# Contents

# 1 Introduction

...

# 2 Comments on the errors

...

# 3 Derivation of Transparent Boundary Conditions for the linearized KdV equation and the dispersion equation

In [2], Transparent Boundary Conditions (TBCs) are derived for the one-dimensional continuous linearized KdV equation (or Airy equation) :

$$u_t + U_1 u_x + U_2 u_{xxx} = h(t,x), \quad t \in \mathbb{R}^+, \quad x \in \mathbb{R} \tag{1}$$

where $U_1 \in \mathbb{R}$, $U_2 \in \mathbb{R}_*^+$ and $h$ is a source term.

For the homogeneous initial boundary value problem

$$u_t + U_1 u_x + U2 u_{xxx} = h(t,x), \quad t \in \mathbb{R}^+, \quad x \in [a,b] \tag{2}$$
$$u(0,x) = u_0(x), \quad x \in [a,b] \tag{3}$$
$$+\text{boundary conditions}$$

the TBCs are given by

$$\begin{cases} u(t,a) - U_2 \mathcal{L}^{-1}\left(\frac{\lambda_1(s)^2}{s}\right) * u_x(t,a) - U_2 \mathcal{L}^{-1}\left(\frac{\lambda_1(s)}{s}\right) * u_{xx}(t,a) = 0 \\ u(t,b) - \mathcal{L}^{-1}\left(\frac{1}{\lambda_1(s)^2}\right) * u_{xx}(t,b) = 0 \\ u_x(t,b) - \mathcal{L}^{-1}\left(\frac{1}{\lambda_1(s)}\right) * u_{xx}(t,b) = 0 \end{cases} \tag{4}$$

where $\mathcal{L}^{-1}$ denotes the inverse Laplace transform, $s \in \mathbb{C}$ is the Laplace frequency and $\lambda_1$ is, among the three roots of the cubic characteristic equation obtained when solving (1) in the Laplace space, the only one with negative real part.

In this paper, we will focus on the special case $U_1 = 0, U_2 = 1$, which results in the KdV with only the dispersive part :

$$u_t + u_{xxx} = 0 \tag{5}$$

In this case, also accordingly to [2], the only root with negative real part is

$$\lambda(s) = \lambda_1(s) = -\sqrt[3]{s} \tag{6}$$

# 4 Approximations for the Transparent Boundary Conditions for the dispersion equation

The computation of the TBCs (4) is not simple due to the inverse Laplace transform that, which makes these conditions to be nonlocal in time. Therefore, we will propose approximations of the root (6) that avoid integrations in time, making the TBCs considerably simpler.

Obviously, as we can see through the results shown in this section, the approximate boundary conditions are not so precise as the ones proposed by [2] (who derives TBCs derived for the discrete linearized KdV equation). Nevertheless, the objectives of our work and the work of [2] are very different : while they seek to minimize the error of the computed solution (compared to the analytical one) due to the boundary conditions, we want here to apply our approximate TBCs as Coupling Boundary Conditions (CBCs) in a Domain Decomposition Method (DDM). Therefore, our objective lays on the convergence of the DDM to the solution of the same problem in the monodomain, independetly of the errors on the external boundaries.

For deriving our approximations, we firstly notice that we can rewrite (4) as

$$\begin{cases} u(t,a) - U_2\mathcal{L}^{-1}\left(\frac{\lambda_1(s)^2}{s}\hat{u}_x(t,s)\right) - U_2\mathcal{L}^{-1}\left(\frac{\lambda_1(s)}{s}\hat{u}_{xx}(t,s)\right) = 0 \\ u(t,b) - \mathcal{L}^{-1}\left(\frac{1}{\lambda_1(s)^2}\hat{u}_{xx}(t,s)\right) = 0 \\ u_x(t,b) - \mathcal{L}^{-1}\left(\frac{1}{\lambda_1(s)}\hat{u}_{xx}(t,s)\right) = 0 \end{cases} \tag{7}$$

where $\hat{u}(s,x) = \mathcal{L}u(t,x)$ is the Laplace transform in $t$ of $u$. Moreover, the inverse Laplace transform satisfies the following properties [3]:

- Linearity :

$$\mathcal{L}^{-1}\left[a_1\hat{u}_1(s,x) + a_2\hat{u}_2(s,x)\right] = a_1u_1(t,x) + a_2u_2(t,x) \tag{8}$$

- First Derivative :

$$\mathcal{L}^{-1}\left[s\hat{u}(s,x)\right] = u_t(t,x) + \mathcal{L}^{-1}\left[u(0,x)\right] = u_t(s,t) + u(0,x)\delta(t) \tag{9}$$

- Second Derivative :

$$\begin{aligned}\mathcal{L}^{-1}\left[s^2\hat{u}(s,x)\right] =& u_{tt}(t,x) + \mathcal{L}^{-1}\left[su(0,x) + u_t(0,x)\right] = \\ & u_{tt}(s,t) + u(0,x)\delta_t(t) + u_t(0,x)\delta(t)\end{aligned} \tag{10}$$

- Convolution :

$$\mathcal{L}^{-1}\left[\hat{u}_1(s,x)\hat{u}_2(s,x)\right] = \mathcal{L}^{-1}\left[\hat{u}_1(s,x)\right] * \mathcal{L}^{-1}\left[\hat{u}_2(s,x)\right] \tag{11}$$

where $\delta(t)$ is the Dirac delta function and $*$ denotes the convolution operator.

The properties (8) to (11) motivate us to approximate the operands of the inverse Laplace transforms in (4) by polynomials in $s$. We will initially use a constant polynomial for this approximation. Before that, we remark the following useful relations between the mentioned operands, as a consequence of (6)

## 4.1 Approximation of the TBCs using a constant polynomial

We will use the constant polynomial $P_0(s) = c$ for approximating $\frac{\lambda^2}{s}$. Moreover, as a consequence of (6), we can approximate the other operands of the inverse Laplace transforms in (4) only in function of $c$ :

$$\frac{\lambda^2}{s} = c \qquad \frac{\lambda}{s} = -c^2 \qquad \frac{1}{\lambda_1(s)^2} = c^2 \qquad \frac{1}{\lambda_1(s)} = -c \tag{12}$$

Replacing (12) in (4) and considering possibly different polynomial approximations for the left and the right boundaries (respectively with the coefficients $c_L$ and $c_R$), we get the approximate Transparent Boundary Conditions :

$$\begin{cases} u(t,a) - cu_x(t,a) + c^2u_{xx}(t,x) = 0 \\ u(t,b) - c^2u_{xx}(t,x) = 0 \\ u_x(t,b) + cu_{xx}(t,x) = 0 \end{cases} \tag{13}$$

Considering a discrete domain with mesh size $\Delta x$ and points $x_0, ..., x_N$ and using finite difference approximations, the approximate TBCs (13) are discretized as

$$\begin{cases} u_0 - c_L \frac{u_1 - u_0}{\Delta x} + c_L^2 \frac{u_0 - 2u_1 + u_2}{\Delta x^2} = 0 \\ u_N - c_R^2 \frac{u_N - 2u_{N-1} + u_{N-2}}{\Delta x^2} = 0 \\ \frac{u_N - u_{N-1}}{\Delta x} + c_R^2 \frac{u_N - 2u_{N-1} + u_{N-2}}{\Delta x^2} = 0 \end{cases} \tag{14}$$

### 4.1.1 Initial numerical experiments

In order to validate our approximation, observe its general behavior when varying the constant approximations $c_L$ and $c_R$ and compare our results with the ones obtained by [2], we will solve the same numerical test presented in this paper. This problem, originally treated by [5] for the study of numerical solutions for the KdV equation, reads

$$u_t + u_{xxx} = 0, \quad x \in \mathbb{R} \tag{15}$$

$$u(0, x) = e^{-x^2}, \quad x \in \mathbb{R} \tag{16}$$

$$u \to 0, \quad |x| \to \infty \tag{17}$$

The fundamental solution of (15) and the exact solution for the problem (15) - (17) are, respectively,

$$E(t, x) = \frac{1}{\sqrt[3]{3t}} Ai\left(\frac{x}{\sqrt[3]{3t}}\right) \tag{18}$$

$$u_{exact}(t, x) = E(t, x) * e^{-x^2} \tag{19}$$

where $Ai$ is the Airy function.

