Predicting the Propagation of Acoustic Waves using Deep Convolutional Neural Networks

Antonio Alguacil, Michaël Bauerheim, Marc Jacob, Stéphane Moreau

AIAA AVIATION Forum
June 15-19, 2020, VIRTUAL EVENT
AIAA AVIATION 2020 FORUM

10.2514/6.2020-2513

Predicting the Propagation of Acoustic Waves using Deep Convolutional Neural Networks.

Antonio Alguacil* and Michael Bauerheim†
ISAE-SUPAERO, Université de Toulouse, 31400 Toulouse, France

Marc C. Jacob‡
ISAE-SUPAERO, Université de Toulouse, 31400 Toulouse, France
Ecole Centrale de Lyon, Université de Lyon, 69130 Ecully, France

Stéphane Moreau§
Université de Sherbrooke, Sherbrooke, Quebéc, J1K2R1, Canada

A novel approach for numerically propagating acoustic waves in two-dimensional quiescent media has been developed through a fully convolutional multi-scale neural network. This data-driven method managed to produce accurate results for long simulation times with a database of Lattice Boltzmann temporal simulations of propagating Gaussian Pulses, even in the case of initial conditions unseen during training time, such as the plane wave configuration or the two initial Gaussian pulses of opposed amplitudes. Two different choices of optimization objectives are compared, resulting in an improved prediction accuracy when adding the spatial gradient difference error to the traditional mean squared error loss function. Further accuracy gains are observed when performing an *a posteriori* correction on the neural network prediction based on the conservation of acoustic energy, indicating the benefit of including physical information in data-driven methods.

Nomenclature

b	=	Gaussian pulse half-width
c_0	=	Sound speed
D	=	Domain characteristic length
G	=	Neural network function
$\mathcal{L}$	=	Loss function
N	=	number of data samples
(x, y)	=	Cartesian coordinates
X_i	=	Neural network input
Y_i	=	Target data
$\hat{Y}_i$	=	Neural network output

ε	=	Amplitude of Gaussian pulse
η	=	Dynamic viscosity
θ	=	Neural network learnable parameters
λ_i	=	Loss function weight for component i
ρ_0	=	Mean fluid density
τ	=	Non-dimensional time

*PhD Student, Département Aérodynamique, Énergétique et Propulsion, ISAE-SUPAER0, BP 54032, F-31055 Toulouse Cedex 4, antonio.alguacil-cabrerizo@isae-supaero.fr

†Associate Professor, Département Aérodynamique, Énergétique et Propulsion, ISAE-SUPAER0, BP 54032, F-31055 Toulouse Cedex 4

‡Professor, Université de Lyon, École Centrale de Lyon, INSA Lyon, Université Claude Bernard Lyon I, CNRS, Laboratoire de Mécanique des Fluides et d'Acoustique, UMR 5509, F-69134 Ecully Cedex, France, AIAA Member

§Professor, Mechanical Engineering Department, AIAA Lifetime Member

$()'$ = Fluctuating quantity

x = Scalar quantity
$\mathbf{x}$ = Vector

I. Introduction

The prediction of noise generated by aero-acoustic sources has been approached in the last 50 years with a large range of numerical and analytical techniques. Most recent numerical methods encompass direct computational aero-acoustics (CAA) or hybrid CAA coupled with acoustic analogies [1]. Direct CAA solves both source and acoustic propagation, which leads to high accuracy but also an extreme CPU cost, thus limiting the approach to academic and benchmark cases. [2–4]. Hybrid methods, however, separate the computation of the hydrodynamic fluctuations in the source regions from far field one, where acoustic fluctuations are propagated up to the observer's position. Source regions are either modelled or calculated through high-order CFD whereas far field radiation calculations derive from less expensive methods, such as Lighthill's acoustic analogy and its variants (Ffowcs Williams–Hawkings (FW-H) equation [5]), or from the resolution of the Linearized Euler Equations (LEE). The main difficulty of hybrid methods is the coupling of the source computation with the acoustic propagation computation [6]. Another difficulty arises when acoustic waves must be propagated in complex geometries or mean flows, such as ones encountered in turbomachinery [3, 7]. A recent work from Pérez Arroyo *et al*. [8] showed a first-time Large Eddy Simulation coupled with FW-H and Goldstein [9] analogies for the noise prediction of the full fan stage of the NASA SDT benchmark. One inherent limitation is the use of expensive high-order numerical schemes to ensure low dissipation and dispersion properties. Therefore, these methods remain computationally expensive in particular in aircraft design phases where multiple model evaluations are needed for an optimum to be found. Surrogate models, where the computational cost is significantly smaller than CAA, is of vital importance for the present needs of industry. Therefore, this paper focuses on an alternative to high-order numerical schemes for the propagation of acoustic waves in a complex medium.

Leveraging the power of Neural Networks as universal function approximators [10], data-driven methods have been increasingly used as approximating models in fluid mechanics-related problems [11]. Deep Convolutional Neural Networks (CNN) show an ability to learn statistical correlations of flow fields for a variety of tasks, such as surrogate modeling of 2D CFD-RANS simulations to predict steady flow fields [12], solving Poisson equations to enforce a divergence-free velocity in incompressible flows [13, 14] or modeling the complete space-time evolution of flow fields [15]. These models rely on non-linear operations to learn spatial correlations from data, in order to predict a new image given an optimization objective. As opposed to traditional approaches where the explicit resolution of equations is needed, Neural Networks build an implicit understanding of physics from data observation and the choice of an objective functional to minimize (called the loss function). This approach can greatly accelerate calculations, while maintaining acceptable accuracy levels.

However, to the authors knowledge, the use of neural networks, in particular CNN, has never been reported for the propagation of acoustic waves. Yet, it is well known that CNN has a high ability to capture spatial coherent patterns, which is typically the case of the radiated acoustic pressure fields which exhibit a spatially coherent topology. The spatio-temporal dependence of acoustic propagation remains however a challenge for data-driven methods, due to the data causality of time-series, and the lack of theoretical background on the errors made by CNNs, in particular when applied to a recurrent task as the one proposed in the following. It suggests that a careful attention must be taken on the training strategies to build the CNN model. Previous works show the usefulness of using Multi-Scale CNNs for seismic wave propagation [16], or coupled Long Short-Term Memory (LSTM)-CNN models for surface waves propagation [17]. Results from these works are mostly qualitative and reveal difficulties for predicting accurately wave propagation over long time predictions. With these models, errors tend to accumulate over time up to a point at which the wavefronts completely loose their coherence. A key aspect of this study is thus to evaluate the error propagated in time by the neural network, to establish training strategies that limit their accumulation of errors, and finally to propose a benchmark database for future works on CNN tackling the propagation of acoustic waves in complex media.

In the present work, a data-driven approach is proposed for the time-propagation of 2D acoustic waves by a Multi-Scale CNN, trained on acoustic wavefields generated by a Lattice-Boltzmann Method CFD code. The network is intended to generate the next time-step of the pressure field given a number of previous steps.

