16èmes Journées de l'Hydrodynamique

27-29 novembre 2018 - Marseille

DÉCOMPOSITION DE DOMAINE POUR DES ÉQUATIONS DE TYPE BOUSSINESQ LINÉARISÉES

DOMAIN DECOMPOSITION METHODS FOR LINEARIZED BOUSSINESQ TYPE EQUATIONS

J.G. CALDAS STEINSTRAESSER[1,3], G. KEMLIN[2], A. ROUSSEAU[1]

joao.caldas@meric.cl ; gaspard.kemlin@inria.cl ; antoine.rousseau@inria.fr

[1]Inria, Team LEMON, Montpellier, France
[2]Inria Chile, Avenida Apoquindo 2827, Las Condes, Santiago, Chile
[3]MERIC, Avenida Apoquindo 2827, Las Condes, Santiago, Chile

Résumé

Nous présentons dans ce papier la dérivation de conditions aux limites transparentes discrètes pour des équations de type Boussinesq linéarisées. Ces conditions sont non-locales en temps et nous évaluons leur précision avec une discrétisation de Crank-Nicolson sur grille décalée. Ensuite, nous utilisons ces conditions dans des méthodes de décomposition de domaine, où elles deviennent locales en temps. Nous évaluons alors leur efficacité en les comparant avec d'autres conditions aux interfaces. Enfin, nous reproduisons ce principe pour les équations non-linéaires.

Summary

In this paper, we derive discrete transparent boundary conditions for a class of linearized Boussinesq equations. These conditions happen to be non-local in time and we test numerically their accuracy with a Crank-Nicolson time-discretization on a staggered grid. We use the derived transparent boundary conditions as interface conditions in a domain decomposition method, where they become local in time. We analyze numerically their efficiency thanks to comparisons made with other interface conditions. Then, we apply the same ideas to nonlinear equations.

I – Introduction

Among the main challenges faced in the mathematical framework of coastal engineering is the study of wave propagation in the nearshore area. One field of research in this topic makes use of the Boussinesq equations for water of varying depth that describe the nonlinear propagation of waves in shallow water. The work of Peregrine [17], Green and Naghdi [11] laid the basis for many Boussinesq-type equations used nowadays. The dispersion properties of these equations have been improved by Nwogu [16] for practical numerical simulation of ocean wave processes from deep to shallow water. In this paper, we work on the equations derived by Nwogu and that can be recalled as follows. Consider a three-dimensional wave field with surface elevation $\eta(x,y,t)$ over a non-constant water depth $h(x,y)$ and with speed $\boldsymbol{u}(x,y,z(x,y),t)=(u,v)$, respectively the speeds along the x and y axis, defined at an arbitrary reference depth $z(x,y)\in[-h(x,y),0]$, that can be chosen in order to optimize the dispersion properties of the equations (see [16]). Figure 1 provides a sketch where we represent these quantities. With $\boldsymbol{\tau}$ being the bottom shear stress, Nwogu obtained (1), which consists of a continuity equation and a momentum equation. These equations have been used in a `C/Matlab` program known as the *Boussinesq Ocean and Surf Zone* (BOSZ) model, developed by Roeber and Cheung [18].

$$\begin{cases} \eta_t+\nabla\cdot[(h+\eta)\,\boldsymbol{u}]+\nabla\cdot\left[\left(\dfrac{z^2}{2}-\dfrac{h^2}{6}\right)h\nabla(\nabla\cdot\boldsymbol{u})+\left(z+\dfrac{h}{2}\right)h\nabla(\nabla\cdot(h\boldsymbol{u}))\right]=0, \\ \boldsymbol{u}_t+(\boldsymbol{u}\cdot\nabla)\boldsymbol{u}+g\nabla\eta+z\left[\dfrac{z}{2}\nabla(\nabla\cdot\boldsymbol{u}_t)+\nabla(\nabla\cdot(h\boldsymbol{u}_t))\right]+\boldsymbol{\tau}=0. \end{cases} \tag{1}$$

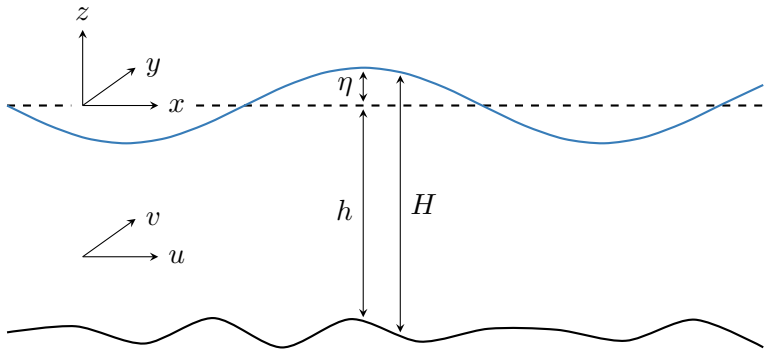

Figure 1: Definition of the quantities η, h, H, $\boldsymbol{u}$.

To simplify the framework, we consider here the 1D equations and ignore the bottom shear stress $\boldsymbol{\tau}$. We also consider a constant flat bottom $h=h_0$. Thus, the total height H can be recovered by the relation $H=h_0+\eta$. Then, we perform a linearization around the equilibrium point $(\bar{\eta},\bar{u})=(0,0)$ to get

$$\begin{cases} \eta_t+h_0u_x+\tilde{h}u_{xxx}=0, & \text{(2a)} \\ u_t+g\eta_x+\bar{h}u_{xxt}=0, & \text{(2b)} \end{cases}$$

to which we now refer as the *linearized Boussinesq equations*. Here, $\tilde{h}$ and $\bar{h}$ are constants defined by

$$\tilde{h}=\left(\frac{z^2}{2}+h_0z+\frac{h_0^2}{3}\right)h_0, \qquad \bar{h}=z\left(\frac{z}{2}+h_0\right). \tag{3}$$

Using the value $z=-0.53753\times h_0$ (the factor comes from the optimization of the dispersion properties of the equations, see [16] for more details) and $h_0=1$, we have that $\bar{h}$ and $\tilde{h}$ are both negative. Setting $\tilde{h}=0$, $\bar{h}=-\varepsilon$ a small parameter, $g=1$ and $h_0=1$ leads to the formulation

of the linearized Green-Naghdi equations, for which discrete transparent boundary conditions have been derived by Kazakova and Noble [13].

