Continuous Assessment 3

MSc in Computer and Mathematical Engineering

Exercise 1

Generate a triangular mesh within a domain with H shape. The legs of the H must have unit thickness while their separation must be also the unit of length. This H must be fitted within the square domain $[0,3] \times [0,3] \subset \mathbb{R}^2$. Plot the resulting mesh.

Use this mesh to make a 2D interpolation of the function $f(x,y) = sin(xy)$ within the domain. Use different resolution for the mesh and compare the results.

In this exercise we are asked to modify the program for obtaining the L^2 projection of a function $f(x,y) = \sin(xy)$ though a triangular mesh that has an H shape in the square $[0,3] \times [0,3] \subset \mathbb{R}^2$. For this purpose, we will use the codes appearing in the book, which appear in the Ex1.m file.

We first start by defining the triangular mesh. As we are asked to change the mesh to an H-shaped mesh, we need to specify its geometry through a matrix, specifying the segment lines and joining them through the nodes. We do so by specifying a concrete matrix that ensures that the mesh will be H-shaped and all line segments will be connected. By specifying a triangle size h_K for each triangle K, we can obtain the desired triangular mesh through Matlab commands.

The only step left is to find the L_2 projection for $f(x,y)$, and we use the L2Projector2D function for achieving this goal. In this case, we do not need to modify too much, but we need to incorporate the parameter h_K for the title of each graph and modify the plots that are shown. Specifically, we create a 3D surface for visualizing the projection by using the triangular mesh, and another one for visualizing the sum of absolute errors of the projection for each point and triangle in the mesh, which is the L_2 error norm.

In order to assess the different results obtained when specifying a different h_K, which is directly related to the approximation accuracy because it determines the number of triangles and interior nodes, we create a loop for obtaining projections for $h_K = 0.5, 0.25, 0.1, 0.075$. The plot of the triangular mesh, a 2D heat mapping for the different triangles in the H-shaped mesh, the resulting 3D surface and the error for each triangle are shown in Figures 1, 2, 3 and 4.

