Contents

1 Introduction

The Korteweg - de Vries (KdV) equation, derived by [7] in 1895, models the propagation of waves with small amplitude and large wavelength, taking in account nonlinear and dispersive effects. In terms of dimensionless but unscaled variables, it can be written as [3]

$$u_t + u_x + uu_x + u_{xxx} = 0 \tag{1}$$ {?}

As done in [9] (and in [4] as a special case of their work), we will focus in this paper on the linearized KdV equation without the advective term :

$$u_t + u_x + uu_x + u_{xxx} = 0 \tag{2}$$ eq:DKdV

to which we will refer as *dispersion equation*.

The work developed here is inspired on [9] and [4]. Nevertheless, our objectives are very different from theirs; to understand these differences, a contextualization must be done, with a brief description of the sources of errors and uncertainties that affect the numerical simulations of physical models.

In a general way, we can group these sources in conceptual modeling errors and numerical errors [8]. In the first group, we can mention conceptual modeling assumptions (for the physical phenomena and the boundary conditions) and uncertainties in the geometry, the initial data, boundary data (missing informations or erros in the measuring method) and in the parameters that play a role in the model [8, 2]. Concerning the numerical errors, we can mention those related to the computational domain size, the temporal errors and the spatial errors. [6, 8]

The total error of the numerical simulation is a contribution of each one of these sources. Knowing and quantifying them is essential to improve the implementation of the physical of model, and, in this context, the separated study of each one of these contributions has a great importance.

Among the types of errors mentioned above, [9] and [4] attempted to reduce the one related to the modeling of the boundary conditions. Such modeling is necessary when one solves in a finite computational domain the problem (3) :

$$\begin{cases} u_t + u_{xxx} = 0, \quad t \in \mathbb{R}^+, \quad x \in \mathbb{R} \\ u(0,x) = u_0(x), \quad x \in \mathbb{R} \\ u \longrightarrow 0, \quad x \longrightarrow \infty \end{cases} \tag{3}$$ eq:problemDKdV

In fact,“in the case when a PDE is employed to model waves on unbounded domain and the numerical simulation is performed, it is a common practice to truncate the unbounded domain by introducing artificial boundaries, which necessitates additional boundary conditions to be designed. A proper choice of these boundary conditions should ensure both stability and accuracy of the truncated initial-boundary value problem." [9]. Although using different approaches, both authors sought to construct Absorbing Boundary Conditions (ABCs) (which simulate the absorption of a wave quitting the computational domain) or Transparent Boundary Conditions (TBCs) (which makes the approximate solution on the computational domain coincides with solution of the whole domain).

In this paper, we will propose a Domain Decomposition Method (DDM) for solving the problem (3) in a finite domain, i.e., we will decompose the computational domain in subdomains and solve (3) in each one of them. This requires the formulation of appropriate conditions on the interface between the subdomains, in order to minimize the error due to the DDM (which we include in the list of numerical errors in the simulation of a physical model). Following the principle of studying each type of numerical error separately, we do not attempt here to minimize the errors due to the introduction of external boundaries of the computational domain (although we also make use of TBCs), but only due to the decomposition of the domain and the introduction of an interface boundary.

As a consequence, our work shall not use the same reference solution as the one used by [9] and [4] : for validating their approaches, they compare their approximate solution with the exact solution of the whole domain. In the other hand, our reference solution will be the solution computed on the computational monodomain, which we will construct with approximate formulations for the TBCs proposed in the mentioned works.

This paper is organized in the following way : in section 2, we recall the exact TBCs derived by [9] for (2) and propose approximations for them, leading to very simple conditions (avoiding, for example, integrations in time) depending on two coefficients. With some numerical experiments, we show that these approximate TBCs work quite well (although not so well as the approaches of [9] and [4]), motivating us to use them in the sequence of our work. In section 3, we describe the Domain Decomposition Method used here and we construct it using our approximate TBCs as Interface Boundary Conditions (IBCs). Small modifications are proposed for these IBCs such that the solution of the DDM problem converges exactly to the reference solution (the solution of the monodomain problem). Finally, we perform a large set of numerical tests in order to optimize the IBCs, in the sense that we search the coefficients for the approximate TBCs that provide the fastest convergence.

2 Approximate Transparent Boundary Conditions for the dispersive equation

⟨sec:TBC⟩
2.1 The exact TBCs for the continuous equation

In [4], Transparent Boundary Conditions (TBCs) are derived for the one-dimensional continuous linearized KdV equation (or Airy equation) :

$$u_t + U_1 u_x + U_2 u_{xxx} = h(t,x), \quad t \in \mathbb{R}^+, \quad x \in \mathbb{R} \tag{4}$$ eq:LKdV

