AN ABSTRACT OF THE THESIS OF

Jack H. Myers for the degree of Master of Science in Computer Science presented on August 1, 2024.

Title: GPU-Accelerated Nonlocal Meshless Lattice Boltzmann Method for 3D Acoustic Propagation

Abstract approved: ____

Dr. Eugene Zhang

Sound propagation is an integral part of realistic scene design within Computer Graphics applications and research. Challenging problems like reflection, refraction, and diffraction require customized and complex numerical models. The lattice Boltzmann method (LBM) utilizes probability distributions of microscopic velocities to calculate macroscopic fluid quantities and is an emerging method for acoustic modeling. Inspired by recent nonlocal methods for acoustics problems in Physics domains, we adopt a nonlocal vector calculus formulation of the divergence operator, deriving a novel discrete solution to the lattice Boltzmann streaming step. Our novel sound propagation framework utilizes a meshless discretization of the spatial dimension to accommodate realistic 3D domains. This formulation enables a parallel CUDA implementation for efficient computation even with the small time steps required for acoustic simulations. We apply our framework to real-world in-

door and outdoor environments containing multiple sound sources and observers that are both static and moving. This capability is found in few existing works. We also show the natural ability of our framework for capturing stereoscopic audio using a pair of offset observers. Moreover, we demonstrate convergence measurements for experiments with known solutions and demonstrate complex acoustic effects by recreating the double-slit and Doppler shift experiments. To our knowledge, our framework is the first to simulate sound propagation in 3D for real-world scenes using a nonlocal LBM formulation.

GPU-Accelerated Nonlocal Meshless Lattice Boltzmann Method for 3D Acoustic Propagation

by

Jack H. Myers

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Presented August 1, 2024
Commencement June 2024

Master of Science thesis of Jack H. Myers presented on August 1, 2024.

APPROVED:

Major Professor, representing Computer Science

Director of the School of Electrical Engineering and Computer Science

Dean of the Graduate School

Jack H. Myers, Author

ACKNOWLEDGEMENTS

TABLE OF CONTENTS

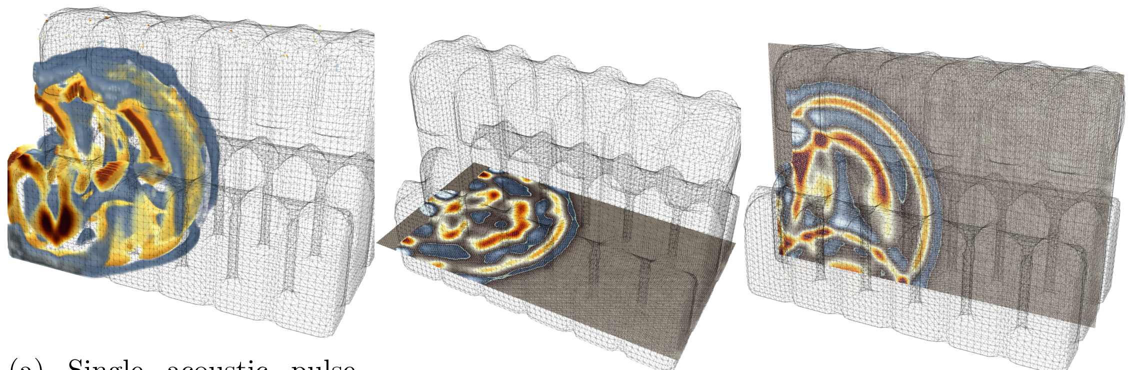
LIST OF FIGURES

LIST OF FIGURES (Continued)

LIST OF FIGURES (Continued)

LIST OF TABLES

Chapter 1: Introduction



(a) Single acoustic pulse, rendered volumetrically. (b) Horizontal section view. (c) Vertical section view

Figure 1.1: We demonstrate the application of our 3D sound propagation framework for a single acoustic pulse propagated through an indoor cathedral scene. The colored spherical shells in (a) indicate the 3D peaks and troughs of the sound waves. We observe that the sound waves propagate through the domain and fill the volume. In a horizontal cross section (b), we observe a diffraction effect around the columns and interference patterns from waves reflected off the columns. In a vertical cross section (c), we note that reflections from the floor and wall boundaries are especially apparent.

Realistic sound propagation plays an important role in complementing the visual components of entertainment and scientific products. The problem of sound propagation involves calculating sounds heard by an observer after the sound waves emitted from one or more sources undergo reflection, refraction, and diffraction within a virtual scene. Accurately modeling these phenomena can greatly enhance the realism and believability of a virtual environment.

Sound is a pressure wave resulting from compressions and expansions of vibrating particles in a medium. Computational fluid dynamics (CFD) provides physically accurate methods for simulating particle motion in fluid mediums such as air or water. Solving the Navier-Stokes equations is a standard approach to capture interactions between pressure and velocity in different mediums, allowing for direct simulation of wave propagation. However, standard numerical solution methods, such as finite difference, finite volume, and finite element, are often challenging to implement efficiently.

The Lattice Boltzmann method (LBM) takes a statistical approach to CFD where the fluid medium is represented using probability distributions. Broadly, the LBM solves the lattice Boltzmann equation (LBE), which describes the evolution of the velocity distribution function for populations of microscopic particles on a lattice grid in such a way that macroscopic fluid dynamics can be recovered. Originally developed as an extension of lattice gas automata [45], the LBE can be interpreted as a finite difference method for simulating the Boltzmann transport equation, which represents the probabilistic advection and collision processes of these particles. Despite their difference in origin, the Navier-Stokes equations can be derived from the LBE through a Chapman-Enskog expansion procedure [17]. Compared to direct Navier-Stokes solution methods, the LBM leads to a simpler implementation due to the linearity of its formulation, easier parallelization due to its local nature, and better representation of microscopic interactions, allowing for modeling of multi-physics dynamical systems. These advantages make the LBM particularly attractive for acoustic simulation.

Lattice Boltzmann methods have been applied with good results to various acoustics applications including wave propagation in both thermoacoustic and aeroacoustic settings, sound absorption in porous media, wind-based musical instruments, and outdoor acoustics [11]. A majority of this work, however, is focused on solving physics and engineering problems on small 2D domains with relatively simple geometry. Several works focus on sound wave propagation in 3D cavities with and without obstacles [6, 5].

For computer graphics applications, most existing work follows either wave-based or ray-based approaches. Wave-based approaches use numerical methods to solve the wave equation, capturing a wide range of acoustic effects, but they are computationally costly and relatively inflexible for dynamic scenes. More efficient ray-based approaches, adapted from geometric optics to model sound propagation, have produced several interactive-time algorithms [3, 63, 13]. However, these ray-based methods suffer inaccuracies at low frequencies and are unable to capture some high-order wave effects. Furthermore, ray-based approaches require explicitly defined observer locations as input and do not compute a global sound pressure or fluid density field.

We have developed a framework for simulating realistic acoustic propagation in 3D specifically designed for GPU parallel implementation. We are motivated by the richer set of acoustic properties that physically accurate methods provide and by the need for an efficient solver that can handle the small time steps inherent in acoustic modeling. We leverage nonlocal vector calculus within our LBM framework to create a fast and accurate explicit meshless method. To our knowledge, our approach is the first to retain the parallelizability of traditional grid based LBM approaches while providing the superior boundary treatment of a meshless implementation, and the first to use a compressible fluid solver for acoustic propagation in 3D computer graphics applications. Because the LBM is a compressible fluid solver, we automatically recover the full range of acoustic affects that can be captured by the 3D wave equation, matching the capability of wave-based approaches. Additionally, our GPU parallel implementation provides an efficient tool for computing global pressure fields, enabling a wider range of options for modeling sound sources and observers than geometric ray-based approaches.

Our framework enables acoustic capabilities not yet seen in computer graphics including the ability to model thousands of sound sources, each of which can have unique shape and motion, with little added computational cost. From our final simulated fluid density field, we can extract audio at any number of static or dynamic observer locations. Furthermore, our method can generate spatial audio without requiring the use of Head-Related Transfer Functions (HRTFs) [16], providing a more straightforward approach for producing immersive soundscapes.

In this paper, we provide details for the results listed above, validating convergence to analytic solutions in several experiments, and analyzing the scalability of our implementation. We demonstrate the practicality of our framework for graphics applications by simulating a pair of complex real-world scenarios. Furthermore, we begin by reviewing relevant background on lattice Boltzmann methods and conclude by discussing the limitations of our framework while highlighting future research directions.

Chapter 2: Literature Review

In this section, we review prior applications of LBM to acoustics problems and review the work most relevant to ours on meshless and nonlocal calculus formulations. We also review existing sound propagation methods that are based on wave and ray models.

Lattice Boltzmann Methods. In recent decades, 3D lattice Boltzmann methods (LBMs) have been widely used within computational fluid dynamics [15, 44, 18, 23] and related fields including porous media [60], additive manufacturing [35], supersonic flows [37], heat transfer [41], and multiphase flows [73]. The lattice Boltzmann method also has a localized nature, lending itself to parallel GPU implementation [51, 25].

The LBM has also been used to study the wave propagation in various applications [11]. Most relevant to our approach are the recent works from Benhamou et al. on LBM sound wave propagation in three dimensional cavities. In [6], they obtained good results compared to analytical solutions for point and circular sound sources in cube-shaped cavities. In [5], they generated planar waves by vibrating one of the cavity walls, demonstrating diffraction effects as the waves interact with obstacles inside the cavity. For a wall obstacle with a small opening, they observed a transition from planar wave fronts to spherical wave fronts after passing through the opening. Also relevant is the work of Salomons et al. [62] who simulated 2D outdoor sound propagation for cases including a free field propagation, propagation over solid and porous ground, propagation over a noise barrier, and propagation in wind. They find that sound wave dissipation in the LBM is generally much greater than real dissipation in air, considering the reduction of kinematic velocity and reduction of lattice spacing as possible fixes to this issue. Compared to these previous examples, we apply our LBM sound propagation framework in much more complex 3D inspired by real-world scenes, including a case with multiple moving sound sources and observers.

