Instructions for this assignment *This assignment is meant to be done in groups of three students. The time allowed to hand in this assignment is three weeks: it will be due May 13th 2022 at 6:15pm. We ask for two deliverables: 1) a python code* ***`euler2d.py`*** *contained in one single file with the functionalities described below and 2) a small report of maximum 2 pages (not including the header page, the figures and the bibliography).*

This assignment requires a work quite similar to what was asked in assignment 2. It is to evolve your 2D advection code to a 2D code that essentially solves the following simplified version of the Linearized Euler Equations (LEE) in dimension 2:

$$\frac{\partial \mathbf{q}}{\partial t} + \frac{\partial \mathbf{A}_x \mathbf{q}}{\partial x} + \frac{\partial \mathbf{A}_y \mathbf{q}}{\partial y} = \mathbf{0}. \tag{1}$$

with

$$\mathbf{q} = \begin{bmatrix} \rho_0 u \\ \rho_0 v \\ p \end{bmatrix} \quad , \quad \mathbf{A}_x = \begin{bmatrix} u_0 & 0 & 1 \\ 0 & u_0 & 0 \\ c_0^2 & 0 & u_0 \end{bmatrix} \quad \text{and} \quad \mathbf{A}_y = \begin{bmatrix} v_0 & 0 & 0 \\ 0 & v_0 & 1 \\ 0 & c_0^2 & v_0 \end{bmatrix}.$$

Those equations can be found in a more general form on page 11 of Dr. Toulorge's slides.

In this simplified version, a constant mean flow (u_0, v_0) is considered in the equations to take into account Doppler's effect only (no sources). In what follows, $\rho_0 = 1\,[\mathrm{kg/m^3}]$ (a good approximation of the density of the air) and $c_0 = 340\,[\mathrm{m/s}]$.

Implement a discontinuous Galerkin scheme to solve the LEE, starting either from your code of assignment 3 or the one provided on Moodle. You should implement a numerical flux of your choice, either the true Riemann solver or the Lax-Friedrichs flux. Pages 17 and 18 of Dr. Toulorge's slides contains all necessary informations to build numerical fluxes.

You should implement the following boundary condition:

- Slip-wall boundary condition, proposed on slide 21,
- Simplified non-reflective boundary condition, proposed on slide 22.

One classical test case is to consider an initial Gaussian pressure distribution centered on $(x, y) = (0, 0)$

$$p(x, y, 0) = \epsilon_1 e^{-\alpha_1 (x^2+y^2)\cdot}, \quad u(x, y, 0) = 0 \quad , \quad v(x, y, 0) = 0.$$

The analytical solution can be found in the appendix of [1], equations B8–B11 (use Mach number $M = 0$ in those equations if there is no mean flow so that $\eta^2 = x^2 + y^2$). For example,

$$p(x, y, t) = \frac{\epsilon_1}{2\alpha_1} \int_0^\infty e^{-\xi^2/4\alpha_1} \cos(\xi t) J_0(\xi \eta) \xi d\xi. \tag{2}$$

This analytical solution is (to my best knowledge) impossible to be expressed in a closer form. Thus, this integration must be done numerically. Note that the integrand of (2) decreases rapidly when ξ increases so it is possible to limit the upper bound of the integral.

As a first step, implement the numerical scheme *without a mean flow* and compare your solution to the analytical solution.

In a second time, take into account a *uniform mean flow*, e.g., $(u_0, v_0) = (U_0, 0)$ and show an experiment that highlights the Doppler effect.

Your function `euler2d` should have the following signature:

```
def euler2d(mesh_filename, dt, m, p, u0, v0, p_init, rktype, interactive):
```

where `u0` and `v0` give the constant mean flow, and `p_init` is a function giving the initial pressure distribution. The other arguments are the same as in assignment 3.

References

[1] Tam, C. K., & Webb, J. C. (1993). Dispersion-relation-preserving finite difference schemes for computational acoustics. Journal of computational physics, 107(2), 262-281.