

An exact non reflecting boundary condition for 2D time-dependent wave equation problems

Original

An exact non reflecting boundary condition for 2D time-dependent wave equation problems / Falletta, Silvia; Monegato, Giovanni. - In: WAVE MOTION. - ISSN 0165-2125. - STAMPA. - 51:1(2014), pp. 168-192.
[10.1016/j.wavemoti.2013.06.001]

Availability:

This version is available at: 11583/2507846 since: 2019-09-04T12:27:49Z

Publisher:

Elsevier, Amsterdam Netherlands

Published

DOI:10.1016/j.wavemoti.2013.06.001

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

Elsevier postprint/Author's Accepted Manuscript

© 2014. This manuscript version is made available under the CC-BY-NC-ND 4.0 license
<http://creativecommons.org/licenses/by-nc-nd/4.0/>. The final authenticated version is available online at:
<http://dx.doi.org/10.1016/j.wavemoti.2013.06.001>

(Article begins on next page)

An exact non-reflecting boundary condition for 2D time-dependent wave equation problems.[☆]

Silvia Falletta, Giovanni Monegato^{a,b}

^a*Dipartimento di Scienze Matematiche, Politecnico di Torino, Italy. Email: silvia.falletta@polito.it*

^b*Dipartimento di Scienze Matematiche, Politecnico di Torino, Italy. Email: giovanni.monegato@polito.it*

Abstract

We consider some 2D wave equation problems defined in an unbounded domain, possibly with far field sources. For their solution, by means of standard finite element or finite difference methods, we propose a Non Reflecting Boundary Condition (NRBC) on the chosen artificial boundary \mathcal{B} , which is based on a known space-time integral equation defining a relationship between the solution of the differential problem and its normal derivative on \mathcal{B} . Such a NRBC is exact, non local both in space and time. We discretize it by using a fast convolution quadrature technique in time and a collocation method in space. Besides showing a good accuracy and numerical stability, the proposed NRBC has the property of being suitable for artificial boundaries of general shapes; moreover, from the computational point of view, it is competitive with well known existing NRBC of local type. It also allows the treatment of far field sources, that do not have to be necessarily included in the finite computational domain, being transparent for both incoming and outgoing waves.

Keywords: wave equation; absorbing boundary conditions; space-time boundary integral equations; numerical methods

1. Introduction

Infinite or unbounded domains are often encountered in mathematical models associated with acoustic, aerodynamic, geophysical, electromagnetic and many other problems. Typically, the phenomenon of interest is local but embedded in a vast surrounding medium. Although the exterior region may not be truly unbounded, the boundary effects are often negligible, so that one further simplifies the problem by replacing the vast exterior by an infinite medium. Mathematical models of natural phenomena usually consist of Partial Differential Equations (PDE), and many standard numerical methods, such as Finite Differences (FD) and Finite Elements (FE), can be used to solve them. These can even handle complex geometries, inhomogeneous media and nonlinearity. However, they require a finite computational domain with prescribed boundary conditions. A key issue is therefore the choice of a bounded computational domain, where one is interested in studying the behavior of the solution, and the introduction of boundary conditions which guarantee that the solution of

[☆]This work was supported by the Ministero dell'Istruzione, dell'Università e della Ricerca of Italy, under the research program PRIN09: Boundary element methods for time dependent problems.

the initial boundary value problem inside the finite computational domain coincides with the restriction to the computational domain of the solution of the original problem, which is defined in the infinite region. The method of Artificial (or Absorbing) Boundary Condition (ABC) consists of introducing an artificial boundary \mathcal{B} that truncates the infinite domain and determines two distinct regions: a bounded domain of interest Ω and a residual infinite domain \mathcal{D} . By analyzing the problem in \mathcal{D} , a Non Reflecting Boundary Condition (NRBC) on \mathcal{B} is derived in order to avoid spurious reflections. Once the NRBC is given, it is used to solve the problem in Ω by using a numerical method such as, for example, finite differences, finite elements, finite volumes or spectral methods.

The NRBC are usually divided into two main categories: exact, that is *non local*, both in space and time, and approximate, i.e., *local* both in space and time.

In the context of exact NRBC, we mention the Dirichlet to Neumann (DtN) method proposed by Givoli [12, 13] for elliptic problems, where the exact artificial boundary condition is based on the DtN map associated with the operator and geometry under consideration. Such a map has been derived for various elliptic linear differential equations ([15, 16, 11, 10, 18, 38, 39, 4, 21]) and has also been applied to time-dependent wave problems ([12, 14, 19]). It is limited to circular and spherical artificial boundaries, and its main drawback stands in the fact that the boundary conditions also depends on the behavior of the solution in the residual infinite domain \mathcal{D} ; thus it requires further efforts to somehow truncate and approximate \mathcal{D} . In [32, 1, 22], the authors proposed fast, low-memory implementations of exact non local conditions, but limited to rectangular or circular (spherical) geometries which, especially in the three dimensional case, could be excessively large with respect to the domain of interest.

The earlier approximate NRBC, still widely used, are those proposed by Engquist and Majda [7]. By using the Laplace-Fourier transform in time and in the plane tangential to the artificial boundary, they derived exact boundary conditions in terms of a pseudo-differential operator, which is then localized through a Taylor approximation up to the second order, and a Padé approximation for higher orders ([5]). Higdon [28, 29] derived a NRBC which is exact for any linear combination of plane waves whose angles of incidence are fixed a priori. In both cases, the high-order derivatives involved in these boundary conditions greatly complicate their use in any numerical scheme. As a result, first and second order boundary conditions were commonly used in practice. Later, high order local non reflecting boundary conditions for the wave equation, which do not involve derivatives of order greater than 2, were proposed by Collino [6], Givoli and Neta [17] and Hagstrom and Warburton [27]. In all these three cases, the methods proposed require a straight-edge boundary, and special treatment of the corner effects. A few years after the appearance of the Engquist and Majda paper [7], Bayliss and Turkel (see [3]) presented a new (approximate) ABC of arbitrarily high order, when \mathcal{B} is a disk/sphere, whose truncation error tends to zero as the radius R of \mathcal{B} tends to infinity. Much later, Hagstrom and Hariharan [26] derived a new formulation of this NRBC, without using derivatives of order greater than 2. For a review of high order local NRBC, not involving high order derivatives, see [14].

Many more papers have been published on this topic, in particular in the last decade; but their number is too large to mention them. We recall, however, that in the case of time harmonic waves, that is, in the case of the Helmholtz equation, local NRBC have been

derived also for elliptical shaped artificial boundaries; see, for example, [35], [36] and their references.

All these papers deal with the construction of NRBC with the property of absorbing only outgoing wave, not outgoing and incoming ones. Therefore, sources must necessarily be included in the computational domain, and this can be a severe drawback. We recall however that in [41], Sections 5.5, 5.6, a finite-difference time-domain method is described, which is capable to deal also with incoming plane waves. These are incorporated using a combined scattered/total field representation.

In the present work, we consider the (non homogeneous) exterior Dirichlet problem for the classical 2D wave equation, and propose for its solution a NRBC which arises from a known space-time boundary integral relationship that the problem solution and its normal derivative must satisfy at the chosen artificial boundary \mathcal{B} . It holds for a (smooth) curve of arbitrary shape; therefore, it can be used also in situations of multiple scattering (see [23], [24]), and even in more general ones. Moreover, it allows the problem to have non trivial data, whose (local) supports do not have necessarily to be included in the Ω domain, as it is usually done, in particular when they are away from the domain of interest. In such a case, our NRBC will naturally include the effects of these data and will be transparent for both outgoing and incoming waves. Such an ABC condition is of exact type and it is given by a linear combination of a single and a double layer operators, which are well known in Boundary Integral Equation (BIE) formulations of PDE problems.

For the discretization of the artificial condition, namely for the approximation of the single and double layer operators, we propose a numerical scheme which is based on a second order Lubich discrete convolution quadrature formula for the discretization of the time integral, coupled with a classical collocation method in space. An estimate for the behaviour of the associated (global) truncation error is derived. When the discretization of the bounded domain Ω , where we apply the chosen finite element or finite difference scheme, is refined, and the time stepsize is simultaneously reduced, the accuracy of the NRBC discretization increases.

If N denotes the number of time steps to be performed, the proposed discretized NRBC requires $O(N \log N)$ operations to compute, for each given collocation (space) point, the associated temporal convolution at all chosen instants. Moreover, when the artificial boundary is a circle, the space non locality is comparable, from the computational point of view, to a locality property. Indeed, in such a case, if M denotes the number of (equidistant) collocation points on the boundary, the computational cost of the construction of the proposed NRBC turns out to be $O(MN \log N)$, which is almost that of a NRBC of local type, both in space and time. For an artificial boundary of general shape, taking advantage of the “sparsity” of the involved (approximate) matrices, the overall CPU time can be reduced significantly, as well as the size of the required storage. This issue is discussed in Section 3.2.

The final numerical scheme is then obtained by coupling such non reflecting condition with a second order finite element or finite difference method. The numerical examples that we present in the final section show that indeed the proposed NRBC is very competitive, from both the accuracy and the computational cost points of view.

The paper is organized as follows. In Section 2 we introduce the proposed space-time artificial boundary condition, and present three possible ways of representing the restriction

of the original PDE problem to the bounded region of interest. In Section 3, first we discretize the chosen formulation, to determine the solution of the original problem in the (bounded) domain of interest, by means of a (space) finite element method, combined with a classical one-step time integration scheme. Then, we apply the proposed numerical approach to some test problems. Several remarks are made.

In Section 4, we replace the (space) finite element discretization by a classical finite difference scheme, and apply the resulting numerical method to a test problem. In this case, we compare the CPU time required by our approach with those required by the corresponding schemes associated with the second order Engquist-Majda and Bayliss-Turkel ABC.

Finally, in Section 5 we make some further comments and draw some conclusions.

2. The model problem

Let $\Omega^i \subset \mathbb{R}^2$ be an open bounded domain with a sufficiently smooth boundary Γ ; define $\Omega^e = \mathbb{R}^2 \setminus \bar{\Omega}^i$. We consider the following wave propagation problem in Ω^e :

$$\begin{cases} u_{tt}^e(\mathbf{x}, t) - \Delta u^e(\mathbf{x}, t) &= f(\mathbf{x}, t) & \text{in } \Omega^e \times (0, T) \\ u(\mathbf{x}, t) &= g(\mathbf{x}, t) & \text{in } \Gamma \times (0, T) \\ u^e(\mathbf{x}, 0) &= u_0(\mathbf{x}) & \text{in } \Omega^e \\ u_t^e(\mathbf{x}, 0) &= v_0(\mathbf{x}) & \text{in } \Omega^e. \end{cases} \quad (1)$$

As often occurs in practical applications, we assume that the initial condition u_0 , the initial velocity v_0 and the source term f are either trivial or have a local support. Since in general one has to determine the solution u^e of the above problem in a bounded subregion of Ω^e , surrounding the physical domain Ω^i , we truncate the infinite domain Ω^e by introducing an artificial smooth boundary \mathcal{B} . This boundary divides Ω^e into two (open) sub-domains: a finite computational domain Ω , which is bounded internally by Γ and externally by \mathcal{B} , and an infinite residual domain \mathcal{D} .

We also assume that the data are smooth and satisfy the required compatibility conditions, which guarantee that the (unknown) solution $u(\mathbf{x}, t)$ is at least C^2 with respect to both variables. That is, we assume (see [9]) $f \in C([0, T], C^2(\bar{\Omega}^e))$, $u_0 \in C^3(\bar{\Omega}^e)$, $v_0 \in C^2(\bar{\Omega}^e)$ and the Dirichlet data $g \in C^2([0, T]; C(\Gamma))$ satisfying the following compatibility conditions:

$$g(\mathbf{x}, 0) = u_0(\mathbf{x})|_{\Gamma}, \quad g_t(\mathbf{x}, 0) = v_0(\mathbf{x})|_{\Gamma}, \quad g_{tt}(\mathbf{x}, 0) = \Delta u_0(\mathbf{x})|_{\Gamma} - f(\mathbf{x}, 0) \quad (2)$$

that in our case reduce to

$$g(\mathbf{x}, 0) = g_t(\mathbf{x}, 0) = g_{tt}(\mathbf{x}, 0) = 0$$

since all data have compact supports. This in turn trivially implies that $u(\mathbf{x}, 0) = u_t(\mathbf{x}, 0) = u_{tt}(\mathbf{x}, 0) = 0, \mathbf{x} \in \Gamma$.

Nevertheless, in Section 3.3 we will also consider some examples where not all these conditions are satisfied.

The artificial boundary is chosen to detect the (bounded) region where one has to compute the problem solution. This region does not necessarily have to contain the supports of the

source term and of the initial data. Thus, in general, the support of a datum will be either in the (bounded) region of interest Ω , or in the residual domain \mathcal{D} . In the latter case it will be taken into account by a corresponding term of the artificial boundary condition formulation.

Remark 2.1. *Although our model problem (1) has constant wave speed, the NRBC we propose equally applies to the case of a non constant wave speed, as long as this decays to a constant value, for all $t \geq 0$, on the chosen artificial boundary \mathcal{B} , and maintain this (constant) value in all of \mathcal{D} .*

To obtain a well posed problem in Ω , we need to impose a proper boundary condition on \mathcal{B} . To do so, we analyze the problem in \mathcal{D} , and we impose on \mathcal{B} the integral relation that the solution u and its (outward) normal derivative $\partial_{\mathbf{n}_D} u$ have to satisfy. Thus, for our problem (1) we introduce the associated single and double layer integral operators, defined by

$$\mathcal{V}\psi(\mathbf{x}, t) := \int_0^t \int_{\mathcal{B}} G(\mathbf{x} - \mathbf{y}, t - \tau) \psi(\mathbf{y}, \tau) d\mathcal{B}_{\mathbf{y}} d\tau,$$

and

$$\mathcal{K}\varphi(\mathbf{x}, t) := \int_0^t \int_{\mathcal{B}} \partial_{\mathbf{n}_D} G(\mathbf{x} - \mathbf{y}, t - \tau) \varphi(\mathbf{y}, \tau) d\mathcal{B}_{\mathbf{y}} d\tau,$$

respectively, where $G(\mathbf{x}, t)$ denotes the \mathbb{R}^2 fundamental solution of the wave equation given in (1), that is:

$$G(\mathbf{x}, t) = \frac{1}{2\pi} \frac{H(t - \|\mathbf{x}\|)}{\sqrt{t^2 - \|\mathbf{x}\|^2}},$$

$\delta(\cdot)$, $H(\cdot)$ being the well known Dirac delta and Heaviside functions. Moreover, recalling the representations (see [9]):

$$I_{u_0}(\mathbf{x}, t) = \frac{\partial}{\partial t} \int_{\mathcal{D}} u_0(y) G(\mathbf{x} - \mathbf{y}, t) dy, \quad (3)$$

$$I_{v_0}(\mathbf{x}, t) = \int_{\mathcal{D}} v_0(y) G(\mathbf{x} - \mathbf{y}, t) dy, \quad (4)$$

$$I_f(\mathbf{x}, t) = \int_0^t \int_{\mathcal{D}} f(y, \tau) G(\mathbf{x} - \mathbf{y}, t - \tau) dy d\tau, \quad (5)$$

for the possible “volume” terms generated by the non trivial source and the non homogeneous initial conditions, following the theory of the standard boundary integral equations, we obtain (for more details see [9]) the (boundary) relationship

$$\frac{1}{2}u(\mathbf{x}, t) = \mathcal{V}\partial_{\mathbf{n}_D} u(\mathbf{x}, t) - \mathcal{K}u(\mathbf{x}, t) + I_{u_0}(\mathbf{x}, t) + I_{v_0}(\mathbf{x}, t) - I_f(\mathbf{x}, t) \quad \mathbf{x} \in \mathcal{B}. \quad (6)$$

Contrarily to what happens in the Boundary Element Method (BEM) method, equation (6) has not to be solved (neither u nor $\partial_{\mathbf{n}_D} u$ are given on \mathcal{B}), but it represents the natural relation that u and its normal derivative have to satisfy at each point of the artificial boundary and at each time t . Relation (6) represents an exact NRBC, non local both in time and in space.

Remark 2.2. We have assumed that the problem data satisfy the smoothness and compatibility conditions which guarantee that $u(\mathbf{x}, t) \in C^2(\overline{\Omega^e} \times [0, T])$. Moreover, we only consider data whose supports are included either in Ω or in \mathcal{D} , which means that the functions u_0, v_0, f are certainly null in a whole (two-sided) \mathbb{R}^2 neighborhood of the artificial boundary \mathcal{B} . From these properties, we immediately deduce that $u(\mathbf{x}, t)$ and $\partial_{\mathbf{n}_D} u(\mathbf{x}, t)$, together with their first two time derivatives, must all vanish on \mathcal{B} at $t = 0$.