As done by [5] and [2], the problem will be solved in the spatial domain $[-6, -6]$, for $0 \leq t \leq T_{max}$, with $T_{max} = 4$. The mesh size is $\Delta x = 12/500 = 0.024$ and, as in [2], the time step is $\Delta t = 4/2560 = 0.0015625$.

For a quantitative evaluation of the results, we will calculate the same errors defined in the paper of [2]. For each instant $t_n = n\Delta t$, we compute the relative $l^2-$error

$$e^n = \frac{\left\| u^n_{exact} - u^n_{computed} \right\|_2}{\left\| u^n_{exact} \right\|_2}$$

and, in the whole time interval, the maximum error $e^n$ and the $l^2$ norm of $e^n$, given respectively by :

$$e_{Tm} = \max_{0 < n < T_{max}} (e^n)$$

$$e_{L2} = \sqrt{\Delta t \sum_{n=1}^{T_{max}} (e^n)^2}$$

In order to verify the influence of $c_L$ and $c_R$ on the computed solutions (and possibly identify a range of values that better approximate the TBCs), we made several tests with all the possible pairs $c_L, c_R \in \{-10, -1, -0.1, 0, 0.1, 1, 10\}^2$. The results were classified accordingly to their errors $e_{L2}$ (a criteria based on the error $e_{Tm}$ gives a similar result). The figure 1 shows, for some instants, a comparison between the best, the worst and the exact solution. For naming the worst result, we did not considered the ones in which the numerical solution diverged (following the arbitrary criteria $e_{L2} > 10$).

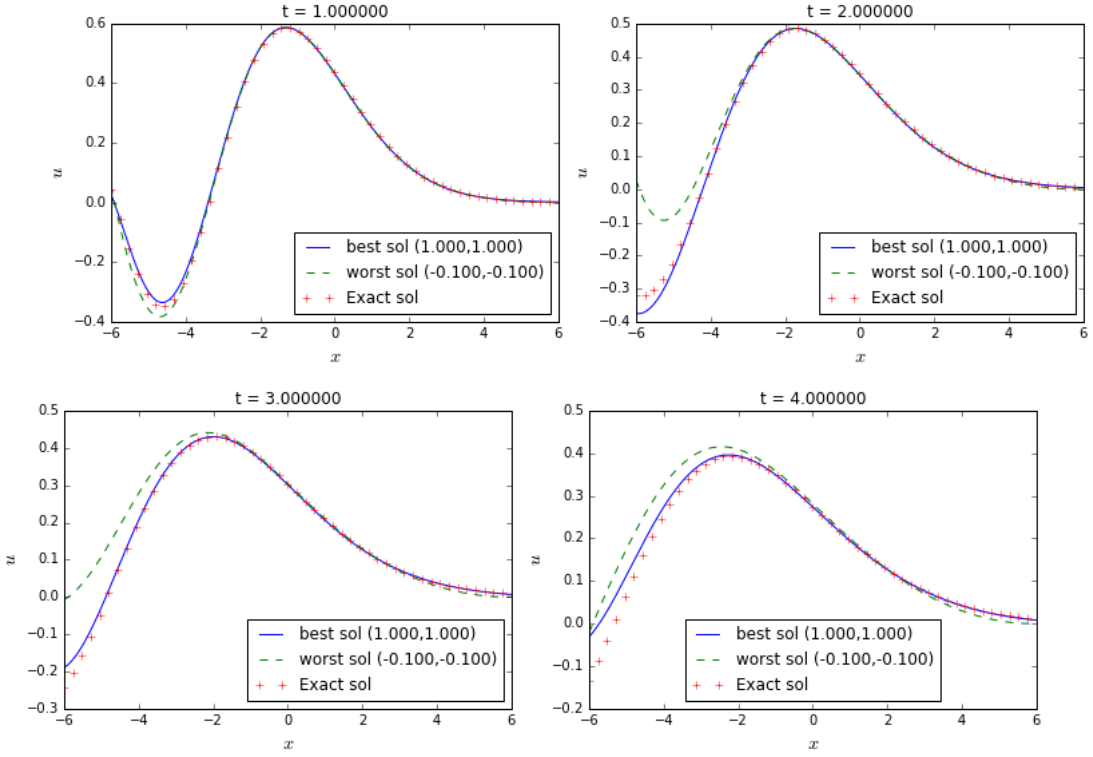



Figure 1: Best and worst solution compared with analytical solution, for the constant polynomial approximation

The table 1 presents the ten tests that presented the smallest $e_{L2}$ :

| $c_L$ | $c_R$ | $e_{L2}$ |
|---|---|---|
| 1.0 | 1.0 | 0.0947 |
| 1.0 | 10.0 | 0.0973 |
| 1.0 | 0.1 | 0.0984 |
| 1.0 | 0.0 | 0.0992 |
| 1.0 | -10.0 | 0.0994 |
| 1.0 | -0.1 | 0.1000 |
| 1.0 | -1. | 0.1016 |
| 10.0 | 1.0 | 0.3470 |
| 10.0 | 0.1 | 0.3474 |
| 10.0 | 0.0 | 0.3475 |

Table 1: Best results (smallest $e_{L2}$) for the constant polynomial approximation

We notice that the results are much sensitive to the coefficient on the left boundary : for a fixed $c_L$, the error is very similar for every $c_R$. This is a consequence of the fact that the solution of this specific problem is practically constant and equal to zero near $x = 6$, but presents strong variations in time near $x = -6$. The best result, as shown in the figure 1, is able to impose this condition on the right boundary, what the worst solution is not able to do. In the case of the left boundary, in despite of a more evident error, the best solutions follows quite well the behavior of the exact solution.