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

For the homogeneous initial boundary value problem

$$\begin{cases} u_t + U_1 u_x + U_2 u_{xxx} = h(t,x), \quad t \in \mathbb{R}^+, \quad x \in [a,b] \\ u(0,x) = u_0(x), \quad x \in [a,b] \\ +\text{boundary conditions} \end{cases}$$

the TBCs are given by

$$\begin{aligned} u(t,a) - U_2 \mathcal{L}^{-1}\left(\frac{\lambda_1(s)^2}{s}\right) * u_x(t,a) - U_2 \mathcal{L}^{-1}\left(\frac{\lambda_1(s)}{s}\right) * u_{xx}(t,a) = 0 \\ u(t,b) - \mathcal{L}^{-1}\left(\frac{1}{\lambda_1(s)^2}\right) * u_{xx}(t,b) = 0 \\ u_x(t,b) - \mathcal{L}^{-1}\left(\frac{1}{\lambda_1(s)}\right) * u_{xx}(t,b) = 0 \end{aligned} \tag{5}$$ eq:continuousTBC

where $\mathcal{L}^{-1}$ denotes the inverse Laplace transform, $*$ the convolution operator, $s \in \mathbb{C}$ is the Laplace frequency and λ_1 is, among the three roots of the cubic characteristic equation obtained when solving (4) 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 on the dispersion equation (2). In this case, accordingly to [9], the only root with negative real part is

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

2.2 Approximation for the TBCs using a constant polynomial

The computation of the TBCs (5) is not simple due to the inverse Laplace transform, 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 [4] (who derives TBCs derived for the discrete linearized KdV equation). Nevertheless, the objectives of our work and the work of [4] 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 Interface Boundary Conditions (IBCs) 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.

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 (5) only in function of c :

$$\frac{\lambda^2}{s} = c \qquad \frac{\lambda}{s} = -c^2 \qquad \frac{1}{\lambda_1(s)^2} = c^2 \qquad \frac{1}{\lambda_1(s)} = -c \tag{7}$$ eq:appP0

Replacing (7) in (5), using some well-know properties of the Laplace Transform (linearity and convolution) 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

$$\begin{aligned} \Theta_1^{c_L}(u,x) = u(t,x) - c_L u_x(t,x) + c_L^2 u_{xx}(t,x) = 0 \\ \Theta_2^{c_R}(u,x) = u(t,x) - c_R^2 u_{xx}(t,x) = 0 \\ \Theta_3^{c_R}(u,x) = u_x(t,x) + c_R u_{xx}(t,x) = 0 \end{aligned} \tag{8}$$ eq:appTBCP0

We notice that the approximation (8) has the same form as the exact TBCs for the equation (2) presented in [9] and [4], being the constants c_L, c_R an approximation for fractional integral operators.

Considering a discrete domain with mesh size Δx and points $x_0, ..., x_N$ and using some finite difference approximations, the approximate TBCs (8) are discretized as

$$\begin{aligned} u_0 - c_L \frac{u_1 - u_0}{\Delta x} + c_L^2 \frac{u_0 - 2u_1 + u_2}{\Delta x^2} = 0 \\ u_N - c_R^2 \frac{u_N - 2u_{N-1} + u_{N-2}}{\Delta x^2} = 0 \\ \frac{u_N - u_{N-1}}{\Delta x} + c_R^2 \frac{u_N - 2u_{N-1} + u_{N-2}}{\Delta x^2} = 0 \end{aligned} \tag{9}$$ eq:appDiscTBCP0

In order to illustrate the results provided by these approximations, we present briefly some numerical tests with the same problem solved by [9] and [4], given by (10a)-(10c) and for which the exact solution is given by (11) :

$$\begin{cases} u_t + u_{xxx} = 0, \quad x \in \mathbb{R} & (10a)\ \boxed{\texttt{eq:testCaseBesse1}} \\ u(0,x) = e^{-x^2}, \quad x \in \mathbb{R} & (10b)\ ?\underline{\texttt{eq:testCaseBesse2}}? \\ u \to 0, \quad |x| \to \infty & (10c)\ \boxed{\texttt{eq:testCaseBesse3}} \end{cases}$$

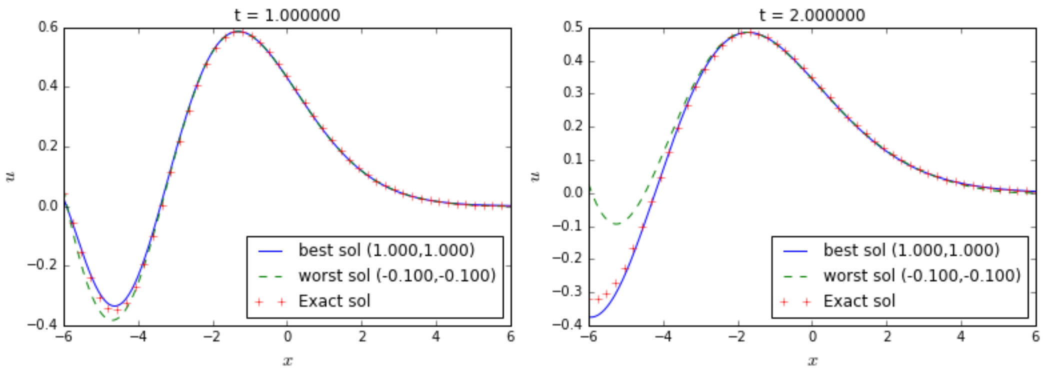
$$u_{exact}(t,x) = \frac{1}{\sqrt[3]{3t}} Ai\left(\frac{x}{\sqrt[3]{3t}}\right) * e^{-x^2} \qquad (11)\ \boxed{\texttt{eq:exactSolution}}$$

where Ai is the Airy function.