The first objective of the paper is to derive transparent boundary conditions for equations (2a) – (2b). Indeed, this system is set on the whole space $\mathbb{R}$ and thus we need to restrict the area of computation to a bounded domain for practical applications. This requires to find suitable boundary conditions. We focus on transparent boundary conditions in order to let waves escaping the domain without any reflection (a phenomenon that we observe for instance with Dirichlet conditions). From a mathematical point of view, we set the problem as follows: given a compactly supported initial data, one derives suitable conditions at the boundaries so that the solution on the bounded domain coincides with the restriction to this domain of the solution computed on the whole domain. In practice, this study can be done in either a continuous or discretized framework. A review of these techniques can be found in [1] where the authors build such conditions for the Schrödinger equation. In the linear case, the study in the continuous framework is carried out by applying the Laplace transform in time and adapt boundary conditions to keep solutions bounded. The Laplace-inverse transform of these conditions results into non-local in time operators. The adaptation of this technique to the discretized framework uses the $\mathcal{Z}$-transform, which is the discrete equivalent of the Laplace transform. Again, the numerical inversion of the obtained discrete conditions yields non-local operators.

The second and main objective of the paper is to test the efficiency of the discrete transparent conditions as interface conditions in a domain decomposition method with the alternating Schwarz method. The interest of this method lies in the possibilities, with few modifications in the original code, to split the original computational domain and/or couple different models (for instance farshore and nearshore). Our work can be seen as a step towards more efficient coupling between different numerical models in coastal engineering. The main difficulty is then to find suitable interface conditions to exchange information between the different subdomains so that the convergence is achieved as fast as possible.

This paper is organized as follows. In the first section, we apply the study from [3, 5, 13] to the linearized Boussinesq equations and obtain discrete transparent boundary conditions. Then, we provide some numerical tests to evaluate the accuracy of such conditions before implementing them in a domain decomposition method. In the second section, we briefly recall the idea of the additive Schwarz method that we are going to use before going on with the adaptation of transparent boundary conditions as interface conditions. We conclude with some numerical experiments to evaluate their efficiency. Finally, we present the tests we did with the nonlinear Serre equations to go on to the nonlinear framework for domain decomposition methods. The first two sections of the paper are available in a more detailed way in [7].

II – Derivation of transparent boundary conditions

First, we introduce the initial value problem that we seek to solve using transparent boundary conditions (TBC):

$$\begin{cases} \eta_t + h_0 u_x + \tilde{h} u_{xxx} = 0, \quad \forall\, x \in \mathbb{R}, \quad t > 0, \\ u_t + g\eta_x + \bar{h} u_{xxt} = 0, \quad \forall\, x \in \mathbb{R}, \quad t > 0, \\ u(x,0) = u_0(x), \quad \eta(x,0) = \eta_0(x), \quad \forall\, x \in \mathbb{R}, \\ u(x,t) \xrightarrow[x\to\pm\infty]{} 0, \quad \eta(x,t) \xrightarrow[x\to\pm\infty]{} 0, \quad \forall\, t > 0. \end{cases} \tag{4}$$

The goal of TBC is to find boundary conditions for a finite domain (for instance $[0, L]$) such that the solution on this domain coincides with the restriction to this domain of the solution of

problem (4). Hence, the problem we will work on is

$$\begin{cases} \eta_t + h_0 u_x + \tilde{h} u_{xxx} = 0, \quad \forall\, x \in [0, L], \quad t > 0, \\ u_t + g\eta_x + \bar{h} u_{xxt} = 0, \quad \forall\, x \in [0, L], \quad t > 0, \\ u(x, 0) = u_0(x), \quad \eta(x, 0) = \eta_0(x), \quad \forall\, x \in [0, L], \\ + \text{ transparent boundary conditions at } x = 0 \text{ and } x = L. \end{cases} \tag{5}$$

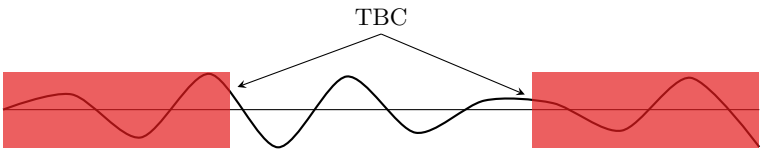

Figure 2: *Transparent* boundary conditions means that the solution we want to compute on $[0, L]$ is a *picture* of the solution on $\mathbb{R}$ restricted to $[0, L]$.

Note that it is possible to decouple equations (2a) – (2b) to obtain an equation on u only. Taking the cross derivatives of (2a) and (2b), we get

$$u_{tt} + \bar{h} u_{xxtt} - g h_0 u_{xx} - g\tilde{h} u_{xxxx} = 0. \tag{6}$$

Thus, system (2a) – (6) is equivalent to (2a) – (2b). We can also note a fourth order space derivative. Therefore, we will need to derive four conditions.

In this section, we follow the steps proposed in [3, 5, 13] to derive discrete TBC that are adapted to the discretized problem. We end this section with some numerical tests to analyze the efficiency of the obtained conditions.