## 4.2 Approximation of the TBCs using a linear polynomial

In a similar way as done above, we approximate $\frac{\lambda^2}{s}$ by $P_1(s) = ds + c$. Then, using relations similar to (12), the inverse Laplace transforms in (7) reads :

$$
\begin{aligned}
\mathcal{L}^{-1}\left(\frac{\lambda^2}{s}\hat{u}_x(s,x)\right) =& \mathcal{L}^{-1}\left[(ds+c)\hat{u}_x(t,s)\right] = \\
& d\mathcal{L}^{-1}\left[\hat{u}_{xt}(t,s)\right] + c\mathcal{L}^{-1}\left[\hat{u}_x(t,s)\right] = \\
& du_{xt}(x,t) + du_x(0,x)\delta(t) + cu_x(x,t) \\
\mathcal{L}^{-1}\left(\frac{\lambda}{s}\hat{u}_{xx}(s,x)\right) =& \mathcal{L}^{-1}\left[-(ds+c)^2\hat{u}_{xx}(t,s)\right] = \\
& -d^2u_{xxtt}(x,t) - d^2\delta_t(t)u_xx(0,x) - d^2\delta(t)u_{xxt}(0,x) + \\
& -2dcu_{xxt}(x,t) - 2dcu_xx(0,x)\delta(t) - c^2u_{xx}(x,t) \\
\mathcal{L}^{-1}\left(\frac{1}{\lambda^2}\hat{u}_{xx}(s,x)\right) =& \mathcal{L}^{-1}\left[(ds+c)^2\hat{u}_{xx}(t,s)\right] = \\
& d^2u_{xxtt}(x,t) + d^2\delta_t(t)u_xx(0,x) + d^2\delta(t)u_{xxt}(0,x) + \\
& 2dcu_{xxt}(x,t) + 2dcu_xx(0,x)\delta(t) + c^2u_{xx}(x,t) \\
\mathcal{L}^{-1}\left(\frac{1}{\lambda}\hat{u}_{xx}(s,x)\right) =& \mathcal{L}^{-1}\left[-(ds+c)\hat{u}_{xx}(t,s)\right] = \\
& -du_{xxt}(x,t) - du_{xx}(0,x)\delta(t) - cu_{xx}(x,t)
\end{aligned}
\tag{20}
$$

Using forward first order forward finite difference approximations for the first derivative in time, second order centered for the second derivative in time, and considering that the initial solution and its derivatives are equal to zero on the boundaries (which is the case of the test case treated here), we obtain the following discretized approximate TBCs :

$$
\begin{aligned}
& u_0^{n+1} - \left(\frac{d_L}{\Delta t} + c_L\right)\left(\frac{u_1^{n+1} - u_0^{n+1}}{\Delta x}\right) + \left(\frac{d_L^2}{\Delta t^2} + \frac{2d_Lc_L}{\Delta t} + c_L^2\right)\left(\frac{u_0^{n+1} - 2u_1^{n+1} + u_2^{n+1}}{\Delta x^2}\right) = \\
& -\frac{d_L}{\Delta t}\left(\frac{u_1^n - u_0^n}{\Delta x}\right) + \left(2\frac{d_L^2}{\Delta t^2} + \frac{2d_Lc_L}{\Delta t}\right)\left(\frac{u_0^n - 2u_1^n + u_2^n}{\Delta x^2}\right) - \frac{d_L^2}{\Delta t^2}\left(\frac{u_0^{n-1} - 2u_1^{n-1} + u_2^{n-1}}{\Delta x^2}\right)
\end{aligned}
\tag{21}
$$

$$
\begin{aligned}
& u_N^{n+1} - \left(\frac{d_R^2}{\Delta t^2} + \frac{2d_Rc_R}{\Delta t} + c_R^2\right)\left(\frac{u_N^{n+1} - 2u_{N-1}^{n+1} + u_{N-2}^{n+1}}{\Delta x^2}\right) = \\
& -\left(2\frac{d_R^2}{\Delta t^2} + \frac{2d_Rc_R}{\Delta t}\right)\left(\frac{u_N^n - 2u_{N-1}^n + u_{N-2}^n}{\Delta x^2}\right) + \frac{d_R^2}{\Delta t^2}\left(\frac{u_N^{n-1} - 2u_{N-1}^{n-1} + u_{N-2}^{n-1}}{\Delta x^2}\right)
\end{aligned}
\tag{22}
$$

$$\frac{u_N^{n+1} - u_{N-1}^{n+1}}{\Delta x} + \left(\frac{d_R}{\Delta t} + c_R\right)\left(\frac{u_N^{n+1} - 2u_{N-1}^{n+1} + u_{N-2}^{n+1}}{\Delta x^2}\right) = \frac{d_R}{\Delta t}\left(\frac{u_N^{n} - 2u_{N-1}^{n} + u_{N-2}^{n}}{\Delta x^2}\right) \tag{23}$$

### 4.2.1 Initial numerical experiments

We repeated the numerical tests made for the approximation with $P_0$, making the coefficients $c_L$ and $d_L$ assume the values in $\{-10, -1, -0.1, 0, 0.1, 1, 10\}$. In order to avoid a too high computation, and also taking in account the remark we made above about the weak dependence of the results on the coefficients for the right boundary, all the tests were done assuming $c_R = c_L$ and $d_R = d_L$.

We present in the table 2 the ten best results. As we can see, the best result is the one where $d_L = d_R = 0$ and $c_L = c_R = 1.$, which corresponds to the best results of the approximations using constant polynomials.

| $d_L = d_R$ | $c_L = c_R$ | $e_{L2}$ |
|---|---|---|
| 0. | 1.0 | 0.0947 |
| 0.1 | 1.0 | 0.1234 |
| 1.0 | 1.0 | 0.2003 |
| 10.0 | 0.1 | 0.2204 |
| -10.0 | 0.1 | 0.2398 |
| 10.0 | 1.0 | 0.2716 |
| -10.0 | 0.0 | 0.2480 |
| -10.0 | 1.0 | 0.3004 |
| 10.0 | 0.0 | 0.2721 |
| 0.0 | 0.1 | 0.3674 |

Table 2: Best results (smallest $e_{L2}$) for the linear polynomial approximation

## 4.3 Partial conclusion

It must be clear that our approach is not better than the one proposed by [2] what, in fact, is not the objective of the work developed here. Indeed, [2] derives TBCs for two discrete schemes, and the worst result among them, using the same $\Delta x$ and $\Delta t$ that we used here, presents an error $e_{L2} \approx 0.005$ for $t = 4$, while our best result has $e_{L2} \approx 0.1$ for the same instant.

However, we can say that the boundary conditions proposed here work relatively well as TBCs, with a very simple implementation and, consequently, low computation and memory costs. Moreover, as shown in the results of

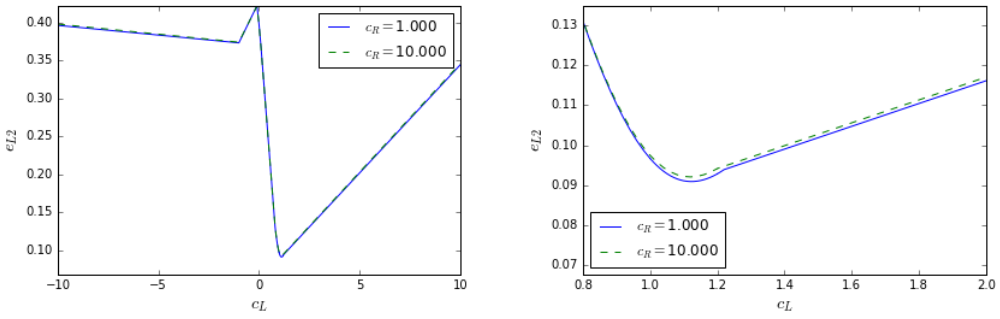
this section, the approximation using a linear polynomial, although the increment in the complexity (including time derivative terms up to the second derivative, what requires the storage of previous computed solutions), does not provide a better approximation for the TBC. Therefore, in the following, we will use only the linear approximation, which will be denoted as

$$\begin{cases} \Theta_1(u,x)_L^c = u(t,x) - c_L u_x(t,x) + c_L^2 u_{xx}(t,x) \\ \Theta_2(u,x)_R^c = u(t,x) - c_R^2 u_{xx}(t,x) \\ \Theta_3(u,x)_R^c = u_x(t,b) + c_R u_{xx}(t,s) \end{cases}$$

## 4.4 Optimization of the approximate TBCs (minimization of the errors compared to the analytical solution)

In this subsection, we investigate deeply the influence of this approximation over the error of the computed solution, compared with the analytical one. For this purpose, we repeat the computations, but with a much more refined range of coefficients, always using as criteria of quality the error $e_{L2}$ of each test.

