DTU Compute
Department of Applied Mathematics and Computer Science

Solving 1D time dependent differential equations using the Spectral/hp-FEM method

Johanne Hvidberg Conradsen, Marcus Kristian Nielsen, Morten Rahbaek Boilesen

Introduction

A classical example of a partial differential equation, (PDE), is the one-dimensional Heat equation

$$\frac{\partial u}{\partial t} = D\frac{\partial^2 u}{\partial x^2} + q\ ,\ (x,t) \in (0,L) \times \mathbb{R}_+.$$

where $u(x,t)$ is the temperature distribution, $q(x,t)$ is a source function, D is the diffusion coefficient (assumed constant). We use a Dirichlet boundary condition $u(0,t) = u(L,t) = 0$, together with an initial condition $u(x,0) = \phi(x)$. Where $\phi(x)$ is some given function specified by the problem.

Methods

The Heat equation is a time-dependent PDE, and to solve it numerically we have to both deal with a discretization of time and space.

Spatial discretization: We initially discretized using finite element method (FEM) and then implemented the spectral element method (SEM). In a p-order SEM, the linear hat functions are replaced by p-order polynomials. At each element, p+1 points is required to fit the polynomial. The local solution is represented on the form

$$u(r) = \sum_{n=0}^{P} \hat{u}_n \psi_n(r)$$

where $\{\psi_n\}$ consists of normalized Legendre polynomials and constitutes an orthonormal basis. We used the Legendre-Gauss-Lobatto points for the point distribution within elements.

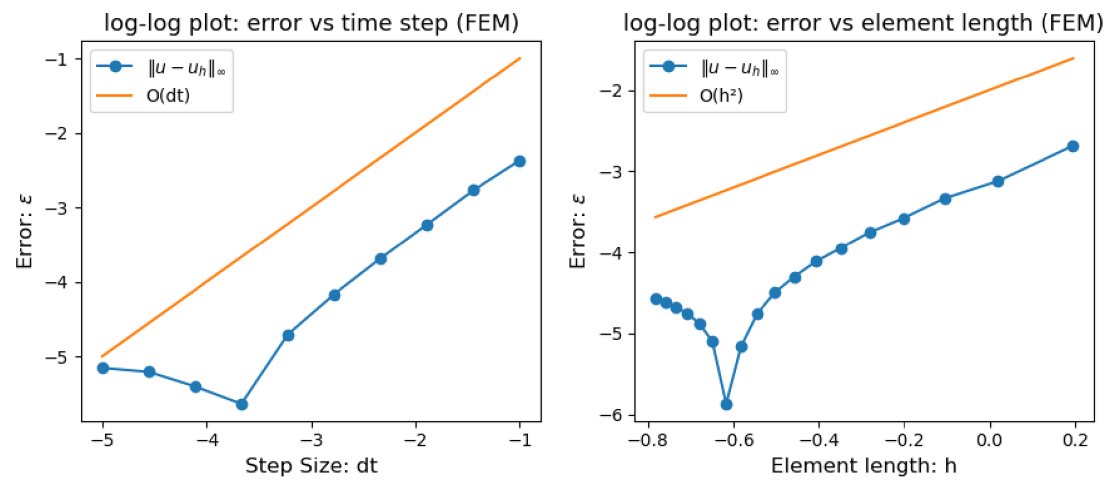
Time discretization: For the time discretization we use the Backward Euler method.

Solving Time-Dependent Problems with FEM

To demonstrate solving time-dependent problems we setup a couple of test problems. For our first test problem we set $q(x,t) = 0$, and as our initial condition we use $\phi(x) = \sin(x)$. In this case one can show that $u(x,t) = \sin(x)e^{-t}$ is the solution. Taking the infinity norm of the difference between the analytical solution u and the numerical u_h, we can find the error ε

$$\varepsilon = \|u(x,t) - u_h(x,t)\|_\infty$$

The FEM is a second order method, $O(h^2)$, and the backward Euler is a first order $O(dt)$. We try to verify these theoretical results by varying the discretization and observing the behavior of the error. This can be seen on the following plots



