Decimation in Time and Space of Finite-Difference Time-Domain Schemes: Standard Isotropic Lossless Model

Federico Fontana, *Senior Member, IEEE,* Enrico Bozzo,
and Marco Novello, *Student Member, IEEE*

Abstract

Finite-Difference Time-Domain schemes permit changes in the grid density on selected regions of the wave propagation domain, which can reduce the computational load of the simulations. One possible alternative to varying the spatial density is to change the simulation temporal rate. This idea looks attractive when the wave signals exhibit pronounced bandwidth fluctuations across time. This is particularly true in sound synthesis, where a physically-based acoustic resonator can be conveniently modeled using such schemes. To overcome the computational constraints that must be met by real-time distributed resonator models, this paper deals with the decimation in time and space of isotropic lossless finite-difference time-domain schemes holding conventional Nyquist-Shannon limits on the bandwidth of the wave signals. Formulae for the reconstruction of these signals at runtime over the interpolated grid are provided for both the 1D and 2D orthogonal case, depending on the ideal boundary conditions (either Neumann or Dirichlet) holding at each side of the grid in connection with the domain side lengths (either even or odd). Together, the boundaries and size determine the type of Discrete Cosine Transform used in the corresponding interpolation formula. Numerical artifacts arising as a consequence of decimating in space in 2D are discussed in terms of dispersion error and aliasing. Considerations concerning the temporal reconstruction of components lying at the decimated Nyquist frequency are addressed in the conclusion.

Index Terms

F. Fontana and E. Bozzo are with the Dipartimento di Matematica e Informatica, University of Udine, Italy, e-mail: {federico.fontana,enrico.bozzo}@uniud.it.

M. Novello is with the Dipartimento di Ingegneria Elettrica, Gestionale e Meccanica, University of Udine, Italy, e-mail: novello.marco@spes.uniud.it.

Multidimensional systems, interpolation, Discrete Cosine Transform, numerical dispersion, second-difference matrix.

EDICS Category: DSP-APPL

Decimation in Time and Space of

Finite-Difference Time-Domain Schemes:

Standard Isotropic Lossless Model

I. INTRODUCTION

Finite-difference time-domain (FDTD) methods provide numerical solutions to several important wave propagation problems. Such methods are useful in a number of application areas, including electromagnetic transmission [1] and sound analysis and synthesis [2], [3]. The transmission and reflection properties of an electromagnetic or air pressure field, as well as an acoustically resonating solid body can be modeled by means of a differential problem in continuous space and time. Once geometric and physical parameters such as dimensionality, spatial extension, medium and boundary characteristics are specified, then FDTD methods use stable numerical schemes whose integration accuracy can be scaled to the available machine computation speed and memory resources.

One critical issue affecting multidimensional numerical simulation is the time needed for integrating the solution along a sufficient temporal window. In spite of the recent progress in this area, (in particular, modular FDTD scheme implementations running on accelerated graphics hardware [4], [5]), fast and accurate solvers are yet to appear. In sound synthesis, the real-time constraint to date seems to be out of reach for all but FDTD grids containing a few thousand nodes; such grids typically correspond to small and/or low resolution domains of integration. This is especially true when 3D propagation must be dealt with, as solid shape descriptions rapidly lead to explosion of the number of nodes. Another option reduces the sampling rate: this allows real-time schemes of the order of a hundred thousand nodes to be defined which are capable of computing a fraction of the auditory range in medium and large room models. For instance, simulating a bandwidth of about 1-2 kHz is especially interesting for the observation of the standing waves appearing in large scale rooms. [**?**].

On the other hand, FDTD methods are appealing for their applicability in object-based interactive sound synthesis and real-time processing. They can describe the elementary parts forming a resonating body, a musical instrument, a reverberant room, or even all of them together in the same scenario [6]. Hence, they are still the best candidates for modeling the nonlinear closed-loop interaction occurring when mechanical or fluid-dynamical energy is transferred from an excitation object to the acoustic resonator

[7], [8]. The accurate reproduction of this interaction is the basis of almost any realistic physics-based sound generation [9]. Alternative approaches to interactive sound synthesis including modal or spectral methods [10], [11] conversely seem, (at least in the short term and in spite of their efficiency) unable to incorporate the nonlinear excitations. These domain transformations become complicated especially when a physical scenario contains multiple interacting objects [?].

Multidimensional force and pressure waves propagating across a lossless and isotropic media modeled by finite difference schemes (more precisely, by a subset of computationally equivalent structures called digital waveguide meshes) have been shown to fill the spectrum until the Nyquist limit [12], [13]. Unfortunately, preserving the precision of the simulation across the whole spectrum (until the Nyquist frequency) requires complex communication among the nodes [14] as well as sophisticated impedance and boundary models [15], [16]. The progressive loss of accuracy with frequency in an FDTD scheme occurs due to

   i) numerical dispersion,
  ii) the approximated geometry of the interface between propagation media having different impedance,
 iii) the finite boundary reflection.

Ultimately, all such artifacts are a consequence of the spatial discretization of the domain where the differential problem was originally defined. In absence of sophisticated models, then, their mitigation requires the use of dense grids running at high sampling rates, resulting in schemes whose Nyquist limit can be many times larger than the auditory band, with consequent waste of computational resources.

In addition to accuracy, physically real sounds (apart from noisy or impulsive events such as sustained air flows or clicks) possess high frequency components only during the attack transient. These usually cover a minor portion of the sonic event; these components rapidly fade during the steady state evolution of the resonating system. Consequently, an FDTD scheme for physics-based sound synthesis requires only a fraction of the otherwise available bandwidth except during and immediately after the excitation.

Literature concerned with the local reduction of computation and memory in FDTD schemes, or *subgridding*, can be found in electromagnetic transmission modeling. The proposed methods put an emphasis on modeling selected regions of a propagation domain using node densities that are commensurate to the local medium impedance and wave speed [17], [18]: once known issues of instability and potential sources of artifacts have been solved or mitigated, subgrids do offer substantial computational advantages. However, to date these methods have found convenient application inside wave propagation models that are essentially stationary [17]. This characteristic is typical in radio transmission, where the information is carried by high-frequency periodic signals.

As mentioned previously, instead of being stationary, audio signals are characterized by the presence of abrupt discontinuities and by frequency components in the audible range that sway a lot across time. As opposed to focusing on changes of the node density across space, as subgridding does, we propose varying the sampling rate of the simulation at runtime. This paper investigates the effects on the FDTD grid following a switch of the sampling rate to a different value.

A family of FDTD schemes whose sampling frequency may be varied at runtime, (depending on the bandwidth of the signals propagating along the grid) would in principle permit dynamic resolution adaptation. This idea has fostered some early research from the authors leading to simulations of time-decimated schemes [25]. For the computation of musical instrument sounds, in which the exciter is often statistically active for less than 0.1% of the total processing time [10], the proposed method could be employed for percussion instrument models [**?**], [6]. The corresponding FDTD scheme would run at high-definition audio rate only during and immediately after every hit from the stick, i.e. across a minor part of the total simulation time.

A. Definition of the problem and content of the paper

Filtering out the wave components above half Nyquist frequency that have already been solved by an FDTD scheme implies that about 50% of the memory locations and 75% of the computations are wasted. This inefficiency occurs when one dimension of propagation is modeled: in the 1D case, in fact, a *decimated* scheme using a grid having half the node density and running at half the sampling rate would in principle produce signals containing the same information.

By continuing this reasoning, if two dimensions are modeled then about 25% memory locations and 12.5% computations are needed to represent the spectral information of the original wave components until the half Nyquist frequency. Fig. 1 shows the nodes (in black) that should be updated at half the sampling rate on an orthogonal FDTD scheme in order to maintain spectral content until the half Nyquist frequency contrasted with the original 8×15 grid providing spectral information until the Nyquist frequency.

Let us consider a uniform P-dimensional orthogonal grid $\mathcal{G}$ forming a hyper-rectangle whose i-th dimension counts M_i nodes. Hence $\mathcal{G}$ counts $M = \prod_{i=1}^{P} M_i$ nodes, and the decimated grid, that is useful until the half Nyquist frequency, contains $\prod_{i=1}^{P} \lfloor M_i/2 \rfloor \leq M/2^P$ nodes (The operator $\lfloor \ \rfloor$ rounds down to the nearest integer). The same relation holds for any (e.g. triangular [12], [13]) grid. Since the decimated scheme can run at half the original sampling rate, then the number of computations that need to be made after the decimation amounts to about $1/2^{P+1}$ times the computations in the original scheme.

