Journal of Computational Physics 259 (2014) 220–241

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

The Double Absorbing Boundary method

Thomas Hagstrom [a], Dan Givoli [b,*], Daniel Rabinovich [b], Jacobo Bielak [c]

[a] Department of Mathematics, Southern Methodist University, Dallas, TX 75275, USA
[b] Department of Aerospace Engineering, Technion — Israel Institute of Technology, Haifa 32000, Israel
[c] Department of Civil and Environmental Engineering, Carnegie Mellon University, Pittsburgh, PA 15213, USA

ARTICLE INFO

Article history:
Received 23 April 2013
Received in revised form 20 November 2013
Accepted 23 November 2013
Available online 3 December 2013

Keywords:
Double absorbing boundary
Absorbing boundary condition
High-order
Perfectly matched layer
Auxiliary variables
Waves
Artificial boundary
Finite differences
Finite elements

ABSTRACT

A new approach is devised for solving wave problems in unbounded domains. It has common features to each of two types of existing techniques: local high-order Absorbing Boundary Conditions (ABC) and Perfectly Matched Layers (PML). However, it is different from both and enjoys relative advantages with respect to both. The new method, called the Double Absorbing Boundary (DAB) method, is based on truncating the unbounded domain to produce a finite computational domain Ω, and on applying a local high-order ABC on *two* parallel artificial boundaries, which are a small distance apart, and thus form a thin non-reflecting *layer*. Auxiliary variables are defined on the two boundaries and inside the layer bounded by them, and participate in the numerical scheme. The DAB method is first introduced in general terms, using the 2D scalar time-dependent wave equation as a model. Then it is applied to the 1D Klein–Gordon equation, using finite difference discretization in space and time, and to the 2D wave equation in a wave guide, using finite element discretization in space and dissipative time stepping. The computational aspects of the method are discussed, and numerical experiments demonstrate its performance.

1. Introduction

Wave problems in unbounded or very large domains appear often in various important applications such as underwater acoustics, electromagnetic scattering and solid earth geophysics, to name just a few. One large class of computational methods to handle such problems is based on truncating the unbounded domain by introducing an artificial boundary, thus defining a finite computational domain of interest. Some numerical technique is applied on or near the artificial boundary, to minimize spurious reflection of outgoing waves.

Since the mid 1990s, two powerful techniques have emerged in this context: the use of a high-order Absorbing Boundary Condition (ABC) and the use of a Perfectly Matched Layer (PML). Research on these methods is still very active. In fact, it is remarkable that finding an excellent computational tool to deal with the unboundedness of wave problems, which has started in the early 1970s, is still not a "closed problem". A search in the article archive ISI shows that during the last 5 years, more than 150 papers were published with the words ABC or PML in the title, and more than 1000 published papers indicated ABC or PML as keywords. What stands behind this is the fact that the computational tool sought must be accurate, stable, efficient and easy to implement, and should apply in various configurations and for various governing equations. All the existing computational techniques for unbounded wave problems can still use improvements in some of these aspects. See the review papers [1–4].

* Corresponding author. Tel.: +1 972 829 3814; fax: +1 972 829 2030.
E-mail addresses: thagstrom@smu.edu (T. Hagstrom), givolid@aerodyne.technion.ac.il (D. Givoli), daniel@aerodyne.technion.ac.il (D. Rabinovich), jbielak@cmu.edu (J. Bielak).

0021-9991/$ – see front matter
http://dx.doi.org/10.1016/j.jcp.2013.11.025

The first PML was devised by Bérenger [5] in 1994 for electromagnetic waves, and since then further developed, analyzed and used in various applications by many authors. See references in the review papers [3,6]. The PML is a layer adjacent to the artificial boundary, in which the governing equations are artificially modified. It possesses two properties at the continuous level: (a) there is a perfect match between the layer and the interior domain, namely any outgoing plane wave produces zero reflection; and (b) the solution decays exponentially when it travels inside the layer. These two properties theoretically guarantee excellent performance of the PML. What may sometimes hamper this theoretical performance is the sensitivity of the PML to the discretization and the need to introduce ad-hoc damping and stretching profiles.

The first high-order ABC was devised by Collino [7] in 1993, and a few other formulations followed by other authors. See references in the review paper [8]. High-order ABCs are local in space and time, like the classical ABCs of Engquist and Majda [9] and Bayliss and Turkel [10], but unlike those, they do not involve high derivatives. Therefore they can be implemented in practice up to any desired order, as opposed to the classical ones that have been implemented up to second order only. In the high-order ABC scheme, the order of the ABC is simply an input parameter. The high derivatives that initially appear when designing a high-order ABC are eliminated by introducing auxiliary variables ϕ_j on the boundary.

Recently the two approaches — high-order ABC and PML — have been compared, theoretically and numerically [4,11], in the frequency domain. They were found to be equally effective, with some relative advantages for both. In fact, although usually derived by very different approaches, recent work has shown that, at the discrete level, the two methods are quite closely related. In particular, it is shown in [12] how to design a non-standard PML with a purely imaginary mesh continuation to exactly annihilate propagating waves at any incidence angle. This non-standard PML is formally equivalent to the high-order ABC proposed by Hagstrom and Warburton [13].

As mentioned above, each of the two classes of techniques has relative advantages. One major disadvantage of high-order ABCs is that they require special treatment at *corners* formed by the intersection of two flat segments of the artificial boundary, and in some cases also at corners between an artificial and a physical boundary. Such special treatment is sometimes cumbersome or even difficult to devise. In contrast, handling corners with PMLs is usually straight forward. Another disadvantage of high-order ABCs is that they are constrained not to include any *normal derivative* of an auxiliary variable ϕ_j on the boundary, since the ϕ_j are discretized in practice only on the boundary. Thus, the ABC is allowed to involve only tangential and temporal derivatives of the ϕ_j. Eliminating the normal derivatives from the ABC operators is sometimes difficult and may require a lot of algebra; a case in point is elastodynamics [14]. PML is also usually easier to incorporate in an existing numerical code.

On the other hand, an important disadvantage of PML is that it is not associated with a clear notion of convergence, except under the expensive scenario of widening a layer where all physical and auxiliary fields are well-resolved. By contrast, in the case of high-order ABCs, with a fixed location of the boundary, one can approach the exact solution arbitrarily closely (up to the discretization error) by increasing the order P of the ABC (with cost that increases only linearly with P). More efficient underresolved PMLs seem to be more sensitive to discretization and to the computational parameters than ABCs. A good design of an ABC at the continuous level usually guarantees good performance at the discrete level. This does not seem to be the general case for PML, where the matching between the solutions in the interior and in the layer at the discrete level is sometimes far from perfect. In addition, the theoretical analysis of a PML is usually more difficult than that for a high-order ABC for the same application. Additional discussion on the comparison of the two types of methods can be found in [3,4,11].

In this paper we present a new method, which shares some features of both the PML and the high-order ABC, but enjoys some of the advantages that each of them lacks. In the new method, called the Double Absorbing Boundary method, or simply DAB, a high-order ABC is applied on *two* parallel artificial boundaries, which are a small distance apart. Auxiliary variables are defined on the two boundaries and in the thin layer bounded by them. Like the PML, the DAB does not require special treatment of corners. The algebra involved is quite simple, since no elimination of normal derivatives is needed. Like in the method of high-order ABCs on a single boundary, DAB is clearly associated with the notion of convergence; one can approach the exact solution arbitrarily closely (up to the discretization error) by increasing the order P, with only linearly-increasing cost. The numerical properties of DAB, like accuracy, stability and sensitivity to discretization, are similar to those of a high-order ABC on a single boundary.

Although DAB is a general approach, and in principle can be used with any high-order ABC applied on the double boundary, here we consider ABCs of the form proposed by Hagstrom and Warburton (H-W) in [13,15]. The ABC formulation in [15], called the Complete Radiation Boundary Condition (CRBC), generalizes that in [13], and leads to an almost uniform-in-time error estimate for both propagating and decaying waves. The H-W ABCs have been incorporated in both finite element and finite difference schemes, and have been shown to be extremely effective in a variety of situations, including those associated with dispersive, stratified, anisotropic and convective media [16–18], and where exterior sources are present (nesting) [19]. Recently a H-W type ABC was applied to problems in elastodynamics [20,21]. This is the first known high-order ABC for elastodynamics that is long-time stable.

Here is an outline of the remaining sections. In Section 2 we introduce DAB in general terms, using the 2D scalar time-dependent dispersive wave equation (Klein–Gordon equation) as a model. In doing so we comment on the comparison of DAB to existing methods. We analyze the accuracy and stability of DAB in Section 3. In Section 4 we apply DAB to the 1D dispersive wave equation, which is perhaps the simplest (but non-trivial) model that still brings to light the main properties of the method, and allows us to discuss computational aspects in detail. We use Finite Difference (FD) discretization in space and time. We present some numerical experiments for this problem in Section 5, which demonstrate the performance of the

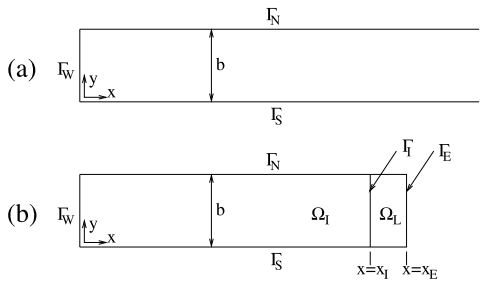

Fig. 1. A semi-infinite waveguide: (a) the original setup, (b) the setup of the problem with truncated domain.

method in 1D. In Section 6 we apply DAB to the 2D wave equation in a wave guide, using Finite Element (FE) discretization in space and dissipative time stepping. This is followed by numerical experiments for the 2D case in Section 7. We end with concluding remarks in Section 8.

2. The Double Absorbing Boundary (DAB) method

2.1. The original unbounded-domain problem

To present the idea underlying DAB, we consider a two-dimensional semi-infinite acoustic wave guide of width b, as shown in Fig. 1(a). A Cartesian coordinate system (x, y) is introduced with the origin at the southwest corner, so that the waveguide is parallel to the x direction, and $y \in [0, b]$. The south, north and west boundaries of the waveguide are denoted Γ_S, Γ_N and Γ_W, respectively.

In the waveguide we consider the dispersive wave equation (Klein–Gordon)

$$\mathcal{W}u \equiv \ddot{u} - c^2\nabla^2 u + su = f. \tag{1}$$

This can be thought of as a model for a membrane on an elastic foundation. Here $c(x, y) > 0$ is the medium wave speed, $s(x, y) \geqslant 0$ is the medium dispersion coefficient (the stiffness of the elastic foundation), and $f(x, y, t)$ is a given source. A superposed dot indicates differentiation with respect to time. Some boundary condition is given on the three boundaries. For simplicity of the presentation, we take the zero Dirichlet condition

$$u = 0 \quad \text{on } \Gamma_S, \Gamma_N, \Gamma_W. \tag{2}$$

Initial conditions, i.e.,

$$u(x, y, t = 0) = u_0(x, y), \tag{3}$$

$$\dot{u}(x, y, t = 0) = \dot{u}_0(x, y), \tag{4}$$

are given, with known functions u_0 and $\dot{u}_0$, which we assume satisfy the wall boundary conditions.

Inside a compact region, denoted Ω_0, the functions c, s, f, u_0 and $\dot{u}_0$ may, in principle, be general. Outside this region, the following simplified conditions hold: (a) the medium is *homogeneous*, namely c and s are constant; (b) there are no sources, namely $f = 0$, and so (1) becomes $\mathcal{W}u = 0$; and (c) the initial values vanish, namely $u_0 = 0$ and $\dot{u}_0 = 0$.

2.2. The problem in the computational domain

We now truncate the semi-infinite domain by introducing the artificial boundary Γ_E, located at $x = x_E$, $0 \leqslant y \leqslant b$. Slightly to the west of Γ_E we set an interface denoted Γ_I, located at $x = x_I$, $0 \leqslant y \leqslant b$. See Fig. 1(b). The entire computational domain bounded by $\Gamma_N \cup \Gamma_W \cup \Gamma_S \cup \Gamma_E$ is denoted Ω. As Fig. 1(b) shows, this domain is divided by the interface Γ_I into two sub-domains: the interior domain Ω_I and a thin *layer* Ω_L. We choose the location of Γ_E and Γ_I such that Ω_0 is strictly contained within Ω_I. Thus, in the layer Ω_L we have that c and s are constant, $f = 0$, and the initial values are zero.

The function u satisfies the Klein–Gordon equation (1) in Ω, the three "wall" boundary conditions (2), and the initial conditions (3) and (4) in Ω. In the layer Ω_L we shall apply a special treatment, with the goal of rendering the solution in the interior domain Ω_I as close as possible to the solution of the original semi-infinite problem in that domain. Thus, Ω_L will act as an absorbing or non-reflecting layer.