Meshless Methods. Meshless numerical methods eliminate the need for meshed discretizations and are known to be advantageous for working with curved and complex boundaries. Taking a step towards a meshless LBM, He et al. [32] proposed uncoupling the computational mesh from momentum discretization by adding an interpolation step between the pair of collision and advection steps, allowing for the use of arbitrary nonuniform mesh grids. However, performing interpolation at each time step becomes especially inefficient for 3D problems. Other methods introduce discretizing the momentum space and physical space separately, using numerical schemes such as finite difference [14], finite volume [72], finite element [39], and discontinuous Galerkin [67] to discretize the physical space. This allows the use of nonuniform body-fitted meshes, but adds computational requirements not only for the additional numerical method used, but also for the creation of a high-quality body-fitted mesh. Musavi et al. [49] presented a true meshless lattice Boltzmann method using the meshless local Petrov-Galerkin (MLPG) formulation based on augmented radial basis functions. This approach was recently successfully extended to aeroacoustics by Gorakifard et al. [24]. On the other hand, MLPG typically involves solving for weights x in a linear system $\mathbf{A}x = b$. Solving this system is known to be difficult to parallelize.

Nonlocal Calculus. Nonlocal models involve integral operators that act on the values of a function over nonlocal neighborhoods, such as a Euclidean ball of radius $\delta > 0$, often referred to as the horizon or interaction radius. Gunzburger and Lehoucq formalized nonlocal calculus for scalar functions [29], followed by Du et al. [21], who extended it to vector functions, allowing for the analysis of nonlocal diffusion equations through similar frameworks as the local differential versions.

Nonlocal calculus is based on incorporating a *radius of influence* for each point in the domain, with contributions from points within this radius accumulated through integration. Emerging research connects nonlocal theory to the field of statistical mechanics [74], which includes the LBM. Both nonlocal calculus and statistical mechanics make use of kernel functions to solve multiscale problems. These are advantageous for acoustic simulations where domains can have complex boundary geometries and obstacles of varying scale.

In recent years, nonlocal calculus has gained renewed interest due to the ubiquity of computational methods. Nonlocal methods have been shown capable of modeling phenomena that are challenging for local differential methods, such as multi-scale systems [4, 2, 20], subsurface transport [64, 33, 8], and stochastic processes [48, 22, 12]. Nonlocal methods have also been applied to acoustics problems in Physics [65, 71] and have long been used in fracture mechanics [54], which is relevant to entertainment graphics applications, such as special effects like explosions.

To our knowledge, ours is the first use of a nonlocal advection formulation within the lattice Boltzmann method for acoustic simulation.

Sound Propagation. Most existing sound propagation methods fall under two main categories: wave based methods and geometric ray-based methods. For a complete review of sound propagation methods, we refer the reader to Liu et al. [42].

Wave-based propagation methods divide the computational space using either a volume-based approach, as in the finite element method [69] and finite difference time domain method [61, 9, 19, 52, 30, 10, 70], or using surface-based approach, as in the boundary element method [53, 28] and equivalent source method [46]. Volume-based divisions are best suited to indoor scenes where the total surface area to volume ratio is high while surface-based approaches are best suited for large outdoor scenes where the scene geometry is sparsely distributed [42]. Raghuvanshi et al. [55] applied adaptive rectangular decomposition to achieve a fast finite-difference time domain (FDTD) method which was further accelerated in a GPU implementation by Mehra et al. [47]. However, interactive wave-based propagation has only been achieved by restricting the simulation to a 2D slice as in [58] or by leveraging precomputed sounds as in [57, 56].

Geometric propagation methods calculate the paths of sound propagation through energy attenuation by assuming that sound travels in straight lines. In recent decades, several interactive time geometric methods have been proposed. Atani et al. [3] achieved this through a combination of ray tracing and acoustic radiance transfer using precomputed transfer operators. Schissler et al. [63] developed an interactive technique for large dynamic environments based on a backward ray tracing algorithm. Cao et al. [13] proposed a bidirectional path tracing method, which converges faster than forward only or backward only methods, and can handle scenes with complex sound occlusions. To address limitations in recreating low frequency acoustic effects like diffraction and occlusion, Rungta et al. [59] presented a combined ray tracing method using precomputed diffraction kernels.

Chapter 3: Lattice Botlzmann Methods Primer

In this section, we review core concepts of the lattice Boltzmann method and additional considerations that are relevant to the problem of sound propagation. For a complete review of the standard LBM formulation, we refer the reader to [36].

3.1 Lattice Boltzmann Method

The Boltzmann transport equation offers a statistical mechanics approach to fluid dynamics, modeling the evolution of probability distributions that represent microscopic fluid particle states. The transport equation is articulated as:

$$\frac{\partial f(\boldsymbol{\xi}, \boldsymbol{v}, t)}{\partial t} + \boldsymbol{v} \cdot \nabla f(\boldsymbol{\xi}, \boldsymbol{v}, t) = \Theta(f(\boldsymbol{\xi}, \boldsymbol{v}, t)). \tag{3.1}$$

Here, f represents a particle distribution function describing the probability of finding a microscopic fluid particle with position $\boldsymbol{\xi}$ and velocity $\boldsymbol{v}$ at time t. The term Θ models the statistical collision process of fluid particles, guiding the distribution towards an equilibrium.

From a macroscopic perspective, fluid properties like density and momentum emerge naturally from the moments of this distribution. The zeroth- and first-order moments can be expressed as:

$$\rho = \int f(\boldsymbol{\xi}, \boldsymbol{v}, t)\ d\boldsymbol{v}, \quad \text{and} \quad \rho\boldsymbol{u} = \int \boldsymbol{v} f(\boldsymbol{\xi}, \boldsymbol{v}, t)\ d\boldsymbol{v}. \tag{3.2}$$

In the above, ρ and $\boldsymbol{u}$ signify the macroscopic fluid density and velocity, respectively. The second-order moment computes the shear-strain tensor or momentum flux. Higher order moments have no physical meaning, but are important in certain collision models.

It is essential to distinguish macroscopic velocities $\boldsymbol{u}$ from microscopic velocities $\boldsymbol{v}$, where the former is an emergent property of the solution while the latter acts as an independent variable in the transport equation and does not represent a solution to it. This requires the equation to be discretized not only in space and time, but also velocity. To discretize velocity, first consider the quantities of interest, the macroscopic fluid properties, and the velocity term's role in computing them. From Equation 3.2, integrating over velocity gives the macroscopic values. It is therefore reasonable to discretize velocity-space using a quadrature method [31] such that the integrals in Equation 3.2 can be rewritten as

$$\rho = \sum_{\alpha} w_\alpha f_\alpha(\boldsymbol{\xi}, t), \quad \text{and} \quad \rho\boldsymbol{u} = \sum_{\alpha} w_\alpha \boldsymbol{c}_\alpha f_\alpha(\boldsymbol{\xi}, t), \tag{3.3}$$

where α is an index for a finite set of velocity values chosen as the quadrature points for solving the integrals. Indexing into the velocity values gives the quadrature point $\boldsymbol{c}_\alpha$ and its associated weight w_α. Moreover, the selection of quadrature points and their weights result from expanding $f(\boldsymbol{\xi}, \boldsymbol{v}, t)$ in $\boldsymbol{v}$ using a Hermite polynomial basis [27, 66]. For these basis polynomials, which are orthonormal, the expansion coefficients actually correspond to lattice velocity moments up to the given degrees [26]. Thus, the choice of quadrature points defines a lattice structure.

Using lattice discretization, Equation 3.1 is rewritten as

$$\frac{\partial f_\alpha(\boldsymbol{\xi}, t)}{\partial t} + \boldsymbol{c}_\alpha \cdot \nabla f_\alpha(\boldsymbol{\xi}, t) = \Theta(f_\alpha(\boldsymbol{\xi}, t)), \tag{3.4}$$

which eliminates the velocity term, instead using n_α linear advection equations that can be solved separately. With the PDEs expressed as functions of only space and time, established methods for numerical solutions can be used to compute emergent properties of the transport equation, i.e., the lattice Boltzmann methods.

Equation 3.1 encapsulates the transport of fictive microscopic particles and their interactions. We derive our nonlocal formulation of the lattice Boltzmann transport equation in Section 4.1. These particles collide and exchange energy while conserving mass and momentum according to the collision operator Θ which guides the system toward a hydrodynamic equilibrium. The probability distribution f is updated according to the velocities after collision. The classic Bhatnagar-Gross-Krook collision model (BGK) describes a single particle reaching this equilibrium after relaxation time τ,

$$\Theta_{BGK}(f_\alpha) = -\frac{f_\alpha - \bar{f}_\alpha}{\tau}, \tag{3.5}$$

where τ is computed from the fluid viscosity ν as $\tau = 3\nu + \frac{1}{2}\Delta t$. The equilibrium value $\bar{f}$ in the above equation for dimension D is given by

$$\bar{f}(\boldsymbol{v}, \rho, \boldsymbol{u}) = \frac{\rho}{(2\pi)^{D/2}} \exp\left(-\frac{||\boldsymbol{v} - \boldsymbol{u}||_2^2}{2}\right). \tag{3.6}$$

Expanding the equilibrium value in $\boldsymbol{v}$ using Hermite polynomials and limiting to second order terms gives

$$\bar{f}_\alpha(\rho, \boldsymbol{u}) \approx w_\alpha \rho \left(1 + \frac{\boldsymbol{c}_\alpha \cdot \boldsymbol{u}}{c_s^2} + \frac{(\boldsymbol{c}_\alpha \cdot \boldsymbol{u})^2}{2c_s^4} - \frac{\boldsymbol{u} \cdot \boldsymbol{u}}{2c_s^2}\right). \tag{3.7}$$

3.2 LBM for Sound Propagation

Compared to the classic BGK model, the central-moment multiple relaxation time model (CM-MRT) is advantageous for low viscosity fluids (with high Reynolds numbers) and has been shown to better handle acoustic dissipation [62]. The CM-MRT collision operator is given by

$$\boldsymbol{\Theta} = \mathbf{K}^{-1}\mathbf{R}\mathbf{K}(\mathbf{f} - \bar{\mathbf{f}})\Delta t, \tag{3.8}$$

where $\mathbf{R}$ is a diagonal matrix containing the relaxation times for each moment. Here the equilibrium moments $\bar{\mathbf{k}} = \mathbf{K}\bar{\mathbf{f}}$, where $\mathbf{K}$ is the transformation matrix, primarily nullify to zero, aiding in determining the inverse analytically. Additionally, these moments are independent of velocity, which ensures Galilean invariance, bolstering stability in simulations with reduced viscosities [40].

