INTERNATIONAL SYMPOSIUM on MUSICAL and ROOM ACOUSTICS
ISMRA 2016
September 11-13, 2016
La Plata, Buenos Aires, Argentina

RA-SM: Simulation and auralization of concert halls / opera houses: Paper 52

Finite difference room acoustics simulation with general impedance boundaries and viscothermal losses in air: Parallel implementation on multiple GPUs

Brian Hamilton[(a)], Craig J. Webb[(a)], Nathaniel D. Fletcher[(b)], Stefan Bilbao[(a)]

[(a)]Acoustics & Audio Group, University of Edinburgh, UK, first.lastname@ed.ac.uk
[(b)]Acoustics & Audio Group, University of Edinburgh, UK, s1534490@sms.ed.ac.uk

Abstract:

Room acoustics modelling requires numerical methods that can simulate the wave behaviour of sound across a wide band of frequencies while taking into account the frequency-dependent characteristics of absorption in air and at walls, but the accurate and stable numerical modelling of complex room geometries under frequency-dependent boundary conditions has remained an elusive problem. Recently, boundary conditions for finite difference/volume time-domain methods have been proposed to simulate frequency-dependent wall impedances in provably-stable numerical schemes based on the viscothermal wave equation over complex room geometries. The purpose of this paper is to investigate these new frequency-dependent boundary conditions in parallel implementations on graphics processing unit (GPU) devices. An efficient implementation of general impedance boundaries combined with the simplest Cartesian viscothermal scheme is presented and shown to be nearly as fast as simpler frequency-independent boundaries in acoustic simulations of a grid-aligned box domain with six frequency-dependent materials and of the Goldener Saal, Musikverein Vienna concert hall, running on up to four Nvidia K20 GPU devices.

Keywords: finite difference time domain (FDTD), room acoustics, computational acoustics, GPGPU

INTERNATIONAL SYMPOSIUM on MUSICAL and ROOM ACOUSTICS
ISMRA 2016
September 11-13, 2016
La Plata, Buenos Aires, Argentina

Finite difference room acoustics simulation with general impedance boundaries and viscothermal losses in air: Parallel implementation on multiple GPUs

1 Introduction

Room acoustics modelling requires numerical methods that can simulate the wave behaviour of sound across a wide band of frequencies while taking into account the frequency-dependent characteristics of absorption in air and at walls [1, 2]. Finite difference time-domain (FDTD) methods for room acoustics simulations have received a great deal of interest over the last twenty years [3–7], but the accurate and stable numerical modelling of complex room geometries under frequency-dependent boundary conditions has remained an elusive problem. Recently, boundary conditions for finite difference and finite volume time-domain methods have been proposed that model general locally-reacting frequency-dependent wall impedances as passive circuit networks, allowing for provably-stable numerical schemes for room acoustic modelling based on the viscothermal wave equation in complex room geometries [8–10]. While inherent numerical dispersion remains a concern [6, 11], this finite volume/difference approach is an important step towards a complete wave-based room acoustic modelling technique, as it is capable of handling realistic scenarios which are currently beyond the scope of other wave-based methods, such as pseudospectral time-domain (PSTD) [12] and adaptive rectangular decomposition (ARD) methods [13].

An important reason for the more recent popularity of FDTD methods is their relative ease of implementation in parallel on graphics processing unit (GPU) devices or multi-core CPUs, and the significant speed-ups possible over single-threaded CPU codes [7,14–18]. The implementation of frequency-dependent boundaries—which, up until recently, has been primarily based on the so-called “digital impedance filter” (DIF) boundaries proposed in [6]—has been a challenge for GPU implementation, with many studies reporting significant slow-downs for the use of frequency-dependent DIF boundaries in comparison to frequency-independent boundaries [7,16,19,20]. As a result, it has been put forth in some studies that a more efficient approach may be to simulate frequency-dependent wall absorption using frequency-*independent* boundary conditions operated over multiple octave bands, with responses recombined using octave-band filters [20–22] (as is typically done in geometric acoustic methods [21]). The supposed efficiency of such an approach lies in the fact that each octave-band simulation requires, in theory, 1/16th of the time required for the next higher octave band (if the grid spacing and time-step are changed accordingly at each octave-band), meaning that the combined simulation time should be approximately $1+1/16+1/16^2+\ldots\approx 1.067$ times that for the highest octave band (neglecting additional pre- and post-processing steps), which is not as significant of a slow-down as those previously reported for implementations of DIF boundaries. Whereas this theoretical increase of 6.7% is achievable with serial implementations [22], performance on a GPU device is not constant across grid sizes, so a multi-band approach on GPU is slightly more costly; e.g., 9.4% increases are reported in [20]. The same applies to ARD and PSTD methods, which are currently limited to multi-band approaches for approximations to frequency-dependent boundaries [12, 23].

While possibly efficient in comparison to DIF boundaries, the multi-band approach introduces many issues into the room acoustics model. For example, there can be geometrical inconsis-

tencies across octave bands due to the use of multiple grid resolutions, whereas if one uses a single grid/mesh/voxelization across all octave bands the multi-band approach is not at all efficient. But perhaps most importantly, the use of multi-band frequency-independent absorption neglects mass and reactance effects that necessarily underlie any frequency-dependent wall behaviour, and is thus non-physical. To avoid such issues, a frequency-dependent approach that is consistent with a physically valid model of room acoustics, such as the one presented in [10], should be pursued, wherein a single simulation can be used to capture the desired acoustic field response across a wide band of frequencies. Hence, the focus of this study is on the parallel implementation of the aforementioned general impedance boundary conditions on GPU devices, with the simple goal of finding a suitable implementation that has reasonable performance in comparison to a frequency-independent counterpart, circumventing the need for multi-band approaches.

