Homework 4
Vorticity-Streamline function method in 2D

In the following problems, several extensions on the MATLAB code implemented in the classroom for the solution of the driven-cavity problem are proposed. The input data for the basic configuration are reported here below:

- domain size: $L_x = L_y = 1\,m$
- north wall velocity: $U_n = 0.1\,m/s$
- kinematic viscosity: $\nu = 0.001\,m^2/s$
- total time of simulation: $t_{tot} = 100\,s$
- number of grid points: 25×25, 51×51, 101×101
- SOR algorithm parameters: $\beta = 1.5$, $max_{it} = 10000$, $max_{error} = 0.0001$

Problem 1: double moving wall

You are requested to modify the MATLAB code developed in the classroom in order to simulate a scenario in both north and south wall are moving parallel to the horizantal axis. The two wall velocities, $U_n$and U_s, can be different. Use the implemented code to explore different combinations of $U_n$and U_s.

Problem 2: modified computational domain

You are requested to modify the MATLAB code implemented in Problem 1 in order to simulate a modified computational domain, according to the figure reported here.

Suggestion: the simplest way to proceed is to still using the complete computational grid (covering the whole rectangular area), but avoiding to solve the equations (Poisson and advection-diffusion equations) in the points belonging to the gray region in the bottom-right corner (i.e. the removed computational domain).

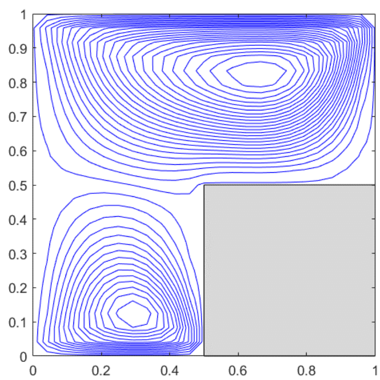

Fig. 1: Modified computational domain for Problem 2

Problem 3: non-uniform grid

You are requested to modify the MATLAB code developed in the classroom to accomodate non-uniform grids in both horizontal and vertical direction.

In case of non uniform grid, the central differencing scheme for the first-order derivative (along the x direction) is:

$$f_x = \frac{\partial f}{\partial x} \approx \frac{f_{i+1} - f_{i-1}}{x_{i+1} - x_{i-1}} \tag{1}$$

For the second-order derivative (along the x direction), the experission is more complex:

$$f_{xx} = \frac{\partial^2 f}{\partial x^2} \approx a_e f_{i+1} - a_p f_i + a_w f_{i-1} \tag{2}$$

where:

$$a_e = \frac{2}{(x_{i+1} - x_{i-1})\,(x_{i+1} - x_i)}$$

$$a_w = \frac{2}{(x_i - x_{i-1})\,(x_{i+1} - x_{i-1})}$$

$$a_p = \frac{2}{(x_i - x_{i-1})\,(x_{i+1} - x_i)}$$

Obviously, similar expressions can be easily derived for derivatives along the y direction.

In particular, it is recommended to test the modified code on the following non-uniform grid, where $\delta_x = 2$ and $\delta_y = 2$ are the so-called *stretching factors*:

$$x_i = \frac{1}{2}\left[1 + \frac{tanh\left(\delta_x\left(\frac{i-1}{N_x-1} - \frac{1}{2}\right)\right)}{tanh\left(\frac{\delta_x}{2}\right)}\right] \quad i = 1, ..., N_x$$

$$y_j = \frac{1}{2}\left[1 + \frac{tanh\left(\delta_y\left(\frac{j-1}{N_y-1} - \frac{1}{2}\right)\right)}{tanh\left(\frac{\delta_y}{2}\right)}\right] \quad j = 1, ..., N_y$$