Acoustic simulations also benefit from attenuating the amplitude of reflections at domain boundaries. Not only does this allow for modeling different materials and reduce echo, but a near-fully absorbent boundary can be used to simulate open-air boundaries. We implement a perfect matching layer (PML) near domain boundaries to model absorption. Unlike a sponge layer, the absorbing layer is designed to match the governing equations, theoretically resulting in no reflected waves [50]. The PML is applied as an added term to Equation 3.4,

$$\frac{\partial f_\alpha}{\partial t} + \boldsymbol{c}_\alpha \cdot \nabla f_\alpha = \Theta_{collision} + \Theta_{PML}. \tag{3.9}$$

In 3D, the PML is defined as

$$\begin{aligned}\Theta_{PML} = &- \left[3\sigma \bar{f}_\alpha^{neq} + 3\sigma^2 Q_1 + \sigma^3 Q_2 + 2\sigma \nabla \cdot (\boldsymbol{c}_\alpha Q_1)\right] \\ &+ \left[\sigma^2 \nabla \cdot (\boldsymbol{c}_\alpha Q_2)\right],\end{aligned} \tag{3.10}$$

where $\bar{f}_\alpha^{neq}$ is the acoustic perturbation of the distribution such that $f_\alpha = \bar{f}_\alpha + \bar{f}_\alpha^{neq}$, and σ is a tuning parameter for altering the amount of acoustic damping. The terms Q_1 and Q_2 are defined implicitly by

$$\frac{\partial Q_1}{\partial t} = \bar{f}_\alpha^{neq}, \quad \text{and} \quad \frac{\partial Q_2}{\partial t} = Q_1.$$

We can integrate these partial derivatives using Euler-Maclaurin quadrature [68] giving

$$Q_1^k = \frac{\Delta t}{2}\left(\bar{f}_\alpha^{neq\ k} + \bar{f}_\alpha^{neq\ k-1}\right), \quad \text{and} \quad Q_2^k = \frac{\Delta t}{2}\left(Q_1^k + Q_1^{k-1}\right),$$

With the explicit forms of Q_1 and Q_2, Equation 3.9 is discretized in space using our nonlocal method presented in section 4.1.

Sound sources can be represented in the LBM spatial domain as perturbations in the fluid density field at specified solution nodes [7]. For a frequency ω measured in rad/s, the condition is defined as:

$$\rho^k = \rho_0 + \rho_a \sin(\omega t^k), \tag{3.11}$$

where ρ_0 is the base density of the medium, and ρ_a denotes the amplitude of the perturbation. For the discrete case the equation becomes:

$$\rho^k = \rho_0 + \rho_a \epsilon^k, \tag{3.12}$$

where ϵ^k represents the kth sample from the sound source's discrete audio data. Such density perturbations can be defined over arbitrary regions, allowing for both point and area sound sources to be modeled.

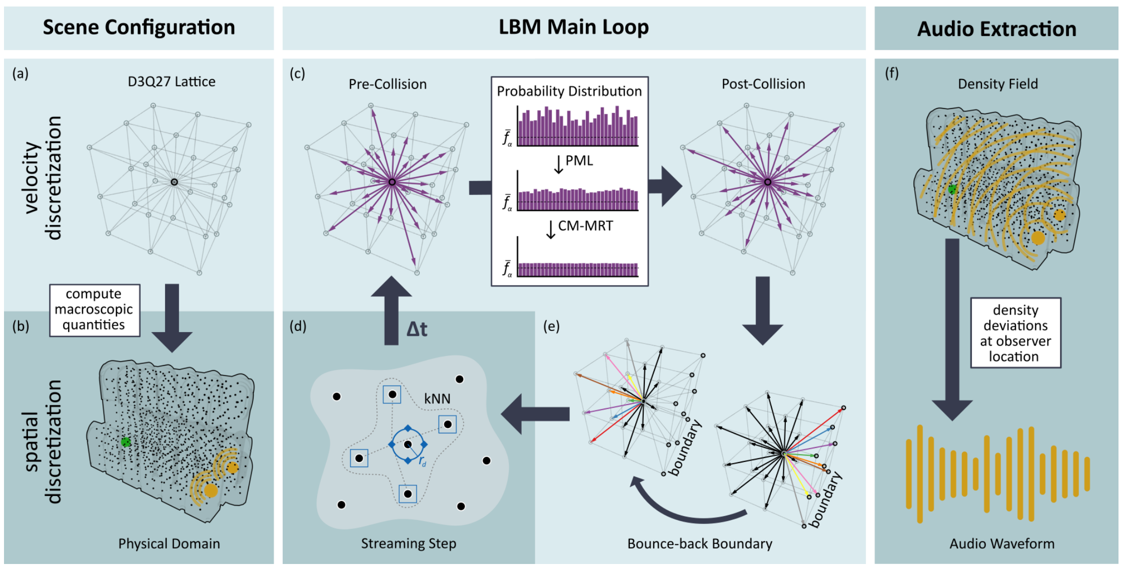


Figure 3.1: Our sound propagation framework contains three main phases: Scene configuration (a,b), LBM time step integration (c,d,e), and audio extraction (f). Macroscopic calculations (dark background) use a meshless space discretization (b) while microscopic calculations (light background) are performed on the D3Q27 lattice grid (a). At the start of each time step loop, macroscopic density and velocity values are calculated, followed by the collision, bounce-back boundary, and streaming steps. In (a) and (f), the observer inside the spatial domain is drawn as a green dot, the sound sources as yellow dots, and the macroscopic solution nodes as small black dots. The collision step (c) illustrates how the velocity probability distributions are affected by the perfect matching layer (PML) applied at boundary nodes and the central moment multiple relaxation time collision model (CM-MRT). In (e), the streaming step quadrature points for a given node are drawn as blue diamonds and interpolation nodes selected by k-nearest neighbors search are highlighted by blue boxes.

Chapter 4: Reformulation of LBE using Nonlocal Advection Equation

Our sound propagation framework is based on a meshless discretization of the Boltzmann transport equation using a method of nonlocal calculus and Gauss quadrature (NC-GQM). We adopt the same velocity discretization as standard LBM, rewriting the subsequent lattice PDEs in their nonlocal form. The integrals of the nonlocal form are then approximated using Gauss quadrature, which allows the continuous position variable to be replaced with discrete values. The spatial discretization assumes no connectivity between points, instead opting for the local radial point interpolation method (LRPIM) to approximate field values at quadrature points. The remaining differential in time is integrated using a Runge-Kutta time stepping scheme.

4.1 Nonlocal Linear Advection Model

Consider the nonlocal divergence operator which maps a two-point vector function to a scalar function:

$$\nabla \cdot \boldsymbol{\psi}(\boldsymbol{x}) \rightarrow \int_{\Omega} (\boldsymbol{\psi}(\boldsymbol{x}) + \boldsymbol{\psi}(\boldsymbol{y})) \cdot \boldsymbol{\alpha}(\boldsymbol{x}, \boldsymbol{y}) \mathrm{d}\boldsymbol{y}, \tag{4.1}$$

where $\boldsymbol{\psi}$ is a vector-valued function, $\boldsymbol{\alpha}(\boldsymbol{x},\boldsymbol{y})$ is a two-point vector valued anti-symmetric kernel function and Ω is a region containing $\boldsymbol{x}$ [21]. Here, we use $\boldsymbol{x}$ to distinguish the spatial domain nodes in a nonlocal vector calculus context from the spatial nodes $\boldsymbol{\xi}$ in Section **??**. Furthermore, we overload the use of α to remain consistent with notational conventions from both LBM and nonlocal vector calculus. The anti-symmetric kernel function is formulated by taking the gradient of a radially symmetric kernel function [38]. In our implementation, we choose the W_4 B-Spline. Substituting the nonlocal divergence operator into Equation 3.4 and replacing $\boldsymbol{\psi}(\boldsymbol{x})$ with $\boldsymbol{c}_\alpha f_\alpha(\boldsymbol{x},t)$ gives the lattice transport equation in its nonlocal form

$$\frac{\partial f_\alpha(\boldsymbol{x},t)}{\partial t} = -\int_\Omega \boldsymbol{c}_\alpha(f_\alpha(\boldsymbol{x},t) + f_\alpha(\boldsymbol{y},t)) \cdot \boldsymbol{\alpha}(\boldsymbol{x},\boldsymbol{y})\mathrm{d}\boldsymbol{y} + \Theta(f_\alpha(\boldsymbol{x},t)). \tag{4.2}$$

In the above, we keep the partial with respect to time, as it will be integrated using a standard explicit time stepping scheme.

Discretizing the reformulated transport equation requires first approximating the integrals using some quadrature method; in this case, the Gauss-Legendre method. The region of integration Ω is spherical with an interaction radius of r_{inter}, so a change of variables is applied:

$$\boldsymbol{y} = x + \begin{pmatrix} r\sin(\phi)\cos(\theta) \\ r\sin(\phi)\sin(\theta) \\ r\cos(\phi) \end{pmatrix} \tag{4.3}$$

where $0 \leq r \leq r_{inter}$, $\frac{-\pi}{2} \leq \phi \leq \frac{\pi}{2}$ and $0 \leq \theta \leq 2\pi$. Then Equation 4.2 can be approximated

$$\frac{\partial f_\alpha(\boldsymbol{x},t)}{\partial t} \approx -\gamma_\theta \gamma_\phi \sum_l^{n_r} \sum_m^{n_\theta} \sum_n^{n_\phi} \gamma_{r,l} \left[\boldsymbol{c}_\alpha (f_\alpha(\boldsymbol{x},t) + f_\alpha(\boldsymbol{y}_{l,m,n})) \right] \cdot r^2 \sin(\phi)\, \boldsymbol{\alpha}(\boldsymbol{x}, \boldsymbol{y}_{l,m,n}) + \Theta(f_\alpha(\boldsymbol{x},t)), \tag{4.4}$$

where $\boldsymbol{y}_{l,m,n}$ is a quadrature point and n_r, n_θ and n_ϕ are the number of points for each spherical dimension. Further, $\gamma_{r,l}$ is the quadrature weight for the l-th radial Gauss-Legendre quadrature point and γ_θ and γ_ϕ are the angular quadrature weights given by $\gamma_\theta = 2\pi/n_\theta$ and $\gamma_\phi = \pi/n_\phi$. For clarity the l, m, and n are consolidated into a single quadrature index q defined accordingly: $n_q = n_r n_\theta n_\phi$, $\ l(q) = q \pmod{n_r}$, $\ m(q) = \lfloor q/n_r \rfloor \pmod{n_\theta}$, $\ n(q) = \lfloor q/(n_r n_\theta) \rfloor$. The condensed version of Equation 4.4 is written as

$$\frac{\partial f_\alpha(\boldsymbol{x},t)}{\partial t} \approx -\gamma_\theta \gamma_\phi \sum_q^{n_q} \gamma_{r,q} \left[\boldsymbol{c}_\alpha (f_\alpha(\boldsymbol{x},t) + f_\alpha(\boldsymbol{y}_q)) \right] \cdot r^2 \sin(\phi)\, \boldsymbol{\alpha}(\boldsymbol{x}, \boldsymbol{y}_{l,m,n}) + \Theta(f_\alpha(\boldsymbol{x},t)), \tag{4.5}$$

