Computational Mechanics Tools
Homework 3 - PDE-Toolbox

Mariano Tomás Fernandez
m.tomasfernandez@gmail.com

Universitat Politècnica de Catalunya — December 13^{th}, 2019

Abstract

In this assignment PDE-Toolbox, a MATLAB tool, was used to solve a *parabolic* partial differential equation. The theoretical convergence was proved with four steps of mesh-refinement. The time-dependence of the PDE was evaluated for different values of t_{end} concluding that the dependence vanishes fastly. Finally, an *elliptic* partial differential equation (no time dependence) was implemented and the error was below $4.0 \cdot 10^{-12}$ when compared to the *parabolic* partial differential equation at $t_{end} = 50$, in agreement with the previous statement.

Problem description

Solve the following problem with the MATLAB PDE Toolbox:

$$u_t - \Delta u = f \quad \text{in } [0, 1]^2,$$

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:

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

where $u_n = \partial u/\partial n$ The analytical solution of this problem is given by the following expression,

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

1. Consider $t_{end} = 10$, solve the problem, and refine the initial mesh up 4 times. Verify that the theoretical convergence holds.
2. How is the solution affected when we modify the final time?
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.

PDE-Toolbox

The partial differential equation (PDE) problem studied is a one-by-one square, which therefore haves an area of one square unit. The problem is solved with the PDE-Toolbox for MATLAB. The PDE-Toolbox module ask for some definitions prior to solving the problem:

1. geometry of interest;
2. type of PDE to solve:
    - elliptic;
    - parabolic;
    - hyperbolic;
3. boundary conditions of the problem for every boundary of the problem of interest;
4. specific properties for the resolution of the PDE being solved (time-step; initial conditions; ending time; among others).

When the previous definitions are settled, the problem is meshed and solved. Geometry and boundary conditions were taken from the description of the assignment. The PDE to solve is a parabolic equation, due to the presence of the $\partial u/\partial t$. Once the parabolic PDE was adopted, the *Parameters* module in the *Solve* menu were setted to match the given conditions for $u(t = 0)$, and $t_{end} = 10$.

1 Theoretical convergence

Once the problem was correctly characterized and solved, four refinements were implemented in the domain (Ω) of the PDE. The obtained results were compared to those obtained by computing the analytical solution of the PDE. The error is defined as the difference between the analytical and the numerical solutions. The values of h were calculated as $h = \sqrt{2 \cdot A/n}$, where n is the total amount of elements in the domain Ω and A is the area of the domain Ω , which in our case is $A = 1$. The theoretical convergence is analysing the error with increasing refinement of the mesh for the numerical solution (reducing h). The obtained results presented a proper convergence of the problem for smaller values of h, see Figure 1. The average slope passing through all the points $(h_i, error_i)$ is $m = 1.83$.

2 Variation of the solution with different end times

In order to analyse the time dependence of the $u(x, y, t)$ function, different t_{end} were used to compute the values of u. Maximum of u (Max(u)) and average of u (Average(u)) values were plotted against logarithm of time, shown in Figure 2. Maximum(u) gets the maximum value of $u(x, y, t_{end})$ considering the results of u_e in each element of the domain at $t = t_{end}$ and average(u) is the sum of all the values of u_e in the domain at $t = t_{end}$ divided by the total number of elements n. The values used for t_{end} are $[1, 2, 3, 4, 5, 6, 7, 10, 25, 50, 90]$. The behaviour for Max($u$) and Average($u$) are similar and appears to be almost constant for $t_{end} \geq 4$. This behaviour is related to the fact that boundary conditions and source terms depend on $\alpha e^{-3t_{end}}$ and using $t \geq 4$ these values are 10^{-6} or less. More detailed information about the graph is presented the Appendix.

