Exact non-reflecting boundary condition for 3D time-dependent multiple scattering-multiple source problems

Exact non-reflecting boundary condition for 3D time-dependent multiple scattering-multiple source problems☆

Silvia Falletta[a], Giovanni Monegato[b]

[a]*Dip. Scienze Matematiche "J.L. Lagrange" - Politecnico di Torino, e-mail: silvia.falletta@polito.it*
[b]*Dip. Scienze Matematiche "J.L. Lagrange" - Politecnico di Torino, e-mail: giovanni.monegato@polito.it*

Abstract

We consider some 3D wave equation problems defined in an unbounded domain, possibly with far field sources. For their solution, by means of standard finite element 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. The computational complexity of the discrete convolution is of order $N \log N$, being N the total number of time steps performed. That of the fully discretized NRBC is $O(N_{\mathcal{B}}^2 N \log N)$, where $N_{\mathcal{B}}$ denotes the number of mesh points taken on $\mathcal{B}$.

Besides showing a good accuracy and numerical stability, the proposed NRBC has the property of being suitable for artificial boundaries of general shapes. It also allows the treatment of far field (multiple) sources, that do not have to be necessarily included in the finite computational domain, being transparent not only for outgoing waves but also for incoming ones. This approach is in particular used to solve multiple scattering problems.

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

1. Introduction

A key issue for solving PDE problems in unbounded domains is the introduction of proper absorbing artificial boundaries, with associated boundary conditions, to define a bounded computational domain of interest where the solution of the initial boundary value problem defined in this domain coincides with the corresponding restriction of the solution of the original problem. Since the mid 90's, two classes of methods have emerged as especially powerful: the Perfectly Matched Layer (PML) method and the method of using local high-order Absorbing (or Non Reflecting) Boundary Conditions (ABC or NRBC).

The PML method, first introduced by Bérenger in 1994 [5], is a very popular technique for constructing artificial boundary conditions which absorb outgoing waves. Since its first

☆This work was supported by the Ministero dell'Istruzione, dell'Università e della Ricerca of Italy, under the research program PRIN 2012: Innovative methodologies for PDE-based numerical modelling.

appearance, it is continuously seeing significant developments. This method essentially consists in surrounding the domain of interest by an artificial absorbing layer in which waves are trapped and (exponentially) attenuated. Thus sources must necessarily be included in the above domain. Then, the new problem defined in the bounded domain is solved by using classical finite difference or finite element methods. For a survey see, for example, [37], Chapter 7, [6], [29]. Besides having some good properties, which make PML an efficient approach for solving several wave propagation problems, it also has some limitations; for a list of these, see for example [20]. In particular, at the moment it does not appear to be an efficient method for solving multiple sources/multiple scattering problems. Therefore, since the solution of these latter is the main goal of our paper, from now on we will consider only the NRBC approach.

The NRBC method consists of introducing an artificial boundary $\mathcal{B}$ that truncates the infinite domain and determines two distinct regions: a bounded one of interest Ω and a residual infinite domain $\mathcal{D}$. By analyzing the problem in $\mathcal{D}$, a non reflecting boundary condition on $\mathcal{B}$ is derived in order to avoid spurious reflections. Once a NRBC has been defined, this is coupled with the condition given on the boundary of the original problem physical domain and with the known initial values, to uniquely define the solution of the corresponding problem in Ω. Then, this latter can be solved, by coupling a time integrator with, for example, a finite difference or a finite element method.

The earlier approximate NRBCs, still widely used, are those proposed by Engquist and Majda [11], which are local and of order up to 2. Much 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 [9], Givoli and Neta [21] 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. When $\mathcal{B}$ is a disk/sphere, Hagstrom and Hariharan [26] derived a new formulation of the classical Bayliss and Turkel [4] NRBC of arbitrarily high order, without using derivatives of order greater than 2. For a review of these methods, see [19].

Many papers have been published on this topic, in particular in the last two decades; their number is too large to mention them. For a review, see for example [17], [18], [19], [14]. All these papers, except for [37], Sections 5.5, 5.6, [23], [24], [14], deal with the construction of NRBC with the property of absorbing only outgoing wave, not waves that are either outgoing or incoming. Therefore, known sources must necessarily be included in the computational domain. However, this can be a severe drawback when, for example, sources are far away from the physical domain. Moreover, the NRBC holds only for a single convex artificial boundary having a special shape, like a circle (sphere) or ellipse (ellipsoid). Only in the last years multiple scattering problems have been examined (see [23], [24]).

Very recently, in [14], we have proposed a global non reflecting boundary condition for the solution of two-dimensional exterior problems for the classical wave equation, which is given by a linear combination of a single and a double layer operators. It is defined by 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 is of exact type, and it holds for a (smooth) curve of arbitrary shape; therefore, it can be used also in situations of multiple scattering, 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, the proposed NRBC naturally includes the effects of these data and it is automatically transparent for outgoing waves as well as for incoming ones.

For the discretization of the artificial condition, namely for the approximation of the single and double layer operators, we have proposed a numerical scheme which is based on a second order Lubich discrete convolution quadrature formula (see [33]) for the discretization of the time integral, coupled with a classical collocation method in space. The computational complexity of the discrete convolution rule is of order $N \log N$, being N the total number of time steps performed. That of the fully discretized NRBC is $O(N_{\mathcal{B}}^2 N \log N)$, where $N_{\mathcal{B}}$ denotes the number of mesh points taken on $\mathcal{B}$. When the discretization of the bounded domain Ω, where we apply the chosen finite element scheme, is refined, and the time stepsize is simultaneously reduced, the accuracy of the NRBC discretization increases.

The numerical examples presented in [14] for the two-dimensional case, show that indeed the proposed NRBC is very competitive, from both the accuracy and the computational cost points of view, with some of local type, such as the second order Engquist-Majda and Bayliss-Turkel ones.

In this new paper, we extend the results presented in [14] to the three-dimensional case, and also consider multiple scattering/multiple source problems. The computational cost of this NRBC is significantly higher than that of corresponding local conditions; however, we believe that its generality and new applications should justify it. In any case, a first attempt to reduce it is described in Section 3.4. Five numerical examples are presented in Section 6, while in the last section we draw some conclusions and outline some possible further improvements.

2. Exact non reflecting boundary conditions for multiple scattering

For the reader convenience, here we recall the NRBC representation and the associated PDE problem formulation in the 3D multiple scattering case. These are obtained by generalizing those described in [14] for the 2D simple scattering problem.

Thus, we consider the problem of a wave propagating through a homogeneous medium in three dimension and impinging upon a scatterer made up of $\boldsymbol{\kappa}$ bounded obstacles of arbitrary shape. We assume that the scatterers are impenetrable and well separated from each other. We denote by $\Omega_k^i \subset \mathbb{R}^3$, with $k = 1, \cdots, \boldsymbol{\kappa}$, $\Omega_k^i \cap \Omega_\ell^i = \emptyset$ for $k \neq \ell$, the open bounded domains with smooth boundaries Γ_k, and by $\Omega^i = \cup_{k=1}^{\boldsymbol{\kappa}} \Omega_k^i \subset \mathbb{R}^3$ the connected open domain bounded by the union of the obstacle boundaries $\Gamma = \cup_{k=1}^{\boldsymbol{\kappa}} \Gamma_k$. Then, we set $\Omega^e = \mathbb{R}^3 \setminus \overline{\Omega^i}$, and consider the following wave propagation problem in Ω^e:

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

As often occurs in practical situations, we assume that the initial values u_0, v_0 and the source term f have local supports.

When one has to determine the solution u^e of the above problem in a bounded subregion of Ω^e, surrounding the physical obstacles Ω_k^i, it is necessary to truncate the infinite domain Ω^e by introducing an artificial smooth boundary $\mathcal{B}$. However, the idea of introducing a single artificial boundary $\mathcal{B}$ that encloses all the obstacles becomes too expensive when, for example, the scatterers are far from each other, and one has to determine the solution u^e only in a neighborhood of each scatterer. In this case, it is preferable to surround each single obstacle by an artificial boundary $\mathcal{B}_k$ and compute the problem solution in the domains of interest.

We denote by Ω_k the subdomain bounded internally by Γ_k and externally by $\mathcal{B}_k$, and by $\Omega = \cup_{k=1}^{\boldsymbol{\kappa}} \Omega_k$ the bounded computational domain of interest. Finally, we set $\mathcal{D} = \mathbb{R}^3 \setminus \overline{\Omega}$. To solve our problem in the Ω domain, we need to prescribe Non Reflecting Boundary Conditions on $\mathcal{B} = \cup_{k=1}^{\boldsymbol{\kappa}} \mathcal{B}_k$, which allow outgoing waves leave Ω_k without spurious reflections. These waves propagate to all other subdomains, and are then reflected by the other scatterers; therefore, subsequently they reenter in Ω_k.

We remark that 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.

Following [12], where a BIE has been derived for the non homogeneous wave equation with non trivial initial data, the derivation of a NRBC in the case of multiple connected artificial boundaries is fairly simple. To write it in a more compact form, we introduce the single and double layer integral operators, defined by

$$\begin{aligned} \boldsymbol{\mathcal{V}}\psi(\mathbf{x},t) &:= \int_0^t \int_{\mathcal{B}} G(\mathbf{x}-\mathbf{y},t-\tau)\psi(\mathbf{y},\tau) d\mathcal{B}_{\mathbf{y}} d\tau \\ &= \sum_{k=1}^{\boldsymbol{\kappa}} \int_0^t \int_{\mathcal{B}_k} G(\mathbf{x}-\mathbf{y},t-\tau)\psi(\mathbf{y},\tau) d\mathcal{B}_{\mathbf{y}} d\tau, \end{aligned}$$

and

$$\begin{aligned} \boldsymbol{\mathcal{K}}\varphi(\mathbf{x},t) &:= \int_0^t \int_{\mathcal{B}} \partial_{\mathbf{n}_{\mathcal{D}}} G(\mathbf{x}-\mathbf{y},t-\tau)\varphi(\mathbf{y},\tau) d\mathcal{B}_{\mathbf{y}} d\tau \\ &= \sum_{k=1}^{\boldsymbol{\kappa}} \int_0^t \int_{\mathcal{B}_k} \partial_{\mathbf{n}_{\mathcal{D}}} G(\mathbf{x}-\mathbf{y},t-\tau)\varphi(\mathbf{y},\tau) d\mathcal{B}_{\mathbf{y}} d\tau, \end{aligned}$$

respectively, where $\partial_{\mathbf{n}_{\mathcal{D}}}$ denotes the outward unit normal derivative on the boundary $\mathcal{B}$ for the problem defined in the residual domain $\mathcal{D}$, and $G(\mathbf{x},t)$ is the fundamental solution of the wave equation given in (1), that is:

$$G(\mathbf{x},t) = \frac{\delta(t-\|\mathbf{x}\|)}{4\pi\|\mathbf{x}\|}, \tag{2}$$

$\delta(\cdot)$, being the well known Dirac delta function. The $NRBC$ on $\mathcal{B}$ is then given by:

$$\frac{1}{2}u(\mathbf{x},t) = \boldsymbol{\mathcal{V}}\partial_{\mathbf{n}_{\mathcal{D}}} u(\mathbf{x},t) - \boldsymbol{\mathcal{K}}u(\mathbf{x},t) + I_{u_0}(\mathbf{x},t) + I_{v_0}(\mathbf{x},t) + I_f(\mathbf{x},t) \quad \mathbf{x} \in \mathcal{B}, \tag{3}$$

where the "volume" terms I_{u_0}, I_{v_0} and I_f are generated by the non homogeneous initial conditions and the non trivial source, respectively. These volume terms have the following integral representations:

$$I_{u_0}(\mathbf{x},t) = \begin{cases} \dfrac{\partial}{\partial t}\displaystyle\int_{\mathrm{supp}(u_0)} u_0(y)G(\mathbf{x}-\mathbf{y},t)dy, & \text{if} \quad \mathrm{supp}(u_0) \subset \mathcal{D} \\ 0, & \text{if} \quad \mathrm{supp}(u_0) \subset \Omega \end{cases} \tag{4}$$

$$I_{v_0}(\mathbf{x},t) = \begin{cases} \displaystyle\int_{\mathrm{supp}(v_0)} v_0(y)G(\mathbf{x}-\mathbf{y},t)dy, & \text{if} \quad \mathrm{supp}(v_0) \subset \mathcal{D} \\ 0, & \text{if} \quad \mathrm{supp}(v_0) \subset \Omega \end{cases} \tag{5}$$

$$I_f(\mathbf{x},t) = \begin{cases} \displaystyle\int_0^t \int_{\mathrm{supp}(f)} f(y,\tau)G(\mathbf{x}-\mathbf{y},t-\tau)dyd\tau, & \text{if} \quad \mathrm{supp}(f) \subset \mathcal{D} \\ 0, & \text{if} \quad \mathrm{supp}(f) \subset \Omega. \end{cases} \tag{6}$$

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

$$\begin{cases} u_{tt}(\mathbf{x},t) - \Delta u(\mathbf{x},t) & = \tilde{f}(\mathbf{x},t) & \text{in}\,\Omega \times (0,T) \\ u(\mathbf{x},t) & = g(\mathbf{x},t) & \text{in}\,\Gamma \times (0,T) \\ \frac{1}{2}u(\mathbf{x},t) + \boldsymbol{\mathcal{V}}\partial_{\mathbf{n}}u(\mathbf{x},t) + \boldsymbol{\mathcal{K}}u(\mathbf{x},t) & = I_{u_0}(\mathbf{x},t) + I_{v_0}(\mathbf{x},t) + I_f(\mathbf{x},t) & \text{in}\,\mathcal{B} \times (0,T) \\ u(\mathbf{x},0) & = \tilde{u}_0(\mathbf{x}) & \text{in}\,\Omega \\ u_t(\mathbf{x},0) & = \tilde{v}_0(\mathbf{x}) & \text{in}\,\Omega, \end{cases} \tag{7}$$

where

$$\tilde{u}_0 = \begin{cases} 0 & \text{if} \quad \mathrm{supp}(u_0) \subset \mathcal{D}. \\ u_0 & \text{if} \quad \mathrm{supp}(u_0) \subset \Omega \end{cases}$$

$$\tilde{v}_0 = \begin{cases} 0 & \text{if} \quad \mathrm{supp}(v_0) \subset \mathcal{D}. \\ v_0 & \text{if} \quad \mathrm{supp}(v_0) \subset \Omega \end{cases}$$

$$\tilde{f} = \begin{cases} 0 & \text{if} \quad \mathrm{supp}(f) \subset \mathcal{D}. \\ f & \text{if} \quad \mathrm{supp}(f) \subset \Omega \end{cases}$$

To discretize the NRBC, i.e., the single and double layer operators, we propose a numerical scheme which is based on a discrete convolution quadrature formula, for the time integral approximation, and a classical collocation method for the space integral discretization. If N denotes the number of time steps to be performed, the proposed NRBC discretization requires $O(N \log N)$ operations to compute, for each given collocation (space) point, the associated temporal convolution at all chosen instants.

We set $u(t)(\mathbf{x}) = u(\mathbf{x},t)$ and 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 $\mathcal{B}$. Following [14], but in a more general setting here, for any given $t \geq 0$ we also introduce the functional spaces

$$X_k = X_k(t) = \{u_k(t) \in H^1(\Omega_k) : u_k(t)|_{\Gamma_k} = g(t)|_{\Gamma_k}\}, \qquad X = X(t) = \prod_{k=1}^{\boldsymbol{\kappa}} X_k$$

and

$$X_{k,0} = \{u_k(t) \in H^1(\Omega_k) : u_k(t)|_{\Gamma_k} = 0\}, \qquad X_0 = \prod_{k=1}^{\boldsymbol{\kappa}} X_{k,0}.$$

Similarly, we set

$$H^{-1/2}(\mathcal{B}) = \prod_{k=1}^{\boldsymbol{\kappa}} H^{-1/2}(\mathcal{B}_k).$$

Then, the problem defined in the domain of interest Ω takes the following form:
given $\tilde{f} \in L^2(\Omega \times (0,T)), \tilde{u}_0 \in X, \tilde{v}_0 \in L^2(\Omega)$, *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*

$$\begin{cases} \frac{d^2}{dt^2}(u(t),w)_\Omega + a(u(t),w) - (\lambda(t),w)_\mathcal{B} &= (\tilde{f}(t),w)_\Omega, & \forall w \in X_0 \\ \frac{1}{2}u(\mathbf{x},t) + \boldsymbol{\mathcal{V}}\lambda(\mathbf{x},t) + \boldsymbol{\mathcal{K}}u(\mathbf{x},t) &= I_{u_0}(\mathbf{x},t) + I_{v_0}(\mathbf{x},t) + I_f(\mathbf{x},t) & \text{on } \mathcal{B} \\ u(0) &= \tilde{u}_0 \quad \text{in } \Omega \\ \frac{du}{dt}(0) &= \tilde{v}_0 \quad \text{in } \Omega. \end{cases} \tag{8}$$

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

$$a(v,w) = \int_\Omega \nabla v \cdot \nabla w = \sum_{k=1}^{\boldsymbol{\kappa}} \int_{\Omega_k} \nabla v_k \cdot \nabla w_k,$$

and, $(v,w)_D = \int_D vw = \sum_{k=1}^{\boldsymbol{\kappa}} \int_{D_k} v_k w_k$, *where, depending on its occurrence,* D *is either* Ω *or* $\mathcal{B}$.