We recall, however, that the theory developed by Lubich (see [34]), for the convolution quadratures that we will use in Section 3.1 to discretize the time integrals of the operators \mathcal{V} and \mathcal{K} defined below, requires data with even higher orders of smoothness and compatibility. For these, see also Remark 4.2 in [9].

The mapping properties of the operators \mathcal{V}, \mathcal{K} have been studied; see, for example, [34], [30]. The notation and the mathematics required to derive them is fairly heavy. Therefore, we simply outline a few of the main ingredients and results. To this end, we denote (see [30]) by $\mathcal{A}(\mu, X, Y)$, X, Y being two Hilbert spaces, the set of analytic functions $F : \mathbb{C}_+ \rightarrow \mathcal{L}(X, Y)$ for which there exists a real number μ such that, given any $\sigma > 0$, for the associated operator $\mathbf{F}(s) : X \rightarrow Y$ we have $\|\mathbf{F}(s)\| \leq C_0 |s|^\mu$, for some $C_0 = C_0(\sigma)$ and all $s : \text{Re}(s) > \sigma$. In the following we denote by \widehat{K} the (time) Laplace transform of K .

It is then known ([30]) that for the kernels of the operators

$$\widehat{\mathcal{V}}(s)\Psi(\mathbf{x}) := \int_{\mathcal{B}} \widehat{\mathcal{V}}(\mathbf{x} - \mathbf{y}, s)\Psi(\mathbf{y})d\mathcal{B}_{\mathbf{y}}, \quad \mathcal{V} = G(\mathbf{x} - \mathbf{y}, t)$$

and

$$\widehat{\mathcal{K}}(s)\Phi(\mathbf{x}) := \int_{\mathcal{B}} \widehat{\mathcal{K}}(\mathbf{x} - \mathbf{y}, s)\Phi(\mathbf{y})d\mathcal{B}_{\mathbf{y}}, \quad \mathcal{K} = \partial_{\mathbf{n}_D} G(\mathbf{x} - \mathbf{y}, t)$$

i.e., of the single and double-layer operators on the boundary for the Helmholtz equation $s^2 u - \Delta u = 0$, respectively, we have:

$$\widehat{\mathcal{V}} \in \mathcal{A}(1, H^{-1/2}(\mathcal{B}), H^{1/2}(\mathcal{B})), \quad \widehat{\mathcal{K}} \in \mathcal{A}(3/2, H^{1/2}(\mathcal{B}), H^{1/2}(\mathcal{B}))$$

where $H^\alpha(\mathcal{B})$ is the classical Sobolev space of order α , defined on the closed curve \mathcal{B} . From these, and recalling that $u(\mathbf{y}, 0)$ and $\partial_{\mathbf{n}_D} u(\mathbf{y}, 0)$, together with their first time derivatives, vanish at $t = 0$ (see Remark 2.2), we obtain (see also [34], p.368) the following mapping properties for the (bounded) operators \mathcal{V} and \mathcal{K} :

$$\mathcal{V} : H_0^{r+1}(0, T; H^{-1/2}(\mathcal{B})) \rightarrow H_0^r(0, T; H^{1/2}(\mathcal{B})), r \geq 0 \quad (7)$$

and

$$\mathcal{K} : H_0^{r+3/2}(0, T; H^{1/2}(\mathcal{B})) \rightarrow H_0^r(0, T; H^{1/2}(\mathcal{B})), r \geq 0. \quad (8)$$

These spaces are defined as follows. Set first $H_0^r(0, T) = \{h|_{(0, T)} : h \in H^r(\mathbb{R}) \text{ with } h \equiv 0 \text{ on } (-\infty, 0)\}$, where $H^r(\mathbb{R})$ denotes the classical Sobolev space of order r defined on the real line. When r is an integer, this space consists of those functions g whose r -th distributional derivative is in $L^2(0, T)$ and which have $h(0) = \dots h^{(r-1)}(0) = 0$. Then:

- $H_0^r(0, T; X)$ is the space of $H_0^r(0, T)$ functions of t , $\phi(\mathbf{x}, t)$, such that, setting $\phi(\mathbf{x}, t) = \phi(t)(\mathbf{x})$, we have $\phi(t) \in X$, with $\|\phi(t)\|_X \|_{H^r(0, T)} < \infty$.

• $H^{1/2}(\mathcal{B})$ and $H^{-1/2}(\mathcal{B})$ are the trace space on the artificial boundary, of $H^1(\Omega)$ functions, and the corresponding dual space, respectively.

Recalling the well known embedding property: $H^r(0, T) \subset C^m[0, T]$ for $r > m + 1/2$, from (7) and (8) we deduce that the assumption $r > 3/2$ guarantees that

$$\mathcal{V} : H_0^r(0, T; H^{-1/2}(\mathcal{B})) \rightarrow C([0, T]; H^{1/2}(\mathcal{B})).$$

Furthermore, if $r > 2$ we also have

$$\mathcal{K} : H_0^r(0, T; H^{1/2}(\mathcal{B})) \rightarrow C([0, T]; H^{1/2}(\mathcal{B})).$$

Here and in the following, $C^m(I; X)$ ($C = C^0$) denotes the space of C^m functions of $t \in I$, such that for each value of t the corresponding function of \mathbf{x} belongs to the space X .

Note that in our case we have $u \in H_0^r(0, T; H^r(\mathcal{B}))$, $\partial_{\mathbf{n}_{\mathcal{D}}} u \in H_0^r(0, T; H^{r-1}(\mathcal{B}))$, where r is at least 2. This is a straight consequence of the assumptions we have made on the problem data (see Remark 2.2). These assumptions also guarantee that all terms of our NRBC (6) are, in particular, continuous functions of $\mathbf{x} \in \mathcal{B}$. Thus, the NRBC it is well defined at any point of the boundary \mathcal{B} .

Remark 2.3. *The above NRBC (6) is mathematically non local both in space and time. However, because of the properties of the Lubich convolution quadrature, the computational cost due to this non locality can be cut down significantly. For example, when we choose as artificial boundary a circle, the true computational cost associated with our boundary condition is essentially of local type. Also the computational cost due to the time non locality is reduced significantly by using the FFT algorithm, to compute simultaneously all the needed quantities. From the computational complexity point of view, in this case the proposed NRBC turns out to be almost of local type.*

These issues will be discussed, with more details, in the examples we will present in Section 3.3.

Remark 2.4. *We have assumed that the local support of each datum is included either in Ω or in \mathcal{D} . In the first case we will obviously have that the corresponding volume integral in (6) vanishes, and the data will be treated by the numerical scheme one chooses to solve the problem in Ω . In the second case, because of the local support, the domain of integration of the volume integrals is not the whole infinite domain \mathcal{D} . Moreover, because of the presence of the Heaviside function in the expression defining the kernel G , each term will require the computation of volume integrals defined on the intersection of the corresponding local support with the disk of radius t centered at the point where the NRBC is collocated.*

The determination of this intersection is particularly simple when the data support is itself a circle, or the data decays smoothly to zero at the boundary of its support, and we trivially embed this latter into a circle, that we will then consider as the data “practical” support.

We finally observe that the computation of I_f is very simple when the source f is concentrated at a point, that is, is of the type $f(\mathbf{x}, t) = h(t)\delta(\mathbf{x} - \mathbf{x}_0)$ (see Section 3.3, Example 6, for details).

From now on, to simplify the description, we assume that the local supports of u_0 , v_0 and f are contained in Ω , so that $I_{u_0} = I_{v_0} = I_f = 0$. The treatment of source and initial data, whose supports are in \mathcal{D} , is postponed to Section 3.3.

Thus, denoting by $\partial_{\mathbf{n}} = \frac{\partial}{\partial \mathbf{n}}$ the outward unit normal derivative at the boundary \mathcal{B} , for the problem defined in the domain Ω , and noting that $\partial_{\mathbf{n}} = -\partial_{\mathbf{n}_D}$, the model problem (defined in the domain of interest Ω) takes the form:

$$\begin{cases} u_{tt}(\mathbf{x}, t) - \Delta u(\mathbf{x}, t) &= f(\mathbf{x}, t) & \text{in } \Omega \times (0, T) \\ u(\mathbf{x}, t) &= g(\mathbf{x}, t) & \text{in } \Gamma \times (0, T) \\ \frac{1}{2}u(\mathbf{x}, t) + \mathcal{V}\partial_{\mathbf{n}}u(\mathbf{x}, t) + \mathcal{K}u(\mathbf{x}, t) &= 0 & \text{in } \mathcal{B} \times (0, T) \\ u(\mathbf{x}, 0) &= u_0(\mathbf{x}) & \text{in } \Omega \\ u_t(\mathbf{x}, 0) &= v_0(\mathbf{x}) & \text{in } \Omega. \end{cases} \quad (9)$$

Remark 2.5. *Recalling some existence and uniqueness results on the solution of problem (1) in proper functional spaces (see [9]), it is not difficult to show that under the same assumptions also problem (9) has a unique solution in the corresponding spaces; this coincides with the restriction of the solution of the former problem in the Ω domain. Indeed, assuming for example that (1) has a unique solution in the space $C^1([0, T]; H^1(\Omega^e))$, this trivially satisfies also the reduced problem (9). Moreover, if by contradiction this latter problem has also another solution in $C^1([0, T]; H^1(\Omega))$, the difference between these two solutions defines a non trivial solution of the associated fully homogeneous version of problem (9). Knowing this solution, hence the corresponding Dirichlet and Neumann data on the artificial boundary \mathcal{B} , we can use the single-double layer potential representation to extend the latter solution into the infinite domain D , to define a non trivial $C^1([0, T]; H^1(\Omega^e))$ solution of the fully homogeneous version of problem (1). Being the solution of this unique, it must necessarily be the trivial one.*

2.1. Restriction of the model problem to the domain of interest

There are mainly three approaches that can be considered to couple our NRBC with the model problem. The first one is given by (9), where all terms are continuous functions of their variables and, therefore, equalities are pointwise defined. This is the so-called **strong formulation** of the problem.

The other two are weak type. To describe them, we introduce the spaces

$$X = \{u \in H^1(\Omega), u = g \text{ on } \Gamma\}$$

and

$$X_0 = \{u \in H^1(\Omega), u = 0 \text{ on } \Gamma\}.$$

We also introduce the additional unknown function $\lambda(\mathbf{x}, t) = \lambda(t)(\mathbf{x}) := \partial_{\mathbf{n}}u(\mathbf{x}, t)$, which is defined only on the boundary \mathcal{B} , in general by means of a trace operator (see [40], Sect. 1.3), and set $u(t)(\mathbf{x}) = u(\mathbf{x}, t)$. Then, the problem defined in the domain of interest Ω consists in finding the couple of unknown functions $u(t), \lambda(t)$ such that one of the following (alternative) formulations is satisfied. These are derived under smoothness requirements which are weaker than those we have previously made.

Variational formulation 1. By considering the weak form in space of the first and third equations of Problem (9), we obtain:

given $f \in L^2(\Omega \times (0, T))$, $u_0 \in X$, $v_0 \in L^2(\Omega)$, find $u(t) \in C^0([0, T]; X) \cap C^1([0, T]; L^2(\Omega))$ and $\lambda(t) \in C^0([0, T]; H^{-1/2}(\mathcal{B}))$ such that

$$\begin{cases} \frac{d^2}{dt^2}(u(t), w)_\Omega + a(u(t), w) - (\lambda(t), w)_\mathcal{B} &= (f(t), w), \quad \forall w \in X_0 \\ \frac{1}{2}(u(t), \varphi)_\mathcal{B} + ((\boldsymbol{\nu}\lambda)(t), \varphi)_\mathcal{B} + ((\boldsymbol{\kappa}u)(t), \varphi)_\mathcal{B} &= 0, \quad \forall \varphi \in H^{1/2}(\mathcal{B}) \\ u(0) &= u_0 \\ \frac{du}{dt}(0) &= v_0. \end{cases} \quad (10)$$

holds in the distributional sense in $(0, T)$, where $a : X \times X \rightarrow \mathbb{R}$ is the classical bilinear form

$$a(v, w) = \int_\Omega \nabla v \cdot \nabla w,$$

and $(v, w)_S = \int_S vw$ ($S = \Omega$ or \mathcal{B}).

Variational formulation 2. Here we consider the weak form only of the wave equation, i.e.,

$$\begin{cases} \frac{d^2}{dt^2}(u(t), w)_\Omega + a(u(t), w) - (\lambda(t), w)_\mathcal{B} &= (f(t), w), \quad \forall w \in X_0 \\ \frac{1}{2}u(\mathbf{x}, t) + \boldsymbol{\nu}\lambda(\mathbf{x}, t) + \boldsymbol{\kappa}u(\mathbf{x}, t) &= 0 \quad \text{on } \mathcal{B} \\ u(0) &= u_0 \\ \frac{du}{dt}(0) &= v_0. \end{cases} \quad (11)$$

for $t \in (0, T]$.

To apply this formulation, we must assume that all equalities are defined in the strong sense, and in particular that all their terms (including the volume terms that appear in the more general situation represented by (6)) are continuous functions of the corresponding variables. This property is certainly guaranteed by the data smoothness and compatibility assumptions we have made at the beginning of this section.

Formulation 1 is certainly of interest; however, its numerical solution using methods such as finite and boundary elements, probably makes it hardly competitive with existing NRBC approaches, in particular when the problem does not have far field sources. We recall that an analogous setting for an elliptic problem has been studied in [8].

From this point of view, the strong formulation and formulation 2 are certainly more appealing, although they require stronger smoothness conditions. Therefore, in the next sections we will consider only these two, and apply them to several test problems. To solve these, formulation 2 will naturally be associated with a (space) finite element method (see Sect. 3), while the space discretization of the strong formulation will be performed by means of finite differences (see Sect. 4).

3. Reduced problem discretization. The Finite Element Method

3.1. A Lubich-collocation method for the NRBC

In this paper we mainly consider smooth problems, i.e., problems defined on domains having a smooth boundary and smooth compatible data. However, as mentioned in [9], the

proposed BIE discretization seems to perform well also in cases where such assumptions are not all satisfied. To define this discretization we adopt the numerical approach which combine a second order (time) convolution quadrature formula of Lubich (see [33]) with a classical space collocation method. This because, in the following, the NRBC will be combined only with second order finite element and finite difference methods. We recall however that there exist also higher order Lubich convolution quadratures (see [37], [2]), that could be combined with higher order space discretizations.

We consider the integral relation (6) defined on a smooth boundary \mathcal{B} , given, for simplicity, by a (smooth) parametric representation. For the time discretization, we split the interval $[0, T]$ into N steps of equal length $\Delta_t = T/N$ and collocate the equation at the discrete time levels $t_n = n\Delta_t$, $n = 0, \dots, N$:

$$\frac{1}{2}u(\mathbf{x}, t_n) + (\mathbf{V}\lambda)(\mathbf{x}, t_n) + (\mathbf{K}u)(\mathbf{x}, t_n) = 0 \quad (12)$$

After having exchanged the order of integration, the time integrals appearing in the definition of the single and double layer operators are discretized by means of the Lubich convolution quadrature formula associated with the order 2 Backward Differentiation Method (BDF) for ordinary differential equations (see [9]). We obtain:

$$(\mathbf{V}\lambda)(\mathbf{x}, t_n) \approx \sum_{j=0}^n \int_{\mathcal{B}} \omega_{n-j}^{\mathbf{V}}(\Delta_t; \|\mathbf{x} - \mathbf{y}\|) \lambda(\mathbf{y}, t_j) d\mathcal{B}_{\mathbf{y}}, \quad n = 0, \dots, N \quad (13)$$

$$(\mathbf{K}u)(\mathbf{x}, t_n) \approx \sum_{j=0}^n \int_{\mathcal{B}} \omega_{n-j}^{\mathbf{K}}(\Delta_t; \|\mathbf{x} - \mathbf{y}\|) u(\mathbf{y}, t_j) d\mathcal{B}_{\mathbf{y}}, \quad n = 0, \dots, N \quad (14)$$

whose coefficients $\omega_n^{\mathcal{J}}$, $\mathcal{J} = \mathbf{V}, \mathbf{K}$, are given by

$$\omega_n^{\mathcal{J}}(\Delta_t; \|\mathbf{x} - \mathbf{y}\|) = \frac{1}{2\pi i} \int_{|z|=\rho} K^{\mathcal{J}} \left(\|\mathbf{x} - \mathbf{y}\|, \frac{\gamma(z)}{\Delta_t} \right) z^{-(n+1)} dz$$

where in this case $K^{\mathbf{V}} = \widehat{G}$ is the Laplace transform of the kernel G appearing in the definition of the single layer operator \mathbf{V} , and $K^{\mathbf{K}} = \widehat{\partial G / \partial \mathbf{n}}$ is the Laplace transform of the kernel $\partial G / \partial \mathbf{n}$ appearing in the definition of the double layer operator \mathbf{K} .

