Wavelets and convolution quadrature for the efficient solution of a 2D space-time BIE for the wave equation

Wavelets and convolution quadrature for the efficient solution of a 2D space-time BIE for the wave equation[1]

S. Bertoluzza[a], S. Falletta[b], L. Scuderi[c]

[a] *I.M.A.T.I - CNR, Pavia , Italy.*
Email: silvia.bertoluzza@imati.cnr.it
[b] *Dipartimento di Scienze Matematiche, Politecnico di Torino, Italy.*
Email: silvia.falletta@polito.it
[c] *Dipartimento di Scienze Matematiche, Politecnico di Torino, Italy.*
Email: letizia.scuderi@polito.it

Abstract We consider a wave propagation problem in 2D, reformulated in terms of a Boundary Integral Equation (BIE) in the space-time domain. For its solution, we propose a numerical scheme based on a convolution quadrature formula by Lubich for the discretization in time, and on a Galerkin method in space. It is known that the main advantage of Lubich's formulas is the use of the FFT algorithm to retrieve discrete time integral operators with a computational complexity of order $\mathcal{R}\log\mathcal{R}$, $\mathcal{R}$ being twice the total number of time steps performed. Since the discretization in space leads in general to a quadratic complexity, the global computational complexity is of order $M^2\mathcal{R}\log\mathcal{R}$ and the working storage required is $M^2\mathcal{R}/2$, where M is the number of grid points on the domain boundary.

To reduce the complexity in space, we consider here approximant functions of wavelet type. By virtue of the properties of wavelet bases, the discrete integral operators have a rapid decay to zero with respect to time, and the overwhelming majority of the associated matrix entries assume negligible values. Based on an a priori estimate of the decaying behaviour in time of the matrix entries, we devise a *time downsampling strategy* that allows to compute only those elements which are significant with respect to a prescribed tolerance. Such an approach allows to retrieve the temporal history of each entry $\boldsymbol{e}$ (corresponding to a fixed couple of wavelet basis functions), via a Fast Fourier Transform, with computational complexity of order $\overline{\mathcal{R}}_{\boldsymbol{e}}\log\overline{\mathcal{R}}_{\boldsymbol{e}}$. The parameter $\overline{\mathcal{R}}_{\boldsymbol{e}}$ depends on the two basis functions, and it satisfies $\overline{\mathcal{R}}_{\boldsymbol{e}} \ll \mathcal{R}$ for a relevant percentage of matrix entries, percentage which increases significantly as time and/or space discretization are refined. Globally, the numerical tests show that the computational complexity and memory storage of the overall procedure are linear in space and time for small velocities of the wave propagation, and even sub-linear for high velocities.

Key words: wave equation; space-time boundary integral equations; multiresolution analysis; down-

[1]This work was supported by GNCS-INDAM 2016 research program: Accoppiamento FEM-BEM non conforme mediante tecniche di decomposizione di dominio di tipo mortar; GNCS-INDAM 2017 research program: Nuove tecniche numeriche per la risoluzione di problemi evolutivi mediante il metodo degli elementi di contorno. The present research has been performed in the framework of MIUR grant Dipartimenti di Eccellenza 20182022, CUP E11G18000350001.

1. Introduction

Let $\Omega^e = \mathbb{R}^2 \setminus \overline{\Omega^i}$ be the complement of a bounded rigid obstacle $\Omega^i \subset \mathbb{R}^2$, having a closed smooth boundary Γ. We consider the following exterior Dirichlet problem for the wave equation:

$$\begin{cases} \frac{1}{c^2} u_{tt}(\mathbf{x},t) - \Delta u(\mathbf{x},t) &= 0 & \text{in } \Omega^e \times (0,T) \\ u(\mathbf{x},t) &= g(\mathbf{x},t) & \text{on } \Gamma \times (0,T) \\ u(\mathbf{x},0) &= 0 & \text{in } \Omega^e \\ u_t(\mathbf{x},0) &= 0 & \text{in } \Omega^e, \end{cases} \tag{1}$$

where c represents the wave velocity.

It is well known that the following single-layer potential representation

$$u(\mathbf{x},t) = \int_\Gamma \int_0^t G(|\mathbf{x}-\mathbf{y}|, t-\tau)\varphi(\mathbf{y},\tau)\, d\tau\, d\Gamma_{\mathbf{y}} \quad \mathbf{x} \in \Omega^e, \quad t \in [0,T] \tag{2}$$

holds, where $G(\mathbf{x},t)$ denotes the fundamental solution

$$G(r,t) = \frac{1}{2\pi} \frac{H\left(t - \frac{r}{c}\right)}{\sqrt{t^2 - \frac{r^2}{c^2}}}, \qquad r = |\mathbf{x}-\mathbf{y}|, \tag{3}$$

$H(\cdot)$ being the Heaviside function. The function φ in (2) is solution of the following Time Dependent Boundary Integral Equation (TDBIE)

$$\int_\Gamma \int_0^t G(r, t-\tau)\varphi(\mathbf{y},\tau)\, d\tau d\Gamma_{\mathbf{y}} = g(\mathbf{x},t), \quad \mathbf{x} \in \Gamma, \quad t \in [0,T] \tag{4}$$

and represents the jump of the normal derivative of u along Γ.

Several numerical approaches have been proposed for solving wave equation problems by means of BIEs. In particular, we mention the pioneering work by Bamberger and Ha Duong for scattering problems in the frequency domain [1], Lubich's convolution quadrature method [2] and the energetic approach [3] for solving time dependent boundary integral equations. In [1, 2], theoretical results on stability and convergence are proved when a Galerkin scheme in space is considered.

Lubich's convolution method for time integral discretization is based on quadrature formulas associated with A-stable methods for ordinary differential equations. Rather than using the explicit expression of the integral equation kernel, Lubich's formulas use its Laplace

transform, which, in the case of wave equations, turns out to have better regularity properties. The major advantage of this approach is that it allows to reduce the computational complexity of the time discretization to an order $\mathcal{R}\log\mathcal{R}$ ($\mathcal{R} = 2N$, where N is the total number of time steps performed), thanks to the Fast Fourier Transform (FFT). The time scheme is generally coupled with a collocation or a Galerkin method in space, leading to a quadratic complexity M^2, where M denotes the number of grid points chosen on the domain boundary. Therefore, the overall (space-time) complexity is of order $M^2\mathcal{R}\log\mathcal{R}$. Such an approach has been successfully applied to wave propagation problems in 2D and in 3D, with Dirichlet, Neumann and mixed boundary conditions (see [4], [5], [6]). Lubich's convolution quadrature method has also been used for the time approximation of non reflecting boundary conditions of exact type prescribed on artificial boundaries, in a FEM-BEM coupling approach for the solution of exterior problems (see [7] and [8] and the very recent work [9] for problems of waves scattered by moving obstacles).

The main drawback of this approach is the high memory requirement, which is M^2N. In fact it requires to store all the matrices involved in the final linear system, which, when standard Lagrangian basis functions are considered for the space approximation, are generally fully populated.

In three dimensions and for the Lubich-collocation approach, the computational cost and the required memory storage can be significantly reduced when the wave propagation velocity is much higher than one. Indeed, in [10] the authors showed that, for high velocities, only a very small number of matrices, let us say the first N_0, with $N_0 \ll N$, are significant with respect to a prescribed tolerance, and the remaining ones can be neglected without affecting the solution accuracy. Another fast technique for the 3D case, which relies on the sparsity of the Runge-Kutta convolution weights, has been proposed in [11]. This technique focuses on the single layer formulation of the boundary integral equation, and combines the fast decay of the convolution weights for the computation of the near field with sparse data techniques for the computation of the far field.

It is known that, contrarily to the 3D case, in 2D the convolution weights do not decay, since the Huygens' principle does not hold. For this reason, the above mentioned fast techniques can not be applied. However, over the last decades, several effective techniques have been developed in such a framework, aimed at reducing computational cost and memory storage. Among these we mention fast multiple methods ([12]), panel clustering ([13], [14]) and hierarchical matrices ([15]). However, for these methods, the FFT algorithm cannot be applied. Recent techniques in [16] and [17] exploit the property of the convolution weight kernels to approximate the tail of the fundamental solution in the time domain, which turns out to be very smooth. In both papers the authors propose the fast and oblivious convolution quadrature to reduce both computational cost and memory requirement for computing the aforesaid tail for long time computation.

In this paper we propose a new and alternative approach which is based on a wavelet type approximation in space, and is applied to the global space-time discretization. It is known that wavelet approximations have the property of yielding sparse matrices when applied

to a wide class of pseudo-differential operators ([18]). Wavelet BEMs have already been considered, for example, in [19], [20], [21], [22] for stationary problems. In [23] the authors apply a wavelet BEM for a time dependent wave equation combined with finite differences in time. To our knowledge, there are no further papers dealing with wavelet approximations in time dependent wave equation problems.

The new approach we propose here combines the good properties of the wavelet Galerkin approximation in space and those of Lubich's convolution quadrature method in time. This has the twofold effect of heavily sparsifying the matrices generated by space discretization and of considerably reducing the computational cost of time discretization. This combination turns out to have an effect that goes well beyond the sum of the independent effects of the two techniques. Indeed, the rapid time decreasing behaviour of the matrix entries allows to devise a time compression strategy consisting in downsampling the FFT involved in Lubich's convolution quadrature method, with a resulting significant reduction of the overall computational complexity. More precisely, denoting by $\{\psi_\lambda,\ \lambda \in \Lambda_L\}$ the wavelet basis for the space discretization, where Λ_L is a suitable finite index set, the computation of the matrix entry corresponding to two basis functions ψ_λ and $\psi_{\lambda'}$, which by Lubich's technique is carried out simultaneously for all times, can be performed with a computational cost of order $\overline{\mathcal{R}}_{\lambda,\lambda'} \log \overline{\mathcal{R}}_{\lambda,\lambda'}$. The parameter $\overline{\mathcal{R}}_{\lambda,\lambda'}$ satisfies $\overline{\mathcal{R}}_{\lambda,\lambda'} \ll \mathcal{R}$ for a large percentage of couples λ, λ', resulting, on one hand, in a consistent memory storage reduction and, on the other hand, in a significant increase the time quadrature efficiency. More precisely, extensive numerical tests show that the computational complexity $\sum_{\lambda,\lambda'} \overline{\mathcal{R}}_{\lambda,\lambda'} \log \overline{\mathcal{R}}_{\lambda,\lambda'}$ and the memory storage $\sum_{\lambda,\lambda'} \overline{\mathcal{R}}_{\lambda,\lambda'}$ of the overall procedure are linear in space and time (the former up to a logarithmic factor) for small velocities and even sub-linear for high velocities.

The paper is organized as follows. In Sections 2.1 and 2.2 we introduce the proposed method, by describing the main steps that lead to the time convolution quadrature formula associated to the single-layer potential BIE (4) and the spatial wavelet discretization, in a rather abstract setting. Convergence estimates are derived for the full discretized scheme. In Sections 2.3.1 and 2.3.2 we present the fundamental properties of wavelet bases that will be used to estimate the decaying behaviour in time of the matrix entries. In Sections 2.3.3 and 2.3.4 we describe the time downsampling strategy that we apply in the new proposed method to retrieve final sparse matrices while maintaining the optimal time computational complexity of the FFT algorithm. Finally, in Sections 3, we apply the proposed numerical approach to several problems, to show its efficiency in terms of accuracy, computational complexity and memory saving.

2. Lubich-wavelet Galerkin BEM

In this section we present the Lubich-wavelet Galerkin BEM that we apply to equation (4). It is based on a BDF2 Lubich convolution quadrature formula in time and a Galerkin method in space, which uses biorthogonal wavelet approximating basis functions. In the sequel whatever refers to this approach will be labelled by the superscript $\mathcal{W}$.

In what follows, the notation $Q1 \lesssim Q2$ (resp. $Q1 \gtrsim Q2$) means that the quantity $Q1$ is bounded from above (resp. from below) by $C \cdot Q2$, where C is a constant that, unless explicitely stated, does not depend on any relevant parameter involved in the definition of $Q1$ and $Q2$. The notation $\simeq$ means that both $Q1 \lesssim Q2$ and $Q1 \gtrsim Q2$ hold. The notation $Q1 \approx Q2$, on the other hand, will simply mean that we approximate $Q1$ with $Q2$.

2.1. Time discretization

For the time discretization, we split $[0, T]$ into N steps of equal length $\Delta_t = T/N$ and collocate equation (4) at the time instants $t_n = n\Delta_t$, $n = 0, \dots, N$:

$$\int_\Gamma \int_0^{t_n} G(r, t_n - \tau)\varphi(\mathbf{y}, \tau)\, d\tau\, d\Gamma_\mathbf{y} = g(\mathbf{x}, t_n), \ \mathbf{x} \in \Gamma, \ n = 0, \dots, N. \tag{5}$$

The time integrals are then discretized by means of Lubich's convolution quadrature rule associated with the BDF method of order 2 (see [24]):

$$\int_0^{t_n} G(r, t_n - \tau)\varphi(\mathbf{y}, \tau)\, d\tau \approx \sum_{j=0}^{n} \omega_{n-j}(\Delta_t; r)\varphi^j(\mathbf{y}), \quad n = 0, \dots, N \tag{6}$$

where we have set $\varphi^j(\mathbf{y}) := \varphi(\mathbf{y}, t_j)$. The coefficients ω_n are defined by the contour integrals

$$\omega_n(\Delta_t; r) = \frac{1}{2\pi\imath} \int_{|z|=\rho} \widehat{G}\left(r, \frac{\gamma(z)}{\Delta_t}\right) z^{-(n+1)}\, dz, \tag{7}$$

where

$$\widehat{G}(r, s) = \frac{1}{2\pi} K_0\left(s\frac{r}{c}\right)$$

denotes the Laplace transform of the fundamental solution G, K_0 being the modified Bessel function of second kind and of order 0. In (7) γ is the characteristic quotient of the Backward Differentiation Formula of order 2 (BDF2), i.e $\gamma(z) = 3/2 - 2z + 1/2z^2$ and ρ is such that for $|z| \le \rho$ the corresponding $\gamma(z)$ lies in the domain of analyticity of $\widehat{G}$ (for details see [2]).

By introducing the polar coordinate $z = \rho e^{\imath\vartheta}$, the integrals in (7) can be efficiently computed by a trapezoidal rule with $\mathcal{R}$ equal steps of length $2\pi/\mathcal{R}$:

$$\omega_n(\Delta_t; r) \approx \frac{\rho^{-n}}{\mathcal{R}} \sum_{m=0}^{\mathcal{R}-1} \widehat{G}\left(r, \frac{\gamma(\rho e^{\imath 2\pi \frac{m}{\mathcal{R}}})}{\Delta_t}\right) e^{-\imath 2\pi n \frac{m}{\mathcal{R}}}. \tag{8}$$

For the computation of the convolution coefficients in (8), a proper choice of the involved parameters, suggested by Lubich in [25], is $\mathcal{R} = 2N$ and ρ such that $\rho^N = \sqrt{\varsigma}$. Indeed, these choices lead to an approximation of ω_n with a relative error of order $\sqrt{\varsigma}$, if $\widehat{G}$ is computed with a relative accuracy bounded by ς. We point out that in the numerical tests we have chosen $\varsigma = 1.0e - 12$.

