DTU Library

Physics-informed neural networks for one-dimensional sound field predictions with parameterized sources and impedance boundaries

Borrel-Jensen, Nikolas; Engsig-Karup, Allan P.; Jeong, Cheol-Ho

Link back to DTU Orbit

Citation (APA):
Borrel-Jensen, N., Engsig-Karup, A. P., & Jeong, C-H. (2021). Physics-informed neural networks for one-dimensional sound field predictions with parameterized sources and impedance boundaries. *Jasa Express Letters*, *1*(12), [122402]. https://doi.org/10.1121/10.0009057

Physics-informed neural networks for one-dimensional sound field predictions with parameterized sources and impedance boundaries ^{EP}

Nikolas Borrel-Jensen, Allan P. Engsig-Karup and Cheol-Ho Jeong

COLLECTIONS

EP   This paper was selected as an Editor's Pick

View Online

Export Citation

ARTICLES YOU MAY BE INTERESTED IN

Deep learning for reduced order modelling and efficient temporal evolution of fluid simulations
Physics of Fluids **33**, 107101 (2021); https://doi.org/10.1063/5.0062546

Introduction to the special issue on machine learning in acoustics
The Journal of the Acoustical Society of America **150**, 3204 (2021); https://doi.org/10.1121/10.0006783

Deep learning-based high-frequency source depth estimation using a single sensor
The Journal of the Acoustical Society of America **149**, 1454 (2021); https://doi.org/10.1121/10.0003603

Physics–informed neural networks for one–dimensional sound field predictions with parameterized sources and impedance boundaries

Nikolas **Borrel-Jensen**,[1,a)] Allan P. **Engsig-Karup**,[2,b)] and Cheol-Ho **Jeong**[1,c)]

[1]*Acoustic Technology, Department of Electrical Engineering, Technical University of Denmark, 2800 Kongens Lyngby, Denmark*

[2]*Department of Applied Mathematics and Computer Science, Technical University of Denmark, 2800 Kongens Lyngby, Denmark*

nibor@elektro.dtu.dk, apek@dtu.dk, chj@elektro.dtu.dk

Abstract: Realistic sound is essential in virtual environments, such as computer games and mixed reality. Efficient and accurate numerical methods for pre-calculating acoustics have been developed over the last decade; however, pre-calculating acoustics makes handling dynamic scenes with moving sources challenging, requiring intractable memory storage. A physics-informed neural network (PINN) method in one dimension is presented, which learns a compact and efficient surrogate model with parameterized moving Gaussian sources and impedance boundaries and satisfies a system of coupled equations. The model shows relative mean errors below 2%/0.2 dB and proposes a first step in developing PINNs for realistic three-dimensional scenes. © 2021 Author(s). All article content, except where otherwise noted, is licensed under a Creative Commons Attribution (CC BY) license (*http://creativecommons.org/licenses/by/4.0/*).

1. Introduction

In computer games and mixed reality, realistic sound is essential for an immersive user experience. The impulse responses (IR) can be obtained accurately and efficiently by numerically solving the wave equation using traditional numerical methods, such as finite element methods,[1] spectral element methods (SEM),[2] discontinuous Galerkin finite element method,[3] and finite-difference time-domain methods.[4,5] For real-time applications spanning a broad frequency range, the IRs are calculated offline due to the computational requirements. However, for dynamic, interactive scenes with numerous moving sources and receivers, the computation time and storage requirement for a lookup database become intractable (in the range of gigabytes) since the IR is calculated for each source-receiver pair. When covering the whole audible frequency range, these challenges become even more extensive. Previous attempts to overcome the storage requirements of the IRs include work for lossy compression,[6] and lately, a novel portal search method has been proposed as a drop-in solution to pre-computed IRs to adapt to flexible scenes, e.g., when doors and windows are opened and closed.[7] A recent technique for handling parameter parameterization and model order reduction for acceleration of numerical models is the reduced basis method (RBM).[8,9] Although very efficient, RBM cannot meet the runtime requirements regarding computation time for virtual acoustics.

In this paper, we consider a new approach using physics-informed neural networks (PINNs)[10–12] including knowledge of the underlying physics (in contrary to traditional "black box" neural networks[13]) to learn a surrogate model for a one-dimensional (1D) domain that can be executed very efficiently at runtime (in the range of ms) and takes up little storage due to their intrinsic interpolation properties in grid-less domains. The applications of PINNs in virtual acoustics are very limited,[14,15] and the main contribution of this work is the development of frequency-dependent and independent impedance boundary conditions with parameterized moving Gaussian sources, making it possible to model sound propagation taking boundary materials properly into account. This work investigated PINNs for virtual acoustics in a 1D domain—still taking the necessary physics into account—making it a possible stepping stone to model realistic and complex 3D scenes for applications, such as games and mixed reality, where the computation and storage requirements are very strict.

[a)]Author to whom correspondence should be addressed, ORCID: 0000-0002-8820-4635.
[b)]ORCID: 0000-0001-8626-1575.
[c)]ORCID: 0000-0002-9864-7317.