As done by [9] and [4], we solved the problem 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 the time step is $\Delta t = 4/2560 = 0.0015625$. We computed, as in [4], the following errors, computed respectively in each time step and in all the time interval :

$$e^n = \frac{\left\| u^n_{exact} - u^n_{computed} \right\|_2}{\left\| u^n_{exact} \right\|_2} \qquad e_{L2} = \sqrt{\Delta t \sum_{n=1}^{T_{max}} (e^n)^2}$$

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} 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 consider the ones in which the numerical solution diverged (following the arbitrary criteria $e_{L2} > 10$). Finally, the table 1 presents the ten tests that presented the smallest e_{L2}.



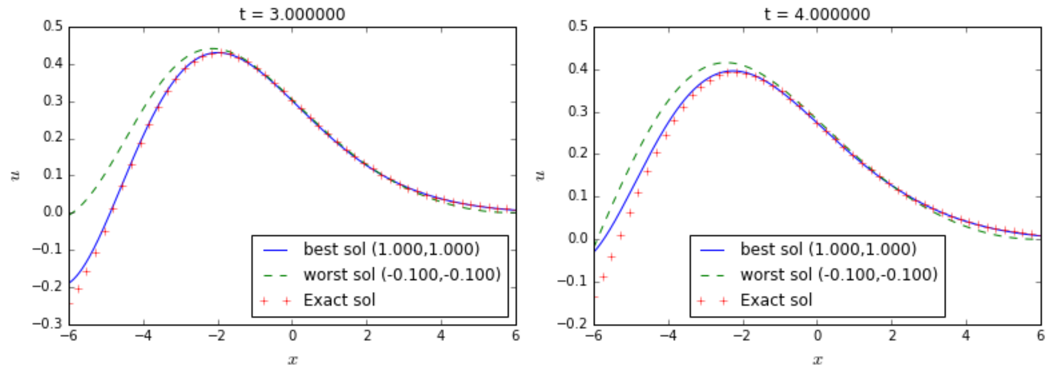


Figure 1: Best and worst solution compared with analytical solution, for the constant polynomial approximation
⟨fig:firstTestsP0⟩

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
⟨tab:firstTestsP0⟩

2.3 Partial conclusion

It must be clear that our approach does not provide better Transparent Boundary Conditions than the one proposed by [4], what, as discussed in the introduction of this paper, is not the objective of the work developed here. Indeed, [4] derives TBCs for two discrete schemes, and the worst result among them, using the same Δx and Δt 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. Nevertheless, considering that our main goal is the application of the TBCs to a Domain Decomposition Method, we focus in minimizing the error due to the Interface Boundary Conditions imposed in this kind of method, and not in the errors due to the external boundary conditions.

In fact, 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. As a development of our approach, we also tested an approximation for $\frac{\lambda^2}{s}$ using a linear polynomial, but, although the increment in the complexity (including time derivative terms up to the second derivative, what requires the storage of previous computed solutions), it does not provide a better approximation for the TBC, in comparison with the approximation using a constant polynomial.

Therefore, in the following of this paper, we will continue using the approximate TBCs given by the operators $\Theta_i^c, \ i = 1, 2, 3$, defined in (8).

3 Application to a Domain Decomposition Method

⟨sec:DDM⟩ The discrete approximations (9) for the Transparent Boundary Conditions for the equation (2) will be applied as Interface Boundary Conditions (IBC) in a Domain Decomposition Method (DDM). Firstly, following [5], we will briefly describe the DDM that we will consider here, and after we will describe and test the incorporation of the proposed IBCs.

3.1 The Schwarz Method

Domain Decomposition Methods allow to decompose a domain Ω in multiple subdomains Ω_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 Ω_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 neighbor subdomains. Therefore, in each interface between the subdomains Ω_i and Ω_j, the boundary condition for the problem in Ω_i is

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

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

Without loss of generality, in the following we will consider a domain Ω decomposed in two non-overlapping subdomains, Ω_1 and Ω_2, with $\Gamma = \Omega_1 \bigcap \Omega_2$.

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

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

In fact, accordingly to [5], the optimal additive Schwarz method for solving the problem

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

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 ∂n_i is the outward normal to Ω_i on Γ , and the D2N (Dirichlet to Neumann) operator is defined by

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

with α defined on Γ. v is solution of the following problem, solved in the complementary set of Ω_i, denoted by Ω_i^c

$$\begin{cases} \mathcal{A}(v) = f \quad \text{in} \quad \Omega_i^c \\ v = 0 \quad \text{on} \quad \partial\Omega_i \backslash \Gamma \\ v = \alpha \quad \text{on} \quad \Gamma \end{cases}$$

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.