2.2. Space discretization

For the space discretization, we consider a standard Galerkin boundary element method. For simplicity, we assume that Γ is given by a global parametric representation. In this case the integration over Γ is reduced to an integration over the parametrization interval. Precisely, let

$$\mathbf{y} = \boldsymbol{\eta}(\theta) = (\eta_1(\theta), \eta_2(\theta)), \quad \theta \in [0, 2\pi)$$

denote the parametrization of Γ. By discretizing the parametrization interval into $M = 2^L$ subintervals, we introduce a finite dimensional space $V_L \subseteq C^0(0, 2\pi)$ satisfying periodic boundary conditions. In view of the matrix compression technique that we are going to employ further on, which will make use of wavelet functions, it is convenient to denote the basis of V_L by $\{\psi_\lambda,\ \lambda \in \Lambda_L\}$, where Λ_L denotes a suitable finite index set of cardinality M, whose form may be more general than the usual $\{1, 2, \cdots, M\}$, as we will see in the next section.

By replacing (6) in (5) and approximating the unknown φ^j by

$$\varphi^j(\boldsymbol{\eta}(\theta)) \approx \sum_{\lambda \in \Lambda_L} d_\lambda^j \psi_\lambda(\theta) =: \varphi_L^j(\theta), \tag{9}$$

we obtain the integral equations

$$\sum_{j=0}^{n} \sum_{\lambda \in \Lambda_L} d_\lambda^j \int_0^{2\pi} \omega_{n-j}(\Delta_t; r(\theta, \sigma)) \psi_\lambda(\theta) |\boldsymbol{\eta}'(\theta)|\, d\theta = g(\boldsymbol{\eta}(\sigma), t_n), \quad \mathbf{x} = \boldsymbol{\eta}(\sigma) \in \Gamma, \tag{10}$$

for $n = 0, \cdots, N$, where now we set $r(\theta, \sigma) = |\boldsymbol{\eta}(\theta) - \boldsymbol{\eta}(\sigma)|$. By using, in a Galerkin approach, test functions from the same basis we end up with the following problem: find d_λ^j, $j = 0, \cdots, N$, $\lambda \in \Lambda_L$, such that for all $n = 0, \cdots, N$ and for all $\lambda' \in \Lambda_L$

$$\begin{aligned} \sum_{j=0}^{n} \sum_{\lambda \in \Lambda_L} d_\lambda^j \int_0^{2\pi} \int_0^{2\pi} \omega_{n-j}(\Delta_t; r(\theta, \sigma))\, \psi_\lambda(\theta) \psi_{\lambda'}(\sigma)\ |\boldsymbol{\eta}'(\theta)|\, |\boldsymbol{\eta}'(\sigma)|\ d\theta\, d\sigma \\ = \int_0^{2\pi} g(\boldsymbol{\eta}(\sigma), t_n)\, \psi_{\lambda'}(\sigma)\ |\boldsymbol{\eta}'(\sigma)|\ d\sigma. \end{aligned} \tag{11}$$

For all $n = 0, \cdots, N$, we will denote by $\boldsymbol{V}^{\mathcal{W},n}$ and $\boldsymbol{g}^{\mathcal{W},n}$ the $M \times M$ matrix and $M \times 1$ vector whose generic elements are, respectively,

$$\boldsymbol{V}_{\lambda,\lambda'}^{\mathcal{W},n} := \int_0^{2\pi} \int_0^{2\pi} \omega_n(\Delta_t; r(\theta, \sigma))\, \psi_\lambda(\theta) \psi_{\lambda'}(\sigma)\ |\boldsymbol{\eta}'(\theta)|\, |\boldsymbol{\eta}'(\sigma)|\ d\theta\, d\sigma, \tag{12}$$

and

$$\boldsymbol{g}_{\lambda'}^{\mathcal{W},n} := \int_0^{2\pi} g(\boldsymbol{\eta}(\sigma), t_n)\, \psi_{\lambda'}(\sigma)\ |\boldsymbol{\eta}'(\sigma)|\ d\sigma. \tag{13}$$

Here, by abuse of notation, we index also matrices and vectors with the elements of Λ_L. More precisely for an $M \times M$ matrix $\boldsymbol{X}$ and an $M \times 1$ vector $\boldsymbol{x}$ we write $\boldsymbol{X}_{\lambda,\lambda'}$ and $\boldsymbol{x}_\lambda$ for $\boldsymbol{X}_{i,i'}$ and $\boldsymbol{x}_i$ with $i = \iota(\lambda)$, $i' = \iota(\lambda')$, where $\iota : \Lambda_L \to \{1, \cdots, M\}$ is any fixed bijection from the general form index set Λ_L to the standard index set $\{1, \cdots, M\}$. Following the natural sequential indexing of the wavelet basis functions, in the sequel we will consider the bijection

$$i = \iota(\lambda) = \iota((\ell, k)) = 1 + \sum_{m=0}^{\ell-1} 2^m + k = 2^\ell + k. \tag{14}$$

The Lubich-Galerkin method leads to a block Toeplitz lower triangular linear system of the form

$$\sum_{j=0}^{n} \boldsymbol{V}^{\mathcal{W},n-j} \boldsymbol{d}^j = \boldsymbol{g}^{\mathcal{W},n} \tag{15}$$

in the unknown vectors $\boldsymbol{d}^n = (d^n_\lambda)_{\lambda \in \Lambda_L}$, for $n = 0, \cdots, N$.

By combining (12) with (8), we obtain

$$\boldsymbol{V}^{\mathcal{W},n}_{\lambda,\lambda'} \approx \frac{\rho^{-n}}{\mathcal{R}} \sum_{m=0}^{\mathcal{R}-1} \tilde{c}^{\mathcal{W}}_{\lambda,\lambda'}(m) e^{-\imath 2\pi n m/\mathcal{R}}, \tag{16}$$

where

$$\tilde{c}^{\mathcal{W}}_{\lambda,\lambda'}(m) = \int_0^{2\pi} \int_0^{2\pi} \widehat{G}\left(r(\theta,\sigma), \frac{\gamma(\rho e^{\imath 2\pi m/\mathcal{R}})}{\Delta_t}\right) \psi_\lambda(\theta)\psi_{\lambda'}(\sigma)\ |\boldsymbol{\eta}'(\theta)|\,|\boldsymbol{\eta}'(\sigma)|\ d\theta\, d\sigma. \tag{17}$$

For fixed $\lambda, \lambda' \in \Lambda_L$, we denote by $c^{\mathcal{W}}_{\lambda,\lambda'} = (c^{\mathcal{W}}_{\lambda,\lambda'}(n))$ the Discrete Fourier Transform (DFT) of $\tilde{c}^{\mathcal{W}}_{\lambda,\lambda'}$:

$$c^{\mathcal{W}}_{\lambda,\lambda'}(n) = \sum_{m=0}^{\mathcal{R}-1} \tilde{c}^{\mathcal{W}}_{\lambda,\lambda'}(m) e^{-\imath 2\pi n \frac{m}{\mathcal{R}}}.$$

The entries of index λ, λ' of the matrices $\boldsymbol{V}^{\mathcal{W},n}$ are then approximated by

$$\boldsymbol{V}^{\mathcal{W},n}_{\lambda,\lambda'} \approx \frac{\rho^{-n}}{\mathcal{R}} c^{\mathcal{W}}_{\lambda,\lambda'}(n), \tag{18}$$

and can be efficiently computed, simultaneously for all $n = 0, \cdots, \mathcal{R}-1$, by an FFT algorithm, with a complexity of order $\mathcal{R} \log \mathcal{R}$.

The stability and convergence of the proposed method follow from Theorem 5.4 in [2], proved by Lubich for the 3D case but, as asserted in [4], valid in the 2D case as well. For the convenience of the reader, we report here the statement of the theorem.

Theorem 2.1. *Let $X_{\Delta_x} \subset L^2(\Gamma)$ be a family of finite dimensional approximation spaces of order $m+1$, that is*

$$\inf_{\psi_{\Delta_x} \in X_{\Delta_x}} \|\psi - \psi_{\Delta_x}\|_{H^{-1/2}(\Gamma)} \le C \Delta_x^{m+3/2} \|\psi\|_{H^{m+1}(\Gamma)}, \qquad \textit{for all } \psi \in H^{m+1}(\Gamma).$$

Let the time discretization method be A-stable and of order p such that $\gamma(z)$ has no poles on the unit circle. Then, for smooth compatible data g, the fully discrete method (Galerkin in space and convolution quadrature in time) admits a unique solution $\varphi^n_{\Delta_x}$ and is unconditionally stable and convergent with optimal order:

$$\|\varphi(\cdot,t_n)-\varphi^n_{\Delta_x}\|_{H^{-1/2}(\Gamma)}=\mathcal{O}(\Delta_t^p)+\mathcal{O}(\Delta_x^{m+3/2}), \tag{19}$$

uniformly over bounded intervals.

As asserted in [4], for the BDF2 method, estimate (19) is valid when assuming in (4) $g\in H^6((0,T),H^{1/2}(\Gamma))$ having all its derivatives up to order 4 vanishing at $t=0$.

We observe that Theorem 2.1 can be applied in the wavelet discretization context, which we will briefly introduce in the following section and whose properties we will exploit in the design of our method.

2.3. Wavelets, matrix compression and time downsampling strategy

2.3.1. Wavelets

Let us now put ourselves in the wavelet framework. We consider the family of spaces $\{V_L\}_{L\geq 0}$ and we assume that it forms a so called Multiresolution Analysis of $L^2(0,2\pi)$ (see [26]), endowed with a compactly supported wavelet basis. More precisely, we assume:

1. *nestedness*: $V_L\subset V_{L+1}$;
2. *scale invariance*: $f(\theta)\in V_L \Leftrightarrow f(2\theta)\in V_{L+1}$;
3. *translation invariant basis*: there exists a compactly supported function $\phi\in L^2(\mathbb{R})$ such that, setting
$$\phi_{L,k}(\theta)=\frac{2^{L/2}}{\sqrt{2\pi}}\sum_{m\in\mathbb{Z}}\phi\left(2^L\frac{\theta}{2\pi}+2^Lm-k\right),\ k=0,\dots,2^L-1, \tag{20}$$
the set $\{\phi_{L,k},\ k=0,\dots,2^L-1\}$ forms a basis for V_L;
4. *wavelet basis*: there exists a compactly supported functions $\psi\in L^2(\mathbb{R})$ such that, setting
$$\psi_{\ell,k}(\theta)=\frac{2^{\ell/2}}{\sqrt{2\pi}}\sum_{m\in\mathbb{Z}}\psi\left(2^\ell\frac{\theta}{2\pi}+2^\ell m-k\right),\ \ell\geq 0,\ k=0,\dots,2^\ell-1, \tag{21}$$
the set
$$\{\psi_\lambda,\ \lambda=(\ell,k)\in\Lambda_L\} \tag{22}$$
with
$$\Lambda_L=\{(-1,0)\}\cup\{(\ell,k),\ell=0,\dots,L-1,\ k=0,\dots,2^\ell-1\},\qquad \psi_{-1,0}:=\phi_{0,0},$$
forms also a basis for the space V_L.

According to this structure, the space V_L has then two bases, the basis $\{\phi_{L,k},\ k = 0,\dots,2^L-1\}$ and the basis (22). We can think of the former as a nodal (or nodal type) basis, for which the degrees of freedom essentially correspond to the nodes of an underlying uniform mesh with mesh size $2\pi/2^L$. The latter is referred to as the *wavelet basis* and has a multilevel structure, where each basis function corresponds to a *level* ℓ and, within the same level, to a *position* $k/2^\ell$. It turns out that such a basis has numerous interesting properties, which are not shared by the nodal basis and which allow to design new efficient numerical schemes. A fast change of basis algorithms, with a fast inverse, is available and is referred to as *Fast Wavelet Transform* (FWT).

Without loss of generality we can assume that $\operatorname{supp}\phi \subseteq [-R,R]$ and $\operatorname{supp}\psi \subseteq [-\tilde{R},\tilde{R}]$ for some positive R and $\tilde{R}$, and that ϕ and ψ are normalized in L^2, so that $\|\phi\|_{L^2(\mathbb{R})} = \|\psi\|_{L^2(\mathbb{R})} = 1$. Observe that, by construction, the functions $\phi_{L,k}$ and $\psi_{\ell,k}$ are 2π-periodic. Moreover, they satisfy the following translation invariance property

$$\phi_{L,k}(x) = \phi_{L,0}(x - 2\pi k 2^{-L}), \qquad \psi_{\ell,k}(x) = \psi_{\ell,0}(x - 2\pi k 2^{-\ell}).$$

Observe also that, when restricted to a (suitably offset) interval of length 2π, the support of $\psi_{\ell,k}$ is centered around $2\pi\, k2^{-\ell}$ and its measure is (less than or equal to) $2\pi \min\{1, 2\tilde{R}\, 2^{-\ell}\}$.

The spaces V_L will satisfy an approximation property, which, in this framework, is usually expressed by assuming that the function ϕ satisfies a *polynomial reconstruction property*: for all p polynomial of degree less than or equal to m, there exists a sequence b_n such that for all $\theta \in \mathbb{R}$

$$p(\theta) = \sum_{n\in\mathbb{Z}} b_n \phi(\theta - n). \tag{23}$$

Observe that, while the above sum is taken over the whole $\mathbb{Z}$, it reduces, for any given value of θ, to a finite sum, due to the compactness of the support of ϕ. Property (23) implies that V_L satisfies the assumption of Theorem 2.1 with $\Delta_x = 2\pi 2^{-L}$. Then, by using the convolution quadrature formula associated to the BDF method of order $p = 2$, for our method we derive the following estimate:

$$\|\varphi(\cdot,t_n) - \varphi_L^n\|_{H^{-1/2}(\Gamma)} = \mathcal{O}(\Delta_t^2) + \mathcal{O}(\Delta_x^{m+3/2}). \tag{24}$$

Furthermore, given any target polynomial degree $\widetilde{m}$, and denoting by $\mathbb{P}_{\widetilde{m}}$ the space of polynomials of degree less than or equal to $\widetilde{m}$, the function ψ can be chosen in such a way that

$$\int_{\mathbb{R}} \psi(\theta) p(\theta)\, d\theta = 0 \quad \forall\, p \in \mathbb{P}_{\widetilde{m}-1}. \tag{25}$$

A consequence of this property, that will play a key role in the forthcoming analysis, is the following Proposition.

Proposition 2.2. *Let $f \in L^2_{\mathrm{loc}}(\mathbb{R})$ be a function satisfying the periodicity condition $f(\sigma + 2\pi) = f(\sigma)$ for all $\sigma \in \mathbb{R}$. Then, for $\lambda = (\ell,k)$ we have*

$$\left| \int_0^{2\pi} f(\sigma)\psi_\lambda(\sigma)\, d\sigma \right| \lesssim \inf_{p\in\mathbb{P}_{\widetilde{m}-1}} \|f - p\|_{L^2(\Omega_\lambda)}$$

with $\widehat{\Omega}_\lambda = [2\pi 2^{-\ell}(k-\tilde{R}), 2\pi 2^{-\ell}(k-\tilde{R})]$.