3. Discretization of the NRBC

In this section we outline the discretization formulas we will use to solve the problem formulation (8). These are very similar to those obtained in [14] for the 2D case; however, in the 3D case we have some favorable properties that we will point out.

3.1. Approximation in time

As in [14], we approximate the single and double layer operators appearing in the $NRBC$ by combining a second order (time) convolution quadrature formula of Lubich (see [33]) with a classical space collocation method. For its 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 $t_n = n\Delta_t$, $n = 0,\ldots,N$:

$$\frac{1}{2}u(\mathbf{x},t_n) + (\boldsymbol{\mathcal{V}}\lambda)(\mathbf{x},t_n) + (\boldsymbol{\mathcal{K}}u)(\mathbf{x},t_n) \;\; = I_{u_0}(\mathbf{x},t_n) + I_{v_0}(\mathbf{x},t_n) + I_f(\mathbf{x},t_n) \tag{9}$$

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 second order Backward Differentiation Method (BDF) for ordinary differential equations (see [12]). We obtain:

$$(\boldsymbol{\mathcal{V}}\lambda)(\mathbf{x},t_n) \approx \sum_{j=0}^{n}\sum_{k=1}^{\boldsymbol{\kappa}}\int_{\mathcal{B}_k}\omega_{n-j}^{\boldsymbol{\mathcal{V}}}(\Delta_t;||\mathbf{x}-\mathbf{y}||)\lambda(\mathbf{y},t_j)\,d\mathcal{B}_{\mathbf{y}},\quad n=0,\ldots,N \tag{10}$$

$$(\boldsymbol{\mathcal{K}}u)(\mathbf{x},t_n) \approx \sum_{j=0}^{n}\sum_{k=1}^{\boldsymbol{\kappa}}\int_{\mathcal{B}_k}\omega_{n-j}^{\boldsymbol{\mathcal{K}}}(\Delta_t;||\mathbf{x}-\mathbf{y}||)u(\mathbf{y},t_j)\,d\mathcal{B}_{\mathbf{y}},\quad n=0,\ldots,N \tag{11}$$

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

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

where

$$K^{\boldsymbol{\mathcal{V}}}(r,s)=\frac{1}{4\pi r}e^{-rs},\qquad K^{\boldsymbol{\mathcal{K}}}(r,s)=-\frac{1}{4\pi r}e^{-rs}\left(\frac{1}{r}+s\right)\frac{\partial r}{\partial\mathbf{n}}. \tag{12}$$

$\gamma(z)=3/2-2z+1/2z^2$ is the so called characteristic quotient of the chosen BDF method of order 2, and the parameter $\rho>0$ is chosen as described below. Note that, contrary to the 2D case, where the corresponding kernels $K^{\mathcal{J}}(r,s)$ have a weak singularity at $s=0$ (see Sect. 3.1 in [13], in particular Remark 3.1), for $r>0$ the above ones are (analytic) entire functions with respect to the variable s. Furthermore, the associated convolution coefficients $\omega_n^{\mathcal{J}}$ have a more favorable behavior (see [34]), which gives rise to more sparse matrices, whose elements actually decay to zero as the time step-size tends to zero (see Sect. 3.4). Also the computation of the matrix elements (16) and (17) below benefits from this behavior.

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

$$\omega_n^{\mathcal{J}}(\Delta_t;r)\approx\frac{\rho^{-n}}{L}\sum_{l=0}^{L-1}K^{\mathcal{J}}\left(r,\frac{\gamma(\rho\exp(il2\pi/L))}{\Delta_t}\right)\exp(-inl2\pi/L). \tag{13}$$

As suggested in ([33]) (see also [14]), we choose $L=2N$ and $\rho=10^{-5}$. This gives a relative accuracy of order $10^{-5}\div 10^{-6}$, which is more than sufficient for our numerical testing. For each given $\mathbf{x}\in\mathcal{B}$, 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.

3.2. Approximation in space

For the space discretization, each surface $\mathcal{B}_k$, of the global artificial boundary $\mathcal{B}$, is approximated by a continuous piecewise triangular surface $\mathcal{B}_{k,\Delta}$, interpolating $\mathcal{B}_k$ at the triangle vertices $\{\mathbf{x}_{k,i}, i=1,\ldots,M_k\}$. We denote by $\Delta_{x,k}$ the mesh size of $\mathcal{B}_{k,\Delta}$, which is given by the maximum triangle diameter.

Further, we set $u_{\mathcal{B}_k}(\mathbf{x},t)=u(\mathbf{x},t)_{|_{\mathcal{B}_k}}$ and $\lambda_{\mathcal{B}_k}(\mathbf{x},t)=\lambda(\mathbf{x},t)_{|_{\mathcal{B}_k}}$, $k=1,\cdots,\boldsymbol{\kappa}$. At each time instant t_j, the unknown function $u_{\mathcal{B}_k}(\cdot,t_j)$ and its normal derivative $\lambda_{\mathcal{B}_k}(\cdot,t_j)$ on $\mathcal{B}_k$ are

approximated by

$$u_{\Delta_{x,k}}(\mathbf{x},t_j) := \sum_{i=1}^{M_k} u_{k,i}^j b_{k,i}(\mathbf{x}), \quad \mathbf{x}\in\mathcal{B}_{k,\Delta} \tag{14}$$

and

$$\lambda_{\Delta_{x,k}}(\mathbf{x},t_j) := \sum_{i=1}^{M_k} \lambda_{k,i}^j b_{k,i}(\mathbf{x}), \quad \mathbf{x}\in\mathcal{B}_{k,\Delta} \tag{15}$$

respectively, where $u_{k,i}^j \approx u_{\mathcal{B}_k}(\mathbf{x}_{k,i},t_j)$, $\lambda_{k,i}^j \approx \lambda_{\mathcal{B}_k}(\mathbf{x}_{k,i},t_j)$, and $\{b_{k,i}\}_{i=1}^{M_k}$ are the classical continuous piecewise linear basis functions associated with the given triangulation.

3.3. Time-space discretization

After having introduced the above time and space discretizations, we collocate the resulting discretized BIE at the (collocation) points $\mathbf{x}_{k,h}$, $h=1,\ldots,M_k$, for each $k=1,\cdots,\boldsymbol{\kappa}$. To write the final system of equations in vectorial notation, we define the matrices

$$(\mathbf{V}_{n-j}^{k,\ell})_{hi} = \int_{\mathcal{B}_\ell} \omega_{n-j}^{\boldsymbol{\mathcal{V}}}(\Delta_t;\|\mathbf{x}_{k,h}-\mathbf{y}\|) b_{\ell,i}(\mathbf{y}) d\mathcal{B}_y, \quad i=1,\cdots,M_\ell \tag{16}$$

$$(\mathbf{K}_{n-j}^{k,\ell})_{hi} = \int_{\mathcal{B}_\ell} \omega_{n-j}^{\boldsymbol{\mathcal{K}}}(\Delta_t;\|\mathbf{x}_{k,h}-\mathbf{y}\|) b_{\ell,i}(\mathbf{y}) d\mathcal{B}_y, \quad i=1,\cdots,M_\ell \tag{17}$$

$\ell=1,\ldots,\boldsymbol{\kappa}$, and the vectors

$$\begin{aligned}
\mathbf{I}_{u_0}^{k,n} &= [I_{u_0}(\mathbf{x}_{k,1},t_n), I_{u_0}(\mathbf{x}_{k,2},t_n),\cdots, I_{u_0}(\mathbf{x}_{k,M_k},t_n)]^T\\
\mathbf{I}_{v_0}^{k,n} &= [I_{v_0}(\mathbf{x}_{k,1},t_n), I_{v_0}(\mathbf{x}_{k,2},t_n),\cdots, I_{v_0}(\mathbf{x}_{k,M_k},t_n)]^T\\
\mathbf{I}_{f}^{k,n} &= [I_{f}(\mathbf{x}_{k,1},t_n), I_{f}(\mathbf{x}_{k,2},t_n),\cdots, I_{f}(\mathbf{x}_{k,M_k},t_n)]^T.
\end{aligned}$$

Then, we introduce the unknown vectors $\mathbf{u}_{\mathcal{B}_\ell}^j = \left[u_{\ell,1}^j,\ldots,u_{\ell,M_\ell}^j\right]^T$ and $\boldsymbol{\lambda}_{\mathcal{B}_\ell}^j = \left[\lambda_{\ell,1}^j,\ldots,\lambda_{\ell,M_\ell}^j\right]^T$, for $\ell=1,\ldots,\boldsymbol{\kappa}$ and $j=0,\ldots,n$, and obtain the following system of equations (see [14] for the case of a single scatterer):

$$\begin{aligned}
\left(\frac{1}{2}\mathbf{I}_k + \mathbf{K}_0^{k,k}\right)\mathbf{u}_{\mathcal{B}_k}^n &+ \sum_{\substack{\ell=1\\ \ell\neq k}}^{\boldsymbol{\kappa}} \mathbf{K}_0^{k,\ell}\mathbf{u}_{\mathcal{B}_\ell}^n + \sum_{\ell=1}^{\boldsymbol{\kappa}}\sum_{j=0}^{n-1} \mathbf{K}_{n-j}^{k,\ell}\mathbf{u}_{\mathcal{B}_\ell}^j + \sum_{\ell=1}^{\boldsymbol{\kappa}} \mathbf{V}_0^{k,\ell}\boldsymbol{\lambda}_{\mathcal{B}_\ell}^n + \sum_{\ell=1}^{\boldsymbol{\kappa}}\sum_{j=0}^{n-1} \mathbf{V}_{n-j}^{k,\ell}\boldsymbol{\lambda}_{\mathcal{B}_\ell}^j\\
&= \mathbf{I}_{u_0}^{k,n} + \mathbf{I}_{v_0}^{k,n} + \mathbf{I}_f^{k,n}, \qquad k=1,\ldots,\boldsymbol{\kappa};\ n=0,\ldots,N
\end{aligned} \tag{18}$$

where the matrix $\mathbf{I}_k$ denotes the identity matrix of order M_k.

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

$$\begin{aligned}
\left(\frac{1}{2}\mathbf{I}_k + \mathbf{K}_0^{k,k}\right)\mathbf{u}_{\mathcal{B}_k}^n &+ \sum_{\substack{\ell=1\\ \ell\neq k}}^{\boldsymbol{\kappa}} \mathbf{K}_0^{k,\ell}\mathbf{u}_{\mathcal{B}_\ell}^n + \sum_{\ell=1}^{\boldsymbol{\kappa}} \mathbf{V}_0^{k,\ell}\boldsymbol{\lambda}_{\mathcal{B}_\ell}^n\\
&= -\sum_{\ell=1}^{\boldsymbol{\kappa}}\sum_{j=0}^{n-1} \mathbf{K}_{n-j}^{k,\ell}\mathbf{u}_{\mathcal{B}_\ell}^j - \sum_{\ell=1}^{\boldsymbol{\kappa}}\sum_{j=0}^{n-1} \mathbf{V}_{n-j}^{k,\ell}\boldsymbol{\lambda}_{\mathcal{B}_\ell}^j + \mathbf{I}_{u_0}^{k,n} + \mathbf{I}_{v_0}^{k,n} + \mathbf{I}_f^{k,n}
\end{aligned} \tag{19}$$

for each $k=1,\ldots,\boldsymbol{\kappa}$.

Remark 3.1. *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 (16), (17), the ω kernel by its discretization (13), and exchanging the integration symbol with that of the quadrature sum (for details see [12]). As described in Example 1 (see Section 6), the computation of the required integrals is performed by using a classical approach based on a ν-point Gauss-Legendre quadrature.*

The evaluation of the volume integrals $\mathbf{I}_{u_0}^{k,n}$, $\mathbf{I}_{v_0}^{k,n}$, $\mathbf{I}_{f}^{k,n}$ has been discussed in [35], where an efficient numerical approach has been proposed for the 2D case and for compactly supported data (see also [12]). For simplicity, in the forthcoming numerical tests we will consider only sources concentrated at a point. This choice extremely simplifies the evaluation of the volume term appearing in the NRBC equation.

3.4. NRBC computational cost and memory space

For simplicity, we consider a single scatterer and the corresponding artificial boundary; the results we obtain can be easily extended to more general situations of multiple scattering. We assume that the chosen Finite Element (FE) grid is regular; furthermore, we denote by N_h the number of its interior points and by $N_{\mathcal{B}}$ that of the points lying on the boundary $\mathcal{B}$. Note that in this case we have $N_{\mathcal{B}} = O(N_h^{2/3})$ as $N_h \to \infty$, i.e., as $\Delta_x \to 0$. Furthermore, as we have already tested in the 2D case, the robustness of the proposed NRBC discretization, and its higher accuracy with respect to that of the associated FEM, may allow a decoupling of the NRBC grid from that of the FEM. That is, one might construct the discretization of the ABC on a subset of the boundary nodes defined by the FE grid. The coupling of the two grids can be performed by a local linear interpolation process.

We further recall that all sums

$$\sum_{j=0}^{n-1} \mathbf{K}_{n-j}^{k,\ell} \mathbf{u}_{\mathcal{B}_\ell}^{j}, \quad \sum_{j=0}^{n-1} \mathbf{V}_{n-j}^{k,\ell} \boldsymbol{\lambda}_{\mathcal{B}_\ell}^{j}, \qquad n = 1, \ldots, N \tag{20}$$

are simultaneously computed by applying a FFT-based algorithm (see [7], Sect. 8.3.1), with a computational cost of $O(N_{\mathcal{B}}^2 N \log N) = O(N_h^{4/3} N \log N)$ flops.

The working space at a first glace appears to be that of $2N_{\mathcal{B}}^2 N$ real numbers; that is, that due to the construction of the above matrices $\mathbf{K}_m, \mathbf{V}_m, m = 0, \ldots, N$. However, this can be significantly reduced. Indeed, as mentioned in Remark 3.1, for each row index, the corresponding row elements of all matrices $\mathbf{K}_m, \mathbf{V}_m, m = 0, \ldots, N$, are simultaneously computed by means of the FFT algorithm. However, before computing, for all matrices, the elements of all rows having the next row index, we set equal to zero those whose size is less than a threshold value; for example, 10^{-5} or 10^{-8}, depending of the final accuracy we want to achieve. Then, we will store, and later use, only the remaining "non zero" row elements. At the end, the total number of elements of each couple of matrices $\mathbf{K}_m, \mathbf{V}_m$ that need to be stored is only a fraction of $2N_{\mathcal{B}}^2$ (see also the following Figures $1 \div 6$). Also the corresponding matrix-vector products will have a computational cost much lower than $N_{\mathcal{B}}^2$.

Some theoretical results, which partially allows us to understand the behavior of the matrices $\mathbf{V}_m$, are reported below. Unfortunately, we cannot derive similar ones for the $\mathbf{K}_m$ matrices, since an explicit representation for the ω-coefficients which define them is not

known. However, the numerical testing we have performed seem to confirm that, as in the 2D case, the behaviors of both matrices are very similar.

Since for the ω-coefficients associated with the $\boldsymbol{\mathcal{V}}$ operator, an explicit analytic representation is known, to verify if the matrices $\mathbf{V}_m$ have some properties, which may be useful to reduce the computational cost of the discretized NRBC, we have obtained the following bounds.