Note that if the number of nodes is even in one direction, then the symmetry of the grid is lost after
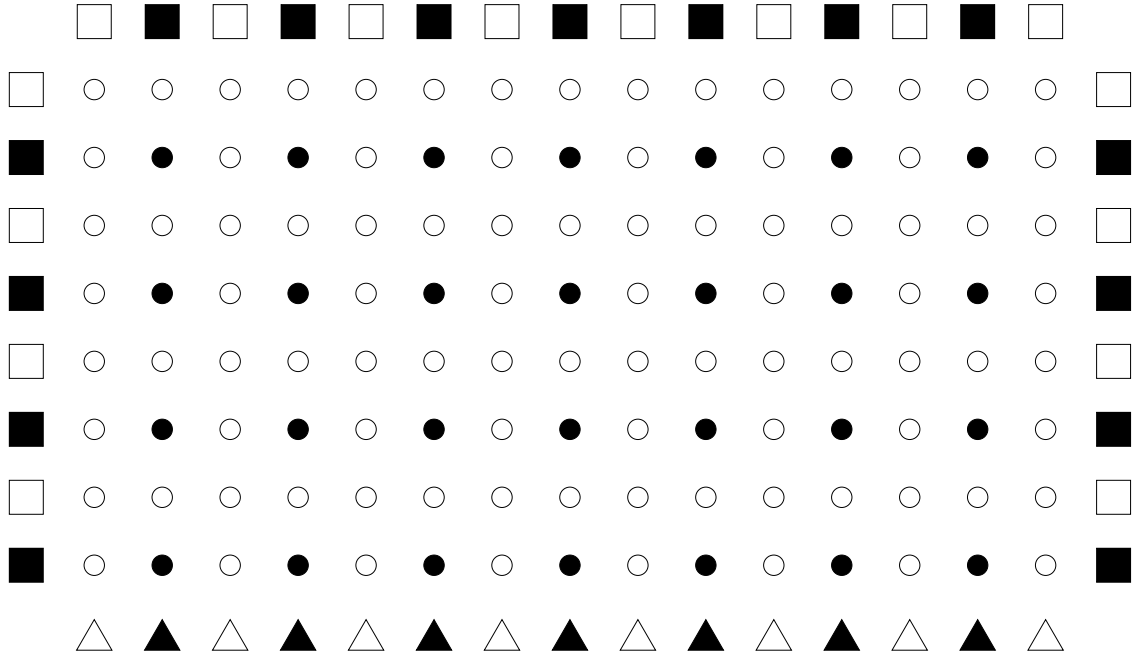
Fig. 1. 8×15 orthogonal FDTD grid and its decimated version. Black nodes and white nodes together provide spectral information until the Nyquist frequency; black nodes alone provide spectral information until half the Nyquist frequency. Square nodes define Neumann boundaries; triangular nodes define Dirichlet boundaries. Black squares and triangles define the decimated boundary. Note that the decimated grid is shifted toward the Dirichlet boundary, hence losing symmetry along the even-sized direction.

the decimation. For instance, in Fig. 1, the black dots are offset toward the bottom edge of the grid. Later we will see that the parity and the boundary conditions along each direction depend on each other through the type of Discrete Cosine Transform representing the decimated second-difference matrix.

We will call Decimated FDTD (DFDTD) scheme the set of black nodes in Fig. 1, along with the operator built upon them. As opposed to subgridding methods (based on spatial interpolations of different kinds) we propose an algebraic rearrangement of the time difference operator approximating the continuous differential problem. The resulting decimation in time and space of the numerical waves does not change the dynamic evolution and, hence, the stability properties of the original scheme.

Our work will deal with the linear second-order case, i.e., with schemes that integrate the signal numerically through the following system of difference equations:

$$\boldsymbol{u}^{n+1} = \boldsymbol{A}_M \boldsymbol{u}^n - \boldsymbol{B}_M \boldsymbol{u}^{n-1}, \tag{1}$$

in which

$$\boldsymbol{u}^n = \begin{pmatrix} u_0^n & \cdots & u_{M_1-1}^n & u_{M_1}^n & \cdots & u_{M_1 M_2-1}^n & u_{M_1 M_2}^n & \cdots & u_{M-1}^n \end{pmatrix}^T \tag{2}$$

is an array containing the values at time $t = n\delta T$ of a continuous signal $u(x_1, \ldots, x_P, t)$ sampled on the grid $\mathcal{G}$. The symbol δT represents the temporal step in seconds and δx_i will be the grid spacing in meters along the direction x_i, moreover ^{T} denotes transposition. The generic element in $\boldsymbol{u}^n$, hence, represents the signal value $u^n_{l_1,\ldots,l_P}$ on the node of spatial coordinate $(l_1, \ldots, l_P)$ at step n. Furthermore, $\boldsymbol{A}_M$ and $\boldsymbol{B}_M$ are $M \times M$ square matrices governing the time-difference operation.

Specifically, our analysis will be limited to 1D and 2D *isotropic lossless* models, both admitting a simple structure of these matrices. A focus on these models has little applicability; however it builds a formal platform, prospectively supporting future investigations on dissipative and/or anisotropic FDTD schemes. In this perspective, (1) neither poses constraints on the grid geometry nor on the spatial communication among nodes; moreover it can incorporate also implicit linear operators provided that the relation $\boldsymbol{C}_M \boldsymbol{u}^{n+1} = \boldsymbol{A}_M \boldsymbol{u}^n - \boldsymbol{B}_M \boldsymbol{u}^{n-1}$ admits an explicit formulation after inversion of $\boldsymbol{C}_M$.

Moving from relation (1), in Sec. II we introduce the 1D and 2D schemes under consideration along with a general form expressing their decimation in time. Based on this form, in Sec. III we derive a decimation in space of the 1D scheme thereby providing the equations which hold the necessary band limitation of the wave signals. These equations reconstruct the original spatial components and, ultimately, the exact runtime interpolation the signal. In Sec. IV a similar procedure is repeated for the 2D case; yet, in this case the exact reconstruction in space requires manipulating the decimated scheme in ways that maintain the structure of the original operator. The error resulting from this manipulation is evaluated, and then interpreted in terms of numerical dispersion. In Sec. V, the reconstruction problem in presence of dispersion and/or aliasing error will be examined through simulations. Sec. VI is finally devoted to understanding why time interpolation occasionally introduces singularities during the computation of the missing samples.

II. LOSSLESS ORTHOGONAL FDTD SCHEMES

In both the 1D and 2D case it can be observed that lossless propagation and reflection imply that $\boldsymbol{B}_M$ is the $M \times M$ identity matrix: $\boldsymbol{B}_M = \boldsymbol{I}_M$.

The simplest FDTD scheme modeling lossless wave transmission on an orthogonal grid is realized by

$$u^{n+1}_l = u^n_{l+1} + u^n_{l-1} - u^{n-1}_l \tag{3}$$

in one dimension, and by

$$u^{n+1}_{l,m} = \frac{u^n_{l+1,m} + u^n_{l-1,m} + u^n_{l,m+1} + u^n_{l,m-1}}{2} - u^{n-1}_{l,m} \tag{4}$$

in two dimensions [18]. Two alternative conditions are typically set to provide lossless boundary wave reflection: Dirichlet or Neumann. Dirichlet corresponds to a constant (usually null) wave field and Neumann to a constant field derivative at the boundary. If we label the boundaries of a 1D grid with -1 and M and M non-boundary nodes with $0, \ldots, M-1$, then when simulating a domain measuring $\ell_0 = (M-1)\delta X$ meters, every step n is such that:

$$u^n_{-1} = u^n_M = 0 \tag{5}$$

for Dirichlet and

$$\frac{u^n_1 - u^n_{-1}}{2} = \frac{u^n_M - u^n_{M-2}}{2} = 0 \tag{6}$$

for Neumann. Now, at the edges of an FDTD scheme governed by (3) it is easy to derive

$$\begin{aligned} u^{n+1}_0 &= u^n_1 - u^{n-1}_0 \\ u^{n+1}_{M-1} &= u^n_{M-2} - u^{n-1}_{M-1} \end{aligned} \tag{7}$$

for Dirichlet and

$$\begin{aligned} u^{n+1}_0 &= 2u^n_1 - u^{n-1}_0 \\ u^{n+1}_{M-1} &= 2u^n_{M-2} - u^{n-1}_{M-1} \end{aligned} \tag{8}$$

for Neumann, where the first derivative has been approximated on u^n_0 and u^n_{M-1} using normalized center differences. This type of approximation is also called *meshpoint* symmetry [19]. The extension of these formulas to 2D grids, as well as the introduction of mixed Dirichlet/Neumann conditions over different boundary regions follows immediately from either (7) or (8).

The structure of $\boldsymbol{A}_M$ in the 1D case follows from (3) and (7) and (8):

$$\boldsymbol{A}_M = \begin{pmatrix} 0 & \alpha_L & & & & \\ 1 & 0 & 1 & & & \\ & 1 & 0 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & 0 & 1 \\ & & & & \alpha_R & 0 \end{pmatrix}, \tag{9}$$

in which α_L and α_R are equal to 1 or 2 depending on the type (Dirichlet or Neumann). The left and right boundaries have been labeled respectively with subscripts L and R. If M is even, then $\boldsymbol{A}_M$ has full rank; conversely, if M is odd then its rank is equal to $M-1$. This property does not depend on the type of the boundary, and conditions the temporal interpolation of the signal as we will see later in Sec. III-A.

In addition, in the 2D case, if we set a uniform boundary type at each side of the rectangular region then from (4), (7) and (8) it follows that

$$\boldsymbol{A}_M = \frac{1}{2}\left(\boldsymbol{I}_{M_y} \otimes \boldsymbol{A}_{M_x} + \boldsymbol{A}_{M_y} \otimes \boldsymbol{I}_{M_x}\right),$$

where $\otimes$ denotes the Kronecker product, and $\boldsymbol{A}_{M_x}$, $\boldsymbol{A}_{M_y}$ have the same structure as in (9) accounting for the wave propagation and reflection respectively along the vertical and horizontal direction. For the details see [?, p. 254]. In particular, for rectangular domains, the signal $\boldsymbol{u}^n$ is more naturally represented by a matrix $\boldsymbol{U}^n$ whose elements correspond to the respective nodes in the orthogonal grid: $\boldsymbol{U}^n(l, m) = u^n_{M_1 l+m}$. For these domains the Kronecker product simplifies in ways that (1) can be written as:

$$\boldsymbol{U}^{n+1} - \frac{1}{2}(\boldsymbol{A}_{M_x}\boldsymbol{U}^n + \boldsymbol{U}^n\boldsymbol{A}_{M_y}) + \boldsymbol{U}^{n-1} = 0. \tag{10}$$

III. 1D DFDTD Scheme

Let $\boldsymbol{B}_M = \boldsymbol{I}_M$ in (1). By rewriting (1) three times respectively for computing $\boldsymbol{u}^n$, $\boldsymbol{u}^{n+1}$, and $\boldsymbol{u}^{n+2}$, we obtain a system of three equations spanning the discrete-time range $[n+2, n-2]$. It is straightforward to eliminate the unknowns $\boldsymbol{u}^{n+1}$ and $\boldsymbol{u}^{n-1}$ from this system, and come up with the following relation:

$$\boldsymbol{u}^{n+2} = (\boldsymbol{A}_M^2 - 2\boldsymbol{I}_M)\boldsymbol{u}^n - \boldsymbol{u}^{n-2}, \tag{11}$$

in which the node values at step $n + 2$ are expressed as a linear combination of $\boldsymbol{u}^n$ and $\boldsymbol{u}^{n-2}$.

