CRUNA – AN OPEN-SOURCE FINITE-DIFFERENCE TIME-DOMAIN SIMULATION AND OPTIMIZATION FRAMEWORK BASED ON ADJOINT GRADIENTS

Arne B. Hölter[1,2], Yannick Schubert[2], Lewin Stein[3], Mario Sroka[2,4], Mathias Lemke[2]

[1] *Audio Communication Group, Technische Universität Berlin, Einsteinufer 17, 10587 Berlin, Germany.*
[2] *Numerical Fluid Dynamics Group, Technische Universität Berlin, Müller-Breslau-Straße 15, 10623 Berlin, Germany.*
[3] *Parallel and Distributed Systems, Supercomputing Group, Zuse Institut Berlin, Takustraße 7, 14195 Berlin, Germany.*
[4] *Department of Engineering Acoustics, Technische Universität Berlin, Einsteinufer 25, 10587 Berlin, Germany.*

This paper is a revised version of the authors' submitted manuscript that was peer-reviewed by at least two anonymous reviewers.

Abstract

Wave-based simulation methods are well suited for analyzing acoustic problems in which wave phenomena such as diffraction and scattering play a dominant role. Among these, the finite-difference time-domain (FDTD) method is a robust and versatile approach. This work presents CRUNA, an open-source 3D FDTD solver with an integrated optimization framework based on adjoint gradient computation. While the modular codebase addresses a wide variety of flow-related problems, this contribution focuses on the acoustic and aeroacoustic components of the solver and their application to wave propagation and optimization tasks.

The solver supports several sets of governing equations, ranging from linearized acoustic formulations to more comprehensive aeroacoustic models. It provides explicit time-stepping schemes, as well as explicit and implicit spatial discretization methods. CRUNA is highly parallelized to enable fast execution in high-performance computing environments. Non-reflective characteristic boundary conditions are implemented, and immersed boundary methods based on effective volume and flow resistivity enable an accurate representation of complex geometries and impedance boundaries.

Representative examples demonstrate the framework, including aeroacoustic simulations, sound source localization and synthesis, and pinna-related transfer function computations. By providing CRUNA as open-source software, this work aims to support transparency, reproducibility, and further development within the computational acoustics community.

Corresponding author: *hoelter@campus.tu-berlin.de.*

Keywords: *finite differences, numerical acoustics, open-source software, adjoint-based optimization*

1. Introduction

Numerical simulations play an essential role in acoustics. They are used in a wide range of applications, from fundamental research to practical engineering applications. The most extensively studied approaches include the finite element method (FEM) [1], the boundary element method (BEM) [2], and the finite-difference time-domain (FDTD) method [3].

While commercial software packages, e.g., COMSOL[1] or Treble[2], provide robust environments for acoustic simulations, there remains a need for open-source solvers that provide greater transparency. Exemplarily, openFOAM [4] is a widely used finite volume (FVM) solver in aeroacoustics. For purely acoustic problems, the GPU-based FDTD solver PFFDTD [5] is primarily used for room and virtual acoustics. However, they are all designed to perform direct (forward) simulations. This is sufficient for evaluating predefined setups. For the inverse task, e.g., finding an optimal setup with respect to a predefined target, complex pipelines with appropriate optimization methods must be built. In ray-based acoustics, a differentiable room acoustic renderer with an inherent optimization algorithm was recently presented [6].

This paper presents CRUNA[3] (compressible reactive unsteady Navier–Stokes adjoint), an open-source FDTD solver for acoustic and aeroacoustic simulations. It includes an inverse optimization pipeline based on the computation of adjoint gradients. Therefore, it covers a wide range of acoustic applications.

2. Numerical Methods

This section presents the governing equations, the spatial and temporal discretization, and the non-reflective

[1] https://www.comsol.com/ [2] https://www.treble.tech/
[3] https://github.com/cruna-toolkit

boundary conditions. Owing to the modular structure of the code, the framework can be readily extended by additional methods or terms on the right-hand side representing boundary conditions or acoustic sources.