2. Methods

We take a data-free approach where only the underlying physics is included in the training and their residual minimized through the loss function, allowing insights into how well PINNs perform for predicting sound fields in acoustic conditions. The Gaussian impulse is used as the initial condition tested with frequency-independent and dependent impedance boundaries. To assess the quality of the developed PINN models, we have used our in-house open-source SEM simulators.[2]

2.1 Governing equations

We consider in the following the use of PINNs for the construction of a surrogate model predicting the solution to the linear wave equation in 1D,

$$\frac{\partial^2 p(x,t)}{\partial t^2} - c^2 \frac{\partial^2 p(x,t)}{\partial x^2} = 0, \quad t \in \mathbb{R}^+, \quad x \in \mathbb{R}, \tag{1}$$

where p is the pressure (Pa), t is the time (s), and c is the speed of sound in air (m/s). The initial conditions (ICs) are satisfied by using a Gaussian source for the pressure part and setting the velocity equal to zero,

$$p(x, t=0, x_0) = \exp\left[-\left(\frac{x-x_0}{\sigma_0}\right)^2\right], \quad \frac{\partial p(x, t=0, x_0)}{\partial t} = 0, \tag{2}$$

with σ_0 being the width of the pulse determining the frequencies to span.

2.2 Boundary conditions

We consider impedance boundaries and denote the boundary domain as Γ (in 1D, the left and right endpoints). We will omit the source position x_0 in the following.

2.2.1 Frequency-independent impedance boundaries

The acoustic properties of a wall can be described by its surface impedance[16] $Z_s = p/v_n$, where v_n is the normal component of the velocity at the same location on the wall surface. Combining the surface impedance with the pressure term $\partial p/\partial \mathbf{n} = -\rho_0(\partial v_n/\partial t)$ of the linear coupled wave equation yields

$$\frac{\partial p}{\partial t} = -c\xi \frac{\partial p}{\partial \mathbf{n}}, \tag{3}$$

where $\xi = Z_s/(\rho_0 c)$ is the normalized surface impedance and ρ_0 denotes the air density (kg/m³). Note that perfectly reflecting boundaries can be obtained by letting $\xi \to \infty$ being the Neumann boundary formulation.

2.2.2 Frequency-dependent impedance boundaries

The wall impedance can be written as a rational function in terms of the admittance $Y = 1/Z_s$ and rewritten by using partial fraction decomposition in the last equation (17)

$$Y(\omega) = \frac{a_0 + \cdots + a_N(-i\omega)^N}{1 + \cdots + b_N(-i\omega)^N} = Y_\infty + \sum_{k=0}^{Q-1} \frac{A_k}{\lambda_k - i\omega} + \sum_{k=0}^{S-1}\left(\frac{B_k + iC_k}{\alpha_k + i\beta_k - i\omega} + \frac{B_k - iC_k}{\alpha_k - i\beta_k - i\omega}\right), \tag{4}$$

where a, b are real coefficients; $i = \sqrt{-1}$ is the complex number; Q is the number of real poles λ_k; S is the number of complex conjugate pole pairs $\alpha_k \pm j\beta_k$; and Y_∞, A_k, B_k, and C_k are numerical coefficients. Since we are concerned with the (time-domain) wave equation, the inverse Fourier transform is applied on the admittance and on the partial fraction decomposition term in Eq. (4). Combining these gives[17]

$$v_n(t) = Y_\infty p(t) + \sum_{k=0}^{Q-1} A_k \phi_k(t) + \sum_{k=0}^{S-1} 2\left[B_k \psi_k^{(0)}(t) + C_k \psi_k^{(1)}(t)\right]. \tag{5}$$

The functions ϕ_k, $\psi_k^{(0)}$, and $\psi_k^{(1)}$ are the so-called accumulators determined by the following set of ordinary differential equations (ODEs) referred to as auxiliary differential equations (ADEs):

$$\frac{d\phi_k}{dt} + \lambda_k \phi_k = p, \quad \frac{d\psi_k^{(0)}}{dt} + \alpha_k \psi_k^{(0)} + \beta_k \psi_k^{(1)} = p, \quad \frac{d\psi_k^{(1)}}{dt} + \alpha_k \psi_k^{(1)} - \beta_k \psi_k^{(0)} = 0. \tag{6}$$

The boundary conditions can then be formulated by inserting the velocity v_n calculated in Eq. (5) into the pressure term of the linear coupled wave equation $\partial p/\partial \mathbf{n} = -\rho_0(\partial v_n/\partial t)$.

2.3 PINNs

Two multi-layer feed-forward neural networks are setup,

$$\hat{f} : (x, t, x_0) \mapsto \mathcal{N}_f(x, t, x_0; \mathbf{W}, \mathbf{b}), \quad \hat{g} : (x, t, x_0) \mapsto \mathcal{N}_{\text{ADE}}(x, t, x_0; \mathbf{W}, \mathbf{b}),$$

where $\mathbf{W}$ and $\mathbf{b}$ are the network weights and biases, respectively; $\mathcal{N}_{\text{ADE}}$ is only applied in case of frequency-dependent boundaries. The networks take three inputs x, t, x_0 corresponding to the spatial, temporal, and source position dimensions. The network $\mathcal{N}_f$ has one output $\hat{p}(x, t, x_0)$ approximating $p(x, t, x_0)$ predicting the pressure; the network $\mathcal{N}_{\text{ADE}}$ is a multi-output network with the number of outputs corresponding to the number of accumulators to approximate as explained in the following.

