Assignment 1 Computational Mechanics Tools, PDE-Toolbox

Jose Raul Bravo Martinez, MSc Computational Mechanics

November 27, 2018

Solve the following problem with the MATLAB PDE Toolbox:

$$\frac{\partial u}{\partial t} - \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) = f \quad in \;\; \Omega = [0,1]^2,$$

where the source term is given by,

$$f(x,y,t) = -3e^{-3t}$$

We consider an initial condition at t = 0:

$$u(x,y,t=0) = x^2 + xy - y^2 + 1$$

and the following boundary conditions

$$u_n(x=0,y,t) = -y$$
$$u_n(x=1,y,t) = 2+y$$
$$u(x,y=0,t) = x^2 + 3e^{-3t}$$
$$u_n(x,y=1,t) = x-2$$

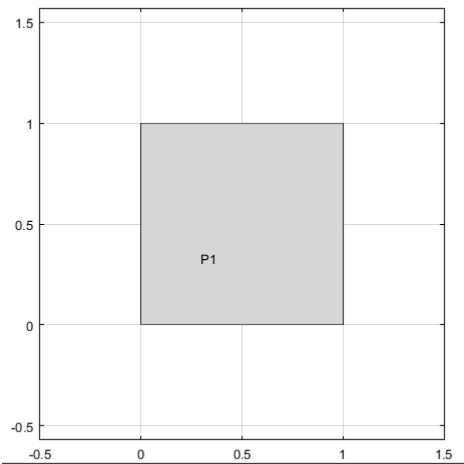
Where $u_n \equiv \partial u/\partial n$.
The analytical solution is given by the following expression,

$$u(x,y,t) = x^2 + xy - y^2 + e^{-3t}$$

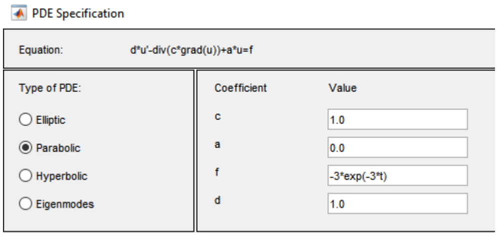
Then,

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

Using the GUI of Matlab to solve PDEs, "PDE Tool", the domain of 1 by one units was created.



On this domain, the PDE specification was set as follows:



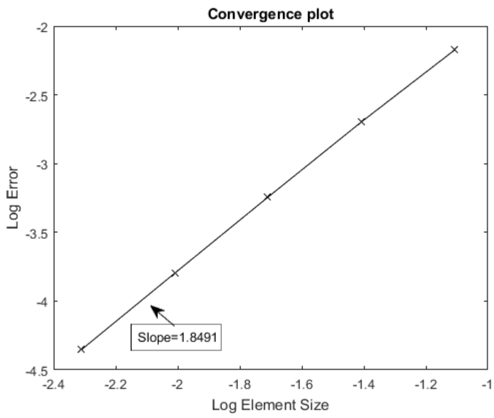
Then, the Neumann and Dirichlet BC were assigned to the corresponding edges and the initial conditions were selected. Since the area of the domain is of 1 unit squared. It is possible to get an estimate for the element size h, which will be used in the convergence plot.

