

Computational Mechanics Tools

PDE – Toolbox assignement

Domenico Pice
12/12/2017

Sommario

1- Equation solution	3
2- Time influence in the solution	9
3- Solution at final time	11

1- Equation solution

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

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

Initial condition with $t=0$

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

Boundary condition:

$$u_n(x = 0, y, t) = -y,$$

$$u_n(x = 1, y, t) = 2 + y,$$

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

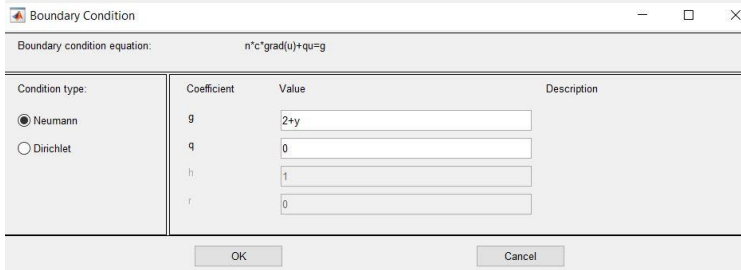
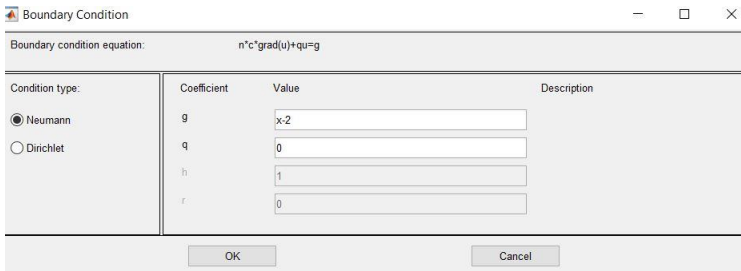
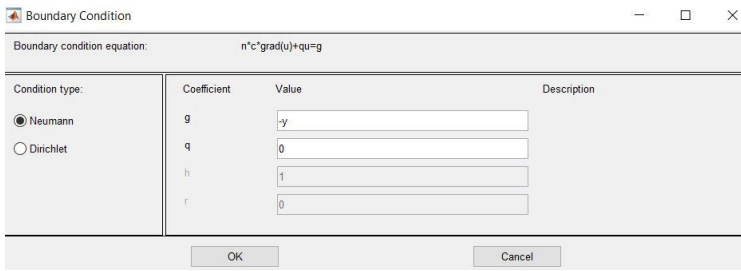
$$u_n(x, y = 1, t) = x - 2,$$

In order to solve the equation with pdetool, it is necessary to insert the equation in the menu: PDE/PDE specification. In this case we have a parabolic equation to write because it depends on the time.

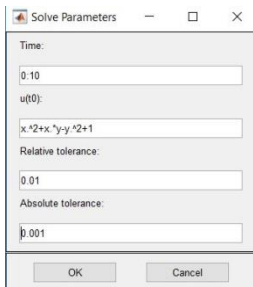
Type of PDE:	Coefficient	Value
<input type="radio"/> Elliptic	c	1
<input checked="" type="radio"/> Parabolic	a	0.0
<input type="radio"/> Hyperbolic	f	-3*exp(-3*t)
<input type="radio"/> Eigenmodes	d	1

Secondly, it is necessary to insert the boundary condition. In this case we have 3 Neumann boundary conditions and 1 Dirichlet boundary condition. It is possible to insert the boundary conditions in the menu: boundary/specify boundary conditions...

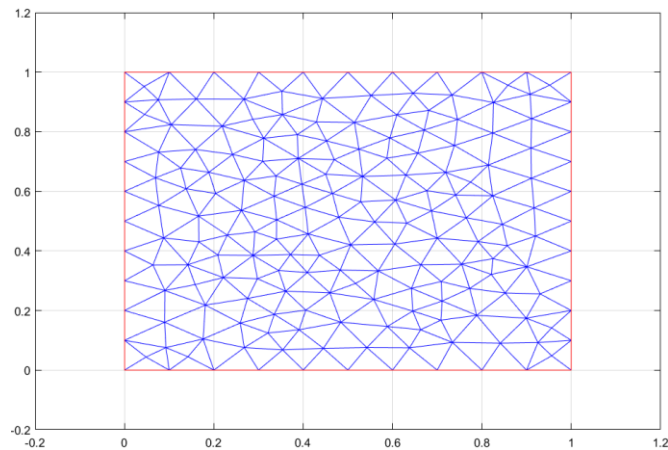
Condition type:	Coefficient	Value	Description
<input type="radio"/> Neumann	g	0	
<input checked="" type="radio"/> Dirichlet	q	0	
	h	1	
	r	x.^2 + exp(-3*t)	