II – 1 Deriving discrete transparent boundary conditions

We introduce here the derivation of *discrete* TBC (DTBC) for the discretized problem. These are boundary conditions directly derived from the discretized problem. We adapt here the method used in [3] for the Schrödinger equation, in [5] for the KdV equation and in [13] for the linearized Green-Naghdi equations. It follows four main steps:

1. Discretize the equations (2a) and (2b).
2. Use the $\mathcal{Z}$-transform on the problem on the complementary set.
3. Find the conditions at the two boundaries in the $\mathcal{Z}$-space.
4. Use the inverse $\mathcal{Z}$-transform to find the transparent boundary conditions.

Discretization

The first step is to find a discretization in time on a staggered grid. Let δx be the spatial-step and δt be the time-step. We build this grid such that $J = L/\delta x$ and $x_j = j\delta x$:

$$0 = x_0 < x_1 < x_2 < \cdots < x_{J-1} < x_J = L.$$

$j < 0$ and $j > J$ denote nodes that are out of the domain we want to work on. The time t^n then stands for $n\delta t$. Using a Crank-Nicolson scheme, (2a) yields

$$\begin{aligned} &\frac{\eta_{j+\frac{1}{2}}^{n+1} - \eta_{j+\frac{1}{2}}^{n}}{\delta t} + \frac{h_0}{2}\left(\frac{u_{j+1}^{n+1} - u_j^{n+1}}{\delta x} + \frac{u_{j+1}^{n} - u_j^{n}}{\delta x}\right) \\ &\qquad + \frac{\tilde{h}}{2}\left(\frac{u_{j+2}^{n+1} - 3u_{j+1}^{n+1} + 3u_j^{n+1} - u_{j-1}^{n+1}}{\delta x^3} + \frac{u_{j+2}^{n} - 3u_{j+1}^{n} + 3u_j^{n} - u_{j-1}^{n}}{\delta x^3}\right) = 0. \end{aligned} \tag{7}$$

Note that the finite differences operator used for the third spatial derivative is centered around $j+\frac{1}{2}$.

Discretizing (2b) is straightforward:

$$\begin{aligned}\frac{u_j^{n+1}-u_j^n}{\delta t}+\frac{g}{2}\left(\frac{\eta_{j+\frac{1}{2}}^{n+1}-\eta_{j-\frac{1}{2}}^{n+1}}{\delta x}+\frac{\eta_{j+\frac{1}{2}}^{n}-\eta_{j-\frac{1}{2}}^{n}}{\delta x}\right)\\ +\frac{\bar{h}}{\delta t}\left(\frac{u_{j+1}^{n+1}-2u_j^{n+1}+u_{j-1}^{n+1}}{\delta x^2}-\frac{u_{j+1}^{n}-2u_j^{n}+u_{j-1}^{n}}{\delta x^2}\right)=0.\end{aligned} \tag{8}$$

Problem on the complementary set

To solve the initial value problem (5), we assume that the initial conditions u_0 and η_0 are compactly supported in $[0,L]$. The derivation of the DTBC associated to our problem can be done by studying the problem on the complementary set of $[0,L]$:

$$\begin{cases}\eta_t+h_0u_x+\tilde{h}u_{xxx}=0, \quad \forall\ x\in\mathbb{R}\backslash[0,L], \quad t>0,\\ u_t+g\eta_x+\bar{h}u_{xxt}=0, \quad \forall\ x\in\mathbb{R}\backslash[0,L], \quad t>0,\\ u(x,0)=0, \quad \eta(x,0)=0, \quad \forall\ x\in\mathbb{R}\backslash[0,L],\\ u(x,t)\xrightarrow[x\to\pm\infty]{}0, \quad \eta(x,t)\xrightarrow[x\to\pm\infty]{}0, \quad \forall\ t>0.\end{cases} \tag{9}$$

$\mathcal{Z}$-transform

The second step is to compute the $\mathcal{Z}$-transform of (9). The $\mathcal{Z}$-transform of the sequence $(u_n)_{n\geqslant 0}$ is defined as a function of the complex variable z:

$$\forall\ |z|>R>0, \qquad \hat{u}(z)=\mathcal{Z}\left\{(u_n)\right\}:=\sum_{n\geqslant 0}u_nz^{-n}, \tag{10}$$

where R is the convergence radius of the series. By applying the $\mathcal{Z}$-transform to (7) and (8), we obtain the following recurrence (in space) relation for $\hat{u}$ after eliminating the dependence in $\hat{\eta}$:

$$\begin{aligned}-\frac{g\tilde{h}}{\delta x^4}\hat{u}_{j+2}+\left(\frac{\tilde{h}}{\delta x^2}s(z)^2-\frac{gh_0}{\delta x^2}+4\frac{g\tilde{h}}{\delta x^4}\right)\hat{u}_{j+1}+\left(s(z)^2\left(1-2\frac{\bar{h}}{\delta x^2}\right)+2\frac{gh_0}{\delta x^2}-6\frac{g\tilde{h}}{\delta x^4}\right)\hat{u}_j\\ +\left(\frac{\tilde{h}}{\delta x^2}s(z)^2-\frac{gh_0}{\delta x^2}+4\frac{g\tilde{h}}{\delta x^4}\right)\hat{u}_{j-1}-\frac{g\tilde{h}}{\delta x^4}\hat{u}_{j-2}=0,\end{aligned} \tag{11}$$

where

$$s(z)=\frac{2}{\delta t}\frac{z-1}{z+1}. \tag{12}$$

Finally, by solving this recurrence relation and finding conditions at the boundaries for $\hat{u}$ that we can translate back with the inverse $\mathcal{Z}$-transform, we find two conditions at each interface, with $\{u^n\}=\left(u^0,\dots,u^n\right)$:

$$\textbf{Left} \qquad \begin{cases}\Gamma_1^l\left(\{u^n\}\right):=u_0^n-(Y_5*u_1)^n+(Y_7*u_2)^n=0,\\ \\ \Gamma_2^l\left(\{u^n\}\right):=u_0^n-(Y_6*u_2)^n+2(Y_9*u_3)^n-(Y_8*u_4)^n=0,\end{cases} \tag{13}$$