Including only information about the underlying physics, the governing partial differential equation (PDE) from Eq. (1) and initial conditions (ICs) from Eq. (2) can be learned by minimizing the mean squared error loss denoted $\|\bullet\|$ as

$$\underset{\mathbf{W},\mathbf{b}}{\text{argmin}}\, \mathcal{L}(\mathbf{W}, \mathbf{b}) = \mathcal{L}_{\text{PDE}} + \lambda_{\text{IC}}\mathcal{L}_{\text{IC}} + \lambda_{\text{BC}}\mathcal{L}_{\text{BC}} + \mathcal{L}_{\text{ADE}}, \tag{7}$$

where

$$\mathcal{L}_{\text{PDE}} = \left\| \frac{\partial^2}{\partial t^2} \mathcal{N}_f(x_f^i, t_f^i, x_{0,f}^i; \mathbf{W}, \mathbf{b}) - c^2 \nabla^2 \mathcal{N}_f(x_f^i, t_f^i, x_{0,f}^i; \mathbf{W}, \mathbf{b}) \right\|, \tag{8a}$$

$$\mathcal{L}_{\text{IC}} = \left\| \mathcal{N}_f(x_{\text{ic}}^i, 0, x_{0,\text{ic}}^i; \mathbf{W}, \mathbf{b}) - \exp\left[-\left(\frac{x_{\text{ic}}^i - x_{0,\text{ic}}^i}{\sigma_0} \right)^2 \right] \right\| + \left\| \frac{\partial}{\partial t} \mathcal{N}_f(x_{\text{ic}}^i, 0, x_{0,\text{ic}}^i; \mathbf{W}, \mathbf{b}) \right\|. \tag{8b}$$

Here, $\{x_{\text{ic}}^i, x_{0,\text{ic}}^i\}_{i=1}^{N_{\text{IC}}}$ denotes the initial N_{IC} data points, $\{x_f^i, t_f^i, x_{0,f}^i\}_{i=1}^{N_f}$ denotes the N_f collocation points for the PDE f, and the penalty weights λ_{IC} and λ_{BC} are used for balancing the impact of the individual terms. The loss function $\mathcal{L}_{\text{BC}}$ will be treated separately for the impedance boundary conditions in the following, where $\{x_{\text{bc}}^i, t_{\text{bc}}^i, x_{0,\text{bc}}^i\}_{i=1}^{N_{\text{BC}}}$ will, correspondingly, be denoting the N_{BC} collocation points on the boundaries. For frequency-dependent boundaries, an auxiliary neural network will be coupled, resulting in the additional loss term $\mathcal{L}_{\text{ADE}}$ in Eq. (7) and is explained in detail in the following.

2.3.1 Frequency-independent impedance boundary loss functions

The frequency-independent boundary condition, Eq. (3), is included in the loss function $\mathcal{L}_{\text{BC}} := \mathcal{L}_{\text{INDEP}}$ satisfied by $\mathcal{L}_{\text{INDEP}} = \|(\partial/\partial t)\mathcal{N}_f(x_{\text{bc}}^i, t_{\text{bc}}^i, x_{0,\text{bc}}^i; \mathbf{W}, \mathbf{b}) + c\xi(\partial/\partial \mathbf{n})\mathcal{N}_f(x_{\text{bc}}^i, t_{\text{bc}}^i, x_{0,\text{bc}}^i; \mathbf{W}, \mathbf{b})\|$.

2.3.2 Frequency-dependent impedance boundary loss functions

For frequency-dependent boundaries, the ADEs need to be solved as well, approximating the ODEs from Eq. (6) by introducing an additional neural network $\mathcal{N}_{\text{ADE}}(x_b, t, x_0; \mathbf{W}, \mathbf{b})$ parameterized by x_b and x_0 for boundary positions and moving sources, respectively. The network has multiple outputs $\mathcal{N}_{\text{ADE}}(x_b, t, x_0; \mathbf{W}, \mathbf{b}) = [\tilde{\phi}_0, \tilde{\phi}_1, \dots, \tilde{\phi}_{Q-1}; \tilde{\psi}_0^{(0)}, \tilde{\psi}_1^{(0)}, \dots,$ $\tilde{\psi}_{S-1}^{(0)}; \tilde{\psi}_0^{(1)}, \tilde{\psi}_1^{(1)}, \dots, \tilde{\psi}_{S-1}^{(1)}]$ corresponding to the *scaled* accumulators determined by a scaling factor $l_{\text{ADE}}^{\bullet}$, mapping $\tilde{\phi}_k = l_{\text{ADE}}^{\phi_k}\hat{\phi}_k$, $\tilde{\psi}_k^{(0)} = l_{\text{ADE}}^{\psi_k}\hat{\psi}_k^{(0)}$, and $\tilde{\psi}_k^{(1)} = l_{\text{ADE}}^{\psi_k}\hat{\psi}_k^{(1)}$, such that $\tilde{\phi}_k, \tilde{\psi}_k^{(0)}, \tilde{\psi}_k^{(1)} : (x_b, t, x_0) \mapsto [-1, 1]$ matching the range of the tanh function used in this work. The scaling factors are independent of the geometry and domain dimensionality, only the material properties determine the amplitude of the accumulators. In this work, the scaling factors are determined using the SEM solver, but might be analytically estimated from the accumulators considering only a single reflection in a 1D domain. A graphical representation of the neural network architectures for approximating the governing physical equations and ADEs is depicted in Fig. 1. The accumulators with parametrized moving sources and boundary positions $\tilde{\phi}_k(x_{\text{bc}}^i, t_{\text{bc}}^i, x_{0,\text{bc}}^i)$, $\tilde{\psi}_k^{(0)}(x_{\text{bc}}^i, t_{\text{bc}}^i, x_{0,\text{bc}}^i)$, $\tilde{\psi}_k^{(1)}(x_{\text{bc}}^i, t_{\text{bc}}^i, x_{0,\text{bc}}^i)$ can be learned by minimizing the mean squared error loss as (arguments omitted)