2.1. Governing equations

Depending on the intended application and effects to be studied, different sets of governing equations are required. For the sake of brevity, only the direct formulations are discussed here. For aeroacoustic simulations, the full compressible volume-penalized (VP) Navier-Stokes equations [7] are

$$\Phi\partial_t(\rho)+\partial_{x_i}(\Phi\rho u_i)=0 \quad (1)$$

$$\Phi\partial_t(\rho u_j)+\partial_{x_i}(\Phi\rho u_i u_j)+\Phi\partial_{x_j}(p)=\Phi\chi(u_j^{\text{solid}}-u_j) + \partial_{x_i}(\Phi\tau_{ij}) \quad (2)$$

$$\Phi\partial_t(\rho e)+\partial_{x_i}(\Phi u_i(\rho e+p))=\partial_{x_i}(\Phi\lambda\partial_{x_i}(T)) + \partial_{x_j}(\Phi\tau_{ij}u_i) \quad (3)$$

where ρ is the density, u_i the velocity components in the three spatial directions $i,j=1,2,3$, p the pressure, and e the total energy; the Einstein summation convention applies. For brevity, we use $\partial_t=\frac{\partial(\cdot)}{\partial t}$ and $\partial_{x_i}=\frac{\partial(\cdot)}{\partial x_i}$. Φ denotes the porosity, i.e., the fraction of the total volume accessible to the fluid phase [8], while the Darcy-like term χ represents the flow resistivity. The velocity inside the modeled material, u_j^{solid}, is taken to be zero in the following. For purely acoustic problems, the terms in blue can be excluded to obtain the nonlinear Euler equations. Therein, λ is the heat diffusion coefficient, T the temperature, and τ_{ij} the stress tensor. If no flow is present, the amplitudes are small and the atmospheric conditions are constant, Eqs. (1)–(3) can be reduced to [3]

$$\partial_t\left(p'\right)+\frac{\rho_0 c_0^2}{\Phi}\partial_{x_j}\left(\Phi u_j'\right)=0 \quad (4)$$

$$\partial_t\left(u_j'\right)+\frac{1}{\rho_0}\partial_{x_j}\left(p'\right)=-\frac{1}{\rho_0}\chi u_j' \quad (5)$$

where c_0 denotes the speed of sound, ρ_0 is the atmospheric density, and the continuity equation is written in terms of the acoustic pressure p'.

2.2. Spatial and temporal discretization

Spatial derivatives in the governing equations are approximated using high-order explicit and implicit finite-difference schemes. Further details, as well as the dispersion and dissipation properties of the schemes, are provided in [3, Sec. 2.2 and Appendix]. Temporal integration can either be carried out by a standard or a low-dispersion, low-dissipation Runge-Kutta scheme of fourth order [9]. Spatial filters of up to tenth order are implemented to suppress spurious waves inherent to the FDTD approach [10].

2.3. Boundary conditions

Boundary conditions are treated in two ways. First, non-reflective boundary conditions (NRBCs) can be applied at the domain boundaries or within a designated boundary region. Local and zonal characteristic boundary conditions (CBCs) are implemented [11]. When local CBCs are employed, an additional sponge layer [12] may be used to improve the NRBC. For impedance-type boundary conditions, such as rigid walls or porous absorbers, the effective volume Φ and the Darcy term χ can be used (see Eqs. (1)–(5)) [8]. A detailed parameter study is presented in [7]. The effective volume Φ requires a smooth transition from air to the obstacle via blending functions. Consequently, a minimum number of grid points, i.e., a finite obstacle thickness, is required. An estimator for this requirement is given in [13].

3. Code structure

The numerical framework is implemented in a modular manner and is primarily written in Fortran 95, with a design that ensures upward compatibility with more recent Fortran standards. It follows a layered architecture in which a generic core provides the main solver infrastructure, while problem-specific components are supplied through user-defined extensions. This separation enables systematic reuse of numerical kernels and facilitates the implementation of new simulation and optimization cases.

Dedicated modules implement the spatial derivatives and metric transformations for the spatial discretization. The time-stepping routines are encapsulated in separate modules and interact with the governing equations through standardized interfaces, enabling straightforward substitution or extension of integration schemes. The numerical filtering is implemented in modules that operate independently in each spatial direction and can be activated at user-defined frequencies to ensure stability in long-time simulations. All modules are selected at compile time through wrapper modules.