As a consequence of the remarks made in the last subsection, this refinement is made higher for the coefficient $c_L$. By making a gradual study, in which we identified at each step the intervals of $c_L$ which give the best results and studied it even more deeply (up to a step 0.01 for the variation of $c_L$), we were able to construct the curves presented in the figure 2 and identify the empirical optimum $c_L = 1.16$ :




(a) General view of all the tested coefficients

(b) Detail for $c_L \in [0.8, 2.0]$

Figure 2: Error of the numerical solution compared to the analytical solution as function of the constant polynomial approximation for the TBC

# 5 Application to a Domain Decomposition Method

The discrete approximations (14) for the Transparent Boundary Conditions for the equation (5) will be applied as an Interface Boundary Conditions (IBC) in a Domain Decomposition Method (DDM). Firstly, following [4], we will briefly describe the DDM that we will consider here, and after we will describe and test the incorporation of the proposed IBCs.

## 5.1 The Schwarz Method

Domain Decomposition Methods allow to decompose a domain $\Omega$ in multiple subdomains $\Omega_i$ (that can possibly overlap) and solve the problem in each one of them. Therefore, one must find functions that satisfies the PDE in each subdomain and that match on the interfaces.

The first DDM developed was the Schwarz method, which consists on an iterative method : in the case of a evolution problem, the solution $u_i^{n,\infty}$, in each time step $t_n$ and each subdomain $\Omega_i$, is computed as the convergence of the solution obtained in each iteration, $u_i^{n,k}$, $\quad k \geq 0$.

We will consider here the additive Schwarz method (ASM), in which the Interface Boundary Conditions are always constructed using the solution $u_j^{n,k-1}$, $\quad j \neq i$ of the previous iteration in the other neighbors subdomains. Therefore, in each interface between the subdomains $\Omega_i$ and $\Omega_j$, the boundary condition for the problem in $\Omega_i$ is

$$\mathcal{B}_i(u_i^{n,k+1}) = \mathcal{B}_i(u_j^{n,k})$$

where $\mathcal{B}$ denotes the operator of the IBC.

Without loss of generality, in the following we will consider a domain $\Omega$ decomposed in two non-overlapping subdomains, $\Omega_1$ and $\Omega_2$, with $\Gamma = \Omega_1 \bigcap \Omega_2$.

When implementing a Schwarz methods, one must define appropriate operators $\mathcal{B}$ such that :

- The solution $u_i$ in each subdomain $\Omega_i$ converges to $u|_{\Omega_i}$, i.e, the solution $u$ of the monodomain $\Omega$ restricted to $\Omega_i$;
- The method shows a fast convergence.

In fact, accordingly to [**?**], the optimal additive Schwarz method for solving the problem

$$\mathcal{A}(u) = f \ \text{ in } \ \Omega$$
$$u = 0 \ \text{ on } \ \partial\Omega$$

where $\mathcal{A}$ is a partial differential operator, is the one which uses as Interface Boundary Conditions the exact Transparent Boundary Conditions, given by

$$B_i(u) = \frac{\partial}{\partial n_i} u + D2N(u)$$

where $\partial n_i$ is the outward normal to $\Omega_i$ on $\Gamma$ , and the D2N (Dirichlet to Neumann) operator is defined by

$$D2N : \alpha(x) \mapsto \frac{\partial}{\partial n_i^c} v \Big|_{\Gamma}$$

with $\alpha$ defined on $\Gamma$. $v$ is solution of the following problem, solved in the complementary set of $\Omega_i$, denoted by $\Omega_i^c$

$$\mathcal{A}(v) = f \ \text{ in } \ \Omega_i^c$$
$$v = 0 \ \text{ on } \ \delta\Omega_i \backslash \Gamma$$
$$v = \alpha \ \text{ on } \ \Gamma$$

The ASM using such exact TBCs is optimal in the sense that it converges in two iterations, and no other ASM can converge faster. Nevertheless, these TBC, in general, are not simple to compute both analtically and numerically. More specifically, they are nonlocal in time, so they must be approximated for an efficient numerical implementation [1]. It is in this context that we propose the implementation of our approximate TBCs as Interface Boundary Conditions for the ASM.

## 5.2 ASM with the approximate TBCs for the dispersive equation

The resolution of the dispersive equation (5) with the Additive Schwarz method, using the constant polynomial approximation for the TBCs, is written as

$$\begin{cases} (u_1^{n,k+1})_t + (u_1^{n,k+1})_{xxx} = 0, \quad x \in \Omega_1, \quad t \geq 0 \\ u_1^{n,0} = u_1^{n-1,\infty}, \quad x \in \Omega_1 \\ \Upsilon_1^{c_L}(u_1^{n+1,k+1}, -L) = 0, \\ \Theta_2^{c_R}(u_1^{n+1,k+1}, 0) = \Theta_2^{c_R}(u_2^{n,k}, 0), \\ \Theta_3^{c_R}(u_1^{n+1,k+1}, 0) = \Theta_3^{c_R}(u_2^{n,k}, 0) \end{cases} \tag{24}$$

$$\begin{cases} (u_2^{n,k+1})_t + (u_2^{n,k+1})_{xxx} = 0, \quad x \in \Omega_2, \quad t \geq 0 \\ u_2^{n,0} = u_2^{n-1,\infty}, \quad x \in \Omega_2 \\ \Theta_1^{c_L}(u_2^{n+1,k+1}, 0) = \Theta_1^{c_L}(u_1^{n,k}, 0) \\ \Upsilon_2^{c_R}(u_2^{n+1,k+1}, L) = 0 \\ \Upsilon_3^{c_R}(u_2^{n+1,k+1}, L) = 0 \end{cases} \tag{25}$$

where $\Upsilon_i, \quad i = 1, 2, 3$, are the external boundary conditions (i.e, $\partial\Omega_i \backslash \Gamma$). These external BCs are independent of the interface BCs. Here, we will consider $\Upsilon_1 = \Theta_1^{c_L=1.0}$, $\Upsilon_2 = \Theta_2^{c_R=0.0}$ and $\Upsilon_3 = \Theta_3^{c_R=0.0}$, which gives

$$\Upsilon_1(u, x) = u - u_x + u_{xx} = 0 \tag{26}$$

$$\Upsilon_2(u, x) = u = 0 \tag{27}$$

$$\Upsilon_3(u, x) = u_x = 0 \tag{28}$$

$$\tag{29}$$

This choice was made based on the easy implementation and the good results provided by the coefficients $c_L = 1.0$ and $c_R = 0.0$ in approximating the analytical solution in $\Omega$ (as shown in the table 1). Nevertheless, it does not have much importance in the study that we will done here. In fact, our purpose is to study exclusively the behavior of the DDM implemented here; therefore, all the results must be compared to a referential solution $u_{ref}$, that can be simply the numerical solution of the monodomain problem. The only restriction for an appropriate study is that the external BCs for computing $u_{ref}$ must be the same $\Upsilon_i, \quad i = 1, 2, 3$ used for each subdomain in the DDM.