The objectives of the study are the following:

(i) To assess the ability of convolutional neural networks to propagate simple acoustic sources.
(ii) To compare extensively the results with reference cases for propagation (single gaussian pulse, two opposed-amplitude gaussian pulses, plane wave).
(iii) To study best practices for training accurately the CNN of interest. A particular attention is drawn on for the choice of the optimization criteria.
(iv) To assess these data-driven simulations by a consistent error analysis

The paper is divided as follows. In Section II Convolutional Neural Networks are presented, along with the training process and the evaluation strategies. Section III describes the Lattice Boltzmann method used for generating the dataset and validates the method in terms of numerical dissipation with respect to analytical test-cases. Section IV shows results for the three aforementioned test-cases of acoustic propagation in a closed domain without interior obstacles. Finally, conclusions are drawn in Section V.

II. Deep Convolutional Neural Network as wave propagator

A. Generalities of CNNs

Deep Convolutional Neural Networks [18, 19] have been successfully used in image recognition tasks since 2012 when for the first time a CNN model won the ILSVRC competition [20], by classifying (*i.e.* assigning labels to images) 1.2 million images of the ImageNet dataset. A typical Convolutional Neural Network architecture consists in an input image being convolved successively by multiple filters to output feature maps [21]. Non-Linear Rectifying Units (ReLu) activation layers[22] are placed in-between convolutional layers in order to learn a non-linear mapping between input and output data. CNNs use sliding filters to learn spatially correlated features from data. First layers scan local features of the input image (for velocity or pressure fields, small characteristic scales) while deeper layers learn high-level representations of the input images, as convolutions scan a larger area of the image.

Each of the convolutional filters is parametrized by learnable weights, *i.e.* modified according to an optimisation objective during the training phase. These optimization criteria are targeted *via* a *loss function*. Therefore a given architecture will output different results for a same input image if trained on different data, or with a different loss function. The main challenge of data-driven techniques is thus to choose a representative dataset for the given problem, as well as an appropiate network architecture and training loss function.

B. Training a Neural Network

A supervised training of a Neural Network G over a dataset composed of N pairs of training samples $(X_i, Y_i)_{1 \leq i \leq N}$ corresponds to the minimization of a loss function $\mathcal{L}$ calculated as an error between the prediction output $G(X_i)$ and a target data Y_i. The minimization problem can be formalized as:

$$\min_{\theta} \quad \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}\left(G(X_i, \theta), Y_i\right) \tag{1}$$

θ denoting the parameters of the network to train (composed of multiplicative weights w_i and additive biases b_i).

Mini-batch Stochastic Gradient Descent techniques are used for minimising the error, even though there are no guarantees that a global optimum will be found, due to the non-linear nature of the problem. As this technique relies on calculating the gradient of the error with respect to the different layer weights, an efficient backpropagation algorithm [23] is used, where analytical gradients of every layer output with respect to their weights are calculated at the forward pass (going from input to output). These gradients are then used in a backward pass for updating all weights of the network, so that the cost function is minimized.

Such operations are performed automatically by the Pytorch open-source library [24] implementing automatic differentiation and optimization algorithms. This framework is used in the present study.

C. Multi-Scale Architecture

Mathieu *et al.* [25] used the Multi-Scale CNN architecture in order to predict the next video frame given 4 concatenated previous frames sampled with equal time-spacing as input. Multiple simple ConvNets at different spatial resolutions are recursively used to output the prediction.

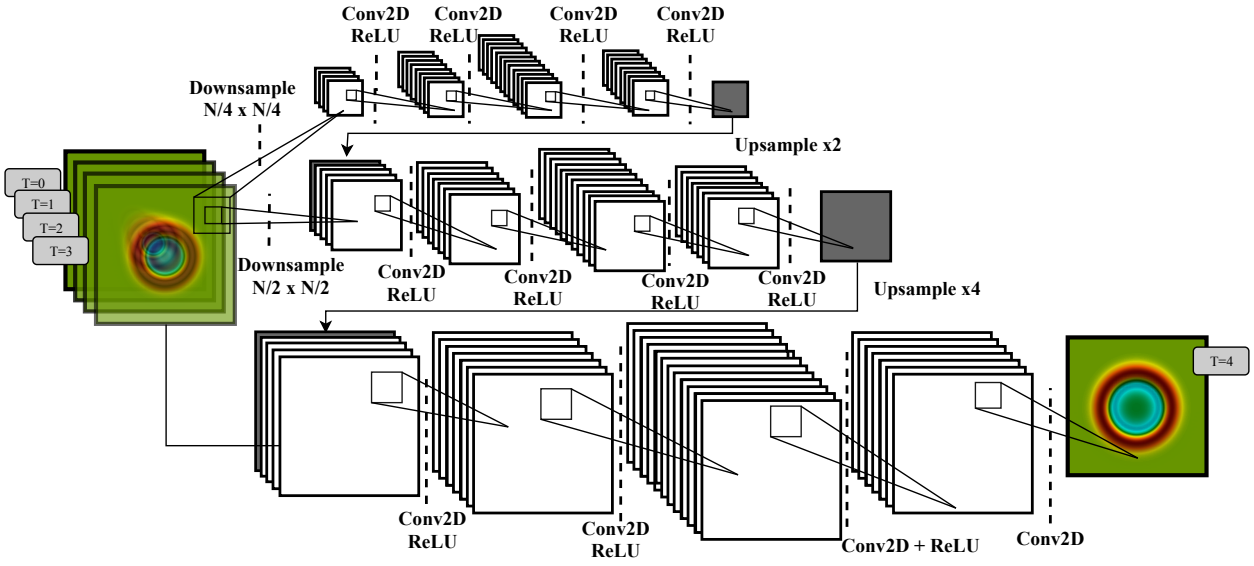


Fig. 1 MultiScale CNN with three scale banks at quarter, half and full-resolution. Input is composed of T consecutive frames $t = 0$, $t = 1$, ..., $t = T$ and outputs frame $t = T + 1$ of resolution NxN cells. Activation layers are ReLUs and depth after each convolution represents number of filters (*i.e.* feature channels).

Reproducing Mathieu's development for completeness, let $N \times N$ be the initial size of input and target (*i.e.* dataset) fields, defined as X^i and Y^i. We want to define a function G such that $G(X)$ approximates Y, *i.e.* $\hat{Y} = G(X)$ where the hat notation denotes an approximation. Let $s_1, ..., s_{N_{scales}}$ be the field sizes of all the network inputs, X_k^i, Y_k^i the downscaled field of X^i and Y^i, and u_k the upscaling operator for scale s_k. For example, if three-scales MultiScale architecture is chosen then $s_1 = \frac{N}{4} \times \frac{N}{4}$, $s_2 = \frac{N}{2} \times \frac{N}{2}$ and $s_3 = N \times N$, and u_k is the interpolating function (bi-linear) to go from one resolution to another. Let G_k be a network that learns a prediction $\hat{Y}_k$ from X_k and a coarse guess denoted $\hat{Y}_{k-1}$.

G_k makes predictions of size s_k and is defined recursively as:

$$\hat{Y}_k = G_k \left(X_k, u_k \left(\hat{Y}_{k-1} \right) \right) \tag{2}$$

The first ConvNet takes the downsampled (quarter scale) input, performs convolutions and applies activation functions while maintaining the same resolution. The output of this first bank is then upsampled to a half-scale and concatenated to the downsampled input field (to the same half-scale), which is then processed by a ConvNet. Finally, this operation is repeated for the full scale bank and a final output is produced.