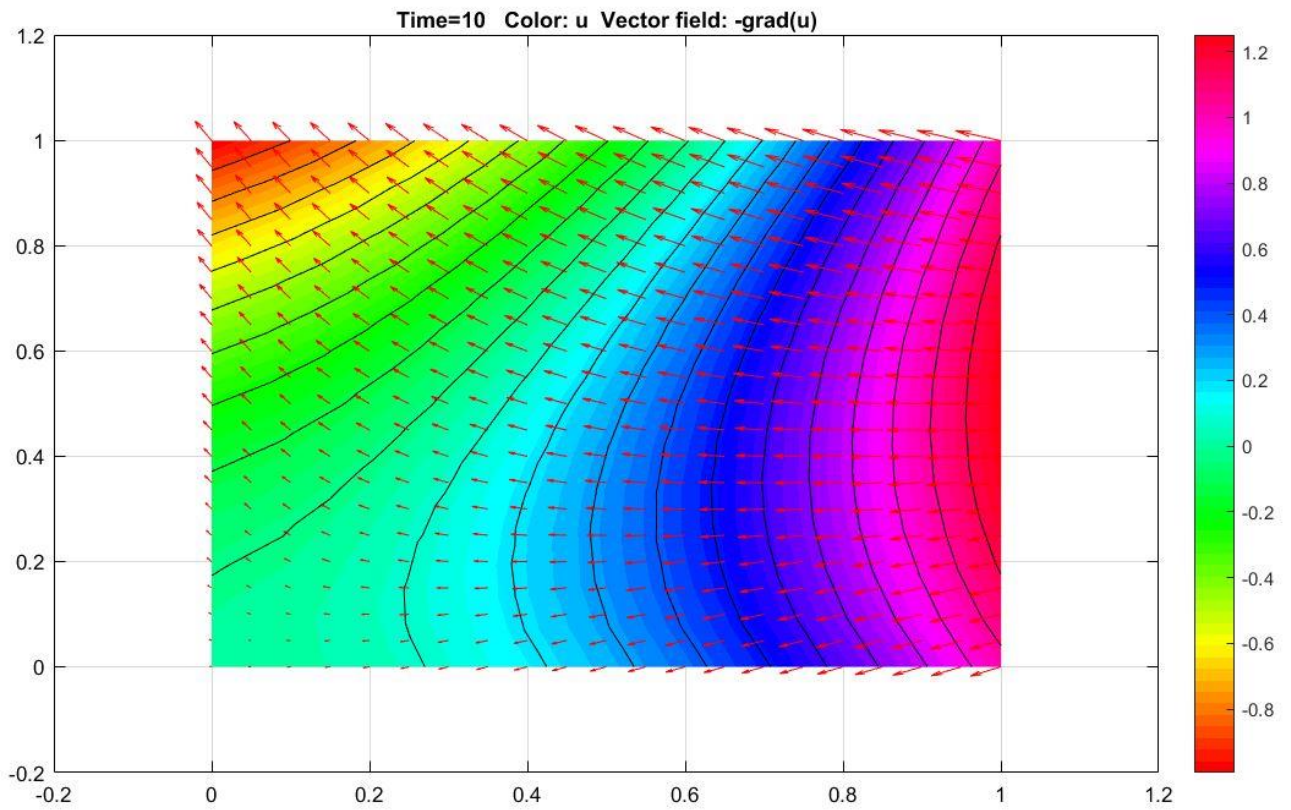
In solve/parameter are inserted the time and the initial conditions.



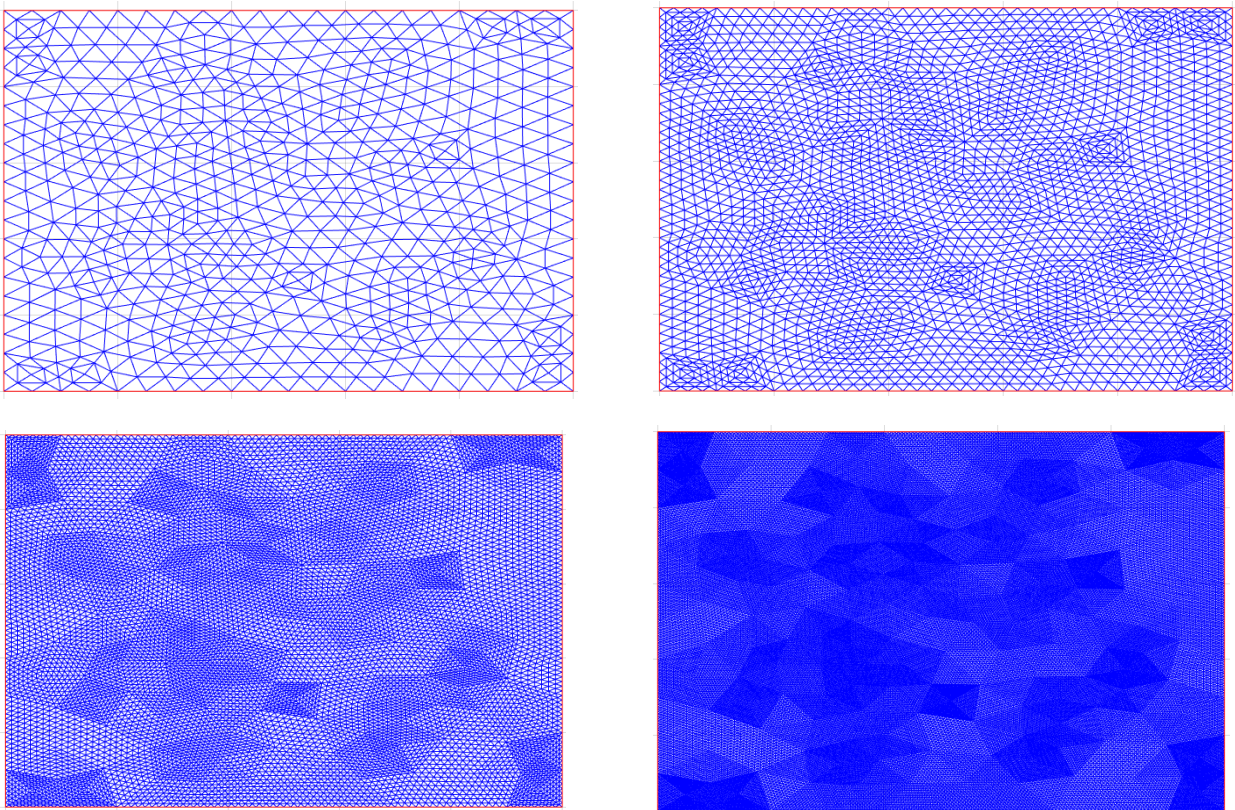
As we define these parameters, we can create the mesh using mesh/initialize mesh.



Finally we can solve the problem: solve/solve pde. The solution obtained is:



In order to verify the convergence of the solution with different meshes, the mesh is refined 4 times, obtaining other 4 similar results.