$$\underset{\mathbf{W},\mathbf{b}}{\text{argmin}}\, \mathcal{L}_{\text{ADE}}(\mathbf{W}, \mathbf{b}) = \sum_{k}^{Q-1} \tilde{\lambda}_{\text{ADE}}^{\tilde{\phi}_k} \mathcal{L}_{\tilde{\phi}_k} + \sum_{k}^{S-1} \left(\tilde{\lambda}_{\text{ADE}}^{\tilde{\psi}_k^{(0)}} \mathcal{L}_{\tilde{\psi}_k^{(0)}} + \tilde{\lambda}_{\text{ADE}}^{\tilde{\psi}_k^{(1)}} \mathcal{L}_{\tilde{\psi}_k^{(1)}} \right), \tag{9}$$

where

$$\mathcal{L}_{\tilde{\phi}_k} = \left\| \frac{\partial}{\partial t} \tilde{\phi}_k + \lambda_k \tilde{\phi}_k - l_{\text{ADE}}^{\phi_k} \mathcal{N}_f \right\|, \tag{10a}$$

$$\mathcal{L}_{\tilde{\psi}_k^{(0)}} = \left\| \frac{\partial}{\partial t} \tilde{\psi}_k^{(0)} + \alpha_k \tilde{\psi}_k^{(0)} + \beta_k l_{\text{ADE}}^{\psi_k^{(0)}}(1/l_{\text{ADE}}^{\psi_k^{(1)}}) \tilde{\psi}_k^{(1)} - l_{\text{ADE}}^{\psi_k^{(0)}} \mathcal{N}_f \right\|, \tag{10b}$$

$$\mathcal{L}_{\tilde{\psi}_k^{(1)}} = \left\| \frac{\partial}{\partial t} \tilde{\psi}_k^{(1)} + \alpha_k \tilde{\psi}_k^{(1)} - \beta_k l_{\text{ADE}}^{\psi_k^{(1)}}(1/l_{\text{ADE}}^{\psi_k^{(0)}}) \tilde{\psi}_k^{(0)} \right\|, \tag{10c}$$
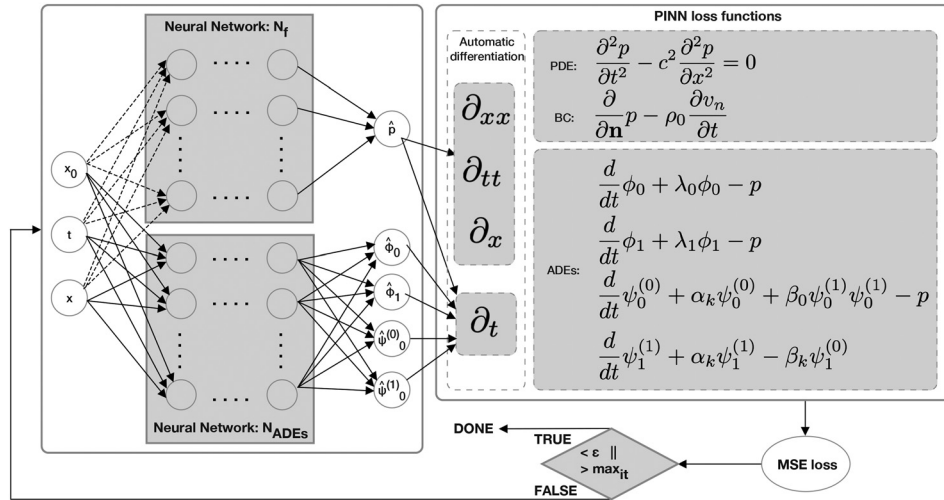
and

Fig. 1. PINN scheme for frequency-dependent boundaries. Left: Two fully connected feed-forward neural network architectures, $\mathcal{N}_f$ (PDE + ICs) and $\mathcal{N}_{\mathrm{ADE}}$ (ADEs). Right: The governing physical equations and ADEs are coupled via the loss function (ICs and scaling terms are omitted for brevity). Training is done when a maximum number of epochs is reached, or the total loss is smaller than a given threshold.

$$\tilde{\lambda}_{\mathrm{ADE}}^{\tilde{\phi}_k} = (1/l_{\mathrm{ADE}}^{\phi_k})\lambda_{\mathrm{ADE}}^{\tilde{\phi}_k}, \quad \tilde{\lambda}_{\mathrm{ADE}}^{\tilde{\psi}_k^{(0)}} = (1/l_{\mathrm{ADE}}^{\psi_k^{(0)}})\lambda_{\mathrm{ADE}}^{\tilde{\psi}_k^{(0)}}, \quad \tilde{\lambda}_{\mathrm{ADE}}^{\tilde{\psi}_k^{(1)}} = (1/l_{\mathrm{ADE}}^{\psi_k^{(1)}})\lambda_{\mathrm{ADE}}^{\tilde{\psi}_k^{(1)}}.$$