This type of network is sparsely used in literature compared with more typical architectures such as U-Net [26]. The main advantage of the model is that it skips the pooling/unpooling pairs of encoder-decoder architectures that tend to act as low-pass filters. The present CNN aims at learning both low and high-wave number characteristics of the input-output mapping. Coarser resolution scales tend to filter high wavenumbers but are able to learn long-range space dependence whereas full-scales convolutions are able to focus also on high wavenumbers.

D. Loss functions

For a set of input and output data, the multi-scale neural network generates predictions that minimize the following loss function:

$$\mathcal{L} = \frac{1}{N} \sum_{k=1}^{N} \left(\lambda_{l2} \mathcal{L}_{l2}^k + \lambda_{gdl} \mathcal{L}_{gdl}^k \right) \tag{3}$$

where the summation is done across N total number of cells of the mini-batch output fields.

$\mathcal{L}_{l2}^k$ minimizes the L2-norm mean-square error (MSE) of the output with respect to the target fields (here corresponding to density fields). $\mathcal{L}_{gdl}^k$ (called Gradient Difference Loss (GDL) and introduced by Mathieu *et al.* [25])

minimizes the L2-norm mean-square error of both x and y- components of the spatial gradients of density fields, discretized with forward first-order finite differences. For a cell defined by indexes (i, j):

$$\mathcal{L}_{gdl}^{k}\left(\hat{Y}_{i,j}^{k}, Y_{i,j}^{k}\right) = \left[\left(\hat{Y}_{i+1,j}^{k} - \hat{Y}_{i,j}^{k}\right) - \left(Y_{i+1,j}^{k} - Y_{i+1,j}^{k}\right)\right]^2 + \left[\left(\hat{Y}_{i,j+1}^{k} - \hat{Y}_{i,j}^{k}\right) - \left(Y_{i,j+1}^{k} - Y_{i,j}^{k}\right)\right]^2 \tag{4}$$

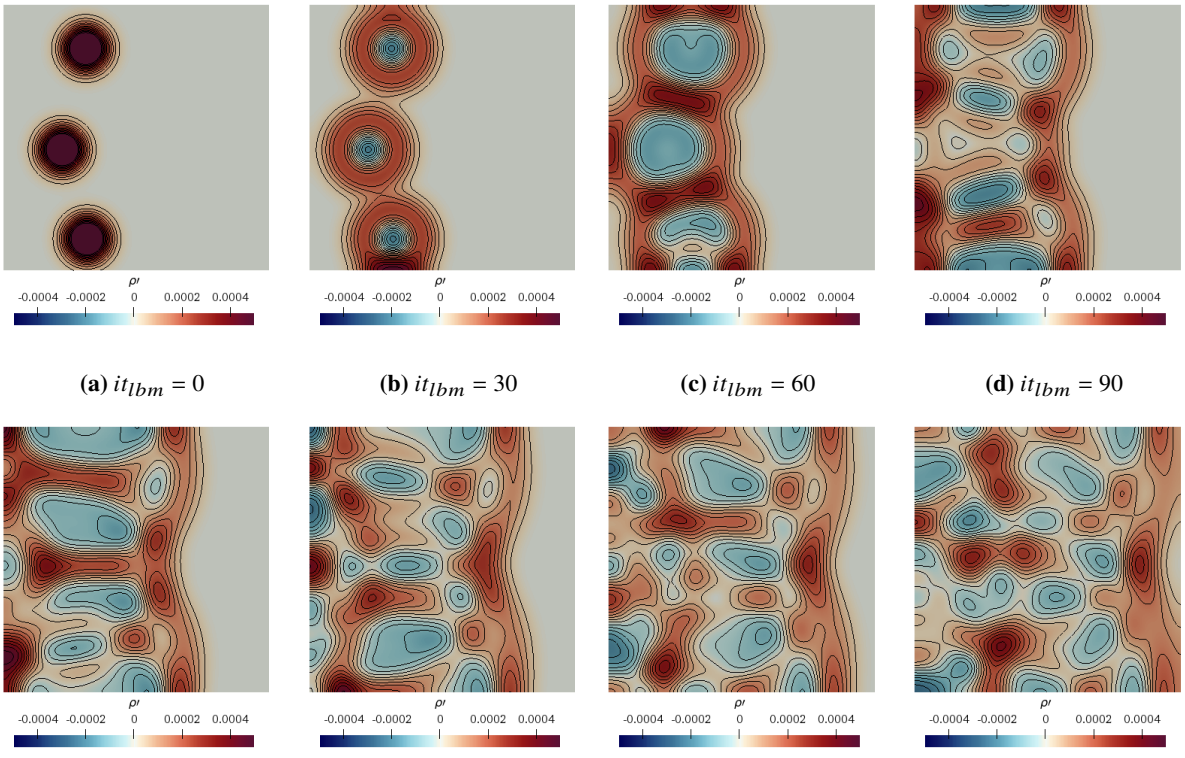
The mean-square error functions remain the classical choice for loss in regression problems. This choice supposes that data is drawn from a Gaussian distribution.Therefore if a multi-modal data distribution is employed, results will tend towards the weighted average between the probability modes. The gradient loss penalizes this feature of the mean-square error loss and forces the output distribution to follow the multi-modal behavior of the target data, and as a result, sharpens the predicted fields.

The choice of weighting parameters λ_{l2} and λ_{gdl} will be carried out in the result section, along with a comparison between a model trained only with the mean-square error loss and one combining both MSE and GDL.

III. Dataset generation

The training dataset used for this study is composed of 2D domains with hard reflecting acoustic walls, no obstacles and gaussian pressure pulses as initial conditions.

A lattice-boltzmann method (LBM) code [27, 28] is used to generate such data. These methods achieve second-order spatial accuracy, yet resulting in a small numerical dissipation similar to a 6^{th} order optimized Navier-Stokes schemes [29], making LBM highly suitable for aeroacoustic predictions [30]. However classical BGK-LBM collision models can produce high frequency instabilities [31] when viscosity is low. As a consequence, a recursive and regularized BGK model (rrBGK-LBM) [32] is applied to maintain code stability with low numerical dissipation.



(a) $it_{lbm} = 0$ **(b)** $it_{lbm} = 30$ **(c)** $it_{lbm} = 60$ **(d)** $it_{lbm} = 90$

(e) $it_{lbm} = 120$ **(f)** $it_{lbm} = 150$ **(g)** $it_{lbm} = 180$ **(h)** $it_{lbm} = 201$

Fig. 2 Example of dataset generation. Each inital condition $(it = 0)$ is stepped 231 LBM iterations in total. Density fields are saved every 3 LBM iterations.