An outline of the rest of the paper is as follows. The model equations and numerical schemes under consideration are presented in Section 2, and an efficient partitioning of the scheme into interior and boundary updates for GPU implementation is presented in Section 3. Section 4 presents simulations for performance testing, including simulations of a concert hall model, followed by conclusions and final remarks in Section 5.

2 Background

2.1 Viscothermal wave equation and general impedance boundary conditions

The partial differential equation of interest in this study is the 3-D viscothermal wave equation, also known as Stokes' wave equation [24]:

$$\partial_t^2 \Psi = c^2 (1 + \tau \partial_t) \Delta \Psi \tag{1}$$

where $\Psi = \Psi(\mathbf{x}, t)$ is the acoustic velocity potential in units m^2/s, $\mathbf{x} = (x, y, z)$ is positional vector in 3-D space, with $\mathbf{x} \in \Omega \subset \mathbb{R}^3$ where Ω is an enclosed space with boundary Γ, and $t \geq 0$ is time. ∂_t denotes a partial derivative with respect to time t, and Δ is the 3-D Laplacian operator. c is the speed of sound in air (e.g., 340 m/s–344 m/s), and τ is a relaxation time associated to classical sound attenuation in air with non-zero bulk viscosity [1, 25] ($\eta = c\tau$ is an associated length in metres, as employed in [11, 26]). Under typical indoor conditions, τ is on the order of 10^{-9} s–10^{-10} s and this equation provides a reasonable first-order approximation to audible sound wave propagation under more general relaxation effects [11, 25].

The sound pressure and vector velocity fields can be very nearly associated to Ψ via [10, 11]:

$$p = \rho \partial_t \Psi, \quad \mathbf{v} = -(1 + \tau \partial_t) \nabla \Psi \tag{2}$$

where ρ is the density of air (e.g., $1.2\,\mathrm{kg/m^3}$), and ∇ is the 3-D gradient operator. These associations are exact for the lossless case $\tau = 0$, and they are almost identical to those resulting from a linearized form of the Navier-Stokes system for $\tau > 0$ in typical indoor conditions for audible frequencies, see [10, 11].

General impedance boundary conditions, relating pressure p and outward normal velocity components $v_\perp = \mathbf{n} \cdot \mathbf{v}$ on the boundary surface with outward normal vector $\mathbf{n}$ at $\mathbf{x} \in \Gamma$, as

presented in [9], based on parallel branches of series RLC circuits, can be expressed as:

$$p = L^{(m)}\partial_t v_\perp^{(m)} + R^{(m)} v_\perp^{(m)} + \frac{1}{C^{(m)}} g^{(m)}, \quad \partial_t g^{(m)} = v_\perp^{(m)}, \quad m = 1, \ldots, M, \qquad v_\perp = \sum_{m=1}^{M} v_\perp^{(m)} \tag{3}$$

where $L^{(m)}, R^{(m)}, C^{(m)}$ are real-valued non-negative series inductances, resistances, and capacitances, respectively, for $m = 1, \ldots, M$, and M is the number of series RLC circuit branches used at $\mathbf{x} \in \Gamma$, and $v_\perp^{(m)}, m = 1, \ldots, M$ are analogues of electrical currents in the RLC branches. Each RLC branch has a frequency-dependent behaviour, and multiple branches may be combined to approximate general frequency-dependent wall absorptions; see, e.g., [9, Section V]. Branch coefficients and M can also vary over the boundary surface.

2.2 Finite difference scheme with viscothermal losses and general impedance boundaries

A provably-stable implicit/explicit finite volume scheme for the above PDE model problem, formulated on fully-unstructured grids, is presented in [10], as an extension of [9]. This study focusses on the practical implementation of the general impedance boundaries therein, but not on the additional use of fitted finite volume cells, so various simplifications are made to the scheme tested here, leading to a simple Cartesian finite difference scheme. Namely, we approximate the domain Ω with congruent cubic cells (staircased boundaries), use fully explicit time-integration, and associate one frequency-dependent wall material to each boundary node.

To serve as an approximation to the continuous field Ψ at discrete grid points in space and time, we introduce the Cartesian grid function $\Psi_{\boldsymbol{i}}^n \cong \Psi(\boldsymbol{i}X, nT_s)$ where $\boldsymbol{i} = (i_x, i_y, i_z) \in \mathbb{Z}^3$ and n is a non-negative integer, and where X is the Cartesian grid spacing and T_s is the time-step ($1/T_s$ is the sample rate of the scheme). Let Ω_X be the region Ω scaled by $1/X$, and let $q_{\boldsymbol{i}}$ be an indicator function taking on the value one when $\boldsymbol{i} \in \mathbb{Z}^3 \cap \Omega_X$ and zero otherwise. The number of nearest neighbours to each grid point inside the domain is represented by $K_{\boldsymbol{i}}$, defined as:

$$K_{\boldsymbol{i}} = K_{i_x,i_y,i_z} = q_{i_x+1,i_y,i_z} + q_{i_x,i_y+1,i_z} + q_{i_x,i_y,i_z+1} + q_{i_x-1,i_y,i_z} + q_{i_x,i_y-1,i_z} + q_{i_x,i_y,i_z-1} \tag{4}$$

and let $\bar{K}_{\boldsymbol{i}} = 6 - K_{\boldsymbol{i}}$. Also, we define $Q_{\boldsymbol{i}}^n$, which represents a weighted spatial average of $\Psi_{\boldsymbol{i}}^n$ over neighbouring points:

$$Q_{\boldsymbol{i}}^n = Q_{i_x,i_y,i_z}^n = \Psi_{i_x+1,i_y,i_z}^n + \Psi_{i_x,i_y+1,i_z}^n + \Psi_{i_x,i_y,i_z+1}^n + \Psi_{i_x-1,i_y,i_z}^n + \Psi_{i_x,i_y-1,i_z}^n + \Psi_{i_x,i_y,i_z-1}^n \tag{5}$$

and similarly, we define $\tilde{Q}_{\boldsymbol{i}}^n$ as follows:

$$\tilde{Q}_{\boldsymbol{i}}^n = \tilde{Q}_{i_x,i_y,i_z}^n = \tilde{\Psi}_{i_x+1,i_y,i_z}^n + \tilde{\Psi}_{i_x,i_y+1,i_z}^n + \tilde{\Psi}_{i_x,i_y,i_z+1}^n + \tilde{\Psi}_{i_x-1,i_y,i_z}^n + \tilde{\Psi}_{i_x,i_y-1,i_z}^n + \tilde{\Psi}_{i_x,i_y,i_z-1}^n \tag{6}$$

where $\tilde{\Psi}_{i_x,i_y,i_z}^n = q_{i_x,i_y,i_z} \Psi_{i_x,i_y,i_z}^n$. Finally, define the sets $\mathcal{I} = \{\boldsymbol{i} \in \mathbb{Z}^3 : q_{\boldsymbol{i}} = 1\}$ (all points inside the domain) and let $K_{\boldsymbol{i}} = 0$ for $\boldsymbol{i} \notin \mathcal{I}$, $\mathcal{I}_\mathrm{i} = \{\boldsymbol{i} \in \mathbb{Z}^3 : K_{\boldsymbol{i}} = 6\}$ (interior points with all neighbours inside), $\mathcal{I}_\mathrm{b} = \{\boldsymbol{i} \in \mathbb{Z}^3 : 0 < K_{\boldsymbol{i}} < 6\}$ (interior points with some neighbours outside, requiring discrete boundary conditions, i.e., boundary nodes). The number of elements in a discrete set Υ is denoted $|\Upsilon|$.

Under the aforementioned simplifications, the finite volume scheme in [10] reduces to the following explicit finite difference scheme, which can be split into two separate updates for

regular interior points ($\boldsymbol{i} \in \mathcal{I}_\mathrm{i}$) and boundary points ($\boldsymbol{i} \in \mathcal{I}_\mathrm{b}$). For regular interior points, the update is:

$$\Psi_{\boldsymbol{i}}^{n+1} = \left(2 - 6\lambda^2\left(1+\tau'\right)\right)\Psi_{\boldsymbol{i}}^{n} + \left(6\lambda^2\tau' - 1\right)\Psi_{\boldsymbol{i}}^{n-1} + \lambda^2(1+\tau')Q_{\boldsymbol{i}}^{n} - \lambda^2\tau' Q_{\boldsymbol{i}}^{n-1}, \qquad \boldsymbol{i} \in \mathcal{I}_\mathrm{i} \tag{7}$$

where $\lambda = cT_s/X$ is the Courant number and $\tau' = \tau/T_s$. Note that for $\tau' = 0$, this reduces to the standard Cartesian scheme for the lossless wave equation [27].

For boundary points ($\boldsymbol{i} \in \mathcal{I}_\mathrm{b}$), the explicit update is carried out in three steps:

$$\Psi_{\boldsymbol{i}}^{n+1} = \frac{1}{1 + \bar{K}_{\boldsymbol{i}}\lambda\beta_{\boldsymbol{i}}/2}\Bigg[\left(2 - \lambda^2 K_{\boldsymbol{i}}\left(1+\tau'\right)\right)\Psi_{\boldsymbol{i}}^{n} + \left(\bar{K}_{\boldsymbol{i}}\lambda\beta_{\boldsymbol{i}}/2 + \lambda^2 K_{\boldsymbol{i}}\tau' - 1\right)\Psi_{\boldsymbol{i}}^{n-1} + \lambda^2(1+\tau')\tilde{Q}_{\boldsymbol{i}}^{n} - \lambda^2\tau'\tilde{Q}_{\boldsymbol{i}}^{n-1}$$
$$- \lambda\bar{K}_{\boldsymbol{i}}\sum_{m=1}^{M_{\boldsymbol{i}}} b_{\boldsymbol{i}}^{(m)}\left(2\hat{D}_{\boldsymbol{i}}^{(m)}\hat{v}_{\perp,\boldsymbol{i}}^{(m),n-1/2} - \hat{F}_{\boldsymbol{i}}^{(m)}\hat{g}_{\boldsymbol{i}}^{(m),n-1/2}\right)\Bigg] \tag{8a}$$

$$\hat{v}_{\perp,\boldsymbol{i}}^{(m),n+1/2} = b_{\boldsymbol{i}}^{(m)}\left((\Psi_{\boldsymbol{i}}^{n+1} - \Psi_{\boldsymbol{i}}^{n-1}) + d_{\boldsymbol{i}}^{(m)}\hat{v}_{\perp,\boldsymbol{i}}^{(m),n-1/2} - 2\hat{F}_{\boldsymbol{i}}^{(m)}\hat{g}_{\boldsymbol{i}}^{(m),n-1/2}\right) \tag{8b}$$

$$\hat{g}_{\boldsymbol{i}}^{(m),n+1/2} = \hat{g}_{\boldsymbol{i}}^{(m),n-1/2} + 0.5\left(\hat{v}_{\perp,\boldsymbol{i}}^{(m),n+1/2} + \hat{v}_{\perp,\boldsymbol{i}}^{(m),n-1/2}\right) \tag{8c}$$

