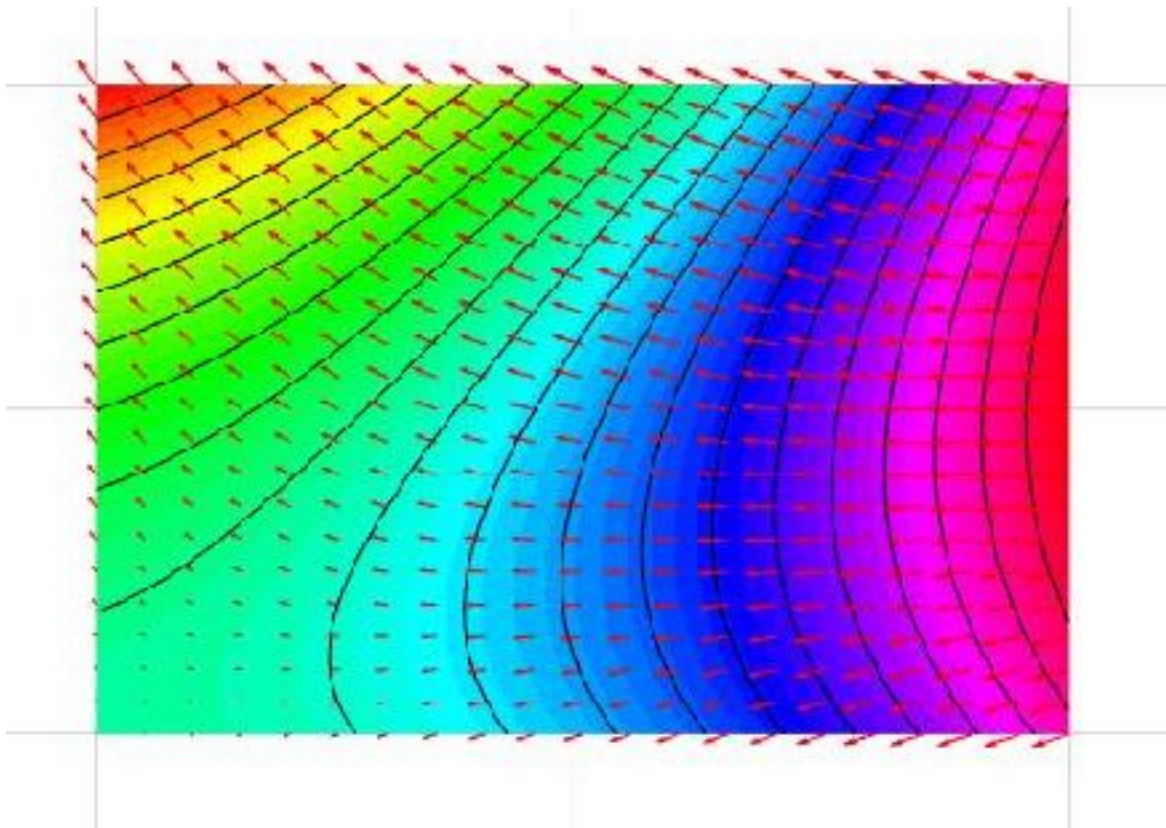

Computational Mechanics Tools

Assignment 2

PDE Toolbox



Introduction

Solve the following problem with the MATLAB PDE Toolbox:

$u_t - \Delta u = f$ in $\Omega = [0, 1]^2$, where the source term is given by

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

We consider an initial condition at $t = 0$:

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

We use Neumann and Laplace Boundary conditions and an initial condition.

In the exercise the analytical solution is given as:

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

Convergence Order

The PDE toolbox is build in application for Matlab. Partial Differential Equation Toolbox™ provides functions for solving partial differential equations (PDEs) in 2-D, 3-D, and time using finite element analysis. I created the geometry and implemented the boundary conditions and the equations.

I set the calculation time to 10 seconds and refine the mesh four times. With the Matlab Script error and the script convergence we calculate the convergence. To understand the convergence of the PDE toolbox its necessary to calculate the difference between the PDE Toolbox Solution with the analytical solution by calculating the absolute difference. The convergence is given by the error dependent from the refinement of the mesh (h).

