UPPSALA UNIVERSITY

SCIENTIFIC COMPUTING FOR PDE

1TD354 12003 HT2022

Project 1 (FDM) - report

Authors:
Linus FALK
Csongor HORVÁTH

September 27, 2022

Introduction

The aim of this work is to solve a scalar wave equation for various boundary conditions with the help of SBP-Projection method. In order to achieve this we will demonstrate the use of the energy method to prove well-posedness for the continuous and semi-discrete case, derive different well-posed boundary conditions, prove stability with the SBP method and verify the order of accuarcy with the help of convergence with analytic solution. The result will be implemented in MATLAB and presented in form of plots and tables.
In the theoretical part we will use the Energy and SBP-Projection methods to derive well-posed BC and for the discrete cases show stability. In the modelling part we use the discrete cases from the theoretical part end implement it to MATLAB using RK4 as time integration method (*note: we did not concluded convergence studies for the RK4 method when we chose the time steps, because the time steps were determined by the assignment text*). In the implementation the unmentioned parameters in the model were always set to the values given by the assignments.
The above methods used in this project are efficient and easy approaches to show well-posedness and stability for linear partial differential equation and model them in one or two dimensions (higher dimension require exponentially more computing capacity).
So, in the below text we will try and present our results for every part of the project.

$$
\begin{aligned}
&u_{tt} = c^2 u_{xx} && -1 \leq x \leq 1, t \geq 0, \\
&L^{(l)}u = \alpha^{(l)}u_t + \beta^{(l)}u + \gamma^{(l)}u_x = 0, && x = -1, t \geq 0, \\
&L^{(r)}u = \alpha^{(r)}u_t + \beta^{(r)}u + \gamma^{(r)}u_x = 0, && x = 1, t \geq 0, \\
&u = f(x), \quad u_t = 0, && -1 \leq x \leq 1, t = 0,
\end{aligned}
\tag{1}
$$

Note that we will use the folowings in the text below:

- $(a, b)_c = \int_{x_l}^{x_r} a * cb$
- $P = I - \overline{H}^{-1} L^T (L\overline{H}^{-1} L^T)^{-1} L$
- f is initial data
- $e_1 = [1, 0, \ldots, 0],\ e_m = [0, \ldots, 0, 1]$

- D_i is the i-th order SBP operator,

where $\overline{H}$ denote a discrete norm

Energy method

We shall use the energy method and then bound the expression we got from it. Assume that u is real valued, so the conjugate does not matter.

$$\begin{array}{lll}
(u_t, u_{tt}) & = (u_t, c^2 u_{xx}) & = u_t c^2 u_x|_{-1}^{1} - (u_{tx}, c^2 u_x) \\
(u_{tt}, u) & = (c^2 u_{xx}, u_t) & = c^2 u_x u_t|_{-1}^{1} - (c^2 u_x, u_{tx}) \\
+ & & \\
\hline
\frac{d}{dt}(u_t, u_t) & = 2c^2 u_x u_t|_{-1}^{1} - \frac{d}{dt}(u_x, u_x)_{c^2} &
\end{array} \tag{2}$$

So we got the energy estimate: $E = \frac{d}{dt}(||u_t||^2 + ||u_x||^2_{c^2}) = 2c^2 u_x u_t|_{-1}^{1}$

Since $c > 0$ so for well posedness we will need to boun the term $u_t u_x|^1 - u_t u_x|^{-1}$.

Assignment 1

For part 1 we get on the left size $\beta^{(l)} u + \gamma^{(l)} u_x = 0$ and here we need a bound that gives us $-u_t u_x \leq 0$ or $u_t u_x \geq 0$.
It means that u_t and u_x should have the same sign or one of them should be zero. It can be guaranteed more ways. First choosing $\beta^{(l)} = 0$ and $\gamma^{(l)}$ to be non zero. Secondly if $\gamma^{(l)} = 0$ and $\beta^{(l)}$ non zero, than u is zero, therefore it's derivatives are zero as well. And the last option is equivalent to choosing $\gamma(l) = 1$ and $\beta^{(l)} \leq 0$. In this case we got $2c|\beta^{(l)}| \cdot u_{xt} u_x = 2c|\beta^{(l)}| \frac{d}{dt} ||u_x||^2 \geq 0$.
On the other boundary, on the right size we got $\beta^{(r)} u + \gamma^{(r)} u_x = 0$ and here we need a bound that gives us $u_t u_x \leq 0$. It means that u_t and u_x should have different sign or one of them should be zero. It can be guaranteed by similar way as in the other boundary. So first $\beta^{(r)} = 0$ and $\gamma^{(r)}$ to be non zero. Secondly to chose $\gamma^{(r)} = 0$ and $\beta^{(r)}$ to be none zero. And the third option is equivalent to choosing $\gamma^{(r)} = 1$ and $\beta^{(r)} \geq 0$.
And we can combine these boundary conditions the two side by choosing an arbitrary BC on both side.

Assignment 2

So we know the energy estimate, which is: $E = \frac{d}{dt}(||u_t||^2 + ||u_x||^2_{c^2}) = 2c^2 u_x u_t|_{-1}^{1}$
First consider the part, where the data is given:
From the given data we got $u_t = cu_x$ on the left and $u_t = -cu_x$ on the right boundary.