To this end we define a sequence of auxiliary variables $\phi_0, \ldots, \phi_{P+1}$ *within the layer domain*. Here P is a chosen parameter that will determine the order of accuracy of the absorbing layer. Each $\phi_j = \phi_j(x, y, t)$ is defined only in the layer Ω_L. The problem for the ϕ_j is given as follows. In the layer, we require each ϕ_j to satisfy the same wave equation as for u, i.e. (1) (with $f = 0$):

$$\mathcal{W}\phi_j \equiv \ddot{\phi}_j - c^2\nabla^2\phi_j + s\phi_j = 0, \quad j = 0, \ldots, P+1, \quad \text{in } \Omega_L. \tag{5}$$

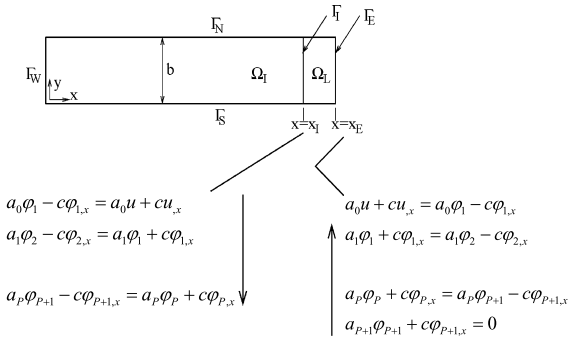

Fig. 2. The "ladder" structure of the DAB, showing the flow of information on the two boundaries bounding the layer. The parameters σ_j were taken to be zero for simplicity.

All the auxiliary variables satisfy zero initial conditions:

$$\phi_j(x, y, t = 0) = 0, \qquad \dot{\phi}_j(x, y, t = 0) = 0, \quad j = 0, \ldots, P + 1, \quad \text{in } \Omega_L. \tag{6}$$

On Γ_S and Γ_N (or more precisely the portions of these boundaries in the layer, i.e., for $x_I \leqslant x \leqslant x_E$ and $y = 0, b$) we apply the same boundary conditions as for u, i.e.,

$$\phi_j = 0, \quad j = 0, \ldots, P + 1, \quad \text{on } \Gamma_S, \Gamma_N. \tag{7}$$

Now we need to define boundary conditions for the ϕ_j on Γ_I and on Γ_E. We will define these boundary conditions in several steps. First, on both boundaries we require the ϕ_j to satisfy the following recursive relations:

$$a_j\dot{\phi}_j + c\phi_{j,x} + \sigma_j\phi_j = \bar{a}_j\dot{\phi}_{j+1} - c\phi_{j+1,x} + \bar{\sigma}_j\phi_{j+1}, \quad j = 0, \ldots, P, \quad \text{on } \Gamma_I, \Gamma_E. \tag{8}$$

Here and elsewhere a comma denotes partial differentiation. The parameters a_j, $\bar{a}_j$, σ_j, and $\bar{\sigma}_j$ satisfy:

$$0 < a_j, \bar{a}_j \leqslant 1, \qquad \sigma_j, \bar{\sigma}_j \geqslant 0. \tag{9}$$

They can be chosen to minimize the reflection coefficient under various assumptions on the simulation time, source location, or frequency content. See, e.g., [15]. Second, on Γ_I we require ϕ_0 and u to coincide in value and slope, i.e.,

$$\phi_0 = u, \qquad \phi_{0,x} = u_{,x} \quad \text{on } \Gamma_I. \tag{10}$$

From these continuity conditions and the fact that both u and ϕ_0 satisfy the same wave equation in Ω_L, it is easy to deduce that ϕ_0 is simply another name for u in the layer, i.e.,

$$\phi_0 \equiv u \quad \text{in } \Omega_L. \tag{11}$$

Finally, on Γ_E we require the "termination condition"

$$a_{P+1}\dot{\phi}_{P+1} + c\phi_{P+1,x} + \sigma_{P+1}\phi_{P+1} = 0 \quad \text{on } \Gamma_E. \tag{12}$$

Remark 1. The relations (8) constitute a high-order ABC (see, e.g., [14,20]). Thus, we impose the ABC on both Γ_I and Γ_E, and therefore the name "Double Absorbing Boundary" for the proposed method.

Remark 2. As we shall see later, on Γ_I we shall regard (8) as a condition for ϕ_{j+1} given ϕ_j. On Γ_E it will be the other way around; we shall regard (8) as a condition for ϕ_j given ϕ_{j+1}. Fig. 2 is an illustration of the "ladder" structure of the DAB, showing the set of boundary conditions applied on each of the two boundaries bounding the layer. For the sake of compactness of the illustration, the parameters σ_j were taken to be zero here, so that the recursive relations do not include an undifferentiated term.

Remark 3. The left (Dirichlet) condition in (10) should be viewed as a condition for ϕ_0 given u, whereas the right (Neumann) condition in (10) should be viewed as a condition for u given ϕ_0. Thus, the latter boundary condition returns the information from the auxiliary variable ϕ_0 in the layer Ω_L to the primary solution u in the interior Ω_I. This is reminiscent of a Dirichlet–to–Neumann (DtN) boundary condition; see, e.g., [22].

Remark 4. The problem for the ϕ_j's ($j = 0, \ldots, P + 1$) consists of $P + 2$ wave equations in Ω_L, i.e., Eq. (5). Therefore we need $P + 2$ boundary conditions on each boundary of Ω_L. This is exactly what Eqs. (7)–(12) provide us with. (Remark 3 should be taken into account in the counting.)

Remark 5. In our previous publications (see, e.g., [14,20]), we introduced recursive relations of the form (8) in the vicinity of an artificial boundary, and then *proved* that the auxiliary variables ϕ_j satisfy the wave equation (5). In the DAB method we impose the recursive relations (8) only on the two boundaries of the layer Ω_L, and we *require* that the ϕ_j satisfy (5) inside this layer.

Remark 6. In our previous publications (see, e.g., [14,20]), we used the fact that the ϕ_j's satisfy the wave equation (5) to eliminate the normal derivative $\phi_{j,x}$ from the ABC, i.e., to replace it by temporal and tangential derivatives. In the DAB method this is not necessary, since the ϕ_j's are defined and discretized in the entire domain Ω_L, and therefore the normal derivative $\phi_{j,x}$ may directly appear in the formulation.

Remark 7. The two previous remarks reveal the main advantage of the DAB method over existing formulations of high-order ABCs using auxiliary variables. In the DAB method one does not have to *prove*, based on given boundary recursion relations, that the auxiliary variables satisfy any wave equation (which may be difficult). Here they are *constructed* to satisfy the wave equation. Also, one does not have to eliminate the normal derivatives of the auxiliary variables (which may be difficult). In addition, special treatment of *corners*, either between an artificial and a physical boundary or between two artificial boundaries, is avoided (although we do not solve any problems with artificial-boundary corners in the present paper). Finally, the possibility to extend the DAB method to more complex media (anisotropic, heterogeneous, etc.) seems promising.

Remark 8. The DAB method involves a layer, just like PML. PML formulations also enjoy some of the properties mentioned above (e.g., no need for special corner treatment). However, the DAB method potentially has some relative advantages over PML. First of all, the DAB method, like any high-order ABC scheme, is associated with a convergence notion in a fixed domain when P increases, whereas with the PML there is no mechanism to improve accuracy within a fixed domain except with the use of singular damping and stretching profiles. Second, PML is notorious for being sensitive to the discretization, whereas high-order ABCs (and therefore hopefully the DAB method as well) are much less so. Finally, it is hoped that the DAB method will be easily generalized to those types of problems for which PML (and also existing ABC schemes) are unstable, e.g., a class of anisotropies in elastodynamics. Careful numerical comparison between the two methods is a major work still to be done.

2.3. Explicit Solution procedure

We propose a solution procedure, based on Eqs. (1)–(12), that is completely explicit. It is important to note, however, that this is not the only explicit solution scheme possible. In fact (and we thank the reviewer for this observation), the entire DAB problem can be handled as a system of hyperbolic equations with coupling conditions via explicit time-stepping methods, for instance by interior-penalty (or other) Discontinuous Galerkin methods [23]. Thus, the procedure outlined below is just one possible approach.

We assume that some space and time discretizations have been applied. We indicate the time-step by the superscript n. The solution procedure consists in the following steps. We assume that all the information at time-step n is known and show how to calculate the information at time-step $n+1$.

- Given at time n: u^n for all grid points in Ω and ϕ_j^n $(j=0,\ldots,P+1)$ for all grid points in Ω_L.
- Compute u^{n+1} at all points except Γ_E, by solving explicitly the discrete version of the ODE (1), using the boundary conditions (2) and the initial conditions (3) and (4).
- Copy the values of u^{n+1} in the layer to the corresponding values of ϕ_0^{n+1}. (Recall (11).)
- Compute ϕ_j^{n+1} $(j=1,\ldots,P+1)$ at all the interior points of the layer Ω_L, namely at all the layer points except Γ_I and Γ_E, by solving explicitly the discrete version of the ODE (5), using the boundary conditions (7) and the initial conditions (6).
- Compute ϕ_{P+1}^{n+1} at Γ_E by solving (12) explicitly.
- Compute ϕ_j^{n+1} at Γ_E for $j=P,P-1,\ldots,0$ recursively, by solving (8) explicitly. (Note the reversed order of the j's. For each j, the right side of (8) is known and the left side is unknown.)
- After the last step ends, there is a new value for ϕ_0^{n+1} at Γ_E. Copy this value to u^{n+1} at Γ_E. (Recall (11).)
- Compute ϕ_{j+1}^{n+1} at Γ_I for $j=0,1,\ldots,P$ recursively, by solving (8) explicitly. (For each j, the left side of (8) is known and the right side is unknown.)
- Move to the next time step: $n \leftarrow n+1$.

This procedure will be applied later to the solution of the 1D Klein–Gordon equation using FD. In principle, it can also be incorporated in a FE scheme, although in this case one has to be careful about the way matrix lumping is done to yield an explicit scheme.

3. Analysis

3.1. Accuracy

First we will prove that *the recursion relations (8) are satisfied in the entire layer* Ω_L and not only on the two boundaries Γ_I ($x = x_I$) and Γ_E ($x = x_E$). To this end, we define, for $j = 0, \ldots, P$,

$$D_{j+1} = \bar{a}_j \dot{\phi}_{j+1} - c\phi_{j+1,x} + \bar{\sigma}_j \phi_{j+1} - a_j \dot{\phi}_j - c\phi_{j,x} - \sigma_j \phi_j \quad \text{in } \Omega_L. \tag{13}$$

From (8) we conclude that $D_{j+1} = 0$ on Γ_I and on Γ_E. Also, since each of the ϕ_j's satisfies the wave equation (1), so does D_{j+1}, i.e., $W D_{j+1} = 0$ in Ω_L. (This is so since the operator appearing in (13) has constant coefficients. In a non-Cartesian coordinate system, this would not be so.) From (7) and (13) we have that D_{j+1} also vanishes on Γ_S and Γ_N. From (6) and (13) we conclude that D_{j+1} satisfies zero initial conditions. There is a unique solution that satisfies these homogeneous differential equation, boundary conditions and initial conditions, and it is the trivial solution $D_{j+1} \equiv 0$. Hence we conclude from (13),

$$a_j \dot{\phi}_j + c\phi_{j,x} + \sigma_j \phi_j = \bar{a}_j \dot{\phi}_{j+1} - c\phi_{j+1,x} + \bar{\sigma}_j \phi_{j+1}, \quad j = 0, \ldots, P, \quad \text{in } \Omega_L. \tag{14}$$

Given that (14) holds in Ω_L, we can now calculate the reflection coefficient R associated with the internal boundary Γ_I, similarly to what has been done with previous ABC schemes. In particular, it is analogous to the analysis of the frequency-domain complete radiation boundary conditions described in [24], where the auxiliary functions are extended from the artificial boundary to the edge of the data region. Performing a Laplace transformation in time (dual variable ω, $\Re\omega > 0$), expanding in a Fourier series in y (dual variable k) and using the fact that the initial data within the layer is 0 we deduce that the transforms of the auxiliary variables ϕ_j have the form

$$\hat{\phi}_j = \mu_j \exp\big[-\gamma(x - x_I)\big] + \rho_j \exp\big[\gamma(x - x_I)\big], \quad j = 0, \ldots, P + 1, \tag{15}$$

where we have introduced

$$\gamma = \frac{1}{c}\big(\omega^2 + c^2 k^2 + s\big)^{1/2}, \quad \Re\gamma > 0. \tag{16}$$