As you can observe in the following plot the numerical convergence which is the calculated error of the PDE toolbox solution has almost the expected convergence order by refining the mesh size. For this equations the expected slope is 2.

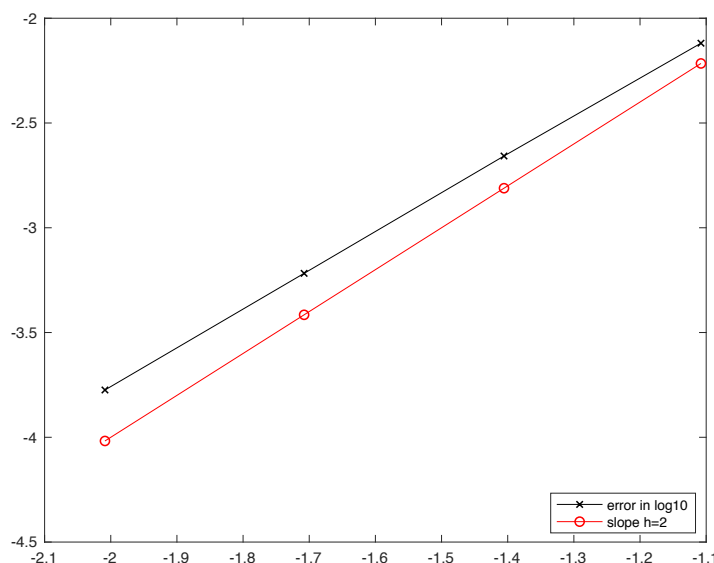


figure 1: log scaled plot of the calculated error over the log of the mesh size h . $t=10s$

In the second task we should compare the convergence order by reducing the calculation time. The new time is one second. There error is almost constant over the time and doesn't equal to the slope of 2.

For short times the PDE toolbox is very certain independent of the mesh quality. The reason is the solver with the with the parabolic PDE.

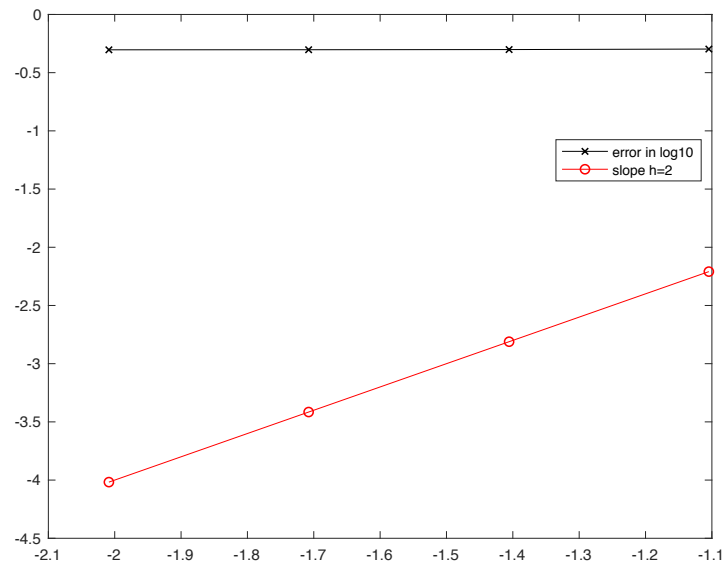


figure 1: log scaled plot of the calculated error over the log of the mesh size h. t=1s

The third task was to find an efficient way to calculate the solution for the time of 50 seconds. I wasn't able to find a solution. A possible way is to calculate the solution with the pdf tool with a very high mesh quality but this is not possible with common solvers. Maybe its possible to calculate with low mesh quality and extrapolate the solution dependent of the convergence behavior of the solution.