On the left boundary we need $u_t u_x \geq 0$. And with the given data we get that it gives us $cu_x^2 \geq 0$, since $c > 0$.
On the right boundary we get the necessary condition which is $-cu_x^2 \leq 0$, since $c > 0$. So here the given data satisfy the statement: $cu_t u_x \leq 0$. Therefore the energy is bounded and so the equation is well-posed.

Now we will propose the general case here. First consider the cases, where one of the unknown is zero. We will discuss only in one boundary. On the other boundary it's clearly follows how to set the unknowns. And after it we can chose BC on both boundary independently.
So further on we will discuss the left boundary conditions only. So generally if exactly two unknown is zero in one boundary is a good condition. So we only need to discuss the parts, where exactly two is none zero. We discussed the case, when α is zero.
Now discuss the case where β is zero. We can assume in this case that $\alpha = 1$. This gives the condition to the BT to be $-2c^2\gamma u_t^2$. So on the left boundary in this case γ should be negative. (And on the right boundary γ should be positive in this case).
Next case is where γ zero. In this case we couldn't find sufficient condition to well posedness due to the expression can't be transformed to a time derivative square.
In the general case, where nothing is zero, we assume $\gamma = -1$. This gives $u_t(\alpha u_t + \beta u) = \alpha u_t^2 + \beta u_t u = \alpha||u_t||^2 + \beta\frac{d}{dt}||u||^2$. So in this case on the left boundary $\alpha, \beta \geq 0$ is working condition for well posedness.

Assignment 3

$$
\begin{aligned}
u_{tt} &= c^2 D_2 u \qquad & t \geq 0, \\
Lu &= 0 \qquad & t \geq 0, \\
u = f, \quad u_t &= 0 & t = 0,
\end{aligned} \tag{3}
$$

Above the 3 equation is the semi-discretized case using SBP operators. Here we use $L = \begin{bmatrix} e_1^T \\ e_m^T \end{bmatrix}$ from assignment 1.

$$
\begin{aligned}
u_{tt} &= Au \qquad & t \geq 0, \\
u = f, \quad u_t &= 0 & t = 0,
\end{aligned} \tag{4}
$$

Above the 4 equation is the semi-discretized SBP-projection approximation of the problem. Where the discretization matrix: $A = Pc^2 D_2 P$.
Apply discretized energy method, multiply with $u^T H$ and add transpose.

$$
\begin{aligned}
u^T H u_{tt} = c^2 u^T HPD_2Pu = c^2 P^T HPD_2PU = c^2 (Pu)^T HD_2Pu = \\
c^2 (Pu)^T HH^{-1}(-M + BD)Pu = c^2 (Pu)^T (-M + BD)Pu = \\
-c^2 (Pu)^T MPu + c^2 (Pu)^T BDPu \\
+u_{tt}^T Hu = -c^2 (Pu)^T M^T Pu + c^2 (Pu)^T BD^T Pu \\
\hline
= u^T H u_{tt} + u_{tt}^T Hu = -c^2 (Pu)^T (M + M^T)Pu + c^2 (Pu)^T (BD + BD^T)Pu \\
= \frac{d}{dt}\|u_t\|_H^2 = -2c^2 (Pu)^T MPu + c^2 (Pu)^2 BDPu + c^2 BD^T Pu
\end{aligned}
$$

Continue with $BD = e_m d_m - e_1 d_1$ and let $w = PV$. Note $BD = BD^T$

$$
\begin{aligned}
\frac{d}{dt}\|v_t\|_H^2 = -2c^2 w^T Mw + 2c^2 w^T (e_m d_m - e_1 d_1)w = \\
-2c^2 w^T Mw + 2c^2 w^T e_m d_m w - 2c^2 w^T e_1 d_1 w = \\
-2c^2 w^T Mw + 2c^2 (e_m^T w)^T d_m w - 2c^2 (e_1^T w)^T d_1 w = -2c^2 w^T Mw
\end{aligned}
$$

In the last step we use that $Lw = 0$. We can see that the semi-discrete solution mimics the continuous and we have proven stability.

Assignment 4

When introducing u_t to the boundary operator we cant use the previous form of semi-discrete approximation. We therefore introducing the variable substitution $v = u_t$. By creating a system of first order equations we can solve it with RK4.

$$\begin{aligned} \begin{bmatrix} u \\ v \end{bmatrix}_t &= \begin{bmatrix} 0 & I \\ c^2 D_2 & 0 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} \\ L \begin{bmatrix} u \\ v \end{bmatrix} &= 0 \\ \begin{bmatrix} u \\ v \end{bmatrix} &= \begin{bmatrix} f \\ 0 \end{bmatrix} \end{aligned} \tag{5}$$

We introduce $e^{(1)}$, $e^{(2)}$ and the Kronecker-operator to form the boundary operator

$$e^{(1)} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, e^{(2)} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{6}$$

$$L \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \alpha_l (e^{(2)} \otimes e_1) + \beta_l (e^{(1)} \otimes e_1) + \gamma_l (e^{(1)} \otimes d_1) \\ \alpha_r (e^{(2)} \otimes e_m) + \beta_r (e^{(1)} \otimes e_m) + \gamma_r (e^{(1)} \otimes d_m) \end{bmatrix} \tag{7}$$