The function $\gamma(z) = 3/2 - 2z + 1/2z^2$ is the so called characteristic quotient of the BDF method of order 2. The parameter ρ is such that for $|z| \leq \rho$ the corresponding $\gamma(z)$ lies in the domain of analyticity of $K^{\mathcal{J}}$.

The Laplace transforms $K^{\mathcal{J}}$ can be computed by using some well known properties of the modified Bessel functions (see formulas 8.486, in particular 11., 16. and 17.) in [20]). In particular, we have that

$$\begin{aligned} K^{\mathbf{V}}(r, s) &= \frac{1}{2\pi} K_0(rs), \\ K^{\mathbf{K}}(r, s) &= -\frac{1}{2\pi} s K_1(rs) \frac{\partial r}{\partial \mathbf{n}}, \end{aligned} \quad (15)$$

where $K_0(z)$ and $K_1(z)$ are the second kind modified Bessel function of order 0 and 1, respectively.

By introducing the polar coordinate $z = \rho e^{i\varphi}$, the above integrals can be efficiently computed by the trapezoidal rule with $L \geq N$ equal steps of length $2\pi/L$:

$$\omega_n^{\mathcal{J}}(\Delta_t; r) \approx \frac{\rho^{-n}}{L} \sum_{l=0}^{L-1} K^{\mathcal{J}} \left(r, \frac{\gamma(\rho \exp(il2\pi/L))}{\Delta_t} \right) \exp(-inl2\pi/L). \quad (16)$$

We choose $L = 2N$ and $\rho^N = \sqrt{\varepsilon}$, since Lubich in ([33]) has shown that this choice leads to an approximation of ω_n with relative error of size $\sqrt{\varepsilon}$, if $K^{\mathcal{J}}$ is computed with a relative accuracy bounded by ε . The choice of ε suggested by Lubich is 10^{-10} . According to the previous statement, this should give a relative accuracy of order 10^{-5} , which is sufficient for the tests we have performed and that we will present in the examples that will follow. For each given $\mathbf{x} \in B$, all the $\omega_n^{\mathcal{J}}$ can be computed simultaneously by the FFT, with $O(N \log N)$ flops. Note that when we choose $L > N$, as in our case, the required $\omega_n^{\mathcal{J}}$ are given by the first N components of the coefficient vector determined by the FFT.

For the space discretization, first we introduce a parametrization of the curve \mathcal{B} , $\mathbf{x} = \boldsymbol{\psi}(x) = (\psi_1(x), \psi_2(x))$ and $\mathbf{y} = \boldsymbol{\psi}(y) = (\psi_1(y), \psi_2(y))$ with $x, y \in [a, b]$. Notice that this requirement is not a restriction. Indeed, since the contour \mathcal{B} can be arbitrarily chosen, we can always define a smooth parametric curve having the desired shape, by taking, for example, a (smooth) parametric cubic spline associated with a chosen set of points in \mathbb{R}^2 .

Then, at every time instant t_j we approximate the (unknown) function $u(\boldsymbol{\psi}(x), t_j)$ and its normal derivative $\lambda(\boldsymbol{\psi}(x), t_j)$ by continuous piecewise linear interpolants, associated with a uniform partition $\{x_k\}_{k=1}^{M+1}$ of the parametrization interval $[a, b]$. These are written in the form

$$\begin{aligned} u(\boldsymbol{\psi}(x), t_j) &\approx u_{\Delta}^{\psi}(x, t_j) := \sum_{i=1}^{M+1} u_i^j N_i(x), & u_i^j &= u(\boldsymbol{\psi}(x_i), t_j) \\ \lambda(\boldsymbol{\psi}(x), t_j) &\approx \lambda_{\Delta}^{\psi}(x, t_j) := \sum_{i=1}^{M+1} \lambda_i^j N_i(x), & \lambda_i^j &= \lambda(\boldsymbol{\psi}(x_i), t_j) \end{aligned} \quad (17)$$

where $\{N_i(x)\}$ are the classical Lagrangian basis functions of local degree 1. These in turns define the associated interpolants of $u(\mathbf{x}, t_j)$ and $\lambda(\mathbf{x}, t_j)$ on the curve \mathcal{B} :

$$u(\mathbf{x}, t_j) \approx u_{\Delta}(\mathbf{x}, t_j) := \sum_{i=1}^{M+1} u_i^j N_i^{\mathcal{B}}(\mathbf{x}), \quad u_i^j = u(\boldsymbol{\psi}(x_i), t_j) \quad (18)$$

$$\lambda(\mathbf{x}, t_j) \approx \lambda_{\Delta}(\mathbf{x}, t_j) := \sum_{i=1}^{M+1} \lambda_i^j N_i^{\mathcal{B}}(\mathbf{x}), \quad \lambda_i^j = \lambda(\boldsymbol{\psi}(x_i), t_j) \quad (19)$$

with $N_i^{\mathcal{B}}(\boldsymbol{\psi}(x)) = N_i(x)$.

Since the role of the NRBC is to define on \mathcal{B} a relationship between the (outgoing/incoming) wave and its normal derivative, which prevents the raising of (spurious) incoming/outgoing waves, the more accurate is the discretized relationship the more transparent this will be. To this end, having chosen a continuous piecewise linear (space) approximant for $u(\boldsymbol{\psi}(x), t_j)$, we use an approximant of the same type also for $\lambda(\boldsymbol{\psi}(x), t_j)$.

Taking into account that the curve \mathcal{B} is closed, we set $u_1^j = u_{M+1}^j$ and $\lambda_1^j = \lambda_{M+1}^j$. Finally, by collocating the fully discretized equation at the points $\xi_h = x_h$, $h = 1, \dots, M$, we obtain (see [9]) the following linear system:

$$\left(\frac{1}{2}\mathbf{I} + \mathbf{K}_0\right) \mathbf{u}_{\mathcal{B}}^n + \sum_{j=0}^{n-1} \mathbf{K}_{n-j} \mathbf{u}_{\mathcal{B}}^j + \mathbf{V}_0 \boldsymbol{\lambda}_{\mathcal{B}}^n + \sum_{j=0}^{n-1} \mathbf{V}_{n-j} \boldsymbol{\lambda}_{\mathcal{B}}^j = 0, \quad n = 1, \dots, N \quad (20)$$

in the unknown vectors $\mathbf{u}_{\mathcal{B}}^j = (u_1^j, \dots, u_M^j)$ and $\boldsymbol{\lambda}_{\mathcal{B}}^j = (\lambda_1^j, \dots, \lambda_M^j)$, $j = 0, \dots, n$. Note that now u_k^j and λ_k^j denote the approximate values, defined by (20), of $u(\boldsymbol{\psi}(x_k), t_j)$ and $\lambda(\boldsymbol{\psi}(x_k), t_j)$, respectively.

The symbol \mathbf{I} denotes the identity matrix of order M , while the matrices \mathbf{V} and \mathbf{K} are given by

$$(\mathbf{V}_{n-j})_{hi} = \int_a^b \omega_{n-j}^{\mathbf{V}}(\Delta_t; \|\boldsymbol{\psi}(x_h) - \boldsymbol{\psi}(y)\|) N_i(y) \|\boldsymbol{\psi}'(y)\| dy, \quad (21)$$

and

$$(\mathbf{K}_{n-j})_{hi} = \int_a^b \omega_{n-j}^{\mathbf{K}}(\Delta_t; \|\boldsymbol{\psi}(x_h) - \boldsymbol{\psi}(y)\|) N_i(y) \|\boldsymbol{\psi}'(y)\| dy. \quad (22)$$

From the computational point of view, supposing to know $\mathbf{u}_{\mathcal{B}}^j$ and $\boldsymbol{\lambda}_{\mathcal{B}}^j$ at the time steps $j = 0, \dots, n-1$, the absorbing condition at time t_n is given by

$$\left(\frac{1}{2}\mathbf{I} + \mathbf{K}_0\right) \mathbf{u}_{\mathcal{B}}^n + \mathbf{V}_0 \boldsymbol{\lambda}_{\mathcal{B}}^n = - \sum_{j=0}^{n-1} \mathbf{K}_{n-j} \mathbf{u}_{\mathcal{B}}^j - \sum_{j=0}^{n-1} \mathbf{V}_{n-j} \boldsymbol{\lambda}_{\mathcal{B}}^j, \quad n = 1, \dots, N. \quad (23)$$

Remark 3.1. As described in [9], for each row index, the corresponding row elements of all the above matrices can be computed simultaneously by means of the FFT algorithm, after replacing, in the representations (21), (22), the ω kernel by its discretization (16), and exchanging the integration symbol with that of the quadrature sum. For the evaluation of the integrals see [9].

For the truncation errors associated with the discrete operators

$$(\mathbf{V}_{\Delta} \lambda)(\mathbf{x}, t_n) := \sum_{j=0}^n \int_{\mathcal{B}} \omega_{n-j}^{\mathbf{V}}(\Delta_t; \|\mathbf{x} - \mathbf{y}\|) \lambda_{\Delta}(\mathbf{y}, t_j) d\mathcal{B}_{\mathbf{y}} \quad (24)$$

$$(\mathbf{K}_{\Delta} u)(\mathbf{x}, t_n) := \sum_{j=0}^n \int_{\mathcal{B}} \omega_{n-j}^{\mathbf{K}}(\Delta_t; \|\mathbf{x} - \mathbf{y}\|) u_{\Delta}(\mathbf{y}, t_j) d\mathcal{B}_{\mathbf{y}} \quad (25)$$

we have constructed, we have the estimates reported in the following proposition.

Proposition 3.2. Under the assumption $u \in C^4([0, T]; H^r(\mathcal{D}))$, $r > 7/2$, we have

$$\max_{0 \leq n \leq N} \|(\mathbf{V} - \mathbf{V}_{\Delta}) \lambda(\cdot, t_n)\|_{L^2(\mathcal{B})} = O(\Delta_t^2) + O(\Delta_x^2)$$

and

$$\max_{0 \leq n \leq N} \|(\mathbf{K} - \mathbf{K}_{\Delta}) u(\cdot, t_n)\|_{L^2(\mathcal{B})} = O(\Delta_t^2) + O(\Delta_x^{3/2})$$

Proof. We describe explicitly the main steps one has to perform to obtain the first estimate, since the case of the \mathcal{K}_Δ operator is very similar. Thus, we rewrite the discretization error as follows:

$$(\mathbf{v} - \mathbf{v}_\Delta)\lambda = (\mathbf{v}\lambda - \mathbf{v}\lambda_\Delta) + (\mathbf{v}\lambda_\Delta - \mathbf{v}_\Delta\lambda) =: R_1^\mathbf{v} + R_2^\mathbf{v}$$

Taking into account the mapping properties of the operators \mathbf{v}, \mathcal{K} (see (7) and (8)), we obtain the bound

$$\|R_1^\mathbf{v}\|_{H^{1/2}(\mathcal{B})} \leq C\|\lambda - \lambda_\Delta\|_{H^{-1/2}(\mathcal{B})} \leq C\|\lambda - \lambda_\Delta\|_{L^2(\mathcal{B})} \leq C_1\Delta_x^2 \quad (26)$$

where the constant C_1 depends only on T . The last bound has been obtained after introducing the curve parametrization $\psi(y)$.

In the case of (25), we have:

$$\|R_1^\mathcal{K}\|_{H^{1/2}(\mathcal{B})} \leq C\|u - u_\Delta\|_{H^{1/2}(\mathcal{B})} \quad (27)$$

A bound for the latter approximation error is obtained by applying the well known interpolation inequality $\|\cdot\|_{H^{1/2}} \leq \sqrt{\|\cdot\|_{H^0}\|\cdot\|_{H^1}}$, since for the H^0 and H^1 error norms we have the bounds $O(\Delta_x^2)$ and $O(\Delta_x^1)$, respectively. This gives:

$$\|R_1^\mathcal{K}\|_{H^{1/2}(\mathcal{B})} \leq C_2\Delta_x^{3/2}$$

from which, a fortiori, the (rough) bound $\|R_1^\mathcal{K}\|_{L^2(\mathcal{B})} \leq C_2\Delta_x^{3/2}$ follows. It seems however reasonable to expect a more accurate bound of the type $O(\Delta_x^2)$.

To obtain a bound for $R_2^\mathbf{v}$, we rewrite this latter in the form:

$$R_2^\mathbf{v}(\mathbf{x}, t_n) = \int_{\mathcal{B}} \sum_{i=1}^{M+1} E_i^\mathbf{v}(\mathbf{x}, \mathbf{y}, t_n) N_i^\mathcal{B}(\mathbf{y}) d\mathcal{B}_\mathbf{y}$$

having set

$$E_i^\mathbf{v}(\mathbf{x}, \mathbf{y}, t_n) = \int_0^{t_n} k(\|\mathbf{x} - \mathbf{y}\|; t_n - \tau) \lambda(\mathbf{y}_i, \tau) d\tau - \sum_{j=0}^n \omega_{n-j}^\mathbf{v}(\Delta_t; \|\mathbf{x} - \mathbf{y}\|) \lambda(\mathbf{y}_i, t_j),$$

where $\mathbf{y}_i = \psi(y_i)$. Recalling the error estimates derived in [34], Theorem 3.1, under the smoothness assumption we have made, we immediately obtain the (uniform) bound

$$|E_i^\mathbf{v}(\mathbf{x}, \mathbf{y}, t_n)| \leq \frac{C_T^\mathcal{B}}{\|\mathbf{x} - \mathbf{y}\|^\epsilon} \Delta_t^2,$$

where $\epsilon > 0$ can be taken arbitrarily small and the constant $C_T^\mathcal{B}$ does not depend on $i, \mathbf{x}, \mathbf{y}$ and n . Since $N_i^\mathcal{B}(\mathbf{y}) = N_i(y) \geq 0$ and $\sum_{i=1}^{M+1} N_i(y) = 1$, we finally have

$$\|R_2^\mathbf{v}\|_{L^2(\mathcal{B})} \leq C\Delta_t^2. \quad \square$$

3.2. The complete discrete scheme

In order to derive the complete numerical method we propose to solve (11), we start by describing the time discretization of its first equation. In general, we choose to perform this latter by using the Crank-Nicolson scheme, of second order and unconditionally stable, which is well suited even for long time intervals, although other methods can be considered as well, in particular explicit ones. We choose an unconditionally stable time integrator mainly because we want to check if the global numerical scheme maintains the same (unconditional) stability properties of the classical coupling Crank-Nicolson-FE. We will however make some comments in the case of a time integrator of explicit type.

Thus, we introduce the variable $v := \frac{\partial u}{\partial t}$, set $b(\lambda(t), w) := (\lambda(t), w)_{\mathcal{B}}$, hence rewrite (11) as follows:

$$\begin{cases} \frac{d}{dt}(v(t), w) + a(u(t), w) - b(\lambda(t), w) &= (f(t), w), \quad \forall w \in X_0 \\ \frac{\partial u}{\partial t}(\mathbf{x}, t) &= v(\mathbf{x}, t), \quad \forall \mathbf{x} \in \Omega \\ \frac{1}{2}u(\mathbf{x}, t) + \mathbf{V}\lambda(\mathbf{x}, t) + \mathbf{K}u(\mathbf{x}, t) &= 0 \quad \text{in } \mathcal{B} \times (0, T] \\ u(\mathbf{x}, 0) &= u_0(\mathbf{x}), \quad \forall \mathbf{x} \in \Omega \\ v(\mathbf{x}, 0) &= v_0(\mathbf{x}), \quad \forall \mathbf{x} \in \Omega. \end{cases} \quad (28)$$

for all $t \in (0, T]$. Denoting by $u^n = u^n(x)$, $v^n = v^n(x)$, $\lambda^n = \lambda^n(x)$ and $f^n = f^n(x)$ the approximations of $u(x, t_n)$, $v(x, t_n)$, $\lambda(x, t_n)$ and $f(x, t_n)$, respectively, and applying the Crank-Nicolson discretization to the first two equations in (28), we have

$$\begin{aligned} \left(\frac{v^{n+1} - v^n}{\Delta_t}, w \right) + a\left(\frac{u^{n+1} + u^n}{2}, w \right) - b\left(\frac{\lambda^{n+1} + \lambda^n}{2}, w \right) &= \left(\frac{f^{n+1} + f^n}{2}, w \right), \quad \forall w \in X_0 \\ \frac{v^{n+1} + v^n}{2} &= \frac{u^{n+1} - u^n}{\Delta_t}, \quad \forall \mathbf{x} \in \Omega. \end{aligned}$$

From the second relation we get:

$$v^{n+1} = \frac{2}{\Delta_t}(u^{n+1} - u^n) - v^n \quad (29)$$

which, inserted in the first relation, leads to

$$\begin{aligned} (u^{n+1}, w) + \frac{\Delta_t^2}{4}a(u^{n+1}, w) - \frac{\Delta_t^2}{4}b(\lambda^{n+1}, w) &= (u^n, w) - \frac{\Delta_t^2}{4}a(u^n, w) + \frac{\Delta_t^2}{4}b(\lambda^n, w) \\ &\quad + \Delta_t(v^n, w) + \frac{\Delta_t^2}{4}(f^{n+1} + f^n, w), \quad \forall w \in X_0 \end{aligned}$$

For the space finite element discretization, we define a regular triangular mesh $\mathcal{T}_h = \{K_i\}$ on Ω , with mesh size bounded by h . This defines a polygonal domain Ω_Δ , having inner and outer boundaries Γ_Δ and \mathcal{B}_Δ , respectively. Below, we denote $f|_{\Gamma_\Delta}$ the restriction of the function f on Γ_Δ . For computational simplicity, in the first equation in (28) we replace Ω by Ω_Δ and \mathcal{B} by \mathcal{B}_Δ . In our NRBC we leave the original boundary \mathcal{B} , to use its parametric representation. Of course, also in this equation we could replace \mathcal{B} by \mathcal{B}_Δ ; however the first choice has given slightly better results. Note that, in spite of this boundary discrepancy, the

final discrete system will involve only the unknown values at the common boundary mesh points. The curve \mathcal{B}_Δ is nothing but a piecewise linear interpolant of \mathcal{B} .