**Remarks on the notation** From here to the end of this paper, we will focus exclusively on the error produced by the DDM, compared to the referential solution, independently of the other possible components of the error compared to the analytical solution (external boundary conditions, error accumulation over the time steps, for example). Therefore, all the following study will be made considering the execution of the method over only one time step, allowing us to suppress the index denoting the instant $t_n$ and use

a clearer notation for the solution : $u_j^i$, where $i$ indicates the subdomain $\Omega_i$ (or, in the case of the reference solution, $i = ref$, and in the convergence of the method, $i = *$) and $j$ indicates the spatial discrete position. In the cases where the iterative process is take in account, we will add the superscript $k$ to indicate the iteration.

Concerning the spatial discretization, the monodomain $\Omega$ will be divided in $2N + 1$ homogeneously distributed points, numbered from 0 to $2N$. In all the analytical description, we will consider that the two subdomains $\Omega_1$ and $\Omega_2$ have the same number of points, respectively $x_0, ..., x_N$ and $x_N, ..., x_{2N}$. The interface point $x_N$ is common to the two domains, having different computed solutions $u_N^1$ and $u_N^2$ in each one of them. Evidently, we expect, at the convergence of the ASM, that $u_N^1 = u_N^2 = u_N^*$

## 5.3 Error in the converged solution

When using approximate TBCs in the ASM, one should guarantee that the converged solutions $u_1, u_2$ satisfies the same equation as the solution $u_{ref}$ of the monodomain problem. In the following paragraphs, we show that this property is not verified by the method (24) - (25) proposed here. Based on that, we will be able to propose corrections for it.

For the interior points of each one of the domains, we will consider a second order spatial discretization of the equation (5).

$$\frac{u_j^i - \alpha_j^i}{\Delta t} + \frac{-\frac{1}{2}u_{j-2}^i + u_{j-1}^i - u_{j+1}^i + \frac{1}{2}u_{j+2}^i}{\Delta x^3} = 0 \tag{30}$$

for $j = 2, ..., N-2$ in the case $i = 1$; for $j = N+2, ..., 2N-2$ in the case $i = 2$; and for $j = 2, ..., 2N-2$ in the case $i = ref$. In the above expression, $\alpha_j^i$ is a given data (for example, the converged solution in the previous time step).

For the points near the boundaries, we use second order uncentered discretizations or the appropriate TBCs. For example, for the point $x_0$ :

$$\frac{u_0^2 - \alpha_0^2}{\Delta t} + \frac{-\frac{5}{2}u_0^2 + 9u_1^2 - 12u_2^2 + 7\frac{1}{2}u_3^2 - \frac{3}{2}u_4^2}{\Delta x^3} = 0 \tag{31}$$

and similarly to the other points near the boundaries.

In the resolution of the problem in $\Omega_1$, two interface boundary conditions are imposed (corresponding to $\Theta_2$ and $\Theta_3$), so the discrete equations for the points $x_{N-1}$ and $x_N$ are written as

$$\Theta_2^{c_R}(u_N^1) = \Theta_2^{c_R}(u_N^2) \implies$$
$$\implies u_N^1 - c_R^2 \frac{u_N^1 - 2u_{N-1}^1 + u_{N-2}^1}{\Delta x^2} = u_N^2 - c_R^2 \frac{u_N^2 - 2u_{N+1}^2 + u_{N+2}^2}{\Delta x^2} \tag{32}$$

$$\Theta_3^{c_R}(u_N^1) = \Theta_3^{c_R}(u_N^2) \implies$$
$$\implies \frac{u_N^1 - u_{N-1}^1}{\Delta x} + c_R \frac{u_N^1 - 2u_{N-1}^1 + u_{N-2}^1}{\Delta x^2} = \frac{u_{N+1}^2 - u_N^2}{\Delta x} + c_R \frac{u_N^2 - 2u_{N+1}^2 + u_{N+2}^2}{\Delta x^2} \tag{33}$$

In the resolution of the problem in $\Omega_2$, only one interface boundary condition is used (corresponding to $\Theta_1$) :

$$\begin{aligned} &\Theta_1^{c_L}(u_N^2) = \Theta_1^{c_L}(u_N^1) \implies \\ \implies & u_N^2 - c_L \frac{u_{N+1}^2 - u_N^2}{\Delta x} + c_L^2 \frac{u_N^2 - 2u_{N+1}^2 + u_{N+2}^2}{\Delta x^2} = \\ & u_N^1 - c_L \frac{u_N^1 - u_{N-1}^1}{\Delta x} + c_L^2 \frac{u_N^1 - 2u_{N-1}^1 + u_{N-2}^1}{\Delta x^2} \end{aligned} \tag{34}$$

In the convergence, the expressions (32) to (34) gives respectively

$$\begin{aligned} & u_N^* - c_R^2 \frac{u_N^* - 2u_{N-1}^* + u_{N-2}^*}{\Delta x^2} = u_N^* - c_R^2 \frac{u_N^* - 2u_{N+1}^* + u_{N+2}^*}{\Delta x^2} \implies \\ \implies & 2c_R^2 \frac{-\frac{1}{2}u_{N-2}^* + u_{N-1}^* - u_{N+1}^* + \frac{1}{2}u_{N+2}^*}{\Delta x^2} = 0 \end{aligned} \tag{35}$$

$$\begin{aligned} & \frac{u_N^* - u_{N-1}^*}{\Delta x} + c_R \frac{u_N^* - 2u_{N-1}^* + u_{N-2}^*}{\Delta x^2} = \\ & \frac{u_{N+1}^* - u_N^*}{\Delta x} + c_R \frac{u_N^* - 2u_{N+1}^* + u_{N+2}^*}{\Delta x^2} \implies \\ \implies & - \frac{u_{N-1}^* - 2u_N^* + u_{N+1}^*}{\Delta x} - 2c_R \frac{-\frac{1}{2}u_{N-2}^* + u_{N-1}^* - u_{N+1}^* + \frac{1}{2}u_{N+2}^*}{\Delta x^2} = 0 \end{aligned} \tag{36}$$

$$\begin{aligned} & u_N^* - c_L \frac{u_{N+1}^* - u_N^*}{\Delta x} + c_L^2 \frac{u_N^* - 2u_{N+1}^* + u_{N+2}^*}{\Delta x^2} = \\ & u_N^* - c_L \frac{u_N^* - u_{N-1}^*}{\Delta x} + c_L^2 \frac{u_N^* - 2u_{N-1}^* + u_{N-2}^*}{\Delta x^2} \implies \\ \implies & - c_L \frac{u_{N-1}^* - 2u_N^* + u_{N+1}^*}{\Delta x} + 2c_L^2 \frac{-\frac{1}{2}u_{N-2}^* + u_{N-1}^* - u_{N+1}^* + \frac{1}{2}u_{N+2}^*}{\Delta x^2} = 0 \end{aligned} \tag{37}$$

Therefore, we can see that, the converged solution of the DDM method satisfies the same equation as the reference solution in all the points $x_j \in \Omega_1$, except in $x_{N-1}$ and $x_N$, and in all the points $x_j \in \Omega_2$, except in $x_N$.