4.2 Local Radial-Point Interpolation

With the required terms for spatial discretization represented in summation form, let the discretization chosen be a set of points with no connectivity. To calculate the macroscopic quantities (from Equation 3.2), we use local radial-point interpolation (LRPIM) to approximate the solution at the quadrature points $f_\alpha(\boldsymbol{\zeta}_q, t)$

from nearby solution nodes. Consider an arbitrary function $\boldsymbol{\chi}(\cdot)$, which can be locally approximated at a point $\boldsymbol{\zeta}_i$ using a radial basis function r_b evaluated at n_b nearby points such that

$$\boldsymbol{\chi}(\boldsymbol{\zeta}_i) \approx \sum_{b=1}^{n_b} B_b(r_b(\boldsymbol{\zeta}_i))a_b + \sum_{p=1}^{4} P_p(\boldsymbol{\zeta}_i)b_p = \boldsymbol{B}^T\boldsymbol{a} + \boldsymbol{P}^T\boldsymbol{b} \tag{4.6}$$

where $B_b(r_b(\boldsymbol{\zeta}_i)) = r_b(\text{dist}(\boldsymbol{\zeta}_i, \boldsymbol{\zeta}_{\text{neighbor}}))$ and a_b is an interpolating weight. We use a multi-quadratic shape function given by $r_b(d) = (d^2 + C_0^2)^\eta$ where C_0 and η are shape parameters. We augment the radial basis functions with a monomial term where $\boldsymbol{P} = \begin{pmatrix} 1 & \boldsymbol{\zeta}_i\hat{\mathbf{i}} & \boldsymbol{\zeta}_i\hat{\mathbf{j}} & \boldsymbol{\zeta}_i\hat{\mathbf{k}} \end{pmatrix}$ where $\hat{\mathbf{i}}, \hat{\mathbf{j}}, \hat{\mathbf{k}}$ are the directions in 3D. Suppose that Equation 4.6 intersects the domain at the supporting solution nodes $S = \{\boldsymbol{\zeta}_i^1, \ldots, \boldsymbol{\zeta}_i^{n_b}\}$. For each $\boldsymbol{\zeta}_i^j \in S$, we add the constraint

$$\sum_{b=1}^{n_b} P_p(\boldsymbol{\zeta}_i^j)a_b = 0 \quad \forall p. \tag{4.7}$$

Then with $\boldsymbol{\chi}^* := (\chi(\boldsymbol{\zeta}_i^1), \ldots, \chi(\boldsymbol{\zeta}_i^{n_b}))^T$, we get the following linear system for the augmented weights:

$$\begin{pmatrix} \boldsymbol{B}_0 & \boldsymbol{P}_0 \\ \boldsymbol{P}_0^T & \mathbf{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{a} \\ \boldsymbol{b} \end{pmatrix} := \boldsymbol{G}_0 = \begin{pmatrix} \boldsymbol{\chi}^* \\ \mathbf{0} \end{pmatrix} \tag{4.8}$$

where $\boldsymbol{B}_0$ and $\boldsymbol{P}_0$ are given respectively as

$$\begin{pmatrix} B_1(r_1(\boldsymbol{\zeta}_i^1)) & \dots & B_{n_b}(r_{n_b}(\boldsymbol{\zeta}_i^1)) \\ \dots & \dots & \dots \\ B_1(r_1(\boldsymbol{\zeta}_i^{n_b})) & \dots & B_{n_b}(r_{n_b}(\boldsymbol{\zeta}_i^{n_b})) \end{pmatrix}, \quad \begin{pmatrix} 1 & (\boldsymbol{\zeta}_i^1)_1 & (\boldsymbol{\zeta}_i^1)_2 & (\boldsymbol{\zeta}_i^1)_3 \\ & \dots & \dots & \\ 1 & (\boldsymbol{\zeta}_i^{n_b})_1 & (\boldsymbol{\zeta}_i^{n_b})_2 & (\boldsymbol{\zeta}_i^{n_b})_3 \end{pmatrix}.$$

Then Equation4.6 can be rewritten as

$$\boldsymbol{\chi}(\boldsymbol{\zeta}_i) = \boldsymbol{G}^T\boldsymbol{G}_0^{-1}\begin{pmatrix} \boldsymbol{\chi}^* \\ \boldsymbol{0} \end{pmatrix} \tag{4.9}$$

where $\boldsymbol{G} = \left(\boldsymbol{B}^T \ \boldsymbol{P}^T\right)^T$.

Let $\boldsymbol{\chi}^*$ be defined in terms of the discretized nonlocal lattice transport equation, $\boldsymbol{\chi}^*_{i,q} := (f_\alpha(\boldsymbol{\zeta}_q^1), \dots, f_\alpha(\boldsymbol{\zeta}_q^{n_b}))$ where f_α is the probability distribution at each node on the velocity lattice (Equation 3.4). Using LRPIM to solve for $f_\alpha(\boldsymbol{\zeta}_q)$,

$$\begin{aligned} \frac{\partial f_\alpha(\boldsymbol{x}_i, t)}{\partial t} \quad \approx \quad & -\gamma_\theta\gamma_\phi \sum_q^{n_q} \gamma_{r,q} \left[f_\alpha(\boldsymbol{x}_i, t) + \boldsymbol{G}^T\boldsymbol{G}_0^{-1}\begin{pmatrix} \boldsymbol{\chi}^*_{i,q} \\ \boldsymbol{0} \end{pmatrix} \right] \\ & \left[\boldsymbol{c}_\alpha \cdot \boldsymbol{\alpha}(\boldsymbol{x}_i, \boldsymbol{\zeta}_q)\right] \cdot r^2 \sin(\phi) + \Theta(f_\alpha(\boldsymbol{x}_i, t)). \end{aligned} \tag{4.10}$$

Although not explicitly shown, $\boldsymbol{G}^T\boldsymbol{G}_0^{-1}$ is unique to each quadrature term.

Typically, the neighboring points surrounding $\boldsymbol{\zeta}_i$ are chosen using a radius of influence. However, this does not lend itself well to GPU implementation as the number of neighbors can vary from point to point. To improve this, we utilize a

k-nearest neighbors search (kNN) in order to constrain the memory footprint to a user-defined constant value. For even more consistency, we use the same set of nearest neighbors as the interpolation nodes for all the quadrature points associated with $\boldsymbol{\zeta}_i$. While kNN methods are known to introduce large dispersive errors, we find them to be insignificant within our framework. This is because we are using kNN to approximate a local radius for LRPIM, a highly-accurate interpolation method, rather than simply convolving a kernel across nearest points to approximate the continuous solution. It is also known that naive search methods for kNN can lead to undesired sampling biases, giving greater weight along certain directions [43]. To overcome this bias, our CUDA-based, partitioned kNN implementation accelerates the search process by introducing a pseudo-random list of distinct search vectors for a cube grid around the search point. The search terminates once all grid cells at a certain distance along the selected directions have been searched and the k-nearest neighbors have been found. We employ the prime modulo hashing function proposed by [34] to generate the randomized list, incorporating a selective array of prime numbers to accommodate varying list sizes within the kNN search.

Equation 4.10 can be simplified to the following:

$$\frac{\partial f_\alpha(\boldsymbol{x}_i, t)}{\partial t} \approx \mathbf{M}\mathbf{f}_\alpha + \Theta(f_\alpha(\boldsymbol{x}_i, t)). \tag{4.11}$$

where $\mathbf{M}$ is a sparse matrix and $\mathbf{f}_\alpha$ is the vector of distribution values for all nodes for the lattice index α.

Finally, to derive temporal integration for the spatial points in our LBM formulation, we adopt the standard 4th-order Runge-Kutta time stepping scheme whose implementation is straightforward.

The discrete solution in Equation 4.11 enables the GPU implementation of our meshless framework because the streaming step is reduced to a series of sparse matrix-vector multiplications. Our use of a kNN search within the LRPIM limits the sparsity of the matrices to k non-zero values per row, meaning our nonlocal streaming step is linear in the number of solution nodes. The collision step is also known to be linear because of its local nature, implying that our complete sound propagation loop is linear. This is comparable to the performance of traditional grid-based LBM formulations. Future work is needed to determine if our implementation achieves this, however, we confirm this linear complexity in the weak scaling plot of Figure 7.1.

Chapter 5: Simulation Results

In this section, we demonstrate the application of our framework to six sound propagation scenarios: an indoor cathedral scene, an outdoor city scene, a Doppler shift experiment, a double slit experiment, an experiment with many sound sources, and a stereoscopic audio experiment. We visualize the results in Paraview [1] to show the interaction between propagated sound waves and the domain boundaries. These scenarios are designed to capture complex acoustic phenomena using a variety of sound source configurations in plausible real-world environments. In the supplementary material we include a video of the cathedral and city scenes as well as the stereoscopic audio experiment. The city scene includes a moving car which produces a noticeable Doppler effect in our video.

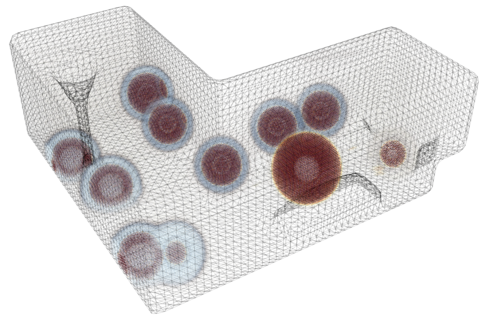
Cathedral Scene. In this scene, we aim to simulate a large indoor environment where the walls, ceilings, and floors all reflect wave energy. Our input sound is a single acoustic pulse. In Figure 1.1, we show three visualizations of the resulting density field inside the 3D domain defined by a cathedral model mesh. This domain validates two desired acoustic properties for propagation: reverberation and diffraction. The walls of the cathedral have only the minimum required damping for a stable solution, which we determined experimentally. This allows us to explore the capacity of our framework for modeling reverberation in an extreme scenario of nearly full acoustic reflection at the domain boundaries. The columns provide additional obstacles and should lead to a diffraction of any incoming waves.

Figure 1.1a shows the pulse propagating in the domain volume. Figure 1.1b shows a horizontal cross-section of the domain. We observe wave fronts, depicted by red-orange bands in the visualizations, appearing behind the columns, indicating diffraction. While visible, these effects could be exaggerated by tuning the wavelength of the initial pulse. Our double slit experiment (see below) displays clearer interference pattens, further demonstrating our framework's capabilities in modeling diffraction. Figure 1.1c shows a vertical cross-section of the domain. We can clearly see wave fronts reflected from the floor of the domain. We also see interference patterns along the left-side wall of the cathedral where the initial and reflected wave fronts interact.