Lemma 3.2. *Let $\Delta_t = T/N$, with $T > 0$ fixed and N arbitrary positive integer. For the ω-coefficients associated with the operator $\boldsymbol{\mathcal{V}}$ the following bounds hold.*

(i) For all integers $1 \le n \le N$ and reals $r > 0$,

$$r|\omega_n(\Delta_t; r)| < 0.05462 \times n^{-\frac{1}{4}} \left(\frac{rN}{nT} e^{-\frac{rN}{nT}+1}\right)^{n/2}. \tag{21}$$

(ii) Let n be a fixed positive integer. For $0 < r_0 \le r \le r_1$, with r_0, r_1 arbitrary real numbers, we have:

$$|\omega_n(\Delta_t; r)| \le C_n \left(N e^{-\alpha_n N}\right)^{n/2}, \tag{22}$$

where

$$C_n = \frac{0.05462}{r_0} n^{-1/4} \left(\frac{r_1 e}{nT}\right)^{n/2}, \qquad \alpha_n = \frac{r_0}{nT}.$$

Proof. In the case of the operator $\boldsymbol{\mathcal{V}}$, the following explicit representation for the ω coefficients has been derived in [25] (see also [34]):

$$\omega_n(\Delta_t; r) = \frac{1}{4\pi r n!} e^{-\frac{3r}{2\Delta_t}} \left(\frac{r}{2\Delta_t}\right)^{n/2} H_n\left(\sqrt{\frac{2r}{\Delta_t}}\right), \ n \ge 0, \tag{23}$$

where $r = \|\mathbf{x} - \mathbf{y}\|, \mathbf{x}, \mathbf{y} \in \mathcal{B}$, and $H_n(x) = 2^n x^n + \dots$ is the n-degree Hermite orthogonal polynomial.

First we consider the case $n = 1 \dots, N$, with $N \to \infty$, that is, $\Delta_t \to 0$. Using the well-known Stirling's formula for the factorial:

$$n! = \sqrt{2\pi} n^{n+\frac{1}{2}} e^{-n+\frac{\theta}{12n}}, \quad 0 < \theta < 1$$

and the bound (see [1], (22.14.17))

$$|H_n(x)| < 1.0865 \times 2^{\frac{n}{2}} e^{\frac{x^2}{2}} \sqrt{n!}$$

from representation (23) we obtain:

$$r|\omega_n(\Delta_t; r)| < \frac{1.0865}{2^{9/4}\pi^{5/4}} n^{-\frac{n}{2}-\frac{1}{4}} e^{\frac{n}{2}} e^{-\frac{1}{2}\frac{rN}{T}} \left(\frac{rN}{T}\right)^{n/2},$$

from which (21) follows.

Then, we consider the case n fixed and $\Delta_t \to 0$. We also assume $0 < r_0 \le r \le r_1$, with r_0, r_1 given real numbers. In this case, bound (22) follows immediately from (21). □

Starting from (21) above, and noting that the function $y = xe^{-x+1}$ is always positive in $(0, \infty)$, is increasing from 0 to its maximum value 1 in the interval $[0, 1]$, and exponentially decaying to zero in $[1, \infty)$, a straightforward calculation then gives the following results.

Corollary 3.3. *Let assume $T > kd_B$, where $0 < r \leq d_B$, d_B denoting the diameter of the artificial boundary $\mathcal{B}$. Then, for all integers n, $N/k \leq n \leq N$, with $k \geq 1$ fixed, we have:*

$$r|\omega_n(\Delta_t; r)| < 0.05462 \times n^{-1/4} \left(\frac{kd_B}{T} e^{-\frac{kd_B}{T}+1} \right)^{\frac{n}{2}}, \tag{24}$$

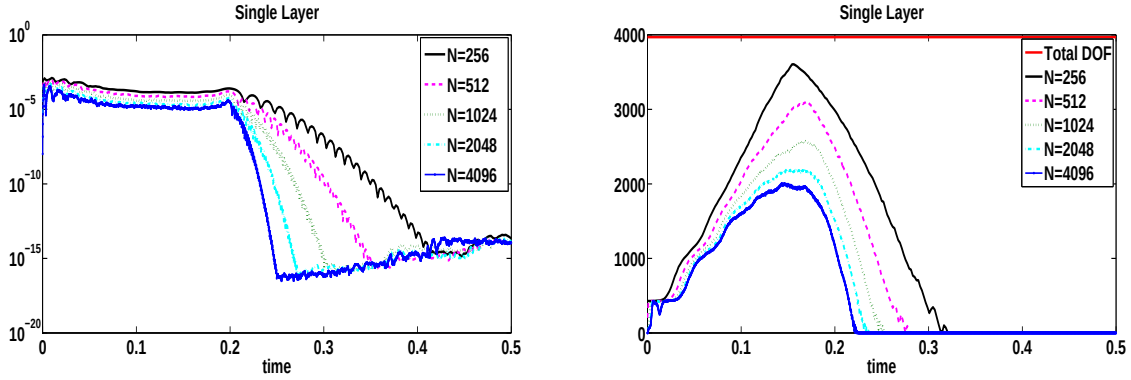
where $0 < \frac{kd_B}{T} e^{-\frac{kd_B}{T}+1} < 1$.

Note that if in bounds (24) we set $x = kd_B/T$, for $0 \leq x \leq 1/4$, i.e. $T \geq 4kd_B$, we have, for example, $y < 0.53$, hence $0.53^{32} < 1.51E-9$ and $0.53^{64} < 2.26E-18$. The smaller is x the smaller is y, hence the faster is the convergence to zero of the associated powers. In particular, for all $r > 0$, the right hand side of (24) tends to zero exponentially, as $N_T \to \infty$. In general, as long as x is away from the abscissa $\bar{x} = 1$, the corresponding value of $y^{n/2}$ decay exponentially to zero, as $N \to \infty$.

Remark 3.4. *From bound (24) it also follows that the larger is T with respect to the diameter of the artificial boundary $\mathcal{B}$, the larger is the value of k that can be taken. Thus for all integers N sufficiently large, all the matrices V_n, $N/k \leq n \leq N$, can be neglected.*

In the following Figures 1-6, we consider an artificial boundary given by a sphere of radius $R = 0.1$, final time instants $T = 0.5, 1, 10$, respectively, and time step T/N. In particular, on the left hand side figures, we report the maximum absolute value of the matrix elements of the matrices $\mathbf{V}_n$ and $\mathbf{K}_n$, for $n = 0, \ldots, N$. On the right hand side figures, we plot, for each n, the number of each matrix element whose size is larger or equal than the threshold value $1E-8$.

Figure 1: Left plot: behavior of the maximum absolute value of the elements of the matrix V_n (Single Layer) versus time. Right plot: number of the non zero entries of the matrix V_n after threshold. $R = 0.1, T = 0.5$



Finally, we recall that in a very general situation, to apply our NRBC we also have to evaluate $3N_{\mathcal{B}}N \log N$ volume integrals (see (19)).

4. Discretization of the boundary value problem

4.1. Time discretization

In principle, any integration scheme can be used for the time discretization of the first equation in (8) (see for example [14], where different time schemes have been considered and

Figure 2: Left plot: behavior of the maximum absolute value of the elements of the matrix K_n (Double Layer) versus time. Right plot: number of the non zero entries of the matrix K_n after threshold. $R = 0.1, T = 0.5$

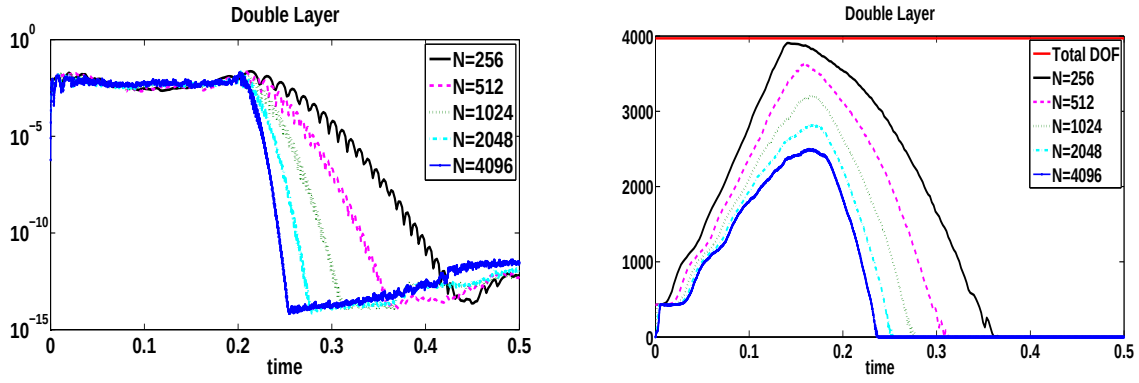


Figure 3: Left plot: behavior of the maximum absolute value of the elements of the matrix V_n (Single Layer) versus time. Right plot: number of the non zero entries of the matrix V_n after threshold. $R = 0.1, T = 1$

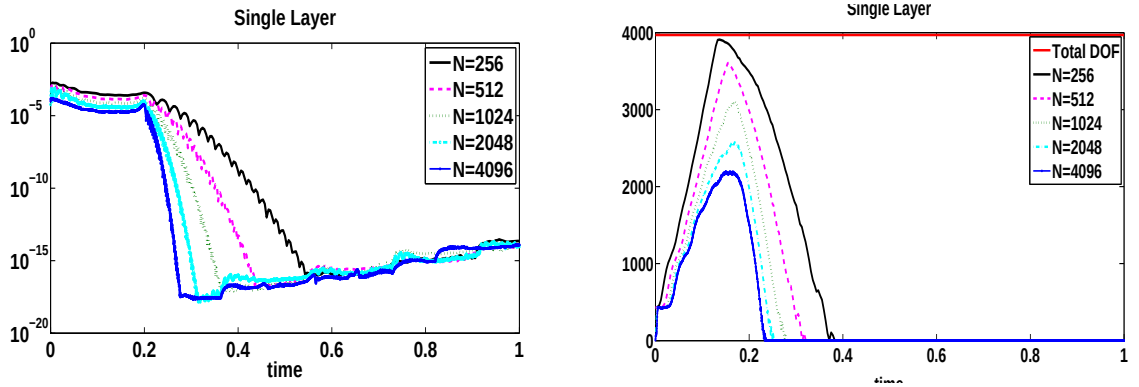
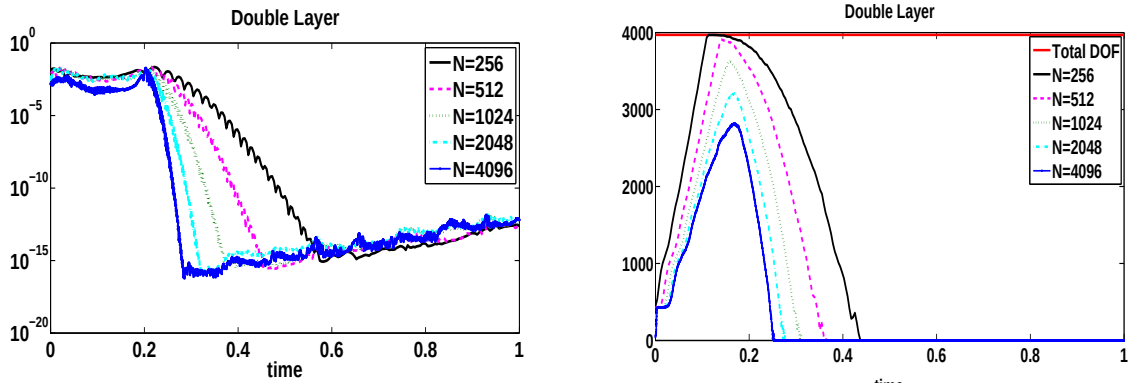


Figure 4: Left plot: behavior of the maximum absolute value of the elements of the matrix K_n (Double Layer) versus time. Right plot: number of the non zero entries of the matrix K_n after threshold. $R = 0.1, T = 1$



compared). For simplicity, we choose to use the Crank-Nicolson integration method, which is of second order and unconditionally stable. Thus, denoting by $v := \frac{\partial u}{\partial t}$ and by $u^n = u^n(x)$, $v^n = v^n(x)$, $\lambda^n = \lambda^n(x)$ and $\tilde{f}^n = \tilde{f}^n(x)$ the approximations of $u(x, t_n)$, $v(x, t_n)$, $\lambda(x, t_n)$ and $\tilde{f}(x, t_n)$, respectively, and applying the Crank-Nicolson discretization, a straightforward

Figure 5: Left plot: behavior of the maximum absolute value of the elements of the matrix V_n (Single Layer) versus time. Right plot: number of the non zero entries of the matrix V_n after threshold. $R = 0.1, T = 10$

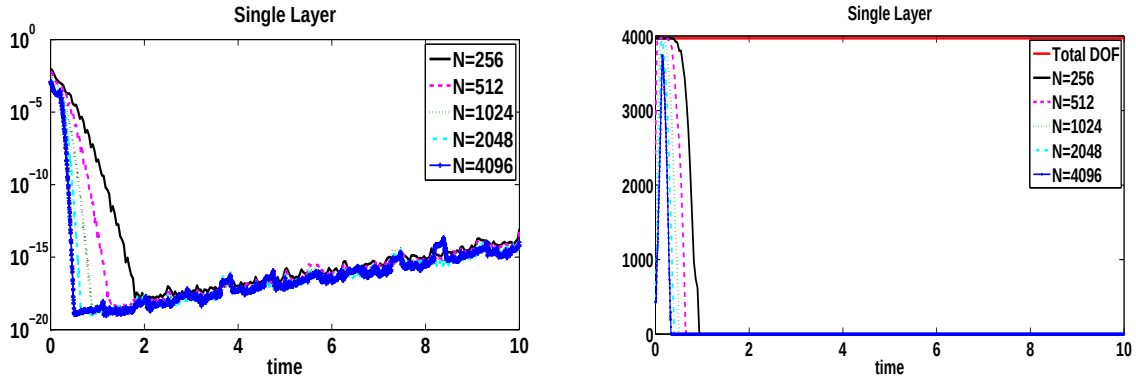
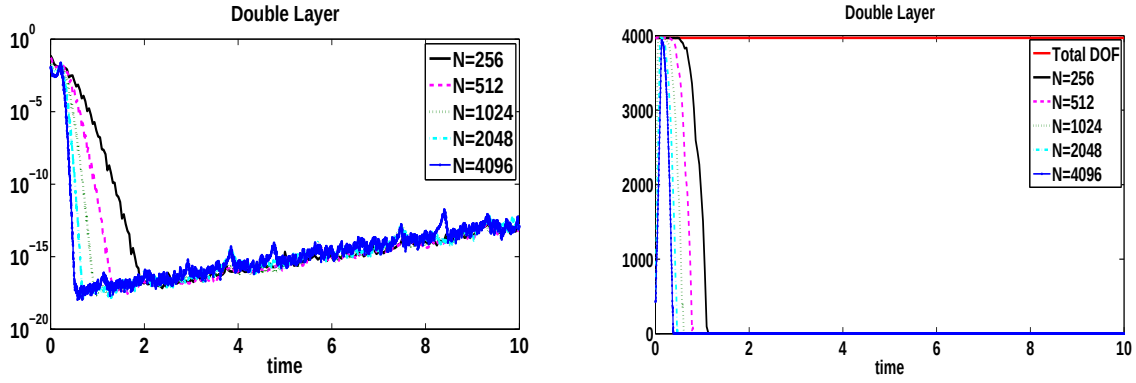


Figure 6: Left plot: behavior of the maximum absolute value of the elements of the matrix K_n (Double Layer) versus time. Right plot: number of the non zero entries of the matrix K_n after threshold. $R = 0.1, T = 10$



calculation leads to the following time-marching scheme (for more details see [14])

$$\begin{cases} (u^{n+1}, w)_\Omega + \frac{\Delta_t^2}{4} a(u^{n+1}, w) - \frac{\Delta_t^2}{4} (\lambda^{n+1}, w)_\mathcal{B} &= (u^n, w)_\Omega - \frac{\Delta_t^2}{4} a(u^n, w) + \frac{\Delta_t^2}{4} (\lambda^n, w)_\mathcal{B} \\ &+ \Delta_t (v^n, w)_\Omega + \frac{\Delta_t^2}{4} (\tilde{f}^{n+1} + \tilde{f}^n, w)_\Omega, \ \forall\, w \in X_0 \\ v^{n+1} = \frac{2}{\Delta_t}(u^{n+1} - u^n) - v^n. \end{cases} \tag{25}$$