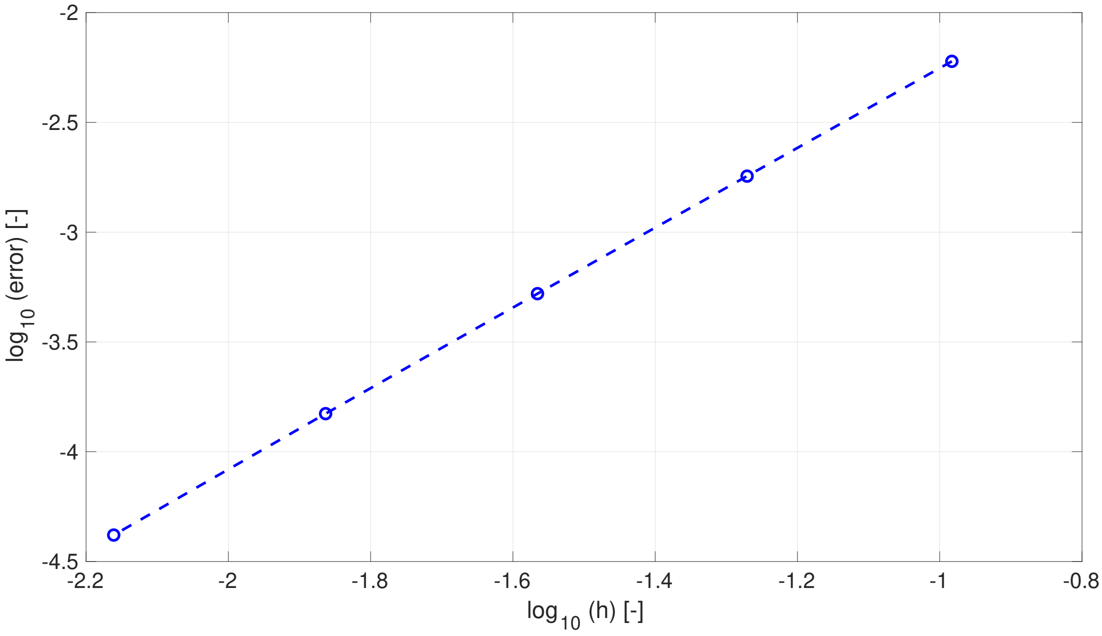


Figure 1: Convergence for increasing number of elements in the domain.

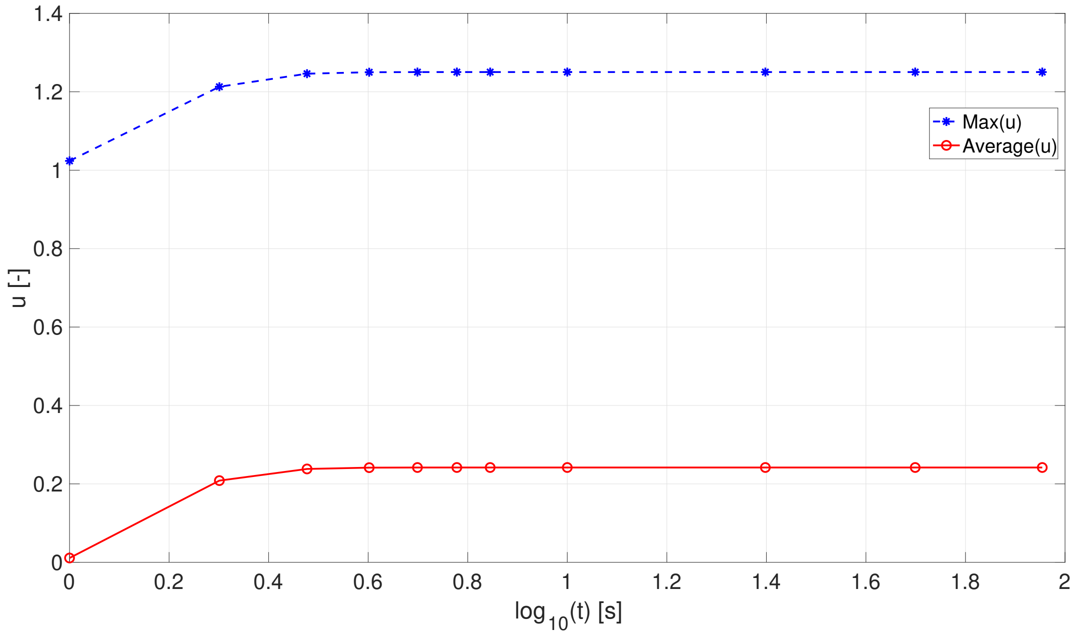


Figure 2: Maximum of u for different maximum of time t.