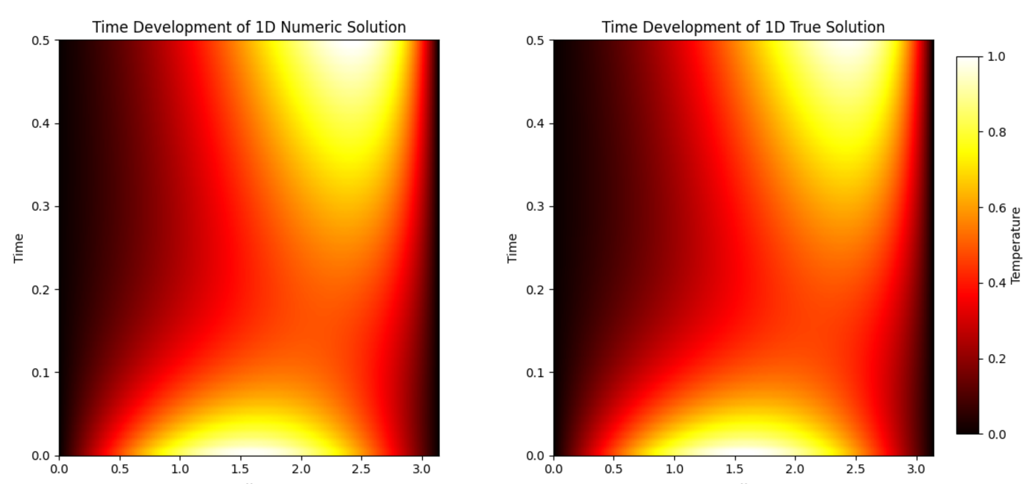
For our second test case we introduce a source term

$$\begin{aligned} q(x,t) = {} & (1-\pi^2)\sin(x)e^{-\pi^2 t} \\ & + kx((\pi - x) - (\pi - x - 1)t)e^{x} \\ & + k(2 - 2(\pi - x) + x)e^{x} t \end{aligned}$$

with an initial condition $\phi(x) = \sin(x)$, this problem has the solution

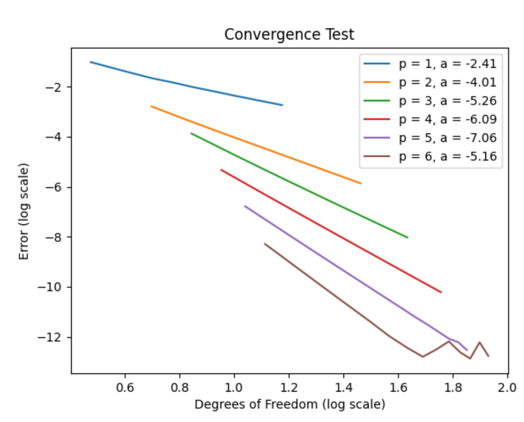
$$u(x,t) = \sin(x)e^{-\pi^2 t} + kx(\pi - x)e^{x} t$$

Below is a comparison of the analytical and the numerical solution found using our FEM solver. The solution was generated using $dt = 0.01$ and 150 elements.



Solving BVP's with SEM

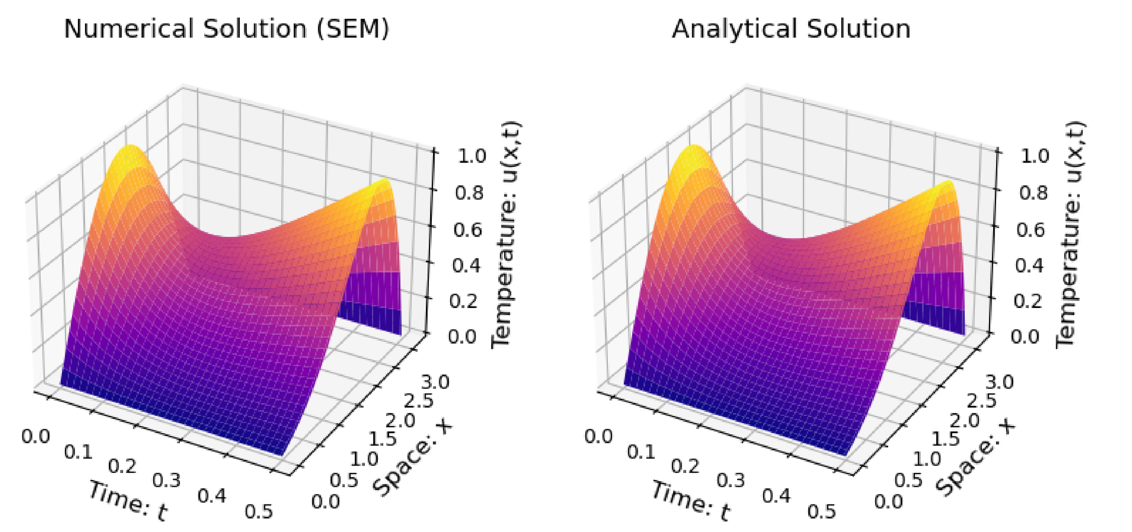
We look at the example from chapter 1 where we have to find the solution to $u'' - u = 0$ with boundary conditions $c = 1$ and $d = e^2$. The plot below presents the convergence test from our solver for different orders of p. The slopes for each convergence test doesn't correspond completely to the expected value p+1, however one can easily see how a larger order gives a faster reduction in the error. In the table below one can see the computational advantage of increasing the order p. It is most significant when going from p = 1 to p = 2.