Proof. We can write, for p polynomial of degree lower than $\widetilde{m}$

$$\begin{aligned}\int_0^{2\pi} f(\sigma)\psi_\lambda(\sigma)\,d\sigma &= \frac{2^{\ell/2}}{\sqrt{2\pi}}\int_0^{2\pi} f(\sigma)\sum_{m\in\mathbb{Z}}\psi\left(2^\ell\frac{\sigma}{2\pi}+2^\ell m-k\right)d\sigma \\ &= \frac{2^{\ell/2}}{\sqrt{2\pi}}\sum_{m\in\mathbb{Z}}\int_{2\pi m}^{2\pi(m+1)} f(\theta-2\pi m)\psi\left(2^\ell\frac{\theta}{2\pi}-k\right)d\theta \\ &= \frac{2^{\ell/2}}{\sqrt{2\pi}}\int_{\mathbb{R}}(f(\theta)-p(\theta))\,\psi\left(2^\ell\frac{\theta}{2\pi}-k\right)d\theta \le \|f-p\|_{L^2(\widehat{\Omega}_\lambda)},\end{aligned}$$

where we used (25), and where the last inequality derives by restricting the integral to $\widehat{\Omega}_\lambda$, which we can do thanks to the definition of the support of ψ, and using the Schwartz inequality (recall that $\|\psi\|_{L^2(\mathbb{R})}=1$). The arbitrariness of p implies the thesis. □

Remark 2.3. *A very large number of discretization spaces fall in the framework that we just described. While many of them have been newly constructed in order to build bases of the form (22), we also find examples from the classical approximation theory, such as piece-wise linear functions, or more generally, B-splines. In addition to the standard nodal basis, the wavelet framework provides such approximation spaces with a wavelet basis, whose properties can be exploited in the design of numerical schemes. As an example, we will consider here the space V_L of piecewise linear functions on an uniform grid with step size $2\pi 2^{-L}$, for which we give the explicit expression of ϕ and ψ:*

$$\phi(t) = \begin{cases} 1-|t|, & |t|\le 1 \\ 0, & else \end{cases}$$

and

$$\psi(t) = \frac{\sqrt{2}}{4}\phi(2t+1)+\frac{\sqrt{2}}{2}\phi(2t)-\frac{3\sqrt{2}}{2}\phi(2t-1)+\frac{\sqrt{2}}{2}\phi(2t-2)+\frac{\sqrt{2}}{4}\phi(2t-3),$$

associated to what, in wavelet terminology, are usually referred to as biorthogonal bior2.2 *compactly supported wavelet functions (see [27] and Figure 1). In Figure 2 we show some periodic basis functions for some levels ℓ. We remark that this wavelet basis has been chosen in the numerical tests. For such an example we have $m=1$ and $\widetilde{m}=2$.*

2.3.2. Matrix compression

It is well known that wavelet bases are an efficient tool for the treatment of integral operators ([28]). In particular, the use of wavelets allows to apply compression strategies which entail a high sparsification of the associated matrices. Indeed these, thanks to the localization and vanishing moments properties of the basis functions, turn out to have a relevant number of negligible entries at each time step. Neglecting the entries that are below

a prescribed tolerance allows to replace the dense system of the conventional BEM into a sparse one.

The key result at the basis of wavelet methods for the solution of boundary integral equations is the following theorem (see [28]), where we recall that $r(\theta,\sigma) = |\boldsymbol{\eta}(\theta) - \boldsymbol{\eta}(\sigma)|$.

Theorem 2.4. *Assume that the operator $\mathcal{K}$ defined by*

$$\mathcal{K}u(\sigma) = \int_0^{2\pi} K(r(\theta,\sigma))u(\theta)\ |\boldsymbol{\eta}'(\theta)|\ d\theta,$$

verifies

1. *$\mathcal{K}$ is a bounded operator, with bounded inverse, from $H^{-1/2}(\Gamma)$ to $H^{1/2}(\Gamma)$, that is, there exist two positive constants C_* and C^*, such that for all $u \in H^{-1/2}(\Gamma)$*

$$C_*\|u\|_{H^{-1/2}(\Gamma)} \le \|\mathcal{K}u\|_{H^{1/2}(\Gamma)} \le C^*\|u\|_{H^{-1/2}(\Gamma)};$$

2. *letting $\tilde{K}(\eta,\sigma) = K(r(\eta,\sigma))$, $\tilde{K}$ is smooth except on the diagonal, and for all $\alpha,\beta \ge 0$ with $\alpha+\beta > 0$ there exists a positive constant $\tilde{C}(\alpha,\beta)$, such that*

$$|\partial_x^\alpha \partial_y^\beta \tilde{K}(x,y)| \le \tilde{C}(\alpha,\beta)|x-y|^{|\alpha|+|\beta|}.$$

Then, setting

$$c_{\lambda,\lambda'} = \int_0^{2\pi}\int_0^{2\pi} K(r(\theta,\sigma))\psi_{\lambda'}(\theta)\psi_\lambda(\sigma)\ |\boldsymbol{\eta}'(\sigma)|\ |\boldsymbol{\eta}'(\theta)|\ d\theta\, d\sigma,$$

and denoting by Ω_λ the support of ψ_λ and by Ω_λ^S its singular support (that is the set of those point where ψ_λ is not smooth, [28]), for $\lambda = (\ell,k)$ and $\lambda' = (\ell',k')$ we have the following two bounds.

1. *Whenever $\delta(\Omega_\lambda,\Omega_{\lambda'}) = \inf_{\sigma\in\Omega_\lambda,\theta\in\Omega_{\lambda'}} |\boldsymbol{\eta}(\theta) - \boldsymbol{\eta}(\sigma)| \ge \kappa 2^{-\min\{\ell,\ell'\}}$ it holds*

$$|c_{\lambda,\lambda'}| \lesssim \frac{2^{-(\ell+\ell')(1/2+\widetilde{m})}}{\delta(\Omega_\lambda,\Omega_{\lambda'})^{2\widetilde{m}}},$$

   (the constant in the inequality depending on κ).
2. *Otherwise, assuming, without loss of generality, that $\ell' < \ell$, it holds*

$$|c_{\lambda,\lambda'}| \lesssim \frac{2^{-\ell(1+\widetilde{m})}2^{\ell'}}{\delta(\Omega_\lambda,\Omega_{\lambda'}^S)^{\widetilde{m}-1}},$$

   with $\delta(\Omega_\lambda,\Omega_{\lambda'}^S)$ defined analogously to $\delta(\Omega_\lambda,\Omega_{\lambda'})$. The implicit constant in the inequalities are independent of λ and λ', while they generally depend on K, and on the parametrization $\boldsymbol{\eta}$ of Γ.

The aim of this paper is to combine these compression techniques with Lubich's convolution quadrature method. Theorem 2.4, for the choice of $K = K^n$ with

$$K^n(r) = \sum_{m=0}^{\mathcal{R}-1} \widehat{G}\left(r, \frac{\gamma(\rho e^{\imath 2\pi m/\mathcal{R}})}{\Delta_t}\right) e^{-\imath 2\pi n m/\mathcal{R}},$$

tells us that the matrices $\boldsymbol{V}^{\mathcal{W},n}$ can be approximated bely sparse matrices for each fixed n. Actually, it turns out that such matrices become sparser and sparser as n increases. Indeed, as we will see, when using wavelet bases, the sequence $\{\boldsymbol{V}^{\mathcal{W},n}_{\lambda,\lambda'}\}_{n=0}^{\mathcal{R}-1}$ (with λ, λ' fixed) of the matrix entries displays, as n increases, a rapid decay to zero and, from a certain point on, its elements assume negligible values. By taking advantage of this property, we will apply a numerical strategy that allows to retrieve the significant elements, up to a prescribed tolerance, by using a downsampled FFT. Such a strategy (to which we will refer as *time downsampling strategy* and that we will describe in Section 2.3.3) will considerably reduce the numerical effort needed for the computation of the matrix elements (18).

As we will see, in order to take full advantage of the downsampling strategy, we need, however, to estimate *a priori*, for each couple λ, λ', the value of $\bar{n} = \bar{n}(\lambda, \lambda')$, such that $|\boldsymbol{V}^{\mathcal{W},n}_{\lambda,\lambda'}| \leq \varepsilon$ for $n \geq \bar{n}$, ε being a prescribed tolerance. To this aim, we need to provide an a priori bound on how $|\boldsymbol{V}^{\mathcal{W},n}_{\lambda,\lambda'}|$ decreases as n increases. While Theorem 2.4 indicates that we can indeed hope in a compression of the matrix, it does not yield an explicit information on the dependence on n of such a compression, which we need in order to choose the parameter $\bar{n}$. Due to the quite cumbersome definition of the kernel K^n, which involves a Laplace transform, combined with the BDF2 scheme and a Discrete Fourier Transform, making such a dependence explicit is not immediately feasible.

In order to propose a criterion for the choice of $\bar{n}$, we instead recall that the following Lemma holds (see [16]).

Lemma 2.5. *There exists $\bar{t}$ depending on Δ_t such that for all $n > (\bar{t} + \delta)\Delta_t$*

$$|\omega_n(\Delta_t; s) - \Delta_t G(s, n\Delta_t)| \leq C(\delta, s)\Delta_t^3$$

with $\delta > 0$ arbitrary and $C(\delta, s)$ a positive constant depending on s and δ.

Using such a lemma, we easily see that, for all $n > (\bar{t} + \delta)/\Delta_t$ we have that

$$\boldsymbol{V}^{\mathcal{W},n}_{\lambda,\lambda'} = \Delta_t \int_0^{2\pi}\int_0^{2\pi} G(r(\theta,\sigma), n\Delta_t)\psi_\lambda(\theta)\psi_{\lambda'}(\sigma)\ |\boldsymbol{\eta}'(\theta)|\,|\boldsymbol{\eta}'(\sigma)|\ d\theta\, d\sigma + \boldsymbol{E}^n_{\lambda,\lambda'}.$$

with

$$\begin{aligned} |\boldsymbol{E}^n_{\lambda,\lambda'}| &\leq \Delta_t^3 \int_0^{2\pi}\int_0^{2\pi} C(\delta, r(\theta,\delta))|\psi_\lambda(\theta)||\psi_{\lambda'}(\sigma)|\ |\boldsymbol{\eta}'(\theta)|\,|\boldsymbol{\eta}'(\sigma)|\ d\theta\, d\sigma \\ &\leq 2^{-(\ell+\ell')/2} C(\delta)\Delta_t^3. \end{aligned}$$

This suggests to look for $\bar{n}$ by exploiting an a priori estimate on

$$\widetilde{\boldsymbol{V}}^{\mathcal{W},n}_{\lambda,\lambda'} = \Delta_t \int_0^{2\pi}\int_0^{2\pi} G(r(\theta,\sigma), n\Delta_t)\psi_\lambda(\theta)\psi_{\lambda'}(\sigma)\ |\boldsymbol{\eta}'(\theta)|\,|\boldsymbol{\eta}'(\sigma)|\ d\theta\, d\sigma.$$

We then start to prove the following Proposition.

Proposition 2.6. *It holds that*

$$|\widetilde{\boldsymbol{V}}^{\mathcal{W},n}_{\lambda,\lambda'}| \lesssim \Delta_t 2^{-\widetilde{m}\min\{\ell,\ell'\}} 2^{-(\ell+\ell')/2} \max_{(\theta,\sigma)\in\Omega_\lambda\times\Omega_{\lambda'}} \max_{|\alpha|=\widetilde{m}} |D^\alpha G_n(\theta,\sigma)| \tag{26}$$

where Ω_λ is the support of ψ_λ.

Proof. In order to bound $|\widetilde{\boldsymbol{V}}^{\mathcal{W},n}_{\lambda,\lambda'}|$ we can use (a two dimensional version of) Proposition 2.2. We then have

$$\left|\int_0^{2\pi}\int_0^{2\pi} G(r(\theta,\sigma), n\Delta_t)\psi_\lambda(\theta)\psi_{\lambda'}(\sigma)\ |\boldsymbol{\eta}'(\theta)|\,|\boldsymbol{\eta}'(\sigma)|\ d\theta\, d\sigma\right| \lesssim \inf_{p\in\mathbb{P}_{\widetilde{m}-1}} \|G_n - p\|_{L^2(\widehat{\Omega}_\lambda\times\widehat{\Omega}_{\lambda'})}. \tag{27}$$

where the 2π bi-periodic function $G_n \in L^2_{\mathrm{loc}}(\mathbb{R}^2)$ is defined by

$$G_n(\theta,\sigma) := G(r(\theta,\sigma), n\Delta_t)\,|\boldsymbol{\eta}'(\theta)|\,|\boldsymbol{\eta}'(\sigma)|\,,$$

(by abuse on notation we also denote by $\boldsymbol{\eta}$ the 2π periodic function coinciding with $\boldsymbol{\eta}$ in $[0,2\pi)$). We bound the right hand side of (27) by a standard argument (see [29]) as

$$\begin{aligned}\inf_{p\in\mathbb{P}_{\widetilde{m}-1}} \|G_n - p\|_{L^2(\widehat{\Omega}_\lambda\times\widehat{\Omega}_{\lambda'})} &\lesssim 2^{-\widetilde{m}\min\{\ell,\ell'\}} |G_n|_{\widetilde{m},\widehat{\Omega}_\lambda\times\widehat{\Omega}_{\lambda'}} \\ &\lesssim 2^{-\widetilde{m}\min\{\ell,\ell'\}} 2^{-(\ell+\ell')/2} \max_{(\theta,\sigma)\in\widehat{\Omega}_\lambda\times\widehat{\Omega}_{\lambda'}} \max_{|\alpha|=\widetilde{m}} |D^\alpha G_n(\theta,\sigma)|. \end{aligned}\tag{28}$$

We finally obtain (26) by observing that, as G_n is a 2π bi-periodic function and as $\Omega_\lambda = \widehat{\Omega}_\lambda \cap [0,2\pi]$ we can replace the maximum over $\widehat{\Omega}_\lambda\times\widehat{\Omega}_{\lambda'}$ with the maximum over $\Omega_\lambda\times\Omega_{\lambda'}$. □

If we assume that the parametrization of Γ is chosen in such a way that $|\boldsymbol{\eta}'(\theta)| = 1$ for all $\theta\in[0,2\pi]$, for $r = r(\theta,\sigma) < c\,n\Delta_t$, it holds:

$$\frac{\partial^2 G_n}{\partial\theta^2}(\theta,\sigma) = \frac{1}{2\pi}\left[\left(\frac{c^{-2}}{((n\Delta_t)^2 - c^{-2}r^2)^{3/2}} + \frac{3c^{-4}r^2}{((n\Delta_t)^2 - c^{-2}r^2)^{5/2}}\right) r_\theta^2 + \frac{c^{-2}r}{((n\Delta_t)^2 - c^{-2}r^2)^{3/2}} r_{\theta\theta}\right].$$

The computation of the derivatives $\dfrac{\partial^2 G_n}{\partial\sigma^2}$ and $\dfrac{\partial^2 G_n}{\partial\theta\partial\sigma}$ yields an analogous expression. Therefore, setting

$$M_{\lambda,\lambda'} := \max_{(\theta,\sigma)\in\Omega_\lambda\times\Omega_{\lambda'}} r(\theta,\sigma),$$

and

$$D_{1,\Gamma} := \max_{|\alpha|=1}\max_{(\theta,\sigma)\in\Gamma\times\Gamma} |D^\alpha r(\theta,\sigma)|, \qquad D_{2,\Gamma} := \max_{|\alpha|=2}\max_{(\theta,\sigma)\in\Gamma\times\Gamma} |D^\alpha r(\theta,\sigma)|,$$

we get the following corollary.

Corollary 2.7. *If the parametrization* $\boldsymbol{\eta}$ *satisfies* $|\boldsymbol{\eta}'(\theta)| = 1$ *for all* $\theta \in [0, 2\pi]$ *we have*

$$|\widetilde{\boldsymbol{V}}^{\mathcal{W},n}_{\lambda,\lambda'}| \lesssim \frac{1}{2\pi}\Delta_t 2^{-\widetilde{m}\min\{\ell,\ell'\}}2^{-(\ell+\ell')/2}\left[\left(\frac{c^{-2}}{((n\Delta_t)^2 - c^{-2}M^2_{\lambda,\lambda'})^{3/2}} + \frac{3c^{-4}M^2_{\lambda,\lambda'}}{((n\Delta_t)^2 - c^{-2}M^2_{\lambda,\lambda'})^{5/2}}\right) D^2_{1,\Gamma} + \frac{c^{-2}M_{\lambda,\lambda'}}{((n\Delta_t)^2 - c^{-2}M^2_{\lambda,\lambda'})^{3/2}} D_{2,\Gamma}\right]. \quad (29)$$

Given now λ, λ' and a threshold ε, the idea is to look for $\bar{n}$ such that for $n > \bar{n}$ we have that the value of the right hand side of (29) is less or equal than ε. The solution will of course depend on the shape of Γ via the derivatives of r with respect to θ and σ.