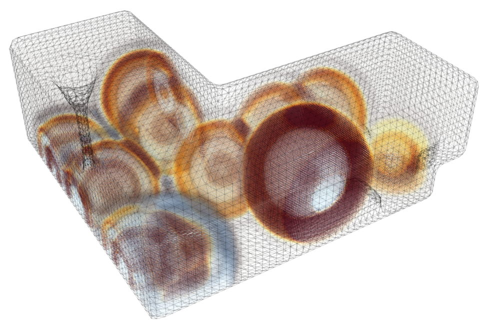
City Scene. In this scenario, we model an outdoor city scene consisting of a sidewalk and alleyway that intersect in an L shape (Figure 5.1). The sidewalk borders a street with moving cars. A single tree between the sidewalk and street acts as an obstacle in the 3D domain. The scene contains a total of 11 sound sources: a car idling in the alley, three pigeons standing along the sidewalk cooing, a water feature containing five distinct point sources along the side of the alley, an air conditioning unit in the back corner of the alley, and a moving car passing by on the street. Figure 5.1a shows the sound waves propagating in the domain before any major interactions occur, identifying locations of the sound sources. This scene also includes three stationary observers and one moving observer walking along the sidewalk and into the alley. For visualization purposes, the start of the input audio for each sound source was aligned such their amplitudes were the same in the first simulation time step. As the simulation progresses further, we observe more interactions between waves from different sources in Figure 5.1b.

This simulation demonstrates that our framework is capable of modeling many different sound sources effectively, where their generated waves interact to form complex interference patterns. Furthermore, the sound sources used include a wide range of input frequencies, differences in amplitudes, and some inputs that are more or less white noise. These are all challenging factors in acoustic simulation, suggesting the robustness of our framework for handling highly varied scenes.

Doppler Effect. Our goal in this scenario is to replicate the Doppler shift effect. We use a 3D rectangular domain, simulating open air boundaries on all sides. The scene contains a single sound source moving left to right along a fixed axis from the center of the domain. Figure 5.2 shows the resulting waves propagated using our framework, where the simulation nodes and a slice along the x-z plane intersecting the origin are both shown. The nodes are visualized using a low opacity so that the wave fronts, which are better demonstrated along the slice, are more clearly visible. We see that the wave front bands are more spread out behind the sound source and more compressed in front along the direction of travel. This visual confirmation indicates that our framework can successfully simulate the compression and expansion of wave fronts for moving sound sources. Additionally, we evaluate the errors between our simulated frequencies and the expected ones from the Doppler effect in Section **??**.



(a) Early propagation time step.



(b) Later propagation time step.

Figure 5.1: Sound propagation at two different time steps in our city scene. Figure (a) indicates the locations of the sound sources. Figure (b) shows interactions with the boundaries as the sound waves begin to propagate and interference patterns between waves from different sound sources.

Double Slit Experiment. Figure 5.3 presents a simulated version of the double slit experiment. The double slit experiment demonstrates how a plane wave incident to a pair of openings in a wall will produce interference patterns due to diffraction on the other side of the wall. Such a pattern is clearly visible in our simulated results. This shows that our framework is capable of modeling diffraction in a way that is not only believably, but accurate. A periodic interference pattern as shown is only possible if the diffraction is behaving in a physically realistic way.

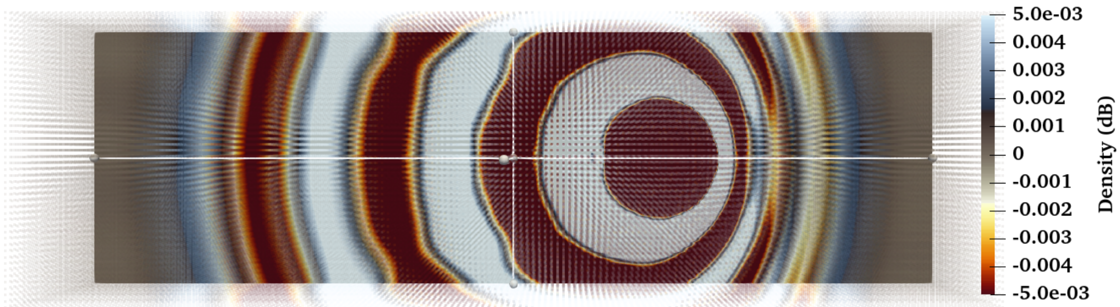


Figure 5.2: Sound propagation in our replication of the Doppler effect. A single sound source travels from the center of the domain toward the right side. As expected, the wave fronts are compressed in the direction of motion and expanded in the opposite direction.

Many Sound Sources. In this experiment, we demonstrate the capability of our framework for modeling into the thousands of independent sound sources. Figure 5.4 visualizes the propagation of 4096 point sound sources on an $8 \times 8 \times 8$ grid within a cube domain with a side length 2m. The blue circle patterns shown in the cross-section slices of the domain indicate the source locations. The red patches around each blue circle indicate interference patterns between sound waves emitted from different sound sources. Due to the high amplitude of accumulated sound from the many sources in this experiment, we do not include the audio results in our supplementary video.

Stereoscopic Sound. To capture stereo spatial sound, we use a 3D model of a simplified human head as an immersed boundary within the simulation. By measuring density variations at each ear and recording the data to dual-channel audio files, we capture spatial audio produced by the various interactions of the waves with the head as well as the offset that occurs by having two listening points.

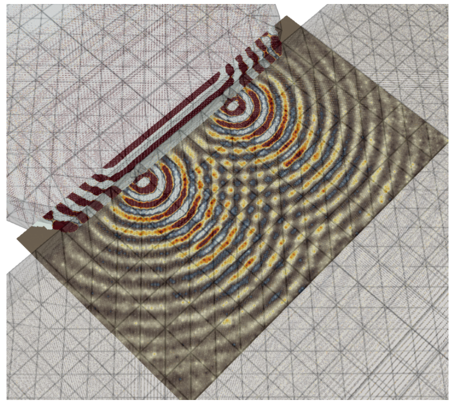


Figure 5.3: Sound propagation results for our double slit experiment with a plane wave (upper left). Diffraction and interference patterns can be seen on the exit side of the two slits (lower right).

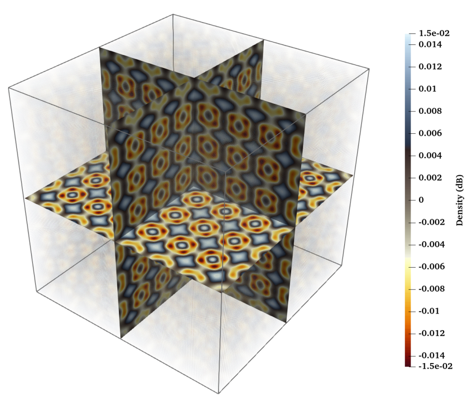


Figure 5.4: Demonstration of our method's ability to scale linearly with the number of sound sources simulated. Our experiment used sinusoidal sound sources equidistantly placed in a cubic domain with side length 2m.

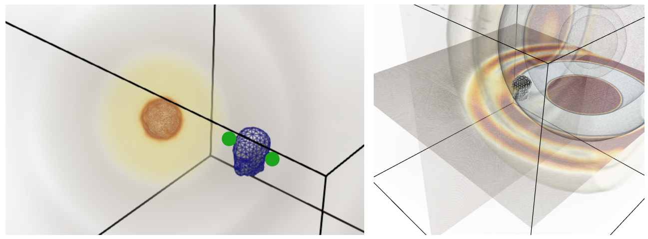


Figure 5.5: Visualization of sound waves from a point source (left side, yellow dot) interacting with a simplified 3D head model. Our framework allows stereoscopic sound to be captured by measuring density variations around each ear (left side, green dots) and recording the data as a two-channel audio file.

Figure 5.5 demonstrates this interaction with an incident wave. The limitation, currently, is that our method does not support moving boundaries, so observers that capture stereo sound must be stationary. Future research is needed to assess the perceived quality of the simulated spatial audio via a user study, which can be compared to existing approaches using head-related transfer functions (HRTFs).

Chapter 6: Accuracy and Convergence

To test for the accuracy and convergence of our framework, we conducted two experiments and performed error analysis. These experiments indicate that our simulated solutions are within acceptable error ranges of the known solutions.

Experiment 1. Following a standard verification method, we simulate a sinusoidal spherical source at the center of a 3D domain for which the analytical solution to density field deviations have the known value

$$\delta\rho(\boldsymbol{x},t) = \begin{cases} \delta\rho(\mathbf{0},t), & 0 \leq ||\boldsymbol{x}|| < r_s \\ \frac{r_s}{||\boldsymbol{x}||}\delta\rho(\mathbf{0}, t - \frac{||\boldsymbol{x}||-r_s}{c_s}), & r_s \leq ||\boldsymbol{x}|| < c_s t - r_s \\ 0, & \text{otherwise} \end{cases} \tag{6.1}$$

where $\delta\rho(\mathbf{0},t) = \rho_a \sin(\omega t)$, r_s is the radius of the sound source and c_s is the speed of sound. Then density field is simply $\rho(\boldsymbol{x},t) = \rho 0 + \delta\rho(\boldsymbol{x},t)$.