4.2. Space discretization

At each time t_n, to compute the unknowns u^n and λ^n we will use a finite element method. To this end, we consider a finite decomposition of each computational domain Ω_k (defined in Section 2) into tetrahedra and we denote by $\Omega_{k,\Delta} = \cup_{\mathcal{T} \in \mathcal{T}_{k,h}} \mathcal{T}$ the finite polyhedral domain, whose mesh size is bounded by h and whose inner and outer boundaries are denoted by $\Gamma_{k,\Delta}$ and $\mathcal{B}_{k,\Delta}$, respectively. Then, we associate with this decomposition the functional spaces

$$X_{k,h} = \{w_{k,h} \in C^0(\Omega_k) : w_{k,h|_\mathcal{T}} \in \mathbb{P}^1(\mathcal{T}), \mathcal{T} \in \mathcal{T}_{k,h}, w_{k,h|_{\Gamma_{k,\Delta}}} = g|_{\Gamma_{k,\Delta}}\} \subset H^1(\Omega_k),$$

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

of (piecewise) linear conforming finite elements in the domain Ω_k associated with the mesh $\mathcal{T}_{k,h}$. We also introduce the space $W_{k,h}$ of (continuous) functions defined on the boundary $\mathcal{B}_k$ by the finite element basis $\{b_{k,i}(x)\}_{i=1}^{M_k}$ (see (14), (15)). Finally, we set

$$X_h = \prod_{k=1}^{\boldsymbol{\kappa}} X_{k,h}, \quad X_{h,0} = \prod_{k=1}^{\boldsymbol{\kappa}} X_{k,h,0}, \quad W_h = \prod_{k=1}^{\boldsymbol{\kappa}} W_{k,h}.$$

The Galerkin formulation of (25) then reads: for each $n = 0, \cdots, N-1$, find $(u_h^{n+1}, \lambda_h^{n+1}) \in X_h \times W_h$ such that, for all $w_h \in X_{h,0}$ we have:

$$\begin{aligned}(u_h^{n+1}, w_h)_\Omega + \frac{\Delta_t^2}{4} a(u_h^{n+1}, w_h) - \frac{\Delta_t^2}{4}(\lambda_h^{n+1}, w_h)_\mathcal{B} = (u_h^n, w_h)_\Omega - \frac{\Delta_t^2}{4} a(u_h^n, w_h) \\ + \frac{\Delta_t^2}{4}(\lambda_h^n, w_h)_\mathcal{B} + \Delta_t (v_h^n, w_h)_\Omega + \frac{\Delta_t^2}{4}(\tilde{f}^{n+1} + \tilde{f}^n, w_h)_\Omega\end{aligned} \tag{26}$$

For every $k = 1, \ldots, \boldsymbol{\kappa}$, let $\{N_i^{\Omega_k}\}_{i \in \mathcal{S}_k}$ denote the set of finite element basis functions defined on the decomposition $\mathcal{T}_{k,h}$, where: $\mathcal{S}_k = \mathcal{S}_{I_k} \cup \mathcal{S}_{\mathcal{B}_k}$, $\mathcal{S}_{I_k}$ is the set of the internal mesh nodes of the polyhedron $\Omega_{k,\Delta}$ and $\mathcal{S}_{\mathcal{B}_k}$ is the set of the mesh nodes lying on the artificial boundary $\mathcal{B}_k$. Note that $b_{k,i} = N_i^{\Omega_k}|_{\mathcal{B}_k}, i \in \mathcal{S}_{\mathcal{B}_k}$. By properly reordering the unknown coefficients of $u_{k,h}^n := u_{h|_{\Omega_k}}^n$, we obtain the (unknown) vector $\mathbf{u}_k^n = [\mathbf{u}_{I_k}^n, \mathbf{u}_{\mathcal{B}_k}^n]^T$, whose two components $\mathbf{u}_{I_k}^n$ and $\mathbf{u}_{\mathcal{B}_k}^n$ represent the unknown values associated with the internal nodes of $\Omega_{k,\Delta}$ and with those on the boundary $\mathcal{B}_k$, respectively. Similarly for the vector $\mathbf{v}_k^n$, containing the unknown coefficients of $v_{k,h}^n$. Finally, we denote by $\boldsymbol{\lambda}_{\mathcal{B}_k}^n$ the unknown vector whose components are the coefficients of the approximant $\lambda_{\triangle_{x,k}}(\mathbf{x}_k, t_n)$ defined in (15).

To rewrite (26) in the matrix form, we consider the system of equations associated with a single computational domain $\Omega_{k,\Delta}$, $k = 1, \ldots, \boldsymbol{\kappa}$, which is given by

$$\begin{aligned}\left(\mathbf{M}^k + \frac{\Delta_t^2}{4}\mathbf{A}^k\right)\mathbf{u}_k^{n+1} - \frac{\Delta_t^2}{4}\mathbf{Q}^k \boldsymbol{\lambda}_{\mathcal{B}_k}^{n+1} &= \left(\mathbf{M}^k - \frac{\Delta_t^2}{4}\mathbf{A}^k\right)\mathbf{u}_k^n + \frac{\Delta_t^2}{4}\mathbf{Q}^k\boldsymbol{\lambda}_{\mathcal{B}_k}^n + \Delta_t \mathbf{M}^k \mathbf{v}_k^n \\ &\quad + \frac{\Delta_t^2}{4}\left(\tilde{\mathbf{f}}_k^{n+1} + \tilde{\mathbf{f}}_k^n\right)\end{aligned} \tag{27}$$

where

$$\mathbf{M^k} = \begin{bmatrix} M_{II}^k & M_{I\mathcal{B}}^k \\ M_{\mathcal{B}I}^k & M_{\mathcal{B}\mathcal{B}}^k \end{bmatrix}, \qquad \mathbf{A^k} = \begin{bmatrix} A_{II}^k & A_{I\mathcal{B}}^k \\ A_{\mathcal{B}I}^k & A_{\mathcal{B}\mathcal{B}}^k \end{bmatrix}, \qquad \mathbf{Q^k} = \begin{bmatrix} Q_{I\mathcal{B}}^k \\ Q_{\mathcal{B}\mathcal{B}}^k \end{bmatrix}.$$

The matrix elements

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

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

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

Equation (27) is finally coupled with

$$\mathbf{v}_k^{n+1} = \frac{2}{\Delta_t}(\mathbf{u}_k^{n+1} - \mathbf{u}_k^n) - \mathbf{v}_k^n \tag{28}$$

and with the discretized NRBC equation

$$\begin{aligned}\left(\frac{1}{2}\mathbf{I}_k + \mathbf{K}_0^{k,k}\right)\mathbf{u}_{\mathcal{B}_k}^{n+1} \quad + \quad & \sum_{\substack{\ell=1\\ \ell\neq k}}^{\boldsymbol{\kappa}} \mathbf{K}_0^{k,\ell}\mathbf{u}_{\mathcal{B}_\ell}^{n+1} + \sum_{\ell=1}^{\boldsymbol{\kappa}} \mathbf{V}_0^{k,\ell}\boldsymbol{\lambda}_{\mathcal{B}_\ell}^{n+1} \\ & = -\sum_{\ell=1}^{\boldsymbol{\kappa}}\sum_{j=0}^{n} \mathbf{K}_{n+1-j}^{k,\ell}\mathbf{u}_{\mathcal{B}_\ell}^{j} - \sum_{\ell=1}^{\boldsymbol{\kappa}}\sum_{j=0}^{n} \mathbf{V}_{n+1-j}^{k,\ell}\boldsymbol{\lambda}_{\mathcal{B}_\ell}^{j} + \mathbf{I}_{u_0}^{k,n+1} + \mathbf{I}_{v_0}^{k,n+1} + \mathbf{I}_f^{k,n+1}.\end{aligned}$$

Having set $\mu = \frac{\Delta_t^2}{4}$, and letting $k = 1, \ldots, \boldsymbol{\kappa}$, we obtain a final linear system $\mathbb{A}\mathbb{X} = \mathbb{B}$ having the following block structure:

$$\mathbb{A} = \begin{pmatrix} \begin{bmatrix} \mathbf{M^1} + \mu\mathbf{A^1} & -\mu\mathbf{Q}^1 \\ \frac{1}{2}\mathbf{I}_1 + \mathbf{K}_0^{1,1} & \mathbf{V}_0^{1,1} \end{bmatrix} & \begin{bmatrix} \mathbf{O} & \mathbf{O} \\ \mathbf{K}_0^{1,2} & \mathbf{V}_0^{1,2} \end{bmatrix} & \cdots & \begin{bmatrix} \mathbf{O} & \mathbf{O} \\ \mathbf{K}_0^{1,\boldsymbol{\kappa}} & \mathbf{V}_0^{1,\boldsymbol{\kappa}} \end{bmatrix} \\ \begin{bmatrix} \mathbf{O} & \mathbf{O} \\ \mathbf{K}_0^{2,1} & \mathbf{V}_0^{2,1} \end{bmatrix} & \begin{bmatrix} \mathbf{M^2} + \mu\mathbf{A^2} & -\mu\mathbf{Q}^2 \\ \frac{1}{2}\mathbf{I}_2 + \mathbf{K}_0^{2,2} & \mathbf{V}_0^{2,2} \end{bmatrix} & \cdots & \begin{bmatrix} \mathbf{O} & \mathbf{O} \\ \mathbf{K}_0^{2,\boldsymbol{\kappa}} & \mathbf{V}_0^{2,\boldsymbol{\kappa}} \end{bmatrix} \\ \vdots & \vdots & \ddots & \vdots \\ \begin{bmatrix} \mathbf{O} & \mathbf{O} \\ \mathbf{K}_0^{\boldsymbol{\kappa},1} & \mathbf{V}_0^{\boldsymbol{\kappa},1} \end{bmatrix} & \cdots & & \begin{bmatrix} \mathbf{M}^{\boldsymbol{\kappa}} + \mu\mathbf{A}^{\boldsymbol{\kappa}} & -\mu\mathbf{Q}^{\boldsymbol{\kappa}} \\ \frac{1}{2}\mathbf{I}_{\boldsymbol{\kappa}} + \mathbf{K}_0^{\boldsymbol{\kappa},\boldsymbol{\kappa}} & \mathbf{V}_0^{\boldsymbol{\kappa},\boldsymbol{\kappa}} \end{bmatrix} \end{pmatrix}$$

$$\mathbb{X} = \begin{pmatrix} \begin{bmatrix} \mathbf{u}_1^{n+1} \\ \boldsymbol{\lambda}_{\mathcal{B}_1}^{n+1} \end{bmatrix} \\ \begin{bmatrix} \mathbf{u}_2^{n+1} \\ \boldsymbol{\lambda}_{\mathcal{B}_2}^{n+1} \end{bmatrix} \\ \vdots \\ \begin{bmatrix} \mathbf{u}_{\boldsymbol{\kappa}}^{n+1} \\ \boldsymbol{\lambda}_{\mathcal{B}_{\boldsymbol{\kappa}}}^{n+1} \end{bmatrix} \end{pmatrix}$$

$$\mathbb{B} = \begin{pmatrix} \begin{bmatrix} (\mathbf{M}^1 - \mu\mathbf{A}^1)\,\mathbf{u}_1^n + \mu\mathbf{Q}^1\boldsymbol{\lambda}_1^n + \Delta_t\mathbf{M}^1\mathbf{v}_1^n + \mu\left(\tilde{\mathbf{f}}_1^{n+1} + \tilde{\mathbf{f}}_1^n\right) \\ -\sum_{\ell=1}^{\boldsymbol{\kappa}}\sum_{j=0}^{n}\mathbf{K}_{n+1-j}^{1,\ell}\mathbf{u}_{\mathcal{B}_\ell}^j - \sum_{\ell=1}^{\boldsymbol{\kappa}}\sum_{j=0}^{n}\mathbf{V}_{n+1-j}^{1,\ell}\boldsymbol{\lambda}_{\mathcal{B}_\ell}^j + \mathbf{I}_{u_0}^{1,n+1} + \mathbf{I}_{v_0}^{1,n+1} + \mathbf{I}_f^{1,n+1} \end{bmatrix} \\ \begin{bmatrix} (\mathbf{M}^2 - \mu\mathbf{A}^2)\,\mathbf{u}_2^n + \mu\mathbf{Q}^2\boldsymbol{\lambda}_2^n + \Delta_t\mathbf{M}^2\mathbf{v}_2^n + \mu\left(\tilde{\mathbf{f}}_2^{n+1} + \tilde{\mathbf{f}}_2^n\right) \\ -\sum_{\ell=1}^{\boldsymbol{\kappa}}\sum_{j=0}^{n}\mathbf{K}_{n+1-j}^{2,\ell}\mathbf{u}_{\mathcal{B}_\ell}^j - \sum_{\ell=1}^{\boldsymbol{\kappa}}\sum_{j=0}^{n}\mathbf{V}_{n+1-j}^{2,\ell}\boldsymbol{\lambda}_{\mathcal{B}_\ell}^j + \mathbf{I}_{u_0}^{2,n+1} + \mathbf{I}_{v_0}^{2,n+1} + \mathbf{I}_f^{2,n+1} \end{bmatrix} \\ \vdots \\ \begin{bmatrix} (\mathbf{M}^{\boldsymbol{\kappa}} - \mu\mathbf{A}^{\boldsymbol{\kappa}})\,\mathbf{u}_{\boldsymbol{\kappa}}^n + \mu\mathbf{Q}^{\boldsymbol{\kappa}}\boldsymbol{\lambda}_{\boldsymbol{\kappa}}^n + \Delta_t\mathbf{M}^{\boldsymbol{\kappa}}\mathbf{v}_{\boldsymbol{\kappa}}^n + \mu\left(\tilde{\mathbf{f}}_{\boldsymbol{\kappa}}^{n+1} + \tilde{\mathbf{f}}_{\boldsymbol{\kappa}}^n\right) \\ -\sum_{\ell=1}^{\boldsymbol{\kappa}}\sum_{j=0}^{n}\mathbf{K}_{n+1-j}^{\boldsymbol{\kappa},\ell}\mathbf{u}_{\mathcal{B}_\ell}^j - \sum_{\ell=1}^{\boldsymbol{\kappa}}\sum_{j=0}^{n}\mathbf{V}_{n+1-j}^{\boldsymbol{\kappa},\ell}\boldsymbol{\lambda}_{\mathcal{B}_\ell}^j + \mathbf{I}_{u_0}^{\boldsymbol{\kappa},n+1} + \mathbf{I}_{v_0}^{\boldsymbol{\kappa},n+1} + \mathbf{I}_f^{\boldsymbol{\kappa},n+1} \end{bmatrix} \end{pmatrix}$$

To discretize, hence solve, our problem (7) we have followed the same approach used in [14] for a single scatterer in the 2D case. However, in the 3D case under consideration, having $\boldsymbol{\kappa}$ scatterers, we have imposed $\boldsymbol{\kappa}$ NRBC. The efficient discretization of these latter is then a key issue for the success of the overall method, both in terms of space memory requirement and computational time.

Note that the matrix $\mathbb{A}$ does not depend on n. Therefore, at each time step only the right hand side $\mathbb{B}$ needs to be updated.

Remark 4.1. *We have tested the stability and convergence of the proposed method. From the results we have obtained, it appears that unconditional stability and convergence is guaranteed, in the given time interval* $[0, T]$*, as long as all the integrals required by the (discretized) NRBC are exactly evaluated. When these integrals are evaluated with a given accuracy* ϵ*, then, for any chosen space discretization parameter* Δ_x*, there exists a time step barrier* $\Delta_0 = \Delta_0(\epsilon)$*, with* $\Delta_0 \to 0$ *as* $\epsilon \to 0$*, such that for* $\Delta_t < \Delta_0$ *instabilities arise before reaching the final time* T*.*

For example, if in the first example of Section 6, where $T = 10$*, we choose* $N = 32, 64, 128, 256, 512, 1024$ *and consider the first two levels of space discretization, no instability appears when we perform the required integration by using a* ν*-point Gauss-Legendre rule, with* $\nu = 2, 2, 2, 8, 12, 20$ *in the case of the first discretization level, and* $\nu = 2, 2, 2, 2, 8, 12$ *in the case of the second level. In both cases, for example, if we take* $\nu = 4$ *when* $N = 512, 1024$*, instabilities appear before reaching the endpoint* $T = 10$*. Thus, the efficient evaluation of the above mentioned integrals is a crucial point for the success of the proposed NRBC. Note however that, as* Δ_x *decreases, the accuracy given by the chosen* ν*-point rule increases, hence the value of* Δ_0 *decreases.*

5. An exact solution for a wave equation problem in the exterior of the unit sphere

To test the convergence of the proposed numerical scheme, and in particular the effectiveness of the proposed NRBC, it is important to have a reference solution at hand.