$$Area = N * \frac{h^2}{2}$$

Where N=number of triangles
then,

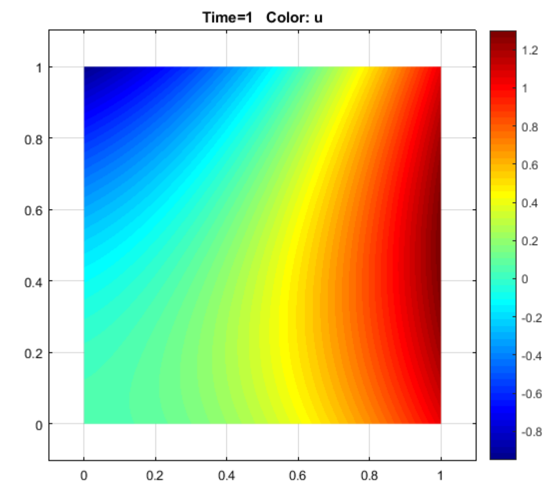
$$h = \sqrt{\frac{2(1)}{N}}$$

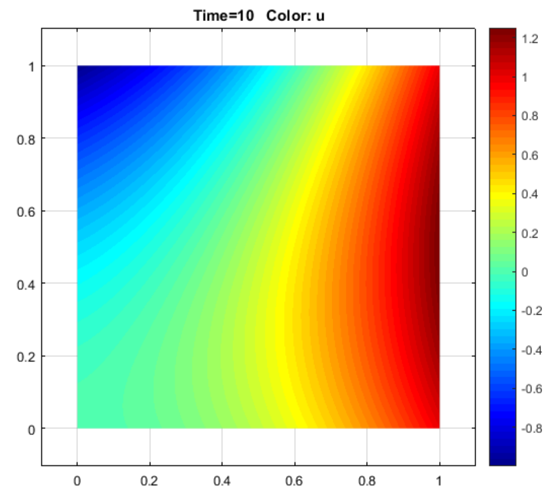
After running the example with different levels of refinement, the following plot was obtained comparing the error and element size.



The theoretical convergence slope for linear elements is of two, the slope obtained is close to the theoretical objective at a value of 1.8491 for the last two points.

2. How is the solution affected when we modify the final time?
In order to analyze the time influence on the solution, let us first consider the solution profile for two different time steps, namely after 1 second, and after 10 seconds.

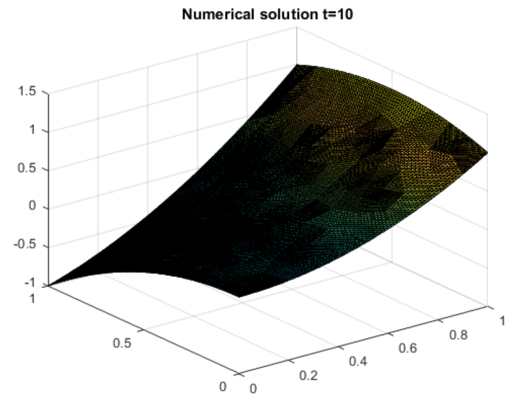


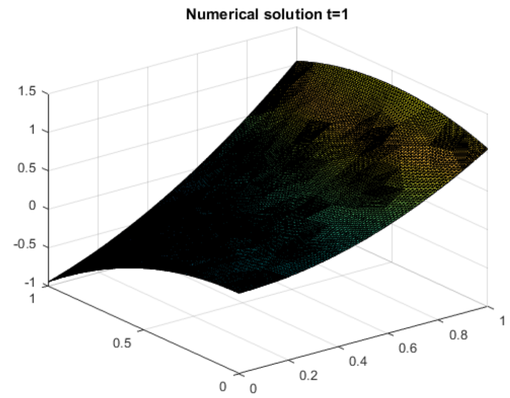


It can be seen that the profile did not change dramatically over this period of time. If one pays attention to the bar next to the plot, it can be seen that the values are a little bit shifted down after 10 seconds. That is, the solution in general moves down preserving the same shape.

In order to visualize it from another angle, one can for example use the provided function errorExample1.m to plot the profile of the solution at t=1 and t=10.

The comparison of the plots is shown next:





The same patter seems to appear, that the solution remains with a similar profile and it just moves down. In order to prove this, the minimum and maximum points are compared for both time steps. The result is shown in the following table:

	After 1 second	After 10 seconds	Difference
Maximum	1.2998	1.2500	0.0498
Minimum	-0.9502	-0.9998	0.0496

Table 1: Comparison of numerical solutions after 1 and 10 seconds

As can be seen, the difference is of approximately 0.05 for both, minimum and maximum cases. Which is of the order of e^{-3}.

3.– We are interested in obtaining the solution at time $t_{end} = 50$. 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.