Let

$$X_h = \{w_h \in C^0(\Omega) : w_{h|_{K_i}} \in \mathbb{P}^1(K_i), K_i \in \mathcal{T}_h, w_{h|_{\Gamma_\Delta}} = g|_{\Gamma_\Delta}\} \subset H^1,$$

$$X_{h,0} = \{w_h \in C^0(\Omega) : w_{h|_{K_i}} \in \mathbb{P}^1(K_i), K_i \in \mathcal{T}_h, w_{h|_{\Gamma_\Delta}} = 0\} \subset H_0^1$$

be the spaces of (piecewise) linear conforming finite elements in the domain Ω associated with the mesh \mathcal{T}_h . Let W_h be the space of (continuous) functions defined on the boundary \mathcal{B} by the finite element basis $\{N_i(x)\}$ (see (18), (19)). The Galerkin formulation of the above equation then reads: for each $n = 0, \dots, N-1$, find $(u_h^{n+1}, \lambda_h^{n+1}) \in X_h \times W_h$ such that, for all $w_h \in X_{h,0}$

$$\begin{aligned} (u_h^{n+1}, w_h) + \frac{\Delta_t^2}{4} a(u_h^{n+1}, w_h) - \frac{\Delta_t^2}{4} b(\lambda_h^{n+1}, w_h) &= (u_h^n, w_h) - \frac{\Delta_t^2}{4} a(u_h^n, w_h) \\ &\quad + \frac{\Delta_t^2}{4} b(\lambda_h^n, w_h) + \Delta_t (v_h^n, w_h) + \frac{\Delta_t^2}{4} (f^{n+1} + f^n, w_h) \end{aligned} \quad (30)$$

Let $\{N_i^\Omega\}_{i \in \mathcal{S}}$ denote the set of finite element basis functions defined on the triangulation \mathcal{T}_h , having set $\mathcal{S} = \mathcal{S}_I \cup \mathcal{S}_B$, where \mathcal{S}_I is the set of the internal mesh nodes and \mathcal{S}_B is the set of the mesh nodes lying on the artificial boundary \mathcal{B} . By properly reordering the unknown coefficients of u_h^n , we obtain the (unknown) vector $\mathbf{u}^n = [\mathbf{u}_I^n, \mathbf{u}_B^n]^T$, whose two components \mathbf{u}_I^n and \mathbf{u}_B^n represent the unknown values associated with the internal nodes and with those on the boundary \mathcal{B} , respectively. Similarly for the vector \mathbf{v}^n , containing the unknown coefficients of v_h^n . Finally, we denote by $\boldsymbol{\lambda}^n$ the unknown vector whose components are the coefficients of the approximant $\lambda_\Delta(\mathbf{x}, t_n)$ defined in (19). Therefore, the matrix form of (30) is given by

$$\begin{aligned} \left(\mathbf{M} + \frac{\Delta_t^2}{4} \mathbf{A}\right) \mathbf{u}^{n+1} - \frac{\Delta_t^2}{4} \mathbf{Q} \boldsymbol{\lambda}^{n+1} &= \left(\mathbf{M} - \frac{\Delta_t^2}{4} \mathbf{A}\right) \mathbf{u}^n + \frac{\Delta_t^2}{4} \mathbf{Q} \boldsymbol{\lambda}^n + \Delta_t \mathbf{M} \mathbf{v}^n \\ &\quad + \frac{\Delta_t^2}{4} (\mathbf{f}^{n+1} + \mathbf{f}^n) \end{aligned} \quad (31)$$

where

$$\mathbf{M} = \begin{bmatrix} M_{II} & M_{IB} \\ M_{BI} & M_{BB} \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} A_{II} & A_{IB} \\ A_{BI} & A_{BB} \end{bmatrix}, \quad \mathbf{Q} = \begin{bmatrix} Q_{IB} \\ Q_{BB} \end{bmatrix}.$$

The matrix elements

$$(\mathbf{M})_{ij} = \int_{\Omega} N_i^\Omega N_j^\Omega, \quad (\mathbf{A})_{ij} = \int_{\Omega} \nabla N_i^\Omega \cdot \nabla N_j^\Omega, \quad i, j \in \mathcal{S}$$

are those of the mass and stiffness matrices, respectively, while those of \mathbf{Q} are given by

$$(\mathbf{Q})_{ij} = \int_{\mathcal{B}} N_i^B N_j^B, \quad i \in \mathcal{S}, j \in \mathcal{S}^B.$$

Equation (31) is finally coupled with (29), that is,

$$\mathbf{v}^{n+1} = \frac{2}{\Delta_t}(\mathbf{u}^{n+1} - \mathbf{u}^n) - \mathbf{v}^n \quad (32)$$

and with the discretized NRBC equation

$$\left(\frac{1}{2}\mathbf{I} + \mathbf{K}_0\right) \mathbf{u}_{\mathcal{B}}^{n+1} + \mathbf{V}_0 \boldsymbol{\lambda}^{n+1} = - \sum_{j=0}^n \mathbf{K}_{n+1-j} \mathbf{u}_{\mathcal{B}}^j - \sum_{j=0}^n \mathbf{V}_{n+1-j} \boldsymbol{\lambda}^j. \quad (33)$$

Remark 3.3. When the boundary \mathcal{B} is that of a circle, and the basis $\{N_i\}$ is associated with a uniform partition of it, then all the matrices \mathbf{V}_j and \mathbf{K}_j , $j = 0, \dots, N$ have a Toeplitz structure. Therefore we only need to construct and store the first row of each matrix, that is, a total of $2M(N+1)$ elements. This property reduces significantly the computational cost of this ABC, as confirmed by the examples we have examined.

When we have to deal with a general boundary, or with a nonuniform partition, then the above property does not hold and we have to compute and store $2(N+1)$ full matrices of order M . Nevertheless, because of the behavior of the coefficients $\omega_j^{\mathcal{J}}$ in the Lubich convolution quadrature formula (see, for example, Figures 3 and 11 in [37]), the matrices \mathbf{V}_j and \mathbf{K}_j , $j = 0, \dots, N$, can be approximated by corresponding very sparse ones (see Section 3.4 below). This phenomenon is even more relevant in the 3D case (see for example [25]), where the corresponding Lubich coefficients $\omega_n(\Delta_t; r)$ represent a smooth approximation of delta Dirac functions (see [37], Figures 4 and 12). Thus, also in the case of a boundary \mathcal{B} with no special properties, the computational cost of our NRBC can be drastically cut down. Of course, also in the Toeplitz case described above, the representative rows can be replaced by their sparse version, thus reducing further the computational cost and the storage.

Unfortunately, for the above numerical scheme, till now we have not been able to derive stability and convergence results. This appears to be a challenging and very hard task. Because of this lack of theoretical results, we have performed an intensive numerical testing on these properties. In the next section we present a sample of the results we have obtained, to show the very good absorbing behavior of the proposed numerical scheme.

3.3. Numerical results

In this section, we present some examples of the numerical testing we have performed by using the approach discussed in the previous section. To measure the accuracy of the approximations we construct, we take a reference “exact” solution obtained by applying the Lubich-collocation boundary element method described in [9] with a very fine discretization. Once the density function is retrieved, the solution at any point in the infinite domain Ω^e is defined by computing the associated potential (see [9] for details). In the following, this solution will be denoted by the acronym BEM. To confirm its accuracy, we have first taken a sufficiently large artificial boundary and set on it the trivial Dirichlet condition, hence applied on the new computational domain the Crank-Nicolson/FE method, with sufficiently fine time step and triangulation. The agreement of the solutions produced by these two approaches has been verified in all the examples we present next. We have however preferred

the former, because, to obtain a sufficiently high relative accuracy ($\approx 1E - 6$), needed to estimate the accuracy of the approximants, the FE method requires an excessively fine space triangulation.

Since in the first three examples the chosen artificial boundary \mathcal{B} is a circle, we compare our solution with the classical (and cheap) first order Engquist-Majda NRBC (see [7]) and with the Sommerfeld NRBC (see [12]). In these examples, we have also applied the second order Engquist-Majda and Bayliss-Turkel methods. However, because of the discrepancy between the boundary \mathcal{B}_Δ of the FE computational domain and that (\mathcal{B}) of the NRBC, and of the presence of the tangential derivative, the solutions we have obtained are not satisfactory (they are very much underestimated). Note that while on \mathcal{B} this derivative is smooth, on \mathcal{B}_Δ it is only piecewise continuous. Indeed, when we rewrite the original PDE problem in polar coordinates, and apply the finite-difference method for the space discretization, the results produced by the second order Engquist-Majda and Bayliss-Turkel methods have the expected behaviors (see Section 4).

In the last three examples we compare the solution we obtain using our NRBC with the “exact” (BEM) solution defined above.

Example 1. As a first example, we apply our numerical scheme to the homogeneous case of Problem (1): the source f and the initial data u_0 and v_0 are zero throughout the infinite exterior domain Ω^e . The boundary Γ is the circle of radius $r = 0.25$, where we prescribe the Dirichlet condition $g(\mathbf{x}, t) = 1$ for all $t \geq 0$. We choose a circular artificial boundary with radius $R = 0.5$, so that Ω is the annulus bounded internally by Γ and externally by \mathcal{B} . Clearly, the solution of this problem is a radial function.

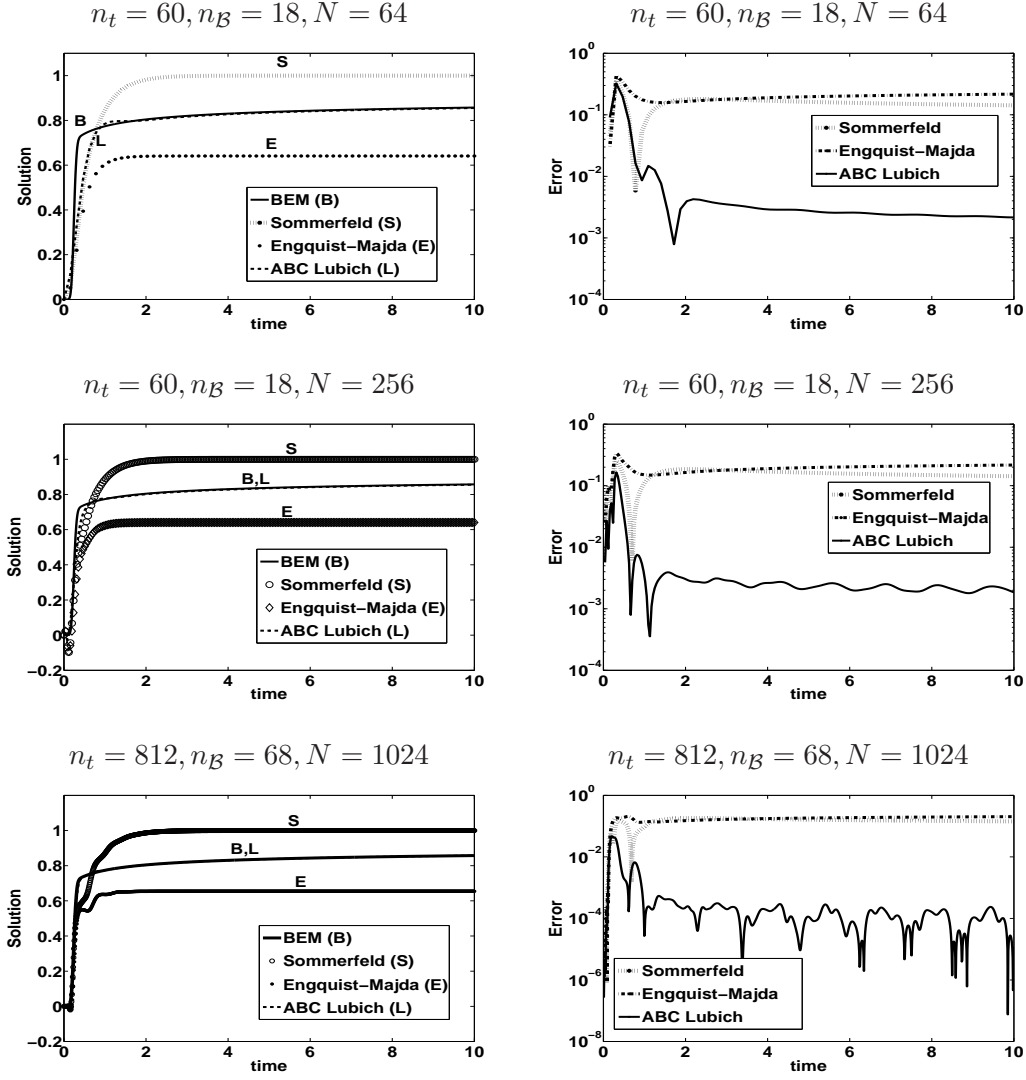
Note that the Dirichlet datum does not satisfy the required compatibility condition. Nevertheless, since this is a typical test which has been considered in several papers on the topic (see [12]), to check the performance of our approach, we have applied it to this problem.

Since our goal is to propose a NRBC for an artificial boundary having a general shape, in the very simple case we consider in this example, as well as in some of the following ones, when we apply the FE method we do not transform the Ω domain by introducing the polar coordinates. Thus we perform an (approximate) domain triangulation in the cartesian coordinates. This means that the Ω domain is itself approximated by an inscribed polygon. Instead, for the discretization of our NRBC we use the parametric representation given for the boundary \mathcal{B} . The boundary mesh is defined by the boundary points of the above domain triangulation, which in our case turns out to be (slightly) non uniform.

For the space discretization we choose an unstructured triangular mesh of n_t triangles, having $n_{\mathcal{B}}$ (not equally spaced) points on the boundary \mathcal{B} . Since on Γ we have $u(\mathbf{x}, 0) = 0$, while $g(\mathbf{x}, 0^+) = 1$, for the time integration we use a first order method. Higher order methods would generate unpleasant oscillations, caused by the mentioned initial datum jump. Thus, we have used the (first order) implicit Euler scheme, which is unconditionally stable.

In Figure 1 we have plotted the behavior of the approximate solution we have obtained at a point belonging to the boundary \mathcal{B} , $t \in [0, 10]$, for different choices of the time steps. It is compared with the exact reference solution, the first order Engquist-Majda and the Sommerfeld NRBC. It can be noticed that our NRBC converges to the exact solution as the time step decreases, while the Engquist-Majda and the Sommerfeld NRBC, as it is well known, give rise to approximations which underestimate and overestimate it, respectively.

Figure 1: Example 1. Solution obtained with the implicit Euler scheme (left column), and corresponding errors (right column).



In the next figures, whenever the exact NRBC (that will be denoted by “ABC Lubich”) perfectly matches the exact reference solution, we use the same graphical sign for the two curves in order to avoid misreading. There and in the following figures, $n_{\mathcal{B}}$ denotes the number of points on \mathcal{B} .

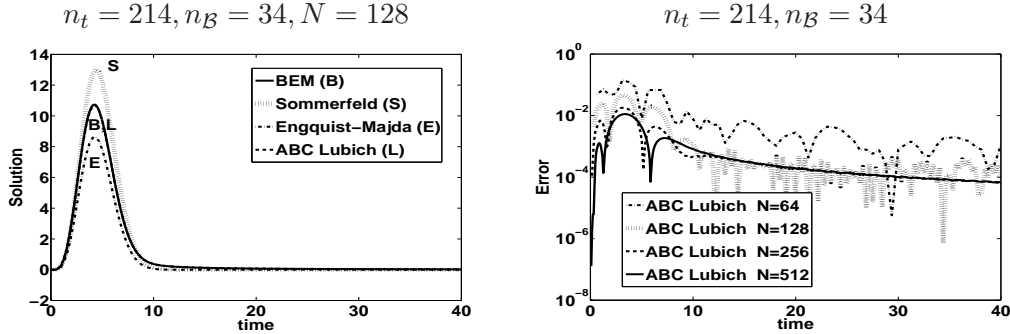
If we replace the implicit Euler method by the explicit one, then, to obtain similar results we have to take smaller step-sizes. The required reduction is apparently higher in the case of our NRBC.