Apply the projection and energy method to verify stability. We multiply with $\begin{bmatrix} u \\ v \end{bmatrix}^T \overline{H}$ and let: $\overline{H} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} H$

So we get SBP-Projection approximation:

$$\begin{aligned} \begin{bmatrix} u \\ v \end{bmatrix}_t &= P \begin{bmatrix} 0 & I \\ c^2 D_2 & 0 \end{bmatrix} P \begin{bmatrix} u \\ v \end{bmatrix} \\ \begin{bmatrix} u \\ v \end{bmatrix} &= \begin{bmatrix} f \\ 0 \end{bmatrix} \end{aligned} \tag{8}$$

$$\begin{bmatrix} u \\ v \end{bmatrix}^T \overline{H} \begin{bmatrix} u \\ v \end{bmatrix}_t = \begin{bmatrix} u \\ v \end{bmatrix}^T \overline{H} P \begin{bmatrix} 0 & 1 \\ c^2 D_1 & 0 \end{bmatrix} P \begin{bmatrix} u \\ v \end{bmatrix} = \left(P \begin{bmatrix} u \\ v \end{bmatrix} \right)^T \overline{H} \begin{bmatrix} 0 & 1 \\ c^2 D_2 & 0 \end{bmatrix} P \begin{bmatrix} u \\ v \end{bmatrix} =$$

$$\left(P \begin{bmatrix} u \\ v \end{bmatrix} \right)^T \begin{bmatrix} 0 & H \\ c^2 H D_2 & 0 \end{bmatrix} P \begin{bmatrix} u \\ v \end{bmatrix} = \left(P \begin{bmatrix} u \\ v \end{bmatrix} \right)^T \begin{bmatrix} 0 & H \\ c^2 H H^{-1}(-M + BD) & 0 \end{bmatrix} P \begin{bmatrix} u \\ v \end{bmatrix}$$

$$= \left(P \begin{bmatrix} u \\ v \end{bmatrix} \right)^T \begin{bmatrix} 0 & H \\ c^2(-M + BD) & 0 \end{bmatrix} P \begin{bmatrix} u \\ v \end{bmatrix}$$

$$+ \left(P \begin{bmatrix} u \\ v \end{bmatrix} \right)^T \begin{bmatrix} 0 & c^2(-M^T + BD^T) \\ H^T & 0 \end{bmatrix} P \begin{bmatrix} u \\ v \end{bmatrix}$$

$$\frac{d}{dt} \| \begin{bmatrix} u \\ v \end{bmatrix} \|_{\overline{H}}^2 = \left(P \begin{bmatrix} u \\ v \end{bmatrix} \right)^T \begin{bmatrix} 0 & H + c^2(-M^T + BD^T) \\ c^2(-M + BD) + H^T & 0 \end{bmatrix} P \begin{bmatrix} u \\ v \end{bmatrix} =^{(1)}$$

$$w^T \begin{bmatrix} 0 & H \\ H^T & 0 \end{bmatrix} w - c^2 w^T \begin{bmatrix} 0 & M \\ M & 0 \end{bmatrix} w + c^2 w^T \begin{bmatrix} 0 & d_m^T e_m^T - d_1^T e_1^T \\ e_m d_m - e_1 d_1 & 0 \end{bmatrix} w$$

Note: $^{(1)}$ *here we assign* $w = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = P \begin{bmatrix} u \\ v \end{bmatrix}$

Here $c^2 w^T \begin{bmatrix} 0 & d_m^T e_m^T - d_1^T e_1^T \\ e_m d_m - e_1 d_1 & 0 \end{bmatrix} w = c^2 (e_m^T w_2)^T d_m w_1 - c^2 (e_1^T w_2)^T d_1 w_1 + c^2 (d_m w_1)^T (e_m^T w_2) - c^2 (d_1 w_1)^T (e_1^T w_2)$

From $Lw = 0$ we got $d_1 w_1 = \frac{1}{c} e_1^T w_2$ and $d_m w_1 = -\frac{1}{c} e_m^T w_2$.

Therefore we got $c^2 (e_m^T w_2)^T d_m w_1 - c^2 (e_1^T w_2)^T d_1 w_1 + c^2 (d_m w_1)^T (e_m^T w_2) - c^2 (d_1 w_1)^T (e_1^T w_2) = -c \left((e_m^T w_2)^T \right)^2 - c (e_m^T w_2)^2 - c \left((e_1^T w_2)^T \right)^2 - c (e_1^T w_2)^2 \leq 0$

So from the BC we got $\frac{d}{dt} \| \begin{bmatrix} u \\ v \end{bmatrix} \|_{\overline{H}}^2 \leq w^T \begin{bmatrix} 0 & H \\ H^T & 0 \end{bmatrix} w - c^2 w^T \begin{bmatrix} 0 & M \\ M & 0 \end{bmatrix} w = w_1^T H w_2 + (H w_2)^T w_1 - c^2 w_1^T M w_2 - c^2 w_2^T M^T w_1 = (w_1^T H w_2) + (w_1^T H w_2)^T - (c^2 w_1^T M w_2) - (c^2 w_1^T M w_2)^T = 2 w_1^T (H - c^2 M) w_2 = 2 w_1^T (H - c^2 M) \frac{d}{dt}(w_1) = \frac{d}{dt} \|w_1\|_{H - c^2 M}^2$.
So we can move the terms left to the energy part. Therfore we showed that, the discretized case is stable for well posed BC.