Since we aim to model acoustics in air, we choose a speed of sound $c_s = 343\text{m/s}$. Figure 6.1 demonstrates the simulated solution for a 1000Hz sine wave at a given time and visually compares the simulated waveforms against expected waveforms along the x-axis of the 3D domain. We plot the error of the simulated results for various frequencies in Fig. 6.2. The x-axis is normalized to the macroscopic node resolution given by $c_s/(f \cdot r_d)$, which is inversely proportional to the radius

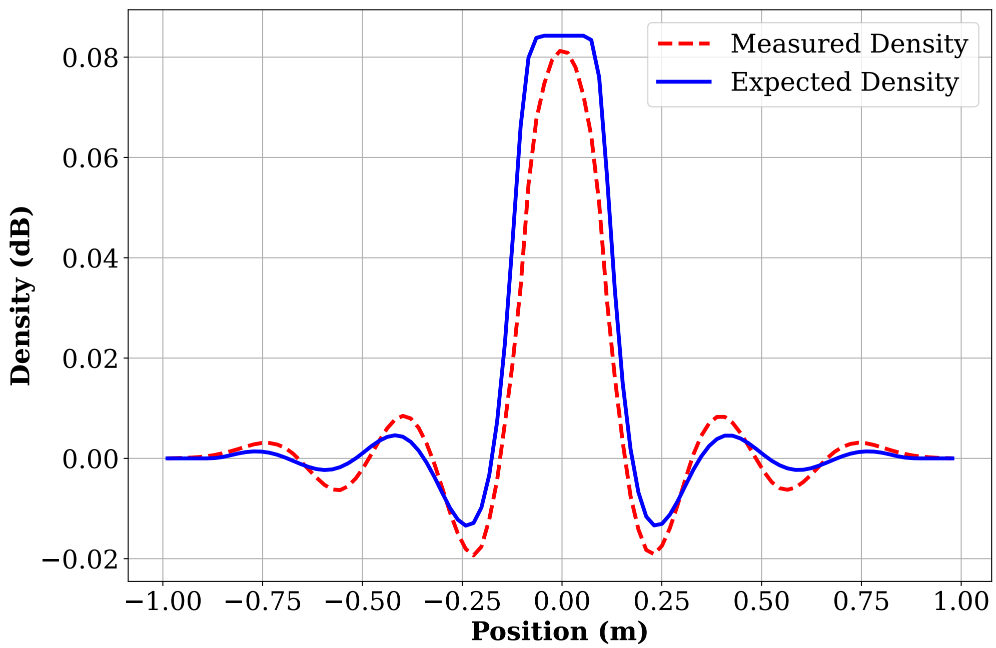


Figure 6.1: Experiment 1 simulation results from of sound propagation from a sinusoidal source verified against the theoretical solution. Expected (blue) density deviations are compared against simulated (red) density deviations along the X-axis of a 3D domain.

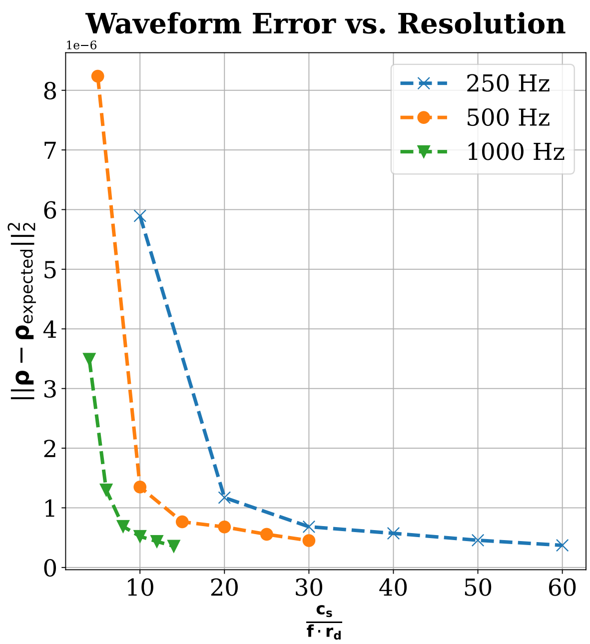


Figure 6.2: Experiment 1 error between our simulated sound waveforms and the analytic solution vs. macroscopic node resolution $c_s/(f \cdot r_d)$ where c_s is the speed of sound and f is the simulated frequency. The simulated waveform was generated via a sine wave at the origin. The magnitude on the vertical axis is 10e-6.

of interaction r_d for our non-local streaming step. This facilitates an intuitive interpretation of what nodal spacing is required to resolve a given frequency.

In Figure 6.2, we observe rapid decreases in error as the node resolution increases. Furthermore, the error for each frequency converges past a certain resolution threshold. For the three frequencies chosen, 250Hz, 500Hz, and 1000Hz, we find that node resolutions of 40, 20, and 10 respectively guarantee solution convergence.

Experiment 2. Our framework can also accurately capture the Doppler shift

Scene	$\mathbf{c}_{\text{sound}}$ (m/s)	f_{target} (Hz)	Sim. Length (s)	Node Count	r_d (mm)	Δt (μs)	Sim. Time (hrs)
City	550	1150	10	1,856,580	80	72	15
Stereo-scopic Sound	343	2000	6	1,642,280	18	25	19
Cathed-ral	550	1200	6	3,964,536	76	69	24

Table 6.1: This table provides timing data for three example problems simulated using a Linux desktop workstation with a AMD Ryzen 5800x CPU, 32GB RAM, and an Nvidia RTX 4090 GPU. These timings are indicative of the computational requirements for simulating a real-world sound propagation problem using our framework. For each demonstration, three user inputs are provided: the speed of sound in the fluid medium, $\mathbf{c}_{\text{sound}}$, the maximum frequency to be resolved, f_{target}, and the length of simulated audio in seconds. These parameters define the nodal spacing r_d and the total number of solution nodes. The time step Δt is set to $.5r_d/\mathbf{c}_{\text{sound}}$ to ensure stability.

effect. We compute the error between the frequency of simulated waveforms and of the expected frequency predicted by the Doppler shift formula $f_o = f_s(c+v_o)/(c+v_s)$ where f_s is the source frequency, c is the speed of sound, v_o is the signed speed of the observer and v_s is the signed speed of the sound source. We perform this experiment for two cases: a stationary observer with the source moving toward them, and the sound source moving away from them. In both cases the motion is axis aligned. To compute the error, we record the waveforms during the time it takes for 12 peaks to pass the observer. We then perform a fast Fourier transform (FFT) on these simulated waveforms and compare with the FFT of the expected waveforms. Figure 6.3 visualizes both the computed errors and normalized discrete

FFTs.

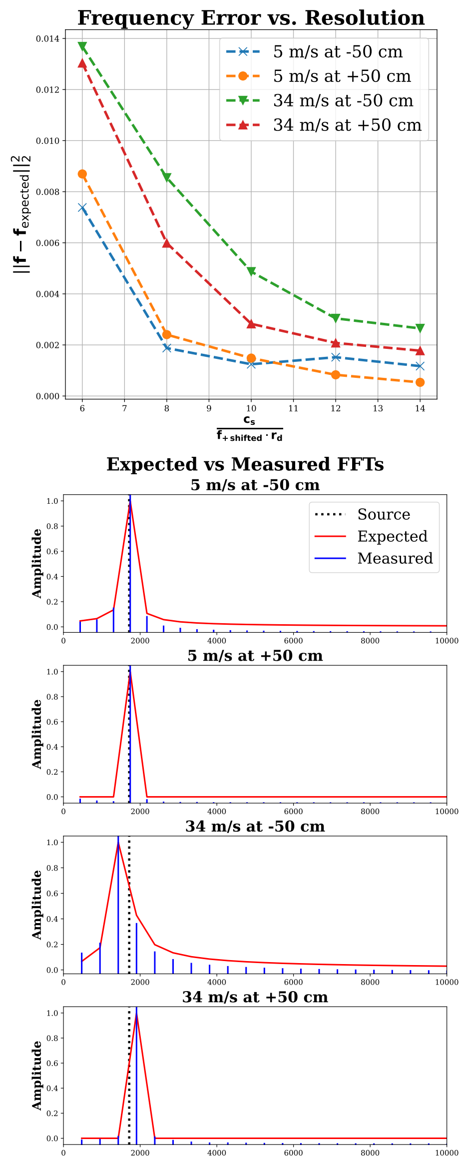


Figure 6.3: Experiment 2 error analysis and FFT plot for our simulated Doppler effect. We included two different sound source speeds in our testing, 5m/s and 34m/s. (Top) Simulation error vs. node resolution $c_s/(f_{+\,\text{shifted}} \cdot r_d)$ where $f_{+\,\text{shifted}}$ is the frequency shift at two observer positions, +50cm and −50cm. (Bottom) Comparison of FFTs for the simulated and expected waveforms at the same two observer positions.

Chapter 7: Performance

In this section, we present an evaluation of our method's performance, focusing on execution time and scalability for practical scenarios.

7.1 General Timings

Here we include three example simulations computed using a Linux desktop workstation with a AMD Ryzen 5800x CPU, 32GB RAM, and an Nvidia RTX 4090 GPU. Two of our examples, the indoor cathedral scene and the outdoor city scene, were designed to be similar to practical real-world scenes like those found in animation products. The results are detailed in Table 6.1. Both examples were simulated for pre-defined lengths, the cathedral scene for six seconds and the city scene for ten seconds, with the time step fixed around 70 microseconds. We note that these problem sizes are large, containing 3,964,536 and 1,856,580 solution nodes respectively. As such the computation times of 24 hours for the cathedral scene and 15 hours for the city scene are within our expectation. At the spatial and temporal resolutions reported in Table 6.1, our resulting audio outputs are of reasonable quality. We apply minimal postprocessing, including only a final low-pass audio filter.

Achieving high-quality audio output within a reasonable time frame necessitates scaling our framework across multiple GPUs. Our scalability experiments indicate the feasibility of such an expansion.

7.2 GPU Scalability

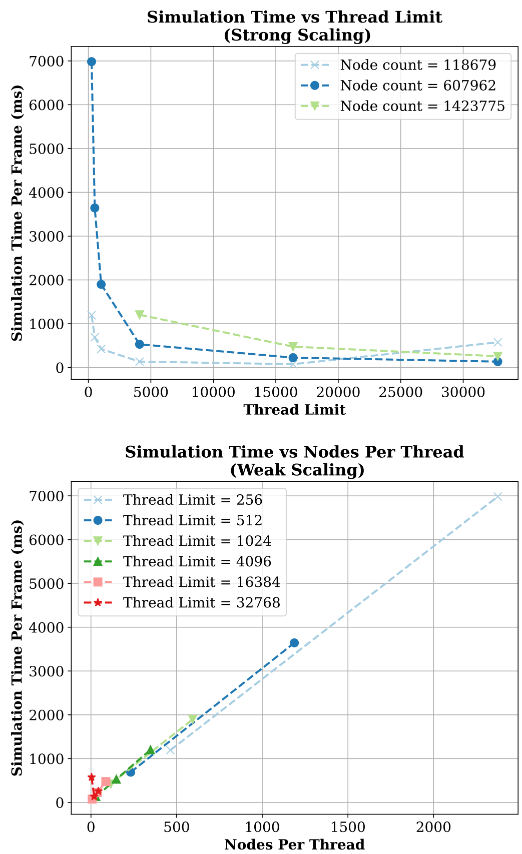


Figure 7.1: Scalability measurements for our framework: (Top) Strong scaling comparing time to simulate against the number of threads. (Bottom) Weak scaling comparing time to simulate against problem size per thread.