3.2 ASM with the approximate TBCs for the dispersion equation

The resolution of the dispersion equation (2) with the Additive Schwarz method, using the constant polynomial approximation for the TBCs, is written as

$$
\begin{cases}
(u_1^{n,k+1})_t + (u_1^{n,k+1})_{xxx} = 0, \quad x \in \Omega_1, \quad t \geq t_0 \\
u_1^{n,0} = u_1^{n-1,\infty}, \quad x \in \Omega_1 \\
\Upsilon_1^{c_L}(u_1^{n+1,k+1}, -L) = 0, \\
\Theta_2^{c_R}(u_1^{n+1,k+1}, 0) = \Theta_2^{c_R}(u_2^{n,k}, 0), \\
\Theta_3^{c_R}(u_1^{n+1,k+1}, 0) = \Theta_3^{c_R}(u_2^{n,k}, 0)
\end{cases}
\tag{12}
$$ eq:problemDDM1

$$
\begin{cases}
(u_2^{n,k+1})_t + (u_2^{n,k+1})_{xxx} = 0, \quad x \in \Omega_2, \quad t \geq t_0 \\
u_2^{n,0} = u_2^{n-1,\infty}, \quad x \in \Omega_2 \\
\Theta_1^{c_L}(u_2^{n+1,k+1}, 0) = \Theta_1^{c_L}(u_1^{n,k}, 0) \\
\Upsilon_2^{c_R}(u_2^{n+1,k+1}, L) = 0 \\
\Upsilon_3^{c_R}(u_2^{n+1,k+1}, L) = 0
\end{cases}
\tag{13}
$$ eq:problemDDM2

where $\Upsilon_i, \quad i = 1, 2, 3$, are the external boundary conditions (i.e, defined on $\partial\Omega_i \backslash \Gamma$).

Considering that we want to analyze and minimize the error due to the application of a Domain Decomposition Method, the reference solution u^{ref} in our study will be the solution of the monodomain problem

$$
\begin{cases}
u_t + u_{xxx} = 0, \quad x \in \Omega, \quad t \in [t_0, t_0 + \Delta t] \\
u(t_0, x) = u^{exact}(t_0, x), \quad x \in \Omega \\
\Upsilon_1(u, -L) = 0, \quad t \in [t_0, t_0 + \Delta t] \\
\Upsilon_2(u, L) = 0, \quad t \in [t_0, t_0 + \Delta t] \\
\Upsilon_3(u, L) = 0, \quad t \in [t_0, t_0 + \Delta t]
\end{cases}
\tag{14}
$$ eq:problemMonodomai

We notice that we will always compare the solutions computed along only one time step. This is necessary for the separated study of the DDM method (without influence, for example, of the error accumulated along the time steps, due to the temporal discretization).

The external BCs $\Upsilon_i, \quad i = 1, 2, 3$ 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

$$
\begin{gathered}
\Upsilon_1(u, x) = u - u_x + u_{xx} = 0 \\
\Upsilon_2(u, x) = u = 0 \\
\Upsilon_3(u, x) = u_x = 0
\end{gathered}
\tag{15}
$$?eq:externalBCsDDM?

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 Ω (as shown in the table 1). Nevertheless, it does not have much importance in the study that we will done here, as we want to study exclusively the behavior of the DDM. 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, as we done in (12)-(13) and (14).

Remarks on the notation As the following study will be made considering the execution of the method over only one time step, we can suppress the index denoting the instant t_n and use a clearer notation for the solution : u_j^i, where i indicates the subdomain Ω_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 taken into account, we will add the superscript k to indicate the iteration.

Concerning the spatial discretization, the monodomain Ω 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 Ω_1 and Ω_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 $u_N^1 = u_N^2 = u_N^*$

3.3 Discretization of the problem

For the interior points of each one of the domains, we will consider a second order spatial discretization of the equation (2).

$$\frac{u_j^i - \alpha_j^i}{\Delta t} + \frac{-\frac{1}{2}u_{j-2}^i + u_{j-1}^i - u_{j+1}^i + \frac{1}{2}u_{j+2}^i}{\Delta x^3} = 0 \qquad (16)$$ eq:FDdiscretization

which is valid 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, α_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 an appoximate TBC. Considering that one TBC is written for the left boundary and two for the right one, we have to impose an uncentered discretization only for the second leftmost point of the domain. For example, for the point x_1 :

$$\frac{u_1^2 - \alpha_1^2}{\Delta t} + \frac{-\frac{5}{2}u_1^2 + 9u_2^2 - 12u_3^2 + 7\frac{1}{2}u_4^2 - \frac{3}{2}u_5^2}{\Delta x^3} = 0 \qquad (17)$$?eq:uncenteredFDdisc

and similarly to the other points near the boundaries.

In the resolution of the problem in Ω_1, two interface boundary conditions are imposed (corresponding to Θ_2 and Θ_3) to the discrete equations for the points x_{N-1} and x_N. On the other hand, in the resolution of the problem in Ω_2, only one interface boundary condition is used (corresponding to Θ_1), being imposed to the point x_N.

Remark : modification of the reference solution Even if the DDM with the proposed Interface Boundary Conditions is compatible with the monodomain problem (which we will see that is not the case), 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 boundary 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 Ω_2. Therefore, this point does not satisfy the same discrete equation as in the reference problem. In order to avoid this incompatibility and allow us to study the behavior of the DDM, we will modify the discretization for the point u_{N+1} in the monodomain problem, using the same second-order uncentered expression :

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

The figure 2 resumes the discretizations imposed to each point in the monodomain and the DDM problems, as described above:

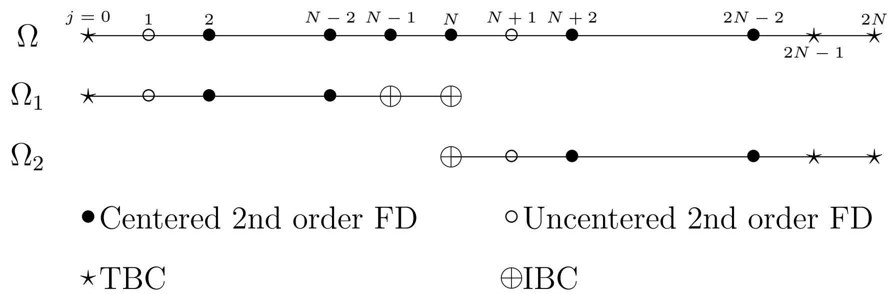


Figure 2: Scheme indicating the discretization imposed to each point in the monodomain and the DDM problems

fig:discretizations⟩

3.4 Corrections for the approximate IBCs

When using approximate TBCs in the ASM, one should guarantee that the converged solutions u^* satisfy the same equation as the solution u_{ref} of the monodomain problem. Nevertheless, usually this property is not verified by DDMs , which is the case of the method proposed here : one can easily see that, in the convergence, the solution u^* does not satisfy the discrete equation (16) on the points where the IBCs are imposed (the poins $x_{N-1}, x_N \in \Omega_1$ and $x_N \in \Omega_2$).

Therefore, we will formulate modified TBCs for the ASM in order to avoid this problem :

$$\begin{aligned}
\Theta_1^{c_L}(u_2^{n+1,k+1}) + \theta_1 &= \Theta_1^{c_L}(u_1^{n,k}) + \theta_1' \\
\Theta_2^{c_R}(u_1^{n+1,k+1}) + \theta_2 &= \Theta_2^{c_R}(u_2^{n,k}) + \theta_2' \\
\Theta_3^{c_R}(u_1^{n+1,k+1}) + \theta_3 &= \Theta_3^{c_R}(u_2^{n,k}) + \theta_3'
\end{aligned} \tag{18}$$ eq:correctedTBC

with θ_i, θ_i' given by

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

$$\theta_2 = \frac{\Delta x}{\Delta t} c_R^2 (u_N^1 - \alpha_N^1)$$
$$\theta_2' = -\frac{\Delta x}{\Delta t} c_R^2 (u_N^2 - \alpha_N^2)$$

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

It is straightforward to verify that the DDM problem with these modifications in the TBCs assure that the converged solution u^* satisfies, in every point, the same discrete equations as the solution u^{ref} of the monodomain problem (14).

We notice that all the modification terms $\theta_i, \theta_i', \ i = 1,2,3$ are of order $O(\Delta x)$ (they are composed of discrete versions of time derivatives and second spatial derivatives multiplied by Δx). It is essential to assure that these terms are small, in order not to cause substantial modifications to the respective approximate TBCs Θ_i.

3.5 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. We will make 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

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

with $\epsilon = 10^{-9}$ and

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

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 $[-10.0, 20.0]$ (chosen after initial tests to identify a proper interval), with a step equal to 0.1 between them (or even smaller, up to 0.005, in the regions near the optimal coefficients), and the maximal number of iterations is set to 100.

3.5.1 Test varying the initial time step and the interface position

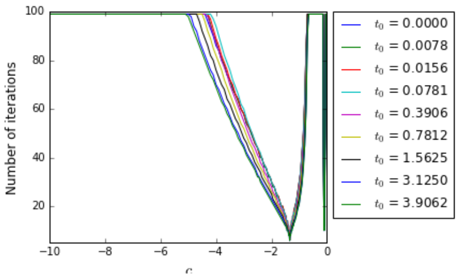
As said above, in the first set of tests we will consider a fixed time step $\Delta t = 20/2560 = 0.0078125$ and a fixed mesh size $\Delta x = 12/500 = 0.024$. Moreover, we will consider two subsets of tests, that will allow us to study the speed of convergence with different initial conditions and different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

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

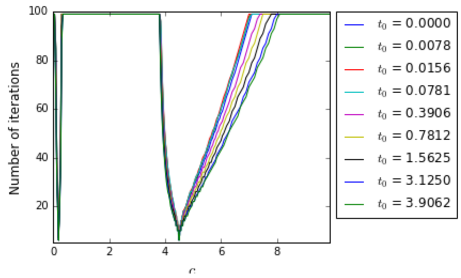
2. Tests varying the position of the interface ($x_{interface} = -L + \alpha 2L$, where $L = 6$ and $0 < \alpha < 1$), for a fixed initial time $t_0 = 0.78125$.

In all the cases, the reference solution u^{ref} will be the solution of the monodomain problem (14).

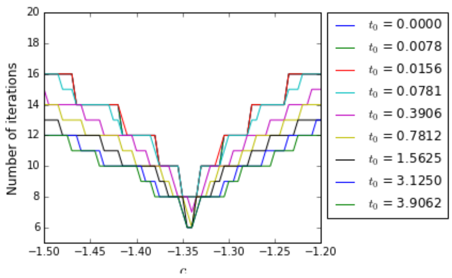
The results are summarized in the figures 3 and 4, with the number of iterations plotted as function of the coefficient c. For the sake of clearness, the results for negative and positive coefficients are shown in separated graphs. They show a very similar behavior of all the curves, with two minimums for $c < 0$ and other two for $c > 0$, which have practically the same value for all the curves (approximately -1.35, -0.10, 0.20 and 4.50). The minima closest to zero are associate to a very discontinuous peak, while the other two are associated with smoother curves. A detail of the curves around each minima are shown in the figures 2c to 2f and 3c to reffig:optimVarInterfacePDetail2. For some curves, the minimum is associated with the coefficients closest to zero, and, for other ones, to the other minimum, but the minimal number of iterations are very similar (between 5 and 7).