We finally remark that in the particular case of a domain Ω with circular boundaries Γ and \mathcal{B} , as in this example, by introducing the polar coordinates we can always reduce the PDE problem to an equivalent one defined on a rectangle. In such a case, we can construct an exact triangulation of this latter domain, having equidistant nodes on the side which identifies the boundary \mathcal{B} . These give then rise to Toeplitz matrices $\mathbf{V}_j, \mathbf{K}_j$.

Example 2. In the same setting of Example 1, we choose $g(\mathbf{x}, t) = t^3 e^{-0.05(x_1^2 + x_2^2 - \sqrt{2}t)^2}$. With this choice, the data compatibility conditions are satisfied and the Crank-Nicolson scheme is applied. Notice that also in this case the solution is a radial function.

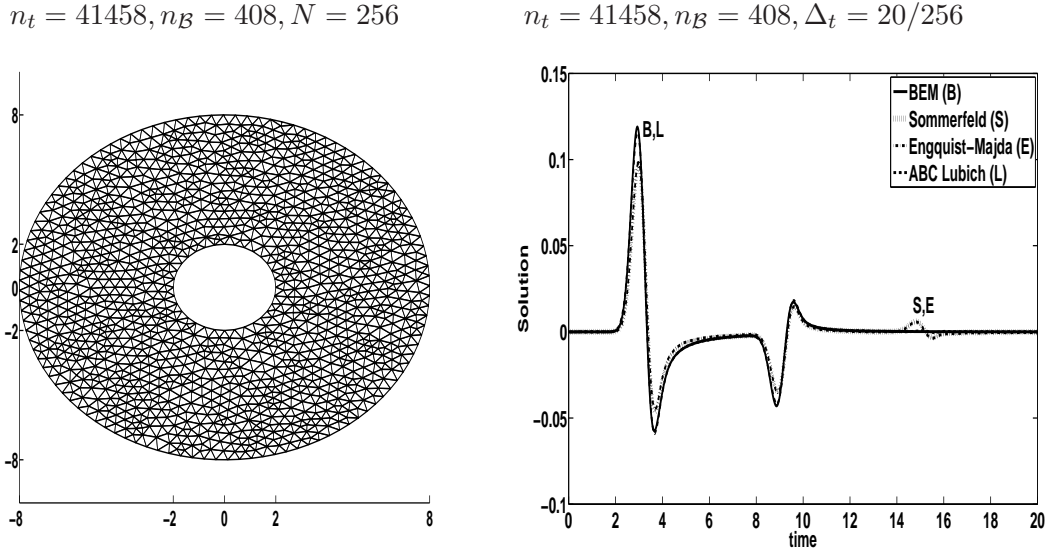
In Figure 2 we plot the behavior of the solution at a mesh point $\mathbf{x} \in \mathcal{B}$, in the time interval $[0, 40]$. In particular, in the left plot we compare the approximants produced by the different NRBC we have considered, taking $n_t = 214$ triangles in Ω and $N = 128$, with the exact solution. On the right plot we show the behavior of the associated error, for some choices of the time discretization step.

Figure 2: Example 2. Solution and corresponding errors.



Example 3. In this example we consider a non homogeneous problem, having zero source and zero initial velocity, but $u_0 \neq 0$, and $g = 0$ on Γ . The initial data is $u_0(x_1, x_2) = e^{-5((x_1-5)^2 + x_2^2)}$. Although u_0 does not have a local support (and thus contradicts one of our assumptions), it decays exponentially fast away from its center $\mathbf{x} = (5, 0)$, in such a way that from the computational point of view it can be regarded as compact and supported in a disk with radius smaller than 3 (at distance 2.7 from its center it assumes approximately values of the order 10^{-16}). The boundary Γ is a circle of radius $r_0 = 2$, and the artificial boundary is a circle of radius $R = 8$, so that the support of u_0 is included in Ω (see Figure 3, left plot). The disk bounded by Γ represents a soft obstacle that acts as a reflecting body.

Figure 3: Example 3. Domain triangulation and solution at $\mathbf{x} \cong (7.93, -1.08)$.



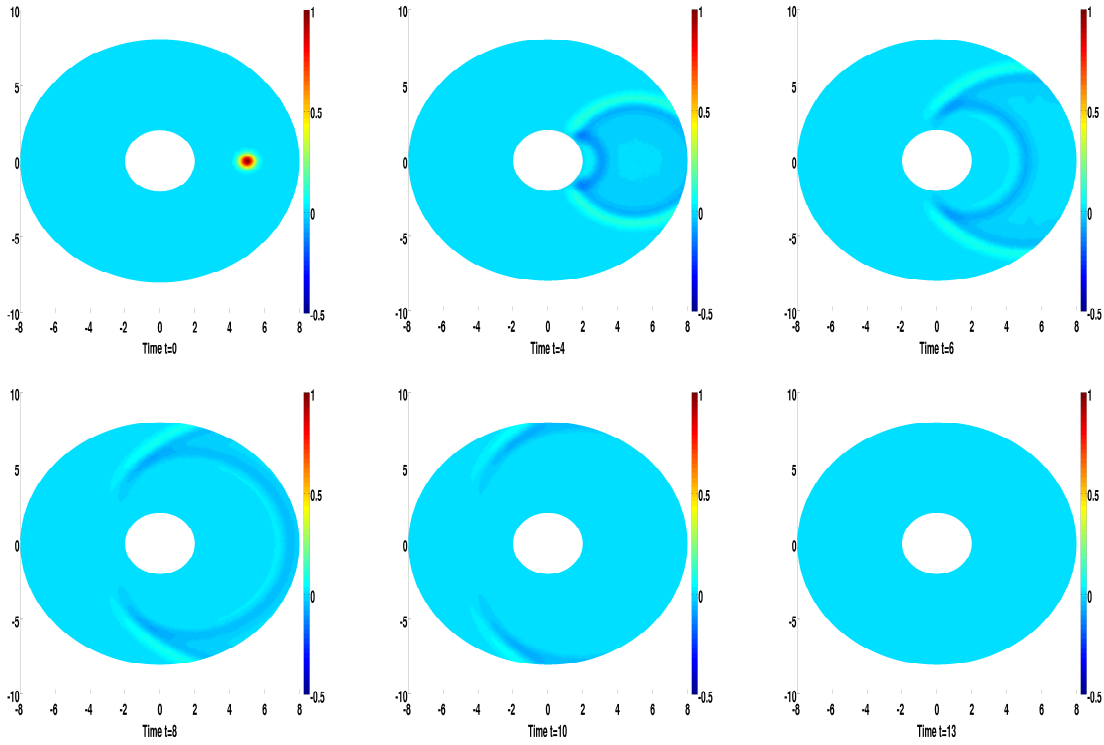
With these choices, the data compatibility conditions are satisfied and the Crank-Nicolson scheme can be applied.

In Figure 3, right plot, we show the behavior of the solution at the boundary mesh point $\mathbf{x} \cong (7.93, -1.08)$ and for $t \in [0, 20]$. We note that the solution is zero until the initial data reaches the artificial boundary around $t = 2$. Approximately at the same time, u_0 also reaches the reflecting boundary Γ and is reflected, so that around $t = 9$ we see another outgoing wave at the artificial boundary \mathcal{B} . After that time, the wave is completely out of the annulus, as the exact solution and the approximate solution with the exact NRBC show, while the other approximate NRBC show spurious waves at $t = 14$. This phenomenon is due to the fact that at $t = 2$ the artificial boundary is not completely transparent for the Engquist-Majda and the Sommerfeld NRBC. Indeed these have generated incoming waves that, after reaching Γ again, have been reflected back. In Figure 4 we show some snapshots of the solution obtained with our exact NRBC at different time steps.

Later, in Section 4, we will solve this same problem by replacing the FE method by the finite difference one, after having introduced the polar coordinates. There, we will also replace the Sommerfeld and the order 1 Engquist-Majda ABC by the second order Engquist-Majda and Bayliss-Turkel ABC. With the space grid and time stepsize chosen there, the approximate solutions generated by these two ABC and by our NRBC overlap with the reference one (see Fig. 19).

Example 4. For some geometries of the physical domain boundary Γ , or of the domain of interest Ω , the choice of a circular artificial boundary \mathcal{B} can be wasteful both from the computational and space memory point of view. In this example, we choose Γ as the ellipse centered at the origin having equation $x_1^2/a^2 + x_2^2/b^2 = 1$, with $a = 2$ and $b = 0.5$. A natural choice of \mathcal{B} could be, for example, the ellipse $x_1^2/a_1^2 + x_2^2/b_1^2 = 1$, where for example we choose $a_1 = 4$ and $b_1 = 1$. We consider the homogeneous problem, with Dirichlet data

Figure 4: Example 3. Snapshots of the solution at different times.



$g(\mathbf{x}, t) = t^3 e^{-t}$. In Figure 5 we compare the solution obtained with the exact NRBC and the exact solution at a mesh point $\mathbf{x} \approx (4, 0)$, for $t \in [0, 40]$, taking $n_t = 784$ triangles and $N = 256$. In Figure 6 we show the snapshots of the solution at different times.

Figure 5: Example 4. Left figure: $n_t = 784$ triangles for the mesh of Ω and the mesh point $\mathbf{x} \approx (4, 0) \in \mathcal{B}$ where the solution is evaluated (bullet); right figure: the behaviors of the exact and approximate solutions at \mathbf{x} .

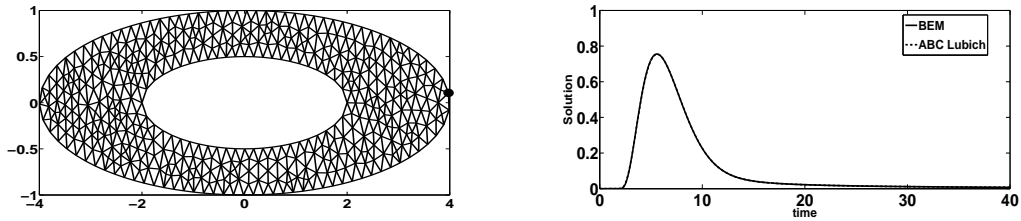


Figure 6: Example 4. Snapshots of the solution at different times.

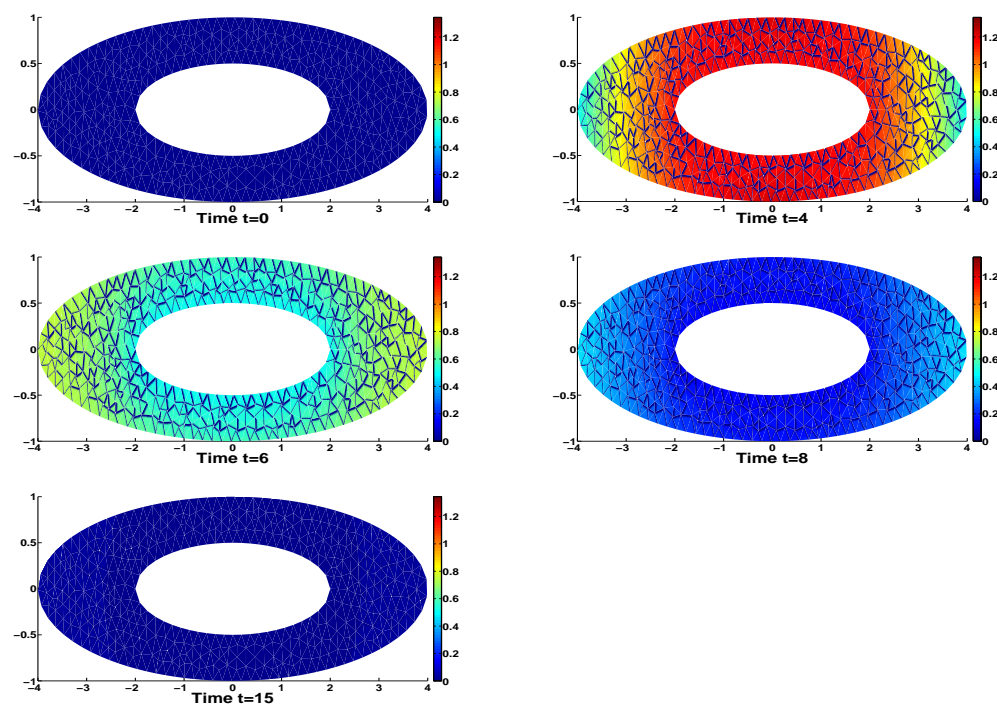
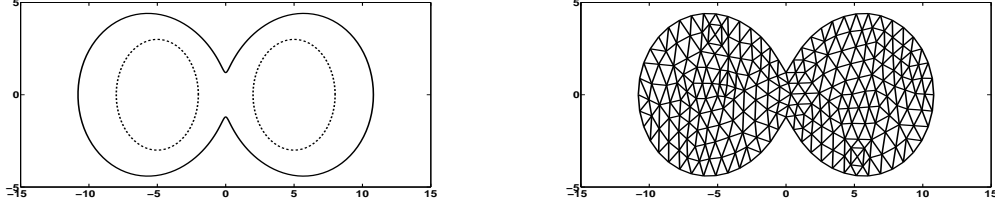


Figure 7: Example 5. Left figure: the nut shape artificial boundary (solid line) and the local support of the initial data u_0 (dashed line). Right figure: a nut shaped domain triangulation.



Example 5. To show the feasibility in the choice of the geometry of the artificial boundary, we apply the proposed scheme to the following problem: we consider the wave equation in the whole \mathbb{R}^2 , with zero source f and zero initial velocity v_0 . The initial datum $u_0(x_1, x_2) = e^{-5((x_1-5)^2+x_2^2)} + e^{-5((x_1+5)^2+x_2^2)}$ is a function with two humps, one centered at $(5, 0)$ and one centered at $(-5, 0)$, both having a “practical” local support included in a circle of radius smaller than 3 and height equals to 1 (notice that u_0 is not compactly supported, but the same considerations we made in Example 3 apply here too). Note that since the computational domain is not convex, in this case we have both outgoing and incoming waves.

Supposing we are interested in knowing the solution in a neighborhood of the center of the humps, we choose as artificial boundary a nut shape curve whose parametric equation is given by

$$\begin{aligned} x_1 &= \rho \cos(\theta), \\ x_2 &= \rho \sin(\theta), \end{aligned}$$

where $\rho = c(1 + e \cos(n\theta))$, with $c = 6, e = 0.8, n = 2$ (see Figure 7).

In Figure 8 we show the snapshots of the solution at different times, obtained with a triangulation of $n_t = 14337$ triangles, and $N = 300$ subdivisions of the time interval $[0, 30]$.

