

## HOMEWORK 2

1. Consider  $t_{\text{end}} = 10$ , solve the problem, and refine the initial mesh up to 4 times. Verify that the theoretical convergence order holds:

As a first step, it is necessary to introduce the conditions of the problem, such as geometry, PDE and boundary conditions, which are defined in the statement of the problem:

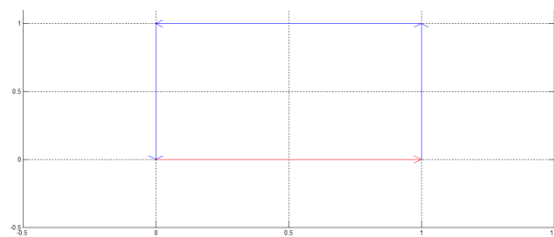
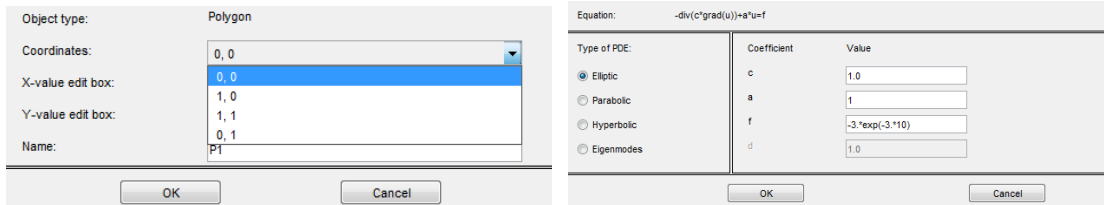


Figure 1 Geometry, Elliptic PDE and boundary condition

After defining the problem it should be defined the desired size of the mesh. I have chosen as the initial length of the element 0.2 and I will refine the initial mesh starting from this value. The size of the element will take the following values: 0.1, 0.05, 0.025.

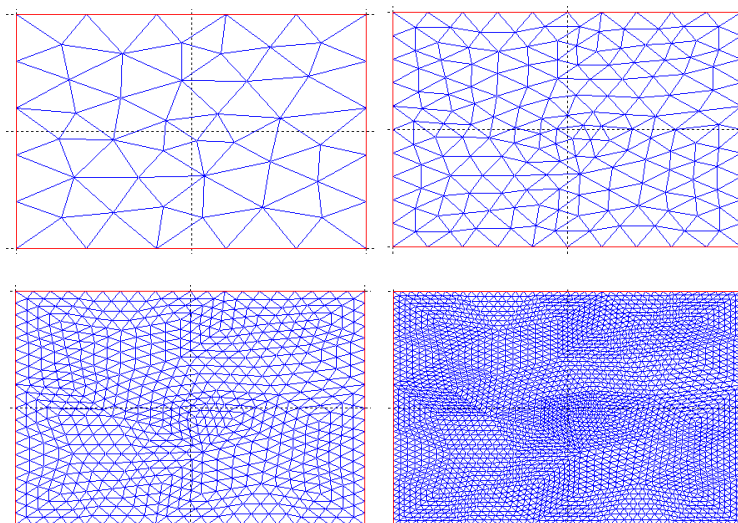
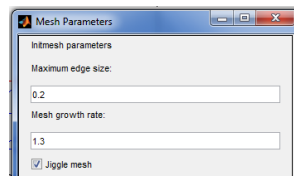
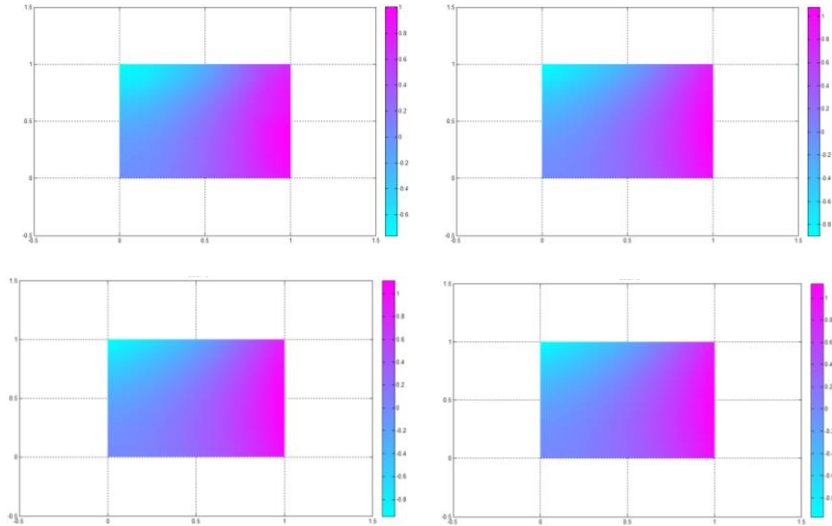