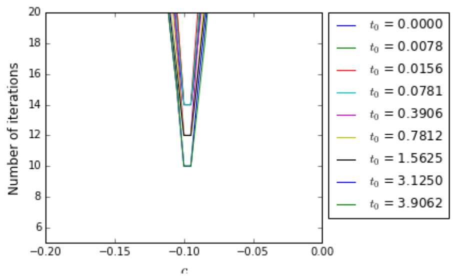
(a) General view for negative coefficients



(b) General view for positive coefficients

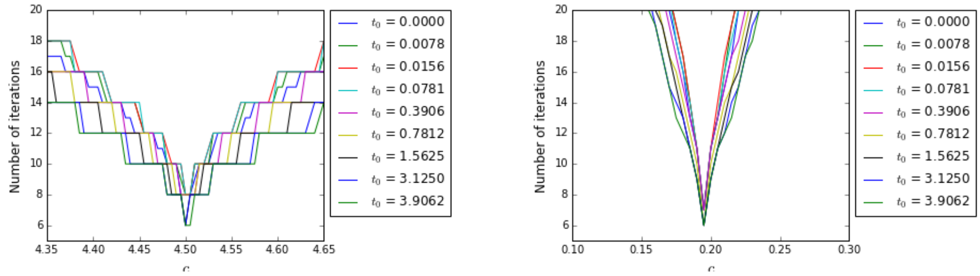


(c) Detail around one of the optimal negative coefficients



(d) Detail around the other optimal negative coefficient

g:optimVarTONDetail⟩ ?⟨fig:optimVarTONDetail2⟩?



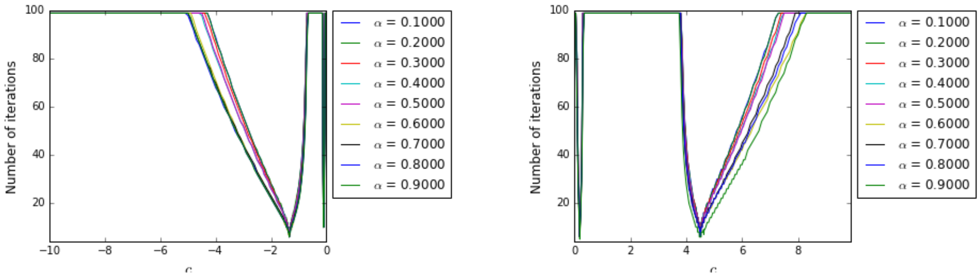
(e) Detail around one of the optimal positive coefficients

(f) Detail around the other optimal positive coefficient

:optimVarT0PDetail⟩? ⟨fig:optimVarT0PDetail2⟩

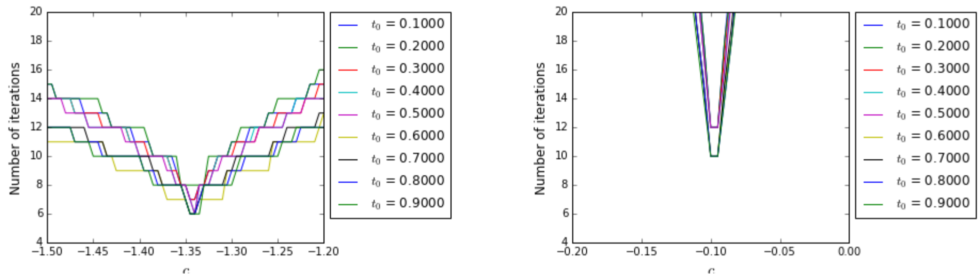
Figure 3: Number of iterations until the convergence as function of the coefficient of the TBC (for a fixed interface and different values of t_0)

⟨fig:optimVarT0⟩



(a) General view for negative coefficients

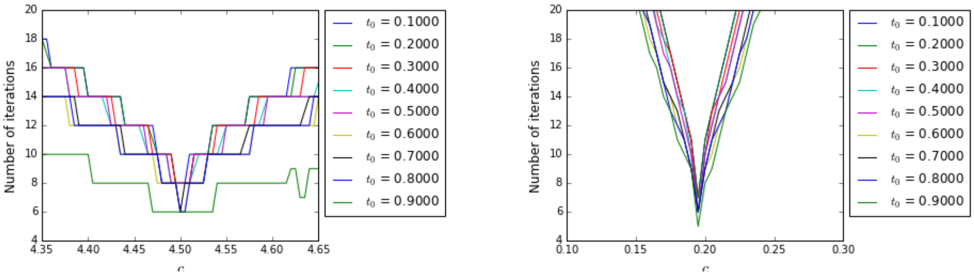
(b) General view for positive coefficients



(c) Detail around one of the optimal negative coefficients

(d) Detail around the other optimal negative coefficient

VarInterfaceNDetail⟩ ?⟨fig:optimVarInterfaceNDetail2⟩?



(e) Detail around one of the optimal positive coefficient

(f) Detail around the other optimal positive coefficient

arInterfacePDetail⟩? ?⟨fig:optimVarInterfacePDetail2⟩?