Appendix

t=10

h: 0.078, 0.0393, 0.196, 0.0098

err: 0.0076, 0.0022, 6.0597e-04, 1.6817e-04

slopes are:

1.8085

1.8534

1.8493

t=1

h: 0.0786, 0.0393, 0.0196, 0.0098

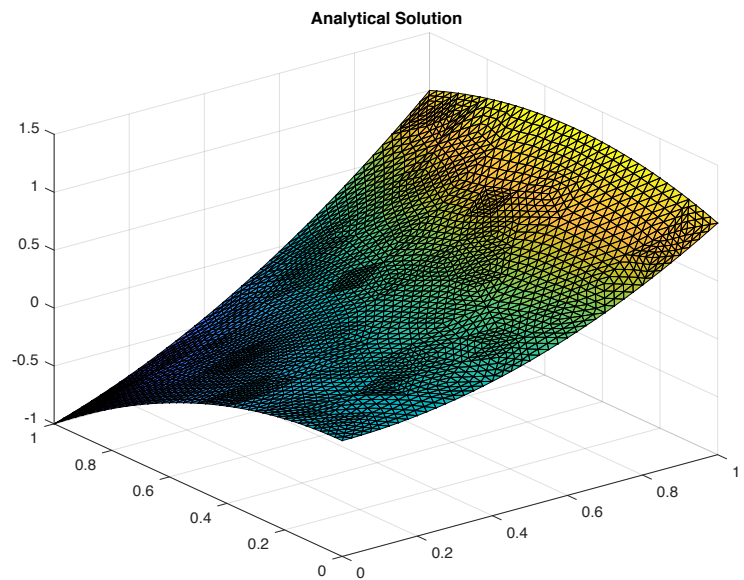
err: 0.5044, 0.4994, 0.4977, 0.4969

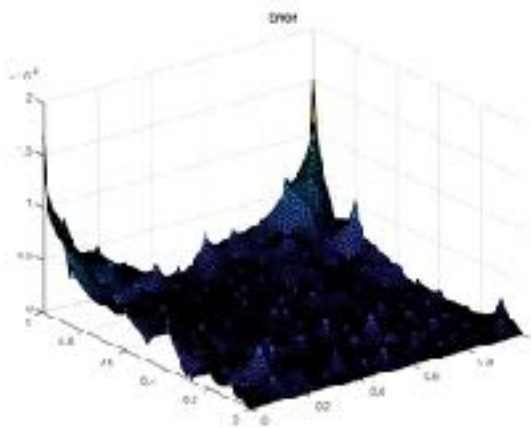
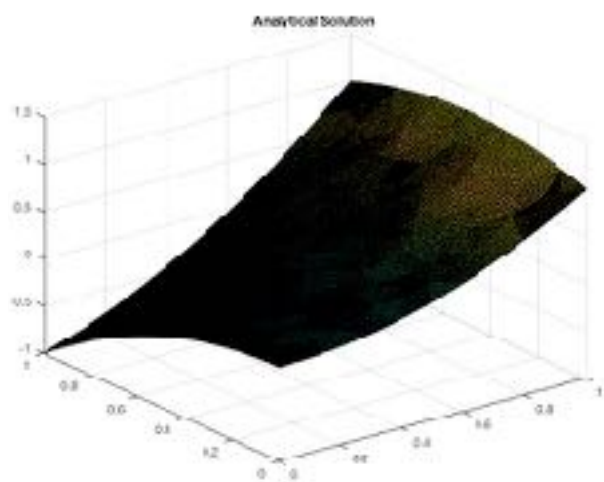
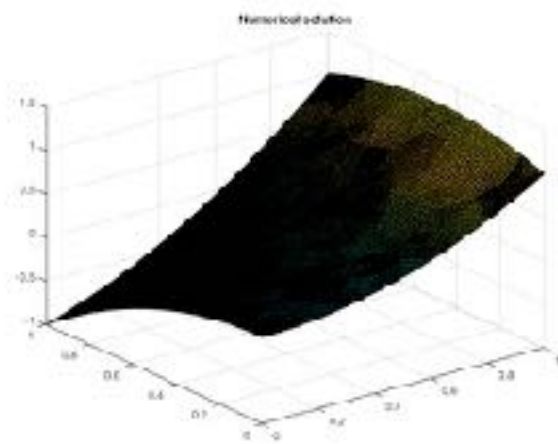
the slopes are:

0.0144

0.0049

0.0023





Example for the error calculation for $t=10s$ and a high mesh quality