The solver incorporates an adjoint-based optimization framework for gradient-based control and design. The framework consists of modules for objective function evaluation, adjoint time integration, gradient computation, and line search procedures. Forward and adjoint solvers share the same discretization and filtering infrastructure, ensuring discrete consistency. An outer optimization loop controls the sequence of direct simulation, adjoint computation, gradient evaluation, and update, enabling efficient large-scale optimization on distributed-memory systems. Figure 1 shows the flowchart, including the direct part in light blue, the adjoint part in light red, and additional components for the adjoint analysis with white background.

Input/output operations are handled by a unified I/O layer based on HDF5[4]. Simulation parameters, field data, and optimization results are stored in structured datasets to support efficient parallel access and post-processing. Auxiliary Python and MATLAB scripts are provided for data reconstruction and visualization. To reduce memory requirements in adjoint simula-

[4] https://www.hdfgroup.org/solutions/hdf5/

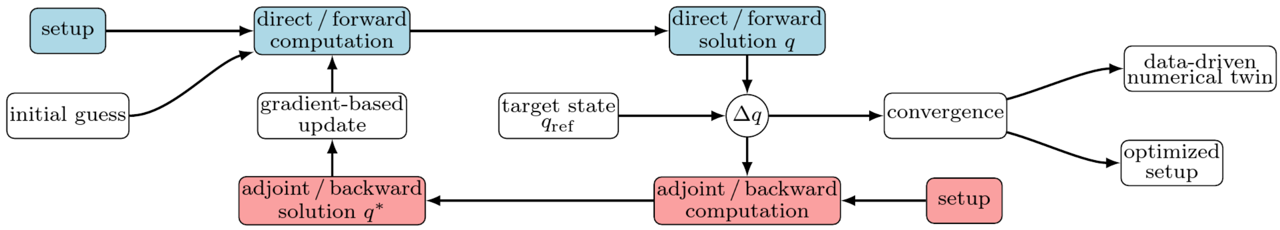

Figure 1: Sketch of CRUNA's flowchart. The components for a direct / forward simulation are highlighted in light blue and can be used independently. Components for solely adjoint / backward simulation are marked in light red and can likewise be employed in a standalone manner. For optimization tasks as well as adjoint-based analyses such as the construction of numerical twins, both light blue and light red as well as the components with white background are used. The variable q comprises the primitive variables of the respective governing equations.

tions, the solver supports subset-based check-pointing, in which forward-state data are written to disk and reloaded during the backward computation.
Problem-specific configurations are implemented through a case-dependent layer that overrides or extends selected core modules. These extensions are compiled together with the core source tree, allowing customization without modification of the main solver.
The framework integrates several external libraries to enhance computational performance and portability. Linear algebra operations for implicit schemes are accelerated using OpenBLAS[5] or Intel MKL[6]. Parallelization is implemented using MPI and OpenMP, enabling efficient execution on modern high-performance computing platforms.
An MPI-based three-dimensional domain decomposition is employed for parallelization, as illustrated in Figure 2(a). Within this framework, data exchange between neighboring sub-blocks is required for the evaluation of spatial derivatives and filtering operations. Since compact finite-difference schemes and filters inherently require access to all grid points along the respective coordinate direction, the computational domain is temporarily rearranged using a pencil decomposition, as shown in Figure 2(b).
During this stage, inter-process communication is performed using collective operations based on `MPI_Alltoallv` within suitable processor sub-groups, rather than point-to-point communication. This approach provides high communication efficiency, particularly on systems with optimized collective communication hardware. The resulting parallelization exhibits competitive performance with nearly linear scaling behavior of up to more than 6000 cores, see Figure 3. This performance level is consistent with the current state of the art for collective-communication-based parallel solvers [14]. The scaling experiments were conducted on the CLX partition of NHR@ZIB[7].

4. Examples

This section presents brief examples of direct simulations for aeroacoustic wave propagation and virtual acoustics (Secs. 4.1, 4.2), adjoint-based source identification (Sec. 4.3), and adjoint-based optimization for source synthesis (Sec. 4.4). For detailed descriptions, analysis, and discussion, the reader is referred to the cited references.