In the special case of an homogeneous problem ($u_0 = v_0 = f = 0$) defined on the surface $\Gamma = \mathbb{S}^2$ of the unit sphere, for the particular Dirichlet datum $g(\mathbf{x}, t) = g(t)Y_n^m(\mathbf{x})$, where Y_n^m are the spherical harmonics of degree n and order m, with $n = 0, 1, \ldots$ and $-n \leq m \leq n$, it is possible to derive an explicit analytic representation for the solution of problem (1).

To this end, by proceeding as the authors of [36] did for an analogous question (the exact solution of the associated BIE defined below in (34)), we consider the Laplace transform of $u(\mathbf{x}, t)$), here denoted by $\hat{u}(\mathbf{x}, s)$, which is the solution of the wave equation reformulated in the frequency domain (Helmholtz equation). Furthermore, we set

$$u(\mathbf{x}, t) = u(t)Y_n^m(\xi), \quad \text{with} \quad \mathbf{x} = r\xi \in \mathbb{R}^3,\ \xi \in \Gamma = \mathbb{S}^2 \tag{29}$$

where, for notational simplicity, we denote by $u(t)$ a function which depends also on r. Then, we use the property that the solution $\hat{u}$ of the associated Helmholtz problem satisfies the relationship

$$\hat{u}(s) = \hat{g}(s)\frac{h_n^{(1)}(isr)}{h_n^{(1)}(is)}, \tag{30}$$

where $h_n^{(1)}$ denotes the spherical Bessel functions of third kind (see [8]). By computing the inverse Laplace transform of (30), we retrieve an analytic expression for the solution of the homogeneous wave equation. This is given by representation (29), with

$$u(t) = \mathcal{L}^{-1}\left(\hat{g}\frac{h_n^{(1)}(ir\cdot)}{h_n^{(1)}(i\cdot)}\right)(t) = g * \mathcal{L}^{-1}\left(\frac{h_n^{(1)}(ir\cdot)}{h_n^{(1)}(i\cdot)}\right)(t).$$

By using known properties of the Bessel functions (see in particular [1] (Sec. 10.1.1) and [22] (formula in Sec. 8.466)), and following [36], we easily get

$$\begin{aligned} h_n^{(1)}(s) &= \sqrt{\frac{\pi}{2s}}H_{n+\frac{1}{2}}^{(1)}(s) = \sqrt{\frac{\pi}{2s}}\sqrt{\frac{2}{\pi s}}(i)^{-(n+1)}e^{is}\sum_{k=0}^{n}(-1)^k\frac{(n+k)!}{k!(n-k)!}\frac{1}{(2is)^k} \\ &= \frac{1}{s}(i)^{-(n+1)}e^{is}y_n\left(-\frac{1}{is}\right), \end{aligned}$$

where $y_n(s) := \sum_{k=0}^{n}(n,k)s^k$, $(n,k) := \frac{(n+k)!}{2^k k!(n-k)!}$ and $H_\nu^{(1)}$ is the first kind Hankel function of order ν. Using this latter expression we obtain:

$$\frac{h_n^{(1)}(irs)}{h_n^{(1)}(is)} = \frac{1}{r}e^{-(r-1)s}\frac{y_n(\frac{1}{rs})}{y_n(\frac{1}{s})}.$$

In the simplest case $n = 0$ we have $y_n(s) = 1$, wherefrom $\frac{h_0^{(1)}(irs)}{h_0^{(1)}(is)} = \frac{1}{r}e^{-(r-1)s}$. By using classical properties of the inverse Laplace transform, we easily get

$$\mathcal{L}^{-1}\left(\frac{h_0^{(1)}(ir\cdot)}{h_0^{(1)}(i\cdot)}\right)(t) = \frac{1}{r}H(t-(r-1))\delta(t-(r-1)),$$

being H the Heaviside function. Thus, we obtain

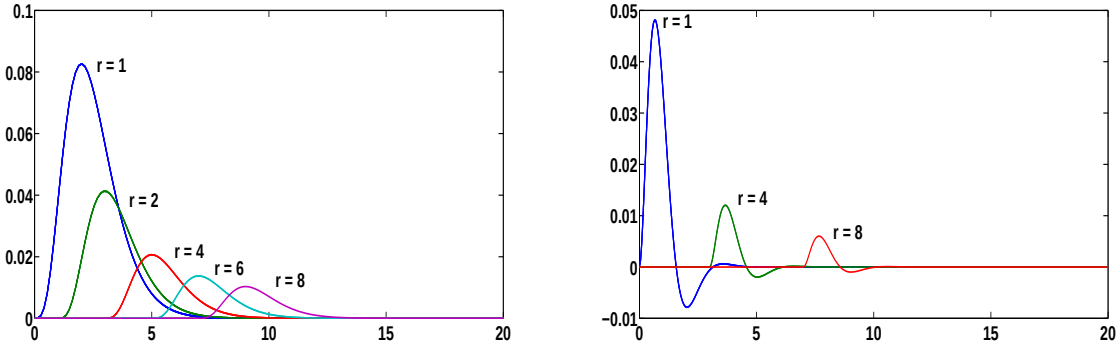
$$\begin{aligned} u(t) &= \frac{1}{r}\int_0^t g(t-\tau)H(\tau-(r-1))\delta(\tau-(r-1))d\tau \\ &= \begin{cases} 0 & \text{if} \quad t \le r-1 \\ \frac{1}{r}g(t-(r-1)) & \text{otherwise.} \end{cases} \end{aligned} \tag{31}$$

Since $Y_0^0(\mathbf{x}) = 1/(2\sqrt{\pi})$, the solution of our problem is:

$$u(\mathbf{x},t) = \frac{1}{2\sqrt{\pi}}u(t) \tag{32}$$

with $u(t)$ defined by (31). Note that this latter is a radial function. In Figure 7 we show the behavior of $u(\mathbf{x},t)$, with respect to the time variable, corresponding to the Dirichlet data $g(\mathbf{x},t) = \frac{1}{2\sqrt{\pi}}t^4e^{-2t}$ (left plot) and $g(\mathbf{x},t) = \frac{1}{2\sqrt{\pi}}\sin(2t)te^{-2t}$ (right plot).

Figure 7: Exact solutions of the homogeneous wave equation for $\Gamma = \mathbb{S}^2$ corresponding to the choice of the Dirichlet data $g(\mathbf{x},t) = \frac{1}{2\sqrt{\pi}}t^4e^{-2t}$ (left plot) and $g(\mathbf{x},t) = \frac{1}{2\sqrt{\pi}}\sin(2t)te^{-2t}$ (right plot). Each curve corresponds to its behavior with respect to time at points having distance r from the origin.



6. Numerical results

In the numerical tests we will perform, the discretization of the three dimensional spatial domain is generated by using the open source Freefem++ library. In particular we use the TetGen software, which allows to generate the tetrahedral mesh of the Ω domain starting from the triangular meshes we define on its two boundaries (see [28]). The computational domains that we will consider for our numerical examples are three dimensional shells bounded internally by a surface Γ and externally by a surface $\mathcal{B}$. The model that represents the shell, which is constructed by TetGen, is a three dimensional Piecewise Linear Complex (PLC). We recall that TetGen does not generate the surface mesh of the PLC; this, together with the choice of the maximum diameter of the triangles generated to approximate the surfaces, must be given as input by the user. Moreover, TetGen does not take the curvature of the surface into account. In what follows, n_T denotes the number of tetrahedra of the decomposition of the computational domain Ω, while nt_Γ and $nt_\mathcal{B}$ denote the number of triangles belonging to the boundaries Γ and $\mathcal{B}$, respectively.

To define a discretization with the TetGen sofware we have downloaded, we first set, on each couple of boundary surfaces $\mathcal{B}$ and Γ, the "sizes" of the corresponding triangulations; these sizes are denoted by $h_{\mathcal{B}}$ and h_Γ, respectively. Starting from these data, TetGen then constructs the associated Ω discretization. Halving $h_{\mathcal{B}}$ and h_Γ does not necessarily implies halving the Ω discretization maximum diameter. Since the discretizations obtained by proceeding as described above were not satisfactory, we have modified them in the central part of Ω, to make them more uniform.

For simplicity, all the problems we will consider have trivial initial data u_0 and v_0, since the evaluation of the corresponding integrals would require a further ad hoc discussion. Unfortunately, we are not aware of published numerical results obtained by solving "true" 3D scattering problems using a FEM+NRBC approach alternative to ours. Those we have found in the literature refer to special cases, such as those of spherical domains and plane waves (see for example [24]), that are first reduced to 2D problems and then solved as such. Therefore, we will not compare the results we obtain using our method with those that could be produced by alternative approaches.

In the first example, we consider a problem whose solution is given by the representation derived in Sect. 5. In the following four examples, since the corresponding exact solutions are not known, to measure the accuracy of the approximations we obtain, we construct a reference "exact" solution obtained by using the following single-layer potential representation associated to (1) (see [12]):

$$u(\mathbf{x},t) = \int_0^t \int_\Gamma G(||\mathbf{x}-\mathbf{y}||, t-\tau)\varphi(\mathbf{y},\tau)\, d\Gamma_{\mathbf{y}}\, d\tau + I_{u_0}(\mathbf{x},t) + I_{v_0}(\mathbf{x},t) + I_f(\mathbf{x},t) \quad \mathbf{x}\in\Omega^e \quad (33)$$

The unknown density function φ is the solution of the following BIE:

$$\int_\Gamma \int_0^t G(r, t-\tau)\varphi(\mathbf{y},\tau)\, d\tau d\Gamma_{\mathbf{y}} = g(\mathbf{x},t) - I_{u_0}(\mathbf{x},t) - I_{v_0}(\mathbf{x},t) - I_f(\mathbf{x},t), \quad \mathbf{x}\in\Gamma, \quad (34)$$

and represents the jump of the normal derivative of u along Γ. The density function is obtained by applying the Lubich-collocation method to equation (34), with a very fine space and time discretization. The solution at any point in the infinite domain Ω^e is then retrieved by computing the associated potential integral representation (33), using the previous discretization. In the following, this solution will be denoted by the acronym BEM.

Since the BEM approach is itself an alternative option for solving problem (1), before proceeding further, we compare it with our $FEM{+}NRBC$ method, both from the computational complexity and the storage requirement.

As in Sect. 3.4, for simplicity we consider the case of a single scatterer with the corresponding artificial boundary. We denote by N_Γ the number of mesh points on the boundary Γ. Following the observations made in the first part of Sect. 3.4, and using the same symbols, the numerical solution of BIE (34), by means of the method proposed in [12], requires the evaluation of $N_\Gamma^2 N \log N$ matrix elements and, in general, of $3N_\Gamma N \log N$ volume integrals. The associated working space is that of $N_\Gamma^2 N$ real numbers. The evaluation of the solution $u(\mathbf{x},t)$ at all N_h interior grid points, as well as at the $N_{\mathcal{B}}$ artificial boundary mesh points, for all N time instants, requires the computation of $(N_h + N_{\mathcal{B}})N_\Gamma N \log N$ matrix elements, plus $3(N_h + N_{\mathcal{B}})N \log N$ volume integrals. Thus the total computational cost of the BEM

approach is given by the evaluation of $(N_h + N_\Gamma + N_\mathcal{B})N_\Gamma N \log N$ matrix elements and, in a very general situation, of $3(N_h + N_\Gamma + N_\mathcal{B})N \log N$ volume integrals.

If we ignore, for simplicity, the volume integral computation, the ratio between the BEM and FEM+NBBC computational complexities is of order $\frac{N_\Gamma(N_h+N_\Gamma+N_\mathcal{B})}{N_\mathcal{B}^2}$. Note also that in the BEM approach, the computational cost of the corresponding volume integral evaluation is much higher than that of the volume integrals required by the FEM+NRBC method (see the last sentence of Sect. 3.4); their ratio is of order $\frac{N_h+N_\Gamma+N_\mathcal{B}}{N_\mathcal{B}}$. Thus, since in the case of a regular grid we have $N_\Gamma, N_\mathcal{B} = O(N_h^{2/3})$, the behavior of these two ratios, as $N_h \to \infty$, is $O(N_h^{1/3})$ for both of them. The ratio between the working space required by the two approaches is $\left(\frac{N_\Gamma}{N_\mathcal{B}}\right)^2$.

Example 1. As a first example, we consider Problem (1) in the case where Ω^i consists of a single scatterer ($\boldsymbol{\kappa} = 1$). We assume that the source f is zero throughout the infinite exterior domain Ω^e. The boundary Γ is the unit sphere, where we prescribe the Dirichlet condition $g(\mathbf{x}, t) = \frac{1}{2\sqrt{\pi}} t^4 e^{-2t}$ for all $t \geq 0$. The solution of this problem is a radial function and its analytical expression is given by (32) and (31).

Case a). First we choose a spherical artificial boundary $\mathcal{B}$ having radius $R = 2$, so that Ω is the shell bounded internally by Γ and externally by $\mathcal{B}$. In Figure 8 we show a section of three different refinements of the shell.

Here, and in the following examples, to evaluate the integrals that appear in the elements of the matrices $\mathbf{V}_\mathbf{m}^{\mathbf{k},\ell}$ and $\mathbf{K}_\mathbf{m}^{\mathbf{k},\ell}$ (see (16) and (17)), we first map each triangle of the approximated artificial surface, where the integrand in non null, into the (standard) reference triangle; then we introduce the polar coordinates and apply a ν-point Gauss-Legendre quadrature rule to each one-dimensional integral. In this first example we take $\nu = 4$. We remark that, because of the discrepancy between the artificial boundary $\mathcal{B}_\Delta$ of the PLC Ω_Δ and that ($\mathcal{B}$) of the $NRBC$, and of the presence of the normal derivative in the kernel $K^{\boldsymbol{\kappa}}$ (see (12)), the solution obtained by integrating over the plain triangles of $\mathcal{B}_\Delta$ is not satisfactory for coarse spatial discretizations and presents spurious reflections that disappear with spatial refinements. Therefore, we perform the integration over the curvilinear triangles by simply projecting a point belonging to the plane triangle to the corresponding point of the surface (see for example Figures 18 and 19 where we compare both types of approximation).

Being the exact solution known, we also compute the following error norm:

$$\mathrm{Err} := \max_{0 \leq n \leq N} \|u(\cdot, t_n) - u_{\Delta_x}(\cdot, t_n)\|_{L^2(\Omega)}. \tag{35}$$

The numerical computation of the quantity

$$\|u(\cdot, t_n) - u_{\Delta_x}(\cdot, t_n)\|^2_{L^2(\Omega)} \approx \sum_{\mathcal{T} \in \mathcal{T}_h} \int_\mathcal{T} |u(\mathbf{x}, t_n) - u_{\Delta_x}(\mathbf{x}, t_n)|^2 d\mathbf{x}$$

has been obtained by applying on each tetrahedron $\mathcal{T}$ a 4-point quadrature rule of order 2 (see [30] for details).

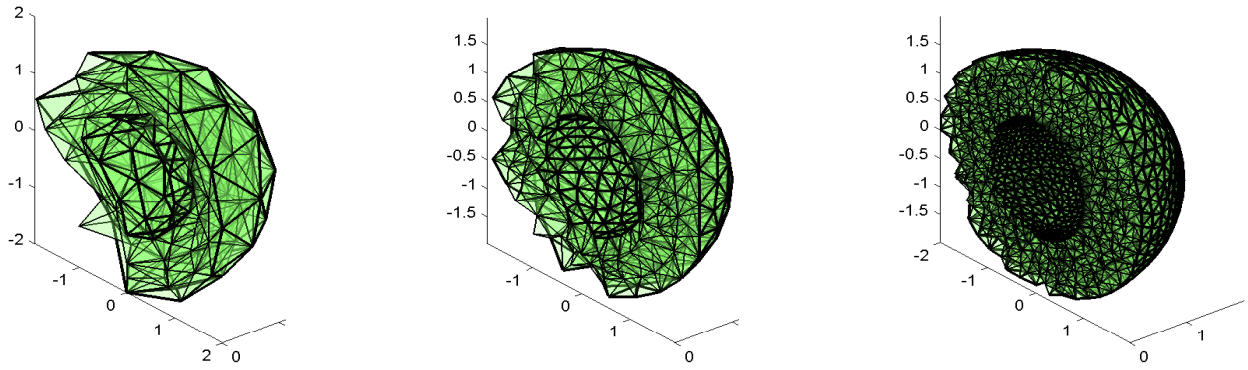
We must remark that, not knowing the maximum diameter reduction produced by halving the sizes of the above defined surfaces triangulations, it becomes impossible to test the

convergence rate of our method. Nevertheless, we believe that the results reported in the two tables below can give some useful information.