To assess the scalability of our proposed framework, we conducted a series of experiments measuring the simulation time for different problem sizes using a range of computational resources. Since our implementation only supports a single GPU so far, we emulate increasing computational resources by limiting the number of threads launched via CUDA. Since the thread count is less than the problem size, multiple nodes are assigned to a single thread implicitly via a grid stride loop. We discuss both strong and weak scaling performance of our framework. The device used to run the scaling simulations is an Nvidia RTX 4090.

Strong Scaling. This metric is evaluated by fixing the total problem size and varying the number of threads. Strong scaling suggests how well a problem can be computed in less time given more resources. Figure 7.1 (top) illustrates the strong performance of our framework in this metric by plotting simulation times vs thread count for various problem sizes. However, as the thread count approaches $\approx$ 16k threads there are noticeable diminishing returns. Beyond this thread count there are no noticeable gains because a hardware limit of the maximum number of resident threads is reached. The diminishing returns seen before this hard limit are expected, as other non-compute tasks such as memory I/O start to have a greater influence over the runtime of each thread. While we employ some known strategies for improving memory task latency, such as pre-fetching and using shared memory blocks, some amount of latency is unavoidable.

Weak Scaling. Contrasting strong scaling, this metric is measured by increasing the problem size proportionally to the number of threads, aiming to maintain a constant workload per thread. Weak scaling suggests how well a problem of a given size and runtime can be executed in the same amount of time at increased scale using additional computational resources to match. Specific to our framework, this metric is important because the simulated domain is limited in size by the available resources of a single GPU. To simulate very large physical domains at an acceptable acoustic fidelity, it is important that our framework scales well according to this metric. Instead of re-running our experiments with problem sizes that are whole number multiples of one another, we evaluate this metric by re-using the data from our strong scaling experiments since the problem sizes span a large range. Figure 7.1 (bottom) plots this data using a separate curve for each thread limit in our experiments. Since all of the curves nearly overlap, we observe that the ratio of compute resources to problem size is approximately 1:1. This indicates that our framework has near-optimal weak scaling.

Chapter 8: Conclusion

We demonstrate effective use of the lattice Boltzmann method (LBM) to simulate acoustic effects in indoor and outdoor 3D scenes. We enable a GPU parallel implementation of our meshless LBM framework that effectively captures complex domain boundaries by formulating the streaming step using nonlocal vector calculus. As shown by our convergence and scalability analysis, this leads to accurate simulations computed in an acceptable amount of time. Our framework accurately reproduces diffraction patterns and wave compressions from benchmark experiments and can incorporate stereoscopic audio using a 3D head model without requiring Head-Related Transfer Functions (HRTFs).

Our framework also has several limitations. The range of resolvable frequencies depends on the spacing of solution nodes. Because of this, simulating sound propagation at higher frequencies can be prohibitively expensive in terms of both space and efficiency, especially in large 3D domains. Even at lower frequencies, our method cannot compete in terms of computation time with the interactive-time performance of some ray-based approaches.

Applying fluid-dynamics to acoustics has seen little representation in the graphics community, and it holds potential for simulating more complex phenomena that are not feasable in existing state-of-the-art techniques. Our future work will focus on enhancing simulation accuracy and scalability, developing an improved parallel implementation distributed across multiple GPUs. The distribution of solution nodes also requires further research, especially around complex boundary geometries. To augment our framework for more dynamic scenes, we also plan to explore moving domain boundaries and varying boundary conditions. We plan to integrate our framework with existing work on acoustics in graphics and animation such as sound generation [70] and sound synthesis [75].

Finally, we plan to develop a complete and systematic evaluation of sound propagation methods, including ray-based, wave-based and physics-based methods, to elucidate the advantages from each approach. Such a study requires both human perception and metric based evaluation techniques. We also plan to compare the effects of numerical methods found in these approaches such as explicit vs. implicit time integration and meshed vs. meshless discretization. We expect that the complexity of graphics applications, including multi-physics and exotic acoustic phenomena, will lead to innovative pairings of modeling approaches and numerical methods.

Bibliography

[1] James Ahrens, Berk Geveci, and Charles Law. *Visualization Handbook*, chapter ParaView: An End-User Tool for Large Data Visualization, pages 717–731. Elsevier Inc., Burlington, MA, USA, 2005.

[2] Bacim Alali and Robert Lipton. Multiscale dynamics of heterogeneous media in the peridynamic formulation. *Journal of Elasticity*, 106:71–103, 2012.

[3] Lakulish Antani, Anish Chandak, Lauri Savioja, and Dinesh Manocha. Interactive sound propagation using compact acoustic transfer operators. *ACM Trans. Graph.*, 31(1), feb 2012.

[4] E Askari, F Bobaru, R B Lehoucq, M L Parks, S A Silling, and O Weckner. Peridynamics for multiscale materials modeling. *Journal of Physics: Conference Series*, 125(1):649–654, 2008.

[5] Jaouad Benhamou, Salaheddine Channouf, and Mohammed Jami. 3d numerical study of sound waves behavior in the presence of obstacles using the d3q15-lattice boltzmann model. In Rajaa Saidi, Brahim El Bhiri, Yassine Maleh, Ayman Mosallam, and Mohammed Essaaidi, editors, *Advanced Technologies for Humanity*, pages 106–115, Cham, 2022. Springer International Publishing.

[6] Jaouad Benhamou, Salaheddine Channouf, Mohammed Jami, Ahmed Mezrhab, Daniel Henry, and Valéry Botton. Three-dimensional lattice boltzmann model for acoustic waves emitted by a source. *International Journal of Computational Fluid Dynamics*, 35(10):850–871, 2021.

[7] Jaouad Benhamou, Mohammed Jami, Ahmed Mezrhab, Valéry Botton, and Daniel Henry. Numerical study of natural convection and acoustic waves using the lattice boltzmann method. *heat Transfer*, 49(6):3779–3796, 2020.

[8] David A. Benson, Stephen W. Wheatcraft, and Mark M. Meerschaert. Application of a fractional advection-dispersion equation. *Water Resources Research*, 36(6):1403–1412, 2000.

[9] Stefan Bilbao. Optimized fdtd schemes for 3-d acoustic wave propagation. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(5):1658–1663, 2012.

[10] Stefan Bilbao and Brian Hamilton. Passive volumetric time domain simulation for room acoustics applications. *The Journal of the Acoustical Society of America*, 145(4):2613–2624, 04 2019.

[11] Johan Augusto Bocanegra, Mario Misale, and Davide Borelli. A systematic literature review on lattice boltzmann method applied to acoustics. *Engineering Analysis with Boundary Elements*, 158:405–429, 2024.

[12] N. Burch, M. D'Elia, and R. B. Lehoucq. The exit-time problem for a markov jump process. *The European Physical Journal Special Topics*, 223(14):3257–3271, 2014.

[13] Chunxiao Cao, Zhong Ren, Carl Schissler, Dinesh Manocha, and Kun Zhou. Interactive sound propagation with bidirectional path tracing. *ACM Trans. Graph.*, 35(6), dec 2016.

[14] Nianzheng Cao, Shiyi Chen, Shi Jin, and Daniel Martínez. Physical symmetry and lattice symmetry in the lattice boltzmann method. *Phys. Rev. E*, 55:R21–R24, Jan 1997.

[15] Shiyi Chen, Zheng Wang, Xiaowen Shan, and Gary D. Doolen. Lattice boltzmann computational fluid dynamics in three dimensions. *Journal of Statistical Physics*, 68:379–400, 1992.

[16] Corey I. Cheng and Gregory H. Wakefield. Introduction to head-related transfer functions (hrtfs): Representations of hrtfs in time, frequency, and space. *J. Audio Eng. Soc*, 49(4):231–249, 2001.

[17] Bastien Chopard, Alexandre Dupuis, Alexandre Masselot, and Pascal Luthi. Cellular automata and lattice boltzmann techniques: An approach to model and simulate complex systems. *Advances in Complex Systems*, 05(02n03):103–246, 2002.

[18] P. V. Coveney, S. Succi, Dominique d'Humières, Irina Ginzburg, Manfred Krafczyk, Pierre Lallemand, and Li-Shi Luo. Multiple-relaxation-time lattice boltzmann models in three dimensions. *Philosophical Transactions of the*

Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences, 360(1792):437–451, 2002.

[19] Didier Dragna, Philippe Blanc-Benon, and Franck Poisson. Time-domain solver in curvilinear coordinates for outdoor sound propagation over complex terrain. *The Journal of the Acoustical Society of America*, 133(6):3751–3763, 06 2013.

[20] Qiang Du, Bjorn Engquist, and Xiaochuan Tian. Multiscale modeling, homogenization and nonlocal effects: Mathematical and computational issues. *Contemporary mathematics*, 754:115–140, 2020.

[21] Qiang Du, Max Gunzburger, R. B. Lehoucq, and Kun Zhou. A nonlocal vector calculus, nonlocal volume-constrained problems, and nonlocal balance laws. *Mathematical Models and Methods in Applied Sciences*, 23(03):493–540, 2013.

[22] Marta D'Elia, Qiang Du, Max Gunzburger, and Richard Lehoucq. Nonlocal convection-diffusion problems on bounded domains and finite-range jump processes. *Computational Methods in Applied Mathematics*, 17(4):707–722, 2017.

[23] Martin Geier, Andreas Greiner, and Jan G. Korvink. Cascaded digital lattice boltzmann automata for high reynolds number flow. *Phys. Rev. E*, 73:066705, Jun 2006.

[24] Mohsen Gorakifard, Clara Salueña, Ildefonso Cuesta, and Ehsan Kian Far. The meshless local petrov–galerkin cumulant lattice boltzmann method: Strengths and weaknesses in aeroacoustic analysis. *Acta mechanica*, 233(4):1467–1483, 2022.

[25] John Gounley, Madhurima Vardhan, Erik W. Draeger, Pedro Valero-Lara, Shirley V. Moore, and Amanda Randles. Propagation pattern for moment representation of the lattice boltzmann method. *IEEE Transactions on Parallel and Distributed Systems*, 33(3):642–653, 2022.

[26] Harold Grad. Note on n-dimensional hermite polynomials. *Communications on Pure and Applied Mathematics*, 2(4):325–330, 1949.

[27] Harold Grad. The profile of a steady plane shock wave. *Communications on Pure and Applied Mathematics*, 5(3):257–300, 1952.

[28] Nail A. Gumerov and Ramani Duraiswami. A broadband fast multipole accelerated boundary element method for the three dimensional Helmholtz equation. *The Journal of the Acoustical Society of America*, 125(1):191–205, 01 2009.