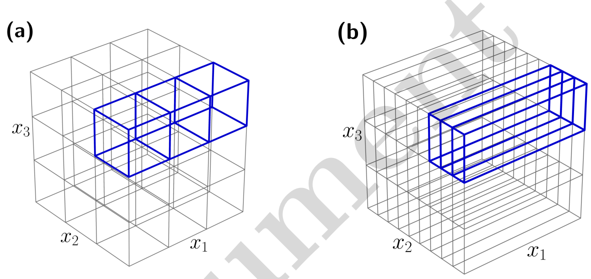

Figure 2: **a** Topological Cartesian domain partitioning for MPI parallelization. **b** Temporary topology for efficient application of implicit schemes; blue processes communicate collectively.

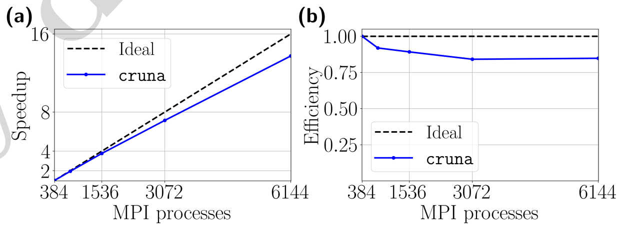

Figure 3: **a** Strong scaling with fixed total grid size, showing nearly linear behavior. **b** Weak scaling with fixed grid size per process, showing a moderate reduction in parallel efficiency. Reproduced from [13] under CC BY 4.0.

4.1. Aeroacoustic simulation of bluff bodies

This example demonstrates CRUNA's capability for direct acoustic and aeroacoustic time-domain simulations, focusing on a canonical bluff-body configuration with porous treatments [7]. The underlying solver advances the full compressible Navier–Stokes equations Eqs. (1)–(3) on a distorted grid using finite-difference discretizations, enabling the simultaneous resolution of unsteady flow features and the resulting acoustic field within a unified framework. Geometries are represented using the immersed boundary formulation that combines a Brinkman-type VP approach with an effective volume representation (Sec. 2.1).
The method is applied to the aeroacoustic sound gen-

[5] https://github.com/OpenMathLib/OpenBLAS
[6] https://www.intel.com/mkl [7] https://nhr.zib.de

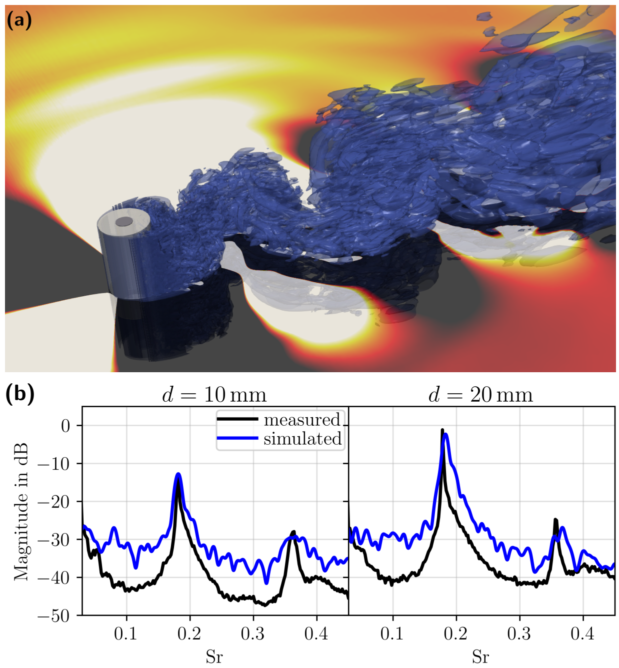

Figure 4: **a** Snapshot illustrating the simulation of a coated cylinder. The visualization of acoustic waves by the summation of the density gradient in the x_1–x_2 plane is in yellow to red, while vortex structures are depicted using an iso-surface of the Q-criterion in blue. **b** Frequency responses of measured and simulated cylinder coated with Basotect for two different thicknesses d, shown as the magnitude of the sound pressure level (SPL) versus the Strouhal number $\mathrm{Sr} = fD/u$, where f is the frequency, D the cylinder coating diameter, and u the inflow velocity. Adapted from [7] under CC BY-NC 4.0.

erated by flow around a porous-coated cylinder at a Reynolds number of $Re = 99000$. The porous coating is modeled by prescribing physically motivated material properties (flow resistivity and porosity) through the VP formulation, linking numerical parameters directly to measurable characteristics of the porous medium. The simulations are performed in three dimensions and consider two different porous coatings, each evaluated at two coating thicknesses. Figure 4(b) illustrates coupled aeroacoustic physics and shows strong agreement between simulation and experiment for different Basotect coating thicknesses. By accurately modeling these realistic porous materials, the framework provides a validated, predictive workflow for aeroacoustic parameter studies.