Figure 2 Meshing

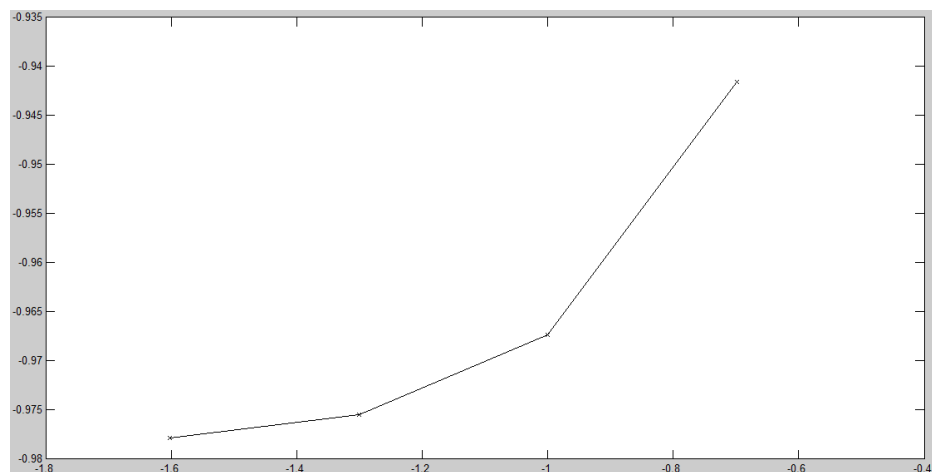
## Computational Mechanics Tools

Once all the variables are defined, the PDE could be solved for each element size executing the Toolbox and the necessary outputs (mesh and solution) have to be exported to the Matlab script to work with them. As it can be seen in the following images, the solution is quite similar but analyzing the numerical data between the roughest mesh and the most detailed mesh the difference is noticeable.



Finally, the error is calculated comparing the solution with the analytical solution. The error decrease when the domain is discretized with smaller elements. This assumption could be seen analyzing the solution of the Node number 5, for example. The variable  $h$  shows different values of the length of the size of the element and the other variable introduces the values of the error at the node 5. The slope is not constant but error decrease when the number of element in the domain increases.

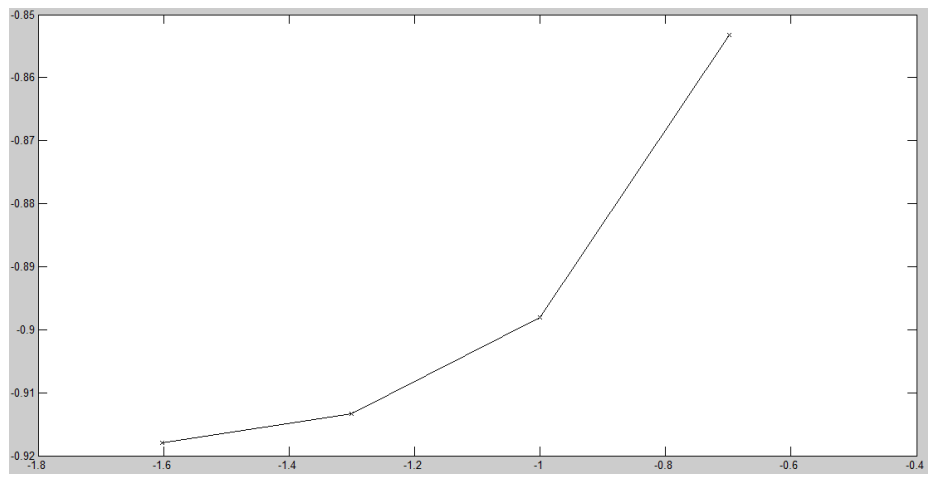
```
% Data
h = [0.2 0.1 0.05 0.025];
err_N5=[0.114404438718293 0.107821845741501 0.105814088664366
0.105227419044675];% Error Node 5
```



The maximum error also meets the assumption:

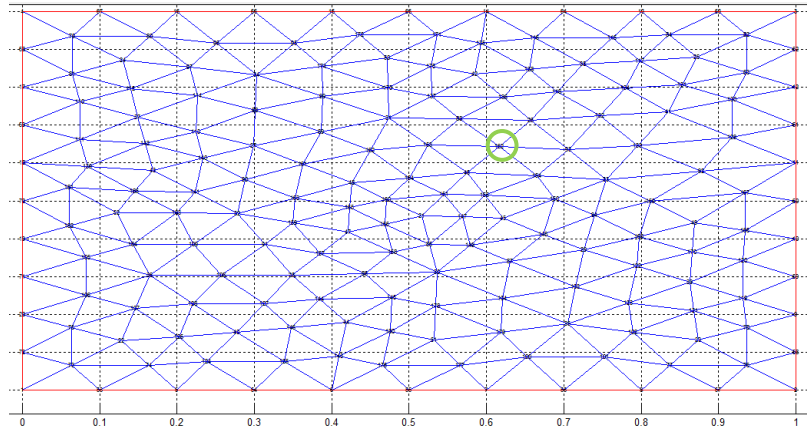
```
% Data
h = [0.2 0.1 0.05 0.025];
err_max=[0.140194119842540 0.127079241220986 0.12223296634107
0.120779286469779];% Error max
```