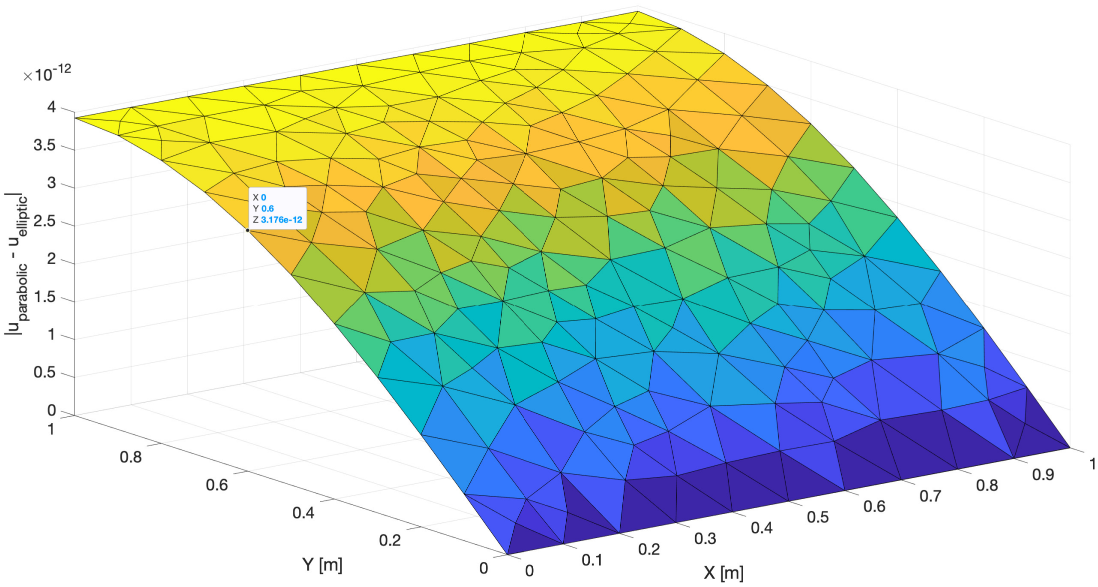


Figure 3: Difference of $u(x, y, t)$ using parabolic and elliptic PDE $|U_{parabolic} - U_{elliptic}|$.

3 Improvement for long time simulations

The idea in this task is to find a way to solve the studied PDE more efficiently. As seen in the previous task, for long time simulations the results can be considered constant. This finding allows us to consider no time dependence and therefore the PDE solved can be treated as an *elliptic* PDE.
To prove this statement numerically, two models were compared. One model is the *parabolic* PDE with $t_{end} = 50$ and the other one is the *elliptic* PDE considering $f(t = 50) \approx 0$ and *no time dependence* $\partial u/\partial t = 0$ by definition. The obtained results shows that the error is below $4 \cdot 10^{-12}$, see Figure 3, which is a very good approximation. Also, considering that in the first case (*parabolic*) the vector result is a matrix of dimensions $(t + 1; n_{elements})$ and for the *elliptic* case the result vector is $(1; n_{elements})$ there is an important reduction of the time needed to solve the problem due to the fact that the system of equations is smaller.