Relation (11) describes a new lossless FDTD scheme, whose dynamic evolution equals the original except for the temporal step that is now twice as long. At this point it is convenient to label with $_{\downarrow Q}$ all symbols that refer to a temporal domain decimated by a factor Q, and thus we rewrite (11) as

$$\boldsymbol{u}^{n+1}_{\downarrow 2} = \boldsymbol{A}_{M\downarrow 2}\boldsymbol{u}^n_{\downarrow 2} - \boldsymbol{u}^{n-1}_{\downarrow 2}, \tag{12}$$

see also [21]. The decimated matrix $\boldsymbol{A}_{M\downarrow 2} = \boldsymbol{A}_M^2 - 2\boldsymbol{I}_M$ has the following structure:

$$\begin{pmatrix}
\alpha_L-2 & 0 & \alpha_L & & & & & \\
0 & \alpha_L-1 & 0 & 1 & & & & \\
1 & 0 & 0 & 0 & 1 & & & \\
& 1 & 0 & 0 & 0 & 1 & & \\
& & \ddots & & & & \ddots & \\
& & & 1 & 0 & 0 & 0 & 1 \\
& & & & 1 & 0 & \alpha_R-1 & 0 \\
& & & & & \alpha_R & 0 & \alpha_R-2
\end{pmatrix}.$$

Notably, $\boldsymbol{A}_{M\downarrow 2}$ computes the decimated signal in time at no additional cost compared to $\boldsymbol{A}_M$, since it still has two non-null elements in each row apart from the first and the last one.

We can sort the elements in $\boldsymbol{u}^n_{\downarrow 2}$ so that the even and odd-indexed values are in the upper and lower part of a new array. This rearrangement is obtained by means of a proper $M \times M$ permutation matrix $\boldsymbol{\Pi}_M$; more precisely, if we define $\boldsymbol{u}^n_{\mathcal{E}}$ and $\boldsymbol{u}^n_{\mathcal{O}}$ to contain the even and odd elements of $\boldsymbol{u}^n$, then $\boldsymbol{\Pi}^T_M \boldsymbol{u}^n = [\boldsymbol{u}^n_{\mathcal{E}}\, \boldsymbol{u}^n_{\mathcal{O}}]^T$. At this point, (12) can be rewritten in the permuted array $\boldsymbol{\Pi}^T_M \boldsymbol{u}^n_{\downarrow 2}$:

$$\boldsymbol{\Pi}^T_M \boldsymbol{u}^{n+1}_{\downarrow 2} = \boldsymbol{\Pi}^T_M \boldsymbol{A}_{M\downarrow 2} \boldsymbol{\Pi}_M \boldsymbol{\Pi}^T_M \boldsymbol{u}^n_{\downarrow 2} - \boldsymbol{\Pi}^T_M \boldsymbol{u}^{n-1}_{\downarrow 2}, \tag{13}$$

in which we make use of the property $\boldsymbol{\Pi}^T_M \boldsymbol{\Pi}_M = \boldsymbol{I}_M$ of the permutation matrix [?, p. 6].

The matrix $\boldsymbol{P}_{M\downarrow 2} = \boldsymbol{\Pi}^T_M \boldsymbol{A}_{M\downarrow 2} \boldsymbol{\Pi}_M$ governs the evolution of the permuted system. This matrix is made of two anti-diagonal blocks containing null elements, plus two diagonal square blocks $\boldsymbol{Q}_{M\downarrow 2}$ and $\boldsymbol{R}_{M\downarrow 2}$. These blocks have sizes equal to either $(M+1)/2$ and $(M-1)/2$ if M is odd, or to $M/2$ if M is even. In either case we conclude that the time-decimated scheme in the 1D case operates through (12) on the even and odd nodes separately via the sub-matrices $\boldsymbol{Q}_{M\downarrow 2}$ and $\boldsymbol{R}_{M\downarrow 2}$:

$$\begin{cases} \boldsymbol{u}^{n+1}_{\mathcal{E}\downarrow 2} & = \boldsymbol{Q}_{M\downarrow 2} \boldsymbol{u}^n_{\mathcal{E}\downarrow 2} - \boldsymbol{u}^{n-1}_{\mathcal{E}\downarrow 2} \\[3mm] \boldsymbol{u}^{n+1}_{\mathcal{O}\downarrow 2} & = \boldsymbol{R}_{M\downarrow 2} \boldsymbol{u}^n_{\mathcal{O}\downarrow 2} - \boldsymbol{u}^{n-1}_{\mathcal{O}\downarrow 2} \end{cases}. \tag{14}$$

If M is even then the square sub-matrices are equal to

$$\boldsymbol{Q}_{M/2} = \begin{pmatrix} \alpha_L - 2 & \alpha_L & & & & \\ 1 & 0 & 1 & & & \\ & & \ddots & & \ddots & \\ & & 1 & 0 & & 1 \\ & & & 1 & \alpha_R - 1 \end{pmatrix}, \tag{15}$$

and

$$\boldsymbol{R}_{M/2} = \begin{pmatrix} \alpha_L - 1 & 1 & & & & \\ 1 & 0 & 1 & & & \\ & & \ddots & & \ddots & \\ & & 1 & 0 & & 1 \\ & & & \alpha_R & \alpha_R - 2 \end{pmatrix}. \tag{16}$$

Now, if the left boundary is modeled with Neumann and the right one with Dirichlet then $\alpha_L = 2$ and $\alpha_R = 1$, hence $\boldsymbol{Q}_{M/2} = \boldsymbol{A}_{M/2}$. Symmetrically, if we swap the boundary types then $\alpha_L = 1$ and $\alpha_R = 2$, hence $\boldsymbol{R}_{M/2} = \boldsymbol{A}_{M/2}$.

Alternatively, if M is odd then the square sub-matrices are such that $\boldsymbol{Q}_{(M+1)/2} = \boldsymbol{A}_{(M+1)/2}$ if both boundaries are modeled with Neumann conditions, and similarly $\boldsymbol{R}_{(M-1)/2} = \boldsymbol{A}_{(M-1)/2}$ if both boundaries are modeled with Dirichlet conditions.

As we discussed in the introduction, the relationship between domain size and boundary conditions constrains the decimation in space: even-sized 1D FDTD grids result in $\boldsymbol{A}_{M/2}$ if one boundary is modeled with Dirichlet conditions and the other with Neumann conditions; odd-sized grids need Dirichlet or Neumann boundaries on both sides instead. The existence of a sub-matrix having the same structure as $\boldsymbol{A}_M$ is necessary for proceeding with the decimation in space, i.e., for limiting the computation to either the first or the second decimated scheme in (14) while possibly reconstructing $\boldsymbol{u}_{\downarrow 2}^n$ at any moment from either $\boldsymbol{u}_{\mathcal{E}\downarrow 2}^n$ or $\boldsymbol{u}_{\mathcal{O}\downarrow 2}^n$.

A. 1D Signal Reconstruction

At this point it is interesting to understand first whether $\boldsymbol{u}_{\downarrow 2}^n$ can be obtained by spatial interpolation of either $\boldsymbol{u}_{\mathcal{E}\downarrow 2}^n$ or $\boldsymbol{u}_{\mathcal{O}\downarrow 2}^n$, and secondly whether $\boldsymbol{u}^n$ can be obtained by temporal interpolation of $\boldsymbol{u}_{\downarrow 2}^n$.

The derivation makes use of an elegant result by Strang, who showed that each Discrete Cosine Transform basis is formed by the eigenvectors of a symmetric *second-difference matrix* [19]. This matrix is equal to $2\boldsymbol{I}_M - \boldsymbol{A}_M$ for an M-point transformation, and classifies all types of Discrete Cosine Transform (DCT) as well as Discrete Sine Transform (DST) depending on how the Dirichlet and Neumann conditions are formulated and then combined together in the corresponding second-difference scheme.