The objective of the problem is to compare the solution obtained with the finite element method with the exact analytical solution. For this purpose the matlab file “errorExample1” has been modified to compute the error at time equal to 10.

```

Editor - C:\Users\dom91\Desktop\ULTIMOPDE\errorExample1.m
errorExample1.m x convergenceExample.m x assignmentpdetool.m x +
1 - format long
2   % Nodal coordinates (PDETOOL - Export mesh)
3 - X = p(1,:)';
4 - Y = p(2,:)';
5
6   % Conectividades
7 - T = t(1:3,:)';
8
9   % Numerical solution (PDETOOL - Export solution)
10 - Sol = u(:,end);
11
12  % Analytical solution
13 - tt=10;
14 - SolA = X.^2 + (X.*Y) - Y.^2 + exp(-3*tt);
15
16
17  % Norm of the maximum error
18 - errMax = max(abs(Sol-SolA))
19
20 - h=sqrt(2/length(T))

```

This operation is computed for every different mesh. It is easy to observe that the accuracy steady increase with the number of elements.

Finally, to compute the slope it is necessary to find the value of the element composing the mesh and the error for every different mesh.

Knowing that: $\frac{h^2}{2} \cdot N = 1$

(Where h is the length of the side of the triangle, and N is the number of triangle in the mesh) we can calculate h as:

$$h = \sqrt{\frac{2}{N}}$$

(this formula is inserted in the error example function).

Inserting in the file “convergenceExample” the values of h and the values of error calculated, we obtain the slope.

```

errorExample1.m x convergenceExample.m x assignmentpdetool.m x +
1   % Data example t=10
2 - h = [0.078086880944303 0.039043440472152 0.019521720236076 0.009760860118038 0.004880430059019];
3 - err = [0.006662135271410 0.001973332296638 5.678745361551441e-04 1.600364442170266e-04 4.442343484845068e-05];

```

Slope values:

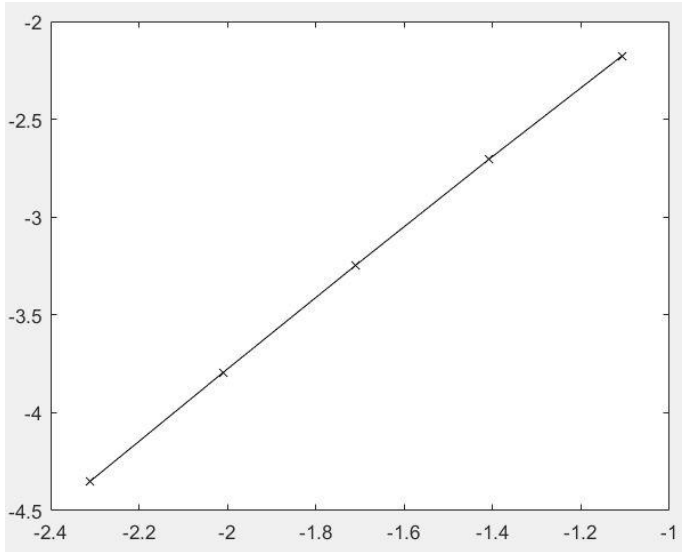
1.755350730319805

1.796989790220328

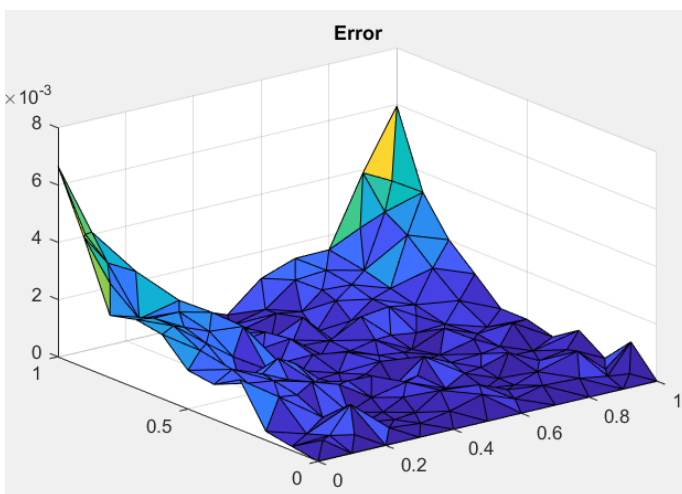
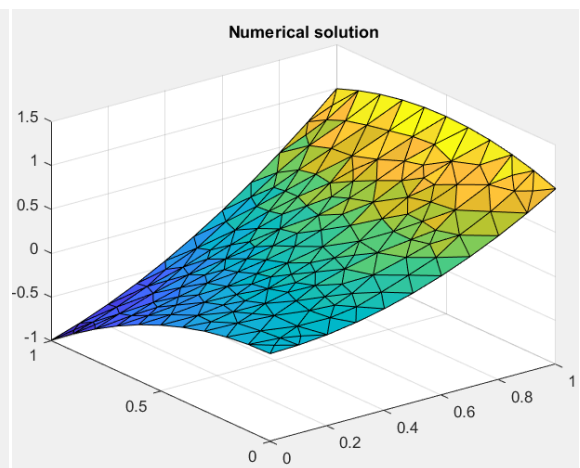
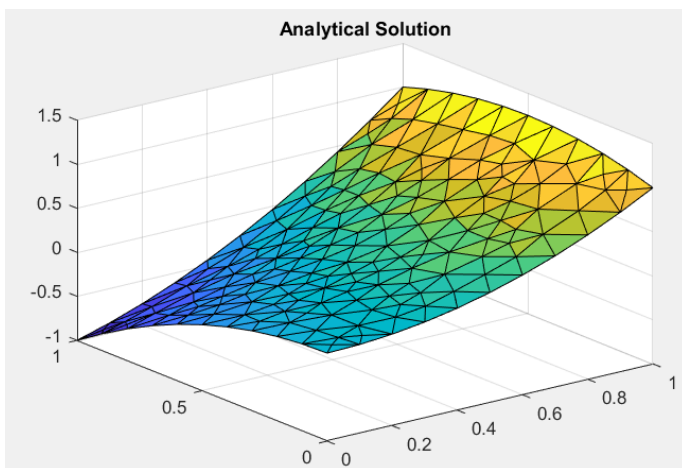
1.827171742363501