Assignment 5

The below pictures were created with the attached MATLAB code $proj\_1D.m$. As asked in the assignment we used 4th order SBP operators and $k = 0.01, m = 101$.

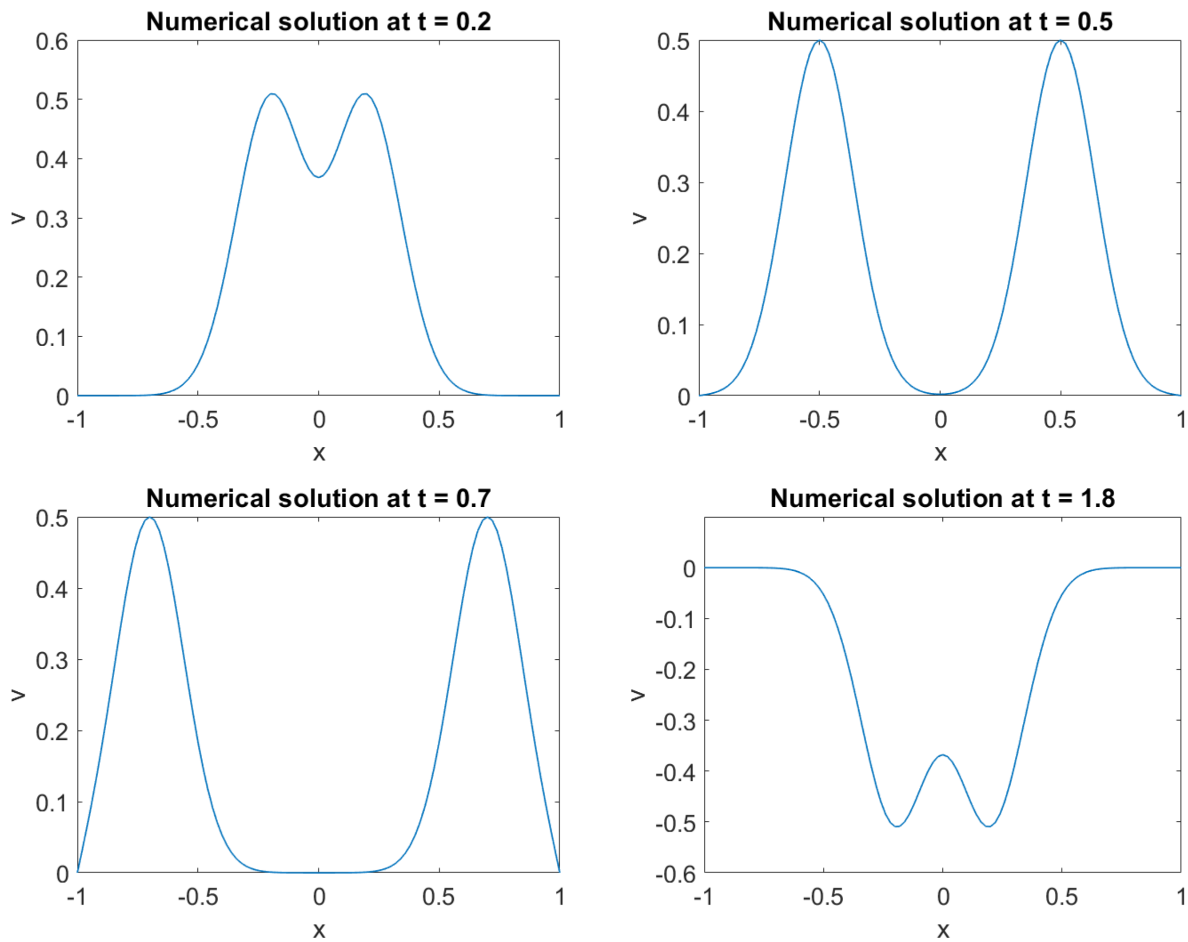

Figure 1: With Dirichlet BC ($u = 0$ $x = x_l, x = x_r$)

In Figure 1 we can see, that as expected from the theorethical part in the case, where BC are Dirichlet the energy is conservated, and that in the boundary the value is fixed in zero.