In Table 1 we report the error norm defined in (35) with respect to some space and time mesh refinements. The successive spatial refinements have been obtained by halving each time the mesh size of both the internal (h_Γ) and external ($h_\mathcal{B}$) boundaries of the shell. The associated refinements of the Ω decomposition are automatically determined by the software TetGen. It is our opinion that the above errors are essentially due to the FE computation. Those due to the NRBC discretization appear negligible. Indeed, if we compute, for example, all the needed integrals with higher accuracy, using a 16-point Gauss-Legendre rule, we obtain the same errors reported in Table 1. As we have already noted in the 2D case, the NRBC is very robust (see the comment we have made at the end of the first paragraph in Sect. 3.4).

Figure 8: Example 1. Case a). A section of the shell having the unit sphere as internal boundary and the sphere of radius 2 as external one: three discretizations into n_T tetrahedra.

$n_T = 1627$ $n_T = 6079$ $n_T = 44892$



We have also computed the problem solution, at three of the grid points produced by TetGen, by using the BEM approach described above, with the same FEM mesh. Since the three grids we have consider do not have interior common points, the tree points P_1, P_2, P_3 we have taken for each grid are those having minimum distance from $Q_1 = (2, 0, 0), Q_2 = (1.5, 0, 0), Q_3 = (1.2, 0, 0)$. The associated errors are also reported for a comparison.

For each chosen discretization level $L1$-$L3$, and $t \in [0, 10]$, in the left plots of Figures 9, 10 and 11 we compare the exact solution with the approximations we have obtained at the grid points P_1, P_2 and P_3 defined above. On the right plots of the same figures we report the associated absolute errors.

In Table 2, for the same mesh refinements of Table 1, we compare the absolute errors given by our method and by the BEM, at the grid points P_1, P_2 and P_3. The label *npoints* denotes the total number of degrees of freedom (internal and $\mathcal{B}$-boundary points). In the last column of Table 2 we report the ratio between the CPU times required by the BEM and by the FEM+NRBC approaches, to determine the solution at all the above defined *npoints*, and for all chosen time instants t_n, $n = 1, \ldots, N$. As expected, the proposed method is

Figure 9: Example 1. Case a). Comparison between the exact solution and the approximate ones at P_1 (left plot) and associated errors (right plot), for the space and time mesh refinements defined in Table 1.

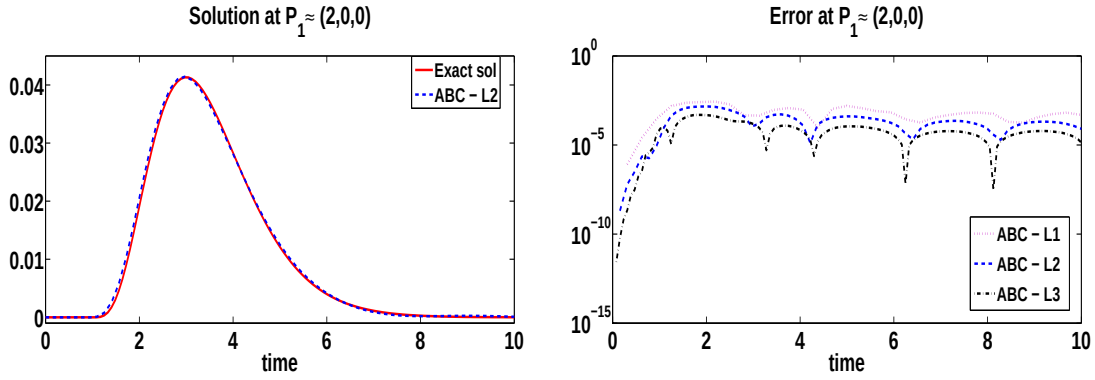


Figure 10: Example 1. Case a). Comparison between the exact solution and the approximate ones at P_2 (left plot) and associated errors (right plot), for the space and time mesh refinements defined in Table 1.

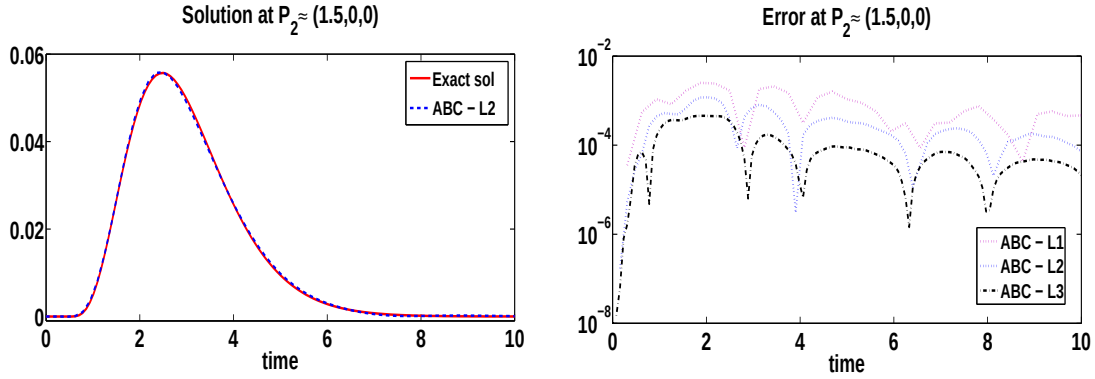
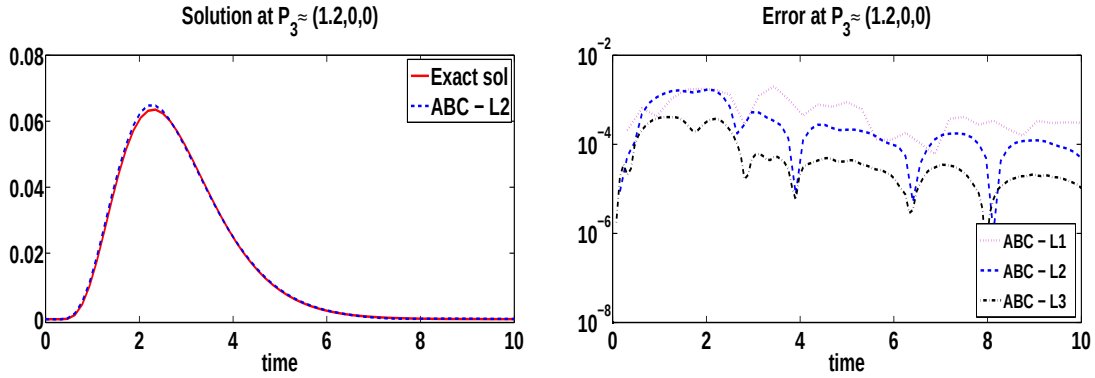


Figure 11: Example 1. Case a). Comparison between the exact solution and the approximate ones at P_3 (left plot) and associated errors (right plot), for the space and time mesh refinements defined in Table 1.



much faster.

As already remarked, the choice of the artificial boundary $\mathcal{B}$ should be flexible and should be dictated by the area where one is interested in knowing the solution, rather than by the property of the ABC of being more or less transparent according to where $\mathcal{B}$ is located. It is well known, for example that local ABCs of low order perform better when $\mathcal{B}$ is located

Table 1: Example 1. Case a). Behavior of the error norm (35) with respect to the space and time mesh refinements, for $t \in [0, 10]$.

level	h_Γ	nt_Γ	$h_\mathcal{B}$	$nt_\mathcal{B}$	n_T	N	$\mathrm{Err}_{\mathrm{FEM}}$
$L1$	.5	122	1	122	1627	32	9.70E-03
$L2$	.25	440	.5	440	6079	64	5.75E-03
$L3$	.125	1772	.25	1772	44892	128	2.63E-03

Table 2: Example 1. Case a). Absolute errors at P_1, P_2, P_3 and FEM+NRBC vs BEM CPU comparison, for the space and time mesh refinements defined in Table 1, $t \in [0, 10]$.

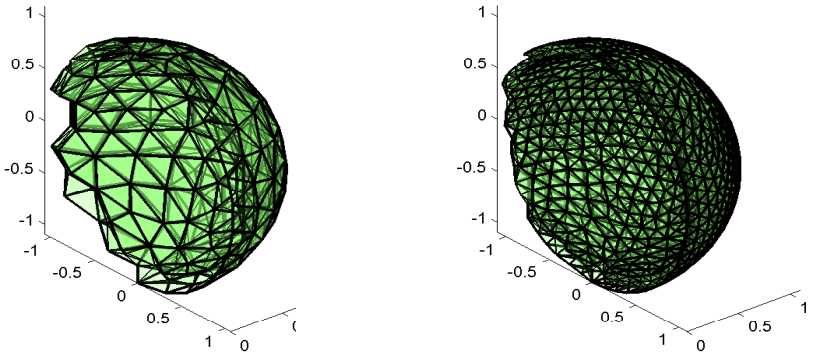
		Err(P1)		Err(P2)		Err(P3)		
level	*npoints*	FEM	BEM	FEM	BEM	FEM	BEM	RCPU
$L1$	265	2.73E-03	3.52E-03	2.51E-03	2.91E-03	9.72E-03	1.66E-03	3.6
$L2$	973	1.46E-03	1.29E-03	1.19E-03	9.18E-04	1.71E-03	4.67E-04	3.4
$L3$	7001	4.92E-04	3.44E-04	4.56E-04	2.52E-04	4.08E-04	1.15E-04	5.7

far from Γ. In our case the choice of $\mathcal{B}$ is arbitrary, both for what concerns its location (see the following example) and its shape (see Example 3).

Case b). Next we take the spherical artificial boundary $\mathcal{B}$ having radius $R = 1.1$, so that the shell Ω is a very thin one (see Figure 12).

Figure 12: Example 1. Case b). A section of the shell having the unit sphere as internal boundary and the sphere of radius 1.1 as external one: two discretizations into n_T tetrahedra.

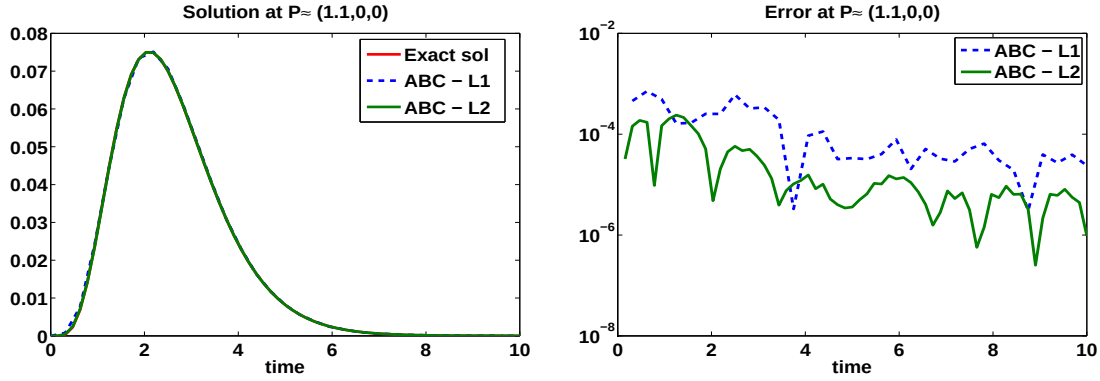
$n_T = 2895$ $\qquad$ $n_T = 12646$



We consider two levels of discretization, corresponding to the choice $L1 : n_T = 2895, N =$

32, and $L2$: $n_T = 12646, N = 64$. In Figure 13 we show the good agreement of the approximate solution with the reference BEM one at a point $P \approx (1.1, 0, 0)$ belonging to the artificial boundary and the corresponding absolute errors.

Figure 13: Example 1. Case b). Comparison between the exact solution and the approximate ones at $P \approx (1.1, 0, 0)$ (left plot) and associated errors (right plot), for the space and time mesh refinements $L1$ and $L2$.



Example 2. Case a). We consider the case of a wave generated by a single source and impinging upon a single body acting as a soft obstacle. The data of the problem are $u_0 = 0$, $v_0 = 0$, $g = 0$ and $f \neq 0$. We recall that, if a source is far from the area of interest, the existing local NRBCs would require to take a much larger domain Ω, to include the source, thus wasting computational time and space memory. This is not the case when we use our NRBC. Indeed, when a source is located in the residual domain $\mathcal{D}$, the source action is taken into account by the proposed artificial boundary condition.

To simplify the computation of the volume term I_f, 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, taking into account the presence of the delta Dirac function in the expression of the fundamental solution (2), we deduce the following simple form of the volume integral I_f defined in (6):

$$I_f(\mathbf{x}, t) = \begin{cases} \frac{h(t-\|\mathbf{x}-\mathbf{x}_0\|)}{4\pi\|\mathbf{x}-\mathbf{x}_0\|}, & \text{for all} \quad \mathbf{x} : \|\mathbf{x} - \mathbf{x}_0\| < t \\ 0 & \text{otherwise.} \end{cases} \tag{36}$$

We place the source $f(\mathbf{x}, t) = t^2 \sin(4t)e^{-t}\delta(\mathbf{x} - \mathbf{x}_0)$ at the point $\mathbf{x}_0 = (5, 0, 0)$; Γ and $\mathcal{B}$ are the surfaces of the spheres of radius $r_0 = .25$ and $R = .5$, respectively, both centered at the origin (see Figure 14 left plot). In Figure 14, right plot, we compare the approximate and the reference (BEM) solutions at the $\mathcal{B}_\Delta$ mesh point P closest to $(.5, 0, 0)$, for $t \in [0, 20]$. The reference solution has been obtained by a discretization of Γ into $nt_\Gamma = 122$ triangles and $N = 1E + 03$ time steps. The approximate solution has been obtained by using a decomposition of the spherical shell into $n_T = 226$ tetrahedra and taking $N = 500$ time steps. We note that the reference and the approximate solutions perfectly match and that the wave is null until the signal f reaches the point P at the time $t \approx 4.5$. Moreover, because of the presence of the term e^{-t} in the source f, the wave vanishes for large times without showing spurious reflections.

Figure 14: Example 2. Case a). Scatterer surrounded by the artificial boundary and external source f (left plot). Comparison between the reference solution and the approximate one at $P \approx (.5, 0, 0)$ (right plot).