1.849007627097741

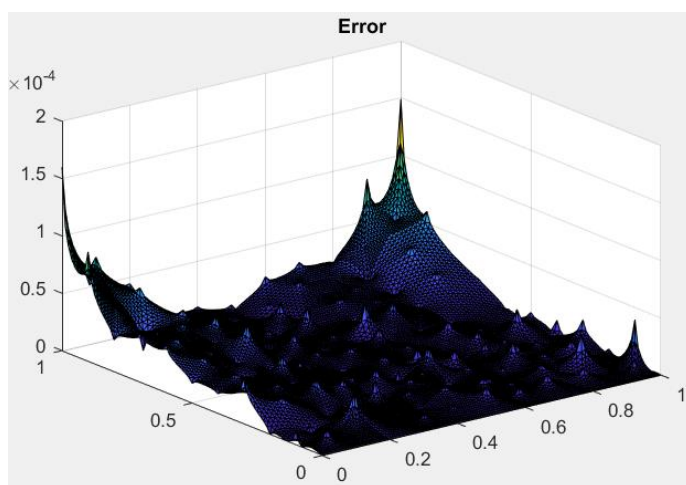
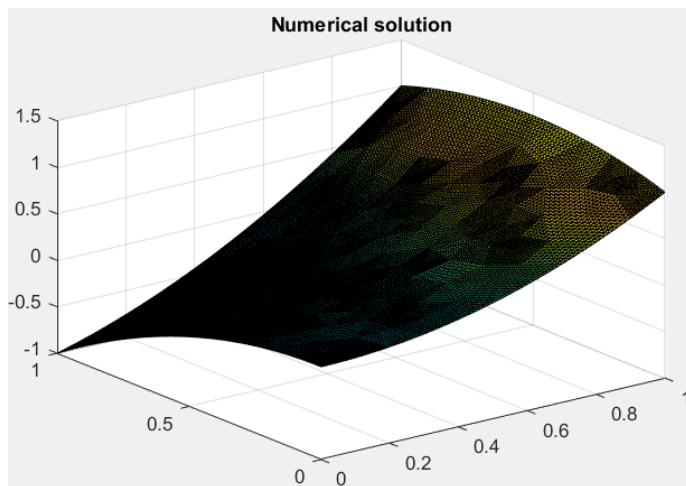
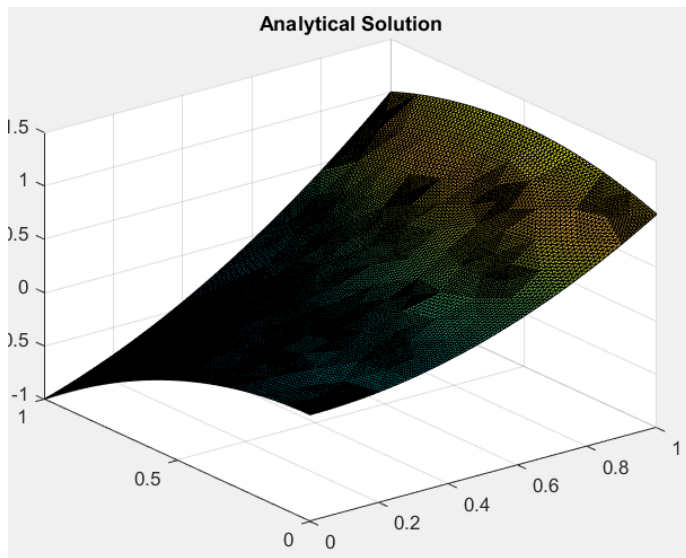
The values are increasing, this means that the numerical method is converging to the exact solution.



In the following graphs we can see the difference between numerical and analytical solution for the most coarse mesh.



In the following graphs we can see the difference between numerical and analytical solution for the most fine mesh.



Comparing the graphs above, it is clear that the error passes from a scale of (10^{-3}) to (10^{-4}) . The solution is converging. With a more refined mesh we can observe the local difference, point by point, in the solution.

2- Time influence in the solution

To verify the time influence in the solution, the procedure above mentioned is repeated inserting a value of time equal to 1 second. The graphs below show the solution for coarse and fine meshes.