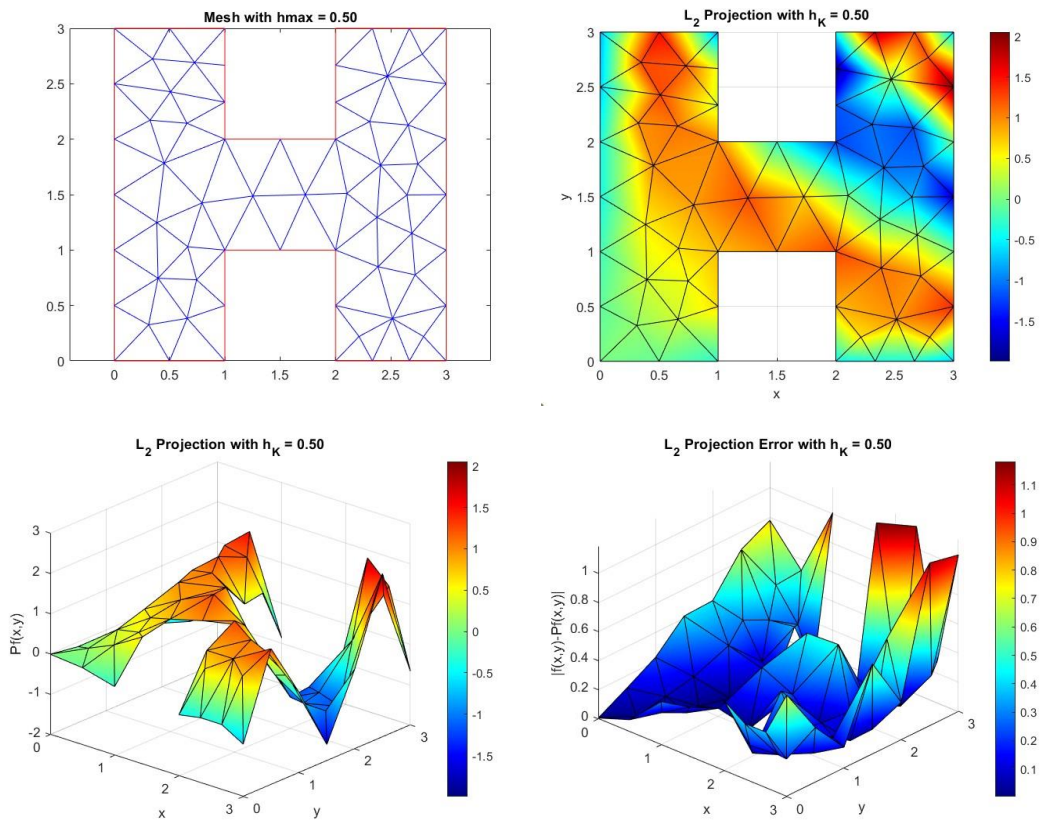

Figure 1. *Surface of the projection made with a triangular mesh with* $h_K = 0.5$

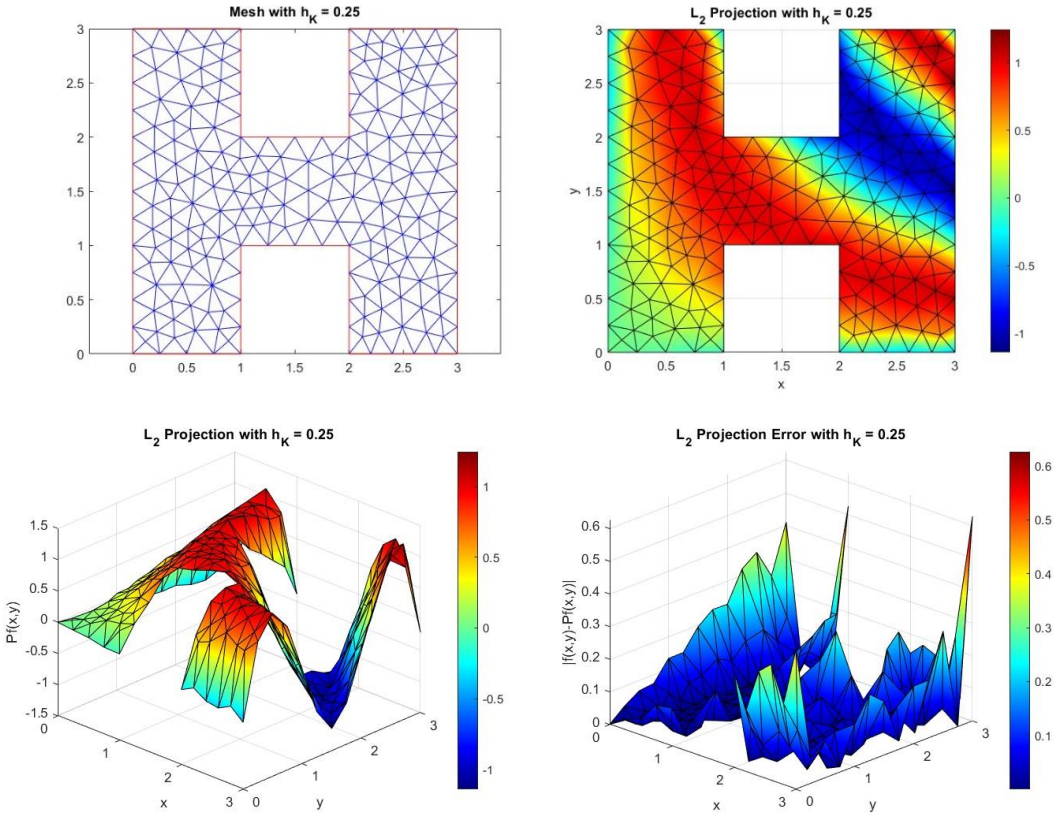

Figure 2. *Surface of the projection made with a triangular mesh with* $h_K = 0.25$