A. Numerical setup: Propagation of Gaussian pulses in a closed domain

The first dataset consists of 500 2D-simulations, with hard walls imposed on the four domain boundaries (Fig. 2). Reflecting walls are modelled as classical half-way bounce-back nodes [33]. Density fields are initialized with a random number (varying between one and four) of Gaussian pulses located at random domain locations. The total density of a Gaussian pulse is defined as

$$\rho(x, y, t = 0) = \rho_0 + \varepsilon \exp\left[-\frac{\ln 2}{b^2}\left((x - c_x)^2 + (y - c_y)^2\right)\right] \tag{5}$$

where ε is the pulse amplitude, b is the pulse half-width, c_x and c_y are the x- and y-coordinates of the pulse center. All units are dimensionless. The reference density is chosen as $\rho_0 = 1$. The computational domain is $D^2 = [0, 100] \times [0, 100]$, each direction being discretized with 200 nodes. The parameters ε and b/D are fixed and equal to 0.001 and 6 respectively, such that 12 lattice nodes discretize the initial pulse half-width (this choice is justified in section III.B). As the background density $\rho_0 = 1$, the fluctuating density ρ' is chosen such that $\rho_0 \gg \rho'$ in order to avoid non-linear effects. Viscosity is set to $\nu = 0.001$ as nil values of this quantity could affect the numerical stability of LBM schemes.

A LBM time-step of $\Delta t_{LBM} = 7.39\ 10^{-3}$ is chosen. A normalized time is defined as $\tau = tc_0/D$, where $c_0 = 340$ is the velocity of sound. This time τ corresponds to the propagation time of an acoustic wave from one boundary to the other. Each training simulation is stopped at $\tau = 1.707$ and density fields are recorded at time-steps of $\Delta\tau = 0.022$. Note that the latter time-step is the one used by the CNN (in this particular case, 3 times the LBM time-step). It suggests that CNN could achieve significant speed-ups compared with LBM by performing low error predictions at arbitrarily large timesteps. The computed LBM fields fields are then packed into groups of 4+1 frames (input+target) for the CNN training. Snapshots from a complete LBM run are shown in Fig. 2.

B. Validation and test cases for training

In order to demonstrate the dissipation behavior of the rrBGK-LBM solver, a benchmark simulation is performed by comparing the free-field propagation of a single density gaussian pulse with the analytical solution, given by a zero-order Bessel function J_0 [34]:

$$\rho(x, y, t) = \rho_0 + \frac{\varepsilon}{2\alpha}\int_0^\infty \exp(-\xi^2/(4\alpha))\cos(\xi t)J_0(\xi\eta)\xi d\xi) \tag{6}$$

where $\alpha = \log 2/b^2$ and $\eta = x^2 + y^2$ represents the radial coordinate in absence of mean flow. A schematic of the problem setup is shown in Fig. 3a.

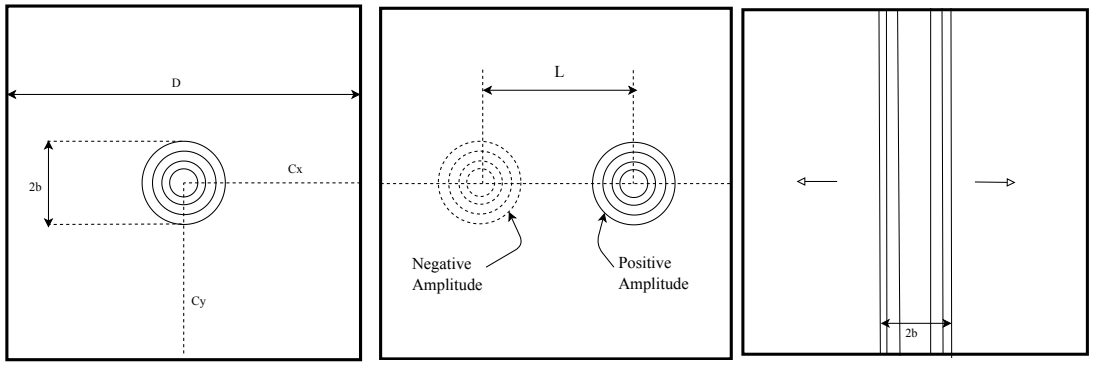
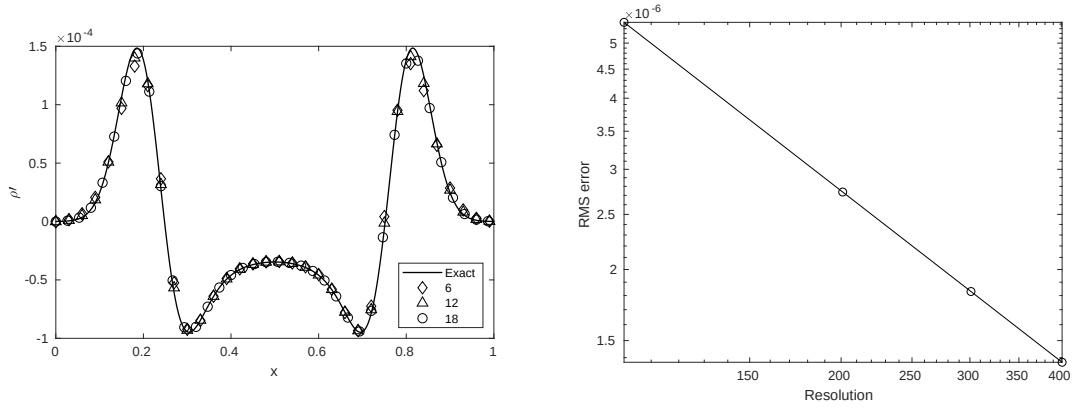


Fig. 3 Schematic of the three test cases. Domain size is $D \times D$. (a) propagation of single Gaussian Pulse of half-wifth b, (b) Two Pulses with opposed initial amplitudes and (c) Plane wave with Gaussian profile propagating in x-direction.

Fig. 4a shows a slice of the density fluctuations in the line $y/D = 0.5$ at time $\tau = 0.26$ before the pulse impinges on any wall. Comparison is made between the analytical solution and various simulations performed with various



(a) **(b)**

Fig. 4 Propagation of Gaussian pulse with initial parameters $\varepsilon = 0.001$ and $b = 6$: (a) Comparison of the cross profiles of density fluctuations $\rho\prime$ for different resolutions of at slice $y/D = 0.5$ and dimensionless time $\tau = 0.26$. The legend indicates the number of lattice points per initial Gaussian pulse half-width $\frac{b}{\Delta x}$. (b) Evolution of the rms error with the lattice resolution.

resolutions $b/\Delta x = 6$, 12 and 18. Results show that at least 12 points per half-width are necessary to capture the propagation of the pulse accurately with a small numerical dissipation error. Fig. 4b shows the evolution of the error with respect to the lattice resolution.

This validation case will also be used for the validation of the Neural Network predictions. Henceforth this single Gaussian pulse case is referred to as "**Test A**". It is the baseline case of the present study, as it is very similar to the training configurations.

In addition to this validation case, two more cases are created for evaluating the network performance:

- **Test B**: Double Gaussian pulse, with opposed initial amplitudes. One pulse is located at $\left(c_x, c_y\right)_1 = (D/2 - L/2, D/2)$ with amplitude $\varepsilon_1 = -\varepsilon$, the other at $\left(c_x, c_y\right)_2 = (D/2 + L/2, D/2)$ with amplitude $\varepsilon_2 = \varepsilon$. A schematic is shown in Fig. 3b. This test aims at verifying the network generalization ability, *i.e.* the capacity to extrapolate a solution when the input falls outside the training space. This test remains nonetheless close to the training data distribution.
- **Test C**: One-dimensional Gaussian plane wave, propagating in the x-direction, as shown in Fig.3c. This configuration is a challenging generalization case since it is very different from the combination of radially expanding waves addressed by the training. Indeed, Sorteberg *et al.* [17] showed that "plane wave" cases were difficult to predict when training their network with only cylindrical pulses.