# Computational Mechanics Tools

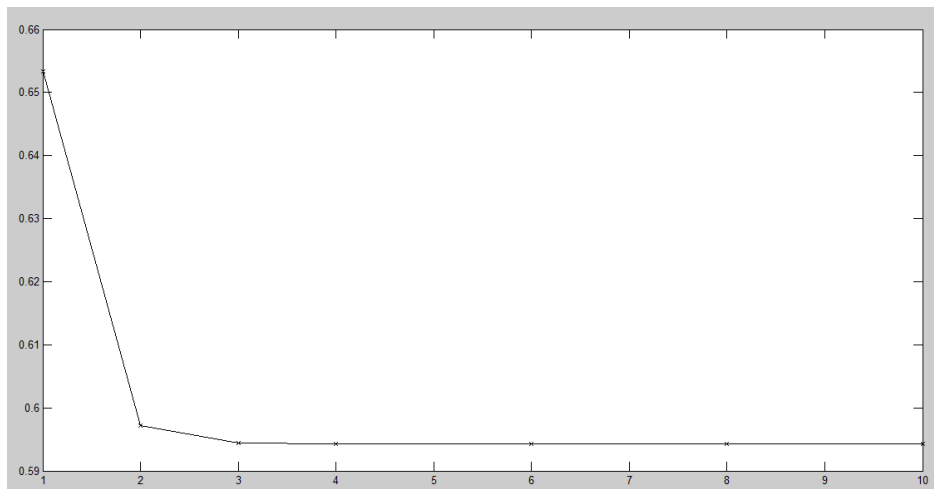


## 2. How is the solution affected when we modify the final time?

The error at the beginning is higher than at the end, due to the initial conditions, and it would decrease over time until stabilizing. In the following example is analyzed the evolution of the error over the time ( $t = 1-10$ ) at a certain node, for example at the node 162 (rounded in green):



```
time=[1 2 3 4 6 8 10];
error=[0.653385645944634 0.597204621361823 0.594407532849941
       0.594268274012969 0.594260995535587 0.594260977494045
       0.594260977449325];
plot(time,error)
```



At the beginning, the error is greater than 0.65 and shortly after ( $t = 3$ ), the error remains practically constant showing values close to 0.59. Hence, the system stabilizes when the time go on.

3. We are interested in obtaining the solution at time  $t_{\text{end}} = 50$ . Find a more efficient manner to solve this problem. You do not need to prove the equivalence mathematically, but you need to provide numerical evidence of the new method.

The value of some terms are depreciable for  $t_{\text{end}} = 50$  because its value is very close to zero. This assumption would facilitate and simplify the problem:

- The source term:

$$f(x, y, t) = -3 e^{-3 \cdot 50} \approx 0 \rightarrow u_t - \Delta u = 0$$

- Boundary condition:

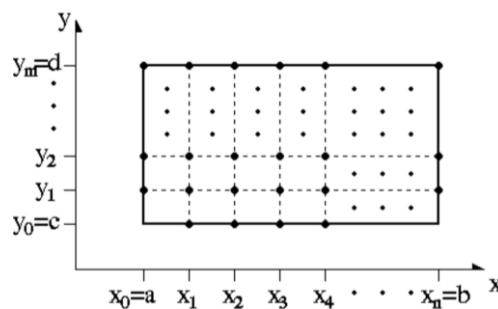
$$\begin{aligned} u_n(x = 0, y, t = 50) &= -y \\ u_n(x = 1, y, t = 50) &= 2 + y \\ e^{-3 \cdot 50} \approx 0 \rightarrow u(x, y = 0, t = 50) &= x^2 \\ u_n(x, y = 1, t = 50) &= x - 2 \end{aligned}$$

- Analytical solution: The solution is a function of two variables (x and y).

$$e^{-3 \cdot 50} \approx 0 \rightarrow u(x, y) = x^2 + xy - y^2$$

The partial differential equations of the Poisson form could be solved using an iterative technique. The Poisson's Equation is,

$$\frac{d^2 u}{dx^2} + \frac{d^2 u}{dy^2} = u_t(x, y)$$



The approximation of the second derivatives are:

$$\frac{d^2 u}{dx^2}(x_i, y_j) = \frac{u(x_{i+1}, y_j) - 2u(x_i, y_j) + u(x_{i-1}, y_j))}{h^2} + O(h^2)$$

$$\frac{d^2 u}{dy^2}(x_i, y_j) = \frac{u(x_i, y_{j+1}) - 2u(x_i, y_j) + u(x_i, y_{j-1}))}{k^2} + O(k^2)$$

$$h = \frac{b - a}{n} \quad n: \text{number of elements}$$

$$k = \frac{d - c}{n}$$

The value of the variables  $O(h^2)$  and  $O(k^2)$  are depreciable and a small error is assumed. Therefore, the Poisson's equation becomes,

## Computational Mechanics Tools

$$\frac{u(x_{i+1}, y_j) - 2u(x_i, y_j) + u(x_{i-1}, y_j))}{h^2} + \frac{u(x_i, y_{j+1}) - 2u(x_i, y_j) + u(x_i, y_{j-1}))}{k^2} = u(x, y)$$

The iterative solution is obtained solving the PDE together with the boundary and initial conditions.