We show the derivation for even values of M, with Neumann conditions on the left boundary and Dirichlet conditions on the right. In this case (14) was shown to be equal to

$$\begin{cases} \boldsymbol{u}_{\mathcal{E}\downarrow 2}^{n+1} = \boldsymbol{A}_{M/2}\boldsymbol{u}_{\mathcal{E}\downarrow 2}^n - \boldsymbol{u}_{\mathcal{E}\downarrow 2}^{n-1} \\ \\ \boldsymbol{u}_{\mathcal{O}\downarrow 2}^{n+1} = \boldsymbol{R}_{M/2}\boldsymbol{u}_{\mathcal{O}\downarrow 2}^n - \boldsymbol{u}_{\mathcal{O}\downarrow 2}^{n-1} \end{cases}. \tag{17}$$

The eigenvalues and eigenvectors of $\boldsymbol{A}_M$ and $\boldsymbol{R}_M$ are known in closed form [19]. In the case under study, the eigenvectors of $\boldsymbol{A}_M$ form the basis $\boldsymbol{C}_M^{(III)} = \begin{pmatrix} \boldsymbol{c}_M^{(III)}(0) & \cdots & \boldsymbol{c}_M^{(III)}(M-1) \end{pmatrix}$ of the M-point DCT of type III, whereas the eigenvectors of $\boldsymbol{R}_M$ form the basis $\boldsymbol{C}_M^{(IV)}$ of the M-point DCT of type IV:

$$\begin{aligned} \boldsymbol{A}_M \boldsymbol{C}_M^{(III)} &= \boldsymbol{C}_M^{(III)} \boldsymbol{\Lambda}_M \\ \\ \boldsymbol{R}_M \boldsymbol{C}_M^{(IV)} &= \boldsymbol{C}_M^{(IV)} \boldsymbol{\Lambda}_M \end{aligned}, \tag{18}$$

in which the matrices of the eigenvectors have elements

$$\boldsymbol{C}_M^{(III)}(i,j) = \cos\tfrac{\pi}{M}i(j+\tfrac{1}{2})$$

$$\boldsymbol{C}_M^{(IV)}(i,j) = \cos\tfrac{\pi}{M}(i+\tfrac{1}{2})(j+\tfrac{1}{2})$$

(19)

and the matrix of eigenvalues has elements only in the diagonal:

$$\boldsymbol{\Lambda}_M(i,j) = \begin{cases} 2\cos\tfrac{\pi}{M}(j+\tfrac{1}{2}), & i=j \\ 0 & \text{otherwise} \end{cases},$$

(20)

with $i,j = 0,\ldots,M-1$.

Similar to the Fourier basis, both the type-III and type-IV DCT admit simple inversion of their matrix. In fact, $\boldsymbol{C}_M^{(IV)-1} = \tfrac{2}{M}\boldsymbol{C}_M^{(IV)}$ and $\boldsymbol{C}_M^{(III)-1} = \boldsymbol{C}_M^{(III)T}\boldsymbol{D}_M^2$, with $\boldsymbol{D}_M^2 = \text{diag}\left(\tfrac{1}{M},\tfrac{2}{M},\ldots,\tfrac{2}{M}\right)$ an $M\times M$ diagonal matrix. Hence,

$$\boldsymbol{A}_M = \boldsymbol{C}_M^{(III)}\boldsymbol{\Lambda}_M\boldsymbol{C}_M^{(III)T}\boldsymbol{D}_M^2$$

$$\boldsymbol{R}_M = \tfrac{2}{M}\boldsymbol{C}_M^{(IV)}\boldsymbol{\Lambda}_M\boldsymbol{C}_M^{(IV)}$$

(21)

Next, we need to recall that an M-point DCT can be factored into two $M/2$-point DCT's, possibly belonging to different types [22]. In our case we make use of the following relation:

$$\boldsymbol{\Pi}_M^T\boldsymbol{C}_M^{(III)} = \begin{pmatrix} \boldsymbol{C}_{M/2}^{(III)} & \boldsymbol{C}_{M/2}^{(III)}\boldsymbol{J}_{M/2} \\ \boldsymbol{C}_{M/2}^{(IV)} & -\boldsymbol{C}_{M/2}^{(IV)}\boldsymbol{J}_{M/2} \end{pmatrix},$$

(22)

in which $\boldsymbol{J}_M$ is an $M\times M$ exchange matrix [?, p. 215]. Since $\boldsymbol{\Pi}_M\boldsymbol{D}_M^2\boldsymbol{\Pi}_M^T = \boldsymbol{D}_M^2$, and by transposing (22), then:

$$\begin{aligned} \boldsymbol{C}_M^{(III)T}\boldsymbol{D}_M^2\boldsymbol{u}_{\downarrow2}^n &= \boldsymbol{C}_M^{(III)T}\boldsymbol{\Pi}_M\boldsymbol{D}_M^2\boldsymbol{\Pi}_M^T\boldsymbol{u}_{\downarrow2}^n \\ &= \boldsymbol{C}_M^{(III)T}\boldsymbol{\Pi}_M\begin{pmatrix} \tfrac{1}{2}\boldsymbol{D}_{M/2}^2\boldsymbol{u}_{\mathcal{E}\downarrow2}^n \\ \tfrac{2}{M}\boldsymbol{u}_{\mathcal{O}\downarrow2}^n \end{pmatrix} \\ &= \begin{pmatrix} \tfrac{1}{2}\boldsymbol{C}_{M/2}^{(III)T}\boldsymbol{D}_{M/2}^2\boldsymbol{u}_{\mathcal{E}\downarrow2}^n + \tfrac{2}{M}\boldsymbol{C}_{M/2}^{(IV)}\boldsymbol{u}_{\mathcal{O}\downarrow2}^n \\ \tfrac{1}{2}\boldsymbol{J}_{M/2}\boldsymbol{C}_{M/2}^{(III)T}\boldsymbol{D}_{M/2}^2\boldsymbol{u}_{\mathcal{E}\downarrow2}^n - \tfrac{2}{M}\boldsymbol{J}_{M/2}\boldsymbol{C}_{M/2}^{(IV)}\boldsymbol{u}_{\mathcal{O}\downarrow2}^n \end{pmatrix} \end{aligned}$$

(23)

If the upper half of the DCT coefficients are null then the lower $M/2$ elements of the vector (23) will satisfy:

$$\frac{2}{M}\boldsymbol{C}_{M/2}^{(IV)}\boldsymbol{u}_{\mathcal{O}\downarrow2}^n = \frac{1}{2}\boldsymbol{C}_{M/2}^{(III)T}\boldsymbol{D}_{M/2}^2\boldsymbol{u}_{\mathcal{E}\downarrow2}^n,$$

(24)

that is,

$$\boldsymbol{u}^n_{\mathcal{O}\downarrow2} = \boldsymbol{C}^{(IV)}_{M/2}\boldsymbol{C}^{(III)\,T}_{M/2}\boldsymbol{D}^2_{M/2}\boldsymbol{u}^n_{\mathcal{E}\downarrow2} = \boldsymbol{C}^{(IV)}_{M/2}\boldsymbol{C}^{(III)\,-1}_{M/2}\boldsymbol{u}^n_{\mathcal{E}\downarrow2}, \tag{25}$$

which proves that $\boldsymbol{u}^n_{\mathcal{O}\downarrow2}$ can be reconstructed without error by a type-III anti-transformation followed by a type-IV transformation of $\boldsymbol{u}^n_{\mathcal{E}\downarrow2}$.

Once we have reconstructed $\boldsymbol{u}^n_{\downarrow2}$ exactly in space from $\boldsymbol{u}^n_{\mathcal{O}\downarrow2}$ and $\boldsymbol{u}^n_{\mathcal{E}\downarrow2}$, then the signal can be interpolated across time. Specifically, from (1) and (12) we obtain

$$\boldsymbol{A}_M\boldsymbol{u}^{n-1} = \boldsymbol{u}^n + \boldsymbol{u}^{n-2} = \boldsymbol{u}^n_{\downarrow2} + \boldsymbol{u}^{n-1}_{\downarrow2}, \tag{26}$$

and finally $\boldsymbol{u}^{n-1}$ after solution of the system (26).

This completes the interpolation procedure when M is even. The condition of having the upper half of the DCT coefficients equal to zero is equivalent to having the energy of $\boldsymbol{u}^n$ bounded within the lower half of the spatial Fourier spectrum computed over the grid.

Similar results can be found if the Dirichlet or Neumann conditions are set on both sides of the boundary. Remember, however, that $\boldsymbol{A}_M$ is singular when M is odd: this prevents interpolation across time directly after matrix inversion in (26). In Sec. VI we will see how this singularity can be handled.

Conversely, the even size of the grid makes the interpolation particularly interesting when M is a power of two. In fact, if $M = 2^K$ then the decomposition (14) can be iterated k times with $k \leq K$, each time via a proper permutation yielding either $\boldsymbol{Q}_{M\downarrow2^k} = \boldsymbol{A}_{M/2^k}$ or $\boldsymbol{R}_{M\downarrow2^k} = \boldsymbol{A}_{M/2^k}$ depending on the boundary conditions. This decomposition permits to compute the corresponding 2^k sub-schemes independently[1]. If the spectral power of the signal is null in the upper $2^k - 1$ sub-bands, then a sufficient condition exists for the reconstruction of the original signal from its decimated version in space by a factor 2^k.

IV. 2D DFDTD SCHEME

We decimate (10) across time and come up with the following scheme:

$$\boldsymbol{U}^{n+1}_{\downarrow2} - \frac{1}{2}(\boldsymbol{A}_{M_x\downarrow2}\boldsymbol{U}^n_{\downarrow2} + \boldsymbol{U}^n_{\downarrow2}\boldsymbol{A}_{M_y\downarrow2}) + \boldsymbol{U}^{n-1}_{\downarrow2} = 0, \tag{27}$$

in which, by analogy with (12), $\boldsymbol{A}_{M_x\downarrow2} = \boldsymbol{A}^2_{M_x} - 2\boldsymbol{I}_{M_x}$ and $\boldsymbol{A}_{M_y\downarrow2} = \boldsymbol{A}^2_{M_y} - 2\boldsymbol{I}_{M_y}$. Again, we can multiply (27) by $\boldsymbol{\Pi}^T_{M_x}$ and $\boldsymbol{\Pi}_{M_y}$ respectively, hence obtaining

$$\widetilde{\boldsymbol{U}}^{n+1}_{\downarrow2} - \frac{1}{2}(\boldsymbol{P}_{M_x\downarrow2}\widetilde{\boldsymbol{U}}^n_{\downarrow2} + \widetilde{\boldsymbol{U}}^n_{\downarrow2}\boldsymbol{P}_{M_y\downarrow2}) + \widetilde{\boldsymbol{U}}^{n-1}_{\downarrow2} = 0 \tag{28}$$

---

[1]We deliberately avoid investigating the links existing between this decomposition and similar matrix rearrangements used by fast DCT algorithms [22]: such an investigation would, in fact, open a chapter going beyond the scope of this work.

with $\boldsymbol{P}_{M_x\downarrow2} = \boldsymbol{\Pi}_{M_x}^T \boldsymbol{A}_{M_x\downarrow2} \boldsymbol{\Pi}_{M_x}$ as well as $\boldsymbol{P}_{M_y\downarrow2} = \boldsymbol{\Pi}_{M_y}^T \boldsymbol{A}_{M_y\downarrow2} \boldsymbol{\Pi}_{M_y}$, and furthermore

$$\widetilde{\boldsymbol{U}}_{\downarrow2}^n = \boldsymbol{\Pi}_{M_x}^T \boldsymbol{U}_{\downarrow2}^n \boldsymbol{\Pi}_{M_y} = \begin{pmatrix} \boldsymbol{U}_{\mathcal{EE}\downarrow2}^n & \boldsymbol{U}_{\mathcal{EO}\downarrow2}^n \\ \boldsymbol{U}_{\mathcal{OE}\downarrow2}^n & \boldsymbol{U}_{\mathcal{OO}\downarrow2}^n \end{pmatrix} \tag{29}$$

which is a permutation containing the even-even, even-odd, odd-even, and odd-odd ordered elements of $\boldsymbol{U}_{\downarrow2}^n$ respectively in the first, second, third, and fourth quadrant.