The frequency-dependent boundary conditions are satisfied by $\mathcal{L}_{\mathrm{DEP}} = \|(\partial/\partial\mathbf{n})\mathcal{N}_f(x_{\mathrm{bc}}^i, t_{\mathrm{bc}}^i, x_{0,\mathrm{bc}}^i; \mathbf{W}, \mathbf{b})$ $+\rho_0[\partial v_n(x_{\mathrm{bc}}^i, t_{\mathrm{bc}}^i, x_{0,\mathrm{bc}}^i)/\partial t]\|$, where v_n is the expression at the boundaries given in Eq. (5) with $\hat{\phi}_k = 1/l_{\mathrm{ADE}}^{\phi_k}\tilde{\phi}_k$, $\hat{\psi}_k^{(0)}$ $= 1/l_{\mathrm{ADE}}^{\psi_k^{(0)}}\tilde{\psi}_k^{(0)}$, and $\hat{\psi}_k^{(1)} = 1/l_{\mathrm{ADE}}^{\psi_k^{(1)}}\tilde{\psi}_k^{(1)}$, ensuring that the accumulators are properly re-scaled. The loss is included as the term $\mathcal{L}_{\mathrm{BC}} := \mathcal{L}_{\mathrm{DEP}}$ in Eq. (7) together with the loss for the ADEs in Eq. (9).

2.4 Setup

TENSORFLOW 2.5.1,[18] SCIANN 0.6.4.7,[19] and PYTHON 3.8.9 with 64 bit floating points for the neural network weights are used. The code for reproducing the results can be found online.[20]

Reference data for impedance boundaries are generated using a fourth-order Jacobi polynomial SEM solver. The grid was discretized with 20 points per wavelength spanning frequencies up to 1000 Hz yielding an average grid resolution of $\Delta x = 0.017$ m, and the time step was $\Delta t = \mathrm{CFL} \times \Delta x/c$, where CFL is the Courant-Friedrichs-Lewy constant (CFL = 1.0 and CFL = 0.1 for frequency-independent and dependent boundaries, respectively). The speed of sound $c = 1$ m/s is used for the PINN setup, implying $\Delta x = \Delta t/c = \Delta t$, which is a normalization introduced to ensure the same scaling in time and space required for the optimization problem to converge. In case of a normalized speed of sound, the effective normalized frequency is correspondingly $f = f_{\mathrm{phys}}/c_{\mathrm{phys}}$, since the wave is now travelling slower compared to the physical setup. To evaluate the results for a physical speed of sound $c_{\mathrm{phys}} = 343$ m/s, the temporal dimension should be converted back as $t_{\mathrm{phys}} = t/c_{\mathrm{phys}}$ s. In case of frequency-dependent boundaries, the velocity from Eq. (5) needs to be normalized accordingly regarding frequency and flow resistivity $\sigma_{\mathrm{mat}} = \sigma_{\mathrm{mat,phys}}/c_{\mathrm{phys}}$ in Miki's model. Fitting the parameters for $c = 1$ m/s yields modified λ_k, α_k, β_k, and Y_∞ values resulting in the exact same surface impedance as for $c = 343$ m/s, but scaled by c_{phys} in frequency range and amplitude. This can be seen from the complex wavenumber and characteristic impedance of the porous medium in Miki's model involving f/σ_{mat} and $2\pi f/c$ not being affected by normalization.[20]

The point distribution in time and space, number of sources, penalty weights λ and scaling factors l_{ADE} for the ADEs are listed in Table 1. Note that the number of (time and space) domain points (30%) per source (7) is $47\,089 \times 0.3/7 = 2018$, satisfying the Nyquist theorem, since $\Delta x = 2/\sqrt{2018} = 0.045$ m (we can use the square root to get the gridpoint distribution in the spatial dimension) resulting in ppw $= \lambda_w/\Delta x = 7.6$ points per wavelength; $\lambda_w = c/f$

Table 1. Number of points in time and space; inner domain, boundaries, and initial condition point distributions; number of evenly distributed sources (srcs); values for the penalty weights λ; scaling factors l for normalizing the accumulators.

#total	#BC	#IC	#inner[a]	#srcs	λ_{IC}	λ_{BC}	$\lambda_{\mathrm{ADE}}^\bullet$	$l_{\mathrm{ADE}}^{\phi_0}$	$l_{\mathrm{ADE}}^{\phi_1}$	$l_{\mathrm{ADE}}^{\psi_0^{(0)}}$	$l_{\mathrm{ADE}}^{\psi_0^{(1)}}$
47,089	45%	25%	30%	7	20	1	10	10.3	261.4	45.9	22.0

[a]The centered Latin hypercube sampling strategy (Ref. 24) is used.

being the wavelength for physical frequency 1000 Hz and physical speed of sound 343 m/s. For the neural network, we have used the ADAM optimizer and the mean-squared error for calculating the losses for both networks. The training was run with learning rate 1e−4 and batch size 512 until a total loss of $\epsilon = 2e − 4$ was reached (roughly 16k and 20k epochs needed for frequency-independent and dependent boundaries, respectively). The relatively big batch size was chosen to ensure that enough initial and boundary points were included in the optimization steps.