We also note that $\Im\gamma$ and $\Im\omega$ always have the same sign; to see this simply note that if $\Re\omega > 0$ then γ is a continuous function, is real only if $\Im\omega = 0$ and satisfies $\gamma \sim \frac{\omega}{c}$ for $|\omega| \gg 1$. Here $\phi_0 \equiv u$, $\mu_0 + \rho_0 = \hat{u}(x_I)$ and $\rho_0 = R\mu_0$, which is the reflection coefficient. We substitute (15) into (14) and (12), and introduce the layer width $L = x_E - x_I$. Since D_j also satisfies the Klein–Gordon equation, its Fourier–Laplace transform must take the same form as (15), say with coefficients $\tilde{\mu}_j$ and $\tilde{\rho}_j$. However, $D_j = 0$ and the functions $\exp[\pm\gamma x]$ are linearly independent, so we conclude that $\tilde{\mu}_j = \tilde{\rho}_j = 0$. Expressing these in terms of the coefficients for the auxiliary functions we find

$$\mu_j(a_j\omega - c\gamma + \sigma_j) = \mu_{j+1}(\bar{a}_j\omega + c\gamma + \bar{\sigma}_j), \quad j = 0, \ldots, P, \tag{17}$$

$$\rho_j(a_j\omega + c\gamma + \sigma_j) = \rho_{j+1}(\bar{a}_j\omega - c\gamma + \bar{\sigma}_j), \quad j = 0, \ldots, P, \tag{18}$$

$$\mu_{P+1}(a_{P+1}\omega - c\gamma + \sigma_{P+1})\exp[-\gamma L] + \rho_{P+1}(a_{P+1}\omega + c\gamma + \sigma_{P+1})\exp[\gamma L] = 0. \tag{19}$$

Now we use these equations recursively to calculate R:

$$\begin{aligned}
R\mu_0 = \rho_0 &= \rho_1 \frac{\bar{a}_1\omega - c\gamma + \bar{\sigma}_1}{a_1\omega + c\gamma + \sigma_1} \\
&= \rho_2 \left(\frac{\bar{a}_2\omega - c\gamma + \bar{\sigma}_2}{a_2\omega + c\gamma + \sigma_2}\right)\left(\frac{\bar{a}_1\omega - c\gamma + \bar{\sigma}_j}{a_1\omega + c\gamma + \sigma_1}\right) \\
&= \rho_{P+1} \prod_{j=0}^{P} \left(\frac{\bar{a}_j\omega - c\gamma + \bar{\sigma}_j}{a_j\omega + c\gamma + \sigma_j}\right) \\
&= -\mu_{P+1}\exp[-2\gamma L]\left(\frac{a_{P+1}\omega - c\gamma + \sigma_{P+1}}{a_{P+1}\omega + c\gamma + \sigma_{P+1}}\right) \prod_{j=0}^{P} \left(\frac{\bar{a}_j\omega - c\gamma + \bar{\sigma}_j}{a_j\omega + c\gamma + \sigma_j}\right) \\
&= -\mu_P \exp[-2\gamma L]\left(\frac{a_P\omega - c\gamma + \sigma_P}{\bar{a}_P\omega + c\gamma + \bar{\sigma}_P}\right)\left(\frac{a_{P+1}\omega - c\gamma + \sigma_{P+1}}{a_{P+1}\omega + c\gamma + \sigma_{P+1}}\right) \prod_{j=0}^{P} \left(\frac{\bar{a}_j\omega - c\gamma + \bar{\sigma}_j}{a_j\omega + c\gamma + \sigma_j}\right) \\
&= -\mu_0 \exp[-2\gamma L]\left(\frac{a_{P+1}\omega - c\gamma + \sigma_{P+1}}{a_{P+1}\omega + c\gamma + \sigma_{P+1}}\right) \prod_{j=0}^{P} \left(\frac{\bar{a}_j\omega - c\gamma + \bar{\sigma}_j}{a_j\omega + c\gamma + \sigma_j}\right) \cdot \left(\frac{a_j\omega - c\gamma + \sigma_j}{\bar{a}_j\omega + c\gamma + \bar{\sigma}_j}\right).
\end{aligned} \tag{20}$$

Hence after rearranging the terms in the product we have

$$R = -\exp[-2\gamma L]\left(\frac{a_{P+1}\omega - c\gamma + \sigma_{P+1}}{a_{P+1}\omega + c\gamma + \sigma_{P+1}}\right)\prod_{j=0}^{P}\left(\frac{a_j\omega - c\gamma + \sigma_j}{a_j\omega + c\gamma + \sigma_j}\right)\cdot\left(\frac{\bar{a}_j\omega - c\gamma + \bar{\sigma}_j}{\bar{a}_j\omega + c\gamma + \bar{\sigma}_j}\right), \tag{21}$$

which is the desired result.

Lastly, we note that each factor in the product with $a_j, \bar{a}_j > 0$ has absolute value less than 1 unless ω is real, and each factor with $\sigma_j, \bar{\sigma}_j > 0$ always has absolute value less than 1. Thus $|R| < 1$ for $\Re\omega > 0$. Moreover, $|R| < 1$ for $\Re\omega = 0$ except when $\gamma = 0$ where $R = -1$. Introducing a time scale, T, and an error tolerance, ϵ, $P = O(\ln(\frac{1}{\epsilon})\cdot\ln(\frac{cT}{L}))$ parameters can be chosen to guarantee that $|R| < \epsilon$ for $\Re\omega = T^{-1}$, which, combined with the analysis in the subsequent section, guarantees an error $O(\epsilon)$ up to time T. See [15] for details.

Remark 9. We note that the degree of accuracy in (21) as defined by the number of factors in the product is $2P+3$. This is higher by 2 than the degree achieved in our previous publications (see, e.g., [25]), which was $2P+1$. The reason is that in the previous publications the conditions for ϕ_0 and for ϕ_{P+1} were different than those taken here; we used $\dot{\phi}_0 + c\phi_{0,x} = \dot{\phi}_1$ instead of $\dot{\phi}_0 + c\phi_{0,x} = \dot{\phi}_1 - c\phi_{1,x}$ and $\phi_{P+1} = 0$ instead of $\dot{\phi}_{P+1} + c\phi_{P+1,x} = 0$. The reason we did that was that these relations enabled us to eliminate the normal derivatives from all the auxiliary variables. In the DAB method this elimination is unnecessary, and hence we can use the same operators in the equations for ϕ_0 and ϕ_{P+1} as for all the other equations. This has the effect of gaining a degree of 2 in the exponent of the reflection coefficient.

Remark 10. In the analysis above we have argued that since each of the ϕ_j's satisfies the wave equation, so does D_{j+1}. This argument led to (14). We note that this argument does not hold true if both the wave equation and the expression for D_{j+1} are vectorial (i.e., non-scalar). This may happen, for example, in the case of elastodynamics, if one is using vectorial recursive relations on the double boundary for any j except the last one. However, a remedy for this situation is to calculate the reflection coefficients directly.

Remark 11. We note that the formula for the reflection coefficient is closely related to the formulas derived for the reflection coefficients governing the acoustic system in first-order form in [15] for a boundary located a distance L from the source plane.

Remark 12. Since the PDE satisfied by the auxiliary functions ϕ_j (Eq. (5)) is part of the definition of the ϕ_j's, one may wonder whether it would be possible to require the ϕ_j's to satisfy a PDE that is different and simpler than the one satisfied by u. In principle this may be possible. However, an inappropriate PDE for the ϕ_j's may have a negative effect on the overall reflection coefficient. In fact, for the 1D problem presented in the next section, we have checked that if s is nonzero in the original wave equation, then taking $s = 0$ in the equations for the ϕ_j may generate an unbounded reflection coefficient.

3.2. Stability

In order to analyze the stability of the DAB scheme (at the continuous level), we first use the estimates of the reflection coefficient obtained above to bound the solution, u, in the domain, $0 \leqslant x \leqslant x_I$. For simplicity we write

$$u = \tilde{u} + (1+t)e^{-t}u_0 + te^{-t}\dot{u}_0. \tag{22}$$

The function $\tilde{u}$ satisfies the Klein–Gordon equation with forcing

$$H = f - \mathcal{W}\big((1+t)e^{-t}u_0 + te^{-t}\dot{u}_0\big), \tag{23}$$

but with zero initial data. Clearly $\tilde{u} = u$ for $x \geqslant x_I$. After Fourier–Laplace transformation we find:

$$\frac{d^2\hat{\tilde{u}}}{dx^2} = \gamma^2\hat{\tilde{u}} - \frac{1}{c^2}\hat{H}. \tag{24}$$

In addition to the physical boundary condition $\hat{\tilde{u}}(0,k,\omega) = 0$ we can obtain an equivalent boundary condition on Γ_I that involves only $\hat{\tilde{u}}$ and none of the auxiliary variables. Again using the Fourier–Laplace transform and the analysis of the preceding section we find:

$$\hat{u}(\Gamma_I,k,\omega) = \mu_0 + \rho_0 = (1+R)\mu_0, \tag{25}$$

$$\hat{u}_x = -\gamma\mu_0 + \gamma\rho_0 = -\gamma(1-R)\mu_0. \tag{26}$$

Thus the equivalent boundary condition takes the form:

$$\hat{u}_x = -\gamma \frac{(1-R)}{(1+R)} \hat{u}. \tag{27}$$

(Recall that $u = \tilde{u}$ and $u_x = \tilde{u}_x$ at $x = x_I$.)

To estimate $\hat{\tilde{u}}$ we introduce $v_\pm = \hat{\tilde{u}}_x \pm \gamma \hat{\tilde{u}}$. Then it is easy to check:

$$\begin{aligned} &\frac{dv_\pm}{dx} = \pm\gamma v_\pm - \frac{1}{c^2}\hat{H}, \\ &v_+(0) = v_-(0), \\ &v_+(x_I) = -R v_-(x_I). \end{aligned} \tag{28}$$

Integrating this system we find

$$\begin{aligned} v_-(x) &= \exp[-\gamma x] v_-(0) - \frac{1}{c^2}\int_0^x \exp\big[-\gamma\big(x-x'\big)\big] H\big(x'\big)\, dx', \\ v_+(x) &= -R\exp\big[\gamma\big(x-2x_I\big)\big] v_-(0) + \frac{R}{c^2}\int_0^{x_I} \exp\big[\gamma\big(x+x'-2x_I\big)\big] H\big(x'\big)\, dx' \\ &\quad + \frac{1}{c^2}\int_x^{x_I} \exp\big[\gamma\big(x-x'\big)\big] H\big(x'\big)\, dx', \\ v_-(0) &= \frac{1}{c^2\big(1+R\exp[-2\gamma x_I]\big)} \int_0^{x_I} \Big(\exp\big[-\gamma x'\big] + R\exp\big[-\gamma\big(2x_I - x'\big)\big]\Big) H\big(x'\big)\, dx'. \end{aligned} \tag{29}$$

Letting C denote a generic constant dependent only on Γ_I and c and recalling that $|R| < 1$ we see that

$$\int_0^{x_I} \big|v_\pm(x)\big|^2\, dx \leqslant C\big(1+V^2\big) \int_0^{x_I} |H(x)|^2\, dx, \tag{30}$$

where

$$V = \max\left| \frac{\exp\big[-\gamma x'\big] + R\exp\big[-\gamma\big(2x_I - x'\big)\big]}{1 + R\exp[-2\gamma x_I]} \right|. \tag{31}$$

Since $\hat{\tilde{u}}_x = (v_+ + v_-)/2$ we can use Parseval's relation to obtain a bound on $\hat{u}_x$ and thus u:

$$\|u_x\|^2_{L^2(0,T,L^2([0,x_I]\times[0,b]))} \leqslant C\big(1+V^2\big)\big(\|f\|^2_{L^2(0,T,L^2([0,x_I]\times[0,b]))} + \|u_{0,x}\|^2_{L^2([0,x_I]\times[0,b])} + \|\dot{u}_{0,x}\|^2_{L^2([0,x_I]\times[0,b])}\big). \tag{32}$$

From this we obtain a similar bound on u using a Poincaré inequality.

Thus we have stability for u so long as a bound on V can be obtained. We note that such a bound is trivial if we take $\Re\omega \geqslant \frac{1}{T}$ and allow C to depend on T. To get a uniform bound, however, we must allow $\Re\omega = 0$. Noting that $|R| \leqslant R_{\inf} < 1$ for $|\omega| \gg 1$, we must only account for the possibility of $1 + R\exp[-2\gamma x_I] = 0$, which occurs when $\gamma = 0$. Using l'Hôpital's rule we see that the formula defining V takes on the limiting value

$$\left| \frac{\frac{dR}{d\gamma}(0) + 2\big(x_I - x'\big)}{\frac{dR}{d\gamma}(0) + 2x_I} \right|. \tag{33}$$

Noting that $R = -1$ and that $|R| < 1$ if $\Re\gamma > 0$ we see that $\Re\frac{dR}{d\gamma}(0) \geqslant 0$. Thus V is bounded and stability for u is established. Note that for typical choices of the parameters we could show that the bound can be made uniform in P.