Figure 4: 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)

g:optimVarInterface⟩

The figure 5 shows the evolution of the error, as function of the iterations, for the five coefficients c that gave the fastest convergences, for a fixed initial instant and a fixed position position of the interface. For other values of t_0 and α this graph is similar, concerning the number of iterations and the fact that the convergence is more regular for the coefficients closest to zero, compared to the other optimal coefficients.

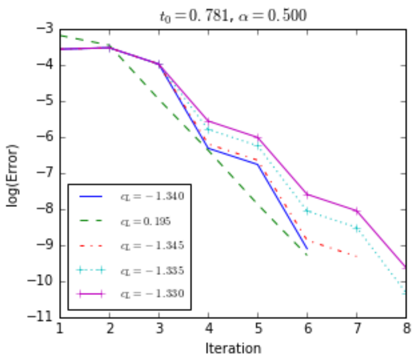


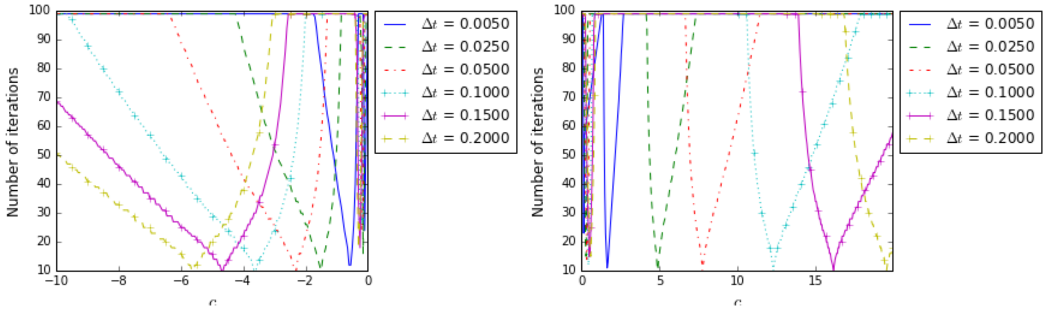
Figure 5: Error evolution with the iterations for the fastest results

⟨fig:errorEvolution⟩

3.5.2 Tests varying Δt and Δx

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 Δt (with fixed $\Delta x = 12/250$) and different values of Δx (with fixed $\Delta t = 0.02$).

The number of iterations as functions of the coefficients, for some of the tests, are shown in the figure 6 and 7. The figure 8 presents the optimal coefficient for each Δt or Δx. Considering the observation we did before about the similar results (i.e, the number of iterations until the convergence) for the four optimal coefficients, we only took into account, for the construction of this curve, the two minima farther from zero : it was done because, as shown in the figures 6 and 7, these minima have a strong dependence on Δt or Δx, and we will seek to study this relation.

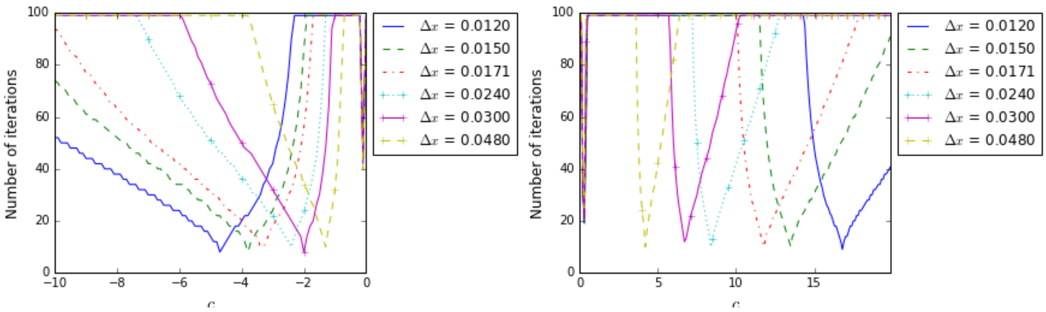


(a) Negative coefficients

(b) Positive coefficients

Figure 6: Number of iterations until the convergence as function of the coefficient of the TBC, for a fixed $2N = 250$ and different values of Δt

fig:niterxCoefVarDt⟩



(a) Negative coefficients

(b) Positive coefficients

Figure 7: 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 Δx

fig:niterxCoefVarDx⟩

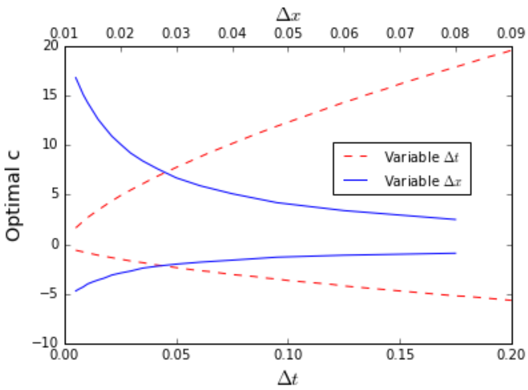
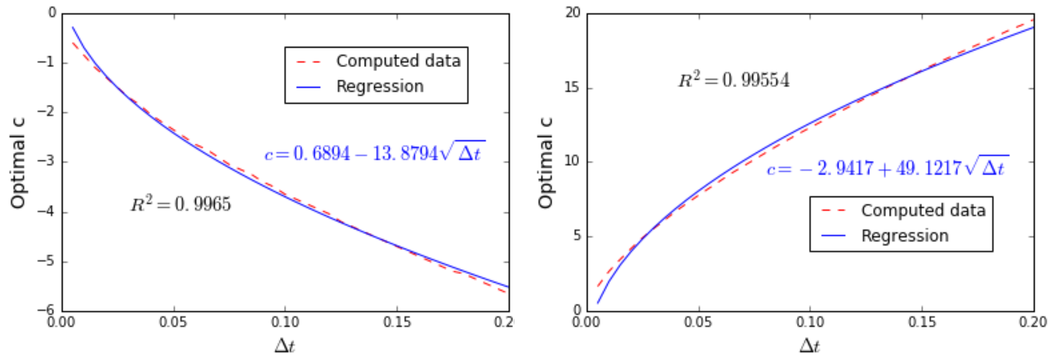