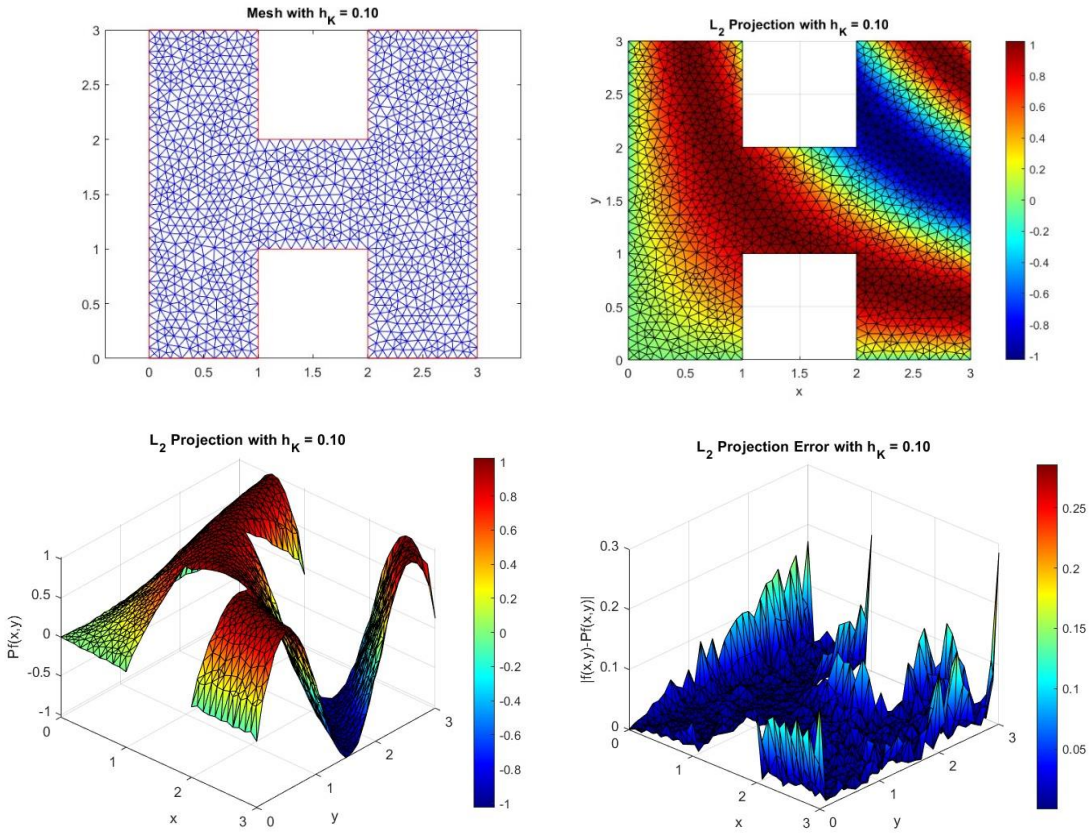

Figure 3. *Surface of the projection made with a triangular mesh with* $h_K = 0.1$