4 Conclusions

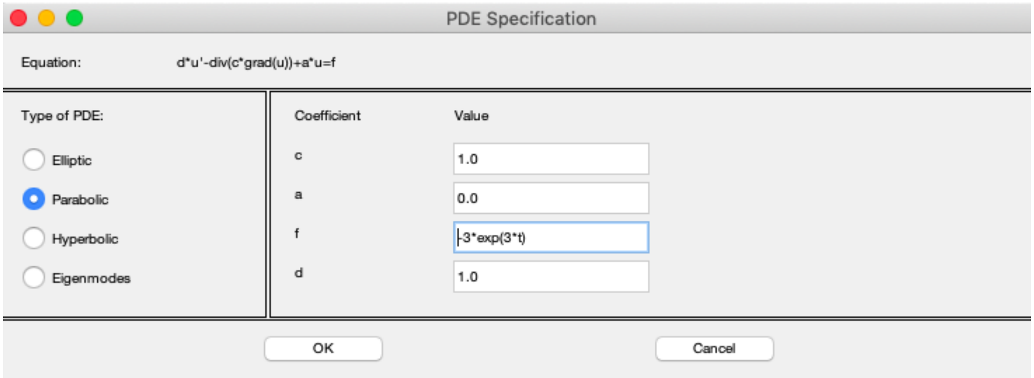
In this assignment the PDE-Toolbox of MATLAB was used to analyse a PDE in a certain domain Ω. The general case of the PDE requires a parabolic description of the solution because of the presence of time derivatives of the unknown function $u(x, y, t)$. The theoretical convergence was proved by discretizing the domain Ω four times and reducing the error as expected. Afterwards, the time dependence of the PDE was evaluated. The results were influenced for values of $t_{end} \leq 4$, and remained almost equal for bigger values of t_{end}. Considering these results, it was finally probed numerically that for $t_{end} = 50$ the difference of solving the PDE using a *Parabolic* or *Elliptic* description is below $4.0 \cdot 10^{-12}$ solving a much less complicated system of equations.

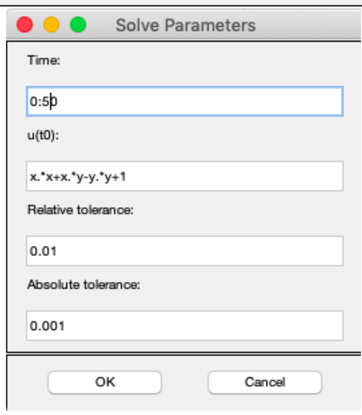
Appendix

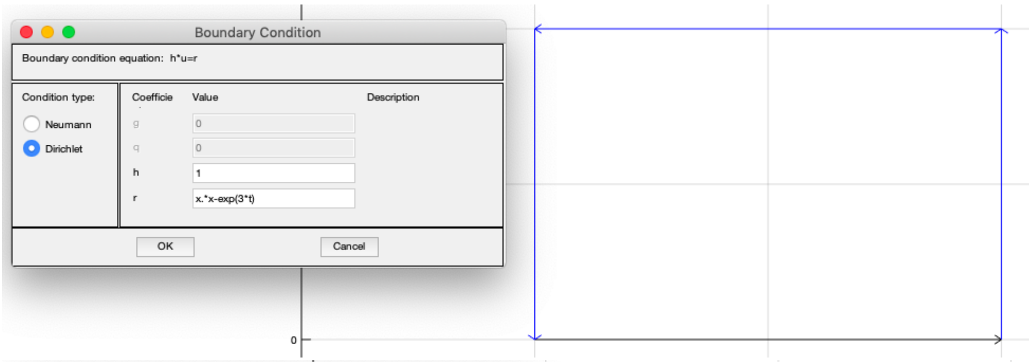
In the following the obtained results for increasing t_{end} are shown.

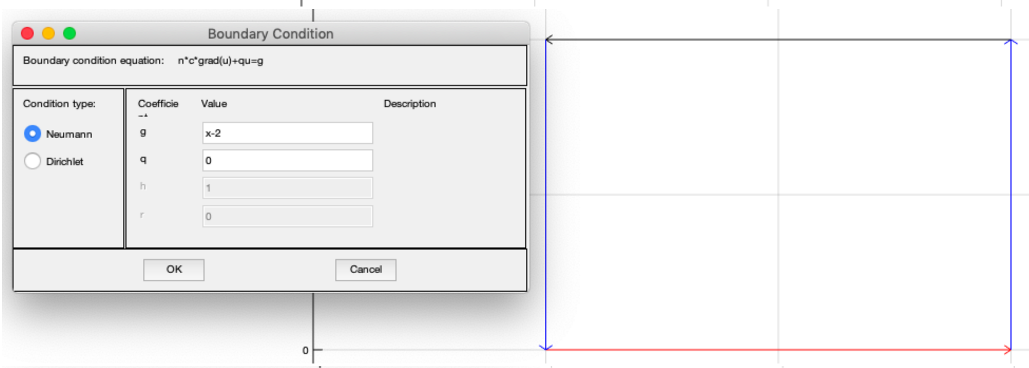
t_{end}	Average(u)	Max(u)
1	0.010926525052259	1.023759650549057
2	0.208362602490835	1.212882112877206
3	0.238073693482186	1.246171590832516
4	0.241405636524575	1.249915396675338
5	0.241737366628505	1.250289123382596
6	0.241773641897925	1.250329977800336
7	0.241772117897548	1.250328258196478
10	0.241770807199974	1.250326779238561
25	0.241771198036082	1.250327219835512
50	0.241771198066461	1.250327219869756
90	0.241771198066393	1.250327219869681