IV. Acoustic wave propagation by a Deep Convolutional Network

This section studies the performance of the Multi-Scale neural network trained with the dataset described in Section III.A, by comparing the resulting predictions for test cases A and B with the corresponding LBM predictions, taken as "Target" data (as usually defined by the Machine Learning community). For case C, treated in Section IV.E, the target solution employed for comparison is the analytical solution for a 1D propagating Gaussian pulse.

A. Training parameters

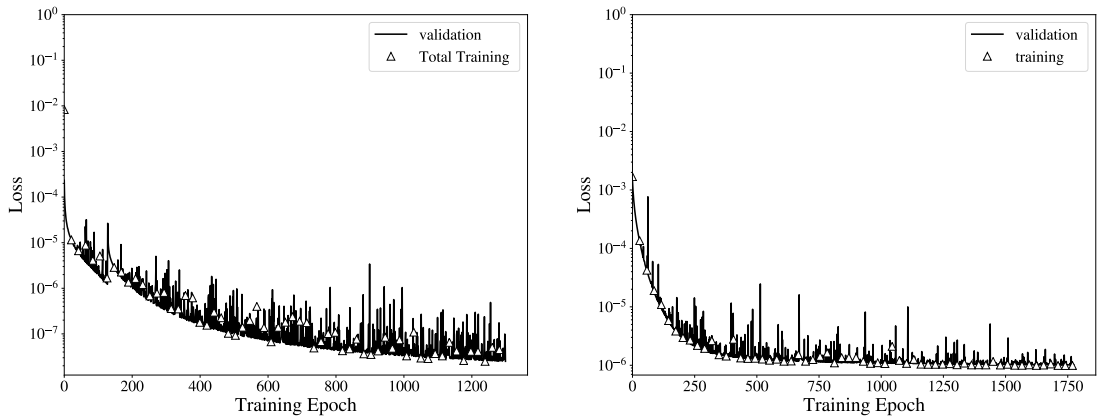
Two types of training strategies are taken into consideration. Training T_1 is carried out by setting the loss function as a mean-square error such that $\lambda_{l2} = 1$ and $\lambda_{gdl} = 0$. Training T_2 sets $\lambda_{l2} = 0.02$ and $\lambda_{gdl} = 0.98$, combining mean-square error and gradient loss functions. The ratio $\lambda_{l2}/\lambda_{gdl}$ is set to obtain similar values of the two parts of the loss function, so that both mean and spatial gradient errors contribute equally to the parameter update during the optimization of the neural network. Recent works [35] have demonstrated the importance of balancing the contributions from the different loss terms in order to increase the accuracy and stability of the neural network training process.

400 simulations from the dataset are employed for training with 77 density fields per simulation at consecutive

timesteps. Training samples are used to calculate losses in order to optimize the network by modifying its weights and biases. The remaining 100 simulations are used for validation purposes: no model optimization is performed with this data. The validation is only used to ensure that a similar accuracy is achieved by the neural network on data unseen during the training phase. During the training, the dataset is processed in batches to produce predictions (forward pass) followed by the update of network weight parameters θ (backward pass to minimise the error). A complete cycle through all dataset samples is named an epoch. This process is repeated until the loss is converged. Training samples are shuffled at each epoch in order to increase data variability. Data points (input and target fields) are also rotated randomly at four right angles in order to perform a second type of data augmentation.

The Adam stochastic optimization algorithm [36] is used, with an initial learning rate set to 1×10^{-4} and a 2% reduction each time the loss value stagnates for more than 10 epochs.

Figure 5 shows the evolution of both training and validation dataset errors during the optimization run. Network weights are optimized until convergence. For T_2, convergence on both components of the loss is reached. These two trainings provide two optimized neural networks that can be now tested on the three benchmarks proposed in Section **??**. Although the error is lower for the MSE optimization, the GDL training is expected to be more robust for configurations exhibiting sudden amplitude changes such as wave reflections against hard walls.



(a) **(b)**

Fig. 5 Evolution of total losses for training samples and validation samples for (a) Training T_1 ($\lambda_{l2} = 1$, $\lambda_{gdl} = 0$) and (b) Training T_2 ($\lambda_{l2} = 0.98$ and $\lambda_{gdl} = 0.02$).

B. Test case A: Gaussian Pulse

Resulting fields for test case A, corresponding to a centered Gaussian pulse left propagating through the closed domain, are shown in Fig. 6 for both T_1 and T_2 for several dimensionless times $\tau = \frac{tc_0}{D}$ where c_0 is the speed of sound and D is the domain length. Note that a single timestep performed by the neural network corresponds to three hundred LBM iterations $(\Delta t)_{NN} = 3(\Delta t)_{lbm}$. Snapshots of the density fluctuations along the line $y/D = 0.5$ are plotted for the target, T_1 and T_2 on Fig. 7 at 18 representative times of the simulation. Good agreement is found between the reference field and predictions up to dimensionless times around $\tau \simeq 1.22$ for both networks.

The initial Gaussian pulse spread is well captured (first row in Fig. 7). Both wall-pulse interactions ($\tau \sim 0.5$) and pulse-pulse interactions ($\tau \sim 1$, after wall reflection) seem to be well predicted. Although the network was trained to predict fields one single time-step ahead from the four input data fields, the recursive prediction of outputs does not perturb the predictions significantly until late times ($\tau = 1.22$ corresponding to 136 recursive predictions). Both networks seem capable of predicting both the mean level and the spatial gradients accurately, without introducing significant numerical dissipation or dispersion.

For times greater than $\tau = 1.22$, a uniform drift in density level is observed. This drift is more significant in training T_1 than T_2. This suggests that spatial gradients continue to be accurately predicted and that this drift is mostly homogeneous. As the inputs are the outputs from previous predictions, the neural network can no longer correct itself once the inputs are already shifted. This is a typical error amplification effect encountered with iterative