4.2. Pinna-related transfer functions

As an example of virtual acoustics, pinna-related transfer functions (PRTFs) are simulated and compared with measurements of the 3D-printed counterpart [15] in Hölter et al. [13]. The mesh of the pinna is shown in Figure 5(a). The wave propagation is modeled using Eqs. (4)–(5), while the pinna geometry is represented via VP, with $\Phi = 10^{-3}$ prescribed in the penalized region defined by a blending function. The source is placed in the ear canal and excited with an initial Gaussian distribution. On a sphere around the pinna with a radius of 0.07 m, 770 receiver points on a Lebedev grid are defined and used to extrapolate the results to the measurement positions at a distance of 1 m by spherical harmonics. Figure 5(b) shows exemplary frequency responses of three out of ten measurement positions for the azimuth $\theta = [0, 45, 90]^\circ$ and the elevation $\varphi = 0^\circ$. For the analysis, we define an absolute error

$$\varepsilon = \left| 20 \log_{10} \left(\frac{|P|}{|P_{\mathrm{ref}}|} \right) \right| \tag{6}$$

where P denotes the simulated frequency response and P_{ref} the measured frequency response. In a frequency range from 0.35 to 20 kHz, the arithmetic mean error is $\varepsilon_{\mathrm{mean}} = 1.36\,\mathrm{dB}$ for a grid spacing of $\Delta x = 0.1\,\mathrm{mm}$ and $\varepsilon_{\mathrm{mean}} = 2.16\,\mathrm{dB}$ for $\Delta x = 0.2\,\mathrm{mm}$. The use of blending functions for the VP allows for the smooth representation of complex geometries without grid adaptations. However, a certain number of grid points is required to represent thin structures. In this case, the thinnest structures have a thickness of approximately 1 mm, which can lead to incorrect notch frequencies, e.g., at $\theta = 90^\circ$ and $f = 10\,\mathrm{kHz}$ for $\Delta x = 0.2\,\mathrm{mm}$. Overall, the results show the applicability of the solver to simulate acoustically complex structures. Further details and discussion can be found in [13].

4.3. Sound source localization

Another application is the identification and localization of acoustic sources. For this purpose, a single adjoint computation is sufficient. The source terms in the adjoint equations are given by measurement signals from a typical microphone-array setup. The local maxima of the aggregated adjoint sensitivities indicate the spatial regions with the greatest influence on the objective function and thus correspond to the most likely source locations. This method is described in detail in [16].

Figure 6 presents an example of acoustic source identification. Figure 6(a) illustrates the setup, with a 64-microphone spiral array (red circles) positioned at $x_3 = 0\,\mathrm{m}$ and four acoustic sources (blue crosses) located at $x_3 = 0.75\,\mathrm{m}$, which emit white noise to generate synthetic measurements at the microphone positions. This scenario represents a well-defined benchmark case, and the corresponding measurement data are publicly available [17].

Figure 6(b) shows the sensitivity distribution in the source plane after $n_t = 400$ adjoint time steps. The four sources are clearly identifiable and correctly localized. The adjoint approach can be used as an alternative to conventional beamforming methods. Owing to the Euler or Navier–Stokes equations acting as physical constraints, the adjoint-based formulation is less sensitive to measurement noise, small microphone-positioning errors, and deviations in the assumed speed of sound caused by temperature variations. Since the adjoint method requires only a small number