$$\textbf{Right} \qquad \begin{cases}\Gamma_1^r\left(\{u^n\}\right):=u_J^n-(Y_1*u_{J-1})^n+(Y_3*u_{J-2})^n=0,\\ \\ \Gamma_2^r\left(\{u^n\}\right):=u_J^n-2(Y_1*u_{J-1})^n+(Y_2*u_{J-2})^n-(Y_4*u_{J-4})^n=0,\end{cases} \tag{14}$$

where (Y_i^n) for $i = 1, \dots, 9$ are coefficients that depend on the equation parameters and that can be computed offline before starting to solve: we don't need to recompute them at each time step. $*$ also denotes the discrete convolution:

$$\forall\ i \in [\![1, 9]\!], \quad (Y_i * u_j)^n := \sum_{m=0}^{n} Y_i^m u_j^{n-m}. \tag{15}$$

Hence, it is important to notice that these conditions *are not* local in time. For more details, and especially the expression of the (Y_i^n), one can refer to [7].

II – 2 Numerical results

We present here some numerical results to analyze the efficiency of the conditions we just derived. We used the following values for the different parameters we can vary ($\bar{h}$ and $\tilde{h}$ resulted from h_0 and z):

$$\begin{gathered} L = 1\,\mathrm{m}, \quad g = 9.81\,\mathrm{m}\cdot\mathrm{s}^{-2}, \quad \delta t = 0.001\,\mathrm{s}, \quad \delta x = 0.01\,\mathrm{m}, \\ h_0 = 1\,\mathrm{m}, \quad z = -0.53753 \times h_0, \quad \bar{h} = -0.39306\,\mathrm{m}^2, \quad \tilde{h} = -0.05973\,\mathrm{m}^3. \end{gathered} \tag{16}$$

For the initial conditions, we used

$$u_0 \equiv 0, \qquad \eta_0(x) = \exp\left(-400 \times \left(x - \frac{L}{2}\right)^2\right). \tag{17}$$

The results are presented in Figure 3. They are quite satisfying as the solution computed with DTBC fits well the reference solution computed on a larger domain. We used the error e^n as relative error at time t^n and e_T as global l^2 error in time:

$$e^n = \frac{\|u_{\mathrm{ref}}(\cdot, t^n) - u_{\mathrm{num}}(\cdot, t^n)\|_{l^2}}{\|u_{\mathrm{ref}}(\cdot, t^n)\|_{l^2}}, \qquad e_T = \sqrt{\delta t \times \sum_{n=1}^{n_{\max}} (e^n)^2}. \tag{18}$$

We used a trapezoidal rule to compute the discrete l^2-norm in space and the reference solution is computed on a larger domain, with the same transparent conditions. The error is expected to be small by definition of the *transparency* of the conditions. For $T = 1$, we get $e_T = 4.5 \cdot 10^{-6}$.

III – Application to domain decomposition methods

The discrete boundary conditions (13) and (14) will be used in this section as *interface* boundary conditions (IBC) in a domain decomposition method (DDM). We briefly describe the DDM that we are going to use before presenting analysis and numerical results.

III – 1 The Schwarz method

Domain decomposition methods are used to split a domain Ω, on which we want to solve a given problem, in multiple domains Ω_i, that can possibly overlap. Then, we can solve the problem in each domain. Hence, one must find functions that satisfy the PDE in each domain and that match with its neighbours on the interfaces, in a sense that has to be defined. The main difficulty of domain decomposition methods lies in the definition of efficient conditions at the interface between subdomains.

The original DDM was developed by Schwarz in 1870 [19] and consists in an iterative method: the solution in the i-th subdomain Ω_i is computed as the limit of a sequence u_i^k, $k \geqslant 0$. At each iteration k, we solve the problem in each subdomain with boundary conditions at the interfaces imposed using functions from the other subdomains. We will consider here the *additive* Schwarz

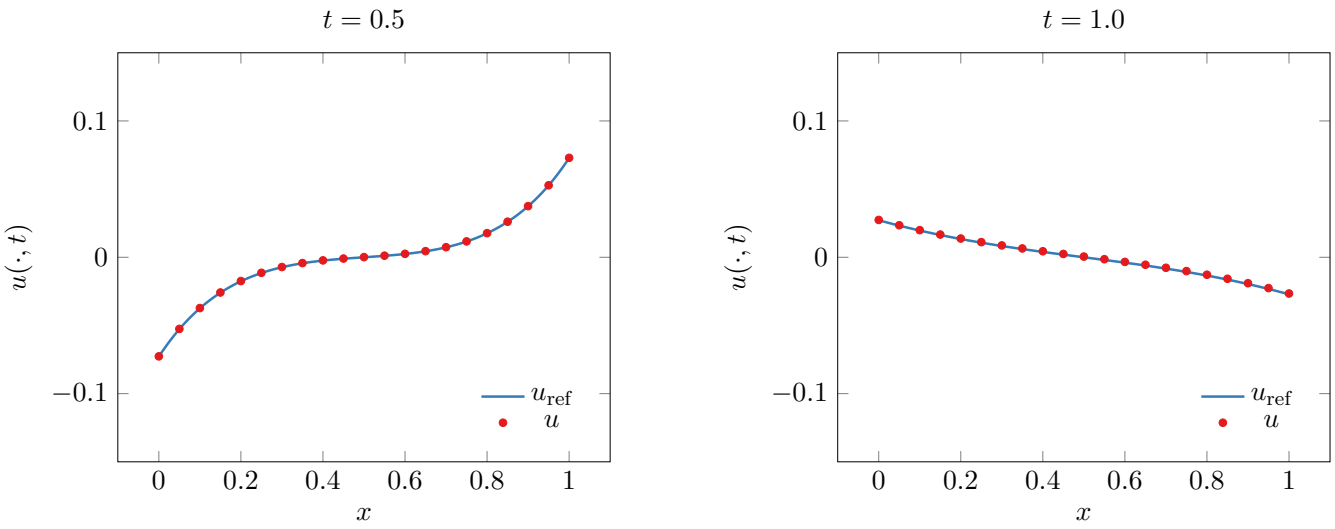

Figure 3: Snapshot at different times of the reference solution u_{ref} (computed on a larger domain with the transparent conditions at the boundaries) and the solution u computed with the same transparent conditions.