We now move on to estimate the auxiliary functions, ϕ_j. First note that we can now express their transforms directly in terms of $v_\pm$. (The formulas listed above for $v_\pm$ also hold when $x > x_I$.) In particular since $v_+(x_I) = -R v_-(x_I)$

$$\begin{aligned} \mu_j &= -\frac{v_-(x_I)}{2\gamma} \prod_{k=0}^{j-1} \frac{a_k\omega - c\gamma + \sigma_k}{\bar{a}_k\omega + c\gamma + \bar{\sigma}_k}, \\ \rho_j &= \frac{v_+(x_I)}{2\gamma} \prod_{k=0}^{j-1} \frac{a_k\omega + c\gamma + \sigma_k}{\bar{a}_k\omega - c\gamma + \bar{\sigma}_k} \end{aligned}$$

$$= \frac{v_-(x_I)}{2\gamma}\exp[-2\gamma L]\left(\frac{a_{P+1}\omega - c\gamma + \sigma_{P+1}}{a_{P+1}\omega + c\gamma + \sigma_{P+1}}\right)$$
$$\times \prod_{k=j}^{P}\left[\left(\frac{a_k\omega - c\gamma + \sigma_k}{a_k\omega + c\gamma + \sigma_k}\right)\cdot\left(\frac{\bar{a}_k\omega - c\gamma + \bar{\sigma}_k}{\bar{a}_k\omega + c\gamma + \bar{\sigma}_k}\right)\right]\cdot\prod_{k=0}^{j-1}\frac{a_k\omega - c\gamma + \sigma_k}{\bar{a}_k\omega + c\gamma + \bar{\sigma}_k}. \tag{34}$$

Introducing

$$P_j = \prod_{k=0}^{j-1}\frac{a_k\omega - c\gamma + \sigma_k}{\bar{a}_k\omega + c\gamma + \bar{\sigma}_k}, \tag{35}$$

$$S_j = \prod_{k=j}^{P}\left(\frac{a_k\omega - c\gamma + \sigma_k}{a_k\omega + c\gamma + \sigma_k}\right)\cdot\left(\frac{\bar{a}_k\omega - c\gamma + \bar{\sigma}_k}{\bar{a}_k\omega + c\gamma + \bar{\sigma}_k}\right) \tag{36}$$

and recalling the formula for $v_-(x)$ we find

$$\hat{\phi}_j = \left(\exp[-\gamma x_I]v_-(0) + \int_0^{x_I}\exp\left[-\gamma\left(x_I - x'\right)\right]\hat{H}\left(x'\right)dx'\right)\cdot P_j$$
$$\times\left(\frac{S_j\exp\left[\gamma\left(x + x_I - 2x_E\right)\right] - \exp\left[-\gamma\left(x - x_I\right)\right]}{2\gamma}\right). \tag{37}$$

Thus we obtain bounds for ϕ_j analogous to (32) with an additional multiplicative constant

$$K_j = \max\left|P_j\frac{S_j\exp\left[\gamma\left(x + x_I - 2x_E\right)\right] - \exp\left[-\gamma\left(x - x_I\right)\right]}{2\gamma}\right|. \tag{38}$$

To bound K_j we first note that $|S_j| \leqslant 1$ while bounds on $|P_j|$ will depend on the particular choice of parameters. However, it is clear that $|P_j|$ is uniformly bounded for $\Re\omega > 0$. In fact, if we make the particular choice $a_j = \bar{a}_j$ and $\sigma_j = \bar{\sigma}_j$ it is less than 1. Thus we need only be concerned with the case of $\gamma = 0$. Noting that $S_j(0) = 1$ again we may apply l'Hôpital's rule to obtain the limiting value

$$\left|P_j(0)\frac{\frac{dS_j}{d\gamma}(0) + 2(x - x_E)}{2}\right|, \tag{39}$$

which is clearly bounded. Thus stability for the auxiliary functions is established.

Remark 13. The Dirichlet condition at $x = 0$ can be replaced by a Neumann condition. The only change in the analysis would be that $v_+(0) = -v_-(0)$, leading to a modified formula for V. The modified formula can also be analyzed, and in fact l'Hôpital's rule is not needed.

This analysis pertains to stability at the continuous level, not to the stability of the fully discrete scheme. We have checked the stability of the DAB method numerically as well, by running the 1D FD scheme and the 2D FE scheme discussed later for very long times. In all cases, and with all orders P, the solution remained stable. More details will be provided later.

4. Application of DAB in 1D: A finite difference scheme

Now we consider the (homogeneous) Klein–Gordon equation (1) in 1D for $u(x,t)$:

$$\mathcal{W}u \equiv \ddot{u} - c^2u'' + su = 0, \quad x > 0,\ t > 0. \tag{40}$$

Here a prime indicates differentiation with respect to x. This can be thought of as a model for a string (or cable) on an elastic foundation. For simplicity we assume that c and s are constant. We assume the left boundary condition

$$u(0,t) = 0. \tag{41}$$

We prescribe initial conditions

$$u(x, t = 0) = u_0(x), \qquad \dot{u}(x, t = 0) = \dot{u}_0(x). \tag{42}$$

We truncate the domain by introducing the artificial boundary Γ_E at $x = x_E$ and the interface Γ_I at $x = x_I$; see Fig. 1 (but imagine that the domain is 1D). As in the 2D case, we introduce auxiliary variables $\phi_j(x,t)$ and require that they satisfy the wave equation (40) in the layer $\Omega_L = \{x \mid x_I < x < x_E\}$:

$$\mathcal{W}\phi_j \equiv \ddot{\phi}_j - c^2\phi_j'' + s\phi_j = 0, \quad j = 0,\ldots,P+1, \quad \text{in } \Omega_L. \tag{43}$$

Eqs. (6), (8)–(12) are also carried over to the 1D case:

$$\phi_j(x,t=0) = 0, \qquad \dot{\phi}_j(x,t=0) = 0, \quad j = 0,\ldots,P+1, \quad \text{in } \Omega_L, \tag{44}$$

$$\dot{\phi}_j + c\phi_j' = \dot{\phi}_{j+1} - c\phi_{j+1}', \quad j = 0,\ldots,P, \quad \text{on } \Gamma_I,\ \Gamma_E, \tag{45}$$

$$\phi_0 = u, \qquad \phi_0' = u' \quad \text{on } \Gamma_I, \tag{46}$$

$$\phi_0 \equiv u \quad \text{in } \Omega_L, \tag{47}$$

$$\dot{\phi}_{P+1} + c\phi_{P+1}' = 0 \quad \text{on } \Gamma_E. \tag{48}$$

Using FD in space and time, we discretize the spatial coordinate with the space increment Δx, and we discretize time with the time-step size Δt. Grid points are indicated by the subscript m and time-steps are indicated by the superscript n. The points at Γ_E and Γ_I are indicated simply by $m = E$ and $m = I$, respectively. The solution procedure outlined in Section 2.3 is implemented by using second-order FD approximation for x and t derivatives. For the efficient writing of the FD formulas of the boundary conditions, we define a few discrete operators:

$$\text{Forward average in space:} \qquad A_x^+ v_m = (v_m + v_{m+1})/2, \tag{49}$$

$$\text{Forward average in time:} \qquad A_t^+ v^n = \left(v^n + v^{n+1}\right)/2, \tag{50}$$

$$\text{Forward difference in space:} \qquad D_x^+ v_m = (v_{m+1} - v_m)/\Delta x, \tag{51}$$

$$\text{Forward difference in time:} \qquad D_t^+ v^n = \left(v^{n+1} - v^n\right)/\Delta t. \tag{52}$$

Note that all these approximations are *centered* around a point "$m+1/2$" residing between grid points m and $m+1$ and around time "$n+1/2$" between times n and $n+1$. With a physical boundary one cannot apply the boundary condition at a grid point "$m+1/2$" which does not lie on the boundary; however, with an artificial boundary this can be done, since the boundary condition is satisfied at the point "$m+1/2$" as well. We also define the constant

$$\omega = \frac{1}{\Delta t} + \frac{c}{\Delta x} \tag{53}$$

for future reference.

We apply the explicit solution procedure outlined in Section 2.3. Here are the details for the FD implementation of this procedure.

- Given at time n: u_m^n for all grid points m in Ω, and $\phi_{j,m}^n$ $(j = 0,\ldots,P+1)$ for all grid points m in Ω_L.
- Compute u^{n+1} at all points except Γ_E, by solving explicitly the discrete version of the ODE (40). This is done by using the standard 2nd-order accurate central difference formula

$$u_m^{n+1} = 2u_m^n - u_m^{n-1} + (c\Delta t/\Delta x)^2\left(u_{m+1}^n - 2u_m^n + u_{m-1}^n\right) - s(\Delta t)^2 u_m^n. \tag{54}$$

  This is not a self-starting formula. To start it and to maintain 2nd-order accuracy, we consider (54) for time $n = 0$ and obtain an equation that involves u_m^1, u_m^0 and u_m^{-1}. We also write the given initial velocity v_m^0 as a central difference of values at times 1 and -1, i.e., $v_m^0 = (u_m^1 - u_m^{-1})/(2\Delta t)$. This gives us two equations with two unknowns, u_m^1 and u_m^{-1}. Solving them and simplifying finally yields the starting formula

$$u_m^1 = u_m^0 + \Delta t v_m^0 + (1/2)(c\Delta t/\Delta x)^2\left(u_{m+1}^0 - 2u_m^0 + u_{m-1}^0\right) - (s/2)(\Delta t)^2 u_m^0. \tag{55}$$

- Copy the values of u^{n+1} in the layer to the corresponding values of ϕ_0^{n+1}. (Recall (11).)
- Compute ϕ_j^{n+1} $(j = 1,\ldots,P+1)$ at all the interior points of the layer Ω_L, namely at all the layer points except Γ_I and Γ_E, by solving explicitly the discrete version of the ODE (43). Again use the standard 2nd-order accurate central difference formula, i.e., (54) where u is replaced by ϕ_j everywhere.
- Compute ϕ_{P+1}^{n+1} at point Γ_E by solving the discrete version of (48) explicitly. We use 2nd-order central approximations for the derivatives, and thus (48) gives

$$A_x^+ D_t^+ \phi_{P+1,E-1}^n + cA_t^+ D_x^+ \phi_{P+1,E-1}^n = 0. \tag{56}$$

  The only unknown in this equation is $\phi_{P+1,E}^{n+1}$, which is involved (unseen) in (56) as a result of the action of the operators A^+D^+. Solving for $\phi_{P+1,E}^{n+1}$ we get the explicit formula

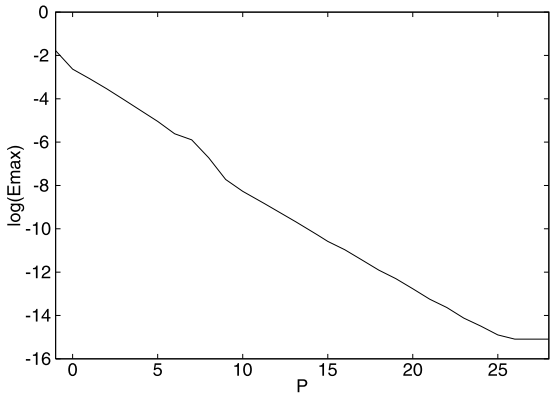

Fig. 3. Logarithm of the maximum error as a function of the boundary conditions order P, for $s = 1$ and $N_L = 3$.

$$\omega\phi^{n+1}_{P+1,E} = -D^+_t\phi^n_{P+1,E-1} - cD^+_x\phi^n_{P+1,E-1} + \frac{1}{\Delta t}\phi^n_{P+1,E} + \frac{c}{\Delta x}\phi^{n+1}_{P+1,E-1}. \tag{57}$$

- Compute ϕ^{n+1}_j at point Γ_E for $j = P, P-1, \ldots, 0$ recursively, by solving the discrete version of (8) explicitly. The 2nd-order central approximation of (8) is

$$A^+_x D^+_t\phi^n_{j,E-1} + cA^+_t D^+_x\phi^n_{j,E-1} = A^+_x D^+_t\phi^n_{j+1,E-1} - cA^+_t D^+_x\phi^n_{j+1,E-1}. \tag{58}$$

  This yields the explicit formula, for $j = P, P-1, \ldots, 0$,

$$\omega\phi^{n+1}_{j,E} = 2A^+_x D^+_t\phi^n_{j+1,E-1} - 2cA^+_t D^+_x\phi^n_{j+1,E-1} - D^+_t\phi^n_{j,E-1} - cD^+_x\phi^n_{j,E-1} + \frac{1}{\Delta t}\phi^n_{j,E} + \frac{c}{\Delta x}\phi^{n+1}_{j,E-1}. \tag{59}$$

- Copy the new value of $\phi^{n+1}_{0,E}$ to u^{n+1} at point Γ_E. (Recall (11).)
- Compute ϕ^{n+1}_{j+1} at point Γ_I for $j = 0, 1, \ldots, P$ recursively, by solving the discrete version of (8) explicitly. The 2nd-order central approximation of (8) is

$$A^+_x D^+_t\phi^n_{j,I} + cA^+_t D^+_x\phi^n_{j,I} = A^+_x D^+_t\phi^n_{j+1,I} - cA^+_t D^+_x\phi^n_{j+1,I}. \tag{60}$$

  This yields the explicit formula, for $j = 0, 1, \ldots, P$,

$$\omega\phi^{n+1}_{j+1,I} = 2A^+_x D^+_t\phi^n_{j,I} + 2cA^+_t D^+_x\phi^n_{j,I} - D^+_t\phi^n_{j+1,I+1} + cD^+_x\phi^n_{j+1,I} + \frac{1}{\Delta t}\phi^n_{j+1,I} + \frac{c}{\Delta x}\phi^{n+1}_{j+1,I+1}. \tag{61}$$

- Move to the next time step: $n \leftarrow n+1$.