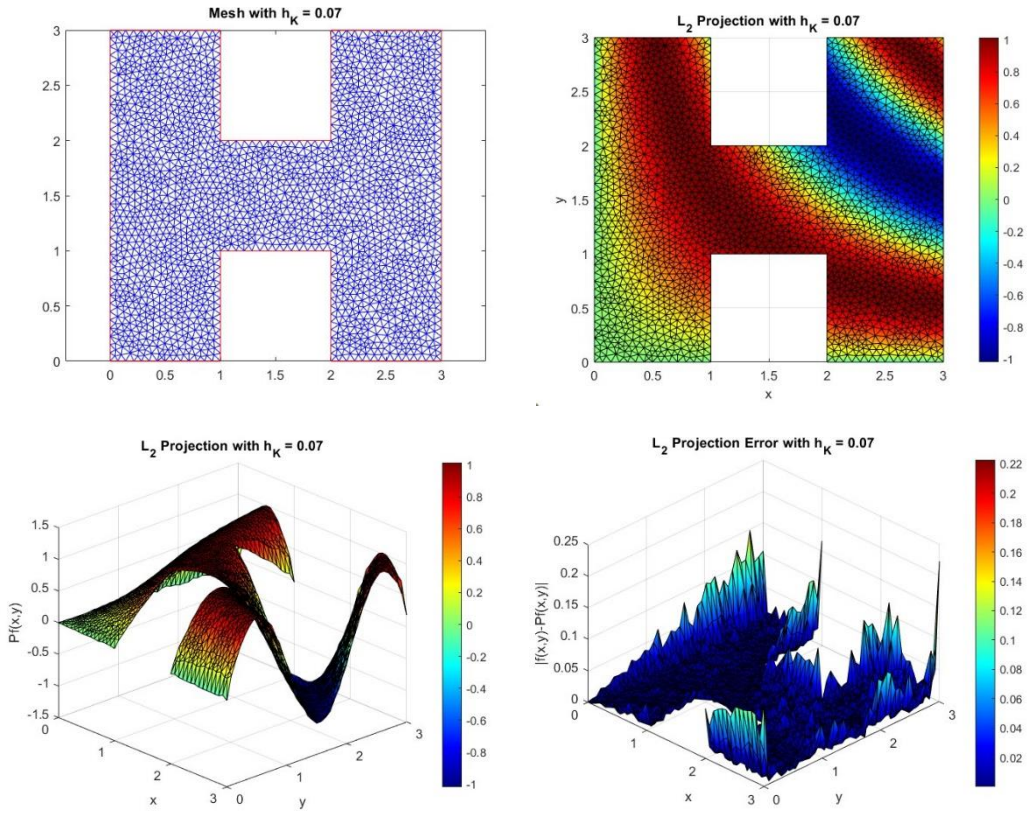

Figure 4. *Surface of the projection made with a triangular mesh with* $h_K = 0.075$

When decreasing the value of h_K, we are increasing the number of triangles in the H-shaped mesh because more triangles are able to fit into it, and hence we have more interior nodes that allow the mesh to better adapt the real surface of the function we want to project. From the graphics, we can observe this is true due to the smoothness of the surface increasing when we consider lower values of h_K.

Because we are increasing the number of interior nodes and we are approximating better, we can see that the error decreases when decreasing h_K by looking at the z axis of the error 3D surface, which covers a lower range of values for each decrease.

Exercise 2

Consider the flux of some fluid within a flat channel with total width of 3 units. Just in the centre of this channel we can find an obstacle with square shape of dimensions 1×1 units. Study the flux of this fluid supposing that the velocity field $[u_x, u_y]$ in each point of the fluid is irrotational, $\nabla \times u = 0$. This is, solve the 2D differential problem of the potential flux:

$$-\Delta\phi = 0, \qquad within\ \Omega$$

with the following boundary conditions

$$n \cdot \nabla\phi = -1 \ \ on\ \Gamma_{in}\ and\ n \cdot \nabla\phi = 1\ on\ \Gamma_{out}$$

$$n \cdot \nabla\phi = 0 \ \ elsewhere$$

Where Γ_{in} represents the boundary where the fluid enters the channel and Γ_{out} the boundary where the fluid leaves the domain. Note that these boundary conditions ensures that all the fluid that enters the channel, leaves the domain and the orthogonal velocities over the boundaries of the domain are null. This implies that the open boundaries are free, then the total length of the channel can be selected freely, but as we want to get good resolution near the object, I suggest to use a total length of 5 units.

We are asked to solve the 2D differential problem for the potential flux of the proposed fluid. For solving it, we need to use the functions proposed in the book, related to solving bidimensional problems like this one. All the code needed for solving this problem appears in the Ex2.m file.

For solving bidimensional problems, we still do a similar approach like the one we used for one-dimensional ones, approximating our answer through solving a linear system which is determined by a stiffness matrix and a load vector, and where we used concepts derived from linear piecewise polynomial approximation (in this case, 2D approximation).