Solve Parameters

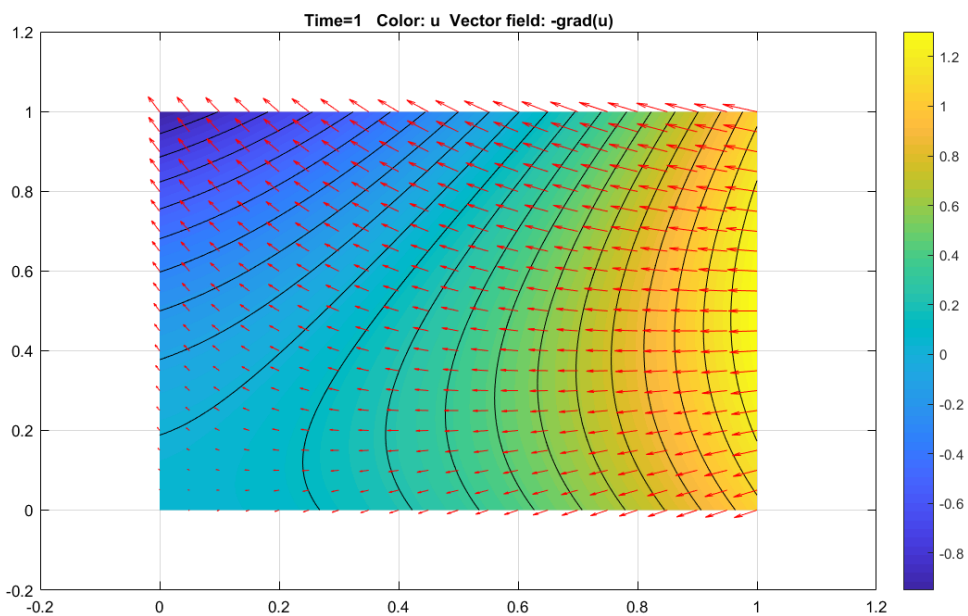
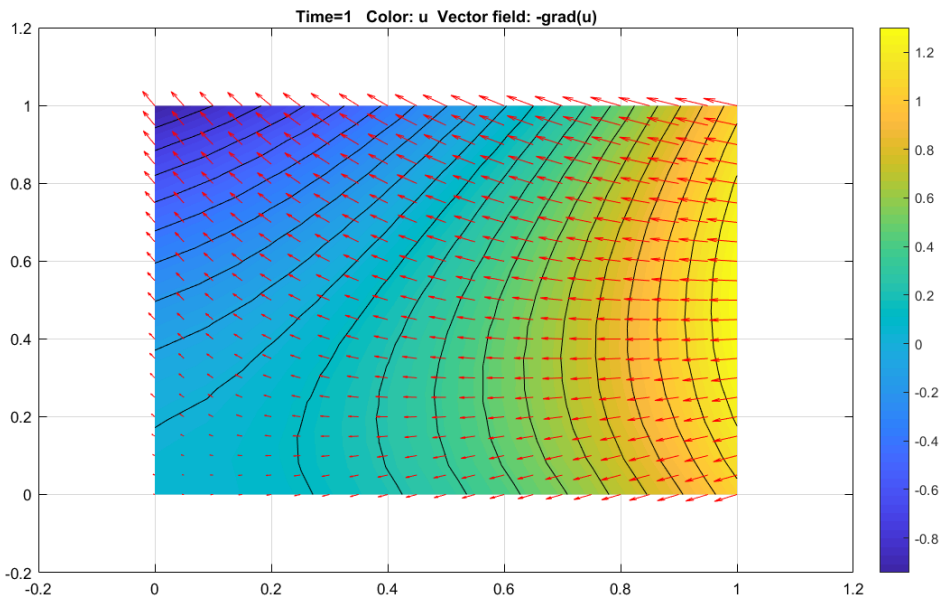
Time:

$u(t_0)$:

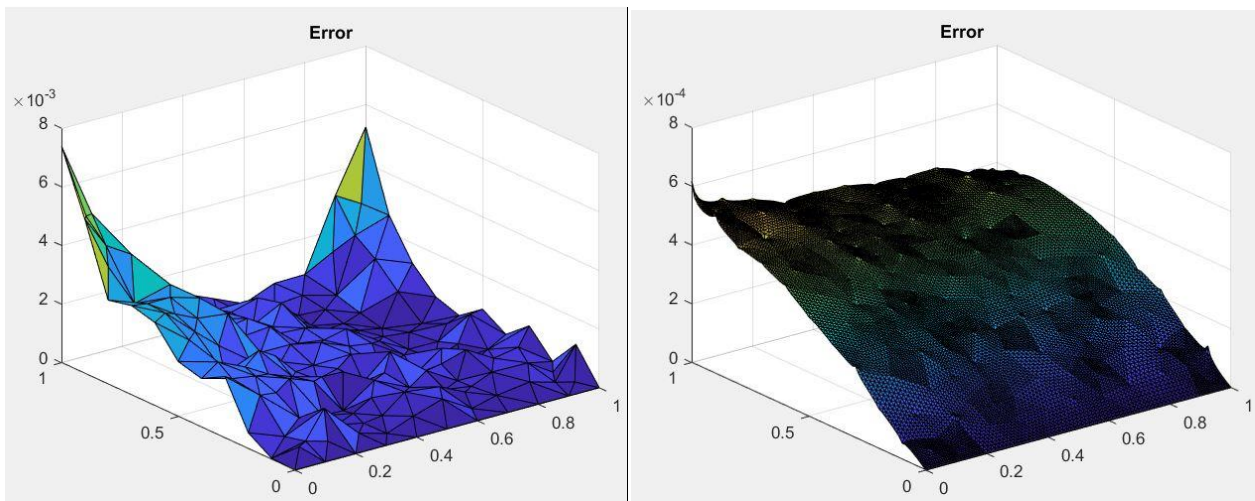
Relative tolerance:

Absolute tolerance:

OK Cancel

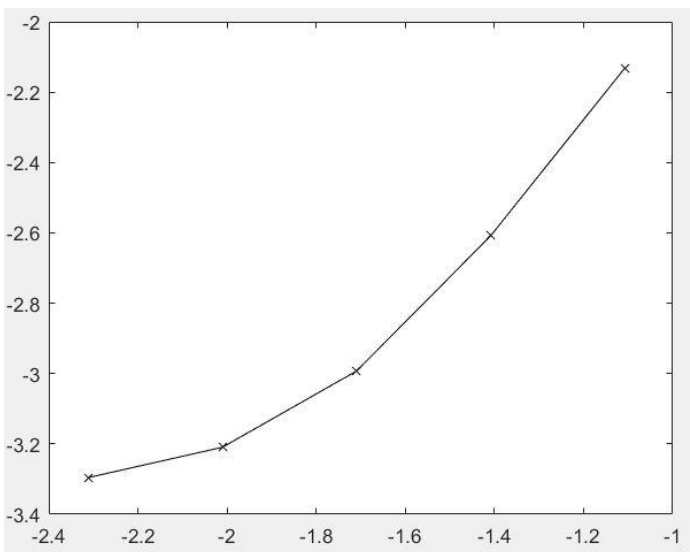


Errors with different meshes:



In this case the convergence is increasing but the error is bigger. This means that time influences the solution. The influence of time is bigger for $t < 1$ because in this partial differential equation we have a negative exponential that depends on t . This means that higher values of t , the exponential suddenly reaches really low and irrelevant values.

```
% example 2 t=1
h = [0.078086880944303 0.039043440472152 0.019521720236076 0.009760860118038 0.004880430059019];
err = [0.007388099174234 0.002464530877717 0.001016772029200 6.172782551673306e-04 5.061650593674028e-04];
```



Slope values for t=1

- 0.286313049024039
- 0.720003372693163
- 1.277316808109969
- 1.583890176063761

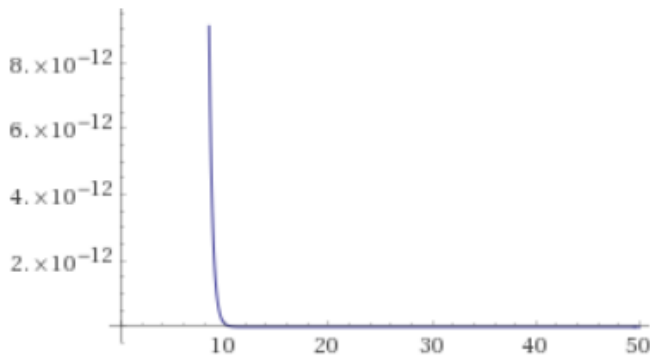
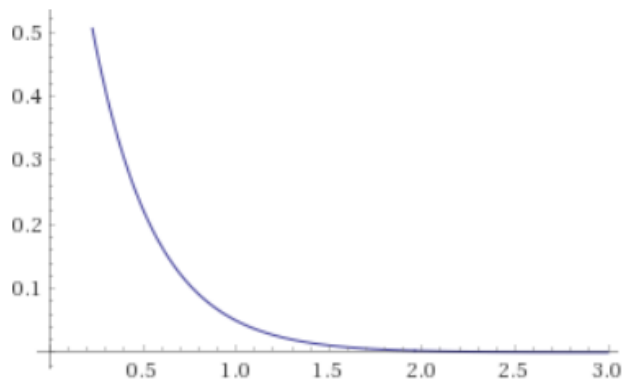
Comparing the results with this new time discretization, it is clear that time influences the solution only at the beginning of the simulation. Indeed the results are similar but not the same.

3- Solution at final time

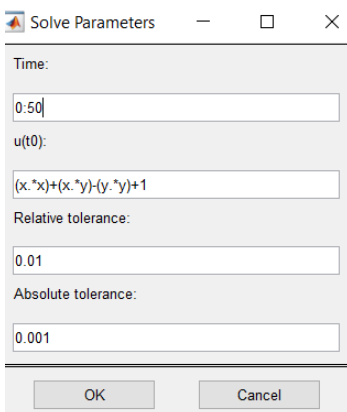
The considerations above are confirmed solving the equation with time equal to 50 seconds.

The analytical explanation of this solution is that the equation $f = \exp(-3t)$ reaches values close to 0 with time higher than 1,5. For this reason, with time equal to 50 seconds, the solution does not depend from the time. This is demonstrated with pde tool, removing the exponential part of the equations in the “boundary conditions” and “pde specification”.

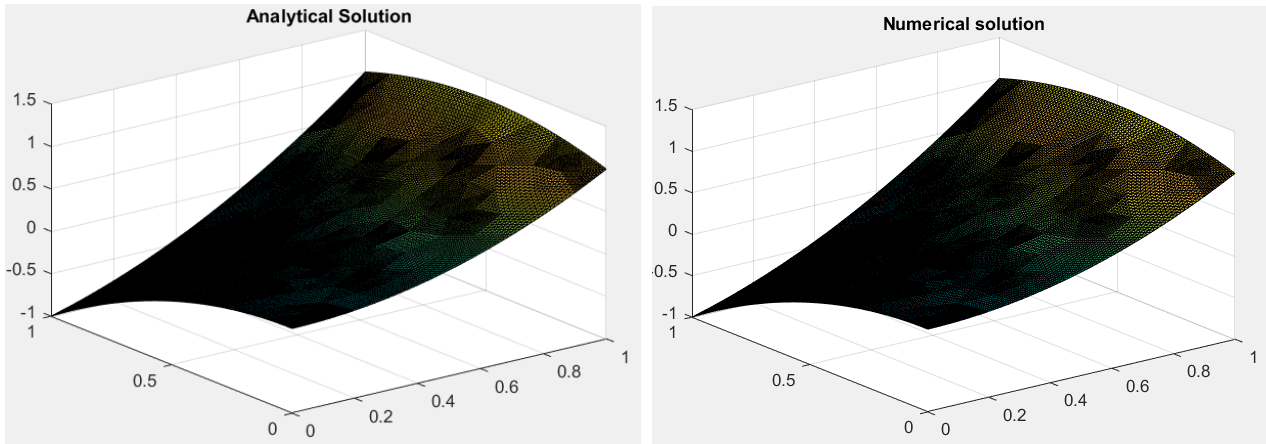
The graphs below show the steady decrease of the function: $f = \exp(-3t)$



To demonstrate the irrelevance of time for $t=50$ the equation is resolved 2 times. At first imposing time equal to 50 without changing the pde specification and the boundary condition, then transforming the equation eliminating the time dependence.