5. Application of DAB in 1D: Numerical experiments

We consider the 1D problem and the FD procedure described in the previous section. We set $c = 1$, while s will be specified later. We take grid spacing of $\Delta x = 0.1$. The location of Γ_I is fixed at $x_I = 9.8$, and so Ω_I contains 99 grid points. The location of the exterior boundary Γ_E is determined by the thickness of the layer, which is left as a free parameter. We take $\Delta t = 0.09$, and 400 time steps. This means that the final time is $T = 36$. As initial conditions, we take a "hat function", starting as u=0 at $x = 0$, varying linearly to $u = 1$ at $x = 0.4$, and varying linearly back to $u = 0$ at $x = 0.8$. The initial velocity is taken as zero. The parameters s (dispersion parameter), N_L (number of grid points in the layer, including the nodes on Γ_I and Γ_E) and P (order of the boundary conditions) will be varied. Lastly we take the simplest possible choice for the boundary condition parameters: $a_j = \bar{a}_j = 1$, $\sigma_j = \bar{\sigma}_j = 0$. These are sufficient to illustrate the stability and convergence of the method. For longer time computations optimized parameters as derived in [15] should be used.

To calculate the error, we also solve a reference problem with a much larger domain. In the reference problem we take $x_E = 30$, with 301 grid points (and thus again $\Delta x = 0.1$). Other parameters are the same as in the problem described above.

We first set $s = 1$, and take $N_L = 3$, namely a single grid point separates Γ_I and Γ_E. Fig. 3 shows the maximum error vs. the order P. The case $P = -1$ corresponds to the classical Sommerfeld condition $\dot{u} + cu' = 0$ applied on Γ_E. It is clear that the error reduces exponentially fast with P. It is observed that very low error levels are obtained, and the error is seen to "saturate" only at the level of machine accuracy. This has not been expected, since the discretization scheme used is only 2nd-order accurate in space and time within the DAB layer. Obviously there is a super-convergence phenomenon occurring in this simple 1D configuration.

Now we set $s = 1$ and $P = 4$ and we vary the layer thickness as defined by N_L. Fig. 4 shows the maximum error vs. N_L, for $N_L = 3, \ldots, 10$. As seen in the figure, the error decreases quite slowly when the layer is made thicker. Thus, it is much

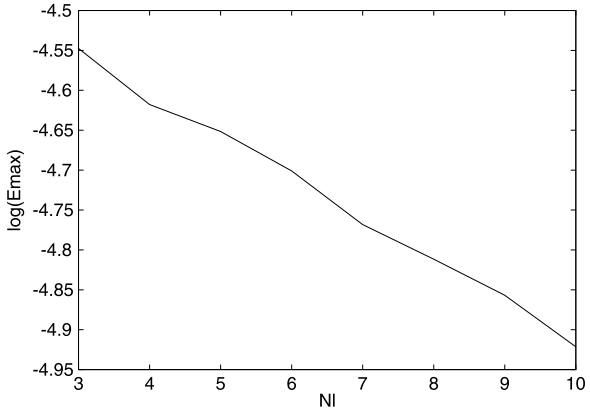

Fig. 4. Logarithm of the maximum error as a function of the layer thickness N_L, for $s = 1$ and $P = 4$.

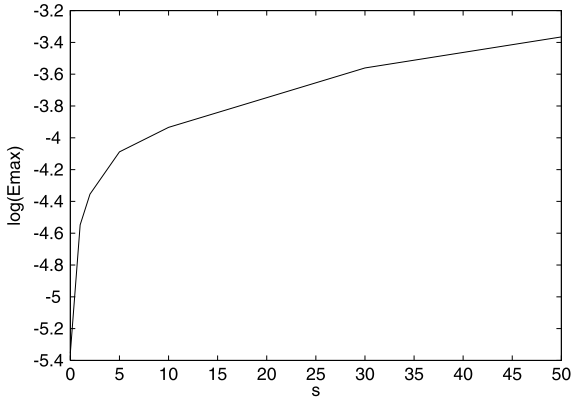

Fig. 5. Logarithm of the maximum error as a function of the dispersion parameter s, for $P = 4$ and $N_L = 3$.

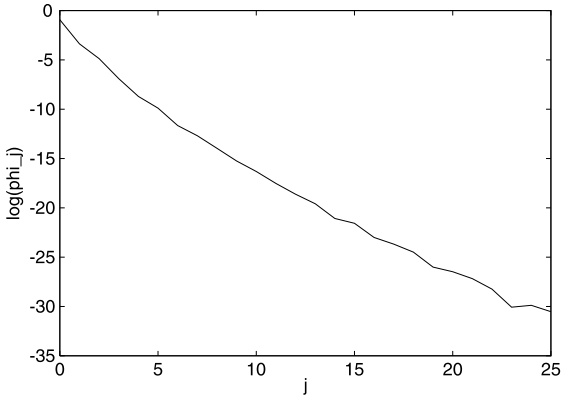

Fig. 6. Logarithm of the auxiliary function ϕ_j at Γ_I as a function of j, for $P = 24$, $s = 1$ and $N_L = 3$.

more effective to fix the layer's thickness N_L and increase the ABC order P than to fix P and increase N_L. This conclusion is consistent with observations made with previous high-order ABC schemes. It should be noted that the minimum layer thickness should be $N_L = 3$ (two cells), since in the case $N_L = 2$ (one cell) there are no interior points in the layer, and therefore, with the FD approximation that we are using, no PDE is solved for the ϕ_j's at all, which amounts to an ill-posed problem.

We set $N_L = 3$ and $P = 4$ and we vary s. As seen in Fig. 5, the maximum error increases with s. This is expected, since with an increased s the dispersion effect increases, which causes the solution to be richer with waves of different wave-lengths and frequencies.

Since the method involves the use of the auxiliary variables ϕ_j in the layer, it is interesting to investigate the behavior of these functions. For $s = 1$, $N_L = 3$ and $P = 24$, Fig. 6 shows the decay of ϕ_j at Γ_I for increasing j. As the figure shows,

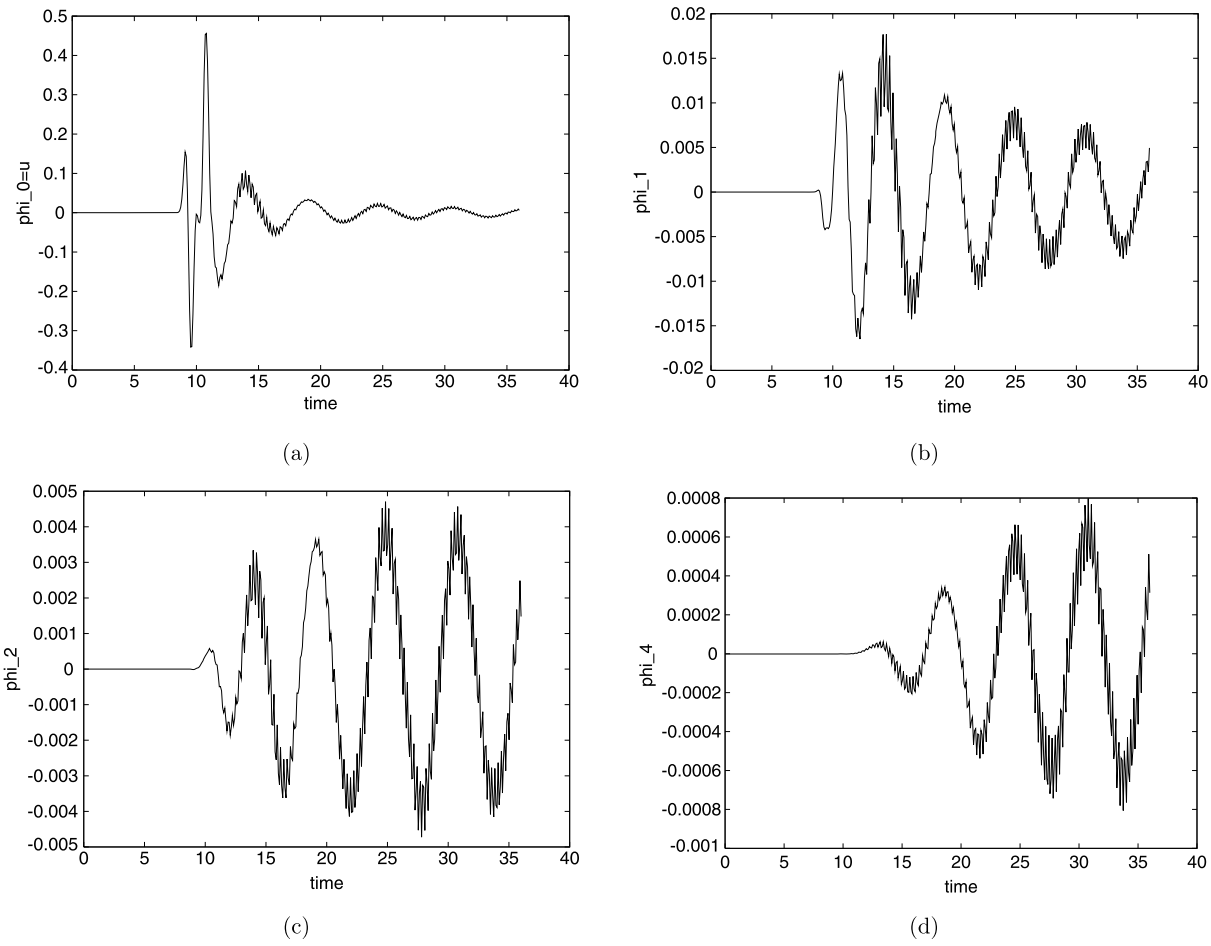

Fig. 7. The auxiliary variable ϕ_j at Γ_I as a function of time, for $P = 3$, $s = 1$ and $N_L = 3$. (a) $\phi_0 = u$; (b) ϕ_1; (c) ϕ_2; (d) ϕ_4.

the relative importance of an auxiliary variable decreases exponentially with its order. This is a favorable situation, that can be exploited by an adaptive scheme, to control the parameter P used during the solution process; see [25]. We note that this property is not possessed by some other ABC schemes, like the Givoli–Neta and the Collino schemes; see [26] for a discussion on this issue.

It is also interesting to investigate the behavior of the auxiliary variables in time. The primary variable u must generally decay in time, since the scheme is stable and the physical energy gradually goes out of the computational domain through the artificial boundary Γ_E (although due to the dispersion there is always some physical energy left in the domain). The behavior of the auxiliary variables is less predictable. For $P = 3$, $s = 1$ and $N_L = 3$, Fig. 7 shows $\phi_0 = u$, ϕ_1, ϕ_2 and ϕ_4 versus time at Γ_I. Indeed the wave amplitude ϕ_0 at Γ_I generally decays in time. The auxiliary function ϕ_1 also seems to decay, but ϕ_2 and ϕ_4 do not decay during the simulation time. (Of course, the magnitude of these non-decaying functions is very small.)

6. Application of DAB in 2D: A finite element scheme

6.1. Strong form

We present a 2D Galerkin Finite Element (FE) formulation for the DAB scheme. The problem considered here is a slight variation of the 2D wave guide problem stated in Section 2.1. See Fig. 1.

All the unknown variables satisfy the Klein–Gordon equation, i.e.,

$$\ddot{u} - c\nabla^2 u + su = 0 \quad \text{in } \Omega, \tag{62}$$

$$\ddot{\phi}_j - c\nabla^2 \phi_j + s\phi_j = 0, \quad j = 1, \ldots, P \quad \text{in } \Omega_L, \tag{63}$$

and are subject to the following initial conditions:

$$u(x,y,0)=u_0 \quad \text{in } \Omega, \tag{64}$$

$$\dot{u}(x,y,0)=v_0 \quad \text{in } \Omega, \tag{65}$$

$$\phi_j(x,y,0)=0,\ j=1,\dots,P \quad \text{in } \Omega_L, \tag{66}$$

$$\dot{\phi}_j(x,y,0)=0, \quad j=1,\dots,P \quad \text{in } \Omega_L, \tag{67}$$

where u_0 and v_0 are given functions.