The problem of finding the values n such that the right hand side of (29) is smaller than ε given can not be solved analytically. However, for a fixed time discretization, once the quantities involved therein are evaluated, an upper bound for $\bar{n}$ can be numerically computed after further bounding the right hand side of (29) with a simpler expression. Just to make an example, let us consider the case of Γ being a circle of radius R. In this case, it is not difficult to obtain that the terms $D_{1,\Gamma}$ and $D_{2,\Gamma}$ can be bounded by R. Therefore, an estimate for $\bar{n}$ follows by exploiting the following estimate

$$|\widetilde{\boldsymbol{V}}^{\mathcal{W},n}_{\lambda,\lambda'}| \lesssim \frac{1}{2\pi}\Delta_t 2^{-\widetilde{m}\min\{\ell,\ell'\}}2^{-(\ell+\ell')/2}\left[\left(\frac{c^{-2}(R^2 + M_{\lambda,\lambda'}R)}{((n\Delta_t)^2 - c^{-2}M^2_{\lambda,\lambda'})^{3/2}} + \frac{3c^{-4}M^2_{\lambda,\lambda'}R^2}{((n\Delta_t)^2 - c^{-2}M^2_{\lambda,\lambda'})^{5/2}}\right)\right]. \quad (30)$$

We finally remark that estimate (30) will be used in the numerical tests of Section 3.

Remark 2.8. *Observe that, while Lemma 2.5 yields that* $\boldsymbol{V}^{\mathcal{W},n}_{\lambda,\lambda'} \sim \widetilde{\boldsymbol{V}}^{\mathcal{W},n}_{\lambda,\lambda'}$ *only for* $n > (\bar{t}+\delta)/\Delta_t$*, in using the right hand side of* (29) *to estimate* $\bar{n}$*, we disregard such a restriction. The numerical tests confirm that this does not lead to an under estimation of* $\bar{n}$*, which would negatively affect the accuracy of the compression scheme, but rather to an over estimation of such a parameter (which implies a higher cost of the resulting scheme). Of course, it would be desirable to have a strategy leading to a sharper estimate of* $\bar{n}$*, which would result in a cheaper but still accurate numerical scheme.*

2.3.3. Time downsampling strategy

From estimate (29) we deduce that, for fixed $\lambda = (\ell, k)$, $\lambda' = (\ell', k')$, the matrix elements have a rapid decay to zero for increasing values of the time instants $t_n = n\Delta_t$. Moreover, this behaviour is influenced both by the levels ℓ and ℓ' of the two indexes, and by the wave propagation speed c. Indeed, as we will notice in the forthcoming numerical tests, for a fixed tolerance ε, the higher the values of ℓ, ℓ' and c are, the smaller the value of $\bar{n}$ is. If, for given λ, λ', we avoid the computation of this negligible values, which we can set to 0,

and settle for only building the matrix entries $\boldsymbol{V}^{\mathcal{W},n}_{\lambda,\lambda'}$ for $n \leq \bar{n}$, we can exploit the discrete Fourier transform properties to write down a time downsampled procedure, leading to both computational and memory saving.

In order to describe the time downsampling procedure, we start by recalling the relation between the FFT and its Inverse Discrete Fourier Transform (IDFT). In particular, the coefficients in (17) can be retrieved by:

$$\tilde{c}^{\mathcal{W}}_{\lambda,\lambda'}(n) = \frac{1}{\mathcal{R}} \sum_{m=0}^{\mathcal{R}-1} c^{\mathcal{W}}_{\lambda,\lambda'}(m) e^{\imath 2\pi mn/\mathcal{R}}, \qquad n = 0, \cdots, \mathcal{R}-1. \tag{31}$$

Let us denote by $\overline{\mathcal{R}}_{\lambda,\lambda'}$, with $\mathcal{R} = Q\overline{\mathcal{R}}_{\lambda,\lambda'}$, the smallest integer, submultiple of $\mathcal{R}$, such that $|\boldsymbol{V}^{\mathcal{W},n}_{\lambda,\lambda'}| \leq \varepsilon$ for $n \geq \overline{\mathcal{R}}_{\lambda,\lambda'}$. According to (18), we have $c^{\mathcal{W}}_{\lambda,\lambda'}(n) \approx \rho^n \mathcal{R} \boldsymbol{V}^{\mathcal{W},n}_{\lambda,\lambda'}$, and hence (since $\rho < 1$) we will have that $|c^{\mathcal{W}}_{\lambda,\lambda'}(n)| \leq \mathcal{R}\varepsilon =: \tilde{\varepsilon}$, for $n \geq \overline{\mathcal{R}}_{\lambda,\lambda'}$.

We rewrite (31) as

$$\begin{aligned} \tilde{c}^{\mathcal{W}}_{\lambda,\lambda'}(n) &= \frac{1}{\mathcal{R}} \sum_{m=0}^{\overline{\mathcal{R}}_{\lambda,\lambda'}-1} c^{\mathcal{W}}_{\lambda,\lambda'}(m) e^{\imath 2\pi mn/\mathcal{R}} + \frac{1}{\mathcal{R}} \sum_{m=\overline{\mathcal{R}}_{\lambda,\lambda'}}^{\mathcal{R}-1} c^{\mathcal{W}}_{\lambda,\lambda'}(m) e^{\imath 2\pi mn/\mathcal{R}} \\ &=: \tilde{c}^{\mathcal{W},1}_{\lambda,\lambda'}(n) + \tilde{c}^{\mathcal{W},2}_{\lambda,\lambda'}(n), \end{aligned}$$

with $|\tilde{c}^{\mathcal{W},2}_{\lambda,\lambda'}(n)| \leq (\mathcal{R} - \overline{\mathcal{R}}_{\lambda,\lambda'})/\mathcal{R}\, \tilde{\varepsilon} \leq \tilde{\varepsilon}$, for $n = 0, \cdots, \mathcal{R}-1$.

By neglecting this second small term and evaluating $\tilde{c}^{\mathcal{W},1}_{\lambda,\lambda'}(n)$ at $n = pQ$, with $p = 0, \cdots, \overline{\mathcal{R}}_{\lambda,\lambda'} - 1$, we get

$$\tilde{c}^{\mathcal{W},1}_{\lambda,\lambda'}(pQ) = \frac{1}{Q\overline{\mathcal{R}}_{\lambda,\lambda'}} \sum_{m=0}^{\overline{\mathcal{R}}_{\lambda,\lambda'}-1} c^{\mathcal{W}}_{\lambda,\lambda'}(m) e^{\imath 2\pi mp/\overline{\mathcal{R}}_{\lambda,\lambda'}}$$

from which, applying the DFT, we deduce

$$c^{\mathcal{W}}_{\lambda,\lambda'}(n) = Q \sum_{p=0}^{\overline{\mathcal{R}}_{\lambda,\lambda'}-1} \tilde{c}^{\mathcal{W},1}_{\lambda,\lambda'}(pQ) e^{-\imath 2\pi np/\overline{\mathcal{R}}_{\lambda,\lambda'}}, \qquad n = 0, \cdots, \overline{\mathcal{R}}_{\lambda,\lambda'} - 1. \tag{32}$$

Therefore, recalling that $\tilde{c}^{\mathcal{W}}_{\lambda,\lambda'} \approx \tilde{c}^{\mathcal{W},1}_{\lambda,\lambda'}$ (up to a term smaller than $\tilde{\varepsilon}$), according to equation (32), to get the significant coefficients $c^{\mathcal{W}}_{\lambda,\lambda'}(n)$ for $n = 0, \cdots, \overline{\mathcal{R}}_{\lambda,\lambda'}$, it is sufficient to compute the coefficients $\tilde{c}^{\mathcal{W}}_{\lambda,\lambda'}(pQ)$ only at the $\overline{\mathcal{R}}_{\lambda,\lambda'}$ downsampled values pQ, $0 \leq p \leq \overline{\mathcal{R}}_{\lambda,\lambda'} - 1$.

Combining (32) and (16), the computation of the significant matrix entries is therefore given by the following formula:

$$\boldsymbol{V}^{\mathcal{W},n}_{\lambda,\lambda'} \approx \frac{\rho^{-n}}{\mathcal{R}} Q \sum_{p=0}^{\overline{\mathcal{R}}_{\lambda,\lambda'}-1} \tilde{c}^{\mathcal{W}}_{\lambda,\lambda'}(pQ) e^{-\imath 2\pi np/\overline{\mathcal{R}}_{\lambda,\lambda'}}, \qquad n = 0, \cdots, \overline{\mathcal{R}}_{\lambda,\lambda'} - 1. \tag{33}$$

It is worthwhile noting that the complexity of the downsampled FFT algorithm (33) is of order $\overline{\mathcal{R}}_{\lambda,\lambda'} \log \overline{\mathcal{R}}_{\lambda,\lambda'}$. As we will see in the numerical tests, for many couples (λ, λ') it turns out that $\overline{\mathcal{R}}_{\lambda,\lambda'} \ll \mathcal{R}$, and this allows a consistent computational cost reduction and, consequently, a high memory saving.

We remark that, when in (17) piece-wise linear nodal approximating basis functions are considered instead of the wavelet basis functions ψ_λ and $\psi_{\lambda'}$, a downsampling strategy could in principle be applied as well. However, in this case, it turns out that the associated coefficients obtained by the FFT are not negligible, and the cutoff procedure is not effective, as the next section shows.

2.3.4. On the efficiency of the downsampling strategy

To show the downsampling strategy efficiency, we consider here, as a benchmark example, Problem (1) where Γ is the circumference of radius $R = 1$ and the final time $T = 10$. The discretization parameters are $L = 8$, $N = 256$ and the threshold $\varepsilon = 1.0e - 10$.

In what follows, we refer to the approach where the piece-wise linear nodal basis $\{\phi_{L,i},\ i = 0, \cdots, M-1\}$ is used as the *standard approach*, and we will label it by the superscript $\mathcal{S}$. The associated coefficients, corresponding to formula (17) where ψ_λ and $\psi_{\lambda'}$ are replaced by $\phi_{L,i}$ and $\phi_{L,j}$ respectively, will be denoted by $\tilde{c}^{\mathcal{S}}_{i,j}$ and the matrix entries obtained by applying the FFT will be denoted by $\boldsymbol{V}^{\mathcal{S},n}_{i,j}$.

In Figures 3 and 4 we compare the cutoff effects for two choices of the velocity: $c = 1$ and $c = 343$ (speed in m/s of propagation of acoustic waves in air). In each figure, the left plots correspond to the behaviour in time of the coefficients $\tilde{c}^{\mathcal{S}}_{i,j}$ (standard approach, top row) and $\tilde{c}^{\mathcal{W}}_{\lambda,\lambda'}$ (wavelet approach, middle and bottom rows). The right plots represent the behaviour in time of the matrix entries $\boldsymbol{V}^{\mathcal{S},n}_{i,j}$ (standard approach, top row) and $\boldsymbol{V}^{\mathcal{W},n}_{\lambda,\lambda'}$ (wavelet approach, middle and bottom rows) computed by the FFT algorithm. The graphs in solid lines refer to the FFT algorithm applied to the whole set of time instants, while the bullets refer to the FFT algorithm applied to the downsampled time instants. To compare entries of the matrices belonging to the same row and column, we use (14) to obtain (i, j) from (λ, λ'). In particular, for $\lambda = (3, 1)$ and $\lambda' = (7, 13)$ we have $(i, j) = (9, 141)$; for $\lambda = (7, 0)$ and $\lambda' = (7, 123)$ we have $(i, j) = (128, 251)$.

As we can see, in the wavelet approach, the downsampling strategy reveals to be crucial since it significantly reduces the computation of $\tilde{c}^{\mathcal{W}}_{\lambda,\lambda'}$ at only $\overline{\mathcal{R}}_{\lambda,\lambda'} \ll \mathcal{R} = 512$ values. For example, for $\lambda = (3, 1)$ and $\lambda' = (7, 13)$, it results $\overline{\mathcal{R}}_{\lambda,\lambda'} = 64$ for $c = 1$ and $\overline{\mathcal{R}}_{\lambda,\lambda'} = 4$ for the higher velocity $c = 343$. For the wavelets associated to higher levels, $\overline{\mathcal{R}}_{\lambda,\lambda'}$ is even smaller,

as expected. For example, for $\lambda = (7, 0)$ and $\lambda' = (7, 123)$ it results $\overline{\mathcal{R}}_{\lambda,\lambda'} = 4$ for $c = 1$, and $\overline{\mathcal{R}}_{\lambda,\lambda'} = 1$ for $c = 343$.

On the contrary, in the standard approach, for $(i, j) = (9, 141)$, the downsampling strategy is not effective because the entries of $\boldsymbol{V}^{\mathcal{S},n}$ are all greater than the chosen threshold ε when n increases. Since the graph corresponding to $(i, j) = (128, 251)$ is very similar, we omit it.

Remark 2.9. *It is worth noting that, in order to apply the downsampled FFT algorithm, $\overline{\mathcal{R}}_{\lambda,\lambda'}$ must be a submultiple of $\mathcal{R}$ and, for this reason, some values $\boldsymbol{V}^{\mathcal{W},n}_{\lambda,\lambda'}$ are computed even if they are below the threshold ε (see for example Figure 3, middle and bottom plots on the right). To reduce further the storage, these coefficients will be dropped successively by applying an a posteriori cutoff strategy, based on the same ε.*

We also highlight that the time downsampling strategy produces matrices $\boldsymbol{V}^{\mathcal{W},n}$ of the final linear system (15) that, for each fixed value of n, turn out to be sparse. Therefore, the resolution of (15),

$$\boldsymbol{V}^{\mathcal{W},0}\boldsymbol{d}^n = \boldsymbol{g}^{\mathcal{W},n} - \sum_{j=0}^{n-1} \boldsymbol{V}^{\mathcal{W},n-j}\boldsymbol{d}^j, \quad n = 0, \cdots, N,$$

takes advantage of the fast sparse matrix-vector product, in particular for what concerns the computation of the right hand side term $\sum_{j=0}^{n-1} \boldsymbol{V}^{n-j,\mathcal{W}}\boldsymbol{d}^j$ for each $n = 0, \cdots, N$.

In the next section we present some numerical examples to test the Lubich-wavelet Galerkin approach presented in Sections 2.1 and 2.2, combined with the time downsampling strategy described in Section 2.3.

3. Numerical results

In the forthcoming numerical tests, we will compare the approach $\mathcal{W}$ with approach $\mathcal{S}$. By virtue of the considerations made in Section 2.3, for the approach $\mathcal{S}$ we will not apply the downsampling strategy, but only the a posteriori cut off of the matrix entries below the threshold ε. Finally, we will also consider the Lubich-wavelet Galerkin approach without time downsampling, where the matrices of the Toeplitz block linear system expressed in the wavelet basis are obtained from the matrices relative to the standard approach, by using the fast wavelet fransform FWT; this latter approach will be detailed later and will be labeled by the capital letter $\mathcal{T}$. In this case too, we will apply only the a posteriori cut off procedure.

For each approach we will compute the approximations of φ, solution of BIE (4), and of the associated single layer potential $u(\mathbf{x}, t)$, solution of Problem (1) and defined by (2), in a chosen point $\mathbf{x} \in \Omega^e$. For the convenience of the reader, we summarize the two main steps of the three algorithms $\mathcal{S}$, $\mathcal{T}$ and $\mathcal{W}$.