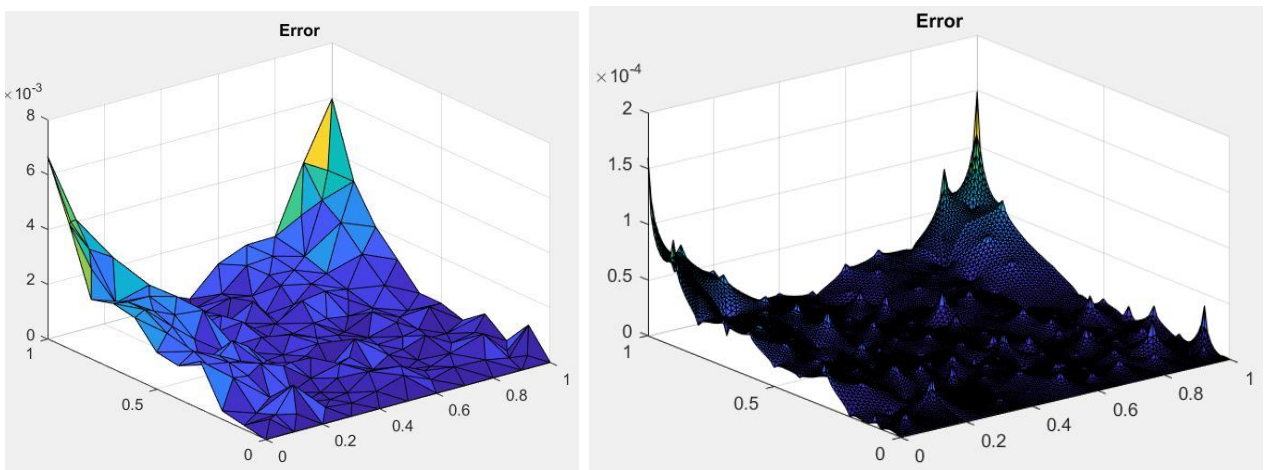


Solutions for the fine mesh:



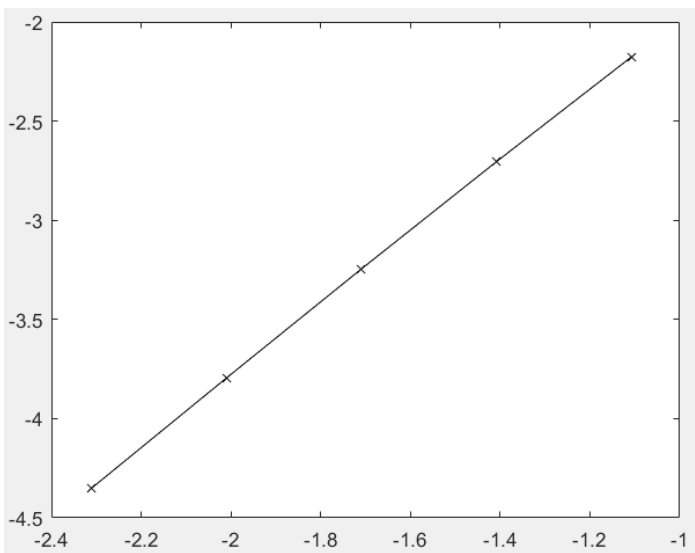
Comparing these results with the ones obtained with $t=10$ we can see that they are really similar. Even the error and the slopes confirm this assumption.

Errors obtained with different meshes:



```
%% example 3 t=50
```

```
h = [0.078086880944303 0.039043440472152 0.019521720236076 0.009760860118038 0.004880430059019];  
err = [0.006662018152399 0.001973319082656 5.678757245667354e-04 1.599212100461545e-04 4.441365703522493e-05];
```



Slope values:

1.755335028525100

1.796977110323700

1.828213947644191

1.848286020113484

In this case the slopes values are increasing and reach almost 2. This means that the solution is correct and converging.

The purpose is to find a more efficient manner to solve the problem. It has been demonstrated before that time is not influent. For this reason the exponential can be deleted. Doing this operation we are transforming the problem from parabolic to elliptic.

PDE Specification

Equation: $-\text{div}(c \cdot \text{grad}(u)) + a \cdot u = f$

Type of PDE:	Coefficient	Value
<input checked="" type="radio"/> Elliptic	c	1.0
<input type="radio"/> Parabolic	a	0.0
<input type="radio"/> Hyperbolic	f	0
<input type="radio"/> Eigenmodes	d	1.0

OK Cancel

For this purpose the Dirichlet boundary condition must be changed.