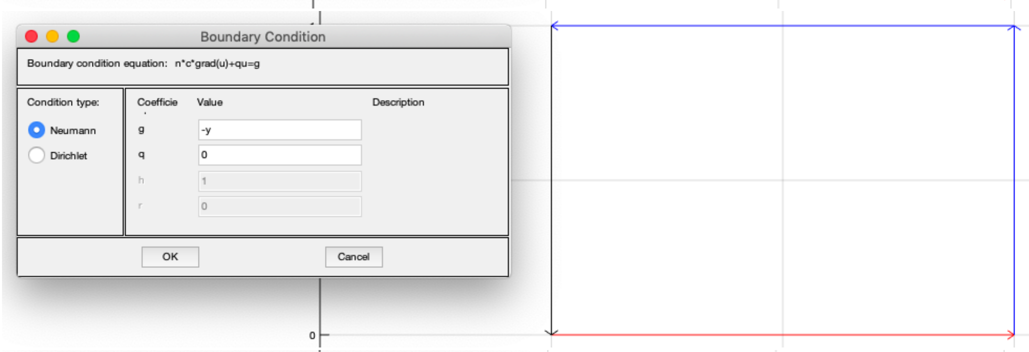
Some of the relevant boundary conditions are plotted for an easier understanding of the problem. Were loaded as shown in the following Figures:

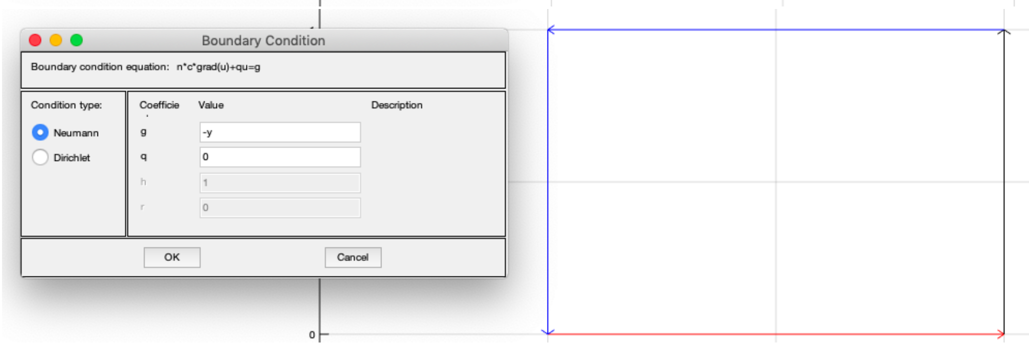





Boundary Condition
Boundary condition equation: h*u=r
Condition type:
Neumann
Dirichlet
Coefficie
Value
Description
g
0
q
0
h
1
r
x.*x-exp(3*t)
OK
Cancel
0


Boundary Condition
Boundary condition equation: n*c*grad(u)+qu=g
Condition type:
Neumann
Dirichlet
Coefficie
Value
Description
g
x-2
q
0
h
1
r
0
OK
Cancel
0


Boundary Condition
Boundary condition equation: n*c*grad(u)+qu=g
Condition type:
Neumann
Dirichlet
Coefficie
Value
Description
g
-y
q
0
h
1
r
0
OK
Cancel
0


Boundary Condition
Boundary condition equation: n*c*grad(u)+qu=g
Condition type:
Neumann
Dirichlet
Coefficie
Value
Description
g
-y
q
0
h
1
r
0
OK
Cancel
0