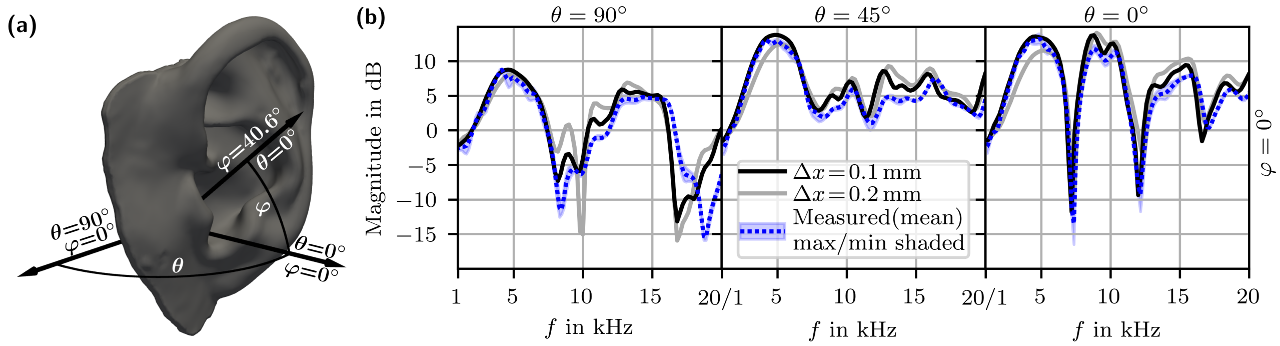

Figure 5: a Sketch of the pinna with angular orientation. **b** Exemplary frequency responses of simulations and acoustic measurements of the pinna. Adapted from [13] under CC BY 4.0.

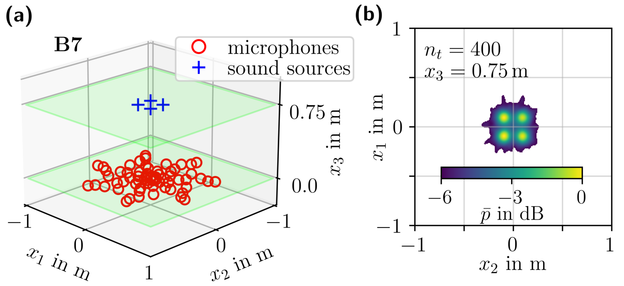

Figure 6: a Basic arrangement for B7 benchmark case. **b** Map of the normalized adjoint-based sensitivity $\bar{p}$ computed for all $n_T = 500$ time steps. Logarithmic color scaling with a range of 6 dB.

of measurement samples, rapidly moving sources can be reliably identified [18].

Furthermore, using the volume penalization method enables source identification in the presence of complex boundaries, allowing for the localization of sources even without a direct line of sight between the sources and the microphones [18].

4.4. Sound source synthesis

Finally, the full optimization loop of Figure 1 is used to reconstruct the directivity of a bassoon. It is an instrument with a particularly complex radiation behavior due to its extended geometry and multiple sound-radiating openings [19].

Figure 7(a) shows the setup. The bassoon is positioned at the center of a spherical microphone array with a radius of 2 m. The experimental reference dataset consists of 2522 microphone positions uniformly distributed on a spherical grid. For the analysis, subsets of 256 and 64 microphones are selected from this dataset to evaluate the performance of the method under reduced spatial sampling conditions.

In the numerical model, a spherical source region with a radius of $r_{\mathrm{sp}} = 0.6$ m is defined around the instrument, corresponding to approximately 40,000 degrees of freedom in the monopole synthesis. Within this region, energy (pressure) source terms are iteratively optimized using an adjoint-based approach constrained by the compressible Euler equations. This monopole synthesis yields a physically consistent numerical twin of the bassoon. This numerical twin reproduces the measured microphone signals while inherently satisfying the governing wave propagation physics, enabling high-resolution extraction of the instrument's directional characteristics.

The reconstructed directivities for the fourth overtone at 624 Hz are shown in Figure 7(c,d), and the reference is depicted in (b). The balloon plots illustrate the results obtained using 256 and 64 microphones, respectively. With 256 microphones, the dominant radiation lobes and the pronounced spatial asymmetry of the bassoon are accurately reproduced. When reducing the number of microphones to 64, the main structural features of the radiation pattern remain clearly identifiable. However, local deviations and smoothing effects become more pronounced as the spatial sampling density decreases.

This example demonstrates that the adjoint-based monopole synthesis framework can generate a physics-consistent numerical twin of a complex musical instrument and reliably reconstruct its multi-lobed directivity pattern even under sparse measurement conditions. Further details and discussion can be found in [19].