The boundary conditions are as follows. On Γ_W we apply a zero Neumann condition, i.e.,

$$u_{,x}=0 \quad \text{on } \Gamma_W. \tag{68}$$

On Γ_I and Γ_E the auxiliary variables ϕ_j satisfy the recursive relations constituting the ABC, i.e.,

$$\dot{\phi}_j + c\phi_{j,x} = \dot{\phi}_{j+1} - c\phi_{j+1,x}, \quad j=0,\dots,P-1 \quad \text{on } \Gamma_E \text{ and } \Gamma_I, \tag{69}$$

$$\dot{\phi}_P + c\phi_{P,x} = 0 \quad \text{on } \Gamma_E. \tag{70}$$

Note that for reasons of programming convenience, the indexing of the ϕ_j is slightly different than that used in Section 4, namely here $j=0,\dots,P$ and the last auxiliary variable is ϕ_P. Also, (69) and (70) are such that in the case $P=0$ only (70) is relevant, and corresponds to the zero-order Sommerfeld-like ABC. On the boundaries Γ_N and Γ_S we consider two possible cases:

Case 1. Periodic boundary conditions:

$$u(x,0,t)=u(x,b,t), \qquad u_{,y}(x,0,t)=u_{,y}(x,b,t). \tag{71}$$

Case 2. Neumann boundary conditions:

$$u_{,y}=0 \quad \text{on } \Gamma_N, \Gamma_S. \tag{72}$$

We also need to impose boundary conditions for the ϕ_j on Γ_S and Γ_N (or more precisely, on the portions of Γ_S and Γ_N which bound the layer Ω_L). For Case 1 this is straightforward, since the entire problem is periodic, namely,

$$\phi_j(x,0,t)=\phi_j(x,b,t), \qquad \phi_{j,y}(x,0,t)=\phi_{j,y}(x,b,t), \quad j=1,\dots,P. \tag{73}$$

For Case 2, we can deduce by induction, based on (72) and the recursive relations (69), that all the ϕ_j satisfy

$$\phi_{j,y}=0 \quad \text{on } \Gamma_N, \Gamma_S. \tag{74}$$

6.2. Weak form

We now outline the weak form of the problem. In Case 1 we define the function spaces

$$\mathcal{S}=\left\{u \,\middle|\, u\in H^1(\Omega),\ u(\Gamma_S)=u(\Gamma_N)\right\}, \qquad \mathcal{S}_L=\left\{\phi_j \,\middle|\, \phi_j\in H^1(\Omega_L),\ \phi_j(\Gamma_S)=\phi_j(\Gamma_N)\right\}, \tag{75}$$

whereas in Case 2 these function spaces are defined simply as $\mathcal{S}=H^1(\Omega)$ and $\mathcal{S}_L=H^1(\Omega_L)$. Then, the weak form of the problem is:

Find $u\in\mathcal{S}$ and $\phi_j\in\mathcal{S}_L$, $j=1,\dots,P$ such that for all $w_0\in\mathcal{S}$ and $w_j\in\mathcal{S}_L$, $j=1,\dots,P$ there holds:

$$c^2\int_\Omega \nabla w_0\cdot\nabla u\,d\Omega + \int_\Omega w_0\ddot{u}\,d\Omega + s\int_\Omega w_0 u\,d\Omega + c\int_{\Gamma_E} w_0\dot{u}\,d\Gamma - c\int_{\Gamma_E} w_0\dot{\phi}_1\,d\Gamma + c^2\int_{\Gamma_E} w_0\phi_{1,x}\,d\Gamma = 0; \tag{76}$$

for $j=1,\dots,P-1$:

$$\begin{aligned} &c^2\int_{\Omega_L}\nabla w_j\cdot\nabla\phi_j\,d\Omega + \int_{\Omega_L} w_j\ddot{\phi}_j\,d\Omega + s\int_{\Omega_L} w_j\phi_j\,d\Omega \\ &\quad + c\int_{\Gamma_E} w_j\dot{\phi}_j\,d\Gamma - c\int_{\Gamma_E} w_j\dot{\phi}_{j+1}\,d\Gamma + c^2\int_{\Gamma_E} w_j\phi_{j+1,x}\,d\Gamma \\ &\quad + c\int_{\Gamma_I} w_j\dot{\phi}_j\,d\Gamma - c\int_{\Gamma_I} w_j\dot{\phi}_{j-1}\,d\Gamma + c^2\int_{\Gamma_I} w_j\phi_{j-1,x}\,d\Gamma = 0; \end{aligned} \tag{77}$$

and for $j = P$:

$$c^2 \int_{\Omega_L} \nabla w_P \cdot \nabla \phi_P \, d\Omega + \int_{\Omega_L} w_P \ddot{\phi}_P \, d\Omega + s \int_{\Omega_L} w_P \phi_P \, d\Omega + c \int_{\Gamma_I} w_P \dot{\phi}_P \, d\Gamma - c \int_{\Gamma_I} w_P \dot{\phi}_{P-1} \, d\Gamma + c^2 \int_{\Gamma_I} w_P \phi_{P-1,x} \, d\Gamma = 0. \tag{78}$$

6.3. Space discretization

Eqs. (76)–(78) are discretized in space using the standard Galerkin FE method. At the global level, the variable u in Ω is replaced by the finite-dimensional approximation

$$u^h(\boldsymbol{x}, t) = \sum_{A \in \eta} d_A^h(t) N_A(\boldsymbol{x}), \quad \boldsymbol{x} \in \Omega, \tag{79}$$

and similarly, the variables ϕ_j in Ω_L $(j = 1, \ldots, P)$, are replaced by

$$\phi_j^h(\boldsymbol{x}, t) = \sum_{A \in \eta^L} \phi_{jA}^h(t) N_A(\boldsymbol{x}), \quad \boldsymbol{x} \in \Omega_L, \quad j = 1, \ldots, P. \tag{80}$$

Here the index A stands for a node number, η and η^L are the sets of all nodes in Ω and in Ω_L, respectively (where in Case 1 the nodes on Γ_N have identical numbers to the corresponding nodes on Γ_S, thus taking account of the periodicity), N_A is the FE shape functions associated with node A, and the d_A^h and ϕ_{jA}^h are unknown functions of time. Note that, as seen by (79) and (80), we choose to use the same shape functions for u^h as for all the ϕ_j^h, although in principle they may be chosen to be different. (In the numerical examples of the next section, the N_A are piecewise-bilinear functions.) At the element level the analogous expansions are

$$u^e(\boldsymbol{x}, t) = \sum_{a=1}^{N_{en}} d_a^e(t) N_a(\boldsymbol{x}), \quad \boldsymbol{x} \in \Omega^e, \tag{81}$$

$$\phi_j^e(\boldsymbol{x}, t) = \sum_{a=1}^{N_{en}} \phi_{ja}^e(t) N_a(\boldsymbol{x}), \quad \boldsymbol{x} \in \Omega_L^e, \quad j = 1, \ldots, P. \tag{82}$$

Here e stands for the element number, N_{en} is the number of element nodes, a is the local (element) node number, Ω^e is the domain of element e, $\Omega_L^e = \Omega_L \cap \Omega^e$, N_a is the element-level shape function associated with the node a, and d_a^e and ϕ_{ja}^e are the values of u^e and ϕ_j^e at node a of element e. Similar expansions are used for the test functions w_0 (in Ω) and w_j, $j = 1, \ldots, P$ (in Ω_L).

Using the approximations (79)–(80) in the weak equations (76)–(78) gives the following system of linear ODEs in time:

$$\boldsymbol{M}_0 \ddot{\boldsymbol{d}} + \boldsymbol{C}_0 \dot{\boldsymbol{d}} + \boldsymbol{K}_0 \boldsymbol{d} + \boldsymbol{G}_0 \dot{\boldsymbol{\phi}}_1 + \boldsymbol{H}_0 \boldsymbol{\phi}_1 = 0; \tag{83}$$

for $j = 1, \ldots, P-1$:

$$\boldsymbol{M}_j \ddot{\boldsymbol{\phi}}_j + \boldsymbol{C}_j \dot{\boldsymbol{\phi}}_j + \boldsymbol{K}_j \boldsymbol{\phi}_j + \boldsymbol{A}_j \dot{\boldsymbol{\phi}}_{j-1} + \boldsymbol{B}_j \boldsymbol{\phi}_{j-1} + \boldsymbol{G}_j \dot{\boldsymbol{\phi}}_{j+1} + \boldsymbol{H}_j \boldsymbol{\phi}_{j+1} = 0; \tag{84}$$

and

$$\boldsymbol{M}_P \ddot{\boldsymbol{\phi}}_P + \boldsymbol{C}_P \dot{\boldsymbol{\phi}}_P + \boldsymbol{K}_P \boldsymbol{\phi}_P + \boldsymbol{A}_P \dot{\boldsymbol{\phi}}_{P-1} + \boldsymbol{B}_P \boldsymbol{\phi}_{P-1} = 0. \tag{85}$$

Here $\boldsymbol{d}$ and $\boldsymbol{\phi}_j$ are vectors whose entries are the unknown nodal values of u^h in Ω and of ϕ_j^h in Ω^L, respectively. Initial conditions, directly obtained from (64)–(67), accompany these equations:

$$\boldsymbol{d}(0) = \boldsymbol{d}_0, \qquad \boldsymbol{\phi}_j(0) = \boldsymbol{0}, \qquad \dot{\boldsymbol{d}}(0) = \boldsymbol{v}_0, \qquad \dot{\boldsymbol{\phi}}_j(0) = \boldsymbol{0}, \quad j = 1, \ldots, P. \tag{86}$$

The element-level expressions for the arrays appearing in (83)–(85) may be extracted from (76)–(78). These expressions are written down in Appendix A. The global arrays appearing in (83)–(85) are obtained from the element arrays (A.7)–(A.23) (see Appendix A) via the usual FE assembly process.

Remark 14. To take full advantage of the high order of the DAB conditions, one should also use a high-order discretization scheme. If one uses low-order FEs, the accuracy of the DAB would be limited by the discretization error. Ideally, one should increase the order of the discretization along with the order of the DAB conditions in order to increase the total accuracy as much as desired and approach the exact solution of the original unbounded domain problem. In [27], a spectral FE scheme

of arbitrary order was used along with a high-order ABC for problems in acoustics, leading to an extremely high accuracy, which reached machine precision. Here, however, we shall use standard low-order FEs in conjunction with the DAB. Even in this case, the advantage of using DAB will be seen clearly, with DAB orders of up to $P = 4$ or $P = 5$. Beyond this order, no increase in accuracy can typically be obtained without increasing the order of the discretization as well.

6.4. Time discretization

The entire semi-discrete system of ODEs (83)–(85) is solved as a whole, using a time-stepping scheme.

In the interior domain Ω_I, any time-stepping scheme may be used in principle. In the layer Ω_L, we have found out experimentally that a *dissipative* time-stepping scheme must be used; otherwise the whole discrete formulation loses its stability. This behavior is reminiscent of other cases of discrete problems involving non-standard ABCs where an energy-preserving time-stepping scheme leads to an instability whereas a dissipative scheme yields a stable scheme. This situation is at the heart of the Goldberg–Tadmor theory for the discrete-level stability of initial boundary value problems [28]. An example of a similar situation can be found, e.g., in [29].

In the present work we use the Newmark family of time-stepping schemes, which involves the two parameters β and γ [30]. In the layer Ω_L (including all the degrees of freedom on the two boundaries Γ_I and Γ_E) we use $\gamma = 0.7$ and $\beta = 0.36$ which produces a dissipative unconditionally-stable implicit first-order scheme. In the interior domain Ω_I the standard parameters $\gamma = 0.5$ and $\beta = 0.25$ are used, which gives a second-order energy-preserving scheme (average acceleration, or trapezoidal). The first-order accuracy of the scheme used in the layer is directly not of much concern, since the layer may be regarded as a purely numerical construction, while in the region of interest the scheme has second-order accuracy in time. However, it is possible that the reduced accuracy could affect the reflection coefficient.

7. Application of DAB in 2D: Numerical experiments

7.1. Numerical setup

In this section, we present numerical results obtained from the FE scheme described in the previous section. For each solved example a reference solution $u^{h,ref}$ is generated which enables the calculation of the error associated with the ABC. This is done by solving the problem in a much longer waveguide in which the waves reflected from the far boundary do not reach back to $x = x_E$ during the simulation. In the reference problem waves generated in Ω do not encounter any boundary or ABC at $x = x_I$ or $x = x_E$, and hence the difference between the solution u^h computed in Ω and the reference solution $u^{h,ref}$ there is approximately the ABC error.

We define two error measures: the global relative space–time error

$$E = \frac{\|u^h - u^{h,ref}\|_{\Omega_I \times [0,T]}}{\|u^{h,ref}\|_{\Omega_I \times [0,T]}}, \tag{87}$$

and the time-dependent error

$$e(t) = \frac{\|u^h(t) - u^{h,ref}(t)\|_{\Omega_I}}{\sqrt{A(\Omega_I)}}. \tag{88}$$

Here $\|\cdot\|_{\mathcal{M}}$ is the l_2 norm calculated on the manifold $\mathcal{M}$, T is the simulation time, and $A(\Omega)$ is the area of Ω. Note that the errors are measured outside of Ω_L.

In the following numerical experiments we set the length of the computational domain (without the absorbing layer) to $x_I = 9.7$, the length of the reference domain to $x_{far} = 20$ and the waveguide width to $b = 3$. The wave velocity is taken as $\sqrt{3}$, and we set $s = 0$ (a non-dispersive medium). We use a uniform FE mesh of square bilinear elements with a mesh parameter of $h = 0.05$ both in Ω and in Ω_L. While the location of the interface Γ_I is fixed, the location of the east boundary Γ_E (or, equivalently, the thickness of the layer) will vary in the experiments described below. The reference mesh has the same density and comprises 24,000 elements. For the time-stepping scheme (see details in the previous section) we use a time step of $\Delta t = 0.005$. the simulation time is taken to be $T = 10$.