Algorithm $\mathcal{S}$

$\mathcal{S}1$: the matrix elements $\boldsymbol{V}_{i,j}^{\mathcal{S},n}$ are computed by using a Gauss-Legendre product quadrature rule in space applied to each interval of regularity of the piece-wise linear basis functions, and by performing the complete FFT algorithm in time.

$\mathcal{S}2$: subsequently, the a posteriori cut off is applied to set equal to zero the matrix entries such that $|\boldsymbol{V}_{i,j}^{\mathcal{S},n}| \leq \varepsilon$ for $n = 0, \cdots, N$.

Algorithm $\mathcal{T}$

$\mathcal{T}1$: the matrices $\boldsymbol{V}^{\mathcal{T},n}$ are retrieved by applying the FWT to $\boldsymbol{V}^{\mathcal{S},n}$. To this aim we use the Matlab function `wavedec`, which returns the wavelet decomposition of the piece-wise linear basis functions at the level L associated to the *bior2.2* wavelets. Storing these latter as columns of a square matrix $\boldsymbol{W}$, we compute $\boldsymbol{V}^{\mathcal{T},n}$ as

$$\boldsymbol{V}^{\mathcal{T},n} = \boldsymbol{W}^{-T}\boldsymbol{V}^{\mathcal{S},n}\boldsymbol{W}^{-1}, \quad n = 0, \cdots, N. \tag{34}$$

$\mathcal{T}2$: then, we apply the a posteriori cutting to the matrix entries such that $|\boldsymbol{V}_{\lambda,\lambda'}^{\mathcal{T},n}| \leq \varepsilon$, for $n = 0, \cdots, N$.

Algorithm $\mathcal{W}$

$\mathcal{W}1$: for a fixed threshold ε and for $\overline{\mathcal{R}}_{\lambda,\lambda'}$ obtained according to the estimate (29), such that $\mathcal{R} = Q\overline{\mathcal{R}}_{\lambda,\lambda'}$ and $|\boldsymbol{V}_{\lambda,\lambda'}^{\mathcal{W},n}| \leq \varepsilon$ for all $n \geq \overline{\mathcal{R}}_{\lambda,\lambda'}$, the matrix elements $\boldsymbol{V}_{\lambda,\lambda'}^{\mathcal{W},n}$ are computed by a Gauss-Legendre product quadrature rule in space applied to each interval of regularity of the wavelet basis functions. In time, we apply the downsampled FFT to the set of coefficients $\tilde{c}_{\lambda,\lambda'}^{\mathcal{W}}(pQ)$, with $p = 0, \cdots, \overline{\mathcal{R}}_{\lambda,\lambda'}$.

$\mathcal{W}2$: subsequently, the elements computed by the downsampled FFT that are below the threshold ε, are further set to zero by an a posteriori cutting based on the same ε (see Remark 2.9). The remaining entries are those actually stored.

We will denote by $\boldsymbol{\varphi}_{M,N}^{*}$ and $\boldsymbol{u}_{M,N}^{*}$, with $* \in \{\mathcal{S}, \mathcal{T}, \mathcal{W}\}$, the corresponding approximate solutions, obtained by using $M = 2^L$ subintervals for the discretization of $[0, 2\pi]$ and N subintervals for $[0, T]$.

We point out that, the a posteriori cutting applied in the approach $\mathcal{S}$ allows a low memory saving, in agreement with the remarks done in Section 2.2 about the time behaviour of the entries of $\boldsymbol{V}^{\mathcal{S},n}$. On the contrary, as expected, the a posteriori cutting applied in the approach $\mathcal{T}$ allows to retrieve the same sparse matrices obtained by combining $\mathcal{W}1$ and $\mathcal{W}2$. However, the cutoff procedure in $\mathcal{T}$ can be performed only after that the matrices $\boldsymbol{V}^{\mathcal{S},n}$ have been

computed and stored for all $n = 0, \cdots, N$. This drawback will be highlighted in Example 4, in which the approaches $\mathcal{S}$ and $\mathcal{T}$ can not be applied because of an out of memory, while the approach $\mathcal{W}$ can be performed. Therefore, $\mathcal{T}$ is here considered for a comparison with $\mathcal{W}$ and to evaluate the sharpness of the a priori estimate (29).

In the following examples, we take as reference solutions $\boldsymbol{\varphi}^{\mathcal{S}1}_{M_\mathrm{e},N_\mathrm{e}}$ and $\boldsymbol{u}^{\mathcal{S}1}_{M_\mathrm{e},N_\mathrm{e}}$ those obtained by the standard approach $\mathcal{S}1$ that uses piece-wise basis functions for the spatial approximation, and without applying any cutting. The discrete parameters M_e and N_e are properly chosen according to the numerical example to which they refer.

To test the accuracy and efficiency of the above described approaches $* \in \{\mathcal{S}, \mathcal{W}, \mathcal{T}\}$, we will consider:

1) in the plots, for a graphic comparison, the absolute errors

$$\mathrm{E}^*_{\boldsymbol{\varphi}}(\mathbf{x}, t) = \left|\boldsymbol{\varphi}^{\mathcal{S}1}_{M_\mathrm{e},N_\mathrm{e}}(\mathbf{x}, t) - \boldsymbol{\varphi}^*_{M,N}(\mathbf{x}, t)\right|$$

and

$$\mathrm{E}^*_{\boldsymbol{u}}(\mathbf{x}, t) = \left|\boldsymbol{u}^{\mathcal{S}1}_{M_\mathrm{e},N_\mathrm{e}}(\mathbf{x}, t) - \boldsymbol{u}^*_{M,N}(\mathbf{x}, t)\right|$$

associated to $\boldsymbol{\varphi}^*_{M,N}$ and $\boldsymbol{u}^*_{M,N}$, respectively;

2) in the tables, the maximum in time of the absolute errors

$$\mathrm{E}^*_{L^2,\boldsymbol{\varphi}} = \max_{t\in[0,T]} \|\boldsymbol{\varphi}^{\mathcal{S}1}_{M_\mathrm{e},N_\mathrm{e}}(\cdot, t) - \boldsymbol{\varphi}^*_{M,N}(\cdot, t)\|_{L^2(\Gamma)} \tag{35}$$

and

$$\mathrm{E}^*_{\infty,\boldsymbol{u}}(\mathbf{x}) = \max_{t\in[0,T]} \left|\boldsymbol{u}^{\mathcal{S}1}_{M_\mathrm{e},N_\mathrm{e}}(\mathbf{x}, t) - \boldsymbol{u}^*_{M,N}(\mathbf{x}, t)\right|; \tag{36}$$

3) the memory saving percentage

$$\mathrm{mem}^*(\%) = \left(1 - \frac{\mathrm{nz}^*}{M^2 N}\right) \cdot 100, \tag{37}$$

M^2N being the total number of matrix elements of the standard approach $\mathcal{S}1$, and nz^* the number of elements that are stored after the cutting.

The goal of the forthcoming numerical experiments is twofold: a) to investigate the sharpness of estimate (29); b) to apply the new proposed method to problems of type (1) and to test the downsampling procedure efficiency, not only in terms of memory saving, but also in terms of accuracy of the approximate solutions $\boldsymbol{\varphi}^*_{M,N}$ and $\boldsymbol{u}^*_{M,N}$.

For simplicity, the domain Ω^i that we consider in all the examples is a disc of radius $R = 1$. By choosing two reference velocities of the acoustic wave, $c = 1$ (low velocity) and $c = 343$ (high velocity), we observe the dependency on c of the discrete integral operators sparsity: it results that the higher the velocity is, the sparser the wavelet matrices are.

Out of a set of extensive numerical test, the results we report here have been selected to present significant benchmark examples. For each of them, we summarize its main goal.

- ▷ In Example 1 we compute the errors (35) and (36) for all the approaches $\mathcal{S}$, $\mathcal{W}$ and $\mathcal{T}$ and report the global convergence rate with respect to the space and time refinement. Moreover, we show the sparsity pattern of the associated matrices at fixed time steps. Finally, we give the computational complexity $\sum_{\lambda,\lambda'} \overline{\mathcal{R}}_{\lambda,\lambda'} \log \overline{\mathcal{R}}_{\lambda,\lambda'}$ and memory saving, for both low and high velocities and by varying the threshold ε.
- ▷ In Example 2 we present a problem for which a coarse space discretization and a large threshold are sufficient to obtain an accurate solution with a high memory saving by the wavelet approaches $\mathcal{T}$ and $\mathcal{W}$, contrarily to the standard one $\mathcal{S}$.
- ▷ In Example 3 we consider a problem for which a fine space discretization and a small threshold are necessary to obtain an accurate solution for all the approaches. Despite the small value of ε the wavelet approaches $\mathcal{T}$ and $\mathcal{W}$ allow a very high memory saving, contrarily to the standard one $\mathcal{S}$.
- ▷ Finally, the last numerical test in Example 4 shows that the new approach $\mathcal{W}$ allows to deal with large scale problems, for which $\mathcal{S}$ and $\mathcal{T}$ can not be applied.

We remark that the implementation at the current stage of development does not employ optimized libraries or parallelization, and it is based on a standard (i.e., sequential) Matlab code.

Example 1. We consider Problem (1) with

$$g(\mathbf{x}, t) = t^4 e^{-2t} \cos(x_1^2 + 2x_2^2)$$

and $T = 10$. In Tables 1 and 2 we report the errors (35) and (36) for the approaches $\mathcal{S}$ and $\mathcal{T}$, and the corresponding estimated order of convergence (EOC), for $\varepsilon = 1.0e-10$ and velocities $c = 1$ and $c = 343$, respectively. In (35) and (36) the reference solutions $\boldsymbol{\varphi}^{\mathcal{S}_1}_{M_\mathrm{e},N_\mathrm{e}}$ and $\boldsymbol{u}^{\mathcal{S}_1}_{M_\mathrm{e},N_\mathrm{e}}$ have been obtained by choosing the discretization parameters $M_\mathrm{e} = 2^9$ and $N_\mathrm{e} = 2048$. In the last two columns we compare the memory saving of the two approaches.

Combining (24) with a known inverse-type inequality for negative Sobolev norms (see, for example, [30]), and recalling that $m = 1$ for the chosen wavelet basis, we obtain the following error estimate

$$\|\varphi(\cdot, t_n) - \varphi_L^n\|_{L^2(\Gamma)} = \mathcal{O}(\Delta_t^2) + \mathcal{O}(\Delta_x^2).$$

The results reported in the tables confirm the expected order of convergence, for fine discretizations when $c = 1$ and even for coarse discretizations when $c = 343$. As it can be seen, in spite of the high compression, the accuracy of the solutions obtained by $\mathcal{T}$ is the same of that given by $\mathcal{S}$, for which the compression is very low and, in some cases, even null.

We remark that the approach $\mathcal{W}$ has produced the same values (at least up to the second significant digit) for all the quantities reported in the two tables, excepted for the values $M = 256$ and $N = 1024$ of the last row, for which the error is slightly larger. This mismatch is due to the spatial numerical quadrature employed for the computation of the matrix entries. Indeed, for the approach $\mathcal{W}$, an "ad hoc" quadrature strategy is needed, and this will be the object of our study in a forthcoming research.

For this benchmark example, we also show the sparsity pattern of the associated matrices $\boldsymbol{V}^{*,n}$ for $* \in \{\mathcal{S}, \mathcal{W}, \mathcal{T}\}$ and we report the corresponding global memory saving.

Choosing, for example, the parameters $L = 8$, $N = 256$ and $\varepsilon = 1.0e-08$, in Figures 5 and 6 we present the structure of the compressed matrices $\boldsymbol{V}^{\mathcal{S},n}$ (first column), $\boldsymbol{V}^{\mathcal{T},n}$ (second column) and $\boldsymbol{V}^{\mathcal{W},n}$ (third, fourth and fifth column), at the time instants t_0 (top row), $t_{N/2}$ (middle row) and t_N (bottom row), for $c = 1$ and $c = 343$, respectively. For what concerns the matrices $\boldsymbol{V}^{\mathcal{W},n}$, to validate the effectiveness of estimate (29), we show the sparsity pattern at two different stages: the effectively computed entries of the matrices obtained by applying only the a priori compression strategy $\mathcal{W}1$ (fourth column); the effectively stored entries of the matrices obtained by applying further the compression strategy $\mathcal{W}2$ (fifth column). For the sake of comparison, we also show the sparsity pattern of the matrices $\boldsymbol{V}^{\mathcal{W},n}$ obtained by using a compression strategy based on an optimal truncation value evaluated "a posteriori", after computing all the entries (third column).

As we can see in Figure 5, for $c = 1$, at the first time instant the compressed matrix generated by $\mathcal{S}$ is sparse, while for the other instants they are fully populated; indeed, for $t_{N/2}$ and t_N the number nz of non-zero entries coincides with $M^2 = (2^L)^2 = 65536$. Consequently, $\text{mem}^{\mathcal{S}} \approx 9\%$ and, hence, the total memory saving in this case is low. Contrarily, the compressed wavelet matrices are all highly sparse; they have a typical *finger structure* at the first time instant and are even sparser at the subsequent instants. In particular, at $t_N = 10$, only 62 entries out of 65536 are stored. In this case $\text{mem}^{\mathcal{W}} = \text{mem}^{\mathcal{T}} \approx 91\%$.

In Figure 6, for $c = 343$, the matrix compression for the approaches $\mathcal{W}$ and $\mathcal{T}$ with respect to the approach $\mathcal{S}$ is even more evident, as expected, because of the higher value of the velocity. Indeed, in the latter case, all the entries of $\boldsymbol{V}^{\mathcal{S},n}$, for all the time instants, are computed and stored, with a consequent null memory saving ($\text{mem}^{\mathcal{S}} = 0\%$). On the contrary, the sparsity in the wavelet approach is stronger, especially when t increases. Incidentally, we point out that for $t_{N/2} = 5$ and $t_N = 10$, only a single entry of the matrices $\boldsymbol{V}^{\mathcal{W},n}$ and $\boldsymbol{V}^{\mathcal{T},n}$, out of 65536, is computed and stored. In this case $\text{mem}^{\mathcal{W}} = \text{mem}^{\mathcal{T}} \approx 99.8\%$. In both cases, the comparison between columns 3 and 4 (a posteriori and a priori estimate) shows the good performance of the a priori cutting strategy.

In Figures 7 and 8 we report the colour scale for $\overline{\mathcal{R}}_{\lambda,\lambda'}$ of the computed entries of $\boldsymbol{V}^{\mathcal{W},n}$ for all instants $t_n, n = 0 \cdots, N$, and for $c = 1$ and $c = 343$, respectively. These figures can be read as an overlap of the sparsity patterns at each time instant and give an idea of the value of $\overline{\mathcal{R}}_{\lambda,\lambda'}$ associated to each entry. The number $\text{nc} = \sum_{\lambda,\lambda'} \overline{\mathcal{R}}_{\lambda,\lambda'}$ reported in the x label represents the total number of the computed entries. It results that, before the a posteriori

cutting, the memory saving is about 69% for $c = 1$ and 99.6% for $c = 343$. The global complexity $\sum_{\lambda,\lambda'} \overline{\mathcal{R}}_{\lambda,\lambda'} \log \overline{\mathcal{R}}_{\lambda,\lambda'}$ of the algorithm is about $3.5e+07$ for $c = 1$ and $9.6e+04$ for $c = 343$, instead of $3.0e+08$ for the approach $\mathcal{S}1$.