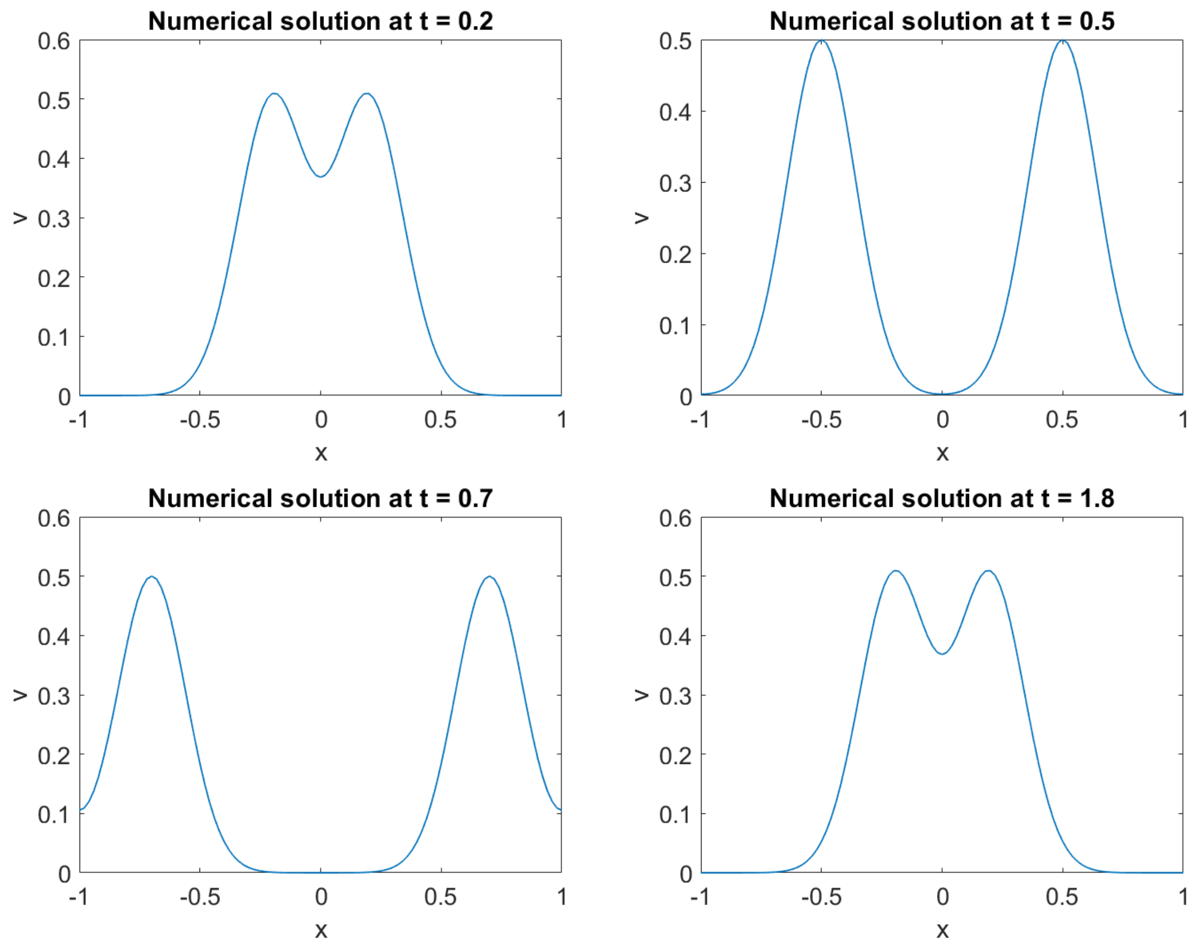

Figure 2: With Neuman BC ($u_x = 0$ $x = x_l, x = x_r$)

In Figure 2 we can see, that as expected from the theoretical part in the case, where BC are Nauman the energy also being conservated, but in this part the derivative in place is zero in the boundary, so the value can move.