The first thing we do is to specify the geometry of the mesh we will be using. In the problem, we are specified to use a 3×5 rectangular mesh with a 1×1 square right in the middle. Therefore, we modify the geometry matrix $\mathbf{g}$ accordingly and obtain the desired mesh, shown in the subsequent pictures in Figure 5. Clearly, when h_K decreases, the mesh becomes less coarse and hence more triangles can approximate our solution.

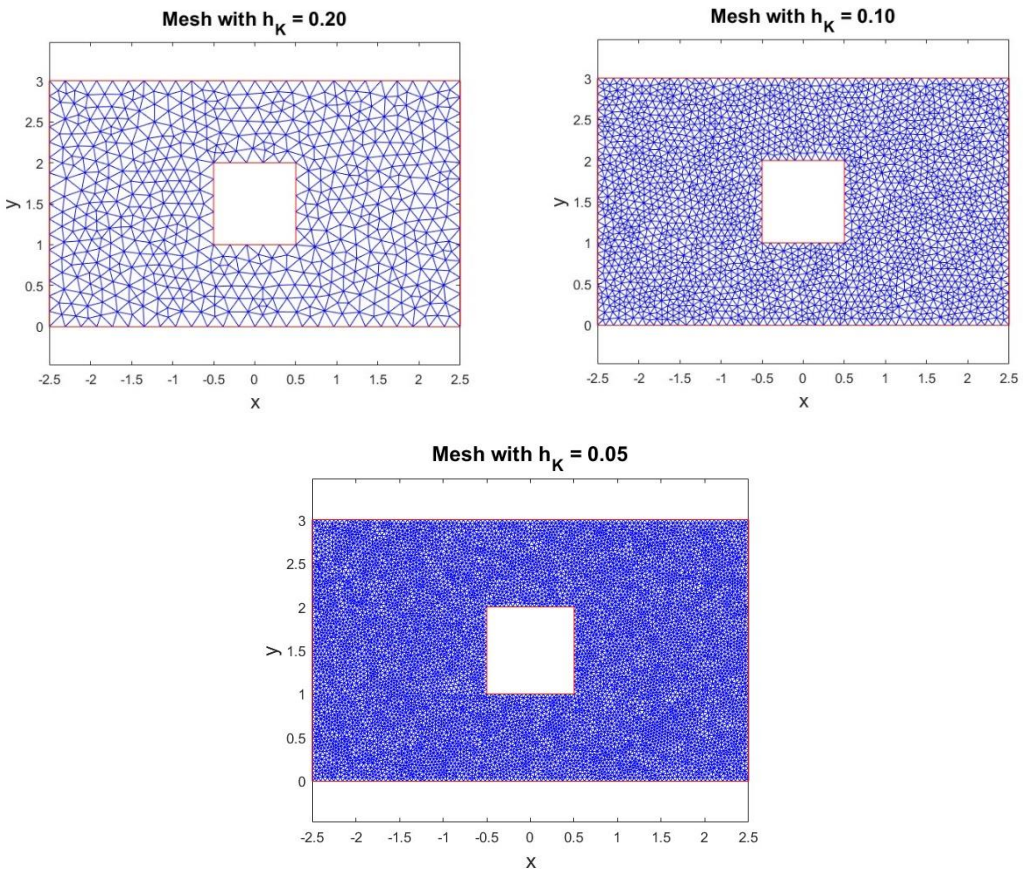

Figure 5. *Meshes for* $h_K = [0.05, 0.1, 0.2]$

As we said, we are using the functions used in the book. In this case, however, we create other auxiliary functions for carrying on our analysis of the resolution. When it comes to the StiffnessAssembler2D, the RobinMassMatrix2D, the RobinLoadVector2D and the RobinAssembler2D functions, these do not change, as we only need to affect the robin conditions specification in order to adapt our problem to that of the example.