method (ASM) in which the interface conditions are always constructed using solutions u_j^{k-1} ($j \neq i$) from the previous step in the neighbour subdomains. Therefore, at each interface between two subdomains Ω_i and Ω_j, the IBC in Ω_i is

$$\mathcal{B}_i\left(u_i^{k+1}\right) = \mathcal{B}_i\left(u_j^{k}\right). \tag{19}$$

Note that it is possible to impose several interface conditions when Ω_i has several neighbours. Initially, the operators $\mathcal{B}_i$ were Dirichlet conditions: $\mathcal{B}_i(u) = u$. For more details on the Schwarz method, the reader can refer to [9, 14, 15].

We now look for an operator $\mathcal{B}_i$ inspired from the DTBC we derived in the previous section. Without loss of generality, we consider the domain $\Omega = [0, L]$ divided into two subdomains Ω_1 (left subdomain) and Ω_2 (right subdomain) that can possibly overlap. Ω is discretized into N nodes, while Ω_1 and Ω_2 are discretized into N_1 and N_2 nodes. The nodes in the subdomains coincide with the nodes in Ω. The interface conditions $\mathcal{B}_1$ and $\mathcal{B}_2$ will have to be such that:

- At each Schwarz iteration, there is a unique discrete solution $u_i^{n,k}$, at each time-step, in each subdomain.
- The solution in each subdomain will have to converge (in the sense of Schwarz) to the solution on Ω restricted to this subdomain.

Note that, as problem (5) is a time-dependent problem, we will perform the Schwarz method at each time-step. To avoid any confusion between the iteration in the Schwarz method and our time-dependent problem, the word *iteration* and the integer k will refer to the Schwarz algorithm whereas *time-step* and n, m will refer to the evolution in time of the problem we are solving.

III – 2 A Schwarz method with transparent conditions at the interface

From [12], we know that transparent boundary conditions are very good candidates for interface conditions in DDM. Thus, the TBC we derived in the previous section might inspire us to set up interface conditions in an additive Schwarz method. We recall that we look for *discrete* interface conditions, so that all the future reasonings will be done with the discrete equations and

conditions. If we focus on the right interface of the left domain, we recall that the transparent boundary conditions are

$$\begin{cases} \Gamma_1^r\left(\left\{u_1^{n+1}\right\}\right) = 0, \\ \Gamma_2^r\left(\left\{u_1^{n+1}\right\}\right) = 0. \end{cases} \tag{20}$$

A first heuristic is to use the left-hand-side with $u_2^{n+1,k}$ to provide a right-hand-side for the computation of $u_1^{n+1,k+1}$. With $\left\{u_1^{n+1}\right\}^k = \left(u_1^0, \ldots, u_1^n, u_1^{n+1,k}\right)$ where all the u_1^m for $0 \leqslant m \leqslant n$ have been computed by the Schwarz algorithm at the previous time-steps, we have

$$\begin{cases} \Gamma_1^r\left(\left\{u_1^{n+1}\right\}^{k+1}\right) = \Gamma_1^r\left(\left\{u_2^{n+1}\right\}^k\right), \\ \Gamma_2^r\left(\left\{u_1^{n+1}\right\}^{k+1}\right) = \Gamma_2^r\left(\left\{u_2^{n+1}\right\}^k\right). \end{cases} \tag{21}$$

Assuming we reached convergence at previous time-steps, we have $u_1^m = u_2^m$ at the nodes in the interface zone for $0 \leqslant m \leqslant n$. Hence, the values of u_1^m and u_2^m cancel each other and the IBC becomes (for the right boundary of the left domain)

$$\begin{cases} u_{1,J_1}^{n+1,k+1} - Y_1^0 \cdot u_{1,J_1-1}^{n+1,k+1} + Y_3^0 \cdot u_{1,J_1-2}^{n+1,k+1} = u_{2,J_{2,1}}^{n+1,k} - Y_1^0 \cdot u_{2,J_{2,1}-1}^{n+1,k} + Y_3^0 \cdot u_{2,J_{2,1}-2}^{n+1,k}, \\ u_{1,J_1}^{n+1,k+1} - 2Y_1^0 \cdot u_{1,J_1-1}^{n+1,k+1} + Y_2^0 \cdot u_{1,J_1-2}^{n+1,k+1} - Y_4^0 \cdot u_{1,J_1-4}^{n+1,k+1} = u_{2,J_{2,1}}^{n+1,k} - 2Y_1^0 \cdot u_{2,J_{2,1}-1}^{n+1,k} + Y_2^0 \cdot u_{2,J_{2,1}-2}^{n+1,k} - Y_4^0 \cdot u_{2,J_{2,1}-4}^{n+1,k}, \end{cases} \tag{22}$$

where $J_{2,1} = N_1 + N_2 - N - 1$ is the index such that the node $J_{2,1}$ of the right domain coincides with the node $J_1 = N_1 - 1$ of the left domain. The operator defined on each side of this system will be denoted as $\mathcal{B}_1$, so that the IBC becomes

$$\mathcal{B}_1\left(u_1^{n+1,k+1}\right) = \mathcal{B}_1\left(u_2^{n+1,k}\right). \tag{23}$$