Holding, for either direction, the same relationships between size and boundary conditions as those seen in the 1D case (i.e., Dirichlet/Neumann or Neumann/Dirichlet for even sizes, and Dirichlet/Dirichlet or Neumann/Neumann for odd sizes), then identical conclusions can be drawn about the structures of the sub-matrices forming $\boldsymbol{P}_{M_x\downarrow2}$ and $\boldsymbol{P}_{M_y\downarrow2}$. In particular, if both M_x and M_y are even and we set the boundary to Neumann at top and left as well as to Dirichlet at bottom and right, then $\boldsymbol{Q}_{M_x\downarrow2} = \boldsymbol{A}_{M_x/2}$ and $\boldsymbol{Q}_{M_y\downarrow2} = \boldsymbol{A}_{M_y/2}$. Therefore, from (28) and in analogy with (17),

$$\begin{cases} \boldsymbol{U}_{\mathcal{EE}\downarrow2}^{n+1} = \frac{1}{2}(\boldsymbol{A}_{M_x/2}\boldsymbol{U}_{\mathcal{EE}\downarrow2}^n + \boldsymbol{U}_{\mathcal{EE}\downarrow2}^n\boldsymbol{A}_{M_y/2}) - \boldsymbol{U}_{\mathcal{EE}\downarrow2}^{n-1} \\[2mm] \boldsymbol{U}_{\mathcal{EO}\downarrow2}^{n+1} = \frac{1}{2}(\boldsymbol{A}_{M_x/2}\boldsymbol{U}_{\mathcal{EO}\downarrow2}^n + \boldsymbol{U}_{\mathcal{EO}\downarrow2}^n\boldsymbol{R}_{M_y/2}) - \boldsymbol{U}_{\mathcal{EO}\downarrow2}^{n-1} \\[2mm] \boldsymbol{U}_{\mathcal{OE}\downarrow2}^{n+1} = \frac{1}{2}(\boldsymbol{R}_{M_x/2}\boldsymbol{U}_{\mathcal{OE}\downarrow2}^n + \boldsymbol{U}_{\mathcal{OE}\downarrow2}^n\boldsymbol{A}_{M_y/2}) - \boldsymbol{U}_{\mathcal{OE}\downarrow2}^{n-1} \\[2mm] \boldsymbol{U}_{\mathcal{OO}\downarrow2}^{n+1} = \frac{1}{2}(\boldsymbol{R}_{M_x/2}\boldsymbol{U}_{\mathcal{OO}\downarrow2}^n + \boldsymbol{U}_{\mathcal{OO}\downarrow2}^n\boldsymbol{R}_{M_y/2}) - \boldsymbol{U}_{\mathcal{OO}\downarrow2}^{n-1} \end{cases}. \tag{30}$$

A. 2D Signal Reconstruction

By applying the transformations (21) to the scheme (27), we can extend the reconstruction in space to the 2D case. In fact, a type-III DCT of $\boldsymbol{U}_{\downarrow2}^n$ can be obtained by reversing the permutation (29) and subsequently transforming:

$$\begin{aligned} \widehat{\boldsymbol{U}}_{\downarrow2}^n &= \boldsymbol{C}_{M_x}^{(III)\,-1} \boldsymbol{U}_{\downarrow2}^n \boldsymbol{C}_{M_y}^{(III)} \\ &= \boldsymbol{C}_{M_x}^{(III)\,-1} \boldsymbol{\Pi}_{M_x} \widetilde{\boldsymbol{U}}_{\downarrow2}^n \boldsymbol{\Pi}_{M_y}^T \boldsymbol{C}_{M_y}^{(III)}, \end{aligned} \tag{31}$$

in which the transformed matrix has been denoted with a hat. By factoring out (22) as

$$\boldsymbol{\Pi}_M^T \boldsymbol{C}_M^{(III)} = \begin{pmatrix} \boldsymbol{C}_{M/2}^{(III)} & \\ & \boldsymbol{C}_{M/2}^{(IV)} \end{pmatrix} \begin{pmatrix} \boldsymbol{I}_{M/2} & \boldsymbol{J}_{M/2} \\ \boldsymbol{I}_{M/2} & -\boldsymbol{J}_{M/2} \end{pmatrix}, \tag{32}$$

then (31) can be rewritten as

$$\widehat{U}_{\downarrow 2}^n = \frac{1}{2} \begin{pmatrix} I_{M_x/2} & I_{M_x/2} \\ J_{M_x/2} & -J_{M_x/2} \end{pmatrix} \cdot$$
$$\cdot \begin{pmatrix} \widehat{U}_{\mathcal{EE}\downarrow 2}^n & \widehat{U}_{\mathcal{EO}\downarrow 2}^n \\ \widehat{U}_{\mathcal{OE}\downarrow 2}^n & \widehat{U}_{\mathcal{OO}\downarrow 2}^n \end{pmatrix} \begin{pmatrix} I_{M_y/2} & J_{M_y/2} \\ I_{M_y/2} & -J_{M_y/2} \end{pmatrix}, \tag{33}$$

Now, if only the top-left quadrant of $\widehat{U}_{\downarrow 2}^n$ is non null then (33) can be split in four subsystems, whose solution and subsequent inverse transformation gives

$$U_{\mathcal{EO}\downarrow 2}^n = U_{\mathcal{EE}\downarrow 2}^n C_{M_y/2}^{(III)} C_{M_y/2}^{(IV)-1}$$
$$U_{\mathcal{OE}\downarrow 2}^n = C_{M_x/2}^{(IV)} C_{M_x/2}^{(III)-1} U_{\mathcal{EE}\downarrow 2}^n \tag{34}$$
$$U_{\mathcal{OO}\downarrow 2}^n = C_{M_x/2}^{(IV)} C_{M_x/2}^{(III)-1} U_{\mathcal{EE}\downarrow 2}^n C_{M_y/2}^{(III)} C_{M_y/2}^{(IV)-1}$$

As in the 1D case, once we have reconstructed $U_{\downarrow 2}^n$ exactly in space then the signal can be interpolated across time via solution of the linear system resulting directly from (10) and (27):

$$\frac{A_{M_x} U^{n-1} + U^{n-1} A_{M_y}}{2} = U^n + U^{n-2} = U_{\downarrow 2}^n + U_{\downarrow 2}^{n-1}. \tag{35}$$

This completes the interpolation procedure when M_x and M_y are even, under the hypothesis of null Fourier power spectrum above the half Nyquist frequency for either spatial direction.

In analogy with (26), the invertibility of the system (35) is not guaranteed. A type-III DCT of this system in fact gives

$$\frac{\Lambda_{M_x} \widehat{U}^{n-1} + \widehat{U}^{n-1} \Lambda_{M_y}}{2} = \widehat{U}^n + \widehat{U}^{n-2}. \tag{36}$$

Hence, element by element, and then forming the reciprocal:

$$\widehat{U}^{n-1}(i,j) = 2 \frac{\widehat{U}^n(i,j) + \widehat{U}^{n-2}(i,j)}{\lambda_{M_x}(i) + \lambda_{M_y}(j)}. \tag{37}$$

We conclude that (35) cannot be solved due to components (i,j) such that $\lambda_{M_x}(i) + \lambda_{M_y}(j)$ is null. An interpretation of this fact will be given in Sec. VI.

B. Dispersion Error

We have been left with the question whether (27) (in spite of its tractability) represents an exact temporal decimation of (10). Unfortunately, if the calculations leading to (11) are applied to (10) then a different 2D scheme is found instead. We keep track of this difference by denoting the signal at step n

in the new scheme with V^n instead of U^n. More precisely, by rewriting (10) three times respectively for computing V^n, V^{n+1} and V^{n+2}, we now obtain the following scheme in decimated notation:

$$V^{n+1}_{\downarrow 2} + 2V^n_{\downarrow 2} + V^{n-1}_{\downarrow 2} -$$
$$- \frac{1}{4}(A^2_{M_x}V^n_{\downarrow 2} + 2A_{M_x}V^n_{\downarrow 2}A_{M_y} + V^n_{\downarrow 2}A^2_{M_y}) = 0, \tag{38}$$

Here, both V^{n+1} and V^{n-1} have been eliminated by summing up the three resulting equations, after pre-multiplying (10) by $A_{M_x}/2$ when computing V^{n+1} as well as post-multiplying (10) by $A_{M_y}/2$ when computing V^{n+2} from the two previous time steps.