5. Conclusion

A FDTD framework for solving forward and inverse problems in acoustics and aeroacoustics has been presented, with emphasis on the implemented numerical methods, the code structure, and four representative application examples. The examples demonstrate the versatility of the framework and include aeroacoustic simulations, the computation of pinna-related transfer functions, sound source localization, and sound source synthesis. Beyond the cases discussed here, the identification of porous absorbers [20] and the adaptation of source driving functions under harsh environmental conditions [21] further highlight the potential of the framework for applications in room acoustics and sound reinforcement.

In addition to acoustical applications, the framework will be extended to flow-related problems in fluid mechanics, as well as to high-frequency applications such as ultrasonic wave propagation.

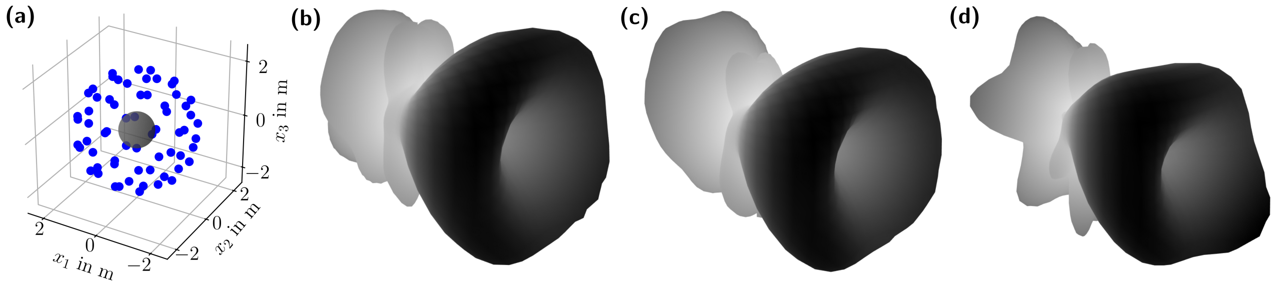

Figure 7: **a** Sketch of the measurement and reconstruction setup. The grey sphere represents the source region used for the monopole synthesis, and the blue dots indicate the microphone positions (illustrated here for 64 microphones). **b** Reference directivity of the bassoon at 624 Hz. **c** Reconstruction of **b** using 256 microphones. **d** Reconstruction of **b** using 64 microphones. Panels **b**–**d** are reproduced from [19] under CC BY-NC-ND 4.0.

Acknowledgments

The authors gratefully acknowledge the financial support provided by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under grant numbers 504470212 and 532453154. They also acknowledge the computing time made available to them on the high-performance computer "Lise" at the NHR@ZIB center. This center is jointly supported by the Federal Ministry of Education and Research and the participating state governments.

References

[1] L. L. Thompson, "A review of finite-element methods for time-harmonic acoustics," *J. Acoust. Soc. Am.*, vol. 119, no. 3, pp. 1315–1330, 2006. DOI: 10.1121/1.2164987.

[2] S. Preuss, C. Gurbuz, C. Jelich, S. K. Baydoun, and S. Marburg, "Recent advances in acoustic boundary element methods," *J. Theor. Comput. Acoust.*, vol. 30, no. 3, p. 2 240 002, 2022. DOI: 10.1142/S2591728522400023.

[3] A. B. Hölter, S. Weinzierl, and M. Lemke, "Finite difference time domain discretization for room acoustic simulation based on the non-linear Euler equations," *Acta Acust.*, vol. 8, no. 75, pp. 1–13, 2024. DOI: 10.1051/aacus/2024071.

[4] H. Jasak, "OpenFOAM: Open source CFD in research and industry," *Int. J. Nav. Archit. Ocean Eng.*, vol. 1, no. 2, pp. 89–94, 2009. DOI: 10.2478/IJNAOE-2013-0011.

[5] J. Saarelma and L. Savioja, "An open source finite-difference time-domain solver for room acoustics using graphics processing units," in *Proc. Forum Acusticum*, Krakow, Poland, 2014, pp. 1–6.

[6] T. Jüterbock, U. Finnendahl, M. Worchel, D. Wujecki, M. Alexa, and S. Weinzierl, "Misuka: An open-source differentiable room acoustic renderer," *Proc. Mttgs. Acoust.*, vol. 58, no. 1, p. 022 004, 2026. DOI: 10.1121/2.0002193.