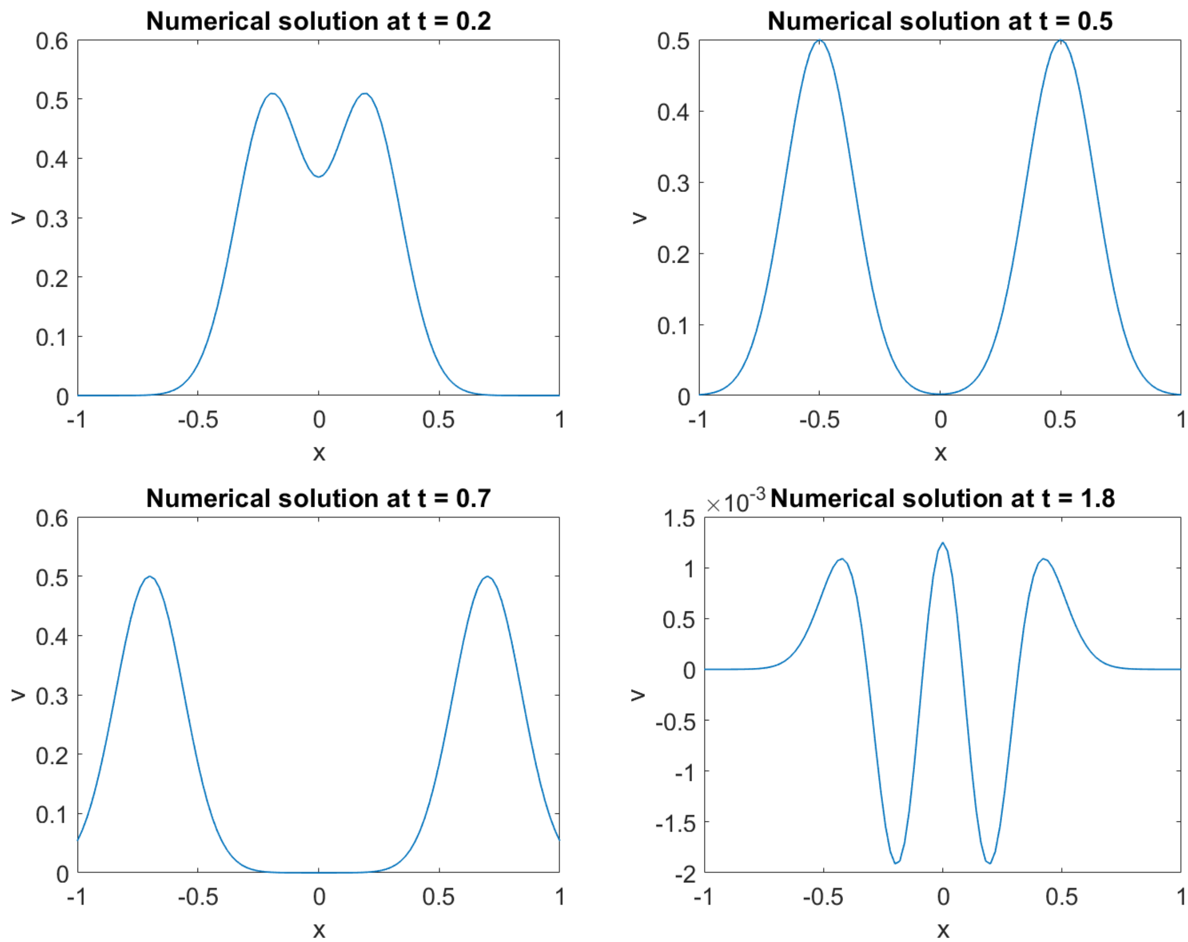

Figure 3: With absorbing BC

In Figure 3 we can see, that as expected from the theoretical part in the case, where BC are absorbing the energy converge to zero (the scale is changing on the figures).

Assignment 6

We implemented the convergent study in the code and will present our result in the below tables. Note that we use $c = 1$ during our simulations. And the analitic solution was given in the project description, so we could just use that and we shouldn't find it for ourself. We just had to implement the calculations in the given times with a specifed set of parameters.
We use $T = 1.8$, $k = 0.1 * h$. Here note that because we use time steps which has a ratio equivalent to $1/m - 1$, so in the case $m = 301$ due to the numerical error in the floating point approximation of 1/3 the end time won't actually be the given value, but in most case it won't cause any problem.

	m=51	m=101	m=201	m=301
order 2	0.019238	0.0048609	0.0012187	0.00054023
order 4	0.0008396	3.9973e-05	2.4964e-06	5.0445e-07
order 6	0.00017189	1.1205e-06	4.1719e-08	1.0852e-08

Table 1: Dirichlet boundary l2 errors for $T = 1.8$

	m=51	m=101	m=201	m=301
order 2	0.019203	0.0048606	0.0012187	0.00054023
order 4	0.00060918	4.8272e-05	2.9719e-06	5.7328e-07
order 6	0.00032086	2.7217e-06	1.1275e-08	1.0322e-08

Table 2: Neumann boundary l2 errors for $T = 1.8$

So, as we expected the numerical error decrease when we use higher order SBP approximation and when we use more grid points.

From the above data we can calculate the q convergence rate.

$$q = \frac{log_{10}(l2_1/l2_2)}{log_{10}(m_2/m_1)} \tag{9}$$

This calculation is correct, since per definition in the description

$$q = \frac{log_{10}(\frac{l2_1}{\sqrt{m_1}}/\frac{l2_2}{\sqrt{m_2}})}{log_{10}(m_2/m_1)} \tag{10}$$

Here $l2_i$ is the $norm(u - v) = l_2$ error for $m = m_i$ grid points.

We will only do these calculations for $m_1 = 51$ and $m_2 = 301$ for the Dirichlet condition and $m_1 = 101, m_2 = 201$ for Neuman. So in this way we will be able to compare the results for the different parameters. Otherwise, it's just the same calculation with the given data above.

	Order=2	Order=4	Order=6
Dirichlet	2.01141	3.92528	5.82694
Neumann	2.01018	4.05073	7.97232

Table 3: q value for Dirichlet $m_1 = 51, m_2 = 301$ and $t = 1.8$, q value for Neumann $m_1 = 101, m_2 = 201$ and $t = 1.8$

From the above table we can conclude that the q value is increasing as we increase the order of SBP operators. This is the expected behaviour. Also as expected in the two viewed case the increasement is similar.

This concludes our task on the numerical errors.

Assignment 7

The pictures below were created with the attached MATLAB codes *proj2D.m* and *proj2D_mixed_boundary.m*. We used 4th order SBP operators and $k = 0.005$ time step and $m = 101$ grid points.
As we can se in Figure 4 and 5, the case with symmetric BC results symmetric figures as expected. And in Figure 6 it's not centrally symmetrical due to the different kind of BC in different directions.