It is worth noting that the operator (23) used in (22) is local in time, contrarily to the operators $\Gamma_{1,2}^r$. Similarly, the IBC for the left boundary of the right domain is

$$\begin{cases} u_{2,0}^{n+1,k+1} - Y_5^0 \cdot u_{2,1}^{n+1,k+1} + Y_7^0 \cdot u_{2,2}^{n+1,k+1} = u_{1,O_{1,2}}^{n+1,k} - Y_5^0 \cdot u_{1,O_{1,2}+1}^{n+1,k} + Y_7^0 \cdot u_{1,O_{1,2}+2}^{n+1,k}, \\ u_{2,0}^{n+1,k+1} - Y_6^0 \cdot u_{2,2}^{n+1,k+1} + 2Y_9^0 \cdot u_{2,3}^{n+1,k+1} - Y_8^0 \cdot u_{2,4}^{n+1,k+1} = u_{1,O_{1,2}}^{n+1,k} - Y_6^0 \cdot u_{1,O_{1,2}+2}^{n+1,k} + 2Y_9^0 \cdot u_{1,O_{1,2}+3}^{n+1,k} - Y_8^0 \cdot u_{1,O_{1,2}+4}^{n+1,k}, \end{cases} \tag{24}$$

where $J_2 = N_2 - 1$ and $O_{1,2} = N - N_2$ is the index such that the node $O_{1,2}$ of the left domain coincides with the node 0 of the right domain. Again, we rewrite this system as

$$\mathcal{B}_2\left(u_2^{n+1,k+1}\right) = \mathcal{B}_2\left(u_1^{n+1,k}\right). \tag{25}$$

Remark 1. Let us notice that the interface conditions need at least 5 nodes in the overlap zone and that it does not allow a non-overlapping ASM. It is not really an issue as the number of nodes required in the overlap zone is small and does not depend on the mesh size.

Finally, the ASM along with these interface conditions reads

$$\begin{cases} \Gamma_1^l\left(\left\{u_1^{n+1}\right\}\right) = 0, \quad \Gamma_2^l\left(\left\{u_1^{n+1}\right\}\right) = 0, & \text{left}, \\ u_1^{n+1,k+1} = f_1\left(u_1^n, u_1^{n-1}\right), & \text{interior}, \\ \mathcal{B}_1\left(u_1^{n+1,k+1}\right) = \mathcal{B}_1\left(u_2^{n+1,k}\right), & \text{right}, \end{cases} \qquad \begin{cases} \mathcal{B}_2\left(u_2^{n+1,k+1}\right) = \mathcal{B}_2\left(u_1^{n+1,k}\right), & \text{left}, \\ u_2^{n+1,k+1} = f_2\left(u_2^n, u_2^{n-1}\right), & \text{interior}, \\ \Gamma_1^r\left(\left\{u_2^{n+1}\right\}\right) = 0, \quad \Gamma_2^r\left(\left\{u_2^{n+1}\right\}\right) = 0, & \text{right}, \end{cases} \tag{26}$$

where f_i stands for the discrete equation from the numerical scheme applied to the interior of domain i.

III – 3 Numerical results

We end this section with numerical results obtained with the implementation of an additive Schwarz method to solve the linearized Boussinesq equations. We used the same parameters than in the previous tests (16). What we are interested in here is the number of Schwarz iterations required by the DDM (26) to converge to the reference solution (given by the solution computed on the domain Ω with discrete transparent conditions at the boundaries). We will use the following stopping criterion:

$$e_{\text{DDM}}^{n,k} \leqslant \varepsilon, \tag{27}$$

where $\varepsilon = 10^{-12}$ and

$$e_{\text{DDM}}^{n,k} = \sqrt{\delta x \left(\sum_{j=0}^{J_1} \left(u_{\text{ref},j}^n - u_{1,j}^{n,k} \right)^2 + \sum_{j=0}^{J_2} \left(u_{\text{ref},O_{1,2}+j}^n - u_{2,j}^{n,k} \right)^2 \right)}. \tag{28}$$

Again, $O_{1,2}$ is the index on the mono-domain of the node corresponding to the first node of the right domain. To study the efficiency of this method, we analyse its convergence at fixed points in time and compare our conditions at the interface to the classical Dirichlet conditions.

Results are presented in Table 1 for the minimum overlap size. They are very satifying as we can see that our interface conditions make the Schwarz algorithm converge in 2 iterations only, where the Dirichlet interface conditions require several hundreds of iterations. As these conditions correspond to the fastest possible convergence (2 iterations), it is not necessary to study the influence of the size of the overlap zone nor the efficiency of a global Schwarz algorithm as it will not converge in less than 2 iterations.

t	TBC	Dirichlet
0.25	2	604
0.5	2	612
0.75	2	546
1.0	2	556

Table 1: Number of iterations required for convergence in the ASM. We used, for the subdomains, $N_1 = 88$ and $N_2 = 18$ so that the overlap zone is of size 5 (we have $N = 101$ from (16)).

IV – Towards transparent boundary conditions for nonlinear problems

Considering the efficiency of these discrete transparent boundary conditions when used at the interface in a domain decomposition method, we thought that the next step would be to use them for solving nonlinear equations. For convenience and by lack of time, we chose to consider the nonlinear Serre equations as a nonlinear solver was already available to us and we could focus on the implementation and the test of the boundary conditions. Applying the same ideas to the nonlinear Boussinesq equations is subject to further study.