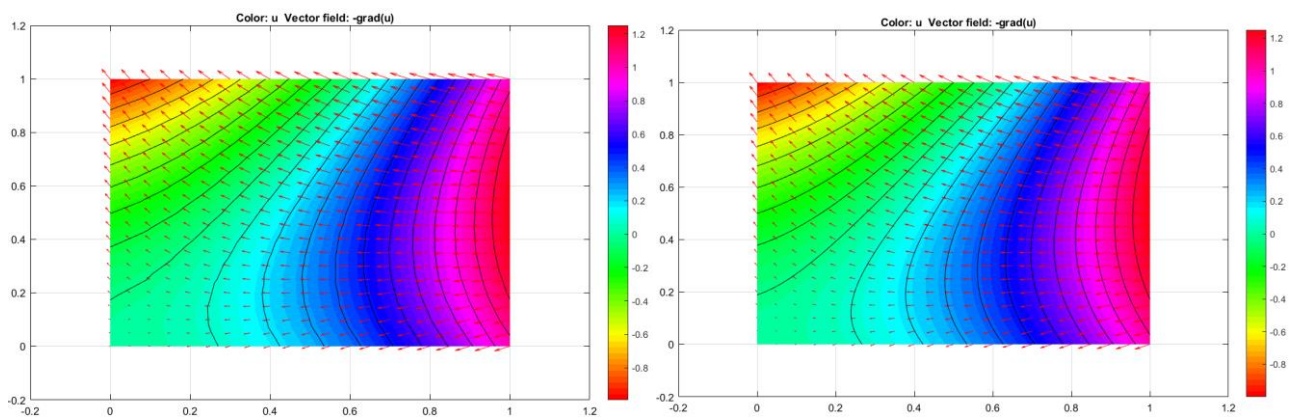
Boundary Condition

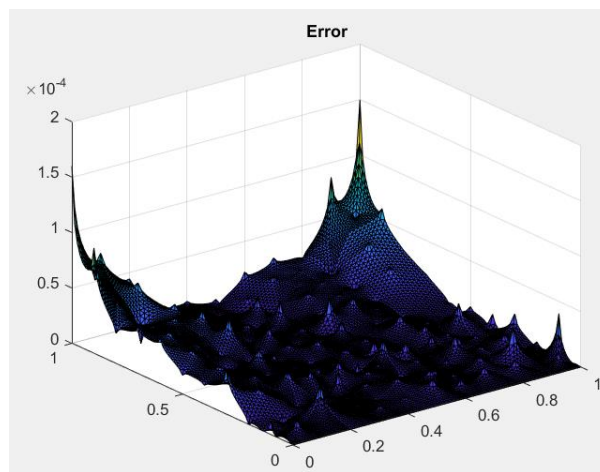
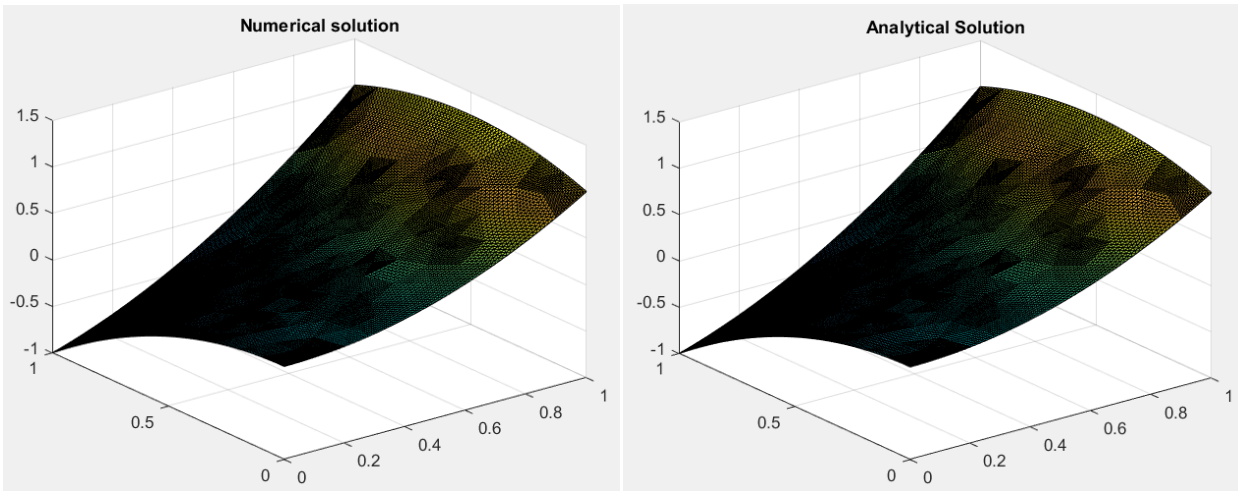
Boundary condition equation: $h \cdot u = r$

Condition type:	Coefficient	Value	Description
<input type="radio"/> Neumann	g	0	
<input checked="" type="radio"/> Dirichlet	q	0	
	h	1	
	r	$(x.^2)$	

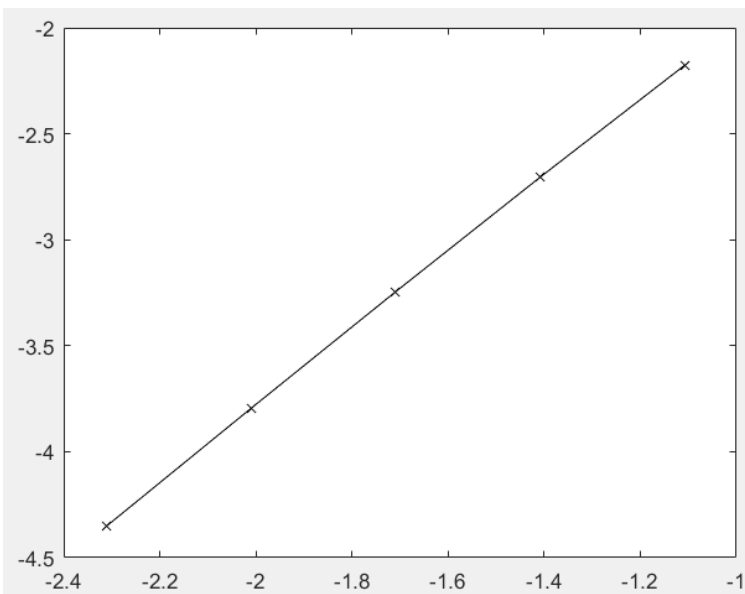
OK Cancel

Solution for coarse mesh and fine mesh.





```
% example 3 t=50 elliptic without exp
h = [0.078086880944303 0.039043440472152 0.019521720236076 0.009760860118038 0.004880430059019];
err = [0.006662018152480 0.001973319082695 5.678757245809463e-04 1.599212100485969e-04 4.441365711382872e-05];
```



Slope values:

- 1.755335028514127
- 1.796977110316110
- 1.828213947658260
- 1.848286017582220

In this case the slopes values are increasing and reach almost 2. This means that the solution is correct and converging.

In this case the results are exactly the same as before. This means that at final time, it is possible to solve an easier elliptic equivalent equation. In this way we use less computational resources.