[29] Max Gunzburger and R. B. Lehoucq. A nonlocal vector calculus with application to nonlocal boundary value problems. *Multiscale Modeling & Simulation*, 8(5):1581–1598, 2010.

[30] Brian Hamilton and Stefan Bilbao. Fdtd methods for 3-d room acoustics simulation with high-order accuracy in space and time. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(11):2112–2124, 2017.

[31] Xiaoyi He and Li-Shi Luo. A priori derivation of the lattice boltzmann equation. *Phys. Rev. E*, 55:R6333–R6336, Jun 1997.

[32] Xiaoyi He, Li-Shi Luo, and Micah Dembo. Some progress in lattice boltzmann method. part i. nonuniform mesh grids. *Journal of Computational Physics*, 129(2):357–363, 1996.

[33] Amit Katiyar, Shivam Agrawal, Hisanao Ouchi, Pablo Seleson, John T. Foster, and Mukul M. Sharma. A general peridynamics model for multiphase transport of non-newtonian compressible fluids in porous media. *Journal of Computational Physics*, 402:109075, 2020.

[34] Mazen Kharbutli, Keith Irwin, Yan Solihin, and Jaejin Lee. Using prime numbers for cache indexing to eliminate conflict misses. In *10th International Symposium on High Performance Computer Architecture (HPCA'04)*, pages 288–299. IEEE, 2004.

[35] Alexander Klassen, Thorsten Scharowsky, and Carolin Körner. Evaporation model for beam based additive manufacturing using free surface lattice boltzmann methods. *Journal of Physics D: Applied Physics*, 47(27):275303, jun 2014.

[36] Tim Krüger, Halim Kusumaatmaja, Alexandr Kuzmin, Orest Shardt, Goncalo Silva, and Erland Magnus Viggen. *The Lattice Boltzmann Method*. Springer, 1 edition, 2016.

[37] Jonas Latt, Christophe Coreixas, Joël Beny, and Andrea Parmigiani. Efficient supersonic flow simulations using lattice boltzmann methods based on

numerical equilibria. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 378(2175):20190559, 2020.

[38] Hwi Lee and Qiang Du. Asymptotically compatible sph-like particle discretizations of one dimensional linear advection models. *SIAM Journal on Numerical Analysis*, 57(1):127–147, 2019.

[39] Taehun Lee and Ching-Long Lin. A characteristic galerkin method for discrete boltzmann equation. *Journal of Computational Physics*, 171(1):336–356, 2001.

[40] Wei Li, Yixin Chen, Mathieu Desbrun, Changxi Zheng, and Xiaopei Liu. Fast and scalable turbulent flow simulation with two-way coupling. *ACM Transactions on Graphics*, 39(4):Art–No, 2020.

[41] Zheng Li, Mo Yang, and Yuwen Zhang. Lattice boltzmann method simulation of 3-d natural convection with double mrt model. *International Journal of Heat and Mass Transfer*, 94:222–238, 2016.

[42] Shiguang Liu and Dinesh Manocha. Sound synthesis, propagation, and rendering: A survey. *CoRR*, abs/2011.05538, 2020.

[43] Steen Magnussen, Erkki Tomppo, and Ronald E McRoberts. A model-assisted k-nearest neighbour approach to remove extrapolation bias. *Scandinavian Journal of Forest Research*, 25(2):174–184, 2010.

[44] Nicos S. Martys and Hudong Chen. Simulation of multicomponent fluids in complex three-dimensional geometries by the lattice boltzmann method. *Phys. Rev. E*, 53:743–750, Jan 1996.

[45] Guy R. McNamara and Gianluigi Zanetti. Use of the boltzmann equation to simulate lattice-gas automata. *Phys. Rev. Lett.*, 61:2332–2335, Nov 1988.

[46] Ravish Mehra, Nikunj Raghuvanshi, Lakulish Antani, Anish Chandak, Sean Curtis, and Dinesh Manocha. Wave-based sound propagation in large open scenes using an equivalent source formulation. *ACM Trans. Graph.*, 32(2), apr 2013.

[47] Ravish Mehra, Nikunj Raghuvanshi, Lauri Savioja, Ming C. Lin, and Dinesh Manocha. An efficient gpu-based time domain solver for the acoustic wave equation. *Applied Acoustics*, 73(2):83–94, 2012.

[48] Ralf Metzler and Joseph Klafter. The restaurant at the end of the random walk: recent developments in the description of anomalous transport by fractional dynamics. *Journal of Physics A: Mathematical and General*, 37(31):R161, 2004.

[49] S. Hossein Musavi and Mahmud Ashrafizaadeh. Meshless lattice boltzmann method for the simulation of fluid flows. *Phys. Rev. E*, 91:023310, Feb 2015.

[50] A Najafi-Yazdi and L Mongeau. An absorbing boundary condition for the lattice boltzmann method based on the perfectly matched layer. *Computers & fluids*, 68:203–218, 2012.

[51] Octavio Navarro-Hinojosa, Sergio Ruiz-Loza, and Moisés Alencastre-Miranda. Physically based visual simulation of the lattice boltzmann method on the gpu: a survey. *The Journal of Supercomputing*, 74:3441–3467, 2018.

[52] Takuya Oshima, Masashi Imano, Yasuhiro Hiraguri, and Yoshikazu Kamoshida. Linearized euler simulations of sound propagation with wind effects over a reconstructed urban terrain using digital geographic information. *Applied Acoustics*, 74(12):1354–1366, 2013.

[53] G. Pavić. A technique for the computation of sound radiation by vibrating bodies using multipole substitute sources. *Acta Acustica united with Acustica*, 92(1):112–126, 2006.

[54] Gilles Pijaudier-Cabot and Zdeněk P. Bažant. Nonlocal damage theory. *Journal of Engineering Mechanics*, 113(10):1512–1533, 1987.

[55] Nikunj Raghuvanshi, Rahul Narain, and Ming C. Lin. Efficient and accurate sound propagation using adaptive rectangular decomposition. *IEEE Transactions on Visualization and Computer Graphics*, 15(5):789–801, 2009.

[56] Nikunj Raghuvanshi and John Snyder. Parametric wave field coding for precomputed sound propagation. *ACM Trans. Graph.*, 33(4), jul 2014.

[57] Nikunj Raghuvanshi, John Snyder, Ravish Mehra, Ming Lin, and Naga Govindaraju. Precomputed wave simulation for real-time sound propagation of dynamic sources in complex scenes. In *ACM SIGGRAPH 2010 Papers*, SIGGRAPH '10, New York, NY, USA, 2010. Association for Computing Machinery.

[58] Matthew Rosen, Keith W. Godin, and Nikunj Raghuvanshi. Interactive sound propagation for dynamic scenes using 2d wave simulation. *Computer Graphics Forum*, 39(8):39–46, 2020.

[59] Atul Rungta, Carl Schissler, Nicholas Rewkowski, Ravish Mehra, and Dinesh Manocha. Diffraction kernels for interactive sound propagation in dynamic environments. *IEEE Transactions on Visualization and Computer Graphics*, 24(4):1613–1622, 2018.

[60] Reza Sadeghi, Mostafa Safdari Shadloo, M Hopp-Hirschler, Abdellah Hadjadj, and Ulrich Nieken. Three-dimensional lattice boltzmann simulations of high density ratio two-phase flows in porous media. *Computers & Mathematics with Applications*, 75(7):2445–2465, 2018.

[61] Shinichi Sakamoto, Ayumi Ushiyama, and Hiroshi Nagatomo. Numerical analysis of sound propagation in rooms using the finite difference time domain method. *The Journal of the Acoustical Society of America*, 120(5):3008–3008, 2006.

[62] Erik M Salomons, Walter JA Lohman, and Han Zhou. Simulation of sound waves using the lattice boltzmann method for fluid flow: Benchmark cases for outdoor sound propagation. *PloS one*, 11(1):1–19, 2016.

[63] Carl Schissler and Dinesh Manocha. Interactive sound propagation and rendering for large multi-source scenes. *ACM Trans. Graph.*, 36(4), jul 2017.

[64] Rina Schumer, David A. Benson, Mark M. Meerschaert, and Boris Baeumer. Multiscaling fractional advection-dispersion equations and their solutions. *Water Resources Research*, 39(1):1022–1032, 2003.

[65] M. Shaat. A general nonlocal theory and its approximations for slowly varying acoustic waves. *International Journal of Mechanical Sciences*, 130:52–63, 2017.

[66] Xiaowen Shan, Xue-Feng Yuan, and Hudong Chen. Kinetic theory representation of hydrodynamics: a way beyond the navier–stokes equation. *Journal of Fluid Mechanics*, 550:413–441, 2006.

[67] Xing Shi, Jianzhong Lin, and Zhaosheng Yu. Discontinuous galerkin spectral element lattice boltzmann method on triangular element. *International Journal for Numerical Methods in Fluids*, 42(11):1249–1261, 2003.

[68] Sigurd Johannes Benedikt Stoll. Lattice boltzmann simulation of acoustic fields, with special attention to non-reflecting boundary conditions. Master's thesis, Institutt for elektronikk og telekommunikasjon, 2014.

[69] Lonny L. Thompson. A review of finite-element methods for time-harmonic acoustics. *The Journal of the Acoustical Society of America*, 119(3):1315–1330, 03 2006.

[70] Jui-Hsien Wang, Ante Qu, Timothy R. Langlois, and Doug L. James. Toward wave-based sound synthesis for computer animation. *ACM Trans. Graph.*, 37(4), jul 2018.

[71] Ke Wang, Yi Chen, Muamer Kadic, Ghangguo Wang, and Martin Wegener. Nonlocal interaction engineering of 2d roton-like dispersion relations in acoustic and mechanical metamaterials. *Communications Materials*, 3:35–, 2022.

[72] Haowen Xi, Gongwen Peng, and So-Hsiang Chou. Finite-volume lattice boltzmann method. *Phys. Rev. E*, 59:6202–6205, May 1999.

[73] Ao Xu, TS Zhao, Liang An, and Le Shi. A three-dimensional pseudo-potential-based lattice boltzmann model for multiphase flows with large density ratio and variable surface tension. *International Journal of Heat and Fluid Flow*, 56:261–271, 2015.

[74] Wu-Wen Yao, Xiao-Ping Zhou, and Qi-Hu Qian. From statistical mechanics to nonlocal theory. *Acta Mechanica*, 233(3):869–887, 2022.

[75] Jinta Zheng, Shih-Hsuan Hung, Kyle Hiebel, and Yue Zhang. Real-time rendering of decorative sound textures for soundscapes. *ACM Trans. Graph.*, 39(6), nov 2020.

APPENDICES

Appendix A: Things

Appendix B: More Things