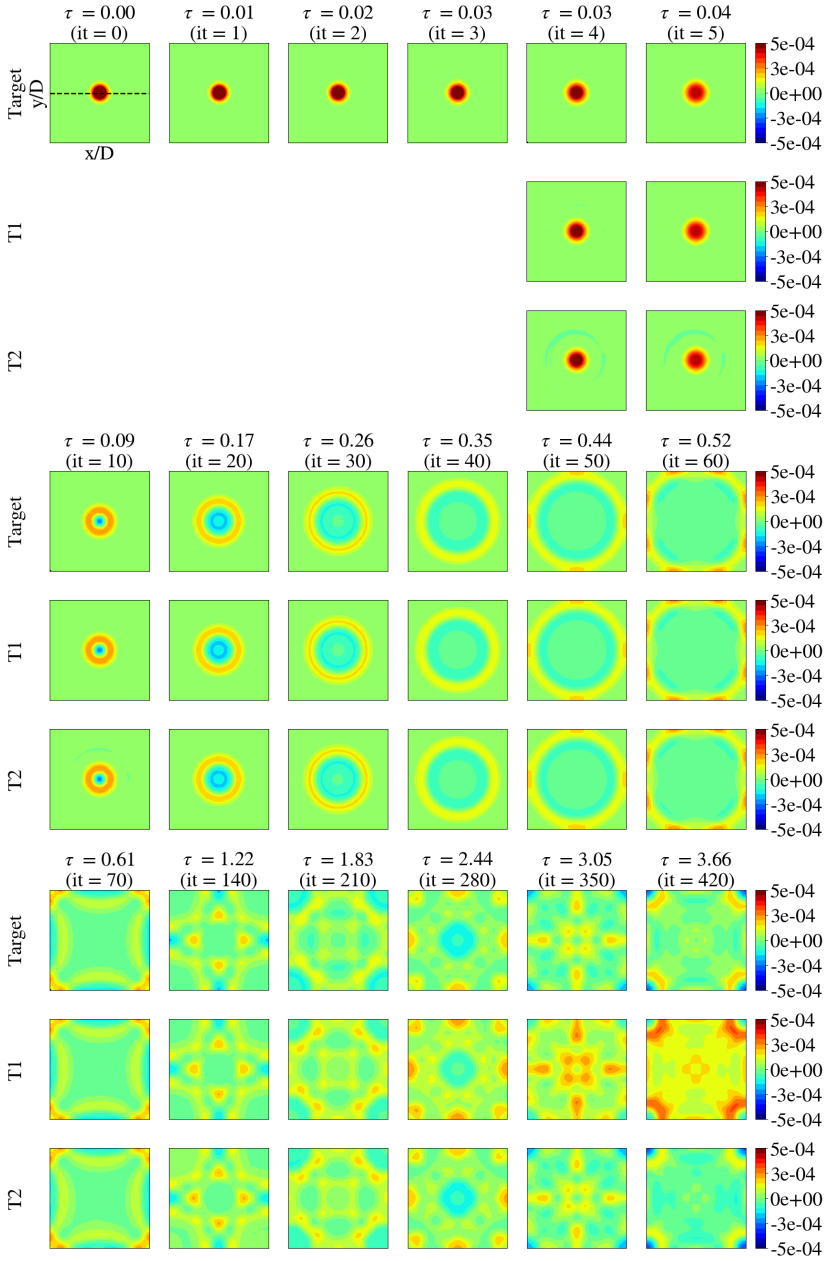


Fig. 6 Results for training T_1 and T_2, compared with LBM target data and test case A (pulse) for dimensionless time $\tau = \frac{t c_0}{D}$.

algorithms. Training T_2 suffers also from this mean level drift, even though the amplitude error remains smaller than in T_1. Furthermore, symmetry of the density field is lost from $\tau = 1.22$ onward, as seen in Fig. 6 (third row). This indicates that errors tend also to accumulate in the prediction of spatial gradients, even if this error remains small. The evolution of the root mean square error relative to the maximum spatial values of the density target ρ' and gradient of the density $\nabla\rho'$ is plotted in Fig. 8 and shown in black. These errors are defined as:

$$RMSE(\hat{\rho\prime}) = \sqrt{\frac{1}{N}\sum^{N}(\hat{\rho\prime} - \rho\prime)^2} \quad \text{and} \quad rRMSE(\hat{\rho\prime}) = \frac{RMSE(\hat{\rho\prime})}{\max(\rho\prime)} \tag{7}$$

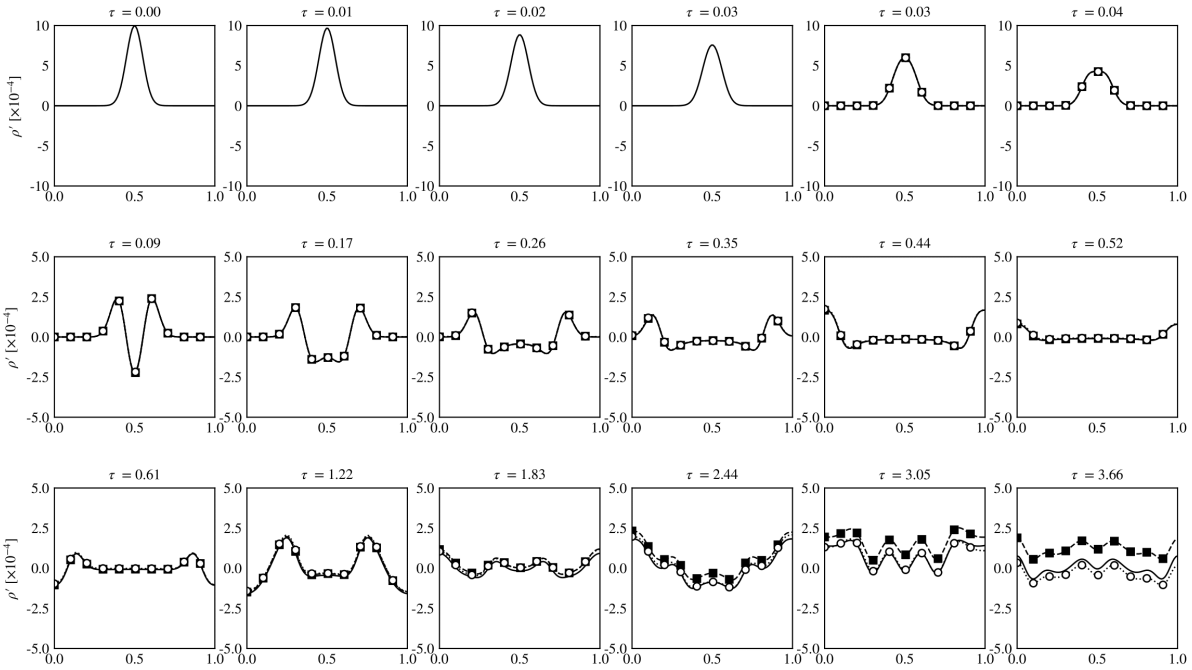


Fig. 7 Slice of density field at $y/D = 0.5$ for Test Case A (pulse). First four images represent input fields, then predictions are performed recursively. (——) Target data (LBM), (-■-) training T_1 (loss in MSE) and (-○-) training T_2 (loss in MSE+GDL)

and:

$$RMSE(\nabla\hat{\rho}\prime) = \sqrt{\frac{1}{N}\sum^{N}||\nabla\hat{\rho}\prime - \nabla\rho||_2^2} \quad \text{and} \quad rRMSE(\nabla\hat{\rho}\prime) = \frac{RMSE(\nabla\hat{\rho}\prime)}{\max(\nabla\rho\prime)} \tag{8}$$

where $||.||_2$ represents the L2 norm.

Both networks show a similar behavior: the error starts at very low levels ($\sim 10^{-4}$) for mean-square density error, corresponding to the converged loss found during training for the validation samples. Then the error grows abruptly, until reaching 3% of the maximum density value for the MSE. The relative error remains however below 5% for times up to $\tau = 0.5$ for $T1$ and $\tau = 1.0$ for $T2$.