IV – 1 Serre equations

The Serre equations are used to describe the propagation of strongly nonlinear waves in shallow water. In 1D, with a flat bottom, these equations have the following form in the variables (h, u):

$$\begin{cases} h_t + (hu)_x = 0, \\ u_t + uu_x + gh_x - \dfrac{1}{3h} \left(h^3 \left(u_{xt} + uu_{xx} - (u_x)^2 \right) \right)_x = 0. \end{cases} \tag{29}$$

This formulation can be found in [8]. The goal here is to derive TBC for these equations by linearizing them and then use the coefficients of the linearized equations as boundary conditions for the nonlinear equation. After linearisation, the exact same work that we did in the previous sections can be followed and we obtain similar discrete boundary conditions, except that this time the convolution coefficients (Y_i^n) are different because the equations are different. Moreover, when using the conditions in a domain decomposition method, it happens that they are exactly the same than (22) and (24), just with different coefficients. The details of the computations are not exposed here.

IV – 2 Numerical results

Transparent boundary conditions

In this paragraph, the goal is to implement and test the TBC as boundary conditions. To this end, we ran the solver twice: once on a big domain and once on a small domain. Results are available in Figure 4 and Figure 5 below. We used two different sets of initial conditions and parameters:

1. A solitary solution, described in [8],

$$h(x,t) = a_0 + a_1 \mathrm{sech}^2(\kappa(x-ct)), \qquad u(x,t) = c\left(1 - \frac{a_0}{h(x,t)}\right), \tag{30}$$

with the following parameters:

$$\begin{gathered} h_0 = a_0 = 1\,\mathrm{m}, \quad a_1 = 0.1\,\mathrm{m}, \quad \delta t = 0.05\,\mathrm{s}, \quad \delta x = 0.757\,\mathrm{m}, \\ g = 9.81\,\mathrm{m}\cdot\mathrm{s}^{-2}, \quad \kappa = \frac{\sqrt{3a_1}}{2a_0\sqrt{a_0+a_1}}, \quad c = \sqrt{g\,(a_0+a_1)}. \end{gathered} \tag{31}$$

2. A gaussian distribution

$$h(x,0) = h_0 + 0.1 \times \exp\left(-400 \times \left(x - \frac{L}{2}\right)^2\right), \qquad u(x,0) \equiv 0, \tag{32}$$

with the following parameters

$$h_0 = 1\,\mathrm{m}, \quad L = 3\,\mathrm{m}, \quad \delta t = 0.001\,\mathrm{s}, \quad \delta x = 0.06\,\mathrm{m}, \quad g = 9.81\,\mathrm{m}\cdot\mathrm{s}^{-2}. \tag{33}$$

It is not surprising to see that the TBC are not totally transparent as we used convolution coefficients that were computed from the linearized version of the nonlinear Serre equations. What is more striking is that the convolution coefficients computed from the discretized Boussinesq equations are more efficient than the one from the Serre equations, even if both do not seem to work as some reflection is observed. This issue is still under investigation.

Domain decomposition

Here again we measured the consistency error (28) generated by the DDM. Results are presented in Table 2. We can see that if the solitary wave seems to be a case were both interface conditions are pretty efficient, the gaussian case is solved faster with the TBC than with Dirichlet as interface conditions. Again, using the Y_i coefficients from the Boussinesq equations seems to be more efficient than using the coefficients from the linearized Serre equations.

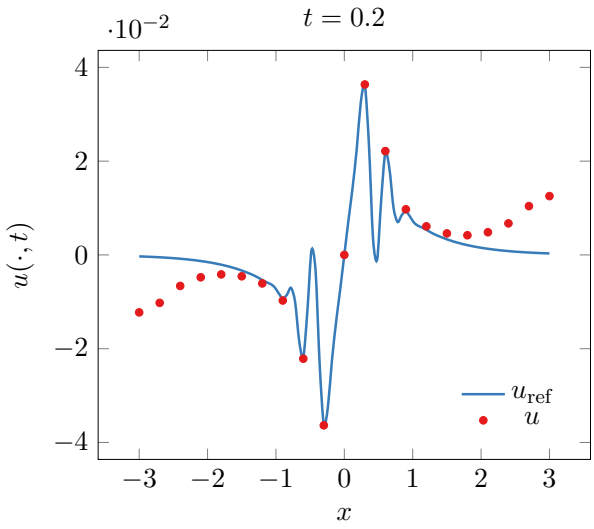

Figure 4: Discrete transparent boundary conditions for the gaussian case, with the Y_i coefficients computed from the linearized Boussinesq equation.

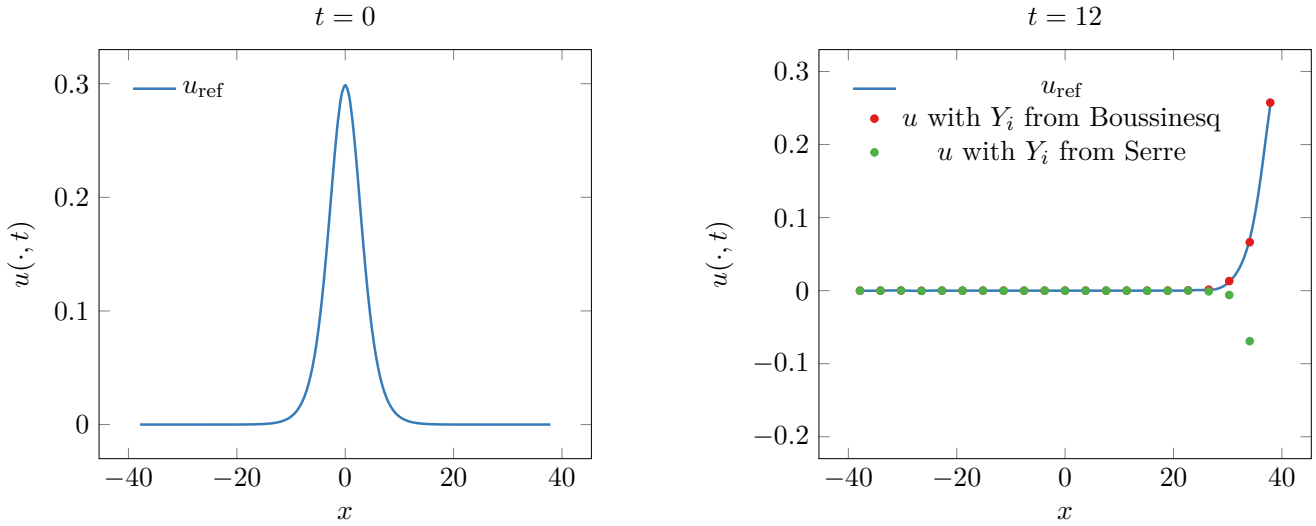

Figure 5: Discrete transparent boundary conditions for the solitary wave. We computed the coefficient for the convolutions in two different ways. (Left) Initial condition. (Right) Y_i computed with the linearized Serre equations *vs* Y_i computed with the linearized Boussinesq equations.