The network architecture of $\mathcal{N}_f$ consists of three layers, each with 256 neurons applying the sine activation function in each layer except for a linear output layer, with proper weight initialization.[21] Using sine activation functions can be seen as representing the signal using Fourier series[22] and is probably the reason for a significantly better convergence compared to using the more common choice of tanh activation functions. However, experiments showed degraded interpolation properties using sine activation functions when the network was trained on grids with source positions distributed more sparsely (0.3 m), even when lowering the number of neurons to prevent overfitting. A reason could be related to the distributed source interval violating the Nyquist sampling theorem $\Delta x < c/(2f) = 0.17$ m and causing aliasing effects, but this remains an open question. Therefore, the source positions were distributed evenly with finer resolution to improve the results between the source positions, consequently resulting in a sparser grid per source by keeping the total number of points the same. Despite the sparser grid, the convergence and final error still showed satisfying results. Distributing the source positions more densely is trivial in a data-free implementation, but if a combination of the underlying physics and simulated/measured data is considered later, a large number of source positions could be practically challenging.

The network architecture of $\mathcal{N}_{ADE}$ consists of three layers, each with 20 neurons applying the tanh activation function in each layer except for a linear output layer, with Glorot normal initialization of the weights.[23] Using the tanh function is an obvious choice since we have chosen to scale the accumulators to take values in the range $[−1, 1]$.

3. Results

Frequency-independent and dependent boundary conditions are tested, each with parameterized moving sources trained at seven evenly distributed positions $\mathbf{x}_0 = [−0.3, −0.2, …, 0.3]$ m and evaluated at five positions $\mathbf{x}_0 = [−0.3, −0.15, 0.0, 0.15, 0.3]$ m. Additional results are included as supplementary materials.[20] The source is satisfied through the initial
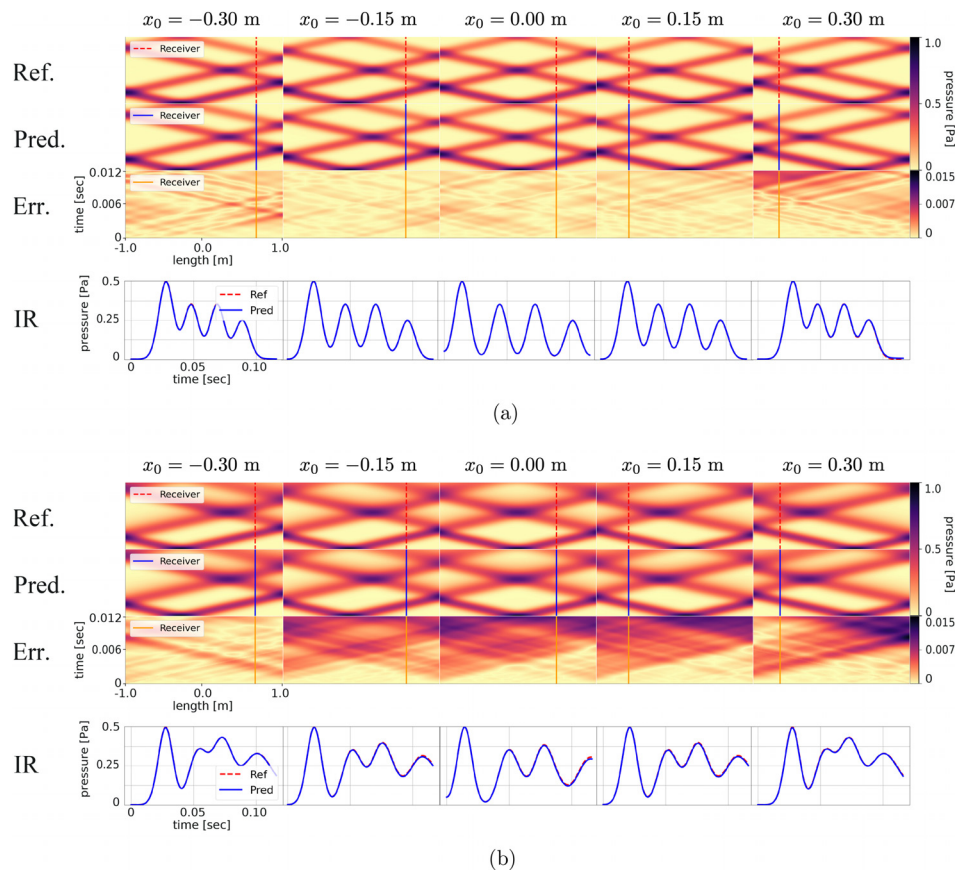

Fig. 2. Wave propagations in a 1D domain $[−1, 1]$ m.

