Internal flow in a channel with a thin symmetric bump (2D Euler equations).

In this project you are going to write a program to solve the 2D Euler equations using a finite volume method. The application is a representative example of an internal flow configuration. It consists of a channel of height $L = 1$ and length $3L$. The computational domain in x should go from $x = -L$ to $x = 2L$, so that the bump starts at $x = 0$, ends at $x = L$ and has the following height:

$$y_0 = \varepsilon\, x \left(1 - \frac{x}{L}\right),\, 0 \leq x \leq L\,, \tag{1}$$

where $\varepsilon = 0.08$ is a constant. Note that the figure shown below has a different coordinate system, please do not get misled.

We are going to use a simple quasi-uniform grid using N_x grid lines along the streamwise direction, x, and N_y grid lines along the wall-normal direction, y. Set a constant value for $\Delta x = 3L/(N_x - 1)$ and divide each vertical line using uniform steps of size $\Delta y = (L - y_0(x))/(N_y - 1)$, where $y_0(x)$ is the bottom wall equation.

You should treat the west boundary as an inlet, the south and north boundaries as walls and the east boundary as an outlet. At the inlet assume atmospheric conditions with static $p_a = 101300$ Pa, $T_a = 288$ K and axial flow, i.e. vertical velocity component, $v = 0$. The condition to be imposed is that the stagnation pressure and the stagnation temperature are known at the inlet. This is done by specifying a value for the upstream Mach number, M_∞, together with p_a and T_a. Recall that M_∞ is not the Mach number at the inlet plane, M_{inlet}, which is a result of your calculation. At the outlet, you should impose the atmospheric pressure p_a.

Recall that in your code you will need to define the volume of the cells (in 2D the cell areas) and the face normals. For the space discretization use a central scheme (for the fluxes) with an artificial dissipation term as described in the lectures. However, we recommend you to substitute the third derivative by a first derivative for simplicity. For the temporal discretization use a 4-stage low-storage Runge Kutta scheme, also described on the lectures. You should solve the time dependent Euler equations until the steady state is reached. **Further guidance you can find on section 11.5 of Hirsch's book.**

Some quantities of interest are the pressure coefficient along the bottom wall, C_p, the local Mach number over the whole domain (as shown in the figure below) or the entropy distribution. Some examples of the analysis that you can do with your code are the following:

1. Study the compressibility effect by varying $M_\infty = 0.2, 0.4, 0.5, 0.6, 0.7$.
2. Solve the problem with various grid resolutions and evalute the influence of the grid.
3. For a given $M_\infty = 0.65$, study the effect of the bump height by varying $\varepsilon = 0.07, 0.08, 0.09$.
4. You can also try the two sets of Runge-Kutta coefficients, and explain the different convergence properties that you observe.

Please note you are not expected to carry out all the analysis described above. You may select the one(s) you consider more interesting.

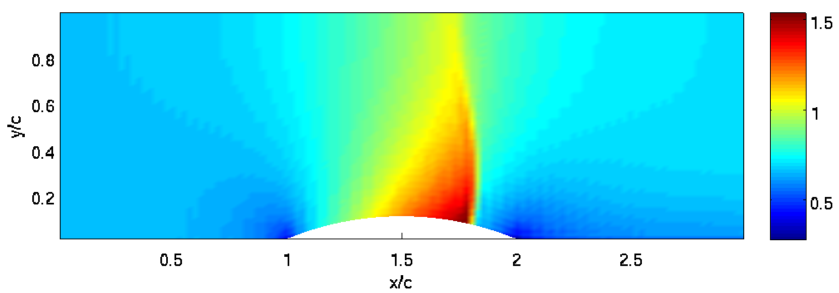

Figure 1: Contours of the local Mach number for a calculation with $p_a = 101300$Pa, $T_a = 288$ K, $M_\infty = 0.7$.

- **Deliverables:** the source code and a short report summarizing your results (maximum 12 pages including figures). In the report you should comment the results and justify your decisions.
- This project is not easy from a programming point of view. Therefore, it is likely that you will not finish the project in a reasonable amount of time. Our recommendation is that you should work on the project for about 30 hours. If after this time, your code is not finished, write a report in which you summarize the status of your code: explain what parts of the code are working and why you are sure that they work, what parts do not work and why you think these parts do not work. If you tried various alternatives to make these parts work, explain this as well. Note that it is very important to show understanding about what you are doing. It is very easy to introduce bugs in your code, so that sometimes, even if you work 100 hours or more, you might end up having an unfinished code. Please follow our recommendation.