### 5.3.1 Analytical proof of error $= O(\Delta x)$

### 5.3.2 Numerical verification of the error

The problem (24) - (25) was solved until the convergence with 5 different uniform spatial discretizations, over one time step (in the interval $[0, \Delta t]$). In each case, the adopted referential solution $u^{ref}$ was the monodomain problem, solved with the same mesh size. Two errors were computed :

$$e^{N,\Gamma} = |u_N^{ref} - u_N^*| \tag{38}$$

$$e^{\Omega,*} = ||u_N^{ref} - u_N^*||_2 = \sqrt{\Delta x \left[ \sum_{j=0}^{N} (u_j^{ref} - u_j^{1,\infty})^2 + \sum_{j=N}^{2N} (u_j^{ref} - u_j^{2,\infty})^2 \right]} \tag{39}$$

corresponding respectively to the error on the interface and the error on the whole domain. In the notation $e^{N,\bullet}$, $N$ refers to the number of space steps in $\Omega$.

We are interested in the behavior of these error as the mesh size changes. As shown in the figure 3, we verify that the DDM proposed here produces a $O(\Delta x)$ error:

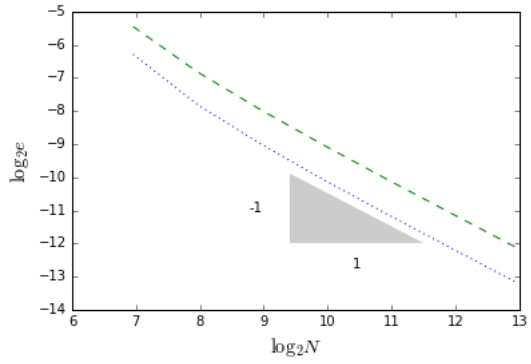



Figure 3: Numerical verification of the order of convergence of the error due to the Domain Decomposition Method

### 5.3.3 Corrections for the approximate TBCs

We will formulate modified TBCs for the ASM method in order to cancel these errors :

$$\begin{cases} \Theta_1^{c_L^*}(u_2^{n+1,k+1}) + \theta_1 = \Theta_1^{c_L^*}(u_1^{n,k}) + \theta_1' \\ \Theta_2^{c_R^*}(u_1^{n+1,k+1}) + \theta_2 = \Theta_2^{c_R^*}(u_2^{n,k}) + \theta_2' \\ \Theta_3^{c_R^*}(u_1^{n+1,k+1}) + \theta_3 = \Theta_3^{c_R^*}(u_2^{n,k}) + \theta_3' \end{cases} \tag{40}$$

with $\theta_i, \theta_i'$ derived below :

**Determination of $\theta_1, \theta_1'$** Our objective is to write the second order centered finite difference discretization (30) for the point $x_N$ :

$$\frac{u_N^* - \alpha_N^*}{\Delta t} + \frac{-\frac{1}{2}u_{N-2}^* + u_{N-1}^* - u_{N+1}^* + \frac{1}{2}u_{N+2}^*}{\Delta x^3} = 0 \tag{41}$$

Defining

$$\theta_1 = c_L \frac{u_{N+1}^2 - u_N^2}{\Delta x} + c_L^2 \frac{\Delta x}{\Delta t}\left(u_N^2 - \alpha_N^2\right) \tag{42}$$

$$\theta_1' = c_L \frac{u_N^1 - u_{N-1}^1}{\Delta x} - c_L^2 \frac{\Delta x}{\Delta t}\left(u_N^1 - \alpha_N^1\right) \tag{43}$$

we have, in the convergence, that

$$\begin{aligned} &\Theta_1^{c_L}(u_N^*) + \theta_1 = \Theta_1^{c_L}(u_N^*) + \theta_1' \implies \\ \implies & u_N^* - c_L \frac{u_{N+1}^* - u_N^*}{\Delta x} + c_L^2 \frac{u_N^* - 2u_{N+1}^* + u_{N+2}^*}{\Delta x^2} + c_L \frac{u_{N+1}^* - u_N^*}{\Delta x} + c_L^2 \frac{\Delta x}{\Delta t}\left(u_N^* - \alpha_N^*\right) = \\ & u_N^* - c_L \frac{u_N^* - u_{N-1}^*}{\Delta x} + c_L^2 \frac{u_N^* - 2u_{N-1}^* + u_{N-2}^*}{\Delta x^2} + c_L \frac{u_N^* - u_{N-1}^*}{\Delta x} - c_L^2 \frac{\Delta x}{\Delta t}\left(u_N^* - \alpha_N^*\right) \implies \\ \implies & 2c_L^2 \frac{-\frac{1}{2}u_{N-2}^* + u_{N-1}^* - u_{N+1}^* + \frac{1}{2}u_{N+2}^*}{\Delta x^2} + 2c_L^2 \frac{\Delta x}{\Delta t}\left(u_N^* - \alpha_N^*\right) = 0 \implies \\ \implies & \frac{u_N^* - \alpha_N^*}{\Delta t} + \frac{-\frac{1}{2}u_{N-2}^* + u_{N-1}^* - u_{N+1}^* + \frac{1}{2}u_{N+2}^*}{\Delta x^3} = 0 \end{aligned} \tag{44}$$

which corresponds to the discretization (30) satisfied in $x_N$.

**Determination of $\theta_2, \theta_2'$** We define

$$\begin{aligned}\theta_2 &= \frac{\Delta x}{\Delta t} c_R^2 (u_N^1 - \alpha_N^1) \\ \theta_2' &= -\frac{\Delta x}{\Delta t} c_R^2 (u_N^2 - \alpha_N^2)\end{aligned} \tag{45}$$

Indeed, we then have, in the convergence

$$\begin{aligned}&\Theta_2^{c_R}(u_N^*) + \theta_2 = \Theta_2^{c_R}(u_N^*) + \theta_2' \implies \\ \implies & u_N^* - c_R^2 \frac{u_N^* - 2u_{N-1}^* + u_{N-2}^*}{\Delta x^2} + \frac{\Delta x}{\Delta t} c_R^2 (u_N^* - \alpha_N^*) = \\ & u_N^* - c_R^2 \frac{u_N^* - 2u_{N+1}^* + u_{N+2}^*}{\Delta x^2} - \frac{\Delta x}{\Delta t} c_R^2 (u_N^* - \alpha_N^*) \implies \\ \implies & 2\frac{\Delta x}{\Delta t} c_R^2 (u_N^* - \alpha_N^*) + 2c_R^2 \frac{-\frac{1}{2}u_{N-2}^* + u_{N-1}^* - u_{N+1}^* + \frac{1}{2}u_{N+2}^*}{\Delta x^2} = 0 \implies \\ \implies & \frac{u_N^* - \alpha_N^*}{\Delta t} + \frac{-\frac{1}{2}u_{N-2}^* + u_{N-1}^* - u_{N+1}^* + \frac{1}{2}u_{N+2}^*}{\Delta x^3} = 0\end{aligned} \tag{46}$$

which corresponds to the discretization (30) satisfied in $x_N$.