where $M_{\boldsymbol{i}}$ is the number of circuit branches employed at grid point $\boldsymbol{i} \in \mathcal{I}_\mathrm{b}$, and $\hat{g}_{\boldsymbol{i}}^{(m),n\pm1/2}$ and $\hat{v}_{\perp,\boldsymbol{i}}^{(m),n\pm1/2}$, $m = 1,\ldots,M_{\boldsymbol{i}}$ are additional variables required at boundary nodes, which have units of m^2/s and can respectively be seen as approximations to analogous continuous variables $cT_s v_\perp^{(m)}$ and $cg^{(m)}$ at staggered times $t = (n \pm 1/2)T_s$ and near $\mathbf{x} = \boldsymbol{i}h$. Also,

$$\beta_{\boldsymbol{i}} = \sum_{m=1}^{M_{\boldsymbol{i}}} b_{\boldsymbol{i}}^{(m)}, \quad b_{\boldsymbol{i}}^{(m)} = \left(2\hat{D}_{\boldsymbol{i}}^{(m)} + \hat{E}_{\boldsymbol{i}}^{(m)} + 0.5\hat{F}_{\boldsymbol{i}}^{(m)}\right)^{-1}, \quad d_{\boldsymbol{i}}^{(m)} = \left(2\hat{D}_{\boldsymbol{i}}^{(m)} - \hat{E}_{\boldsymbol{i}}^{(m)} - 0.5\hat{F}_{\boldsymbol{i}}^{(m)}\right), \quad m = 1,\ldots,M_{\boldsymbol{i}} \tag{9}$$

where $\hat{D}_{\boldsymbol{i}}^{(m)}$, $\hat{E}_{\boldsymbol{i}}^{(m)}$, $\hat{F}_{\boldsymbol{i}}^{(m)}$ are respectively $Y_0 L^{(m)}/T_s$, $Y_0 R^{(m)}$, and $Y_0 T_s/C^{(m)}$ for the materials associated to boundary points $\boldsymbol{i} \in \mathcal{I}_\mathrm{b}$, with $Y_0 = (\rho c)^{-1}$.

The boundary update simplifies to a frequency-independent impedance condition under the constraints $M_{\boldsymbol{i}} = 1$, $\hat{D}_{\boldsymbol{i}}^{(1)} = \hat{F}_{\boldsymbol{i}}^{(1)} = 0$, and where $\hat{E}_{\boldsymbol{i}}^{(1)} \geq 0$ would represent a real-valued characteristic wall impedance (and $\beta_{\boldsymbol{i}}$ a characteristic admittance, as in [26]). The frequency-independent case does not require (8b) and (8c), nor does it require additional storage at boundaries.

Using energy techniques, it can be proven that the entire scheme is stable under the following condition on the time-step [10]:

$$T_s \leq \sqrt{c^{-2}X^2/3 + \tau^2} - \tau \tag{10}$$

Note that the implementation of these general impedance boundary conditions does not impose additional constraints on the scheme since this is also the stability condition obtained through von Neumann analysis for the free-space viscothermal scheme (i.e., (7) when $\mathcal{I}_\mathrm{i} = \mathbb{Z}^3$) [11,26]. It can also be shown that numerical stored energy including accumulated losses is conserved to machine precision, see [10].

Practical operation of the scheme requires three full states $\Psi_{\boldsymbol{i}}^{n+1}, \Psi_{\boldsymbol{i}}^{n}, \Psi_{\boldsymbol{i}}^{n-1}$ (for $\boldsymbol{i} \in \mathcal{I}$) to be stored in memory, and $3M_{\boldsymbol{i}}$ additional states at boundary nodes ((8c) can overwrite in place). It is assumed that the number of wall materials and branches per material is small enough that the

associated coefficients ($b_{\boldsymbol{i}}^{(m)}, d_{\boldsymbol{i}}^{(m)}, \hat{D}_{\boldsymbol{i}}^{(m)}, \hat{F}_{\boldsymbol{i}}^{(m)}$, $m = 1, \ldots, M_{\boldsymbol{i}}$, and $\beta_{\boldsymbol{i}}$) can be stored in a negligible space of memory (e.g., constant memory on GPUs). Also, the storage of $K_{\boldsymbol{i}}$, which can be obtained in a "voxelization" pre-processing step (see, e.g., [17, 28]), can be a small portion of total memory used (e.g., one byte per $K_{\boldsymbol{i}}$) since $0 \leq K_{\boldsymbol{i}} \leq 6$.

A discrete pressure field at times $t = (n + 1/2)T_s$ can be recovered with $p_{\boldsymbol{i}}^{n+1/2} = \rho(\Psi^{n+1} - \Psi^n)/T_s$, or the entire scheme can be rewritten in terms of pressure simply by replacing $\Psi_{\boldsymbol{i}}^n$ with a grid function $p_{\boldsymbol{i}}^n$, in which case the normalised boundary variables $\hat{g}_{\boldsymbol{i}}^{(m),n\pm1/2}$ and $\hat{v}_{\perp,\boldsymbol{i}}^{(m),n\pm1/2}$ take on alternative interpretations (one time derivative higher and with units of pressure).