Table 2. Time domain errors for source/receiver pairs (x_0, x) measured in meters in what follows at positions $s_0 = (-0.3, 0.64)$, $s_1 = (-0.15, 0.58)$, $s_2 = (0.0, 0.5)$, $s_3 = (0.15, -0.58)$, $s_4 = (0.3, -0.66) \mu_{\mathrm{rel}}(x, x_0) = (1/N) \sum_{i=0}^{N-1} [|\hat{p}(x, t^i, x_0) - p(x, t^i, x_0)|/p(x, t^i, x_0)]$ is the relative mean error over time within -60 dB range and $\infty_{\mathrm{abs}}(x, x_0) = \max\{|\hat{p}(x, t^i, x_0) - p(x, t^i, x_0)| : i = 0 \cdots N - 1\}$ Pa is the maximum absolute error for source/receiver pair s_i.

	s_0		s_1		s_2		s_3		s_4	
	μ_{rel}	∞_{abs}	μ_{rel}	∞_{abs}	μ_{rel}	∞_{abs}	μ_{rel}	∞_{abs}	μ_{rel}	∞_{abs}
Freq. indep.	0.7%	0.004	0.3%	0.002	0.5%	0.001	0.4%	0.002	1.5%	0.006
Freq. dep.	0.5%	0.004	1.5%	0.008	1.7%	0.011	1.9%	0.009	1.1%	0.005

condition modeled as a Gaussian impulse from Eq. (2) with $\sigma_0 = 0.2$ spanning frequencies up to 1000 Hz. The speed of sound $c_{\mathrm{phys}} = 343$ m/s and air density $\rho_0 = 1.2$ kg/m^3 are used for all studies.

First, we test the frequency-independent boundary condition with normalized impedance $\xi = 5.83$ depicted in Fig. 2(a). Frequency-independent boundaries with corresponding wave propagation animations available from Mm. 1. Then, we test the frequency-dependent impedance boundary condition, where the boundary is modeled as a porous material mounted on a rigid backing with thickness $d_{\mathrm{mat}} = 0.10$ m with an air flow resistivity of $\sigma_{\mathrm{mat,phys}} = 8000$ Nsm^{-4}. The surface impedance Y of this material is estimated using Miki's model[25] and mapped to a two-pole rational function in the form of Eq. (4) with $Q = 2$ and $S = 1$ using a vector fitting algorithm[26] yielding the coefficients for Eq. (5). The results are depicted in Fig. 2(b). Frequency-dependent boundaries with corresponding wave propagation animations available from Mm. 2.

> Mm. 1. Frequency-independent boundaries, animation. Same parameters as Fig. 2(a). Frequency-independent boundaries. File of type "mp4" (1.3 MB).

> Mm. 2. Frequency-dependent boundaries, animation. Same parameters as Fig. 2(b). Frequency-dependent boundaries. File of type "mp4" (1.6 MB).

We observe that the shape of the wave propagations is well captured, and the impulse responses also fit the reference solutions very well for all boundary types. The relative mean error $\mu_{\mathrm{rel}}(x, x_0) = (1/N) \sum_{i=0}^{N-1} [|\hat{p}(x, t^i, x_0) - p(x, t^i, x_0)|/p(x, t^i, x_0)]$ within -60 dB and absolute maximum error $\infty_{\mathrm{abs}}(x, x_0) = \max\{|\hat{p}(x, t^i, x_0) - p(x, t^i, x_0)| : i = 0 \cdots N - 1\}$ of the impulse responses originating from various source and receiver positions are summarized in Table 2. Relative errors are below 2%/0.2 dB for all predictions. The absolute maximum errors are below 0.011 Pa for all predictions indicating that no severe outliers are present.

4. Conclusion and future work

A novel method is presented for predicting the sound field in a 1D domain for impedance boundaries and parameterized moving Gaussian sources using PINNs. A coupled system of differential equations, consisting of the governing physical equations and a system of ODEs predicting the accumulators of the ADEs, was used for training the PINN. The equations for the ADEs depend only on time t but were parameterized to take boundary and source positions into account, yielding a very flexible implementation. The results are promising, with relative mean errors below 2%/0.2 dB for all cases. The approach taken by learning a compact surrogate model that is inexpensive to evaluate at runtime shows potential to overcome current numerical methods' limitations in modeling flexible scenes, such as moving sources.

Compared to standard numerical methods, the PINN method takes up to three orders of magnitude more time to converge. Therefore, to solve realistic problems in 3D, the convergence rate needs to be improved. This is partly due to the need for fairly large amounts of grid points with 70% of the points located at the initial time step and at boundaries where penalty weights are also needed for balancing each loss term. Formulating an ansatz imposing initial and boundary conditions directly could overcome this problem.[27] Also, considering other architectures taking (discrete) time-dependence into account instead of optimizing the entire spatiotemporal domain at once might improve the learning rate and produce more precise results. Moreover, we have observed challenges in the global optimizer for a larger domain size and/or by increasing the frequency due to the ratio between zero and non-zero pressure values. Domain decomposition methods[28] have been introduced to overcome this limitation. In ongoing work, more complex benchmarks are being considered.

References and links

[1]T. Okuzono, T. Otsuru, R. Tomiku, and N. Okamoto, "A finite-element method using dispersion reduced spline elements for room acoustics simulation," Appl. Acoust. **79**, 1–8 (2014).

[2]F. Pind, A. P. Engsig-Karup, C.-H. Jeong, J. S. Hesthaven, M. S. Mejling, and J. Strømann-Andersen, "Time domain room acoustic simulations using the spectral element method," J. Acoust. Soc. Am. **145**(6), 3299–3310 (2019).