Example 6. In the final example, we aim at simulating those situations where one is interested in knowing the solution at points that are away from sources. For simplicity we suppose $u_0 = 0, v_0 = 0$, and $f \neq 0$. In this case, the artificial boundary \mathcal{B} is chosen in such a way that the source f is locally supported in the residual domain \mathcal{D} , while it is zero in Ω . Therefore, the artificial boundary condition reads:

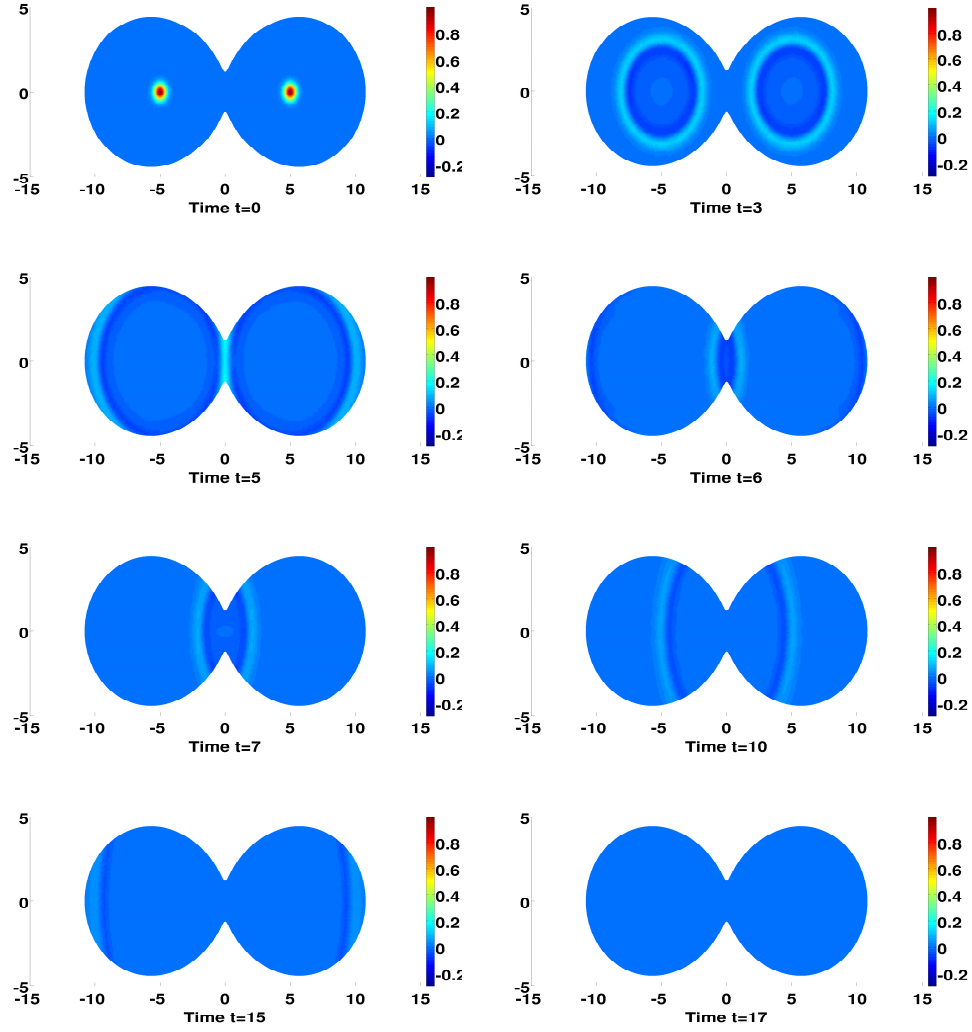
$$\frac{1}{2}u(\mathbf{x}, t) + \mathcal{V}\lambda(\mathbf{x}, t) + \mathcal{K}u(\mathbf{x}, t) = I_f(\mathbf{x}, t) \quad \text{in } \mathcal{B} \times (0, T].$$

In particular, we consider a source concentrated at a point \mathbf{x}_0 : $f(\mathbf{x}, t) = h(t)\delta(\mathbf{x} - \mathbf{x}_0)$, where $h(t)$ is a given smooth function. With this choice, the volume integral I_f (see (5)) has the following simpler form:

$$I_f(\mathbf{x}, t) = \int_0^t h(\tau)G(\mathbf{x} - \mathbf{x}_0, t - \tau)d\tau.$$

For the computation of the volume integral I_f , we apply the Lubich convolution technique. It is beyond the scope of the paper to enter into the details of the numerical evaluation of the volume terms, even for more general sources; this will be the subject of a future work.

Figure 8: Example 5. Snapshots of the solution at different times.



We compare the solution obtained with the above mentioned approach and the usual one, which consists of including the source into the finite computational domain Ω . We recall that, if the source is far from the area of interest, the last approach would require a much larger domain Ω , thus wasting computational time and space memory. In our test we place the source f at $\mathbf{x}_0 = (4, 0)$; Γ and \mathcal{B} are the circles of radius $r_0 = 1$ and R , respectively, both centered at the origin. We choose first $R = 2$ and then $R = 5$. In the first case, f is external to the finite computational domain, while in the second case it is included in the annulus bounded by Γ and \mathcal{B} .

We analyze three different cases, according to the choice of the function h :

$$h(t) = \begin{cases} 10^3 t^4 e^{-t} & (a) \\ 10^3 t^4 e^{-t} \sin(5t) & (b) \\ \sin(5t) & (c). \end{cases}$$

In each case, we compare the solution obtained at the mesh point $\mathbf{x}_{P_1} = (1.9995, 0.0436)$ for the external source, and at $\mathbf{x}_{P_2} = (2.0758, -0.0154)$ for the internal source, with $t \in [0, T]$. Since the finite domain is different in the above mentioned cases, the mesh points where we evaluate the solution are not exactly the same, so we do not overlap the two plots, but we compare them qualitatively. In particular, we plot the solutions obtained in the case $R = 5$ (left) and $R = 2$ (right). The case (a), with $T = 20$, is shown in Figure 9, while the case (b), with $T = 40$, in Figure 10. In both cases the wave is evanescent because f represents a vanishing source. On the contrary, in the case (c), where we have taken $T = 40$, the wave has a periodic constant oscillatory behavior (see Figure 11).

Finally, in Figure 12 we plot the approximate solution we have obtained at \mathbf{x}_{P_1} when the source (c) is located at $\mathbf{x}_0 = (10, 0)$ and $R = 2$. The result obtained for $R = 12$, in which case \mathbf{x}_0 belongs to the (bounded) computational domain Ω , requires a much finer time integration step, hence a very high CPU time, due to the oscillating behavior of the source. Because of this, we have omitted it.

We have also performed the same tests by replacing $\sin(5t)$ with $\sin(100t)$ in the expression of h . We have obtained very good results when \mathbf{x}_0 is external to Ω while, due to the very high oscillatory behavior of the source, the case of the internal source would have required a very fine time interval partition, that we could not perform in a reasonable time.

The last tests show, in particular, that it is more efficient to include the source term f in the I_f term of our NRBC, than to have to treat it as the right hand side of the wave equation.

Remark 3.4. *When we refine the discretization associated with the basic methods, chosen to solve the PDE problem on the Ω domain, to improve their accuracies the discretization of the proposed NRBC will also be (simultaneously) refined, assuming that integrals have been evaluated with sufficiently high accuracy. Thus, from this point of view, the NRBC is not an approximate one, such as those constructed in [6], [17], [26] and [27], where the truncation error of the ABC does not decrease, unless the number of terms defining the latter is increased, and the associated parameters are selected in an appropriate way, to reduce the spurious reflection below the desired level. This overhead is by no means negligible.*

Figure 9: Example 6. The internal source case (left), the external source case (right) for $h(t) = 10^3 t^4 e^{-t}$, $\mathbf{x}_0 = (4, 0)$.

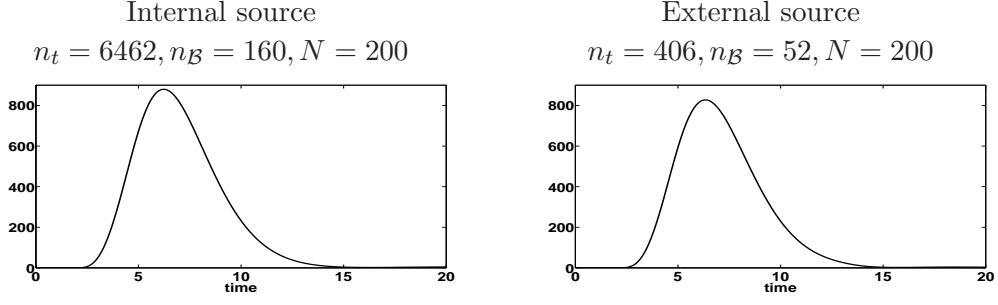


Figure 10: Example 6. The internal source case (left), the external source case (right) for $h(t) = 10^3 t^4 e^{-t} \sin(5t)$, $\mathbf{x}_0 = (4, 0)$.

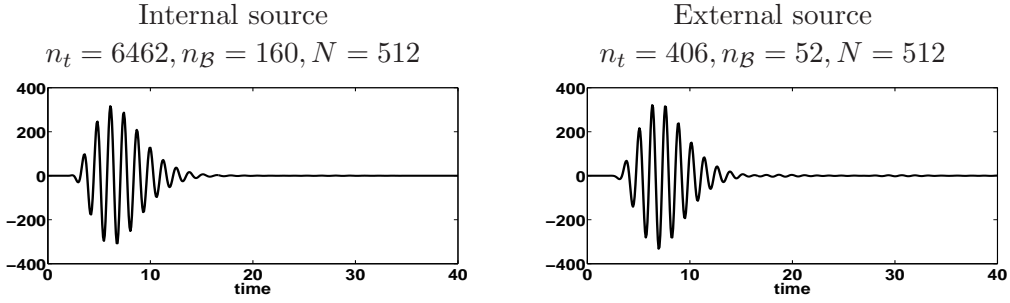


Figure 11: Example 6. The internal source case (left), the external source case (right) for $h(t) = \sin(5t)$, $\mathbf{x}_0 = (4, 0)$.

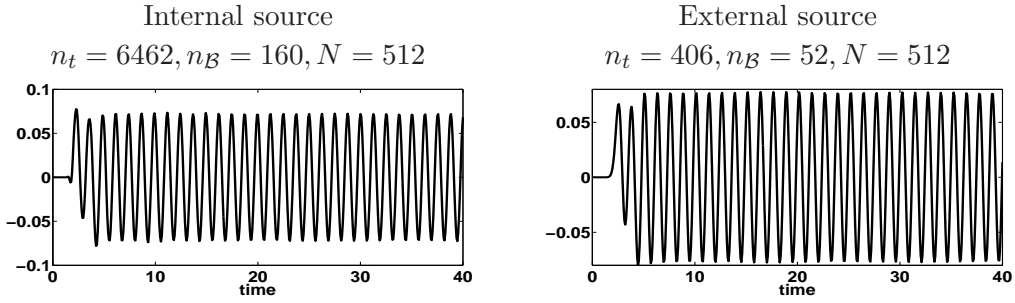
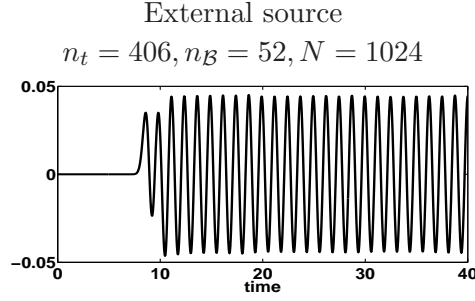


Figure 12: Example 6. The external source case for $h(t) = \sin(5t)$, $\mathbf{x}_0 = (10, 0)$.



3.4. Some remarks on the sparsity of the single and double layer operators

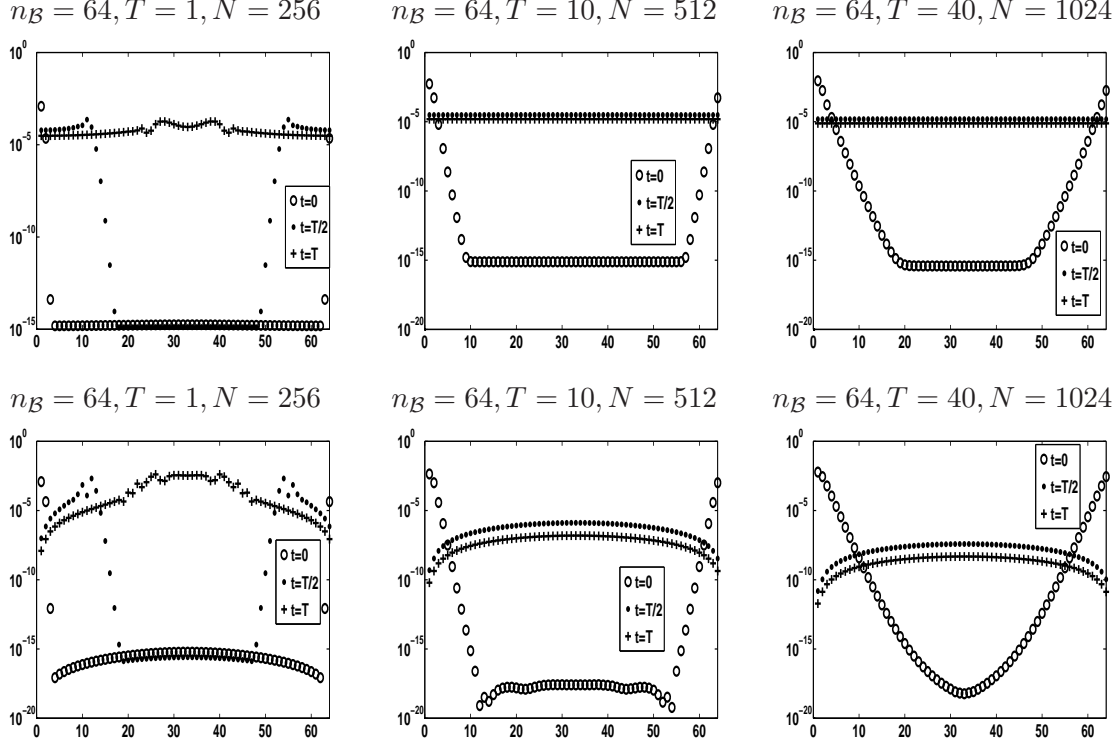
As already pointed out in the previous sections, the matrices \mathbf{V}_j and \mathbf{K}_j , $j = 0, \dots, N$ have many elements which are very small or even negligible. Therefore these matrices can be approximated by corresponding sparse ones. This property allows in principle to reduce the space memory storage and to speed up the computation. In the following, we discuss this aspect of the proposed NRBC, in the setting of some of the numerical examples we have described in the previous section.

In the context of Examples 1 and 3, where \mathcal{B} is the circle of radius $R = 0.5$ and $R = 8$ respectively, the choice of a uniform partition on \mathcal{B} allows to take advantage of the Toeplitz structure of the matrices \mathbf{V}_j and \mathbf{K}_j for each time step $j = 0, \dots, N$. Therefore, we need to construct and store only the first row of each matrix. In Figures (13) and (14) we show the semilogarithmic plot of the values of this row with respect to the number of nodes on \mathcal{B} and for different times, for the case $R = 0.5$ and $R = 8$. In particular, we plot the matrices \mathbf{V}_j and \mathbf{K}_j for $j = 0, N/2, N$ corresponding to the time steps $t = 0, T/2, T$ for different choices of T . We note that, for the time step $t = 0$ both \mathbf{V}_0 and \mathbf{K}_0 have few non negligible elements; these two matrices are the only ones involved in the final linear system that needs to be solved (see equation (33)). On the contrary, the sparsity of the remaining matrices reduces for large values of j . We recall that these matrices give a contribution to the right hand side term, namely they multiply the (known) solutions at the previous time steps.

For the case of the more general nut shape of Example 5, we have performed the same analysis. However, since the matrices do not have the Toeplitz structure anymore, we have first computed all of them and we have then cut all the elements which are below the threshold parameter $\varepsilon = 1e - 06$. The number of the remaining elements is denoted by n_z . In Figures 15, 16 we show their spy plot for different times. In both cases, the solutions obtained with the choice of $\varepsilon = 1e - 06$ are the same as the ones obtained by retaining all the elements of the matrices.

For simplicity, we have taken the same value of the above threshold parameter for all the \mathbf{V}_j and \mathbf{K}_j matrices. Further investigation for a possible more efficient choice of it is however needed. Also the use of special algorithms that take into account the sparsity property of the matrices is an argument that has not been treated in this work, but which is worthwhile to be analyzed.

Figure 13: Circular boundary \mathcal{B} : $R = 0.5$, $n_{\mathcal{B}} = 64$. The first row of the single layer matrices \mathbf{V}_j (top row), and of the double layer matrices \mathbf{K}_j (bottom row).



In the testing we have performed, the non reflecting property of our NRBC has shown to be very robust with respect to the above element cut strategy.

Figure 14: Circular boundary \mathcal{B} : $R = 8$, $n_{\mathcal{B}} = 64$. The first row of the single layer matrices \mathbf{V}_j (top row), and of the double layer matrices \mathbf{K}_j (bottom row).