3 An efficient splitting algorithm for GPU implementations

In this section, we present an efficient partitioning of the scheme that is suitable for parallel implementation on GPU devices. Assume $\Psi_{\boldsymbol{i}}^n$ and $\Psi_{\boldsymbol{i}}^{n-1}$ are initialised according to initial conditions for the PDE problem, and $\Psi_{\boldsymbol{i}}^n = \Psi_{\boldsymbol{i}}^{n-1} = 0$ for $\boldsymbol{i} \notin \mathcal{I}$. Also, we assume that global memory is allocated to each state over the smallest bounding box of points $\mathcal{B} \subset \mathbb{Z}^3$ for which $\mathcal{I} \subset \mathcal{B}$, yet still comprising a one-layer thick halo of zero-valued ghost points surrounding $\mathcal{I}$.

The first step of the partitioned scheme is then to calculate what corresponds to a scheme with rigid boundary terminations:

$$\Psi_{\boldsymbol{i}}^{n+1} := \left(2 - K_{\boldsymbol{i}}\lambda^2(1+\tau')\right)\Psi_{\boldsymbol{i}}^n + \left(K_{\boldsymbol{i}}\lambda^2\tau' - 1\right)\Psi_{\boldsymbol{i}}^{n-1} + \lambda^2\left(Q_{\boldsymbol{i}}^n - \tau' Q_{\boldsymbol{i}}^{n-1}\right), \quad \boldsymbol{i} \in \mathcal{I} \tag{11}$$

where here ":=" is an assignment operator (e.g., "=" in the C programming language [29], and not to be confused with "defined as"). The second step applies a correction to $\Psi_{\boldsymbol{i}}^{n+1}$ over boundary nodes for frequency-dependent (or frequency-independent) wall absorption:

$$\Psi_{\boldsymbol{i}}^{n+1} := \frac{1}{1 + \bar{K}_{\boldsymbol{i}}\lambda\beta_{\boldsymbol{i}}/2}\left[\Psi_{\boldsymbol{i}}^{n+1} + 0.5\bar{K}_{\boldsymbol{i}}\lambda\beta_{\boldsymbol{i}}\Psi_{\boldsymbol{i}}^{n-1} - \lambda\bar{K}_{\boldsymbol{i}}\sum_{m=1}^{M_{\boldsymbol{i}}} b_{\boldsymbol{i}}^{(m)}\left(2\hat{D}_{\boldsymbol{i}}^{(m)}\hat{v}_{\perp,\boldsymbol{i}}^{(m),n-1/2} - \hat{F}_{\boldsymbol{i}}^{(m)}\hat{g}_{\boldsymbol{i}}^{(m),n-1/2}\right)\right], \quad \boldsymbol{i} \in \mathcal{I}_{\mathrm{b}} \tag{12}$$

followed by the updates (8b) and (8c) for $\boldsymbol{i} \in \mathcal{I}_{\mathrm{b}}$ in the frequency-dependent case. This splitting of the scheme into two steps leads to a straightforward design of two CUDA kernels (provided at [30]). At least in the frequency-dependent case, partitioning the scheme into two kernels is preferable to having one kernel that implements both (7) and (8) using conditional statements, as this would lead to branch divergence and would require additional lookup tables in order to fetch values of $\hat{v}_{\perp,\boldsymbol{i}}^{(m),n-1/2}$ and $\hat{g}_{\boldsymbol{i}}^{(m),n-1/2}$ for $\boldsymbol{i} \in \mathcal{I}_{\mathrm{b}}$, assumed to be stored contiguously in a block of global memory of size $3|\mathcal{I}_{\mathrm{b}}| \times \max_{\boldsymbol{i}\in\mathcal{I}_{\mathrm{b}}} M_{\boldsymbol{i}}$ elements. On the other hand, a kernel that operates over a list of boundary nodes will require some non-contiguous memory reading for values of $\Psi_{\boldsymbol{i}}^{n+1}$ and $\Psi_{\boldsymbol{i}}^{n-1}$, but this is not a major concern for performance when $|\mathcal{I}_{\mathrm{b}}|/|\mathcal{I}| \ll 1$, as will be seen in the next section.

In order to implement this scheme across multiple GPU cards, state memory and additional boundary data and variables must be split appropriately for the number of GPU cards used. $\Psi_{\boldsymbol{i}}^{n+1}$ state variables require overlapping halos and data transfers along GPU-card partition interfaces. This can be accomplished using techniques described in [16–18, 31]. For large grids, the amount of data that needs to be transferred is small; e.g., for four GPUs cards, less than 0.5% of the total state is transferred between cards at each time-step. Boundary variables and

data do not require data transfers between cards since (12) is a locally-reacting update.

4 Performance testing

In this section we present three test cases that measure the performance of the proposed splitting algorithm and quantify the computation time required for general impedance boundaries of up to $M \leq 9$ branches. The following tests are carried out using single and double floating-point precision on Nvidia Tesla K20 GPU cards (up to four), each having approximately 5 GB of global memory. In the single-card case, 3-D thread blocks of size 32×2×2 are used for the CUDA kernel that carries out (11) (with enough threads issued to cover $\mathcal{B}$), and thread blocks of size 128×1 are used for the kernel that carries out (11) (enough threads issued to cover $\mathcal{I}_\mathrm{b}$).[1] These kernels are provided at [30], and multi-card CUDA kernels are straightforward extensions of those single-card kernels, for which the number of thread blocks is adapted accordingly to the number of GPU devices used. Also, for the following tests $c = 340\,\mathrm{m/s}$ and $c\tau = 2\mathrm{e}\text{-}6\,\mathrm{m}$, corresponding, approximately, to indoor conditions at 15°C and 40% relative humidity [11, 33].

The modelling of the frequency-dependent materials with complex impedances is beyond the scope of this paper, but for the purposes of this study frequency-dependent absorptions are modelled by fitting general impedance boundary coefficients to frequency-independent octave-band absorption coefficients (obtained from, e.g., ODEON [34]), using simplex methods in order to minimise a cost function based on the reflection magnitude error (on a logarithmic frequency scale). An example fit is shown in Fig. 1. For approaches to more detailed modelling of frequency-dependent materials starting from transfer matrix models, see [9, Section V].