In this case, we change the Kappa1, gD1 and gN1 functions that represent the Robin condition parameters. We fix the first two to zero and in the second one, we specify for the values 1 for $x < -2.49$ and -1 for $x > 2.49$. This is because, in the mathematical derivations, we use the negative sign, so that we have to change the sign in our conditions for consistency with the requirement of the fluid moving to the right.

$$-\boldsymbol{n} \cdot \nabla \boldsymbol{\phi} = \kappa(\boldsymbol{\phi} - g_D) - g_N \;\Rightarrow\; \boldsymbol{n} \cdot \nabla \boldsymbol{\phi} = -\kappa(\boldsymbol{\phi} - g_D) + g_N$$

$$\Rightarrow \begin{cases} \boldsymbol{n} \cdot \nabla \boldsymbol{\phi} = 1 \\ \boldsymbol{n} \cdot \nabla \boldsymbol{\phi} = -1 \end{cases} \Rightarrow \begin{cases} -\boldsymbol{n} \cdot \nabla \boldsymbol{\phi} = -1 \\ -\boldsymbol{n} \cdot \nabla \boldsymbol{\phi} = 1 \end{cases}$$

Finally, because we want to study the flux of this fluid, we also constructed functions based on the velocity potential, velocity field and Bernoulli pressure isocontours plots, such that we just need to pass the necessary matrices and vectors in order for them to be computed. Due to no h_K being specified; we use $h_K = [0.05, 0.1, 0.2]$ and repeat the procedure for each of these values. We show the resulting velocity potential plots in

Figure 6, the velocity field plots in Figure 7 and the Bernoulli pressure plots in Figure 8, and we comment the results afterwards.