As mentioned previously, in addition to the accuracy tests described below, the *stability* of the scheme was tested numerically. To this end, the example models in the following subsections were run for 500,000 time steps, namely up to $T = 2500$, and for various values of ABC order P (up to $P = 10$). In all cases the scheme was observed to be stable.

7.2. Pulse propagation

The first case considered is the propagation of a two-dimensional pulse. On the north and south boundaries we use periodic boundary conditions (see (71)). As initial conditions we prescribe

$$u_0(x, y) = \begin{cases} [(x - 8.5)^2 - 1]^2 \sin(\frac{2\pi y}{b}) & \text{for } 7.5 \leqslant x \leqslant 9.5, \\ 0 & \text{otherwise}, \end{cases} \tag{89}$$

$$v_0(x, y) = 0. \tag{90}$$

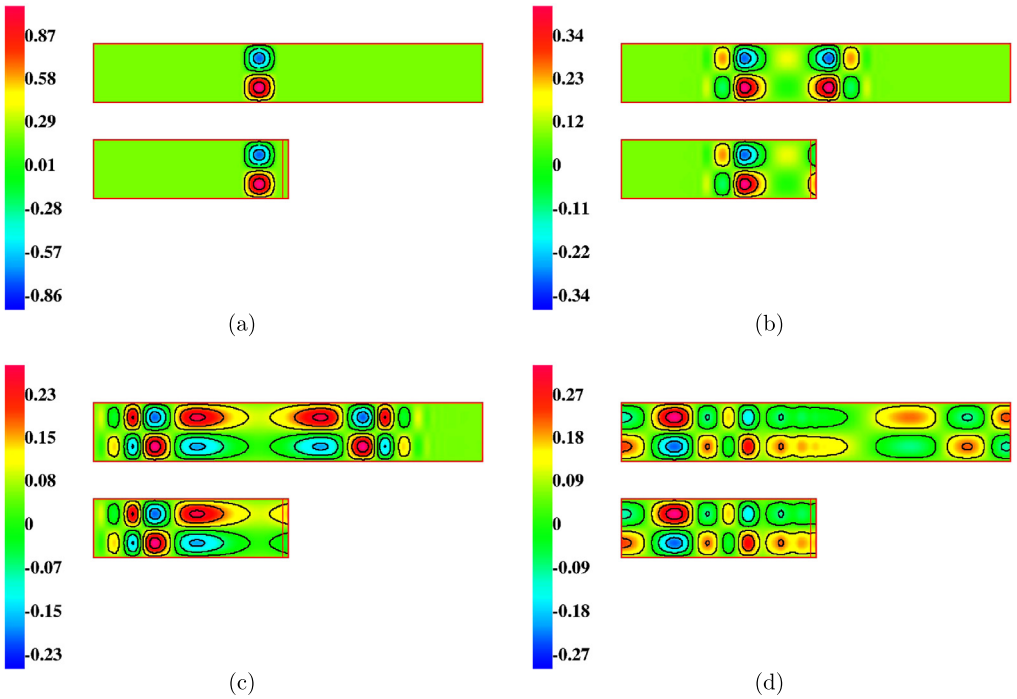

Fig. 8. Pulse propagation with periodic north and south conditions: snapshots of reference solution $u^{h,ref}$ (upper) and actual solution u^h (lower), at (a) $t = 0$; (b) $t = 2$; (c) $t = 4.5$; (d) $t = 10$.

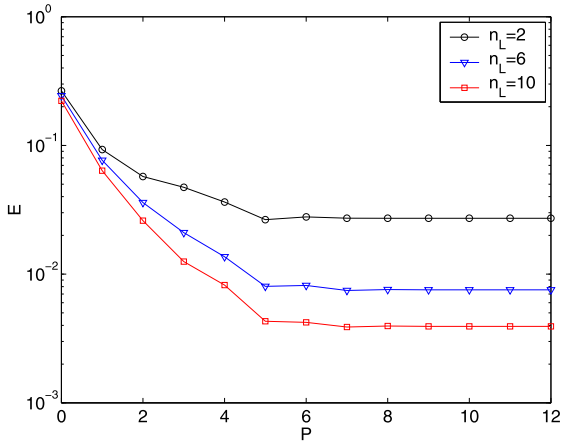

Fig. 9. Pulse propagation: space–time error as a function of P.

Snapshots of the solution for u at various times for $P = 10$ and with $n_L = N_L - 1 = 6$ elements across the width of Ω_L are shown on Fig. 8. Each snapshot consists of two sub-plots: the upper one depicts the reference solution $u^{h,ref}$ and the lower one is the computed solution u^h in the truncated domain using the double ABC. In the lower plot, the layer is indicated by showing the vertical boundary Γ_I. The plots show that there is excellent agreement between the reference and actual solutions.

Fig. 9 shows the global space–time error E as a function of the ABC order P for three values of layer thickness n_L. As in the 1D FD scheme, a thicker layer leads to higher accuracy, as the figure shows. In addition, for all values of n_L, the error decreases with increasing P and "saturates" at about $P = 6$. (This is in contrast to the super-convergent results obtained in 1D.) This level of accuracy is the level of discretization error, below which the error cannot be reached without refining the mesh and time-step. The error obtained with $n_L = 6$ and large P is about 0.4%. In comparison, for $P = 0$ (Sommerfeld condition) the error is about 25%.

Fig. 10 shows the evolution of the space-error $e(t)$ during the simulation for the case $n_L = 6$ and $P = 10$. Note that in this example the initial pulse is located very close to the layer, and arrives at Γ_E at time $t = 0.5/\sqrt{3} = 0.3$. At this time

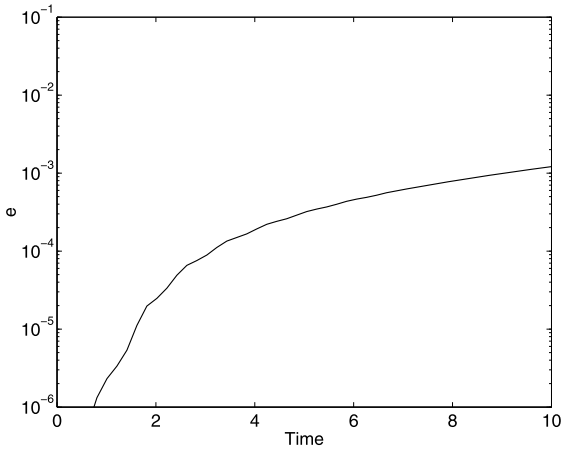

Fig. 10. Pulse propagation: space-error as a function of time, for $n_L = 6$ and $P = 10$.

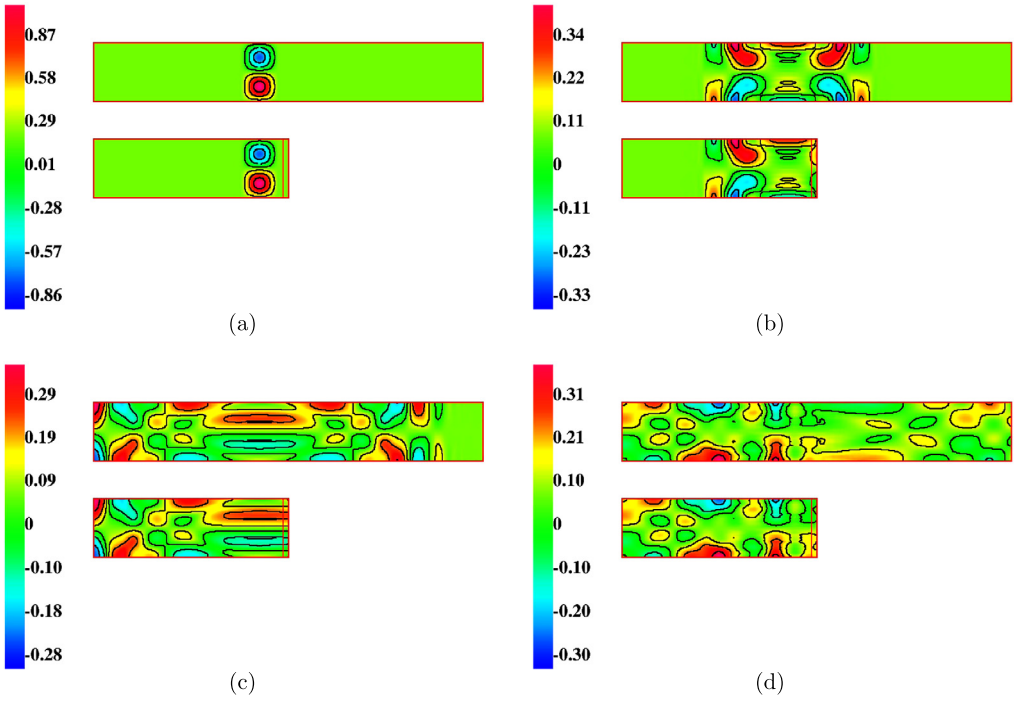

Fig. 11. Pulse propagation with zero Neumann north and south conditions: snapshots of reference solution $u^{h,ref}$ (upper) and actual solution u^h (lower), at (a) $t = 0$; (b) $t = 2$; (c) $t = 5$; (d) $t = 10$.

there is a certain "jump" in the error, but the error is so small that it is below the lowest value in the log scale shown in the figure.

We now repeat this experiment, but using homogeneous Neumann conditions (Eq. (72)) instead of periodic conditions on the north and south boundaries. Snapshots of the solution at various times for $P = 10$ and with $n_L = 6$ are shown in Fig. 11. This figure should be compared with Fig. 8 for the periodic case. The solution pattern is completely different, and the periodic wave packet structure is replaced here with rich multi-frequency content. Again the agreement between the actual and reference solutions is excellent. Error graphs for this case are very similar to those shown in Fig. 9 and Fig. 10 for the periodic case, and hence are omitted here.

7.3. Surface loading

In this example we take zero initial conditions and apply a zero Neumann boundary condition on the south boundary and a persistent surface loading on the north boundary; i.e., at $y = b$ we have $c^2 u_{,y} = h$, where

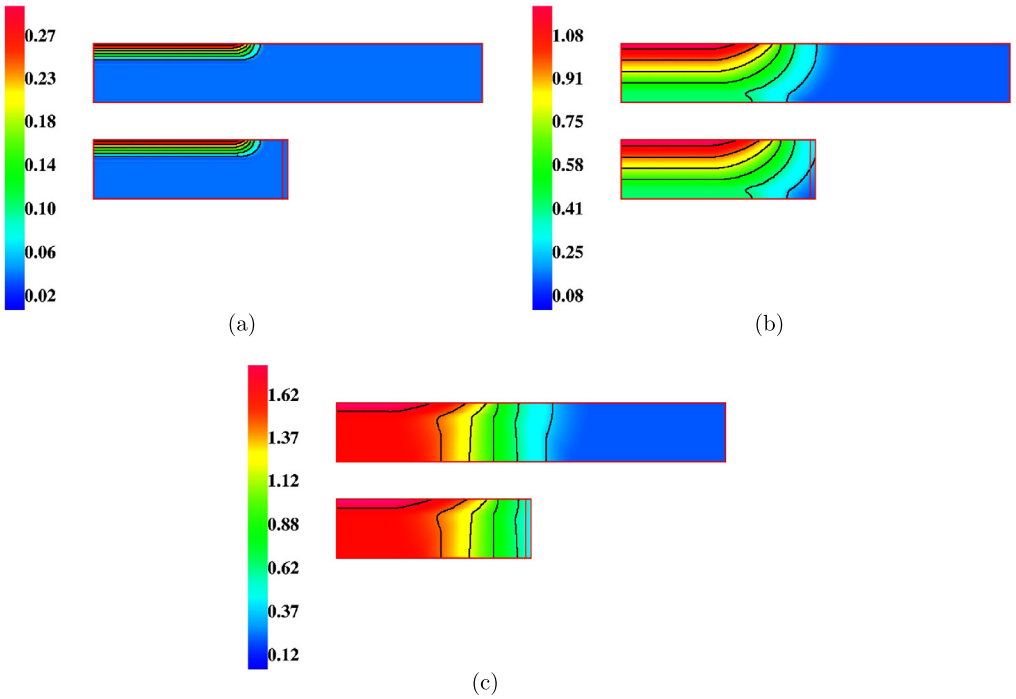

Fig. 12. Persistent surface load: snapshots of reference solution $u^{h,ref}$ (upper) and actual solution u^h (lower), at (a) $t = 0$; (b) $t = 2$; (c) $t = 3$.