**Determination of $\theta_3, \theta_3'$** Using (46) in (33), we get

$$-\frac{u_{N-1}^* - 2u_N^* + u_{N+1}^*}{\Delta x} + 2c_R \Delta x \frac{u_N^* - \alpha_N^*}{\Delta t} = 0 \tag{47}$$

As, when correcting $\Theta_2$, we have already written the discretization (30) for $x_N \in \Omega_1$, now our objective is to write it for the point $x_{N-1}$ :

$$\frac{u_{N-1}^* - \alpha_{N-1}^*}{\Delta t} + \frac{-\frac{1}{2}u_{N-3}^* + u_{N-2}^* - u_N^* + \frac{1}{2}u_{N+1}^*}{\Delta x^3} = 0 \tag{48}$$

what can be achieved by defining

$$\begin{aligned}\theta_3 = 2\frac{\Delta x}{\Delta t}\left[-\Delta x(u_{N-1}^1 - \alpha_{N-1}^1) - c_R(u_N^1 - \alpha_N^1)\right] + \frac{u_{N-3}^1 - 2u_{N-2}^1 + u_{N-1}^1}{\Delta x} \\ \theta_3' = 0\end{aligned} \tag{49}$$

In fact, in the convergence,

$$
\begin{aligned}
&\Theta_3^{c_R}(u_N^*) + \theta_3 = \Theta_3^{c_R}(u_N^*) + \theta_3' \implies \\
\implies &\frac{u_N^* - u_{N-1}^*}{\Delta x} + c_R \frac{u_N^* - 2u_{N-1}^* + u_{N-2}^*}{\Delta x^2} + 2\frac{\Delta x}{\Delta t}\left[-\Delta x(u_{N-1}^* - \alpha_{N-1}^*) - c_R(u_N^* - \alpha_N^*)\right] + \\
&\frac{u_{N-3}^* - 2u_{N-2}^* + u_{N-1}^*}{\Delta x} = \frac{u_{N+1}^* - u_N^*}{\Delta x} + c_R \frac{u_N^* - 2u_{N+1}^* + u_{N+2}^*}{\Delta x^2} \implies \\
\implies &-\frac{u_{N-1}^* - 2u_N^* + u_{N+1}^*}{\Delta x} + 2c_R\Delta_x \frac{u_N^* - \alpha_N^*}{\Delta t} + \\
&2\frac{\Delta x}{\Delta t}\left[-\Delta x(u_{N-1}^* - \alpha_{N-1}^*) - c_R(u_N^* - \alpha_N^*)\right] + \frac{u_{N-3}^* - 2u_{N-2}^* + u_{N-1}^*}{\Delta x} = 0 \implies \\
\implies &-2\frac{-\frac{1}{2}u_{N-3}^* + u_{N-2}^* - u_N^* + \frac{1}{2}u_{N+1}^*}{\Delta x} - 2\frac{\Delta x^2}{\Delta t}(u_{N-1}^* - \alpha_{N-1}^*) = 0 \implies \\
\implies &\frac{u_{N-1}^* - \alpha_{N-1}^*}{\Delta t} + \frac{-\frac{1}{2}u_{N-3}^* + u_{N-2}^* - u_N^* + \frac{1}{2}u_{N+1}^*}{\Delta x^3} = 0
\end{aligned}
$$

**Remark : modification of the reference solution** The modifications proposed above for the Interface Boundary Conditions effectively allows the points $x_{N-1}, x_N \in \Omega_1$ and $x_N \in \Omega_2$ to satisfy the same discrete equation as in the monodomain problem. Nevertheless, the solution of the DDM does not converge exactly to $u^{ref}$, for a reason that does not depend on the expression of the IBCs, but on the fact that for each domain we write two IBCs in the left boundry and only one on the right. We are using a second order centered discretization for the third spatial derivative (which uses a stencil of two points in each side of the central point), implying that we must write an uncentered discretization for the point $x_{N+1}$ when solving the problem in $\Omega_2$. Therefore, this point does not satisfy the same discrete equation as in the reference problem. In order to avoid this problem and allow us to verify that our method is able to correct the error of the DDM, we modify the discretization for the point $u_{N+1}$ in the monodomain problem, using the same second-order uncentered expression :

$$
\frac{u_{N+1}^2 - \alpha_{N+1}^2}{\Delta t} + \frac{-\frac{5}{2}u_{N+1}^2 + 9u_{N+2}^2 - 12u_{N+3}^2 + 7\frac{1}{2}u_{N+4}^2 - \frac{3}{2}u_{N+1}^2}{\Delta x^3} = 0 \quad (50)
$$

## 5.4 Optimization of the IBCs (speed of convergence)

Our objective now is to optimize the IBCs in the sense of minimizing the number of iterations of the ASM until the convergence. Therefore, similarly to the optimization of the TBCs made in the section 4, we will made a very large set of tests in order to find the coefficients $c_L$ and $c_R$ (i.e., the

constant polynomial approximation for the TBC) that provides the fastest convergence. In a first moment, we will make this study with fixed time step and space step, in order to analyze exclusively the influence of the coefficient, and after we will introduce these two parameters into the study.

As we are interested in the speed with which the solution of the DDM method converges to the reference solution, the criteria of convergence used is

$$e^{\Omega,k} \leq \epsilon \tag{51}$$

with $\epsilon = 10^{-9}$ and $e^{\Omega,k}$ defined in (39).

In order to simplify the tests and avoid too expensive computations, we will always consider $c_L = c_R = c$ in this optimization. The range of tested coefficients is $[-2.25, 2.25]$, with a step of 0.01 between them, and the maximal number of iterations is set to 100.

As a last remark, also based on the fact that we are interested only in the study of the DDM method (without influence, for example, of the error accumulated along the time steps, due to the temporal discretization), we remember that all the optimization tests will be made along only one time step.

### 5.4.1 Test varying the initial time step and the interface position

As said above, in the first set of tests we will consider a fixed time step $\Delta t = 20/2560 = 0.0078125$ and a fixed mesh size $\Delta x = 12/500 = 0.024$. Nevertheless, we will consider two subsets tests, that will allow us to study the speed of convergence with different initial conditions :

1. Tests varying the initial time step $t_0$, with the interface in the center of the monodomain $\Omega = [-6, 6]$;

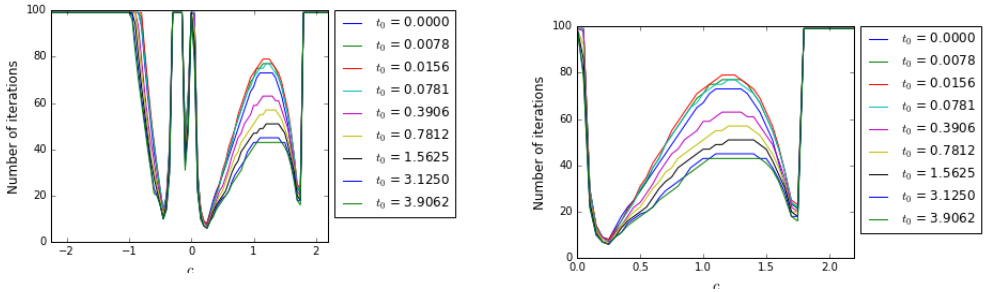
2. Tests varying the position of the interface ($x_{interface} = -L + \alpha 2L$, where $L = 6$ and $0 < \alpha < 1$), for a fixed initial time $t_0 = 0.78125$

In all the cases, the reference solution $u^{ref}$ will be the solution of the monodomain problem

$$\begin{cases} u_t + u_{xxx} = 0, \quad x \in \Omega, \quad t \in [t_0, t_0 + \Delta t] \\ u_(t_0, x) = u^{exact}(t_0, x), \quad x \in \Omega \\ \Upsilon_1(u, -L) = 0, \quad t \in [t_0, t_0 + \Delta t] \\ \Upsilon_2(u, L) = 0, \quad t \in [t_0, t_0 + \Delta t] \\ \Upsilon_3(u, L) = 0, \quad t \in [t_0, t_0 + \Delta t] \end{cases}$$

where $u^{exact}$ is the exact solution given by 19 and $\Upsilon_i,\ i = 1, 2, 3$ are the external boundary conditions given by 26.