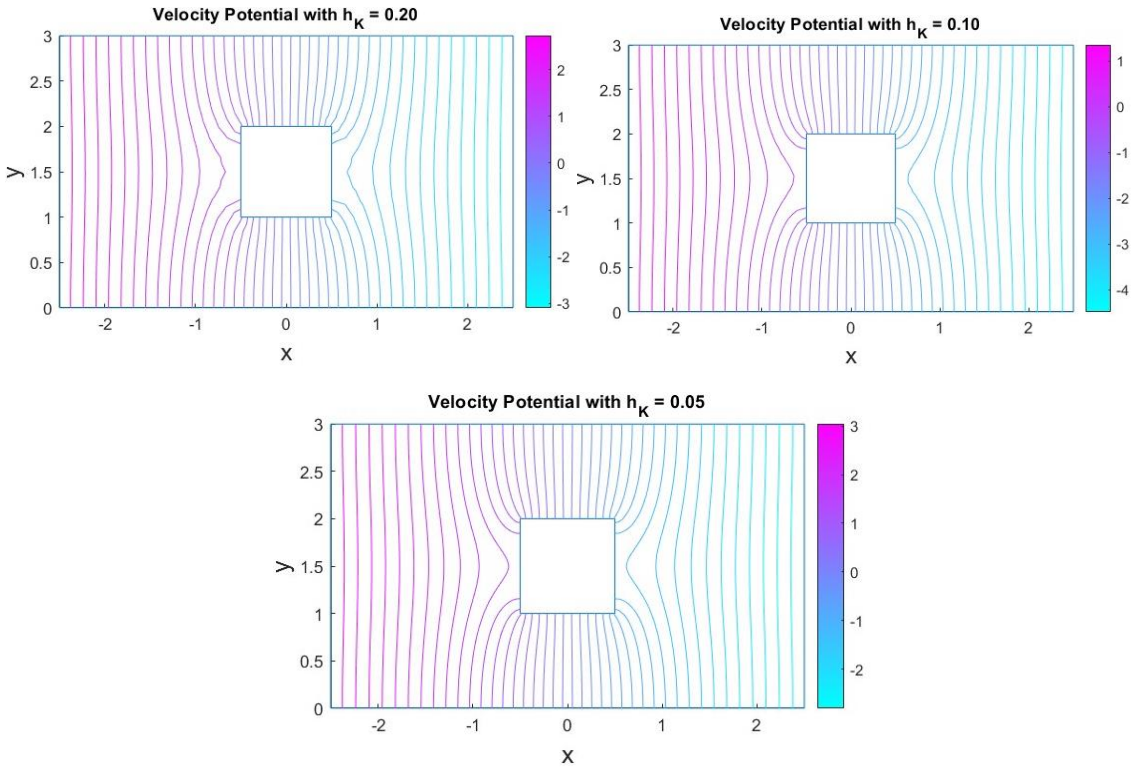

Figure 6. *Velocity potential plots for* $h_K = [0.05, 0.1, 0.2]$

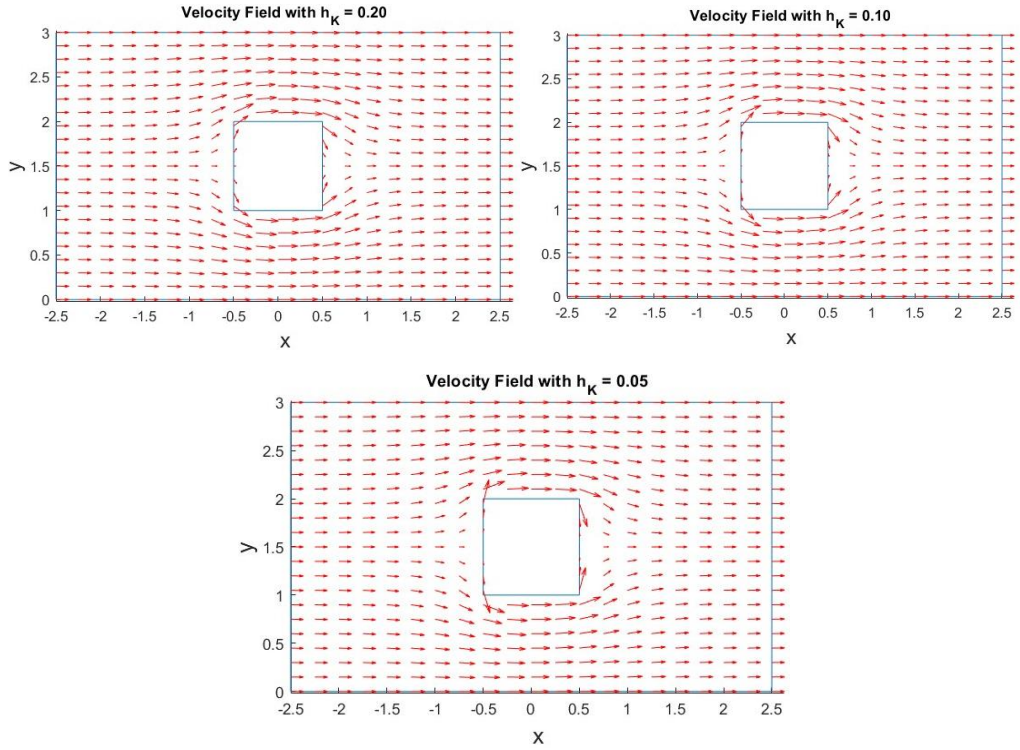

Figure 7. *Velocity filed plots for* $h_K = [0.05, 0.1, 0.2]$

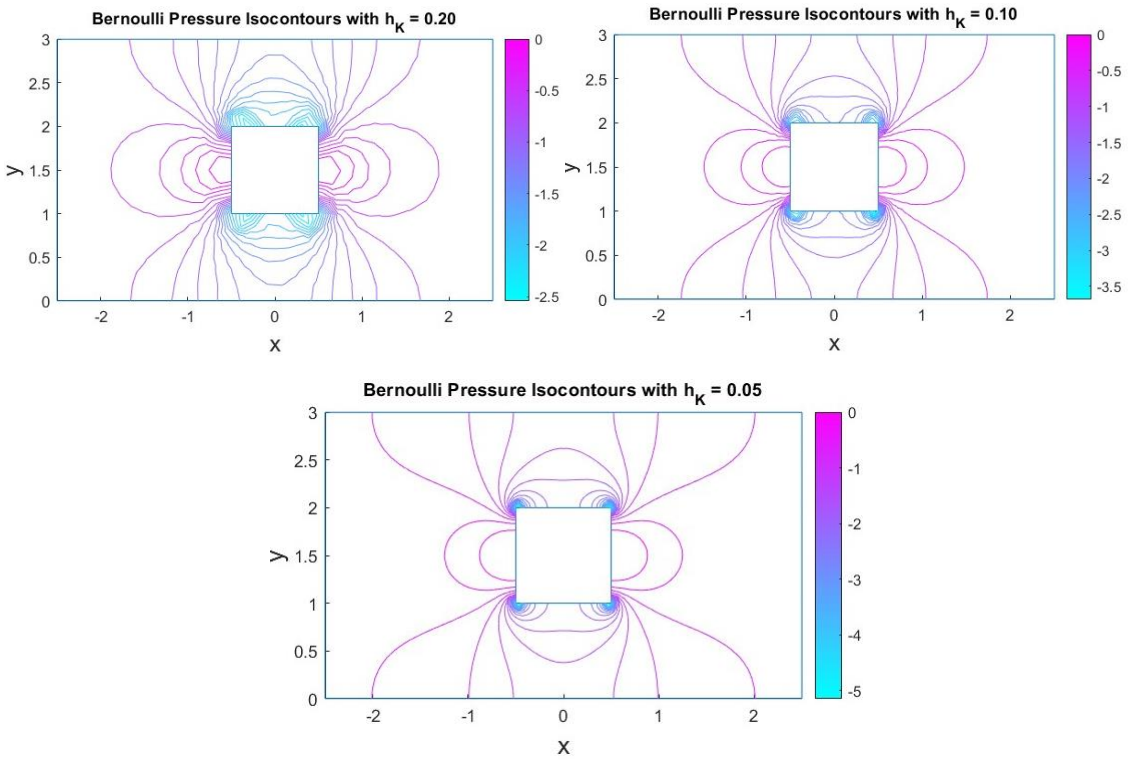

Figure 8. *Bernoulli pressure isocontours plots for* $h_K = [0.05, 0.1, 0.2]$

As we can see, the mesh increases the number of triangles as we decrease h_K, which is obvious. However, this allows to better approximate our solution for this boundary value problem and has visible implications in the resulting plots.

The velocity potential plots show the distribution of velocity potential within the domain, which is plotted in a way one can see both the mesh and the potential. As the size h_K decreases, we can see that the iscontour lines become smoother, which is reasonable given that that a lower h_k increases the number of interior nodes to approximate our solution, making it more precise.

For the velocity field plots, we can see all vectors point to the right direction (due to our change in the boundary conditions), and we can see how higher values of h_K lead to less realistic behaviour of the fluid. We can spot that the vectors near the squared obstacle point to different directions depending on the value of h_K, being those for $h_K = 0.05$ the most realistic ones, as it is intuitive that the fluid will flow to the vertices of the square in order to keep flowing forward. For lower values, we can see that the directions is not vertical enough, which shows how a less coarse mesh makes a better job when approximating the real behaviour of the fluid we are modelling.

Finally, we comment the Bernoulli pressure isocontour results. In this case, we can see that a finer mesh like $h_K = 0.05$ allows to compute more accurately the rapid changes in the velocity and pressure around the square. We can see how the pressure is lower at the corners, because the flow accelerates when intercepting with the square and tries to

flow to the sides. Even though all graphs capture the behaviour on the corners and the surroundings of the square, we can see that by using more triangles we get smoother isocontours.

All in all, the numerical solution obtained for the different h_K values is realistic in the sense that it mimics relatively accurately the beahviour of a real fluid given the results seen in the velocity potential, field and Bernooulli pressure.