If one analyses the terms involving time in the equations. Namely:

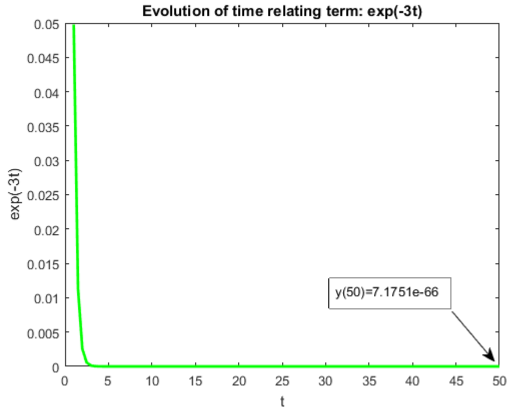
$$Source\ Term : -3e^{-3t} \tag{1}$$

$$Dirichlet\ BC : x^2 + e^{-3t}$$

$$Analytical\ Solution : x^2 + xy - y^2 + e^{-3t}$$

They all contain the term e^{-3t}.

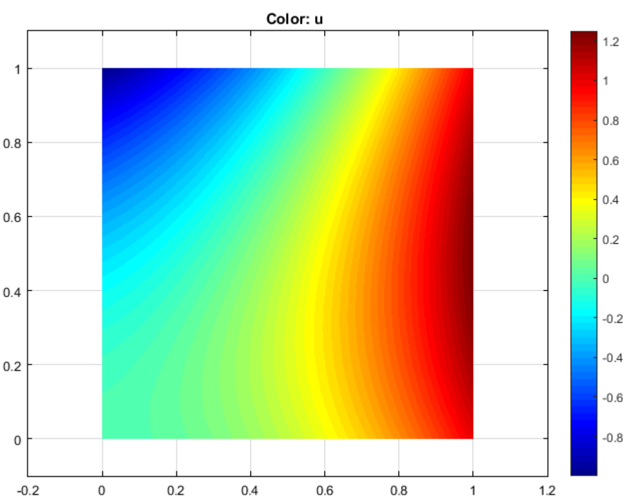
Let us plot the function $y = e^{-3t}$ to see how it evolves.



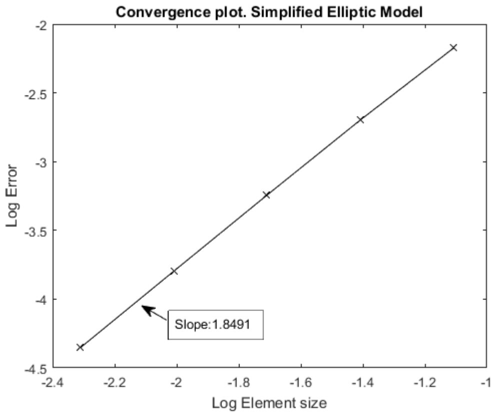
From this, it results clear that one can neglect the time relating term in the expressions, and only keep the spatial ones. This results on an elliptic problem.

Once again, using PDE Tool from Matlab, one can state the domain and boundary conditions; but now neglecting the term time. The main change of couse is the type of PDE to solve, which is now Elliptic.

Initial conditions are not relevant, are not to be given, and the solution takes only one step. Next is the plot of the profile obtained.

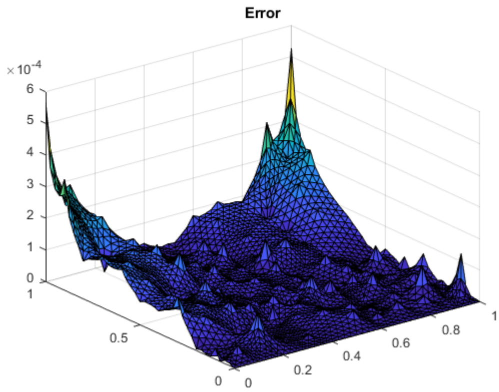


The convergence plot for the elliptic problem is shown next:



As can be observed, the solution convergence rate is exactly the same as it was for the Parabolic model.

One can also plot the error using the function errorExample1.m. The result is shown next:



For this elliptic case, no differences appear in the error up to the 14th decimal place, whether the time term is included or not in the analytical solution to make the comparison against the numerical one. Therefore, one can assume that this implementation is equivalent, and far less costly, as one has to solve only once the equations; compared to thousands of times using the Parabolic formulation.