Finally, to give an idea of the global storage requirement behaviour with respect to space and time refinements, in Figure 9 we report the memory saving $\text{mem}^{\mathcal{S}}(\%)$ and $\text{mem}^{\mathcal{W}}(\%)$ with respect to increasing values of degrees of freedom d.o.f $= MN$. In particular, we consider $M = N = 2^{\ell}$, for $\ell = 6, \ldots, 10$ and we vary the threshold parameter $\varepsilon = 1.0e-09, 1.0e-10, 1.0e-12, 1.0e-14$. As we can see, the behaviour of $\text{mem}^{\mathcal{S}}(\%)$ confirms that $\mathcal{S}$ does not benefit from the cutting strategy, independently of ε and c. On the other hand, the behaviour of $\text{mem}^{\mathcal{W}}(\%)$ clearly illustrates that the cutting strategy is effective for the approach $\mathcal{W}$ and that the memory storage depends on ε for small numbers of d.o.f, but it is asymptotically independent of ε and c.

In Figure 10, we consider a fixed threshold $\varepsilon = 1.0e-12$ and $M = N = 2^{\ell}$, for $\ell = 2, \ldots, 10$. The figure shows that the global memory storage for $\mathcal{S}$ behaves like that of $\mathcal{S}1$, i.e. M^2N for both choices of the velocities $c = 1$ (left plot) and $c = 343$ (right plot). On the contrary, we observe a linear growth $\mathcal{O}(MN)$ of the storage requirement for the approach $\mathcal{W}$ for $c = 1$ (right plot) starting from $M = N = 2^7$, and less than linear for the higher velocity $c = 343$ (right plot) starting from $M = N = 2^5$.

Example 2.

We consider Problem (1) with $c = 1$,

$$g(\mathbf{x}, t) = t^4 e^{-2t}$$

and $T = 10$. Since the resulting solution does not depend on the variable $\mathbf{x}$, being g independent of $\mathbf{x}$, it is sufficient to consider the coarse spatial discretization of level $L = 3$. We discretize the time interval $[0, T]$ in $N = 256$ subintervals and we choose the large threshold parameter $\varepsilon = 1.0e-04$.

In the left plots of Figures 11 and 12 we show the behaviour in time of the approximate solutions $\boldsymbol{\varphi}^{\mathcal{W}}_{M,N}$ and $\boldsymbol{u}^{\mathcal{W}}_{M,N}$ (green dashed line), $\boldsymbol{\varphi}^{\mathcal{S}}_{M,N}$ and $\boldsymbol{u}^{\mathcal{S}}_{M,N}$ (red dotted line), $\boldsymbol{\varphi}^{\mathcal{T}}_{M,N}$ and $\boldsymbol{u}^{\mathcal{T}}_{M,N}$ (blue dotted-dashed line). In the right plots, we report the corresponding absolute errors. The reference solutions $\boldsymbol{\varphi}^{\mathcal{S}_1}_{M_e,N_e}$ and $\boldsymbol{u}^{\mathcal{S}_1}_{M_e,N_e}$ are obtained by choosing the discretization parameters $M_e = M = 2^3$ and $N_e = N$. We remark that the memory savings are $\text{mem}^{\mathcal{T}} = \text{mem}^{\mathcal{W}} \approx 77\%$ and $\text{mem}^{\mathcal{S}} \approx 6\%$.

As we can see, the worst solution is that associated to the approach $\mathcal{S}$, regardless the cutoff strategy has maintained almost all the entries with a resulting low memory saving. It is worth noting that, for this approach, the threshold ε reveals to be too coarse. Contrarily, the wavelet approach allows to obtain an accurate solution with a high memory saving. We point out that the discrepancy in accuracy between the approaches $\mathcal{W}$ and $\mathcal{T}$ is essentially due to the numerical computation of the spatial integrals which appear the matrix entries definition, as already remarked in the previous example.

Example 3.

We consider here Problem (1) with $c = 343$,

$$g(\mathbf{x}, t) = t^4 e^{-t} \cos(x_1^2 + 4x_2^2)$$

and $T = 10$. The discretization parameters are $L = 8$ and $N = 256$, and the threshold is $\varepsilon = 1.0e-12$. The reference solutions $\boldsymbol{\varphi}^{\mathcal{S}_1}_{M_\mathrm{e},N_\mathrm{e}}$ and $\boldsymbol{u}^{\mathcal{S}_1}_{M_\mathrm{e},N_\mathrm{e}}$ are obtained by choosing the discretization parameters $M_\mathrm{e} = M = 2^8$ and $N_\mathrm{e} = N$. With these choices, it results that almost the whole storage of the matrices $\boldsymbol{V}^{\mathcal{S},n}$ associated to the standard approach is necessary, so that the memory saving is $\mathrm{mem}^{\mathcal{S}} \approx 0.01\%$; this justifies that the errors associated to the density and potential functions have the machine precision order. On the other hand, the total memory saving for $\mathcal{W}$ and $\mathcal{T}$ is $\mathrm{mem}^{\mathcal{W}} = \mathrm{mem}^{\mathcal{T}} \approx 98\%$. The maximum value of the corresponding errors is approximately $1.0e-03$ for $\boldsymbol{\varphi}^{\mathcal{W}}_{M,N}$, $1.0e-06$ for $\boldsymbol{\varphi}^{\mathcal{T}}_{M,N}$, $1.0e-07$ for $\boldsymbol{u}^{\mathcal{W}}_{M,N}$ and $1.0e-10$ for $\boldsymbol{u}^{\mathcal{T}}_{M,N}$.

Example 4.

As last example, we present a case for which the approximate solution requires a very fine space and time discretization, that turn out to be prohibitive for the approaches $\mathcal{S}$ and $\mathcal{T}$ in terms of memory space. Indeed our PC prevented the execution of $\mathcal{S}$ and $\mathcal{T}$ because of an out of memory. Therefore, the compression strategy of the new approach revealed to be crucial for this simulation.

For this example, we consider a wave that propagates with velocity $c = 343$, generated by the highly oscillating in time Dirichlet datum

$$g(\mathbf{x}, t) = t^2 e^{-t} \sin(512t) \cos(x_1^2 + 32x_2^2).$$

We solve BIE (4) in the time interval $[0, 1]$ by using $L = 9$ and $N = 4096$. These fine discretization parameters are necessary to reproduce accurately the oscillating behaviour of the density function and of the potential. The threshold is $\varepsilon = 1.0e-10$. In Figure 15 we show the solutions $\boldsymbol{\varphi}^{\mathcal{W}}_{M,N}$ and $\boldsymbol{u}^{\mathcal{W}}_{M,N}$, computed with a memory saving $\mathrm{mem}^{\mathcal{W}} \approx 99.6\%$. In particular, in the top-left plot we show $\boldsymbol{\varphi}^{\mathcal{W}}_{M,N}(\mathbf{x}, t)$ at $\mathbf{x} = (-1, 0)$ for $t \in [0, T]$, in the top-right plot $\boldsymbol{\varphi}^{\mathcal{W}}_{M,N}(\mathbf{x}, T)$ for $\mathbf{x} \in \Gamma$ and in the bottom plot $\boldsymbol{u}^{\mathcal{W}}_{M,N}(\mathbf{x}, t)$ at the external point $\mathbf{x} = (2, 0)$ for $t \in [0, T]$.

4. Conclusions

We have considered a boundary integral formulation of an exterior 2D wave propagation problem. For the resolution of the corresponding BIE, we have used a wavelet Galerkin method in space coupled with a Lubich convolution quadrature method in time. The coupling of the two schemes is new and, although wavelet approximations have been successfully applied to BEMs, there is not much work on their use for the resolution of time-dependent

BIEs. We have devised an approach, denoted by $\mathcal{W}$, that combines the good properties of the wavelet approximation in space and those of Lubich's convolution quadrature method in time. Based on an a priori estimate of the decaying behaviour of the matrix entries and on a downsampled FFT, this approach offers two main advantages: a high matrix compression and a considerable computational cost reduction for the time discretization.

We have compared this approach with: the approach $\mathcal{S}$ that uses piece-wise linear spatial approximating functions and Lubich's quadrature method in time, with an a posteriori cutting; the approach $\mathcal{T}$ that uses wavelet spatial approximating functions, by means of a wavelet transform, and Lubich's quadrature in time, with an a posteriori cutting.

As shown in Section 3, the approach $\mathcal{S}$ does not permit a matrix compression strategy (independently of the wave propagation velocity) and, consequently, it does not allow neither a computational cost reduction nor a memory saving.

On the contrary, thanks to wavelet properties, the approach $\mathcal{T}$ permits a significant matrix compression (depending on the wave propagation velocity), but this allows only an a posteriori cutting and a computational cost reduction in the resolution of the final linear system. Indeed, this approach can not be taken into account for memory saving since the compression is possible only after that the matrices associated to the approach $\mathcal{S}$ have been entirely computed and stored.

Therefore, the new approach $\mathcal{W}$, which allows simultaneously computational cost reduction and memory saving, turns out to be crucial when applied to large scale problems, for which the computation and the storage of all the matrices become prohibitive.

Acknowledgements

We aknowledge the anonymous reviewer for his helpful and constructive comments that greatly contributed to improving the final version of the paper.

References

[1] A. Bamberger, T. Ha Duong, Formulation variationelle espace-temps pour le calcul par potentiel retardé de la diffraction d'une onde acoustique, Math. Meth. in the Appl. Sci 8 (1986) 405–435.

[2] C. Lubich, On the multistep time discretization of linear initial-boundary value problems and their boundary integral equations, Numer. Math. 67 (3) (1994) 365–389.

[3] A. Aimi, M. Diligenti, C. Guardasoni, I. Mazzieri, S. Panizzi, An energy approach to space-time Galerkin BEM for wave propagation problems, Internat. J. Numer. Methods Engrg. 80 (2009) 1196–1240.

[4] S. Falletta, G. Monegato, L. Scuderi, A space-time BIE methods for nonhomogeneous exterior wave equation problems. The Dirichlet case, IMA J. Numer. Anal. 32 (1) (2012) 202–226.

[5] S. Falletta, G. Monegato, L. Scuderi, A space-time BIE method for wave equation problems: the (two-dimensional) Neumann case, IMA J. Numer. Anal. 34 (1) (2014) 390–434.

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

[7] S. Falletta, G. Monegato, An exact non reflecting boundary condition for 2D time-dependent wave equation problems, Wave Motion 51 (1) (2014) 168–192.

[8] S. Falletta, G. Monegato, Exact non-reflecting boundary condition for 3D time-dependent multiple scattering-multiple source problems, Wave Motion 58 (2015) 281–302.

[9] S. Falletta, BEM coupling with the FEM fictitious domain approach for the solution of the exterior Poisson problem and of wave scattering by rotating rigid bodies, IMA J. Numer. Anal. 38 (2) (2018) 779–809.

[10] S. Falletta, G. Monegato, L. Scuderi, On the discretization and application of two space-time boundary integral equations for 3D wave propagation problems in unbounded domains, App. Num. Math. 124 (2018) 22–43.

[11] L. Banjai, M. Kachanovska, Fast convolution quadrature for the wave equation in three dimensions, J. Comput. Phys. 279 (2014) 103–126.

[12] L. Greengard, V. Rokhlin, A new version of the fast multipole method for the Laplace equation in three dimensions, Acta Numerica 6 (1997) 229–269.

[13] W. Hackbusch, The Panel Clustering Technique for the Boundary Element Method, Vol. 9/1, Springer, 1987.

[14] S. Falletta, S. Sauter, The panel-clustering method for the wave equation in two spatial dimensions, J. Comput. Physics 305 (2016) 217–243.

[15] W. Hackbusch, Z. Nowak, On the fast matrix multiplication in the boundary element method by panel clustering, Numer. Math. 54 (4) (1989) 463–491.

[16] L. Banjai, V. Grühne, Efficient long-time computations of time-domain boundary integrals for 2D and dissipative wave equation, J. Comp. Appl. Math. 235 (2011) 4207–4220.

[17] L. Banjai, M. Lopez-Fernandez, A. Schädle, Fast and oblivious algorithms for dissipative and two-dimensional wave equations, SIAM J. Num. Anal. 55 (2017) 621–639.

[18] W. Dahmen, S. Prössdorf, R. Schneider, Wavelet approximation methods for pseudodifferential equations. II. Matrix compression and fast solution 1 (1993) 259–335.

[19] H. Harbrecht, R. Schneider, Wavelet Galerkin schemes for 2D-BEM, Vol. 121 of Oper. Theory Adv. Appl., Birkhäuser, Basel, 2001, pp. 221–260.

[20] G. Beylkin, R. Coifman, V. Rokhlin, Fast wavelet transforms and numerical algorithms. I, Comm. Pure Appl. Math. 44 (2).

[21] G. C. Hsiao, A. Rathsfeld, Wavelet collocation methods for a first kind boundary integral equation in acoustic scattering, Adv. Comput. Math. 17 (4) (2002) 281–308.

[22] C. Lage, C. Schwab, Wavelet Galerkin algorithms for boundary integral equations, SIAM J. Sci. Comput. 20 (6) (1999) 2195–2222.

[23] K. Koro, K. Abe, Application of Haar wavelets to time-domain BEM for the transient scalar wave equation, IOP Conference Series: Materials Science and Engineering 10 (1) (2010) 1–10.

[24] C. Lubich, Convolution quadrature and discretized operational calculus. I, Numer. Math. 52 (1988) 129–145.

[25] C. Lubich, Convolution quadrature and discretized operational calculus. II, Numer. Math. 52 (1988) 413–425.

[26] I. Daubechies, Ten Lectures on Wavelets, SIAM, 2004.

[27] A. Cohen, I. Daubechies, J. Feauveau, Bi-orthogonal bases of compactly supported wavelets, Comm. Pure Appl. Math. 45 (1992) 485–560.

[28] W. Dahmen, H. Harbrecht, R. Schneider, Compression techniques for boundary integral equations – Asymptotically optimal complexity estimates, SIAM J. Numer. Anal. 43 (6) (2006) 2251–2271.

[29] P. G. Ciarlet, The Finite Element Method for Elliptic Problems, North-Holland, Amsterdam, 1978.

[30] I. Graham, W. Hackbusch, S. Sauter, Finite elements on degenerate meshes: inverse-type inequalities and applications, IMA J. Numer. Anal. 25 (2005) 379–407.

Figure 1: The functions ϕ and ψ associated to the biorthogonal *bior2.2* wavelets.

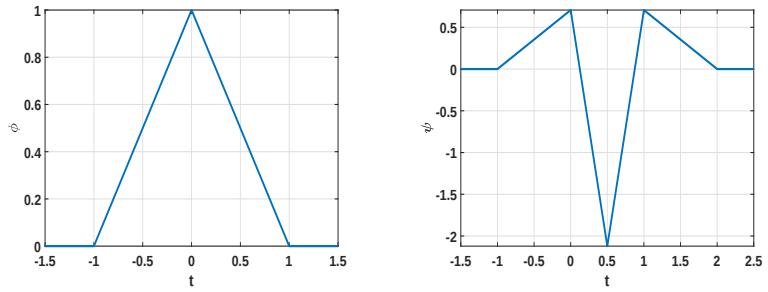

Figure 2: scaling function $\psi_{-1,0}$ and wavelet basis functions $\psi_{0,0}$, $\psi_{1,0}$, $\psi_{1,1}$, $\psi_{2,0}$ and $\psi_{3,3}$ for the choice $L = 5$.