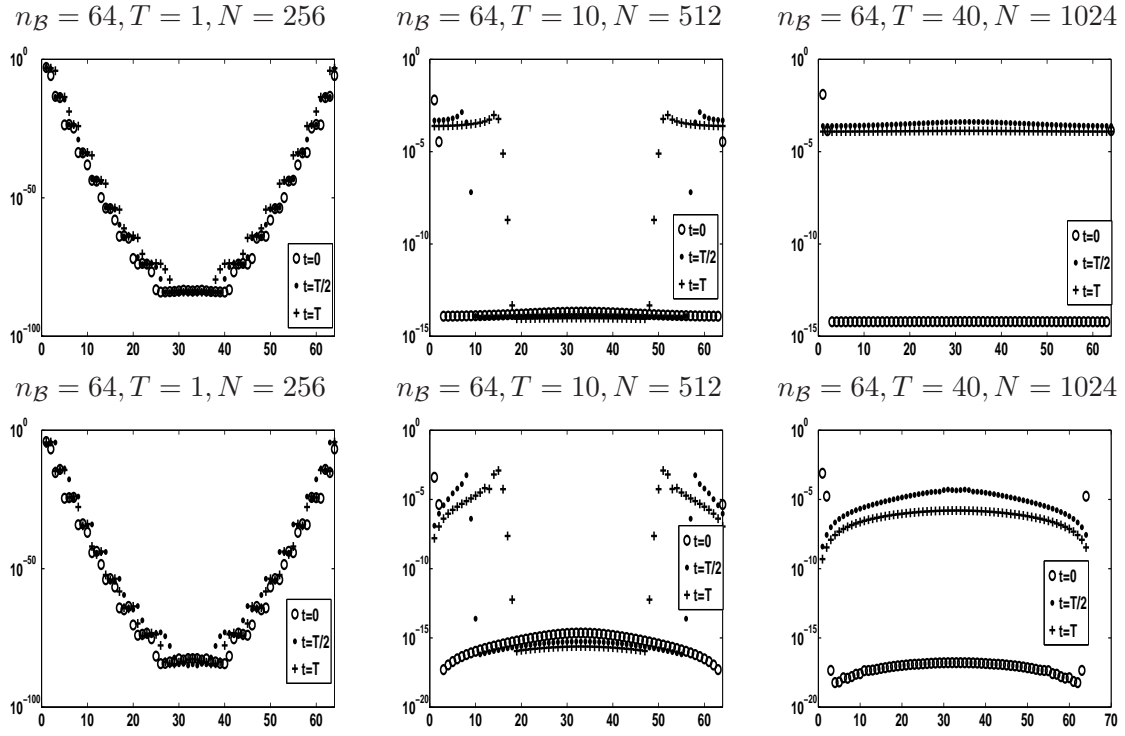


Figure 15: Nut shape boundary \mathcal{B} : $n_{\mathcal{B}} = 32$. The single layer matrices \mathbf{V}_j (top), the double layer matrices \mathbf{K}_j (bottom), $T = 1$.

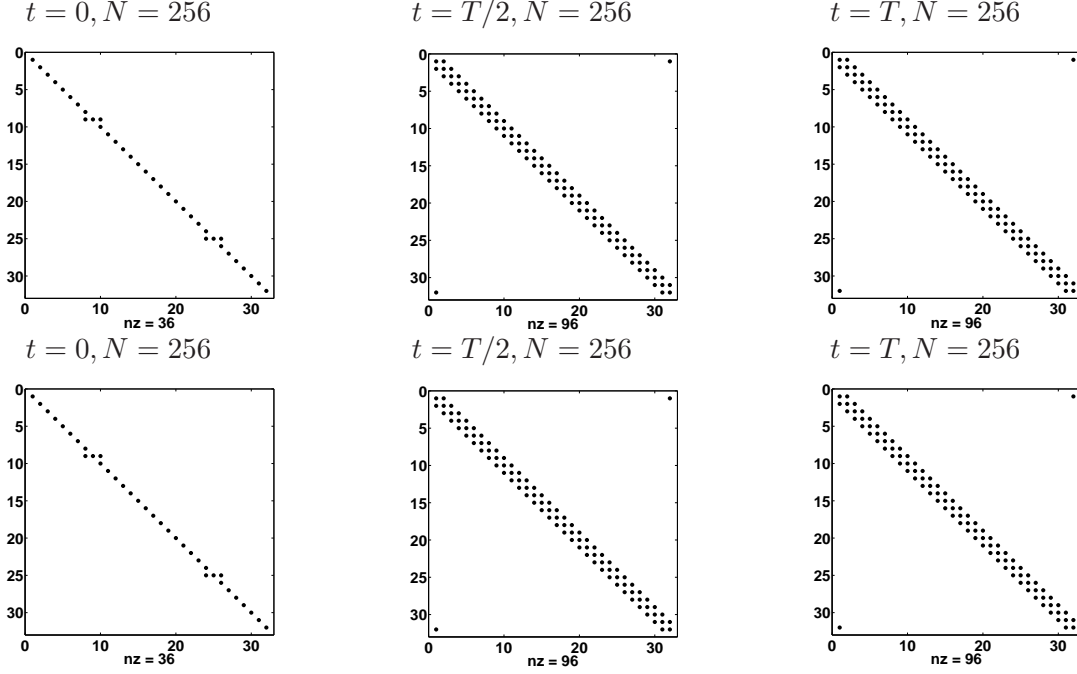
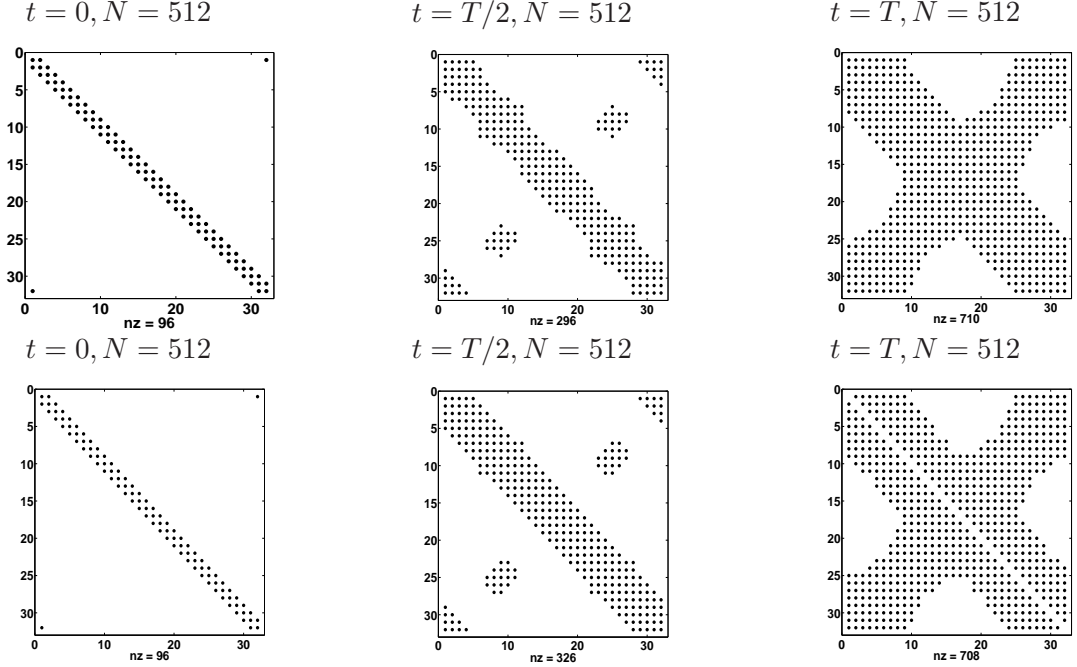


Figure 16: Nut shape boundary \mathcal{B} : $n_{\mathcal{B}} = 32$. The single layer matrices \mathbf{V}_j (top), the double layer matrices \mathbf{K}_j (bottom), $T = 10$.



In the latter case, if we take $T = 20$ and $N = 1024$, leaving unchanged the values of the remaining parameters, while the (approximate) matrices V_j and K_j , $j = 0, N/4, N/2$, coincide with the corresponding ones in Figure 16, V_N and K_N are practically full. In the following we analyze this behavior.

In the case of the V_j matrix, the associated $\omega_j^{\mathbf{V}}$ coefficients (see (21)) satisfy (see Remark 2 in [37]) the following property:

$$\omega_m^{\mathbf{V}}(\Delta_t; \|\psi(x_h) - \psi(y)\|) = \frac{1}{\Delta_t} \omega_m^{\mathbf{V}}\left(1; \frac{\|\psi(x_h) - \psi(y)\|}{\Delta_t}\right).$$

To analyze the behavior of these coefficients, we denote by I_i the support of the basis function $N_i(y)$, $i = 0 : M$ in (21), and define $d_{hi} = \|\psi(x_h) - \psi(y)\|/\Delta_t$, $y \in I_i$, where x_h is a chosen collocation point and we assume Δ_t fixed. Then, as Figure 3 in [37] shows, for all the abscissas d_{hi} that are on the right of a particular point, which appears to be approximately equal to j , the corresponding values of $\omega_j^{\mathbf{V}}$ are negligible. Thus, as j grows, the number of negligible elements of the matrix $h - th$ row becomes smaller. When d_{hi} is fixed and we let $j = 0 : N$, we have a reverse behavior of the same $\omega_j^{\mathbf{V}}$ coefficients (see Figure 11 in [37]). In this case the $\omega_j^{\mathbf{V}}$'s are negligible when j is smaller than a value approximately equal to d_{hi} . Similar remarks can be made on the behavior of the $\omega_m^{\mathbf{K}}$ coefficients.

Note that in Figure 16, the pattern of the nonzero elements is also related to the nut shape of the chosen artificial boundary \mathcal{B} .

Since in the 3D case the coefficients $\omega_j^{\mathcal{J}}(1; d_h)$ decay exponentially to zero as d_{hi} moves away from a value $d_h^* \approx j$, both from the left and right and sides (see [37]), we expect to have a (much) higher degree of sparsity. Indeed, as some preliminary testing we have performed shows, by examining the analytic representation of the $\omega_j^{\mathcal{J}}$ coefficients, we expect all matrices V_j, K_j to be banded, with a very small bandwidth. From this representation it also follows that when we assume T fixed, and we increase N , that is, we reduce the time step Δ_t , the degree of sparsity of the (approximate) matrices increases.

4. Reduced problem discretization. The Finite Difference Method

Although our main goal is the coupling of the NRBC with finite element methods, to further test the performance of our NRBC, in this last section we associate it also with a classical second order finite different scheme for the PDE space discretization.

To fix the ideas, we take Γ as the circle of radius r and \mathcal{B} as the circle of radius $R > r$. By using the polar coordinates $x = \rho \cos(\theta)$, $y = \rho \sin(\theta)$, $\rho \in [r, R]$, $\theta \in [0, 2\pi]$, we transform the bounded computational domain Ω into the rectangle $[r, R] \times [0, 2\pi]$. For simplicity, we consider the homogeneous problem, with $f = 0$ and zero initial conditions, so that in polar coordinates the wave equation takes the following well known form:

$$u_{tt} - (u_{\rho\rho} + \frac{1}{\rho^2}u_{\theta\theta} + \frac{1}{\rho}u_{\rho}) = 0.$$

We consider a discretization of Ω into cells, obtained by a uniform partition of $[r, R]$ and $[0, 2\pi]$ into M_ρ and M_θ subintervals, respectively. Such a choice determines the set of nodes (ρ_i, θ_j) , $i = 0, \dots, M_\rho$, $j = 0, \dots, M_\theta$, where $\rho_i = r + ih_\rho$, $h_\rho = (R-r)/M_\rho$ and $\theta_j = jh_\theta$, $h_\theta =$

$2\pi/M_\theta$. Then, in general we apply the Crank-Nicolson scheme for the time integration, and the classical second order finite difference scheme for the Laplace operator. Without entering into the details of the implementation, we remark that we use the periodicity property at $\theta = 2\pi$. Moreover, since we collocate the equation also at the nodes at $\rho = R$, where the solution is not known, to maintain the second order accuracy, we introduce the fictitious nodes $(\rho_{M_\rho+1}, \theta_j)$, $j = 0, \dots, M_\theta - 1$. The unknowns, at the time level $t_n = n\Delta_t$, are the values u_{ij}^n of the solution at the grid points (ρ_i, θ_j) , $i = 1, \dots, M_\rho$, $j = 0, \dots, M_\theta - 1$, and the values λ_j^n of the normal derivative of u at $\rho = R$, for $j = 0, \dots, M_\theta - 1$. The fictitious nodes are then eliminated by approximating the normal derivative λ_j^n with the symmetric finite difference formula

$$\lambda_j^n = \frac{u_{M_\rho+1,j}^n - u_{M_\rho-1,j}^n}{h_\rho} + O(h_\rho^2).$$

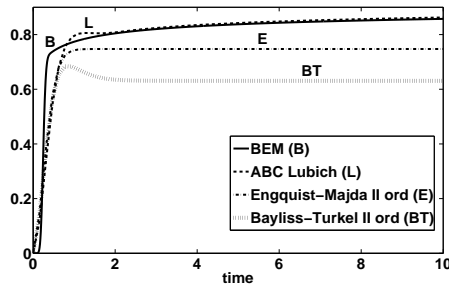
The final scheme is obtained by coupling the above Crank-Nicolson - finite difference method with our NRBC. The latter is approximated by the Lubich-collocation scheme described in Section 3.1.

We have applied this finite difference (FD) scheme to the previous Examples 1-3, that have been solved in Section 3.3 by applying the finite element method. The results are comparable. In the following, we report some of those we have obtained in the first two examples (see Figures 17, 18). In particular, we report the behavior of the problem solution at the mesh points $\mathbf{x} \in \mathcal{B}$, in the chosen time interval $[0, T]$. As in Section 3.3, in the case of Example 1 we replace the Crank-Nicolson method by the implicit Euler one; this because of the problem data non compatibility at $t = 0$.

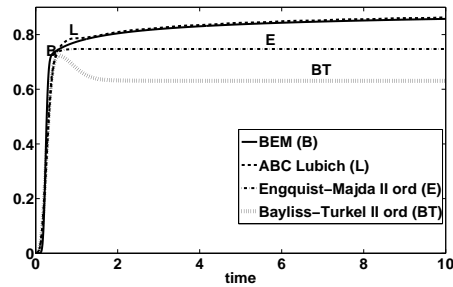
Since after introducing the polar coordinates, the artificial boundary \mathcal{B} is (exactly) mapped into a side of the transformed (rectangular) domain, we also test the behavior of the finite-difference scheme when our NRBC is replaced by the second order Engquist-Majda and Bayliss-Turkel ABC. In the right hand side picture in Fig. 18, we have not reported the approximation generated by the Engquist-Majda ABC. This because, after a behavior similar to that given by the Bayliss-Turkel ABC, up to about $t = 20$, it starts presenting increasingly high oscillations, that blow up at $t = 40$.

Figure 17: Finite differences. Solution behavior at $\mathbf{x} \in \mathcal{B}$.

Example 1. $M_\rho = 8, M_\theta = 8, N = 64$



Example 1. $M_\rho = 8, M_\theta = 8, N = 256$



To have an idea of the computational overhead due to our NRBC, we have compared the CPU times required to solve the above PDE problems by using our exact NRBC and the

Figure 18: Finite differences. Solution behavior at $\mathbf{x} \in \mathcal{B}$.

Example 1. $M_\rho = 30, M_\theta = 30, N = 1024$ Example 2. $M_\rho = 16, M_\theta = 16, N = 128$

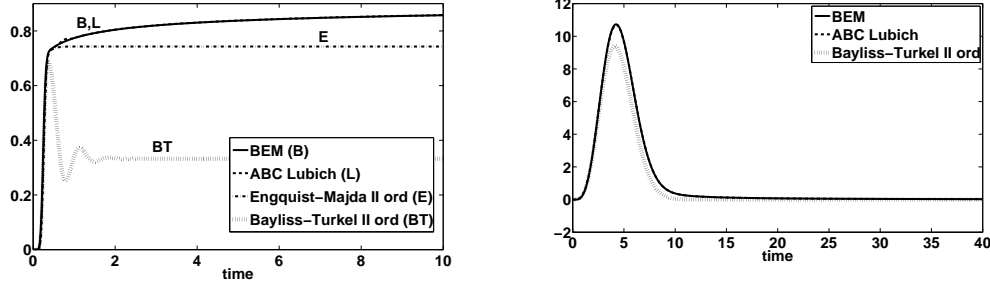
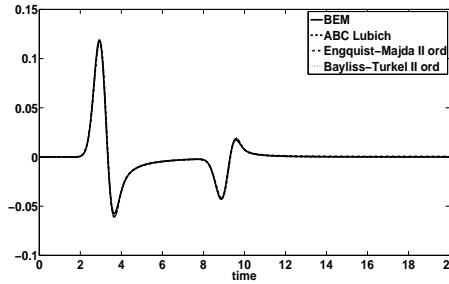


Figure 19: Finite differences. Solution behavior at $\mathbf{x} \cong (7.93, -1.08) \in \mathcal{B}$.

Example 3. $M_\rho = 200, M_\theta = 200, N = 512$



second order Engquist-Majda and Bayliss-Turkel ABC. The CPU time required by the latter two approaches has shown to be the same; therefore we report the ratio R_{CPU} between the CPU time of our method and that of the other two. With reference to the above Figures 17, 18, and in the same order, these are: 8.5, 13.8, 46, 7.5, respectively.

Note that this comparison is not fair, since we have used the same values of the mesh parameters M_ρ, M_θ, N , and we have not taken into account the accuracies produced by the three approaches. The required accuracy could be achieved by taking, for each NRBC, very different parameter values. Indeed, in the cases reported in the above first three figures, the approximants generated by the local ABC are very unsatisfactory. In the last figure, the Bayliss-Turkel approximation (slightly) underestimates the true solution, and its behavior does not change if we further increase the mesh parameters. In this case, to increase the accuracy, one should increase the artificial boundary radius R . Thus, in the next example, we compare our approach with those of Engquist-Majda and Bayliss-Turkel, by applying them to the problem of Example 3, taking $M_\rho = M_\theta = 200, N = 512$. The graph of the three approximate solutions at the chosen boundary mesh point are reported in Fig. 19. The corresponding value of R_{CPU} is 5.01. The reference solution is the one we have computed using the BEM method. In this case, the three approximate solutions and the reference one overlap (see Fig. 19).