The scheme (38), then, decimates (10) exactly in time. As opposed to what happened in the 1D case, there is no exact way to put this scheme in the same form as (27). This discrepancy is not surprising: in fact, (38) characterizes a time-decimated FDTD scheme having the same *dispersion* as the original one.

Dispersion is a known phenomenon affecting numerical wave propagation, which prevents multidimensional schemes from simulating the isotropic behavior of continuous media exactly [23]. In particular, with increasing grid sparsity, the multidimensional waves exhibit widening dispersion. Since spatial decimation leads to node distances that are twice as long, one must expect that a 2D DFDTD scheme is affected by greater dispersion compared to the original scheme. Now, if (38) also permitted exact decimation in space then the resulting DFDTD scheme would paradoxically be affected by the same dispersion as the original one; hence, (38) cannot exhibit exact decimation in space.

Dispersion is computed by performing a Von Neumann analysis [23], linking each wave component of spatial frequency (ξ_x, ξ_y) traveling along the grid across two adjacent temporal steps through the amplification factor $\nu(\xi_x, \xi_y)$:

$$\mathcal{F}\{u^n\}(\xi_x, \xi_y) = \nu(\xi_x, \xi_y)\,\mathcal{F}\{u^{n-1}\}(\xi_x, \xi_y), \tag{39}$$

in which $\mathcal{F}\{u^n\}(\xi_x, \xi_y)$ is the 2D Fourier transform of the signal $u_{l,m}$ at step n. In particular, the argument of $\nu(\xi_x, \xi_y)$ is the temporal phase difference between the spatial component values computed on such two steps, hence it is proportional to their relative phase velocity. For 2D schemes such as (4), the Von Neumann analysis concludes that

$$\nu(\xi_x, \xi_y) = |\nu(\xi_x, \xi_y)|e^{i\arg\nu(\xi_x,\xi_y)} = e^{i\arctan\frac{\sqrt{4-b(\xi_x,\xi_y)^2}}{b(\xi_x,\xi_y)}}, \tag{40}$$

with $b(\xi_x, \xi_y) = \cos(2\pi\xi_x\delta x) + \cos(2\pi\xi_y\delta y)$.

We define the dispersion error to be the ratio between the phase difference in (27), given by (40) when the distance between adjacent nodes is such that $\delta x = \delta y = 2\delta X$, and the phase difference in (38), again
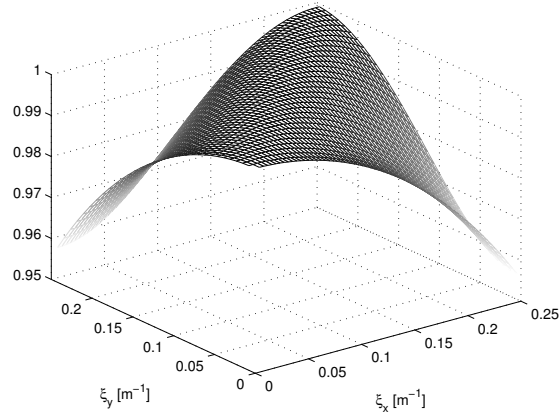
Fig. 2. Dispersion error $D_{\downarrow 2}(\xi_x, \xi_y)$ with node distance $\delta X = 1$.

given by (40) when the spatial components travel the original grid across two time steps:

$$
\begin{aligned}
D_{\downarrow 2}(\xi_x, \xi_y) &= \frac{\mathcal{F}\{u^n\}(2\xi_x, 2\xi_y)/\mathcal{F}\{u^{n-1}\}(2\xi_x, 2\xi_y)}{\mathcal{F}\{u^n\}(\xi_x, \xi_y)/\mathcal{F}\{u^{n-2}\}(\xi_x, \xi_y)} \\
&= \frac{\arg \nu(2\xi_x, 2\xi_y)}{2 \arg \nu(\xi_x, \xi_y)} \\
&= \frac{\arctan \frac{\sqrt{4-b(2\xi_x, 2\xi_y)^2}}{b(2\xi_x, 2\xi_y)}}{2 \arctan \frac{\sqrt{4-b(\xi_x, \xi_y)^2}}{b(\xi_x, \xi_y)}}
\end{aligned}
. \tag{41}
$$

This error is limited to the spatial frequency components lying below half the Nyquist value $\xi_{\text{MAX}} = 1/(2\delta X)$ m^{-1}.

In Fig. 2 we have plotted $D_{\downarrow 2}(\xi_x, \xi_y)$ for positive spatial frequency values within the half Nyquist frequency, i.e. in the range $[0, \xi_{\text{MAX}}/2] = [0, 1/(4\delta X)]$ along both directions, additionally setting a node distance $\delta X = 1$.

V. SIMULATIONS

A simulation of the schemes (27) and (38) confirms the existence of the error shown in Fig. 2. Fig. 3 shows the outputs $v^n_{\downarrow 2}$ and $u^n_{\downarrow 2}$ from a node near one corner of a 64×64 grid that has been clamped at two neighboring edges through Dirichlet boundaries and meanwhile set free at the other two edges through Neumann boundaries. The simulation has been initialized to compute the component (28,2) by means of either (38) (black line) or (27) (gray line). Scheme (38), in fact, anticipates scheme (27) in ways that the respective components go into phase opposition after about 15.5 periods. Now,
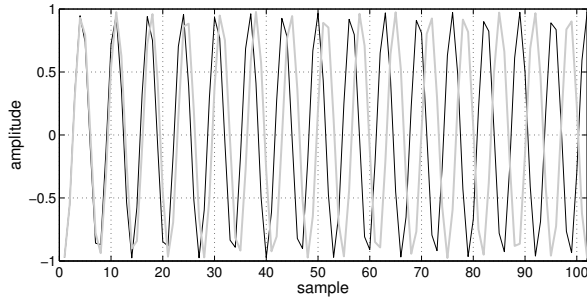
Fig. 3. Outputs from a 64×64 grid respectively realizing the scheme (38, black line) and (27, gray line), both initialized with mode (28,2).

$(\xi_x, \xi_y)_{(28,2)} = (28/128, 2/128) = (0.219, 0.016)$ and finally, from (41), $D_{\downarrow 2}(0.219, 0.016) = 0.967$. This means that the latter wave component is about 3.3% slower than the former. Hence, after 15.5 periods it gets delayed by $0.033 \cdot 15.5 = 0.511$ periods, and is close to being in opposite phase, as shown in the plots.

In the following we give temporal plots showing the effects of signal reconstruction for three different conditions: i) undispersed, non-aliased components, ii) dispersed, non-aliased components, and iii) undispersed, aliased components. The initialization is done by setting the grid at $n = 0$ to the desired initial value, e.g. to reproduce a spatial mode or a superposition of modes like that in Fig. 5. The reconstruction is figured out by starting from $U^n_{\mathcal{EE}\downarrow 2}$ and $U^{n-1}_{\mathcal{EE}\downarrow 2}$ and then computing, in sequence, $U^{n-1}_{\downarrow 2}$ and $U^n_{\downarrow 2}$ using (34), $\widehat{U}^{n-1}_{\downarrow 2} = \widehat{U}^{n-2}$ and $\widehat{U}^n_{\downarrow 2} = \widehat{U}^n$ using (31), and $\widehat{U}^{n-1}$ using (37). Finally, the inverse transformation of $\widehat{U}^{n-1}$ gives, along with $U^n_{\downarrow 2} = U^n$, the initial conditions for the computation of the reconstructed scheme (10).

All such conditions are applied to the previous scheme. As before, signals have been picked up nearby one corner. Notice, however, that "non-aliased" means that the grid initialization at temporal step $n = 0$ does not alias in space; this constraint does not prevent the same signal from aliasing in time, as it happens in practice with any initial excitation that is applied when the grid is at rest.

i) Undispersed wave components that can be decimated without aliasing follow the original evolution accurately (Fig. 4). Deviations occur because of some unavoidable spectral leakage due to the aforementioned time aliasing, even in the case of the low-frequency excitation selected for the example; these deviations are especially visible when a wavefront (that is always rich in high frequencies) reaches the pick-up node. Once (i.e., at step $n = 970$) the original grid is restored, the reconstructed signal follows the decimated signal trajectory and not the original trajectory
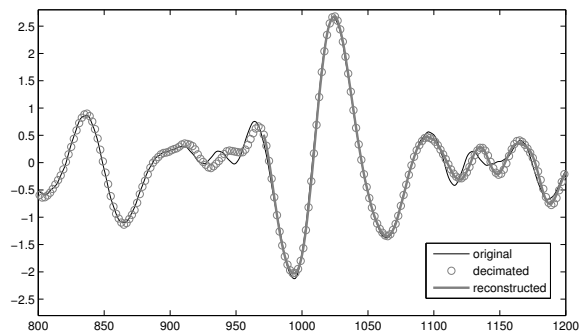
Fig. 4. Condition i). Output from a 64×64 grid (thin black line), its decimated version (gray circles), and reconstruction starting at temporal step 970 (thick gray line). Original grid initialized with mode (2,2).
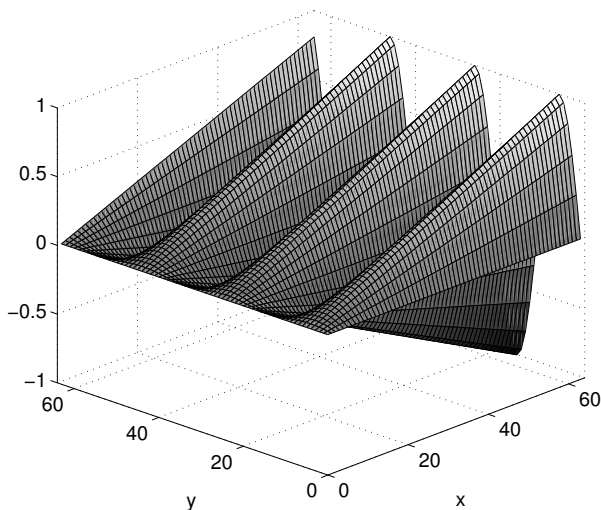