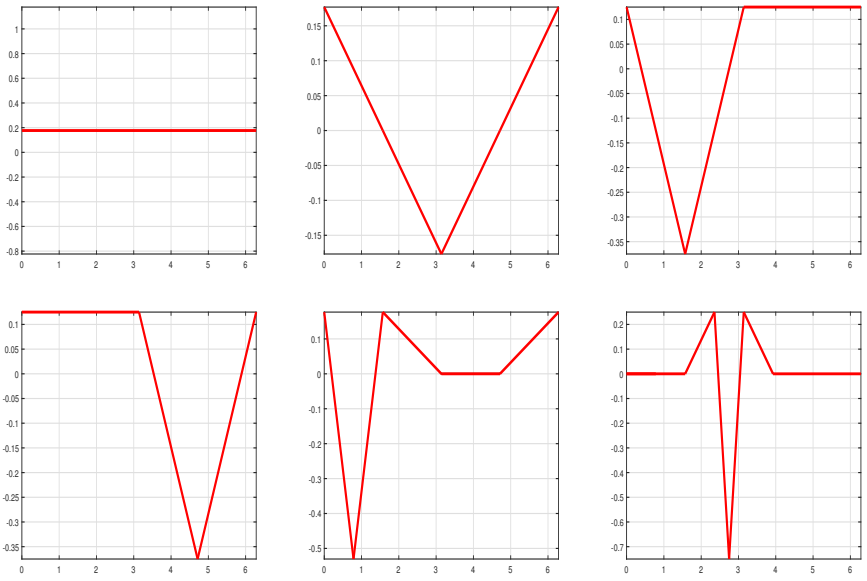

Figure 3: coefficients $\tilde{c}^{\mathcal{S}}_{i,j}(n)$ and corresponding matrix entries $\boldsymbol{V}^{\mathcal{S},n}_{i,j}$ with $(i,j) = (9,141)$ (top). Coefficients $\tilde{c}^{\mathcal{W}}_{\lambda,\lambda'}(n)$ and corresponding matrix entries $\boldsymbol{V}^{\mathcal{W},n}_{\lambda,\lambda'}$ with $\lambda = (3,1), \lambda' = (7,13)$ (middle) and with $\lambda = (7,0), \lambda' = (7,123)$ (bottom). $c = 1$.

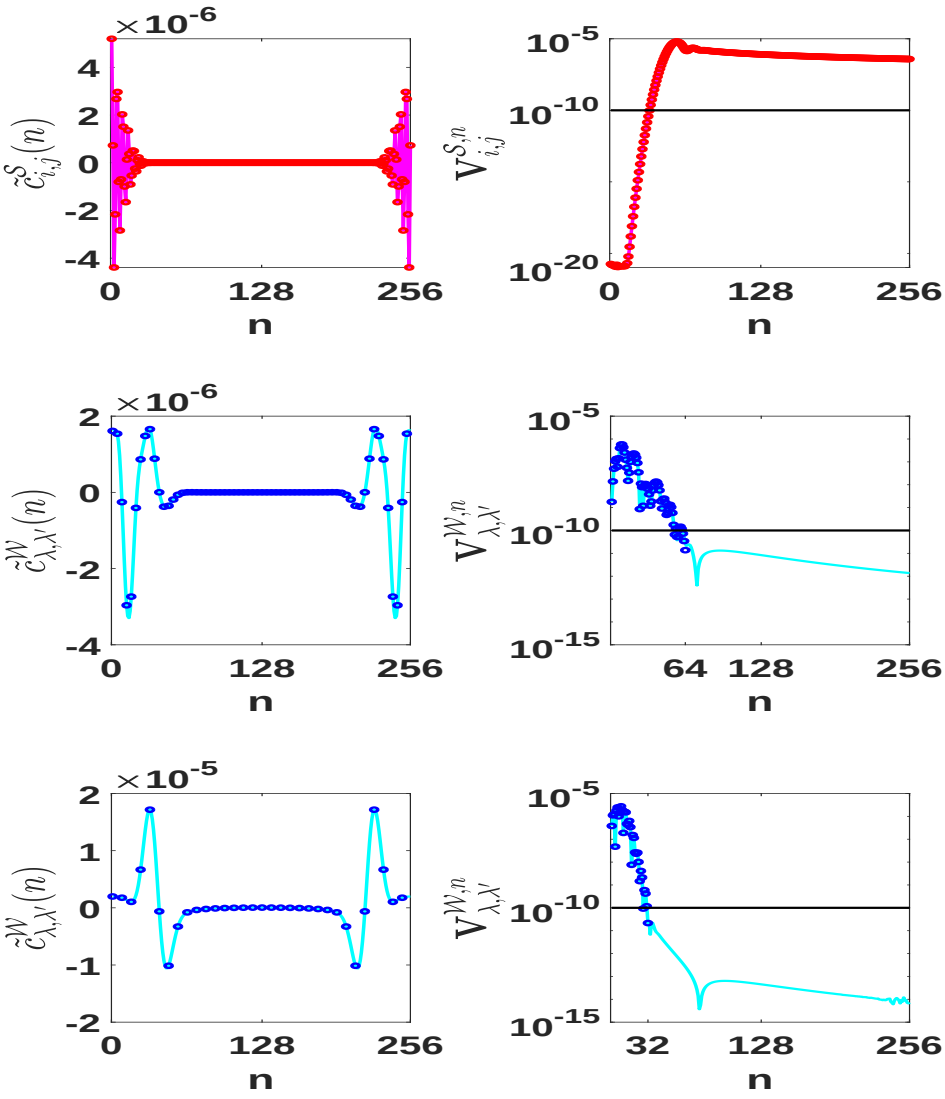

Figure 4: coefficients $\tilde{c}^{\mathcal{S}}_{i,j}(n)$ and corresponding matrix entries $\boldsymbol{V}^{\mathcal{S},n}_{i,j}$ with $(i,j) = (9,141)$ (top). Coefficients $\tilde{c}^{\mathcal{W}}_{\lambda,\lambda'}(n)$ and corresponding matrix entries $\boldsymbol{V}^{\mathcal{W},n}_{\lambda,\lambda'}$ with $\lambda = (3,1), \lambda' = (7,13)$ (middle) and with $\lambda = (7,0), \lambda' = (7,123)$ (bottom). $c = 343$.

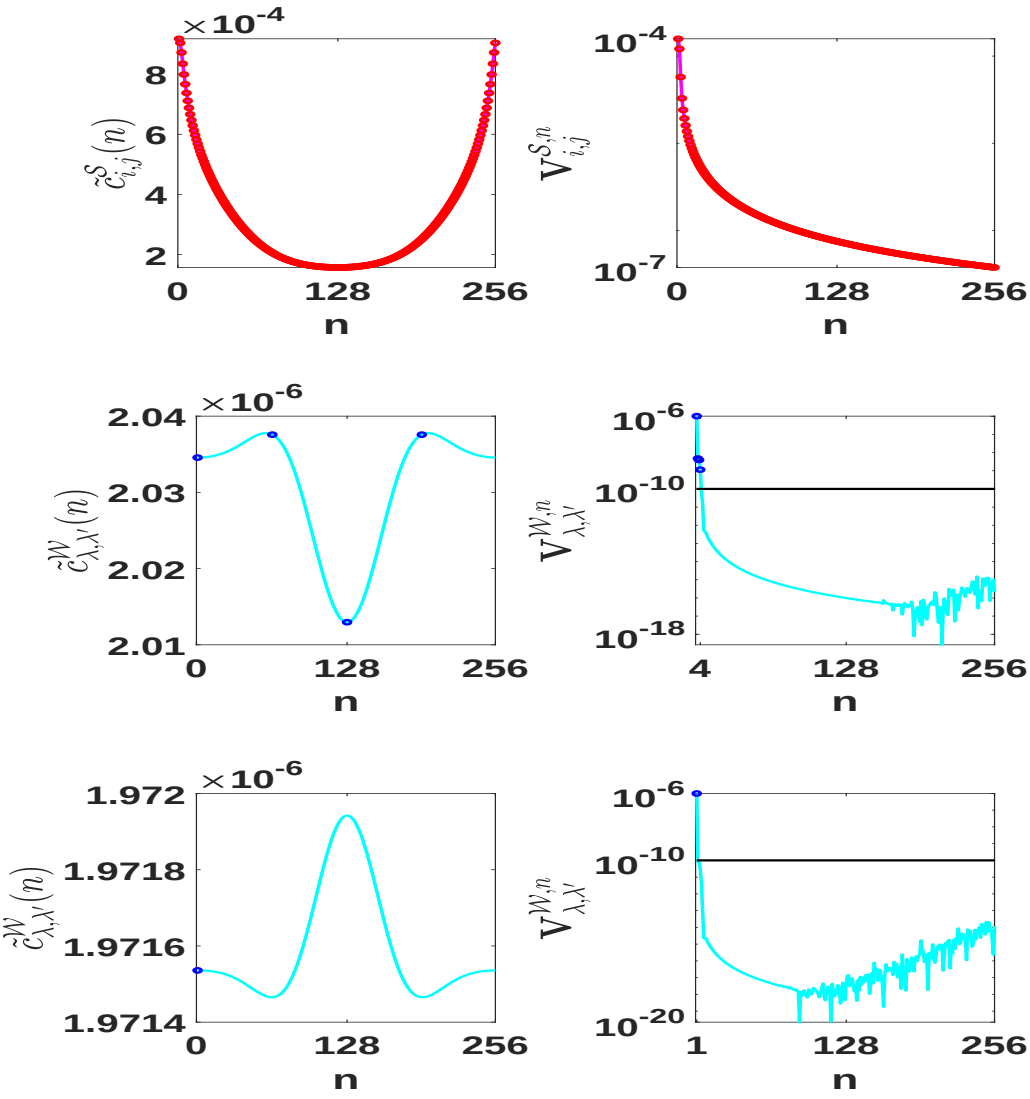

Table 1: Example 1. Errors, EOC, mem*(%), $* = \mathcal{S}, \mathcal{T}$. $T = 10$, $c = 1$, $\varepsilon = 1.0e-10$, $\mathbf{x} = (2, 0)$.

M	N	$\mathrm{E}^{\mathcal{S}}_{L^2,\varphi}$	EOC	$\mathrm{E}^{\mathcal{T}}_{L^2,\varphi}$	EOC	$\mathrm{E}^{\mathcal{S}}_{\infty,u}(\mathbf{x})$	EOC	$\mathrm{E}^{\mathcal{T}}_{\infty,u}(\mathbf{x})$	EOC	mem$^{\mathcal{S}}$	mem$^{\mathcal{T}}$
4	16	$6.31e-01$		$6.31e-01$		$1.76e-02$		$1.76e-02$		0%	63%
			1.2		1.2		0.9		0.9		
8	32	$2.69e-01$		$2.69e-01$		$9.16e-03$		$9.16e-03$		0%	28%
			1.5		1.5		1.4		1.4		
16	64	$9.38e-02$		$9.38e-02$		$3.55e-03$		$3.55e-03$		0%	13%
			0.4		0.4		1.8		1.8		
32	128	$6.92e-02$		$6.92e-02$		$1.00e-03$		$1.00e-03$		3%	7%
			0.9		0.9		2.0		2.0		
64	256	$3.50e-02$		$3.50e-02$		$2.55e-04$		$2.55e-04$		6%	34%
			1.5		1.5		1.9		1.9		
128	512	$1.20e-02$		$1.20e-02$		$6.89e-05$		$6.89e-05$		9%	70%
			2.0		2.0		2.3		2.3		
256	1024	$2.99e-03$		$3.08e-03$		$1.38e-05$		$1.38e-05$		10%	87%

Table 2: Example 1. Errors, EOC, mem*(%), $* = \mathcal{S}, \mathcal{T}$. $T = 10$, $c = 343$, $\varepsilon = 1.0e-10$, $\mathbf{x} = (2, 0)$.

M	N	$\mathrm{E}^{\mathcal{S}}_{L^2,\varphi}$	EOC	$\mathrm{E}^{\mathcal{T}}_{L^2,\varphi}$	EOC	$\mathrm{E}^{\mathcal{S}}_{\infty,u}(\mathbf{x})$	EOC	$\mathrm{E}^{\mathcal{T}}_{\infty,u}(\mathbf{x})$	EOC	mem$^{\mathcal{S}}$	mem$^{\mathcal{T}}$
4	16	$3.65e-01$		$3.65e-01$		$6.83e-05$		$6.83e-05$		0%	63%
			0.5		0.5		-0.6		-0.6		
8	32	$2.57e-01$		$2.57e-01$		$1.07e-04$		$1.07e-04$		0%	39%
			2.0		2.0		4.2		4.2		
16	64	$5.57e-02$		$5.57e-02$		$5.97e-06$		$5.97e-06$		0%	70%
			2.0		2.0		1.9		1.9		
32	128	$1.36e-02$		$1.36e-02$		$1.60e-06$		$1.60e-06$		0%	90%
			2.0		2.0		2.0		2.0		
64	256	$3.32e-03$		$3.32e-03$		$4.03e-07$		$4.03e-07$		0%	97%
			2.1		2.1		2.0		2.0		
128	512	$7.91e-04$		$7.91e-04$		$9.69e-08$		$9.69e-08$		0%	99%
			2.3		2.3		2.3		2.3		
256	1024	$1.58e-04$		$1.61e-04$		$2.00e-08$		$2.00e-08$		0%	99.7%

Figure 5: Example 1. Sparsity pattern of $\boldsymbol{V}^{\mathcal{S},n}$ (column 1), $\boldsymbol{V}^{\mathcal{T},n}$ (column 2) and $\boldsymbol{V}^{\mathcal{W},n}$ (columns 3,4,5) at t_n, for $n = 0, 128, 256$. $T = 10$, $c = 1$, $\varepsilon = 1.0e-08$, $M = N = 2^8$.

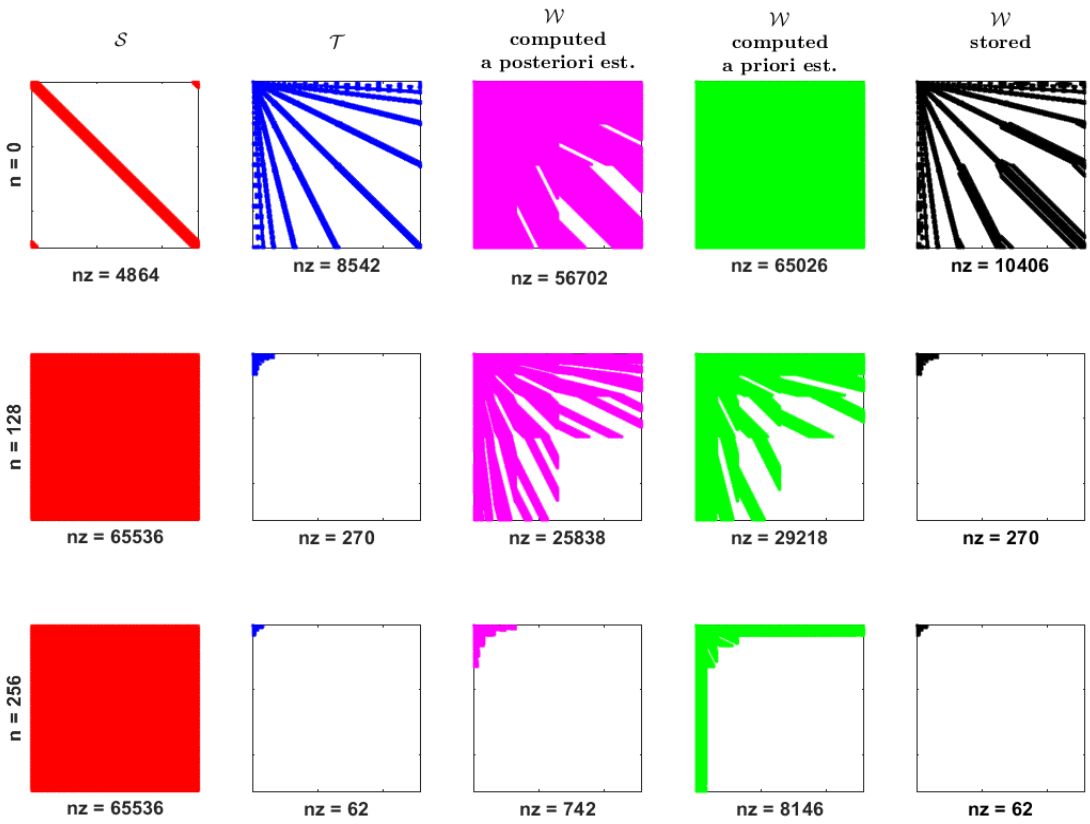

Figure 6: Example 1. Sparsity pattern of $\boldsymbol{V}^{\mathcal{S},n}$ (column 1), $\boldsymbol{V}^{\mathcal{T},n}$ (column 2) and $\boldsymbol{V}^{\mathcal{W},n}$ (columns 3,4,5) at t_n, for $n = 0, 128, 256$. $T = 10$, $c = 343$, $\varepsilon = 1.0e-08$, $M = N = 2^8$.