4.1 Test case 1: Box domain with six wall materials – single GPU card

In the first test case we consider a grid-aligned box domain of size 10m×6m×3m, excited with an impulsive source and using different impedances for each wall, having up to nine branches in the frequency-dependent case. For double precision testing we choose $X = 10\,\mathrm{mm}$, resulting in a grid of size 1002×602×302 (approx. 182 million) grid points, and for single precision we choose $X = 8\,\mathrm{mm}$, resulting in a grid of size 1252×752×377 (approx. 355 million) grid points. In both cases, the time-step T_s is set according to the stability limit (10) (sample rates of 58.9 kHz and 73.6 kHz, respectively). Also in both cases, $|\mathcal{I}_\mathrm{b}|/|\mathcal{I}| \approx 0.01$. Simulations are carried out for frequency-independent absorptions with different reflection coefficients for each wall, and for frequency-dependent materials with $M = 1, \ldots, 9$ branches, using kernels provided at [30]. In each case, 500 time-steps are carried and run-times are recorded using CUDA event timers [32].

For the frequency-independent cases, throughputs are 6300 Megavoxels/s (Mvox/s) and 3892 Mvox/s in single- and double-precision, respectively, for this test case, which give indications that the proposed splitting approach achieves good performance, since the reported throughputs are comparable to throughputs for similar test cases reported in [11, 18, 26] using a single kernel to update the entire scheme. In comparison to the case of a rigid box, relative computation time increases (RCTI) are 2.0% and 1.6% for frequency-independent cases using single and double precision, respectively.

As for the case of frequency-dependent boundaries, RCTIs for $1 \leq M \leq 9$ branches are plotted in

[1] For an overview of the CUDA programming framework, see [32], and for an overview of CUDA programming applied to finite-difference room acoustics models, see [18].

Fig. 2 using the computation times of the frequency-independent cases as references. It can be seen that, compared to the frequency-independent case, the updating of boundary nodes—here only 1% of the total number of nodes—results in approximately 1% RCTI per branch used in double-precision, and RCTIs are smaller in single-precision. At nine branches, the RCTI is in fact 6.6% for single-precision, which is faster than theoretically possible for a multi-band approach using the kernels tested here.

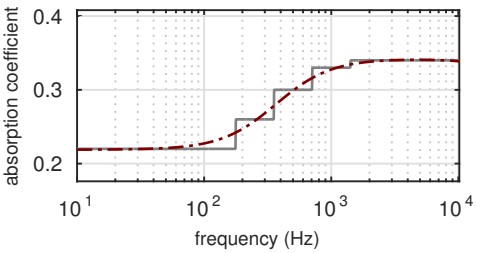


Figure 1: Absorption coefficients obtained from ODEON [34] for chair surfaces for concert hall model used in Section 4.2 (gray line), and a fitted general impedance boundary model using $M = 6$ branches (dotted line).

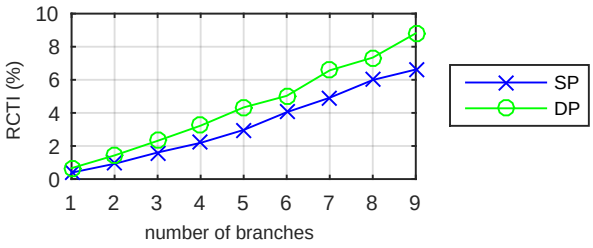
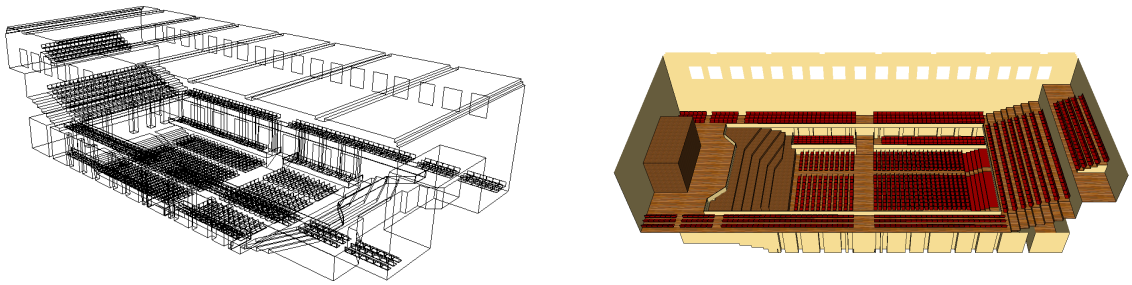


Figure 2: Relative computation time increases (RCTI) for frequency-dependent boundaries with M branches compared to calculation of box with frequency-independent walls, in single-precision (SP) and double-precision (DP).

4.2 Test case 2: Concert hall model – multiple GPU cards

For the second test case, we consider a simplified concert hall simulation run on up to four Nvidia K20 GPU devices. In particular, we consider the *Goldener Saal*, Musikverein Vienna, a shoebox-style concert hall with highly-regarded acoustics [35]. A 3-D surface model is constructed—based on one freely available with ODEON [34, 36] (modified to be watertight, and such that chairs and balconies have volume)—as shown in Fig. 3. The surface mesh has a 15,000 m^3 bounding box, an interior volume of 12,425 m^3, and surfaces totalling 6467 m^2, approximated with 12,550 polygonal faces (32,398 triangles). Surfaces are associated to five materials: plasterboard, glass (windows), wood panelling, and materials for floors and chairs. Absorption coefficients for these materials are taken from ODEON [34]; see Fig. 1.