For longer times ($\tau \geq 0.5$), the relative error in the density fields keeps growing steadily up to values close to 1 in the case of $T1$, *i.e.* the relative error is of the order of the signal itself. This behavior comes mostly from the uniform drift, as the errors on field gradients are one order of magnitude lower than errors on the density field.

As a way to demonstrate the previous statement, an *a posteriori* correction based on the acoustic energy conservation is used to improve the neural network predictions in the following section.

Note that this *a posteriori* correction will modify the current prediction $\hat{\rho}'(t)$ to match some characteristics of the target field $\rho'(t)$. Yet, this latter quantity is not known during inference, and therefore cannot be used: the correction should be derived only using the current network prediction, and possibly the initial target field only, *i.e.* $\rho'(t = 0)$, which is known. Consequently, a link between $\rho'(t = 0)$ and $\rho'(t)$ is necessary to propose such a correction. In the present study, an energy conservation is proposed to relate these two quantities, and generate a physics-based *a posteriori* correction of the neural network, called "Energy Preserving Correction" (EPC).

C. Energy preserving correction (EPC)

The acoustic propagation takes place in a linear regime, *i.e.* the wave is not dissipated and the LBM viscosity is low in order to preserve that hypothesis. The acoustic energy follows the conservation law in integral form:

$$\frac{d}{dt}\int_\Omega E dx + \int_{\partial\Omega} \mathbf{I}\cdot\mathbf{n} d\sigma = -\int_\Omega \mathcal{D} dx \tag{9}$$

where E represents the acoustic energy density and $\mathbf{I}$ the acoustic intensity:

$$\begin{aligned} E &= \frac{p'}{2\rho_0 c_0^2} + \frac{\rho_0 v'^2}{2} \\ \mathbf{I} &= p'\mathbf{v'}. \end{aligned} \tag{10}$$

Such relationships are only valid for a uniform isentropic field at rest ($M = 0$). The dissipation $\mathcal{D}$ is nil. The mean energy flux across a closed domain containing no sources is zero. Since velocity fluctuations are related to pressure fluctuations *via* the Euler equation and the density fluctuations are proportional to pressure fluctuations, the total energy conservation yields the total density conservation. Assuming a positive uniform drift, let ϵ (≥ 0) be the correction constant for the density field, such that:

$$\int_\Omega (\hat{\rho}'(t) - \epsilon)^2 dx = \int_\Omega \rho'(t=0)^2 dx \tag{11}$$

By symmetry of the problem:

$$\int_\Omega (\hat{\rho}'(t))^2 dx = \int_\Omega (\rho'(t=0) + \epsilon)^2 dx \tag{12}$$

Define

$$\begin{aligned} \hat{P} &= \frac{\int_\Omega \hat{\rho}'(t) dx}{\int_\Omega dx} \\ P_0 &= \frac{\int_\Omega \rho'(t=0) dx}{\int_\Omega dx}. \end{aligned} \tag{13}$$

By substracting Eqs. (12) in (11) yields:

$$\int \epsilon^2 dx - 2\int \rho'(t)\epsilon dx = -\int \epsilon^2 dx - 2\int \rho'(t=0) dx \tag{14}$$

Since the drift was found uniform, ϵ can move away of the integrals, leading to the second order equation:

$$\epsilon^2 - \epsilon\hat{P} + \epsilon P_0 = 0 \tag{15}$$

This equation has the null solution $\epsilon = 0$, which could correspond to no correction of the predicted field, and the following non-trivial solution is found for the correction:

$$\epsilon = P_0 - \hat{P} \tag{16}$$

Equation (16) reveals that the intuitive correction of the drift obtained by removing the mean difference between the current prediction and the initial target field, corresponds actually to the energy conservation in time.

In the following, this EPC is implemented, and the accuracy gain is evaluated on the several benchmark cases. The EPC is applied to both networks T_1 and T_2 at each recursive prediction, effectively creating two new corrected networks T_1^C (*i.e.* mse loss coupled with *a posteriori* correction) and T_2^C (*i.e.* training with mse and gradient loss and *a posteriori* correction). The error evolution for test A is shown in Fig. 8, in red color. As expected, the evolution of the gradient error is identical with or without correction (Fig. 8b). Such behaviour demonstrates that the correction has no effect on the error made by the neural network on spatial gradients, while it improves the error on the mean density level significantly (Fig. 8a). Interestingly, network T_2^C (*i.e.* using both gradient loss and MSE during training) has a greater error than T_1^C between $\tau = 0.2$ and $\tau = 1.2$, even though it remains around the 10% threshold. For longer times, T_1^C and

T_2^C errors follow similar trends and levels. This seems to suggest that overall, both networks perform relatively similarly when the *a posteriori* correction is applied, and that accurate solutions are found even for times not seen during the training (only times samples for $\tau \leq 1.7$ were present in training data).

It can be concluded that both neural networks are able to predict the propagation of a single pulse recursively, which is a very similar case to the training ones. With EPC, the long-time error is reduced by one order of magnitude. This highlights the high capability of a physics-informed neural network to reproduce physics compared with standard data-driven only methods. Next sections will discuss the ability of the network to predict propagation of acoustic waves with initial conditions that have *not* been sampled during the training process. It will thus be checked if and how well the network is able to effectively learn the underlying Green's function for other types of initial conditions than those of the training.

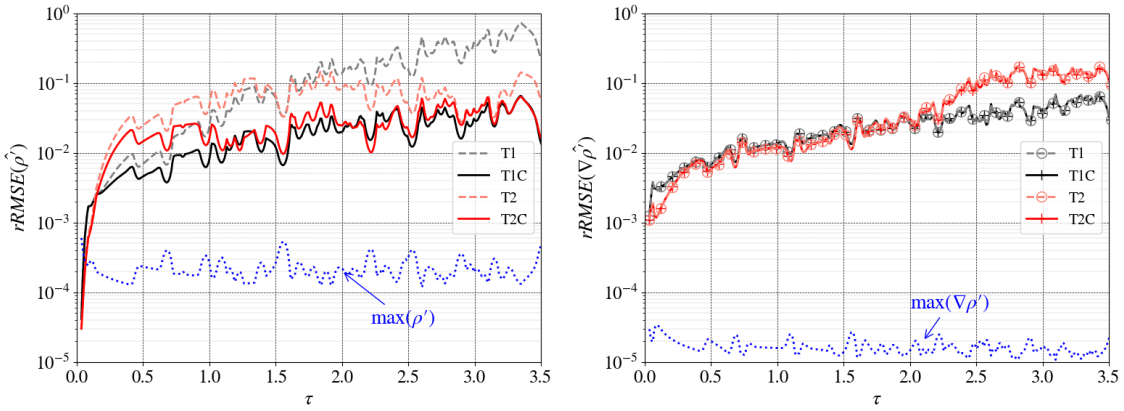


Fig. 8 Comparison of error evolution for trainings $T1$, $T2$, $T1C$ and $T2C$ for test Case A (pulse). (a) Rms error of density relative to the maximum value of density $\max(\rho\prime)$ at each time-step (in blue) and (b) rms error for the sum of density gradients relative to $\max(\nabla_x\rho\prime, \nabla_y\rho\prime)$ at each time-step (in blue).