[3]A. Melander, E. Strøm, F. Pind, A. Engsig-Karup, C.-H. Jeong, T. Warburton, N. Chalmers, and J. S. Hesthaven, "Massive parallel nodal discontinuous Galerkin finite element method simulator for room acoustics," Int. J. High Perform. Comput. Appl. (2020), http://infoscience.epfl.ch/record/279868 (Last viewed 30/10/2021).

[4]D. Botteldoorena, "Finite-difference time-domain simulation of low-frequency room acoustic problems," J. Acoust. Soc. Am. **98**(6), 3302–3308 (1995).

[5]B. Hamilton and S. Bilbao, "FDTD methods for 3-D room acoustics simulation with high-order accuracy in space and time," IEEE/ACM Trans. Audio Speech Lang. Proc. **25**(11), 2112–2124 (2017).

[6]N. Raghuvanshi and J. Snyder, "Parametric wave field coding for precomputed sound propagation," ACM Trans. Graph. **33**(4), 1 (2014).

[7]N. Raghuvanshi, "Dynamic portal occlusion for precomputed interactive sound propagation," arXiv:2107.11548 (2021).

[8]J. Hesthaven, G. Rozza, and B. Stamm, *Certified Reduced Basis Methods Parametrized Partial Differential Equations* (Springer, Berlin, 2015) pp. 1–131.

[9]H. S. Llopis, A. P. Engsig-Karup, C.-H. Jeong, F. Pind, and J. S. Hesthaven, "Efficient numerical room acoustic simulations with parametrized boundaries using the spectral element and reduced basis method," arXiv:2103.11730 (2021).

[10]D. C. Psichogios and L. H. Ungar, "A hybrid neural network-first principles approach to process modeling," AIChE J. **38**(10), 1499–1511 (1992).

[11]I. Lagaris, A. Likas, and D. Fotiadis, "Artificial neural networks for solving ordinary and partial differential equations," IEEE Trans. Neural Networks **9**(5), 987–1000 (1998).

[12]M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," J. Comput. Phys. **378**, 686–707 (2019).

[13]Z. Fan, V. Vineet, H. Gamper, and N. Raghuvanshi, "Fast acoustic scattering using convolutional neural networks," *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing—Proceedings 2020-May* (2020), pp. 171–175.

[14]B. Moseley, A. Markham, and T. Nissen-Meyer, "Solving the wave equation with physics-informed deep learning," arXiv:2006.11894 (2020).

[15]M. Rasht-Behesht, C. Huber, K. Shukla, and G. E. Karniadakis, "Physics-informed neural networks (PINNs) for wave propagation and full waveform inversions," arXiv:2108.12035 (2021), pp. 1–29.

[16]H. Kuttruff, *Room Acoustics*, 6th ed. (CRC Press, Boca Raton, 2016), p. 322.

[17]R. Troian, D. Dragna, C. Bailly, and M. A. Galland, "Broadband liner impedance eduction for multimodal acoustic propagation in the presence of a mean flow," J. Sound Vib. **392**, 200–216 (2017).

[18]Google, "TensorFlow" (2021), https://www.tensorflow.org/ (Last viewed 30/10/2021).

[19]E. Haghighat and R. Juanes, "SciANN: A Keras/TensorFlow wrapper for scientific computations and physics-informed deep learning using artificial neural networks," Comput. Methods Appl. Mech. Eng. **373**, 113552 (2021).

[20]See supplementary material at https://www.scitation.org/doi/suppl/10.1121/10.0009057 for source code and addition results for Neumann boundaries, accumulator predictions, runtime efficiency of the surrogate model, and detailed explanation of the normalization for the frequency-dependent impedance boundary formulation.

[21]V. Sitzmann, J. N. P. Martel, A. W. Bergman, D. B. Lindell, and G. Wetzstein, "Implicit neural representations with periodic activation functions," arXiv:2006.09661 (2020).

[22]N. Benbarka, T. Höfer, H. ul-moqeet Riaz, and A. Zell, "Seeing implicit neural representations as fourier series," arXiv:2109.00249 (2021).

[23]X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, Vol. 9 of Proceedings of Machine Learning Research*, edited by Y. W. Teh and M. Titterington, PMLR, Chia Laguna Resort, Sardinia, Italy (2010), pp. 249–256.

[24]M. Stein, "Large sample properties of simulations using Latin hypercube sampling," Technometrics **29**(2), 143–151 (1987).

[25]Y. Miki, "Acoustical properties of porous materials-modifications of Delany-Bazley models" J. Acoust. Soc. Jpn. (E) **11**(1), 19–24 (1990).

[26]B. Gustavsen and A. Semlyen, "Rational approximation of frequency domain responses by vector fitting," IEEE Trans. Power Deliv. **14**(3), 1052–1061 (1999).

[27]N. Sukumar and A. Srivastava, "Exact imposition of boundary conditions with distance functions in physics-informed deep neural networks," arxiv.org/abs/2104.08426 (2021).

[28]K. Shukla, A. D. Jagtap, and G. E. Karniadakis, "Parallel physics-informed neural networks via domain decomposition," arxiv.org/abs/2104.10013 (2021).