Fig. 5. Condition ii). Initial condition.

since the spectral information above the half Nyquist frequency has been obviously lost during the decimation.

ii) Dispersed wave components are introduced in the grid using the initial excitation shown in Fig. 5, in which a sinusoidal wave along one direction has been weighted by a ramp oriented in the opposite direction. This signal is decimated with low aliasing, but soon deviates from the original trajectory because of the different dispersion error in the original and the decimated scheme (Fig. 6, above), expressed by (41). The switch back to the original scheme at step $n = 970$ restores the original
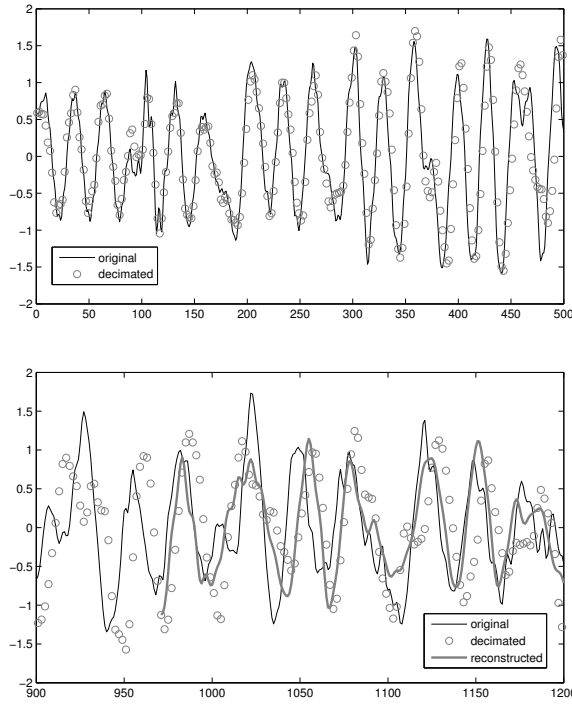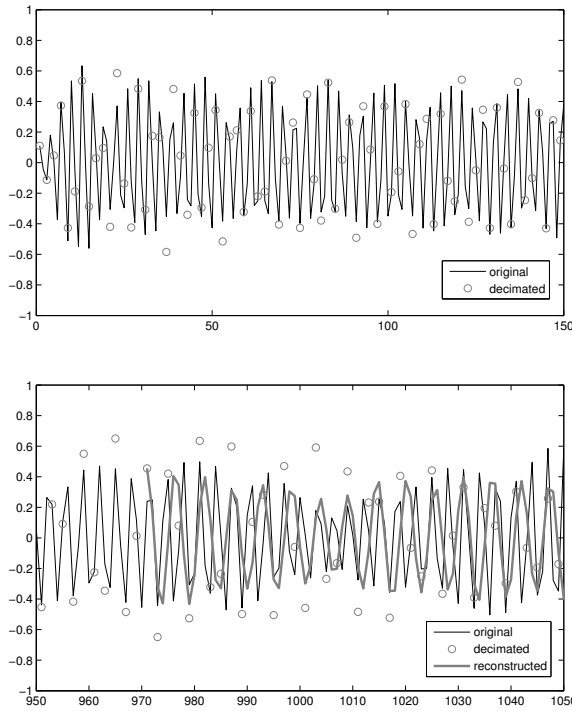
Fig. 6. Condition ii). Above: output from a 64×64 grid (thin black line) and its decimated version (gray circles). Below: in addition to the previous two signals, reconstruction starting at temporal step 970 (thick gray line). Original grid initialized with the signal shown in Fig. 5.

dispersion, in ways that the reconstructed signal in turn deviates from the decimated and also the original trajectory (Fig. 6, below), again by (41).

iii) Undispersed wave components that are decimated with aliasing (e.g. the diagonal traveling wave associated to the mode (41,41)), fold over in a new component within the half Nyquist frequency (Fig. 7, above). The switch back to the original scheme at step $n = 970$ preserves this component, and the reconstructed signal progressively overwrites the decimated trajectory since the reconstruction inherits the original initial condition (Fig. 7, below).

VI. CONSIDERATIONS ON THE RANK OF THE SECOND-DIFFERENCE MATRICES

The computation of (3) provides an exact discrete solution of the 1D standard wave equation, once the differential problem in continuous time and space has been instantiated with boundary and initial conditions [18]. In particular, Eqs. (3) and (4) compute spatial components that are located exactly at the Nyquist frequency: For example, an odd-sized 1D grid configuration such that $\boldsymbol{u}^{n-1} = (1, -1, \ldots, -1, 1)$

Fig. 7. Condition iii). Above: output from a 64×64 grid (thin black line) and its decimated version (gray circles). Below: in addition to the previous two signals, reconstruction starting at temporal step 970 (thick gray line). Original grid initialized with mode (41,41).

and $\boldsymbol{u}^n = (-1, 1, \ldots, 1, -1)$, holding Neumann conditions at both boundaries. Such components appear at any pick-up node in form of a discrete-time signal $u^n = (-1)^n$, which cannot be interpolated into an analog output according to the Nyquist-Shannon theorem.

Specifically under Neumann or Dirichlet boundary conditions, standing waves appear in the continuous solution in the form of cosine or sine functions whose spatial frequency is a multiple of the *fundamental wavelength* [2] $2\ell_0$. Consequently, the spatial period of the fundamental component is equal to $4\pi\ell_0$. If c_0 is the wave propagation speed of the medium, the corresponding temporal component is thus periodic by angular frequency $\omega_0 = 2\pi f_0 = \pi c_0 / \ell_0$.

Such components, then, form a numerable set of infinitely many *harmonic* functions that are proportional to either

$$y_{NN}(kx) = \cos\left(\frac{\pi k x}{\ell_0}\right) \quad \text{or} \quad y_{DD}(kx) = \sin\left(\frac{\pi k x}{\ell_0}\right) \tag{42}$$

[2] Recall that in Sec. II we defined ℓ_0 as the physical length of the domain.

respectively for Neumann or Dirichlet conditions, with[3] $k = 0, 1, \ldots$.

As also mentioned in Sec. III, decimation in presence of an odd-sized grid is possible if both boundaries are subject to the same condition (e.g. Neumann or Dirichlet). Under these conditions, the second-difference matrix $\boldsymbol{A}_M$ has a simple null eigenvalue $\boldsymbol{\Lambda}_M\left(\frac{M-1}{2}, \frac{M-1}{2}\right)$ with associated eigenvector either

$$\boldsymbol{c}_M^{(I)}\left(\tfrac{M-1}{2}\right) = \cos\frac{\pi}{2}i \quad \text{or} \quad \boldsymbol{s}_M^{(I)}\left(\tfrac{M-1}{2}\right) = \sin\frac{\pi}{2}(i+1) \tag{43}$$

with $i = 0, \ldots, M-1$, holding the Neumann or Dirichlet condition respectively. Now, by (18) it descends that $\boldsymbol{c}_M^{(I)}(\frac{M-1}{2})$ and $\boldsymbol{s}_M^{(I)}(\frac{M-1}{2})$ are the $\frac{M-1}{2}$-th eigenvector of the type-I DCT and the type-I DST respectively, both computed as usual on M nodes [19].

Both of these eigenvectors are parallel to signal components that are located exactly at half the Nyquist frequency. In the light of the Nyquist-Shannon theorem, these components cannot be reconstructed from the decimated signal. On the other hand they are reproduced by the decimated scheme described at the beginning of this section, and additionally are part of the solution of the differential problem. In fact, in both cases any reasonable numerical solution must contain a fundamental component proportional from (42) to either $y_{NN}(x)$ or $y_{DD}(x)$. For example, in the Neumann case, the fundamental numerical component must have discrete wavelength $2\tilde{\ell}_0 = 2(M-1)\delta X$, discrete wave speed $\tilde{c}_0 = \delta X/\delta T$, and hence discrete angular frequency

$$\tilde{\omega}_0 = 2\pi f_0 \delta T = 2\pi \frac{\tilde{c}_0}{2\tilde{\ell}_0}\delta T = \frac{\pi}{M-1} \quad \text{rad/sample}. \tag{44}$$

Hence, $\boldsymbol{c}_M^{(I)}(\frac{M-1}{2})$ has angular frequency equal to $(M-1)\,\tilde{\omega}_0/2 = \pi/2$, that is, a multiple integer of ω_0 as soon as one naturally sets the domain length and the discrete wave speed to be equal to the respective physical counterparts: $\tilde{\ell}_0 = \ell_0$ and $\tilde{c}_0 = c_0$.

In Sec. II we also stated that if M is even then $\boldsymbol{A}_M$ has full rank. In fact, holding the Neumann and Dirichlet conditions alternatively on either boundary, the differential problem exhibits standing wave components that are proportional to

$$y_{ND}(kx) = \cos\left((k + \tfrac{1}{2})\frac{\pi}{M}\frac{x}{\delta X}\right), \tag{45}$$

with $k = 0, 1, \ldots$ and $x \in [0, M\delta X]$.

Once again, from (19) we can see that $\boldsymbol{c}_M^{(III)}(\frac{M-1}{2})$ would be parallel to the component at the half Nyquist frequency if M were odd. Since M is even instead, then the standing wave $y_{ND}(\frac{M-1}{2}x)$ is not accounted for by the numerical solution of the original scheme.

---

[3]The standing wave component resulting from $k = 0$ in practice does not oscillate below the fundamental, since its period is infinitely long.