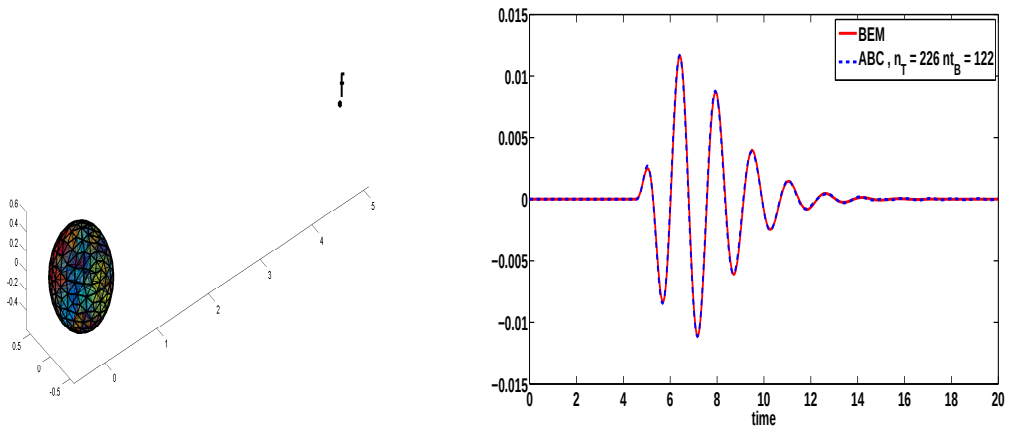
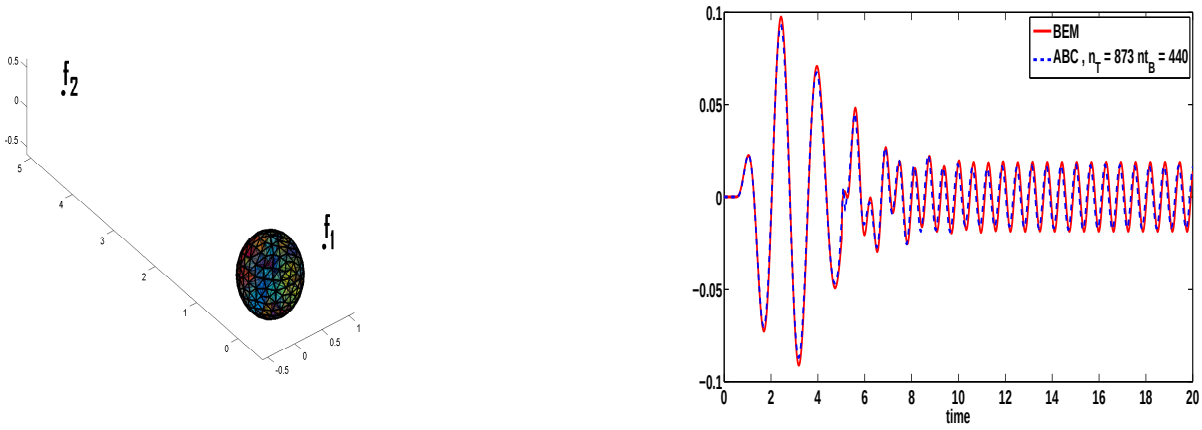


Figure 15: Example 2. Case b). Scatterer surrounded by the artificial boundary and external sources f_1 and f_2 (left plot). Comparison between the reference solution and the approximate one at $P \approx (.5, 0, 0)$ (right plot)



Case b). The treatment of a single source, external to the finite computational domain, can be easily extended to several sources, compactly supported and having disjoint supports. In this case, the volume term I_f that appears in the NRBC consists of the sum of several volume terms I_{f_i}, each of which is generated by the corresponding source f_i. In the next example, in the same setting of the case a), we consider the two point sources

$$\begin{aligned} f_1(\mathbf{x}, t) &= h_1(t)\delta(\mathbf{x} - \mathbf{x}_1) = t^2 \sin(4t) e^{-t} \delta(\mathbf{x} - \mathbf{x}_1) \\ f_2(\mathbf{x}, t) &= h_2(t)\delta(\mathbf{x} - \mathbf{x}_2) = \cos(10t)\delta(\mathbf{x} - \mathbf{x}_2), \end{aligned}$$

located in $\mathbf{x}_1 = (1, 0, 0)$ and $\mathbf{x}_2 = (0, 5, 0)$, respectively. It is easy to check that the volume term of the NRBC is given by $I_f = I_{f_1} + I_{f_2}$, where I_{f_i}, $i = 1, 2$ are given by (36).

In Figure 15 we show the history of the wave at the $\mathcal{B}_\Delta$ mesh point $P \approx (.5, 0, 0)$, for

$t \in [0, 20]$. Also in this case the reference and the approximate solutions match. At a time $t \approx 0.5$ the effect of the first source f_1 is visible at P, and when the effect of this source vanish (because of the presence of the e^{-t} term) the oscillatory behavior of the wave is due to the persistence of the signal f_2.

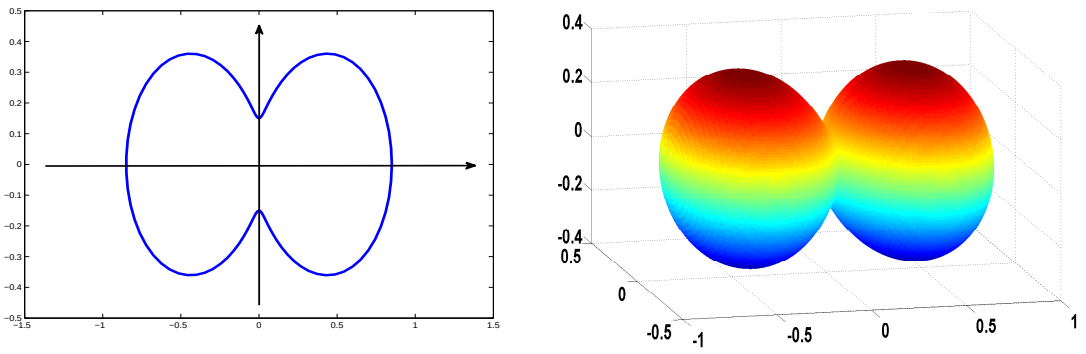
Example 3. The ductility of the artificial boundary is another important characteristic of the proposed NRBC. In this example we consider a single scatterer whose boundary Γ is the nut shape surface obtained by rotating the one dimensional curve

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

where $\rho(\theta) = c(1 + e\cos(n\theta))$, $c = 0.5$, $e = 0.7$, $n = 2$ and $\theta \in [0, 2\pi]$, along the x-axis (see Figure 16). The parametric representation of the three dimensional nut is therefore given by

$$\begin{aligned} X(\theta, \varphi) &= x(\theta) \\ Y(\theta, \varphi) &= \cos(\varphi)y(\theta) \\ Z(\theta, \varphi) &= \sin(\varphi)y(\theta). \end{aligned} \tag{37}$$

Figure 16: Example 3. The nut shape curve for the choice $c = 0.5$, $e = 0.7$ and $n = 2$ (left plot) and the corresponding three dimensional surface (right plot)



We prescribe on Γ the Dirichlet datum $g(\mathbf{x}, t) = \frac{1}{2\sqrt{\pi}}t^4e^{-2t}$ for all $t \geq 0$. In order to study the behavior of the solution in a thin region surrounding Γ, we choose the artificial boundary $\mathcal{B}$ having the same shape of Γ, that is the nut shape surface obtained with the choice of the parameters $c = 1$, $e = 0.7$ and $n = 2$. A cross section of the resulting finite computational domain, bounded internally by Γ and externally by $\mathcal{B}$, and a decomposition of it into tetrahedra, is represented in Figure 17. In Figure 18 we plot the behaviors of the BEM reference solution and of the approximate ones at a $\mathcal{B}_\Delta$ mesh point $P_1 \approx (1.7, 0, 0)$, for $t \in [0, 10]$. The approximations are obtained with different refinements of the spatial mesh and for a fixed refinement of the time interval into $N = 100$ steps. In particular we compare the approximation obtained when the matrix elements of the matrices $\mathbf{V_m}$ and $\mathbf{K_m}$ are computed by integrating over the plain triangles ("ABC plain" acronym for the solution, right plot) and the one obtained when the matrix elements are computed by integrating over

the curvilinear triangles ("ABC curv" acronym for the solution, left plot). We note that, with the same mesh refinement, the second approach produces a more accurate approximation with respect to the first one. In Figure 19 we compare the reference and approximate solutions at the $\mathcal{B}_\Delta$ point $P_2 \approx (0, 0.3, 0) \in \mathcal{B}_\Delta$, located in the nut narrowing part.

Figure 17: Example 3. A decomposition of the finite computational domain into tetrahedra

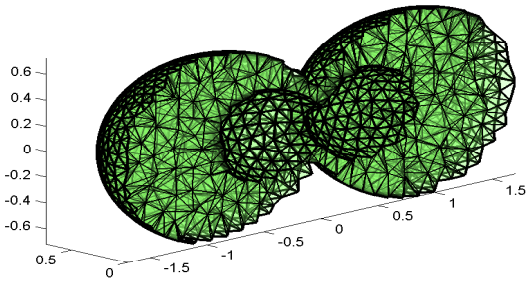
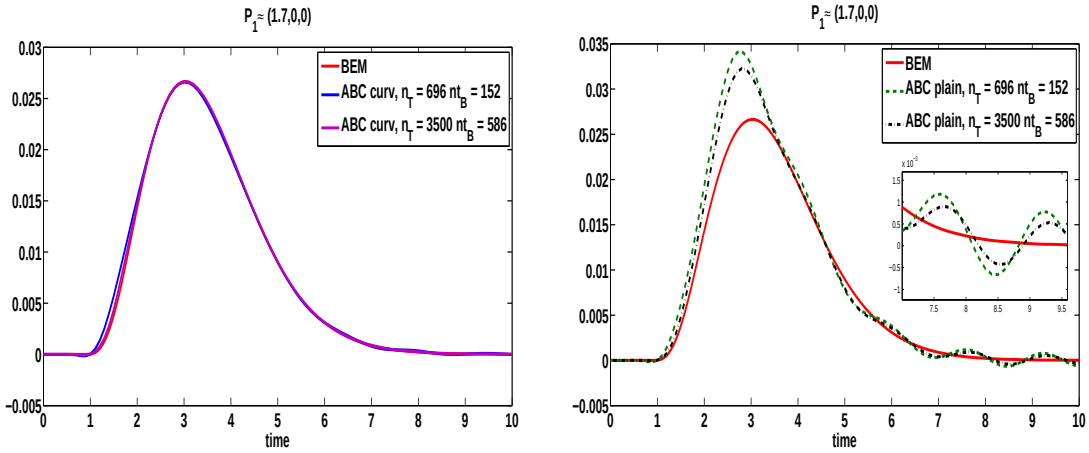
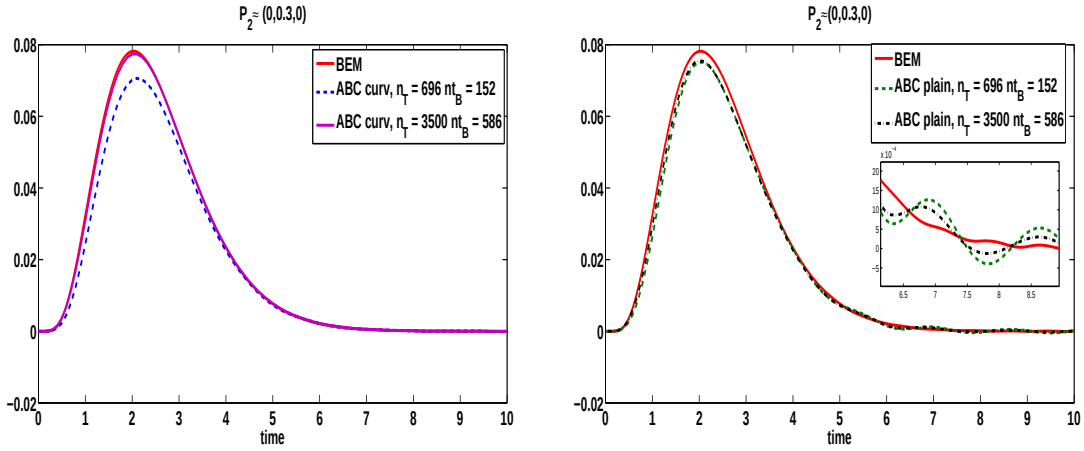


Figure 18: Example 3. Comparison between the reference solution and the approximate one at $P_1 \approx (1.7, 0, 0)$ by curvilinear triangles (left plot) and plain triangles (right plot), $N = 100$.



Example 4. In this example, we consider a multiple scattering problem. In particular we consider two disjoint spherical bodies whose boundaries Γ_1 and Γ_2 are spherical surfaces both of radius $r = 1$, centered at the origin and at $C = (6, 0, 0)$, respectively. As in Example 1, we assume that the source f and the initial data u_0 and v_0 are zero throughout the infinite exterior domain Ω^e. For all $t \geq 0$, we prescribe the Dirichlet condition $g(\mathbf{x}, t) = \frac{1}{2\sqrt{\pi}} t^4 e^{-2t}$ on Γ_1, and set $g(\mathbf{x}, t) = 0$ on Γ_2. Therefore, the second body acts as a soft reflecting obstacle. We compare the solution of this problem with the one of Example 1, case a), where a single scatterer is considered. To this end we choose the artificial boundary $\mathcal{B}_1$ as the spherical surface centered at the origin and with radius $R_1 = 2$, and the artificial boundary $\mathcal{B}_2$ as the

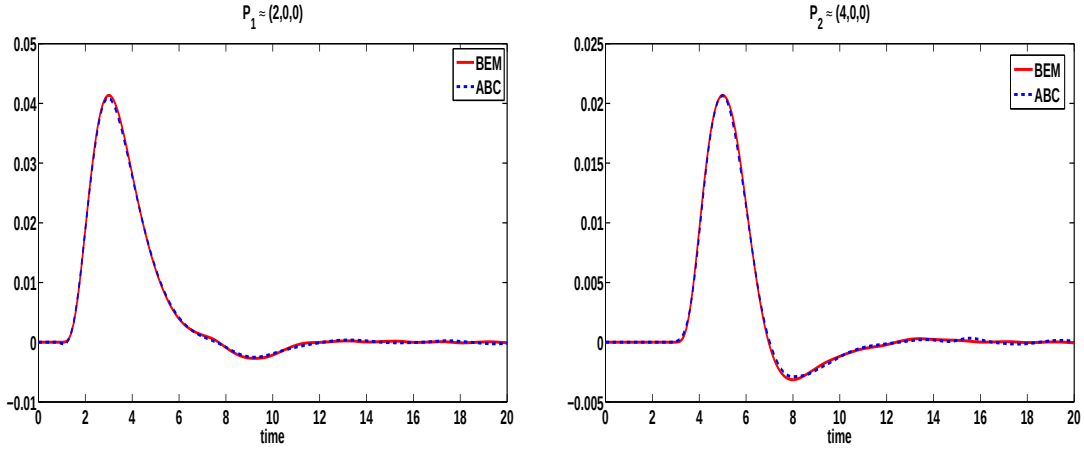
Figure 19: Example 3. Comparison between the reference solution and the approximate one at $P_2 \approx (0, 0.3, 0)$ by curvilinear triangles (left plot) and plain triangles (right plot), $N = 100$.



spherical surface centered at C and with radius $R_2 = 2$. We recall that the choice of $\mathcal{B}_1$ and $\mathcal{B}_2$ is quite flexible also in the multiple scattering case; indeed we have performed many other tests with different choices of the radius of $\mathcal{B}_1$ and $\mathcal{B}_2$ that are not shown here because redundant.

In Figure 20 we compare the BEM reference solution and the corresponding approximate one at the points $P_1 \approx (2, 0, 0), P_1 \in \mathcal{B}_{1,\Delta}$ and $P_2 \approx (4, 0, 0), P_2 \in \mathcal{B}_{2,\Delta}$, for $t \in [0, 20]$. Each reference solution has been obtained by a discretization of both surfaces into $nt_{\mathcal{B}_1} = nt_{\mathcal{B}_2} = 122$ triangles, and by performing $N = 1E + 03$ (equal) time steps in the interval $[0, 20]$. The approximate solutions are given by a decomposition of both spherical shells into $n_T = 24224$ tetrahedra and $N = 200$. It can be noticed that the solution at P_1 coincides with the exact solution represented in Figure 9 (right plot) until the time instant $t \approx 7$, when the effect of the second obstacle comes into play.

Figure 20: Example 4. Comparison between the reference solution and the approximate one at $P_1 \approx (2, 0, 0)$ (left plot) and at $P_2 \approx (4, 0, 0)$ (right plot), corresponding to the choice $n_T = 24224$ and $N = 200$.



Example 5. Case a). As last example, we consider an incident plane wave $u^{inc}(\mathbf{x},t) = e^{-5(\mathbf{x}_3-0.3-t)^2}$ that impinges upon two spherical scatterers. The two spheres have radius $r = 0.25$ and are centered at $C_1 = (0,0,-3)$ and $C_2 = (0,0,3.1)$, along the axis z. The total wave field is given by the sum of the incident wave u^{inc} and the scattered one u^{scatt}; this latter is the solution of (1) with null initial data, null source and Dirichlet boundary condition $g(\mathbf{x},t) = -u^{inc}(\mathbf{x},t)$ on the surface of the two obstacles. We compute the solution in two distinct regions: the two spherical shells Ω_1 and Ω_2 bounded externally by the spherical surfaces of radius 0.5 and centered at C_1 and C_2, respectively.

We point out that at the time instant $t = 0$ the Dirichlet condition takes values that are of the order of the machine precision, so that the required compatibility conditions are numerically satisfied. In Figure 21 we plot the behavior, in the time window $[0,20]$, of the total field at four points: the points $P_1 \approx (0,0,-2.5)$ and $P_2 \approx (0,0,2.6)$ that belong to the artificial boundaries of Ω_1 and Ω_2, respectively, and $P_3 \approx (0,0,-2.625)$ and $P_4 \approx (0,0.375,-3)$ that are points interior to the spherical shell Ω_1. The numerical solution has been obtained with a partition of the shells into $n_T = 5299$ tetrahedra and $N = 400$ time steps.

The incident wave is centered at $\mathbf{x}_3 = 0.3$, $t = 0$, and starts moving from the bottom to the top in the z direction. It first enters the computational domain Ω_2 at $t \approx 2$ and bumps against the obstacle centered at C_2, thus generating a scattered wave that leaves Ω_2 at $t \approx 4$ and reaches the domain Ω_1 at $t \approx 8$ (see right and left top plots in Figure 21). At the interior points P_3 and P_4 the effect of the wave that is further scattered by the obstacle centered at C_1 and propagates in Ω_1 is visible (see left and right bottom plots in Figure 21). For the sake of comparison, we have reported on the same graph the behavior of the solution obtained by the BEM.