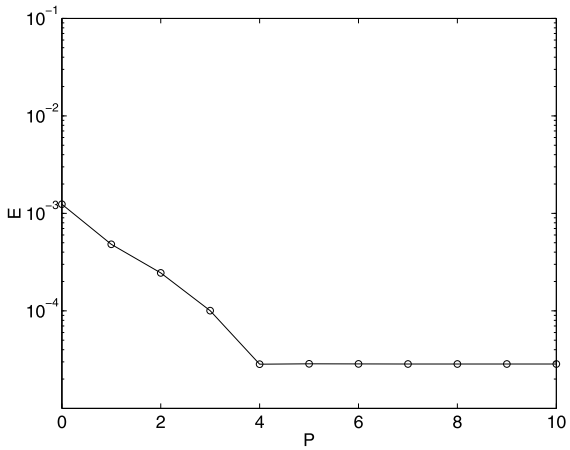

Fig. 13. Persistent surface load: space–time error as a function of P, for $n_L = 6$.

$$h(x,t) = \begin{cases} 1 & \text{for } 0 \leqslant x \leqslant 8, \\ 0 & \text{for } x > 8. \end{cases} \tag{91}$$

Note that the surface load starts to act at $t = 0$ and remains constant in time. This problem has been chosen since it has served as a benchmark problem in some of our previous publications using other methods. At early times oblique waves impinge on the south part of the artificial boundary Γ_E, which may potentially cause slow convergence with respect to increasing P; see, e.g., [14] for a similar problem in the context of elasticity.

The actual and reference solutions for $P = 10$ and $n_L = 6$ are shown in Fig. 12. Snapshots are shown for the early times $t = 0.5, 2, 3$, since in this example steady state is reached rapidly, and the solution does not change much at later times. As in the previous example, the agreement between the reference and actual solutions is excellent.

We fix $n_L = 6$. Fig. 13 shows the time–space error E as a function of P. The convergence seems to be as fast as in the pulse propagation example. The error values are much smaller here. Fig. 14 shows the evolution of the space-error $e(t)$ during the simulation, for $P = 10$. No large errors are observed in early times.

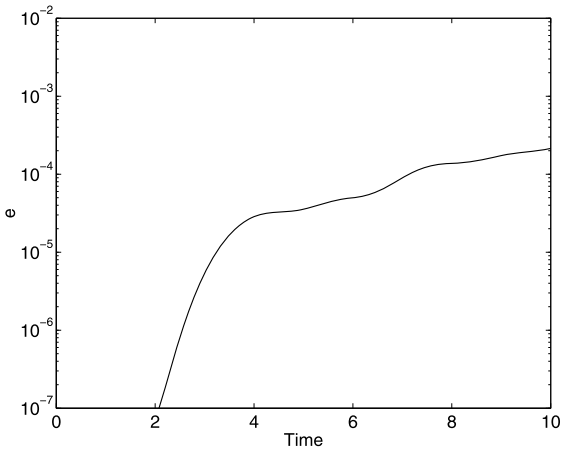

Fig. 14. Persistent surface load: space-error as a function of time, for $P = 10$ and $n_L = 6$.

8. Concluding remarks

In this paper the foundation of the Double Absorbing Boundary (DAB) method for solving unbounded domain problems has been laid. This method shares some of the properties of using ABC on a single boundary and of PML, but has some relative advantages with respect to both. Extensive comparison of DAB and PML merits a separate investigation that is still to be done.

Several important extensions will be considered in our future work. Four of them are:

- The behavior of DAB in the presence of corners joining two straight artificial boundaries will be investigated. The use of DAB on a boundary with corners is expected to be as easy as the case with PML, and thus to be free of the difficulties associated with high-order ABCs on a single boundary in the presence of corners.
- The application of DAB for heterogeneous media will be studied. Since the ABCs used in this method involve only derivatives normal to the artificial boundaries Γ_I and Γ_E, it is expected that extension to a layered medium (with straight layers whose interfaces are normal to Γ_I and Γ_E) should be straight forward.
- Extension to an anisotropic medium also seems possible, although it would require a more involved adaptation. It would be especially interesting to see how DAB behaves in those cases of anisotropy where standard PMLs are unstable.
- The possibility to apply the DAB approach to first-order hyperbolic systems will be explored.
- Extension of the DAB method to elastodynamics is currently underway, with excellent preliminary results, and will be reported in a forthcoming publication.

Acknowledgements

This work was supported by the US–Israel Binational Science Foundation (BSF), grant number 890020 (Technion number 2011303). The work of D.G. was also supported by the fund provided through the Lawrence and Marie Feldman Chair in Engineering and by the Technion's Seniel Ostrow Research Fund; that of T.H. by NSF grant OCI-0904773 and ARO grant W911NF-09-1-0344; and that of J.B. by the U.S. National Science Foundation Award No. NSF OCI-0749227. Any conclusions or recommendations in the paper are those of the authors and do not necessarily represent the views of the BSF, NSF or ARO. The authors also thank Eliane Bécache for her comments on the manuscript.

Appendix A. Element-level expressions for the 2D FE formulation

The element-level expressions for the arrays appearing in (83)–(85) may be extracted from (76)–(78). To write them down, we first define the following generic quantities:

$$I^1_{ab} = \int_{\Omega^e} N_a N_b \, d\Omega, \tag{A.1}$$

$$I^2_{ab} = \int_{\Omega^e} N_{a,k} N_{b,l} \, d\Omega, \tag{A.2}$$

$$I^3_{ab} = \int_{\Gamma_E} N_a N_b \, d\Gamma, \tag{A.3}$$

$$I^4_{ab} = \int_{\Gamma_I} N_a N_b \, d\Gamma, \tag{A.4}$$

$$I^5_{ab} = \int_{\Gamma_E} N_a N_{b,x} \, d\Gamma, \tag{A.5}$$

$$I^6_{ab} = \int_{\Gamma_I} N_a N_{b,x} \, d\Gamma. \tag{A.6}$$

Here the N_a are the element-level shape functions (see (81) and (82)). Then the element-level arrays are as follows. The arrays associated with Eq. (83) are:

$$(M_0)^e_{ab} = I^1_{ab}, \tag{A.7}$$

$$(C_0)^e_{ab} = cI^3_{ab}, \tag{A.8}$$

$$(K_0)^e_{ab} = c^2 I^2_{ab} + sI^1_{ab}, \tag{A.9}$$

$$(G_0)^e_{ab} = -cI^3_{ab}, \tag{A.10}$$

$$(H_0)^e_{ab} = c^2 I^5_{ab}. \tag{A.11}$$

The element-level arrays associated with Eq. (84) are:

$$(M_j)^e_{ab} = I^1_{ab}, \tag{A.12}$$

$$(C_j)^e_{ab} = cI^3_{ab} + cI^4_{ab}, \tag{A.13}$$

$$(K_j)^e_{ab} = c^2 I^2_{ab} + sI^1_{ab}, \tag{A.14}$$

$$(A_j)^e_{ab} = -cI^4_{ab}, \tag{A.15}$$

$$(B_j)^e_{ab} = c^2 I^6_{ab}, \tag{A.16}$$

$$(G_j)^e_{ab} = -cI^3_{ab}, \tag{A.17}$$

$$(H_j)^e_{ab} = c^2 I^5_{ab}. \tag{A.18}$$

Finally, the element-level arrays associated with Eq. (85) are:

$$(M_P)^e_{ab} = I^1_{ab}, \tag{A.19}$$

$$(C_P)^e_{ab} = cI^4_{ab}, \tag{A.20}$$

$$(K_P)^e_{ab} = c^2 I^2_{ab} + sI^1_{ab}, \tag{A.21}$$

$$(A_P)^e_{ab} = -cI^4_{ab}, \tag{A.22}$$

$$(B_P)^e_{ab} = c^2 I^6_{ab}. \tag{A.23}$$

The global arrays appearing in (83)–(85) are obtained from the element arrays (A.7)–(A.23) via the usual FE assembly process.

References

[1] D. Givoli, Non-reflecting boundary conditions: A review, J. Comput. Phys. 94 (1991) 1–29.
[2] S.V. Tsynkov, Numerical solution of problems on unbounded domains. A review, Appl. Numer. Math. 27 (1998) 465–532.
[3] T. Hagstrom, Radiation boundary conditions for the numerical simulation of waves, Acta Numer. 8 (1999) 47–106.
[4] D. Givoli, Computational absorbing boundaries, in: S. Marburg, B. Nolte (Eds.), Computational Acoustics of Noise Propagation in Fluids, Springer, Berlin, 2008, pp. 145–166, Chapter 5.
[5] J.P. Bérenger, A perfectly matched layer for the absorption of electromagnetic waves, J. Comput. Phys. 114 (1994) 185–200.
[6] A. Bermudez, L. Hervella-Nieto, A. Prieto, R. Rodriguez, Perfectly matched layers for time-harmonic second order elliptic problems, Arch. Comput. Methods Eng. 17 (2010) 77–107.
[7] F. Collino, High order absorbing boundary conditions for wave propagation models. Straight line boundary and corner cases, in: R. Kleinman, et al. (Eds.), Proc. 2nd Int. Conf. on Mathematical & Numerical Aspects of Wave Propagation, SIAM, Delaware, 1993, pp. 161–171.
[8] D. Givoli, High-order local non-reflecting boundary conditions: A review, Wave Motion 39 (2004) 319–326.
[9] B. Engquist, A. Majda, Radiation boundary conditions for acoustic and elastic wave calculations, Commun. Pure Appl. Math. 32 (1979) 313–357.
[10] A. Bayliss, E. Turkel, Radiation boundary conditions for wave-like equations, Commun. Pure Appl. Math. 33 (1980) 707–725.
[11] D. Rabinovich, D. Givoli, E. Bécache, Comparison of high-order absorbing boundary conditions and perfectly matched layers in the frequency domain, Int. J. Numer. Methods Biomed. Eng. 26 (2010) 1351–1369 (formerly Commun. Numer. Meth. Engng.).
[12] S. Asvadurov, V. Druskin, M. Guddati, L. Knizherman, On optimal finite difference approximation of PML, SIAM J. Numer. Anal. 41 (2003) 287–305.

[13] T. Hagstrom, T. Warburton, A new auxiliary variable formulation of high-order local radiation boundary conditions: Corner compatibility conditions and extensions to first order systems, Wave Motion 39 (2004) 327–338.
[14] D. Rabinovich, D. Givoli, J. Bielak, T. Hagstrom, A finite element scheme with a high order absorbing boundary condition for elastodynamics, Comput. Methods Appl. Mech. Eng. 200 (2011) 2048–2066.
[15] T. Hagstrom, T. Warburton, Complete radiation boundary conditions: minimizing the long time error growth of local methods, SIAM J. Numer. Anal. 47 (2009) 3678–3704.
[16] T. Hagstrom, A. Mar-Or, D. Givoli, High-order local absorbing conditions for the wave equation: Extensions and improvements, J. Comput. Phys. 227 (2008) 3322–3357.
[17] E. Bécache, D. Givoli, T. Hagstrom, High order absorbing boundary conditions for anisotropic and convective wave equations, J. Comput. Phys. 229 (2010) 1099–1129.
[18] T. Hagstrom, E. Bécache, D. Givoli, K. Stein, Complete radiation boundary conditions for convective waves, Commun. Comput. Phys. 11 (2012) 610–628.
[19] A. Mar-Or, D. Givoli, High order global-regional model interaction: Extension of Carpenter's scheme, Int. J. Numer. Methods Eng. 77 (2009) 50–74.
[20] D. Baffet, J. Bielak, D. Givoli, T. Hagstrom, D. Rabinovich, Long-time stable high-order absorbing boundary conditions for elastodynamics, Comput. Methods Appl. Mech. Eng. 241–244 (2012) 20–37.
[21] D. Rabinovich, D. Givoli, T. Hagstrom, J. Bielak, Stress-velocity complete radiation boundary conditions, J. Comput. Acoust. (2013), accepted for publication.
[22] D. Givoli, I. Patlashenko, Dirichlet-to-Neumann boundary condition for time-dependent dispersive waves in three-dimensional guides, J. Comput. Phys. 199 (2004) 339–354.
[23] J.S. Hesthaven, T. Warburton, Nodal Discontinuous Galerkin Methods, Springer, New York, 2008.
[24] S. Kim, T. Hagstrom, Complete radiation boundary conditions for the Helmholtz equation, I: Waveguides, preprint, 2013.
[25] T. Hagstrom, M. de Castro, D. Givoli, D. Tsemach, Local high order absorbing boundary conditions for time-dependent waves in guides, J. Comput. Acoust. 15 (2007) 1–22.
[26] D. Givoli, T. Hagstrom, I. Patlashenko, Finite element formulation with high order absorbing boundary conditions for time-dependent waves, Comput. Methods Appl. Mech. Eng. 195 (2006) 3666–3690.
[27] L. Kucherov, D. Givoli, High order absorbing boundary conditions incorporated in a spectral element formulation, Int. J. Numer. Methods Biomed. Eng. 26 (2010) 1130–1143 (formerly Commun. Numer. Meth. Engng.).
[28] M. Goldberg, E. Tadmor, Convenient stability criteria for difference approximations of hyperbolic initial-boundary value problems, Math. Comput. 44 (1985) 361–377.
[29] D. Givoli, D. Cohen, Non-reflecting boundary conditions based on Kirchhoff-type formulae, J. Comput. Phys. 117 (1995) 102–113.
[30] T.J.R. Hughes, The Finite Element Method, Prentice Hall, Englewood Cliffs, NJ, 1987.