Case	**Dirichlet**	**TBC**	
		Serre	**Boussinesq**
Gaussian	$120 \sim 160$	18	9
Solitary	11	8	8

Table 2: Convergence of the additive Schwarz method, for a precision of $\varepsilon = 10^{-15}$, for an overlap size of 5.

V – Conclusion

In this paper, we have been using discrete transparent boundary conditions for a class of Boussinesq equations. As expected these conditions (non-local in time) provide very satisfying results with respect to the wave reflection at boundaries. When implemented in a domain decomposition framework, the new conditions happen to be both very efficient and local in time, which provides the best possible framework for the simulation in large domains using decomposition techniques. Adapting this idea to nonlinear equations gives less impressive results but still, the convergence of the domain decomposition has been enhanced.

References

[1] X. Antoine, A. Arnold, C. Besse, M. Ehrhardt, and A. Schaedle. A Review of Transparent and Artificial Boundary Conditions Techniques for Linear and Nonlinear Schrödinger Equations. *Communications in Computational Physics*, 4:729–796, 2008.

[2] A. Arnold. Numerically Absorbing Boundary Conditions for Quantum Evolution Equations. *VLSI Design*, 1998.

[3] A. Arnold and M. Ehrhardt. Discrete Transparent Boundary Conditions for the Schrödinger Equation. *Rivista di Matematica della Università di Parma*, 6, 2001.

[4] A. Arnold, M. Ehrhardt, and I. Sofronov. Discrete transparent boundary conditions for the Schrodinger equation: Fast calculation, approximation, and stability. *Communications in Mathematical Sciences*, 1, 2003.

[5] C. Besse, M. Ehrhardt, and I. Lacroix-Violet. Discrete Artificial Boundary Conditions for the Korteweg-de Vries Equation. *Numerical Methods for Partial Differential Equations*, 2015.

[6] J. G. Caldas Steinstraesser, R. Cienfuegos, J. D. Galaz Mora, and A. Rousseau. A Schwarz-based domain decomposition method for the dispersion equation. 2017. `https://hal.inria.fr/hal-01617692/document`, accepted in *Journal of Applied Analysis and Computation*.

[7] J. G. Caldas Steinstraesser, G. Kemlin, and A. Rousseau. A domain decomposition method for linearized Boussinesq-type equations. 2018. Preprint, available at `https://hal.inria.fr/hal-01797823`.

[8] R. Cienfuegos and J. Carter. The kinematics and stability of solitary and cnoidal wave solutions of the Serre equations. *European Journal of Mechanics B - Fluids*, 30(3):259–268, 2011.

[9] M. J. Gander. Schwarz methods over the course of time. *Electronic Transactions on Numerical Analysis*, 31:228–255, 2008.

[10] J.-F. Gerbeau and B. Perthame. Derivation of Viscous Saint-Venant System for Laminar Shallow Water; Numerical Validation. *Discrete and Continuous Dynamical Systems - Series B*, 1(1):89–102, 2000.

[11] A. E. Green and P. M. Naghdi. A derivation of equations for wave propagation in water of variable depth. *Journal of Fluid Mechanics*, 78(2):237–246, 1976.

[12] C. Japhet and F. Nataf. The Best Interface Conditions for Domain Decomposition Methods: Absorbing Boundary Conditions. In *Absorbing Boundaries and Layers, Domain Decomposition Methods: Applications to Large Scale Computations*, pages 348–373, New York, 2001. L. Tourrette and L. Halpern (eds.), Nova Science Publisher, Inc.

[13] M. Kazakova and P. Noble. Discrete transparent boundary conditions for the linearized Green-Naghdi system of equations. 2017. `https://arxiv.org/pdf/1710.04016.pdf`.

[14] P.-L. Lions. On the Schwarz Alternating Method I. In *First International Symposium on Domain Decomposition Methods for Partial Differential Equations*, pages 1 – 42, Philadelphia, 1988. R. Glowinski, G.H. Golub, G.A. Meurant, J. Périaux (eds.), SIAM.

[15] P.-L. Lions. On the Schwarz Alternating Method III: A variant for Nonoverlapping Subdomains. In *Third International Symposium on Domain Decomposition Methods for Partial Differential Equations*, pages 202–223, Houston, 1989. T.F. Chan, R. Glowinski, J. Périaux, O. Widlund (eds.), SIAM.

[16] O. Nwogu. Alternative Form of Boussinesq Equations for Nearshore Wave Propagation. *Journal of Waterway, Port, Coastal, and Ocean Engineering*, 119(6):618–638, 1993.

[17] D. Peregrine. Long waves on a beach. *Journal of Fluid Mechanics*, 27:815–827, 1967.

[18] V. Roeber and K. F. Cheung. Boussinesq-type model for energetic breaking waves in fringing reef environment. *Coastal Engineering*, 70:1–20, 2012.

[19] H. A. Schwarz. Ueber einen Grenzübergang durch alternirendes Verfahren. *Vierteljahrsschrift der Naturforschenden Gesellschaft in Zürich*, 15:272–286, 1870.