Case b) Finally, in Figure 22, we report the solution obtained by adding, to the problem described above, the exterior source term $f(\mathbf{x},t) = H(t-2)f_1(\mathbf{x},t-2)$, where $f_1(\mathbf{x},t) = t^2 sin(4t)e^{-t}\delta(\mathbf{x})$ is the source function centered at the origin of the axis. The effect of the wave generated by the source f is clearly visible at the point P_1, at $t \approx 4$, before that the effect of the wave generated by the scattered signal starts summing to it at $t \approx 8$. Similar considerations hold for the points P_3 and P_4 too. On the Contrary, at P_2 the effect of the scattered wave remains unchanged until $t \approx 6$, when the external source comes into play, since the incident wave reaches the obstacle centered at C_2 earlier than the wave generated by the source f.

Figure 21: Example 5. Case a). Comparison between the reference solution and the approximate one at $P_1 \approx (0, 0, -2.5)$ (top-left plot), $P_2 \approx (0, 0, 2.6)$ (top-right plot), $P_3 \approx (0, 0, -2.625)$ (bottom-left plot) and $P_4 \approx (0, 0.375, -3)$ (bottom-right plot) corresponding to the choice $n_T = 5299$, $T = 20$ and $N = 400$.

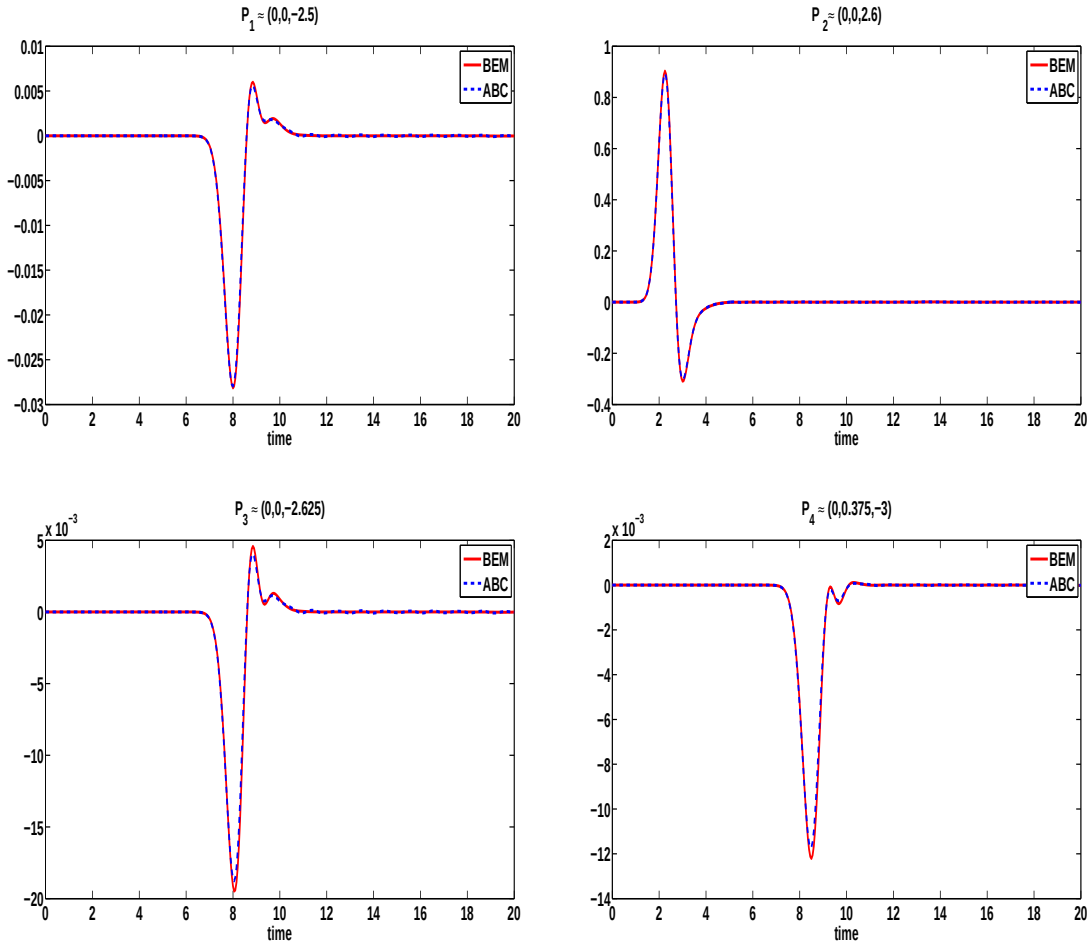
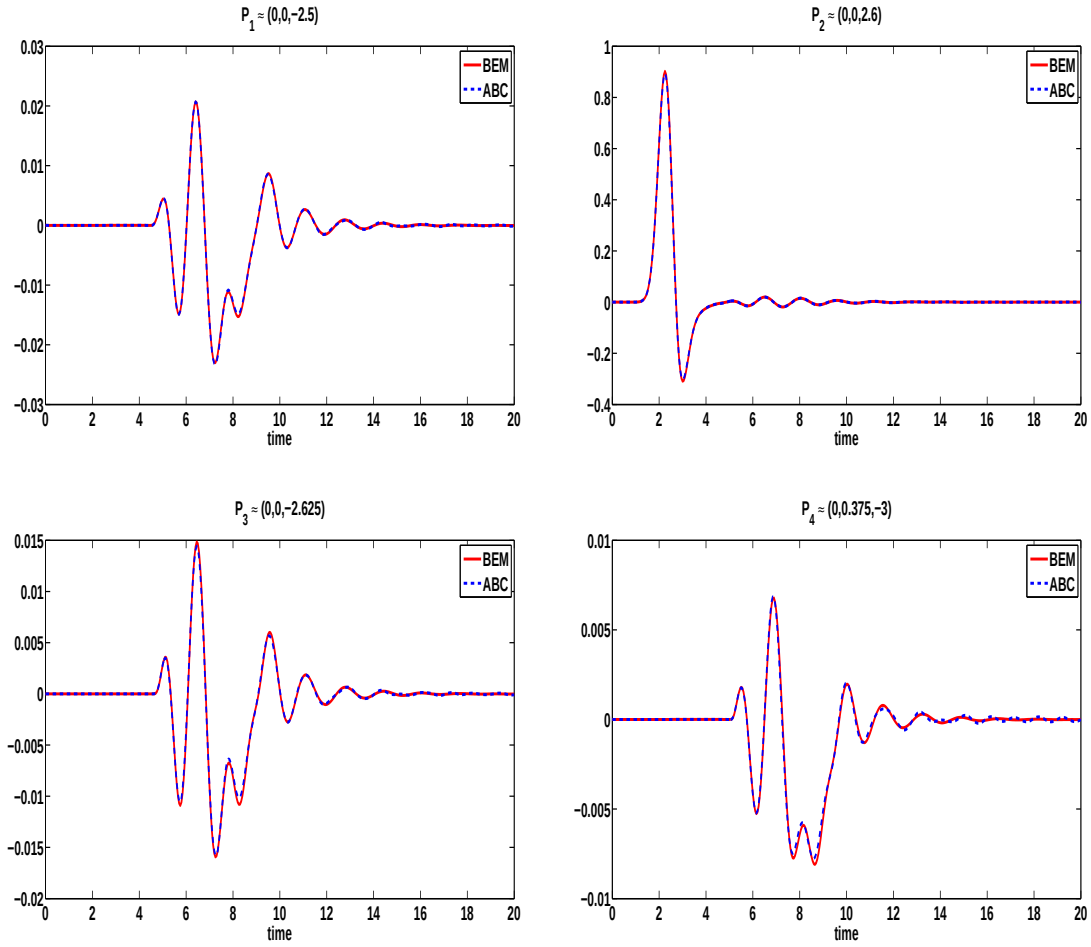


Figure 22: Example 5. Case b). Comparison between the reference solution and the approximate one at $P_1 \approx (0, 0, -2.5)$ (top-left plot), $P_2 \approx (0, 0, 2.6)$ (top-right plot), $P_3 \approx (0, 0, -2.625)$ (bottom-left plot) and $P_4 \approx (0, 0.375, -3)$ (bottom-right plot) corresponding to the choice $n_T = 5299$, $T = 20$ and $N = 400$.



7. Conclusions

In [24] the authors have constructed a fully local ABC, to deal with 3D multi-scattering problems. As for all local conditions, sources must be included in the computational domain. Moreover, the shape of artificial boundary the authors consider is that of a spherical surface, one for each obstacle, although they remark that ellipsoidal ones could also be taken. Its computational cost is $O(N_{\mathcal{B}}N)$. We must however remark that this ABC requires, as stated by the authors, a number of auxiliary functions that must be judiciously chosen, depending on the problem or desired accuracy. Furthermore, a local spline interpolation on the obstacle boundaries is also required. All the examples they present have a symmetry that allows their reduction to corresponding 2D problems, which are then solved by coupling the proposed ABC with a finite-difference scheme. It is however not clear how their approach should be applied in the case of obstacles having more general shapes, and with the finite difference method replaced by the FE one.

Our approach is more general. Obstacles and artificial boundaries can have any (smooth) shape. Furthermore we can even have multi sources, which do not have to be necessarily included in the computational domain. The NRBC naturally takes into account their effects. One does not have to separate incoming waves from outgoing ones. The NRBC will be transparent for each one of them. Finally, we also mention that in principle, we could even have obstacles which are rotating independently from each other. Indeed, some numerical testing we have already performed in the 2D case have given very promising results.

This generality has however a cost, both in terms of CPU and space memory, which is certainly higher than that of the above mentioned local condition. But we believe that a deeper investigation on these aspects should lead to further savings. The goal is to have a computational cost close to that of the associated FEM.

Besides the computational cost and the space memory, another possible drawback of the proposed NRBC is that the time interval of integration must be fixed in advance. Moreover, if the computation of the integrals defining the NRBC is not performed with the needed accuracy, as pointed out in Remark 4.1, instabilities might arise before reaching the final instant T. But increasing the number of quadrature nodes means to increase the NRBC computational cost. Nevertheless, in our opinion there is still room for reducing these drawbacks, including the computational complexity and the working space. This includes the use of discrete convolution quadratures alternative to those of Lubich (see [10], [36], [16]), which should allow the construction of highly sparse $\mathbf{K}_m$ and $\mathbf{V}_m$ matrices, with the position of the non zero elements known a priori, but at the cost of loosing the FFT benefits; the use of higher order Lubich convolution rules (see for example [34], [2]) or of time integration formulas which do not require to fix a priori the final time instant T and to proceed with constant time step-size (see [31], [32]). Finally, we recall that very recently data-sparse techniques, such has panel-clustering, $\mathcal{H}$-matrices and high-frequency fast multipole methods, have been used to reduce the overall computational cost of a 3D space-time BIE Galerkin discretization (see [15], [3]). However, the use of these strategies for reducing the computational cost of our NRBC is still at an early stage and needs further investigation.

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., sequen-

tial) Matlab® codes.

Acknowledgements

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

References

[1] M. Abramowitz and I. A. Stegun, Handbook of Mathematical Functions, National Bureau of Standards, U.S. Department of Commerce, Applied Mathematics Series 55, 1967.

[2] Banjai, L. and Kachanovska, M., Sparsity of Runge-Kutta convolution weights for the three-dimensional wave equation, BIT 54(4), 2014, 901-936.

[3] Banjai, L. and Kachanovska, M., Fast convolution quadrature for the wave equation in three dimensions, J. Comput. Phys. 279 , 2014, 103-126.

[4] A. Bayliss, E. Turkel, Radiation boundary conditions for wave-like equations, Comm. Pure Appl. Math. 33 (1980) 707-725.

[5] J. P. Bérenger, A Perfectly Matched Layer for the Absorption of Electromagnetic Waves, J. Comp. Physics. 114 (1994) 185–200.

[6] A. Bermúdez, L. Hervella-Nieto, A. Prieto, R. Rodríguez, Perfectly Matched Layers, in: Computational Acoustic of Noise Propagation in Fluids - Finite and Boundary Elements (S. Marburg, B. Nolte, eds), Springer, 2008, Chapter 6, 167-196.

[7] D.A. Bini, B. Meini, Fast algorithms with applications to Markov chains and queueing models, in:*Fast Reliable Algorithms for Matrices with Structure* (T. Kailath, A.H. Sayed, eds.), SIAM, Philadelphia, PA, 1999, 211-243.

[8] S.N. Chandler-Wilde, P. Monk, Wave-number-explicit bounds in time-harmonic scattering. SIAM J. Math. Anal. 39 (2008), no. 5, 1428-1455.

[9] 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.

[10] P.J. Davis, D.B. Duncan, Convolution-in-time approximations of time domain boundary integral equations, SIAM J. Sci. Comput. 35(1), 2013, B43-B61.

[11] B. Engquist, A. Majda, Absorbing boundary conditions for numerical simulation of waves, Math. Comput. 31 (1977) 629-651.

[12] 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.

[13] S. Falletta, G. Monegato, L. Scuderi, A space-time BIE method for nonhomogeneous exterior wave equation problems. The Neumann case, IMA J. Numer. Anal. 32 (2014) 202-226.

[14] S. Falletta, G. Monegato, An exact non reflecting boundary condition for 2D time-dependent wave equation problems. Wave Motion 51 (2014), no. 1, 168-192.

[15] S. Falletta, S. Sauter, The panel-clustering method for the wave equation in two spatial dimensions, submitted.

[16] S. Falletta, L. Scuderi, A new boundary element integration strategy for retarded potential boundary integral equations, App. Num. Math., 94 (2015) 106-126.

[17] D. Givoli, Numerical Methods for Problems in Infinite Domains, Elsevier, Amsterdam, 1992.

[18] D. Givoli, Recent advances in the DtN FE method, Archives of Computational Methods in Engineering 6 (1999) 71-116.

[19] D. Givoli, High-order local non-reflecting boundary conditions: A review, Wave Motion 39 (2004) 319-326.

[20] D. Givoli, Computational absorbing boundaries, in: Computational Acoustic of Noise Propagation in Fluids - Finite and Boundary Elements (S. Marburg, B. Nolte, eds), Springer, 2008, Chapter 5, 145-166.

[21] D. Givoli, B. Neta, High-order non-reflecting boundary scheme for time dependent waves, J. Comput. Phys. 186 (2003) 24-46.

[22] I. Gradshteyn, I. Ryzhik, Table of Integrals, Series and Products. Academic Press, 1965.

[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. Hagström, S. I. Hariharan, A formulation of asymptotic and exact boundary conditions using local operators, Appl. Numer. Math. 27 (1998) 403-416.

[27] T. Hagström, 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] F. Hecht, New development in FreeFem++, J. Numer. Math. 20 (2012), no. 3-4, 251-265.

[29] P. Joly, An elementary introduction to the construction and the analysis of perfectly matched layers for time domain wave propagation. SeMA Journal 57 (2012) 5-48.

[30] P. Keast, Moderate-degree tetrahedral quadrature formulas, Comp. Meth. Appl. Mech. Engrg., 55 (1986), 339–348.

[31] M. Lopez-Fernandez, S. Sauter, Generalized convolution quadrature with variable time stepping IMA J. Numer. Anal. 33(4), 2013, 1156-1175.

[32] M. Lopez-Fernandez, S. Sauter, Generalized convolution quadrature with variable time stepping. Part II: algorithm and numerical results, Appl. Num. Math. 94 (2015) 88-105.

[33] Ch. Lubich, Convolution quadrature and discretized operational calculus. I, Num. Math. 52 (1988) 129-145.

[34] G. Monegato, L. Scuderi, M.P. Stanić, Lubich convolution quadratures and their application to problems described by space-time BIEs, Numerical Algorithms 56 (2011) 405-436.

[35] G. Monegato, L. Scuderi, A space-time BIE method for 2D mixed wave equation problems, Appl. Math. Comp. 259 (2015) 1046-1070.

[36] S. Sauter, A. Veit, A Galerkin method for retarded boundary integral equations with smooth and compactly supported temporal basis functions, Numer. Math. 123 (2013), no. 1, 145–176.

[37] A. Taflove, S.C. Hagness, Computational Electrodynamics. The finite-difference time-domain method, Hartech House, Boston-London, 2005.