Figure 8: Optimal coefficients as function of the time step and the space step

CoefVarDxDtCorrectN⟩

The figure 8 suggests that some kind of regressions could be done in order to obtain a curve of the optimal coefficient in function of Δt or Δx. The shapes of the numerical curves indicate that the optimal c shall be function of $\sqrt{\Delta t}$ and $\frac{1}{\Delta x}$. In fact, as shown in the figures 9 and 10, these models provide very good approximations (specially in the case $c = c(\Delta x)$), both quantitatively (with all the coefficients of determination R^2 bigger than 0.99) and qualitatively .

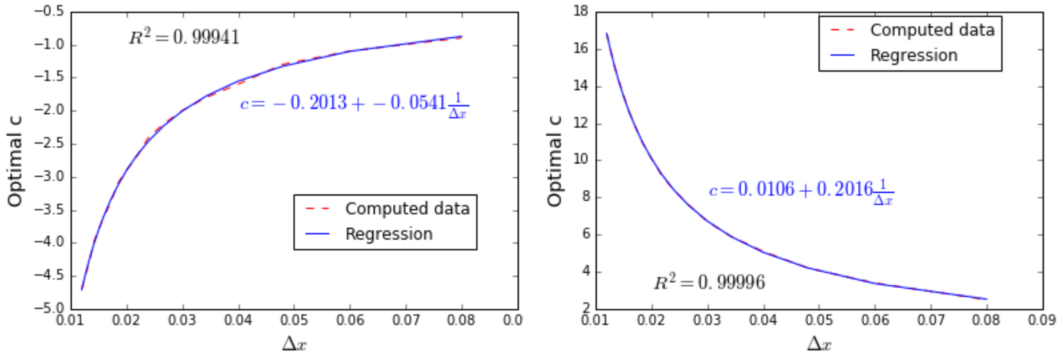


(a) Regression for negative coefficients

(b) Regression for positive coefficients

Figure 9: Numerical and regression curve for the optimal coefficient as function of Δt

⟨fig:regressionDt⟩



(a) Regression for negative coefficients

(b) Regression for positive coefficients

Figure 10: Numerical and regression curve for the optimal coefficient as function of Δt

⟨fig:regressionDx⟩

3.6 Partial conclusion

The results presented in this section show that the Domain Decomposition Method proposed here, consisting in the Additive Schwarz Method with our approximate TBCs, is able to provide a fast convergence toward the solution of the monodomain problem. Furthermore, using the corrected TBCs (18), this convergence is exact. Therefore, we reached our goals of solving the dispersion equation in a finite domain divided in two subdomains.

Moreover, the results of the optimization tests are very satisfying regarding a more general application of our method. Firstly, for fixed spatial and temporal discretizations, we obtained optimal coefficients for the method independently of the initial solution and the size of the subdomains (i.e., independently of the initial instant and the position of the interface). Secondly, we obtained good regression curves for the optimal coefficient as function of Δt or Δx, which could allow the application of the model, with fast convergence, for tests different from the ones made in this study.

4 Conclusion

References

Xaviereta l2008 [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.

balagurusamy2008 [2] E. Balagurusamy. *Numerical methods*. Tata McGraw-Hill, 2008.

BBM1971 [3] T. B. Benjamin, J. L. Bona, and J. J. Mahony. Model equations for long waves in nonlinear dispersive systems. *Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 272(1220):47–78, 1972.

besse2015 [4] C. Besse, M. Ehrhardt, and I. Lacroix-Violet. Discrete Artificial Boundary Conditions for the Korteweg-de Vries Equation. working paper or preprint, Jan. 2015.

Japhet2003 [5] C. Japhet and F. Nataf. The best interface conditions for Domain Decompostion methods : Absorbing Boundary Conditions. http://www.ann.jussieu.fr/ nataf/chapitre.pdf.

karniadakis1995 [6] G. Karniadakis. Toward a numerical error bar in CFD. *ASM Journal of Fluids Engineer*, 117(1):7–9, 1995.

kdv1895 [7] D. J. Korteweg and G. de Vries. On the change of form of long waves advancing in a rectangular canal and on a new type of long stationary waves. *Philosophical Magazine*, 5(39):422–443, 1895.

roache1997 [8] P. J. Roache. Quantification of uncertainty in Computational Fluid Dynamics. *Annual Review of Fluid Mechanics*, 29:123–160, 1997.

zheng2008 [9] 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.