The 2D case extends this interpretation. Briefly, the system (35) does not admit solution in correspondence of the standing wave components that are parallel to the eigenvectors and whose associated eigenvalues sum up to zero. For instance, if all boundary regions are modeled by Neumann or Dirichlet in a 2D FDTD scheme with odd size on both directions then such components reduce to the null pair $\left(\lambda_{M_x}\left(\frac{M_x-1}{2}\right), \lambda_{M_y}\left(\frac{M_y-1}{2}\right)\right)$, accounting for orthogonal standing waves both oscillating at the half Nyquist frequency.

More specifically, one wonders whether combinations exist in which two non-null eigenvalues $\lambda_{M_x}(i), \lambda_{M_y}(j)$ with opposite sign sum up to zero. Indeed, this cannot happen as far as the scheme processes components only within the half Nyquist frequency: $i \leq (M_x - 1)/2$ and $j \leq (M_y - 1)/2$. One example is a square grid having a Neumann/Dirichlet boundary configuration on both directions, for which $M_x = M_y = M$ is even. The sum in fact amounts to

$$
\begin{aligned}
\lambda_M(i) + \lambda_M(j) &= 2\cos\tfrac{\pi}{M}\left(i + \tfrac{1}{2}\right) + 2\cos\tfrac{\pi}{M}\left(j + \tfrac{1}{2}\right) \\
&= 4\cos\tfrac{\pi}{2M}(i + j + 1)\cos\tfrac{\pi}{2M}(i - j),
\end{aligned}
\tag{46}
$$

with $i, j = 0, \ldots, M - 1$. From (46) it follows that all components such that $i + j = M - 1$ cannot be reconstructed. As we have seen for 1D schemes, the only pair of components until the half Nyquist frequency fulfilling this condition would be the null pair, i.e., $i = j = (M-1)/2$. This pair does not exist since the length is even along both directions. Also in this case, there are no numerical standing waves having angular frequency equal to $(M - 1)\tilde{\omega}_0/2 = \pi/2$, i.e., oscillating at the half Nyquist frequency.

In conclusion, it is worth mentioning that both (26) and (37) admit a generalized solution if their rank is not full. This solution makes use of the Moore-Penrose *pseudo-inverse* [24], which in practice neglects the dependent eigenvectors. By avoiding the singularity at the half Nyquist frequency, accurate wave interpolations have been obtained at runtime also in presence of simple absorbing boundaries, i.e., with $B_M \neq I_M$ [25]. Yet, the reliability of such solutions cannot be explained based on the results presented here and, hence, remains to be rigorously studied.

VII. CONCLUSION

Standard orthogonal 1D and 2D FDTD schemes modeling isotropic lossless wave propagation have been shown to admit decimation and interpolation in time and space of their grids at runtime, holding conventional Nyquist-Shannon hypotheses on the bandwidth of the wave signals. Although lossless models have limited application, the results found so far encourage the investigation of FDTD scheme decimation also in the presence of boundary absorption and/or viscosity of the transmission medium. Likely such an investigation may ask for substituting the DCT, of which we made intensive use while developing

our formal derivations, with other mathematical tools able to incorporate dissipation: to this regard, interesting results have been found which allow for switching the sampling frequency of digital filters at runtime [21]; concerning FDTD methods instead, so far we have been able to decimate in time and then interpolate simple dissipative models at runtime by merely applying, in sequence, formulas (11) and (26) for the 1D case as well as (27) and (37) for the 2D case, in which we put frequency-independent dissipative terms in the respective second-difference matrices [25]. These simulations look attractive in terms of a prospective development of the proposed concepts, incorporating dissipation as well.

The generalization of the work done here to the 3D case does not promise to bring answers as important as those posed by the questions on the decimation of dissipative schemes. Rather, a gap which may be conveniently filled has been left at the end of Sec. III, concerning the potentially insightful relationship existing between the proposed decimation/interpolation procedure and the known matrix division and merge operations that are correspondingly implemented by fast DCT algorithms.

REFERENCES

[1] D. Sullivan, *Electromagnetic simulation using the FDTD method.* Wiley-IEEE Press, 2013.

[2] D. Botteldooren, "Finite-difference time-domain simulation of low-frequency room acoustic problems," *J. of the Acoustical Society of America*, vol. 98, no. 6, pp. 3302–3308, 1995.

[3] S. Bilbao, "Optimized FDTD schemes for 3-D acoustic wave propagation," *IEEE Trans. on Audio, Speech and Language Processing*, vol. 20, no. 5, pp. 1658–1663, 2012.

[4] C. J. Webb and S. Bilbao, "Computing room acoustics with CUDA-3D FDTD schemes with boundary losses and viscosity," in *Proc. Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP).* Prague, Czech Republic: IEEE, May 2011, pp. 317–320.

[5] S. Bilbao, A. Torin, P. Graham, J. Perry, and G. Delap, "Modular physical modeling synthesis environments on GPU," in *Proc. of the International Computer Music Conference*, Athens, Greece, Sep. 14–20 2014.

[6] S. Bilbao and C. Webb, "Physical modeling of timpani drums in 3D on GPGPUs," *J. of the Audio Engineering Society*, vol. 61, no. 10, Oct. 2013.

[7] F. Avanzini and D. Rocchesso, "Modeling collision sounds: Non-linear contact force," in *Proc. Conf. on Digital Audio Effects (DAFX-01)*, Limerick, Ireland, Dec. 2001, pp. 61–66.

[8] S. Papetti, F. Avanzini, and D. Rocchesso, "Numerical methods for a nonlinear impact model: a comparative study with closed-form corrections," *IEEE Trans. on Audio, Speech and Language Processing*, vol. 19, no. 7, pp. 2146–2158, 2011.

[9] D. Rocchesso, R. Bresin, and M. Fërnstrom, "Sounding objects," *IEEE Multimedia*, vol. 10, no. 2, 2003.

[10] B. Bank, S. Zambon, and F. Fontana, "A modal-based real-time piano synthesizer," *IEEE Trans. on Audio, Speech and Language Processing*, vol. 18, no. 4, pp. 809–821, 2010, special Issue on Virtual Analog Audio Effects and Musical Instruments.

[11] N. Bonneel, G. Drettakis, N. Tsingos, I. Viaud-Delmon, and D. James, "Fast modal sounds with scalable frequency-domain synthesis," in *ACM SIGGRAPH 2008 papers*, ser. SIGGRAPH '08. New York, NY, USA: ACM, 2008, pp. 24:1–24:9. [Online]. Available: http://doi.acm.org/10.1145/1399504.1360623

[12] F. Fontana and D. Rocchesso, "Signal-theoretic characterization of waveguide mesh geometries for models of two-dimensional wave propagation in elastic media," *IEEE Trans. on Speech and Audio Processing*, vol. 9, no. 2, pp. 152–161, Feb. 2001.

[13] G. Campos and D. Howard, "On the computational efficiency of different waveguide mesh topologies for room acoustic simulation," *IEEE Trans. on Speech and Audio Processing*, vol. 13, no. 5, pp. 1063–1072, 2005.

[14] B. Hamilton and S. Bilbao, "Fourth-order and optimised finite difference schemes for the 2-D wave equation," in *Proc. Conf. on Digital Audio Effects (DAFX-13)*, Maynooth, Ireland, Sep 2-5 2013, p. 015120.

[15] K. Kowalczyk and M. van Walstijn, "Modeling frequency-dependent boundaries as digital impedance filters in FDTD and K-DWM room acoustics simulations," *J. of the Audio Engineering Society*, vol. 56, no. 7/8, pp. 569–583, 2008. [Online]. Available: http://www.aes.org/e-lib/browse.cfm?elib=14401

[16] S. Bilbao, "Modeling of complex geometries and boundary conditions in finite difference/finite volume time domain room acoustics simulation," *IEEE Trans. on Audio, Speech and Language Processing*, vol. 21, no. 7, pp. 1524–1533, 2013.

[17] B. Donderici and F. L. Teixeira, "Improved FDTD subgridding algorithms via digital filtering and domain overriding," *IEEE Trans. on Antennas and Propagation*, vol. 53, no. 9, pp. 2938–2951, 2005.

[18] S. Bilbao, *Wave and Scattering Methods for the Numerical Integration of Partial Differential Equations*. New York: John Wiley & Sons., 2004.

[19] G. Strang, "The discrete cosine transform," *SIAM review*, vol. 41, no. 1, pp. 135–147, 1999.

[20] G. H. Golub and C. F. V. Loan, *Matrix Computations*. Baltimore: The John Hopkins University Press, 1989.

[21] F. Fontana, S. Zambon, and E. Bozzo, "Rate switching filters: Model and efficient approximation," *IEEE Trans. on Signal Processing*, vol. 62, no. 5, pp. 1290–1304, March 2014.

[22] G. Plonka and M. Tasche, "Fast and numerically stable algorithms for discrete cosine transform," *Linear Algebra and its Applications*, vol. 394, pp. 309–345, 2005.

[23] J. Strikwerda, *Finite Difference Schemes and Partial Differential Equations*. Pacific Grove, CA: Wadsworth & Brooks, 1989.

[24] A. Ben-Israel and T. N. E. Greville, *Generalized inverses*, 2nd ed., ser. CMS Books in Mathematics/Ouvrages de Mathématiques de la SMC, 15. New York: Springer-Verlag, 2003, theory and applications.

[25] M. Novello, F. Fontana, and E. Bozzo, "Decimation of finite-difference time-domain schemes in 1D and 2D boundary-absorbing acoustic model simulations," in *Proc. of the 2014 IEEE International Conference on Acoustic, Speech and Signal Processing*. Florence, Italy: IEEE, 2014, pp. 8227–8231.

PLACE
PHOTO
HERE

Federico Fontana Biography text here.

Enrico Bozzo Biography text here.

Marco Novello Biography text here.