν - kinematic fluid viscosity [m²/s]

μ – dynamic fluid viscosity [kg/m-s] ρ - density [kg/m³] L – characteristic length, here h [m]

Description of the experimental data in the folder 'dane_eksperyment' and their graphical presentation

'vx_x0.xy' – x-velocity profile along y distance at $x=0$ m

'vx_x6.xy' – x-velocity profile along y distance at $x=0.06$ m

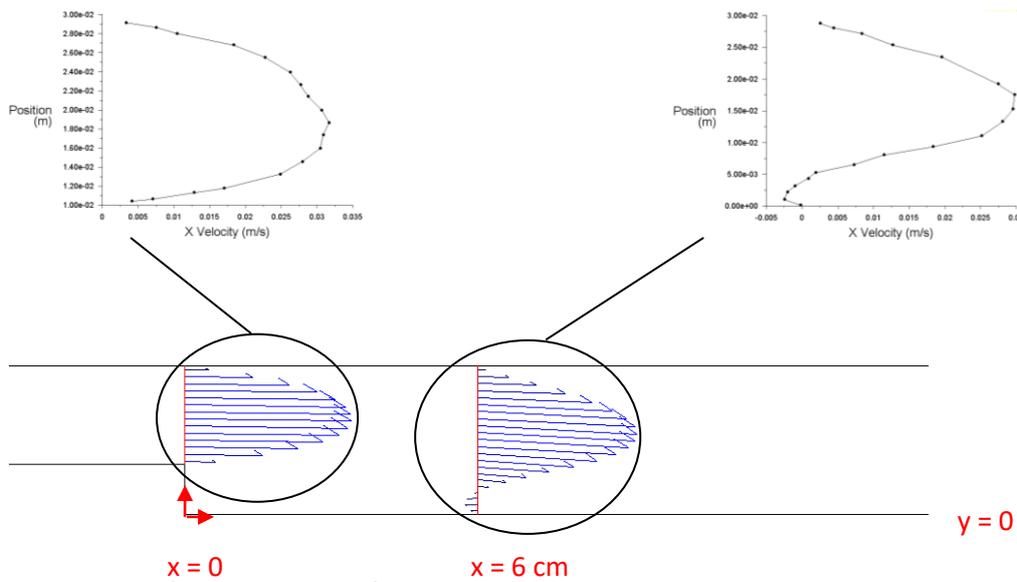


Fig. 3 Cuts at locations $x=0$ and 6.

'reattachment-point-vs-re.dat' – reattachment lengths for different Reynolds numbers

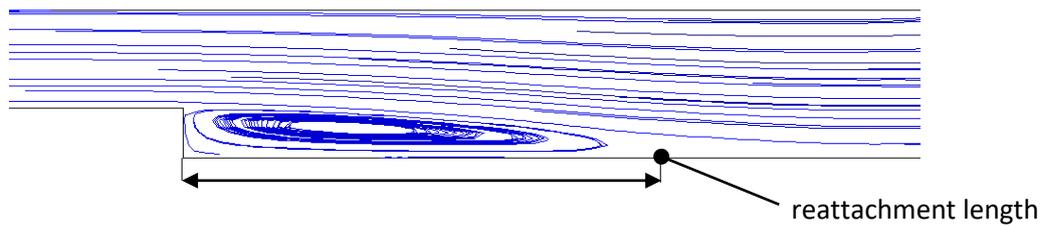


Fig. 4 Streamlines and reattachment point.