For a final test, we consider Example 2 with $g(\mathbf{x}, t) = t^3 e^{-0.5(x_1^2 + x_2^2 - \sqrt{2}t)^2}$ and $[0, T] =$

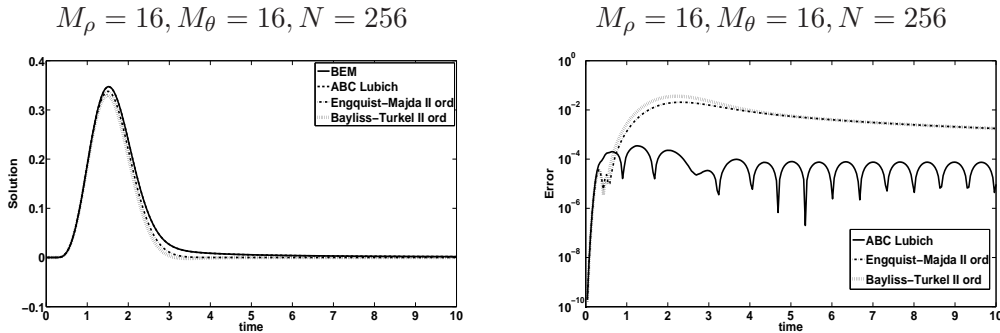
$[0, 20]$. After comparing our NRBC with those of Engquist-Majda and Bayliss-Turkel, we test the rate of convergence of our approach.

In Figure 20 we show the behavior of the solutions produced by the three approaches at a mesh point $\mathbf{x}_P \in \mathcal{B}$, and that of the associated (absolute) errors. As already pointed out, since the mesh on \mathcal{B} is uniform, when we apply our NRBC we take advantage of the Toeplitz structure of the single and double layer matrices \mathbf{V}_j and \mathbf{K}_j for each time step $j = 0, \dots, N$. In this case, the computational cost for the approximation of the exact NRBC is significantly reduced.

In Tables 1, we report the ratio R_{CPU} between the CPU times that, in our testing, the exact NRBC and the second order Engquist-Majda/Bayliss-Turkel approaches have required to solve the PDE problem, for different choices of the discretization parameters M_ρ, M_θ and N . In particular, in Table 1 a) we choose $M_\rho = M_\theta = N$. Note that R_{CPU} gets closer to 2 as the values of the parameters increase, namely, when the discretizations of the domain Ω and of the time interval $[0, T]$ are (simultaneously) refined. A similar comment applies to Table 1 b), where we fix the time step size $\Delta_t = T/16$ and decrease the mesh size h of the space discretization. In this case R_{CPU} approaches 1.

In Table 1 c) we fix the finite difference mesh, by choosing for example $M_\rho = 16, M_\theta = 16$, and refine only the time step by repeatedly doubling N . The value of the ratio R_{CPU} is higher, although it increases very mildly; this is due to the overhead required for updating the right hand side of our NRBC (33). We remark however that in the case of a non trivial problem, the values $M_\rho = 16, M_\theta = 16$ are fairly small, and in general, to increase the accuracy, one has to increase simultaneously all the parameters M_ρ, M_θ and N .

Figure 20: Finite differences. Solution and corresponding errors at $\mathbf{x} \in \mathcal{B}$.



Finally, we check the order of accuracy of the proposed numerical scheme, with reference to the case associated with Table 1 a). For each chosen value of N , we have computed the corresponding ℓ^2 -norm of the space approximation absolute error $e_N(t_n)$ at the time instant t_n , and the estimated order of convergence $EOC = EOC(t_n) = \log_2(e_N(t_n)/e_{N+1}(t_n))$. The results we have obtained are reported in Table 2. As reference solution, we have taken that obtained by using the space-time BEM described in [9], with a sufficiently fine discretization, to guarantee about 6 significant digits. The symbol “—” means that the maximum achievable accuracy has been obtained.

Although we do not have an error estimate for our (global) numerical scheme, we expect a rate of convergence of order 2, uniformly with respect to t_n and in the L^2 -norm with

Table 1: Ratio R_{CPU} between the CPU time of the exact and the second order Engquist-Majda NRBCs.

a): $M_\rho = M_\theta = N$.

b): $M_\rho = M_\theta, N = 16$.

c): $M_\rho = M_\theta = 16$.

M_ρ	R_{CPU}
8	3.58
16	4.63
32	4.47
64	3.16
128	2.73
256	2.70

M_ρ	R_{CPU}
8	5.39
16	5.41
32	4.37
64	2.60
128	1.49
256	1.01

N	R_{CPU}
8	4.79
16	4.63
32	5.58
64	6.29
128	7.01
256	8.08

Table 2: ℓ^2 -norm absolute error and convergence order for $M_\rho = M_\theta = N$, $T = 10$

N	$e_N(T/8)$	$EOC(T/8)$	$e_N(T/4)$	$EOC(T/4)$	$e_N(T)$	$EOC(T)$	$\ e_N\ _\infty$	EOC
8	$3.34E-02$		$5.75E-02$		$1.36E-02$		$6.79E-02$	
		0.42		1.45		0.07		1.37
16	$2.49E-02$		$2.11E-02$		$1.30E-02$		$2.62E-02$	
		3.14		1.46		6.19		1.78
32	$2.83E-03$		$7.66E-03$		$1.77E-04$		$7.66E-03$	
		0.02		2.43		3.37		1.28
64	$2.79E-03$		$1.41E-03$		$1.72E-05$		$3.14E-03$	
		1.74		2.37		2.16		1.91
128	$8.36E-04$		$2.74E-04$		$3.84E-06$		$8.36E-04$	
		2.08		2.21				2.04
256	$1.97E-04$		$5.94E-05$		--		$2.03E-04$	

respect to \mathbf{x} . We recall that for a similar finite difference scheme for a classical Dirichlet problem, the standard error estimate is usually given in the uniform norm with respect to the time variable. In this case, the associated error constant depends only on T , not on t_n . Therefore, it is the behavior of the maximum error, given at the knots of the time interval partition, that defines the convergence order of the method. In the last two columns of Table 2, we have reported this error and the associated EOC value. The pointwise error constant depends on t_n , and might have an erratic behavior, as N increases, in particular when N is not sufficiently large.

We have also compared the CPU time required by the finite difference scheme associated with our NRBC and with the second order Engquist-Majda ABC, when the time integration is performed by using an explicit (conditionally stable) formula. We remark that in this case, we can take advantage of the explicitness of this formula, so that in our approach the only system we have to solve, at each time step, is of the following form:

$$\mathbf{V}_0 \boldsymbol{\lambda}^n = \mathbf{b}^n \quad (34)$$

with \mathbf{b}^n known. Note that to reduce the computational cost of this latter, we can approximate, as suggested by Fig. 13-16, the matrix \mathbf{V}_0 by a very sparse version of it, which is almost banded with a very small bandwidth. When the time stepsize is small, this latter can even be almost diagonal.

In this case, besides being forced to choose the discretization parameters properly, to guarantee the method stability, the R_{CPU} ratio is very similar to that we have when the

Crank-Nicolson time integrator is used. Note that for our approach, the required CPU time can be further reduced by taking advantage of the sparsity of the approximated versions of the matrices $\mathbf{K}_j, \mathbf{V}_j$, which define the above known term \mathbf{b}^n .

5. Conclusions

Boundary integral equation formulations for elliptic PDE problems are nowadays a standard tool for solving some of these problems. They have also been interpreted as a relationship between the (unknown) solution and its normal derivative, which has then been used to define exact NRBC, to be associated with finite difference or finite element numerical schemes. Their analogues for time-dependent problems have not received the same attention, and their development is still at an early stage.

In these last years a few papers have appeared, to solve exterior problems for the wave equation by means of space-time BIE. A common feature of these papers is the use of special convolution quadratures developed by Ch. Lubich in the late 80's, to discretize the time integral. These are then coupled with a standard Galerkin or collocation method for the space discretization. Only for the Galerkin case theoretical results have been derived.

In this paper we have proposed to use such space-time BIE to define, as in the case of the above mentioned elliptic problems, a NRBC for the solution of 2D exterior wave equation problems, in a bounded domain of interest, by means of standard FE or FD methods.

To discretize this BIE, we have coupled a second order Lubich rule with a space collocation method defined by a continuous piecewise linear approximant. The proposed NRBC has the following main features:

- it is of exact type;
- although it is non local in time and space, from the computational complexity point of view, the proposed discretization of it is almost local with respect to both variables. This property follows from the special properties of the coefficients of the Lubich rule;
- it allows to take smooth artificial boundaries of arbitrary shape. Moreover, some numerical testing we have performed seems to confirm that its good non reflecting property is maintained even when the chosen boundary has corners.
- it is transparent for both outgoing and incoming waves, and it naturally allows the treatment of non homogeneous data. Far field sources do not have to be necessarily included in the finite computational domain;
- we have applied it to multi-scattering problems and obtained results very similar to those reported in Section 4.
- its generalization to 3D problems is straightforward (see [9]).

We finally recall that very recently, a first attempt to construct a one-step/variable step method, obtained by coupling a Lubich quadrature with a Galerkin (space) method, has been proposed and studied (see [31]) to solve the space time BIE associated with the wave problem.

Unfortunately, for the Lubich-collocation approach we have proposed, no theoretical results have been till now derived, except for the straightforward one reported in Proposition 3.2. In particular, no mathematical proofs have been obtained for the stability and convergence of the coupling of our NRBC with the chosen FE or FD methods. Because of this,

to verify if these properties hold, we have performed an intensive numerical testing, which seems to confirm that indeed they hold. Further investigation is however still needed.

All the numerical computation has been performed on a PC with Intel Core2® Quad Q6600 (2.40GHz). To perform our numerical testing we have written standard (i.e., sequential) Matlab® codes.

Acknowledgment. The authors are grateful to the referees for their valuable comments.

References

- [1] B. Alpert, L. Greengard, T. Hagstrom, Rapid evaluation of nonreflecting boundary kernels for time-domain wave propagation, *SIAM J. Num. Anal* 37 (2000) 1138-1164.
- [2] L. Banjai, M. Messner, M. Schanz, Runge-Kutta convolution quadrature for the boundary element method, *Comput. Methods Appl. Mech. Engrg.* 245-246 (2012) 90-101.
- [3] A. Bayliss, E. Turkel, Radiation boundary conditions for wave-like equations, *Comm. Pure Appl. Math.* 33 (1980) 707-725.
- [4] G. Ben-Porat, D. Givoli, Solution of unbounded domain problems using elliptic artificial boundaries, *Commun. Num. Meth. Engng* 11 (1995) 735-741.
- [5] R. Clayton, B. Engquist, Absorbing boundary conditions for acoustic and elastic wave equation, *Bulletin of the Seismological Society of America* 67 (1977) 1529-1540.
- [6] F. Collino, High order absorbing boundary conditions for wave propagation models. Straight line boundary and corner cases, in: R. Kleinmann, et. al. (Eds.) *Proc. 2nd Int. Conf. on Mathematics and Numerical Aspects of Wave Propagation*, SIAM, Philadelphia, 1993, pp. 161-171.
- [7] B. Engquist, A. Majda, Absorbing boundary conditions for numerical simulation of waves, *Math. Comput.* 31 (1977) 629-651.
- [8] S. Falletta, B.P.Lamichhane, Mortar finite elements for a heat transfer problem on sliding meshes, *Calcolo* 46 (2009) 131-148.
- [9] S. Falletta, G. Monegato, L. Scuderi, A space-time BIE method for nonhomogeneous exterior wave equation problems. The Dirichlet case, *IMA J. Numer. Anal.* 32 (2012) 202-226.
- [10] D. Givoli, A combined Analytic-Finite Element Method for elastic shells, *Int.J. of Solids and Structures* 26 (1990) 185-198.
- [11] D. Givoli, Finite element analysis of long cylindrical shells, *AIAA Journal* 28 (1990) 1331-1333.
- [12] D. Givoli, *Numerical Methods for Problems in Infinite Domains*, Elsevier, Amsterdam, 1992.

- [13] D. Givoli, Recent advances in the DtN FE method, *Archives of Computational Methods in Engineering* 6 (1999) 71-116.
- [14] D. Givoli, High-order local non-reflecting boundary conditions: A review, *Wave Motion* 39 (2004) 319-326.
- [15] D. Givoli, J.B. Keller, Exact non-reflecting boundary conditions, *J. Comp. Phys.* 82 (1989) 172-192.
- [16] D. Givoli, J.B. Keller, A Finite Element Method for large domains, *Comp. Meth. Appl. Mech. Engng.* 76 (1989) 41-66.
- [17] D. Givoli, B. Neta, High-order non-reflecting boundary scheme for time dependent waves, *J. Comput. Phys.* 186 (2003) 24-46.
- [18] D. Givoli, S. Vigdergauz, Artificial boundary conditions for 2D problems in geophysics, *Comput. Meth. Appl. Mech. Engng.* 110 (1993) 87-101.
- [19] D. Goldman, P.E. Barbone, Dirichlet to Neumann maps for the representation of equipment with weak nonlinearities, *ASME Noise control and acoustic division* 22 (1996) 71-76.
- [20] I.S. Gradshteyn, I.M. Ryzhik, *Table of Integrals, Series, and Products*, Academic Press, New York, 2007.
- [21] M.J. Grote, J.B. Keller, On nonreflecting boundary conditions, *J. Comput. Phys.* 122 (1995) 231-243.
- [22] M.J. Grote, J.B. Keller, Nonreflecting boundary conditions for time dependent scattering problems, *J. Comput. Phys.* 127 (1996) 52-65.
- [23] M.J. Grote, Ch. Kirsch, Nonreflecting boundary condition for time-dependent multiple scattering, *J. Comput. Phys.* 221 (2007) 41-62.
- [24] M.J. Grote, I. Sim, Local nonreflecting boundary condition for time-dependent multiple scattering, *J. Comput. Phys.* 230 (2011) 3135-3154.
- [25] W. Hackbusch, W. Kress, S. Sauter, Sparse convolution quadrature for time domain boundary integral formulations of the wave equation, *IMA J. Numer. Anal.* 29 (2009) 158-179.
- [26] T. Hagstrom, T. Warburton, A formulation of asymptotic and exact boundary conditions using local operators, *Appl. Numer. Math.* 27 (1998) 403-416.
- [27] 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.
- [28] R.L. Higdon, Absorbing boundary conditions for difference approximations to the multidimensional wave equation, *Math. Comp.* 47 (1986) 437-459.

- [29] R.L. Higdon, Numerical absorbing boundary conditions for the wave equation, *Math. Comp.* 49 (1987) 65-90.
- [30] A.R. Laliena, F-J. Sayas, Theoretical aspects of the application of convolution quadrature to scattering of acoustic waves, *Num. Math.* 112 (2009) 637-678.
- [31] M. Lopez-Fernandez, S. Sauter, Generalized convolution quadratures with variable time stepping, *IMA J. Numer. Anal.*, doi: 10.1093/imanum/drs034(2013)
- [32] C. Lubich, A. Schädle, Fast convolution for non-reflecting boundary conditions, *SIAM J. Sci. Comput.* 24 (2002) 161-182.
- [33] Ch. Lubich, Convolution quadrature and discretized operational calculus. I, *Num. Math.* 52 (1988) 129-145.
- [34] Ch. Lubich, On the multistep time discretization of linear initial-boundary value problems and their boundary integral equations, *Num. Math.* 67 (1994) 365-389.
- [35] D.B. Meade, G.W. Slade, A.F. Peterson, K.J. Webb, Comparison of local radiation boundary conditions for the scalar Helmholtz equation with general boundary shapes, *IEEE Trans. Antenn. Propagat.* 43 (1995) 6-10.
- [36] M. Medvinsky, E. Turkel, On surface radiation conditions for an ellipse, *J. Comput. Appl. Math.* 234 (2010) 1647-1655.
- [37] G. Monegato, L. Scuderi, M.P. Stanić, Lubich convolution quadratures and their application to problems described by space-time BIEs, *Numerical Algorithms* 56 (2010) 405-436.
- [38] I. Patlashenko, D. Givoli, Non reflecting finite element schemes for three-dimensional acoustic waves, *J. Computational Acoustics* 5 (1997) 95-115.
- [39] I. Patlashenko, D. Givoli, DtN maps for unbounded wave guides, *J. Comput. Phys.* 143 (1998) 200-223.
- [40] A. Quarteroni, A. Valli, *Numerical Approximation of Partial Differential Equations*, Springer, Berlin, 1994.
- [41] A. Taflov, S.C. Hagness, *Computational Electrodynamics. The finite-difference time-domain method*, Hartech House, Boston-London, 2005.