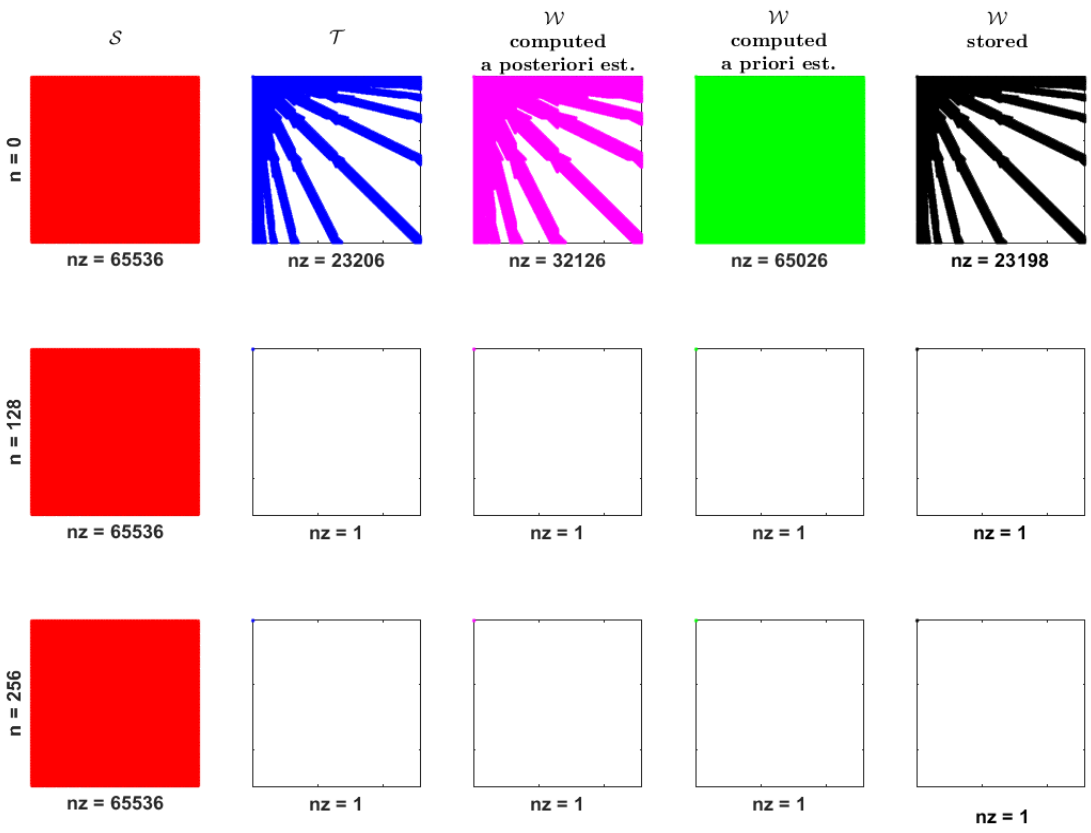

Figure 7: Example 1. Colour scale for $\overline{\mathcal{R}}_{\lambda,\lambda'}$ of the all computed entries $\boldsymbol{V}^{\mathcal{W},n}_{\lambda,\lambda'}$ for $t_n, n = 0 \cdots, N$. $T = 10$, $c = 1$, $\varepsilon = 1.0e-08$, $M = N = 2^8$.

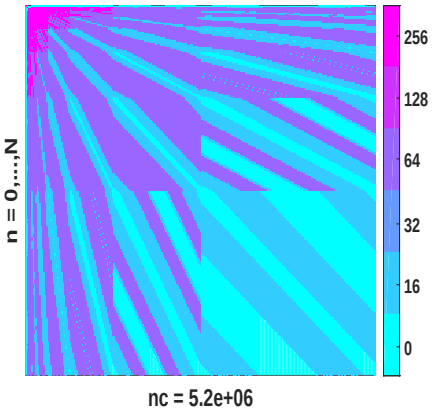

Figure 8: Example 1. Colour scale for $\overline{\mathcal{R}}_{\lambda,\lambda'}$ of the all computed entries $\boldsymbol{V}_{\lambda,\lambda'}^{\mathcal{W},n}$ for $t_n, n = 0 \cdots, N$. $T = 10$, $c = 343$, $\varepsilon = 1.0e-08$, $M = N = 2^8$ (left plot), with a zoom on the first 16×16 block (right plot).

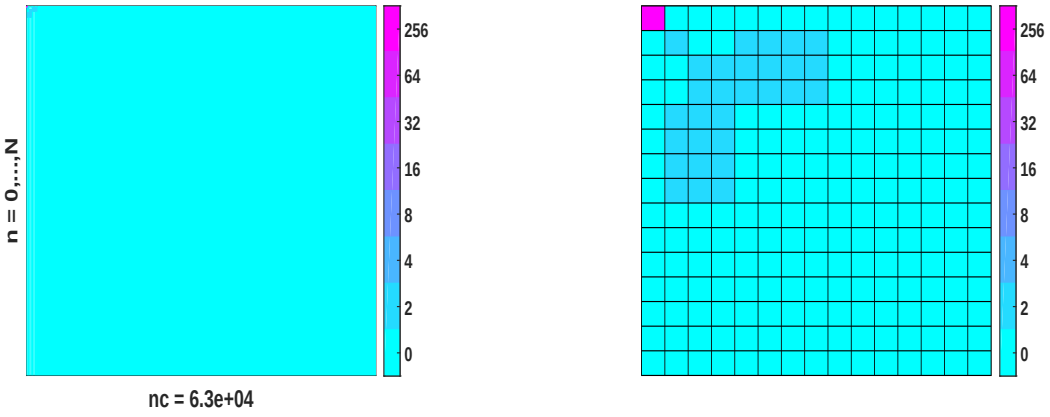

Figure 9: Example 1. Memory saving mem* for $* = \mathcal{S}, \mathcal{W}$, with respect to increasing values of d.o.f and different thresholds ε. $T = 10$, $c = 1$ (left plot) and $c = 343$ (right plot).

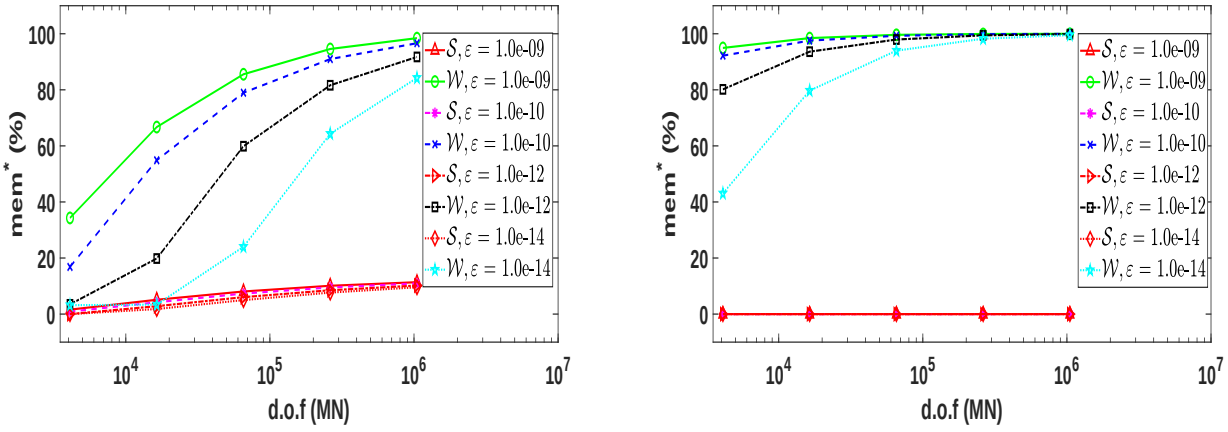

Figure 10: Example 1. Memory storage for $\mathcal{S}$ and $\mathcal{W}$, with respect to increasing values of d.o.f. $T = 10$, $c = 1$ (left plot) and $c = 343$ (right plot), $\varepsilon = 1.0e - 12$.

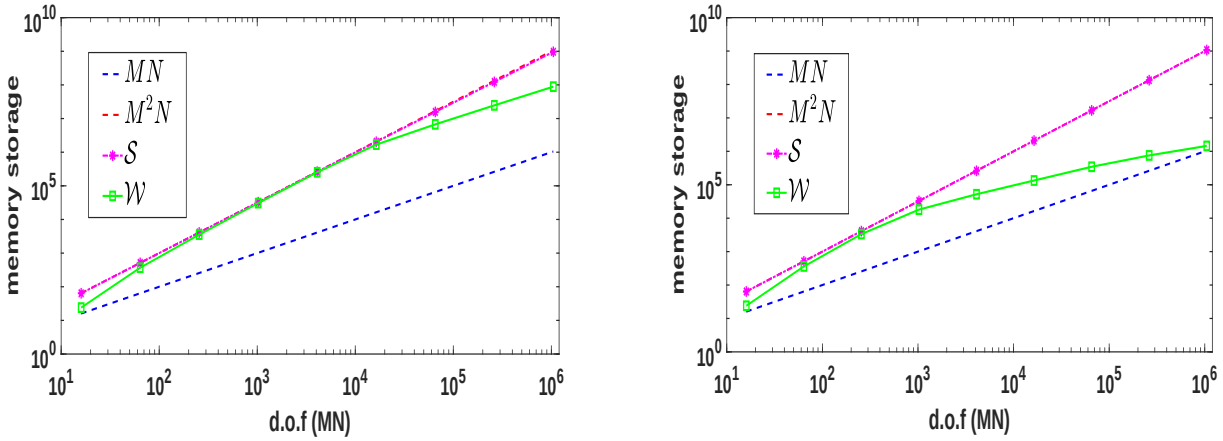

Figure 11: Example 2. Approximate functions $\boldsymbol{\varphi}^*_{M,N}(\mathbf{x}, t)$, $* = \mathcal{S}, \mathcal{T}, \mathcal{W}$, $\mathbf{x} = (-1, 0)$ (left plot) and corresponding absolute errors (right plot). $T = 10$, $c = 1$, $\varepsilon = 1.0e - 04$, $M = 2^3$, $N = 256$.

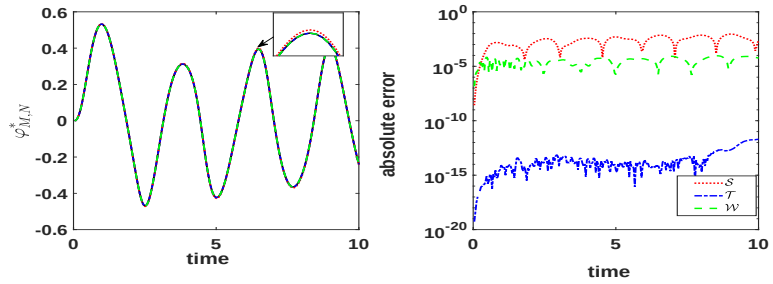

Figure 12: Example 2. Approximate functions $\mathbf{u}^*_{M,N}(\mathbf{x}, t)$, $* = \mathcal{S}, \mathcal{T}, \mathcal{W}$, $\mathbf{x} = (2, 0)$ (left plot) and corresponding absolute errors (right plot). $T = 10$, $c = 1$, $\varepsilon = 1.0e - 04$, $M = 2^3$, $N = 256$.

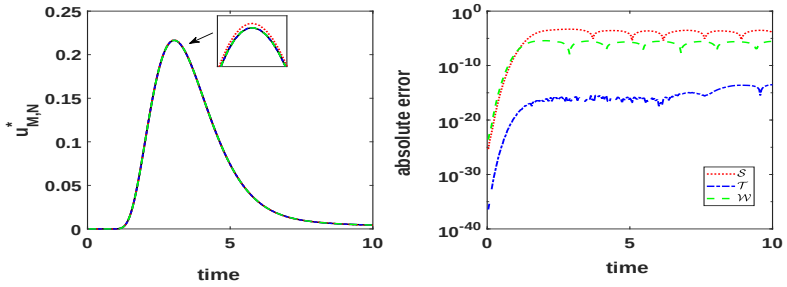

Figure 13: Example 3. Approximate functions $\boldsymbol{\varphi}^*_{M,N}(\mathbf{x},t)$, $*=\mathcal{S},\mathcal{T},\mathcal{W}$, $\mathbf{x}=(-1,0)$ (left plot) and corresponding absolute errors (right plot). $T=10$, $c=343$, $\varepsilon=1.0e-12$, $M=N=2^8$.

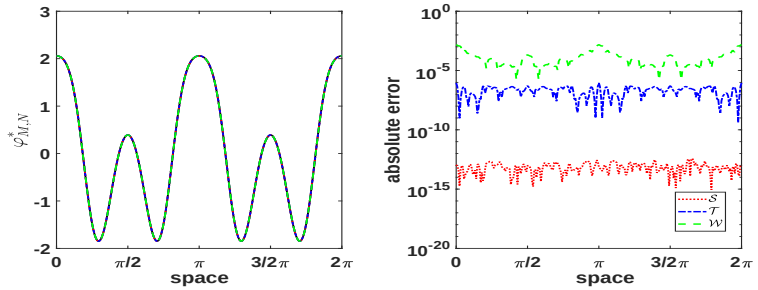

Figure 14: Example 3. Approximate functions $\boldsymbol{u}^*_{M,N}(\mathbf{x},t)$, $*=\mathcal{S},\mathcal{T},\mathcal{W}$, $\mathbf{x}=(2,0)$ (left plot) and corresponding absolute errors (right plot). $T=10$, $c=343$, $\varepsilon=1.0e-12$, $M=N=2^8$.

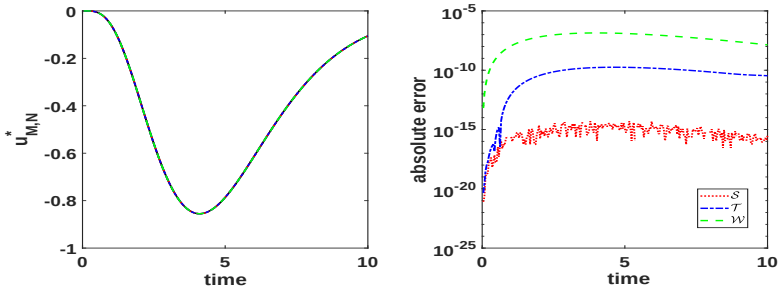

Figure 15: Example 4. Approximate function $\boldsymbol{\varphi}^{\mathcal{W}}_{M,N}(\mathbf{x},t)$, in time for $\mathbf{x}=(-1,0)$ (top-left) and in space for $t=T$ (top-right), and the potential solution $\boldsymbol{u}^{\mathcal{W}}_{M,N}(\mathbf{x},t)$, for $\mathbf{x}=(2,0)$ (bottom). $T=1$, $c=343$, $\varepsilon=1.0e-10$, $M=2^9$, $N=4096$.