[7] Y. Schubert, E. Sarradj, and M. Lemke, "A volume penalization method for acoustic and aeroacoustic simulation of solid and porous bluff bodies," *Comput. Fluids*, vol. 299, p. 106 683, 2025. DOI: 10.1016/j.compfluid.2025.106683.

[8] M. Lemke and J. Reiss, "Approximate acoustic boundary conditions in the time-domain using volume penalization," *J. Acoust. Soc. Am.*, vol. 153, no. 2, pp. 1219–1228, 2023. DOI: 10.1121/10.0017347.

[9] J. Berland, C. Bogey, and C. Bailly, "Low-dissipation and low-dispersion fourth-order Runge–Kutta algorithm," *Comput. Fluids*, vol. 35, pp. 1459–1463, 2006. DOI: 10.1016/j.compfluid.2005.04.003.

[10] D. V. Gaitonde and M. R. Visbal, "Pade-type higher-order boundary filters for the Navier-Stokes equations," *AIAA J.*, vol. 38, no. 11, pp. 2103–2112, 2000. DOI: 10.2514/2.872.

[11] K. W. Thompson, "Time dependent boundary conditions for hyperbolic systems," *J. Comput. Phys.*, vol. 68, no. 1, pp. 1–24, 1987. DOI: 10.1016/0021-9991(87)90041-6.

[12] A. Mani, "Analysis and optimization of numerical sponge layers as a nonreflective boundary treatment," *J. Comput. Phys.*, vol. 231, pp. 704–716, 2012. DOI: 10.1016/j.jcp.2011.10.017.

[13] A. B. Hölter, M. Lemke, S. Weinzierl, and F. Brinkmann, "Modeling head- and pinna-related transfer functions using boundary elements and finite differences with volume penalization," *npj Acoust.*, vol. 2, no. 16, pp. 1–11, 2026. DOI: https://doi.org/10.1038/s44384-026-00052-x.

[14] D. Pekurovsky, "P3DFFT: A framework for parallel computations of Fourier transforms in three dimensions," *SIAM J. Sci. Comput.*, vol. 34, no. 4, pp. C192–C209, 2012. DOI: 10.1137/11082748X.

[15] S. T. Prepeliţă, J. Gómez Bolaños, M. Geronazzo, R. Mehra, and L. Savioja, "Pinna-related transfer functions and lossless wave equation using finite-difference methods: Validation with measurements," *J. Acoust. Soc. Am.*, vol. 147, no. 5, pp. 3631–3645, 2020. DOI: 10.1121/10.0001230.

[16] M. Sroka, E. Sarradj, and M. Lemke, "Moving source detection using microphone array measurements based on Euler equations," *J. Acoust. Soc. Am.*, vol. 158, no. 1, pp. 419–432, 2025. DOI: 10.1121/10.0037089.

[17] *Benchmarking array analysis methods, BTU Cottbus-Senftenberg*, https://www.b-tu.de/fg-akustik/lehre/aktuelles/arraybenchmark, last seen June 17, 2026.

[18] M. Sroka, E. Sarradj, and M. Lemke, "Adjoint-based identification of moving sound sources considering rigid walls," in *Proc. DAS—DAGA*, Copenhagen, Denmark, 2025, pp. 839–842. DOI: 10.71568/dasdaga2025.298.

[19] M. Lemke, A. B. Hölter, and S. Weinzierl, "Physics-informed identification and interpolation of the directivity of sound sources," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 33, pp. 372–385, 2025. DOI: 10.1109/TASLP.2024.3519872.

[20] A. B. Hölter, M. Lemke, and S. Weinzierl, "Simulation and adjoint-based identification of a room acoustical benchmark case," in *Proc. DAS—DAGA*, Copenhagen, Denmark, 2025, pp. 580–583. DOI: 10.71568/dasdaga2025.208.

[21] L. Stein, F. Straube, J. Sesterhenn, S. Weinzierl, and M. Lemke, "Adjoint-based optimization of sound reinforcement including non-uniform flow," *J. Acoust. Soc. Am.*, vol. 146, no. 3, pp. 1774–1785, 2019. DOI: 10.1121/1.5126516.