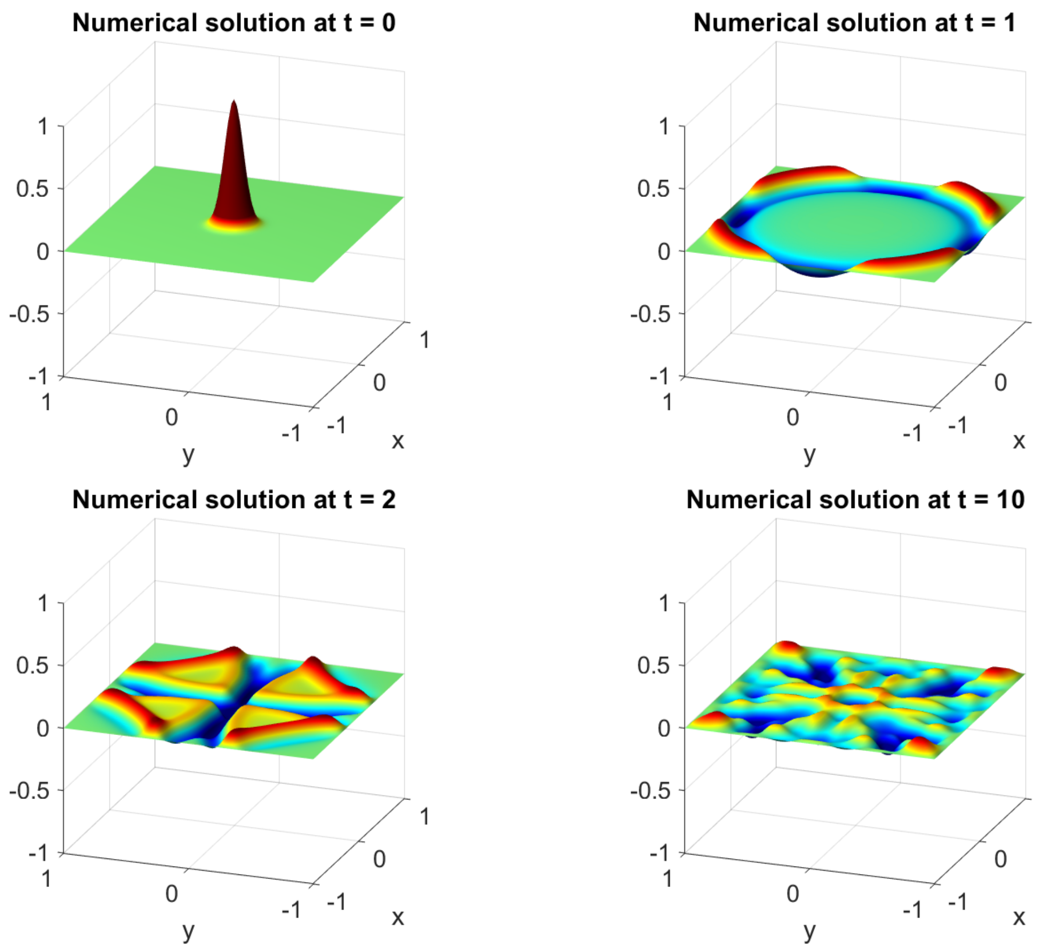

Figure 4: 2D with Dirichlet BC

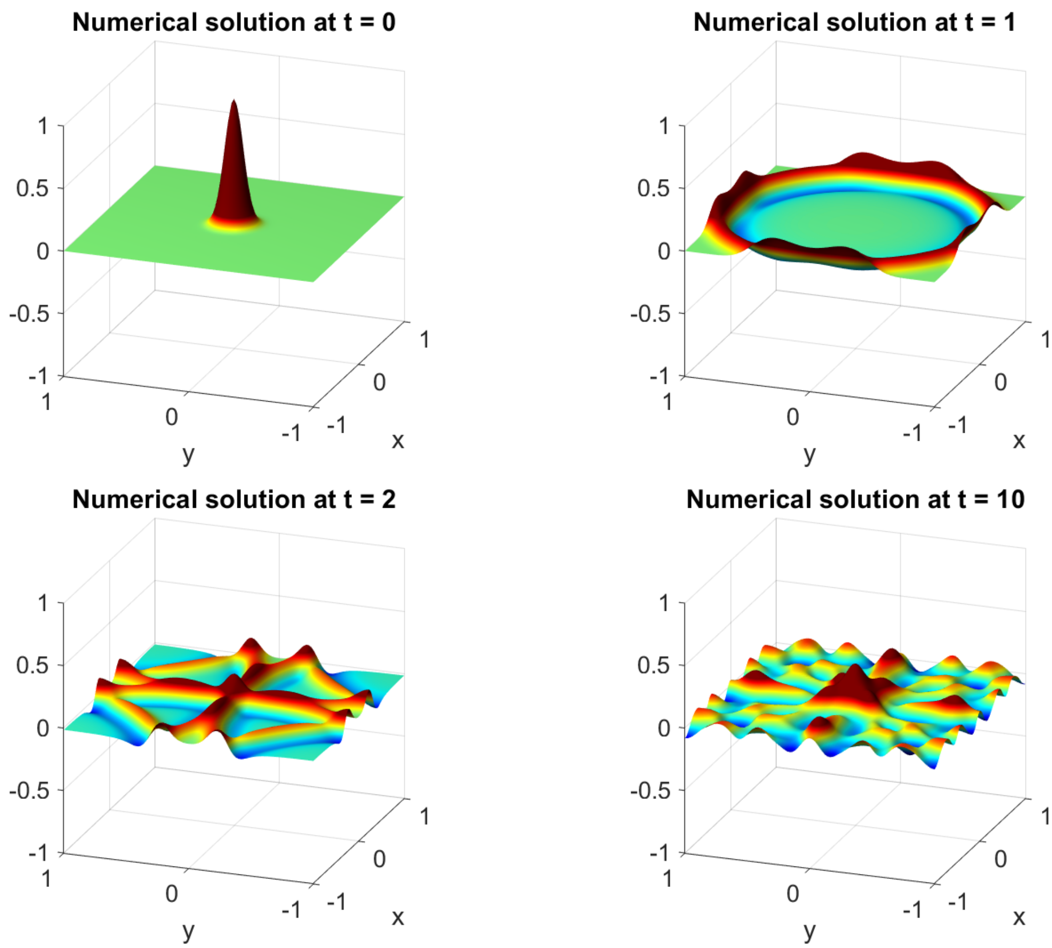

Figure 5: 2D with Neuman BC

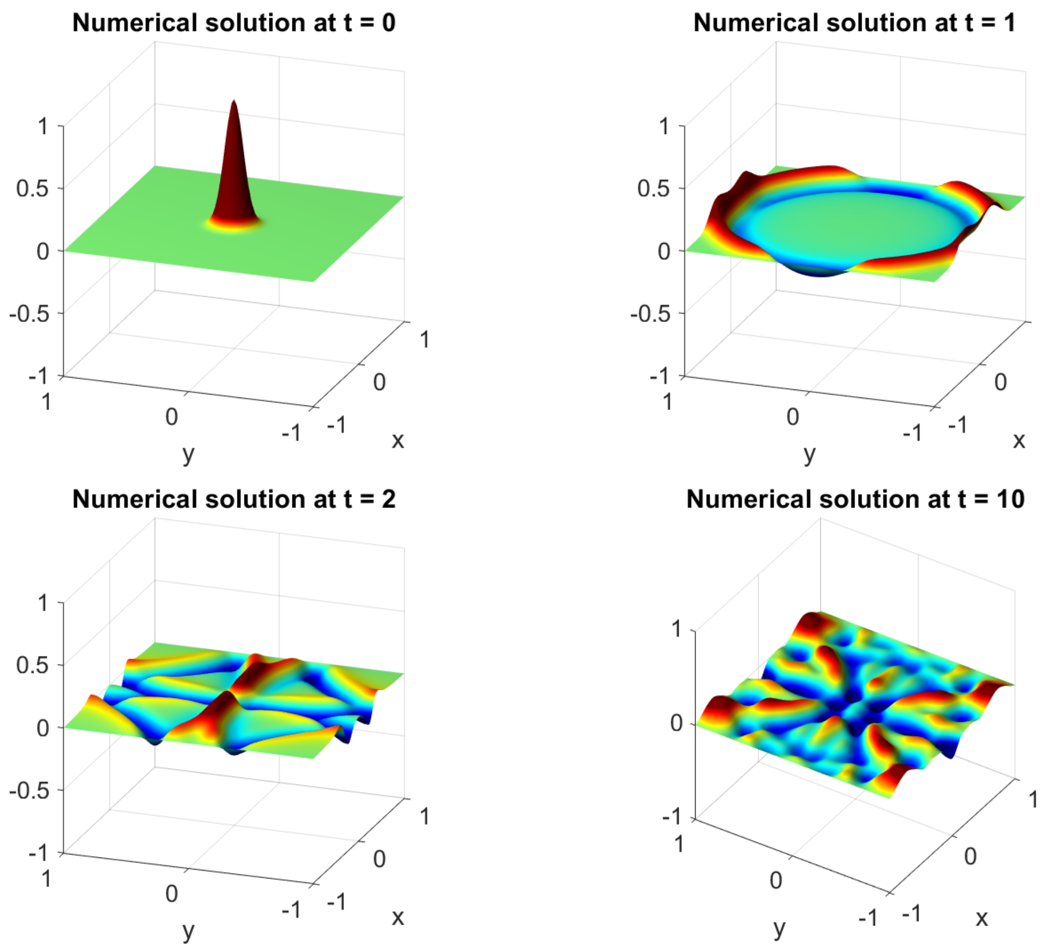

Figure 6: 2D with mixed BC, in axe x Dirichlet in the y axe Neuman BC is given

Conclusions

The project contained variety of methods which we learned in the lectures. Similar examples were presented for all the parts during lectures, but it wasn't repetitive, due to a new idea that was needed to complete the task due to the u_{tt} part. This part made the problem particularly challenging to solve (e.g. the discrete case in assignment 4).
The implementation part was mostly straightforward from the models given and the hardest part was to avoid making errors and finding them, because in scientific computation it's not so easy to find errors in your code (e.g. when you mix up indexes). But hopefully we managed to correct all our errors in the codes due to the effort of the group works.
We would recommend using this method for numerically modelling simpler 1 or 2 dimensional PDEs. But it doesn't seem to be an easily scalable approach, for more complicated and bigger problems the computing capacity fails (e.g. for m=301 the 2D case throws memory error even with sparse matrixes).