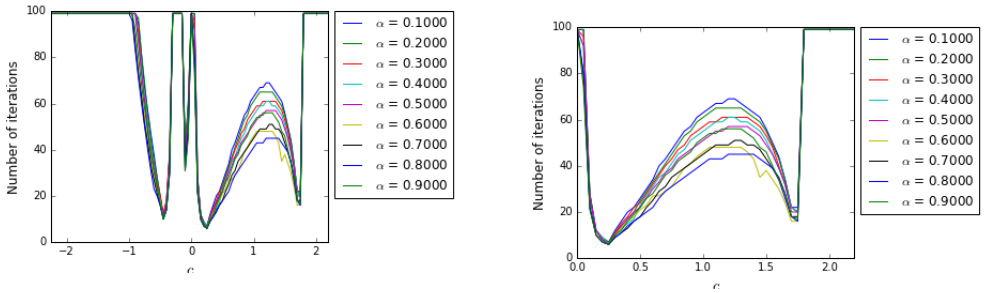
The results are summarized in the figures 4 and 5, with the number of of iterations plotted as function of the coefficient $c$. They show a very similar behavior of these curves, and, instead of some differences in the zone of convergence of the method and a separation for the values of $c$ that result in higher numbers of iterations, all the curves reaches the minimum (6 iterations) on the same point ($c = 0.25$).




(a) General view for positive and negative coefficients

(b) Detail for $c \geq 0$.

Figure 4: Number of iterations until the convergence as function of the coefficient of the TBC (for a fixed interface and different values of $t_0$)




(a) General view for positive and negative coefficients

(b) Detail for $c \geq 0$

Figure 5: Number of iterations until the convergence as function of the coefficient of the TBC (for a fixed $t_0$ and different positions of the interface

The figure 6 shows the evolution of the error, as function of the iterations, for the five coefficients $c$ that gave the fastest convergences.

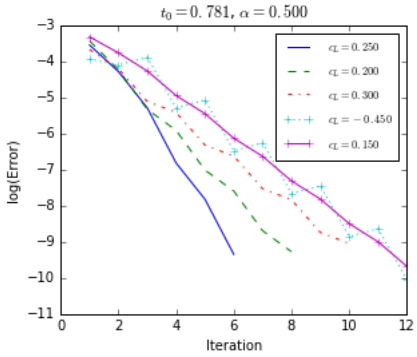



Figure 6: Error evolution with the iterations for the fastest results

### 5.4.2 Tests varying $\Delta t$ and $\Delta x$

After verifying that the method behaves similarly for every initial condition (i.e., every $t_0$) and every position of the interface, we will now keep these parameters fixed ($t_0 = 0$ and $\alpha = 0.5$) and make new tests with different values of $\Delta t$ (with fixed $\Delta x = 12/500$) and different values of $\Delta x$ (with fixed $\Delta t = 0.02$).

The number of iterations as functions of the coefficients, for some of the tests, are shown in the figures 7 and 8. The figure 8 presents the minimal number of iterations for each $\Delta t$ or $\Delta x$.

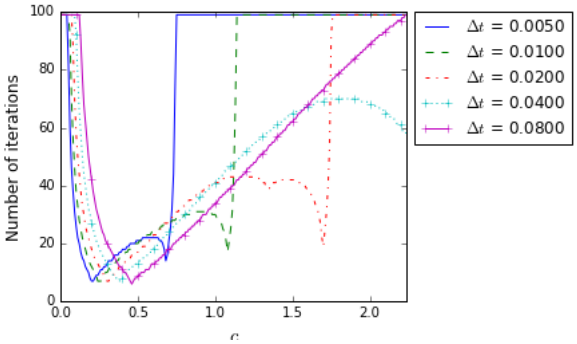



Figure 7: Number of iterations until the convergence as function of the coefficient of the TBC (for a fixed $2N = 250$ and different values of $\Delta t$)

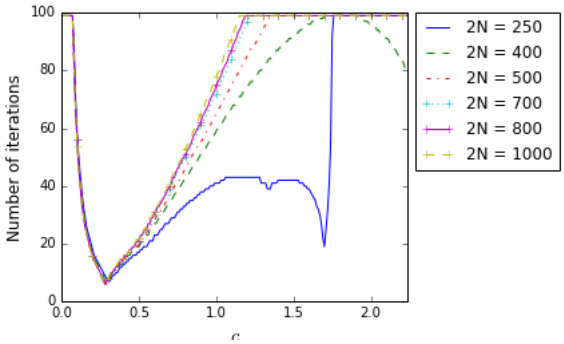



Figure 8: Number of iterations until the convergence as function of the coefficient of the TBC (for a fixed $\Delta t = 0.02$ and different values of $\Delta x$)

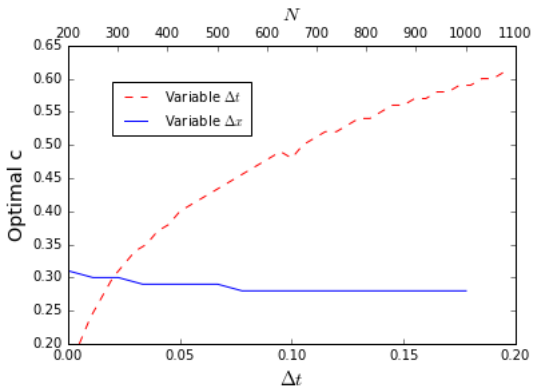



Figure 9: Optimal coefficients as function of the time step and the space step

## 5.5 Partial conclusion

...

# 6 Conclusion

# References

[1] X. Antoine, A. Arnold, C. Besse, M. Ehrhardt, and C. Schädle. A review of Transparent Boundary Conditions or linear and nonlinear Schrödinger equations. *Communications in Compuational Physics*, 4(4):729–796, October 2008.

[2] C. Besse, M. Ehrhardt, and I. Lacroix-Violet. Discrete Artificial Boundary Conditions for the Korteweg-de Vries Equation. working paper or preprint, Jan. 2015.

[3] P. Cheung. Lecture 6 - Frequence-domain analysis : Laplace Transform. [Imperial College of London - Signals and Linear Systems.

[4] C. Japhet and F. Nataf. The best interface conditions for Domain Decompostion methods : Absorbing Boundary Conditions. http://www.ann.jussieu.fr/ nataf/chapitre.pdf.

[5] C. Zheng, X. Wen, and H. Han. Numerical solution to a linearized kdv equation on unbounded domain. *Numerical Methods for Partial Differential Equations*, 24(2):383–399, 2008.