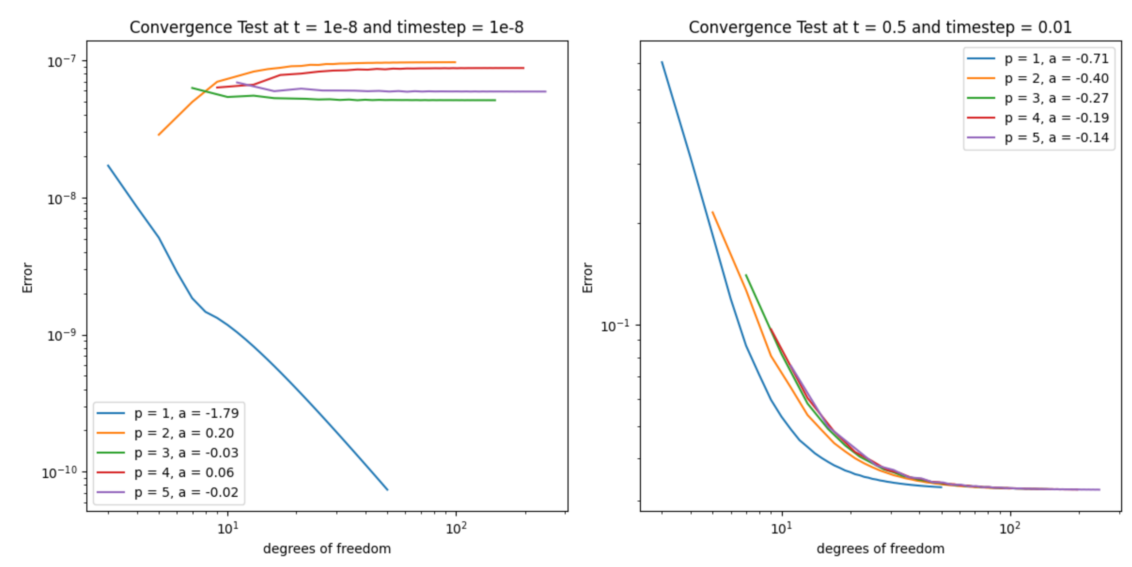
p	Nr of elements	DGF	CPU time	CO2eq kg
1	600	601	12.209	5.6e-5
2	15	31	0.046	2.29e-7
3	5	16	0.013	6.69e-8
4	2	9	0.0001	2.06e-8
5	1	6	0	0
6	1	7	0	0

Solving Time-Dependent Problems with SEM

We here solve the second test case explained in section 3, but now using the SEM solver. Below is seen a 3d plot comparison of the analytical and numerical solution. The solution was generated using dt = 0.005, p=3 and 50 elements.



We investigate the relation between spatial convergence and order p by 1) measuring error after one small time step and 2) measuring error after several time steps. In the left hand plot we would expect something similar to the convergence plot in section 4, but this is not the case, indicating there are still errors in the implementation. In the right hand plot we see that when raising the number of time steps our error converges to the same value.



Conclusions

Our SEM solver works for the time-independent part and we have been able to show how higher order basis functions gives faster convergence and uses less CPU time. In our implementation of the SEM solver for time-dependent problems we have experienced some problems with the convergence tests. One reason could be that the error from the backwards Euler method dominates the error from the SEM solver and therefore makes the order of p obsolete. Another reason could be that we still have mistakes in our implementation of the time-dependent SEM solver. For further research we have learned that we should have more test cases for our functions to make sure that our functions work in general and not only in special cases.