(a) wireframe view

(b) exposed view with surfaces coloured

Figure 3: Simplified computer model of the Goldener Saal, Musikverein Vienna concert hall.

For this test case we are interested in the RCTI of the frequency-dependent case in comparison

to a frequency-independent case (using a single absorption coefficient for each material). Also, the maximum throughput possible across one to four cards is of interest. To these ends, the concert hall is voxelized at different grid resolutions, in order to use as much memory as possible over one to four cards, with up to six branches per material in single precision. The synchronous halo swapping technique, as described in, e.g., [18, 31] (also used in [17]), is employed for data transfers across cards. Simulations are run for five seconds of output, using an impulsive source located at the stage of the hall, and taking an output next to one chair. Performance results are presented in Table 1. As expected, the use of multiple GPU cards permits higher throughputs, although multi-card throughputs are less than the single-card throughput multiplied by number of GPU cards used. This is simply because of latencies introduced by non-asynchronous halo-swapping. On the other hand, RCTIs compared to same-sized simulations under frequency-independent boundaries are again reasonable, and are, in fact, somewhat improved over the single card simulation, most likely due to halo-swapping being the dominating cause of latencies. Sound examples from these simulations are available at [30].

Table 1: Performance results for multi-card simulations of concert hall. Listed are: grid spacings (X), sample rates ($1/T_s$), total memory used, runtimes per second of output, throughputs in Mvox/s, ratios of points in bounding box to points interior to concert hall ($|\mathcal{I}|/|\mathcal{B}|$), ratios of boundary points to interior points ($|\mathcal{I}_\mathrm{b}|/|\mathcal{I}|$), and relative computation time increases (RCTI) over same-size simulations under frequency-independent boundaries.

# GPUs	grid spacing	sample rate	memory used	runtime/second-output	throughput	$\lvert\mathcal{I}\rvert/\lvert\mathcal{B}\rvert$	$\lvert\mathcal{I}_\mathrm{b}\rvert/\lvert\mathcal{I}\rvert$	RCTI
1	35.0 mm	16.83 kHz	5.2 GB	15 min 27 s	5239 Mvox/s	0.80	0.018	5.89%
2	27.9 mm	21.11 kHz	10.1 GB	20 min 41 s	9691 Mvox/s	0.80	0.014	2.70%
3	24.4 mm	24.14 kHz	15.0 GB	30 min 42 s	11161 Mvox/s	0.80	0.013	2.58%
4	22.2 mm	26.53 kHz	19.8 GB	35 min 27 s	14108 Mvox/s	0.80	0.011	2.98%

5 Conclusions and Final Remarks

In this study, we presented an efficient parallel implementation on Nvidia K20 GPUs of the simplest Cartesian finite difference scheme for the viscothermal wave equation, with general impedance boundary conditions modelling frequency-dependent wall absorption. It was shown that frequency-dependent boundaries could be incorporated into the scheme under reasonable compute-time increases over simpler frequency-independent counterparts, and sufficient to circumvent the need for multi-band approaches. A simulation of the acoustics of the Goldener Saal, Musikverein Vienna concert hall was performed over four GPU cards, with a throughput of 14,108 megavoxels/s. Improvements to this work could include: the use of asynchronous halo-swapping, which masks memory transfers thereby speeding-up the multi-card simulations presented here [31]; the use of face-centered cubic grids for reduced numerical dispersion [37]; and fitted cells to adapt to non-Cartesian boundary surfaces [9, 10], which would necessarily require additional data at boundaries and further investigations for efficient implementation on GPUs.

Acknowledgements This work was supported by the European Research Council under grant StG-2011-279068-NESS, and by the Natural Sciences and Engineering Research Council of Canada.

References

[1] P. M. Morse and K. U. Ingard, *Theoretical acoustics*. Princeton University Press, 1968.
[2] H. Kuttruff, *Room acoustics*. CRC Press, 2009.
[3] D. Botteldooren, "Finite-difference time-domain simulation of low-frequency room acoustic problems," *JASA*, vol. 98, pp. 3302–3308, 1995.
[4] L. Savioja, T. J. Rinne, and T. Takala, "Simulation of room acoustics with a 3-D finite difference mesh," in *Proc. Int. Computer Music Conf. (ICMC)*, (Danish Institute of Electroacoustic Music, Denmark), pp. 463–466, 1994.
[5] G. R. Campos and D. M. Howard, "On the computational efficiency of different waveguide mesh topologies for room acoustic simulation," *IEEE Trans. Speech and Audio Processing*, vol. 13, no. 5, pp. 1063–1072, 2005.
[6] K. Kowalczyk and M. van Walstijn, "Room acoustics simulation using 3-D compact explicit FDTD schemes," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 19, no. 1, pp. 34–46, 2011.
[7] C. Spa, A. Rey, and E. Hernandez, "A GPU implementation of an explicit compact FDTD algorithm with a digital impedance filter for room acoustics applications," *IEEE/ACM Trans. Audio, Speech, and Language Processing*, vol. 23, no. 8, pp. 1368–1380, 2015.
[8] S. Bilbao, "Modeling of complex geometries and boundary conditions in finite difference/finite volume time domain room acoustics simulation," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 21, pp. 1524–1533, July 2013.
[9] S. Bilbao, B. Hamilton, J. Botts, and L. Savioja, "Finite volume time domain room acoustics simulation under general impedance boundary conditions," *IEEE/ACM Trans. Audio, Speech, and Language Processing*, vol. 24, no. 1, pp. 161–173, 2016.
[10] S. Bilbao and B. Hamilton, "Wave-based room acoustics simulation: Explicit/implicit finite volume modeling of viscothermal losses and frequency-dependent boundaries," *J. Audio Engineering Society*, 2016. (to appear).
[11] B. Hamilton, S. Bilbao, and C. J. Webb, "Improved finite difference schemes for a 3-D viscothermal wave equation on a GPU," in *Proceedings of Forum Acusticum*, (Krakow, Poland), 2014.
[12] M. Hornikx, T. Krijnen, and L. van Harten, "openPSTD: The open source pseudospectral time-domain method for acoustic propagation," *Computer Physics Communications*, vol. 203, pp. 298–308, 2016.
[13] N. Raghuvanshi, R. Narain, and M. C. Lin, "Efficient and accurate sound propagation using adaptive rectangular decomposition," *IEEE Trans. Vis. and CG*, vol. 15, no. 5, pp. 789–801, 2009.
[14] A. Southern, D. Murphy, G. Campos, and P. Dias, "Finite difference room acoustic modelling on a general purpose graphics processing unit," in *Proc. of the 128th AES Convention*, (London, UK), 2010.
[15] J. Sheaffer and B. Fazenda, "FDTD/K-DWM simulation of 3D room acoustics on general purpose graphics hardware using compute unified device architecture (CUDA)," *Proc. Institute of Acoustics*, vol. 32, no. 5, 2010.
[16] N. Borrel-Jensen, "Real-time auralisation of the lower frequency sound field using numerical methods on the GPU," M.Sc. thesis, RWTH Aachen University and University of Copenhagen, 2012.
[17] J. Saarelma and L. Savioja, "An open source finite difference time-domain solver for room acoustics using graphics processing units," in *Proceedings of Forum Acusticum*, (Krakow, Poland), 2014.
[18] C. J. Webb, *Parallel computation techniques for virtual acoustics and physical modelling synthesis*. Ph.D. thesis, University of Edinburgh, 2014.
[19] L. Savioja, "Real-time 3D finite-difference time-domain simulation of low-and mid-frequency room acoustics," in *Proc. Digital Audio Effects (DAFx)*, vol. 1, (Graz, Austria), p. 75, 2010.
[20] J. Sheaffer, B. M. Fazenda, D. T. Murphy, and J. A. S. Angus, "A simple multiband approach for solving frequency dependent problems in numerical time domain methods," in *Proceedings of Forum Acusticum*, pp. 269–274, 2011.
[21] L. Savioja and U. P. Svensson, "Overview of geometrical room acoustic modeling techniques," *JASA*, vol. 138, no. 2, pp. 708–730, 2015.
[22] S. Oxnard, D. O'Brien, J. van Mourik, and D. Murphy, "Frequency-dependent absorbing boundary implementations in 3D finite difference time domain room acoustics simulations," in *Proc. Euronoise*, (Maastricht, Netherlands), 2015.
[23] N. Morales, R. Mehra, and D. Manocha, "A parallel time-domain wave simulator based on rectangular decomposition for distributed memory architectures," *Applied Acoustics*, vol. 97, pp. 104–114, 2015.
[24] M. J. Buckingham, "Causality, Stokes' wave equation, and acoustic pulse propagation in a viscous fluid," *Physical Review E*, vol. 72, no. 2, p. 026610, 2005.
[25] A. D. Pierce, *Acoustics: an introduction to its physical principles and applications*. McGraw-Hill New York, 1989.
[26] C. J. Webb and S. Bilbao, "Computing room acoustics with CUDA - 3D FDTD schemes with boundary losses and viscosity," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, (Prague, Czech Republic), pp. 317–320, 2011.
[27] G. E. Forsythe and W. R. Wasow, *Finite-difference methods for partial differential equations*, pp. 378–382. New York: Wiley, 1960.
[28] H. Karlsson, "Solid voxelization algorithm for acoustic modeling," Master's thesis, Aalto University, Espoo, Finland, 2014.
[29] B. W. Kernighan and D. M. Ritchie, *The C Programming Language*. Prentice-Hall, 1988.
[30] "Accompanying website." http://www2.ph.ed.ac.uk/~s1164563/ISMRA2016. Accessed: 2016-07-25.
[31] C. J. Webb and A. Gray, "Large-scale virtual acoustics simulation at audio rates using three dimensional finite difference time domain and multiple GPUs," in *Proc. Int. Cong. Acoustics (ICA)*, (Montréal, Canada), 2013.
[32] NVIDIA Corporation, *CUDA C Programming Guide*. PG-02829-001_v7.0, Mar. 2015.
[33] "Acoustics – attenuation of sound during propagation outdoors. Part 1: Calculation of the absorption of sound by the atmosphere," Standard ISO 9613–1: 1993, International Organization for Standardization, Geneva, Switzerland, 1993.
[34] G. M. Naylor, "Odeon—another hybrid room acoustical model," *Applied Acoustics*, vol. 38, no. 2, pp. 131–143, 1993.
[35] L. Beranek, *Concert halls and opera houses: music, acoustics, and architecture*. Springer Science & Business Media, 2012.
[36] H. Shiokawa and J. H. Rindel, "Comparisons between computer simulations of room acoustical parameters and those measured in concert halls," Tech. Rep. 89,2007, Research Institute of Industrial Technology, Nihon University, 2007.
[37] B. Hamilton and C. J. Webb, "Room acoustics modelling using GPU-accelerated finite difference and finite volume methods on a face-centered cubic grid," in *Proc. Digital Audio Effects (DAFx)*, (Maynooth, Ireland), pp. 336–343, Sept. 2013.