D. Test case 2A: Opposed Gaussian pulses

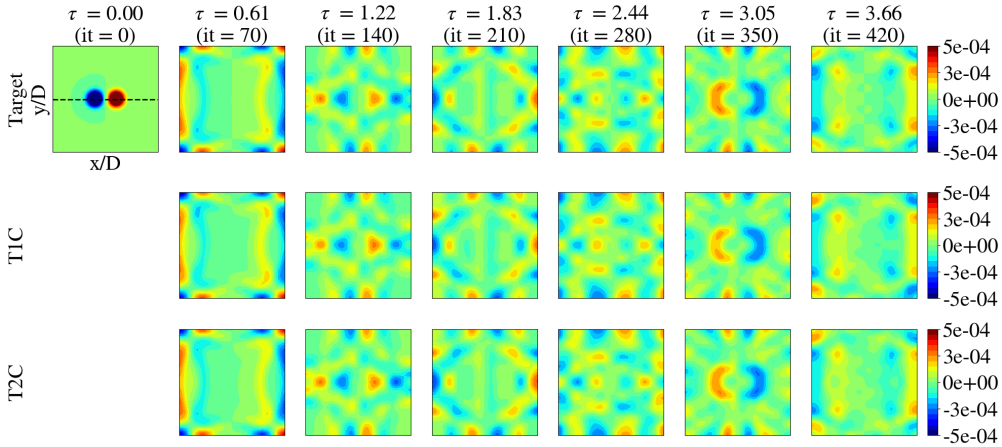


Fig. 9 Results for training T_1^C and T_2^C and test case B (two opposed pulses) for dimensionless time $\tau = \frac{tc_0}{D}$

The second test case consists of two Gaussian pulses, with opposed initial amplitudes, propagating in a closed domain. Trainings T_1^C and T_2^C (EPC neural networks) are evaluated. Figure 9 displays snapshots of density fields for both networks. Snapshots of the density fluctuations along the line $y/D = 0.5$ are plotted for the target T_1^C and T_2^C onf

Fig. 10 for the same dimensionless times as in previous figures, and Figure 11 shows the error evolution. Network T_2^C continues to predict both evolutions accurately for times up to $\tau \sim 1.2$ or even further. It shows a clear advantage over T_1^C as mean square errors are systematically lower for networks using gradient loss during training. In fact, for T_2^C, errors remain below the 10% threshold, whereas for training T_1^C some peaks with a 30% error relative to the *rms* density value are found before $\tau = 3.0$. As seen in the density fields (Fig. 9), at these times wave signals exhibit complex spatial patterns, with many local extrema over short distances, thus making the gradient prediction harder for a network that has not seen those patterns during training. Both networks continue nonetheless to provide overall accurate gradient predictions, and manage to capture both pulse-pulse and pulse-wall interactions (two front-waves adding or subtracting their amplitudes during a short period of time). In fact, Fig. 11 shows bumps in the gradient error appearing periodically, which correspond to these strong interactions phases. It seems that, although the error tends to grow, the RMSE of gradients tends to recover after these interactions, and grows again at the next one. This phenomenon might be attributed to the strong unsteadiness appearing during such events, which have not been frequently sampled during training.

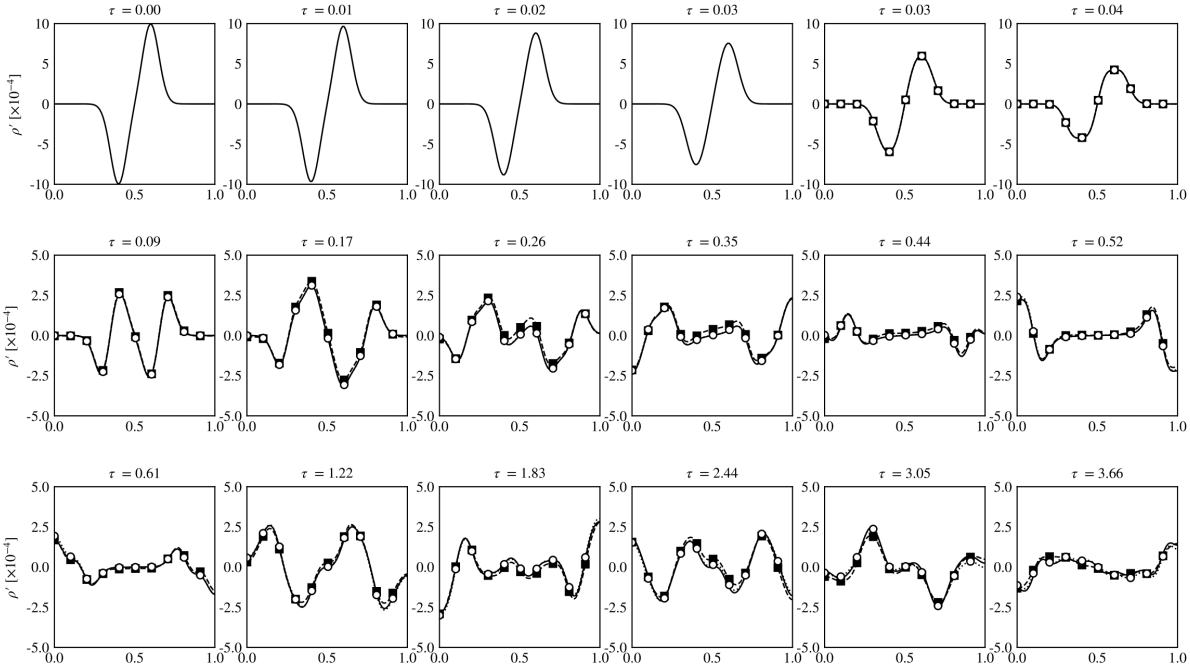


Fig. 10 Slice of density field at $y/D = 0.5$ for Test Case 2A (two opposed pulses). Four first images represent input fields, then predictions are performed recursively. (——) Target data (LBM), (-■-) training T_1^C (loss in MSE + correction) and (-○-) training T_2^C (loss in MSE+GDL+correction)

E. Test case C: Plane Gaussian pulse

A third test case is studied with both networks, where initial conditions correspond to the analytical solution for the one-dimensional propagation of a Gaussian pulse, propagating on both directions along the x-direction. This solution is then extruded into the y-direction, in order to obtain 200×200 grids as for the training data and fed into the neural network. Figures 12 and 13 show respectively the evolution along the x-axis at at $y/D = 0.5$ and the associated error. This case differs from the previous ones in that no complex pattern should appear, just two plane Gaussian pulses bouncing on the walls and interacting at the domain center. Since the networks were trained exclusively with cylindrical pulses, the major challenge for the network lies in its ability to understand that the plane-wave pulse must remain straight and coherent. As already mentioned in section III.B, this was reported as the critical difficulty of the LSTM-CNN approach proposed by Sorteberg *et al.* [17] for seismic waves. Good agreement is found for times below $\tau = 1.22$, which correspond to the free-field initial propagation and the first wall interaction. All wall reflections are clearly marked in the error curves by the sudden variation of the relative error. The local error diminution is mostly due to a

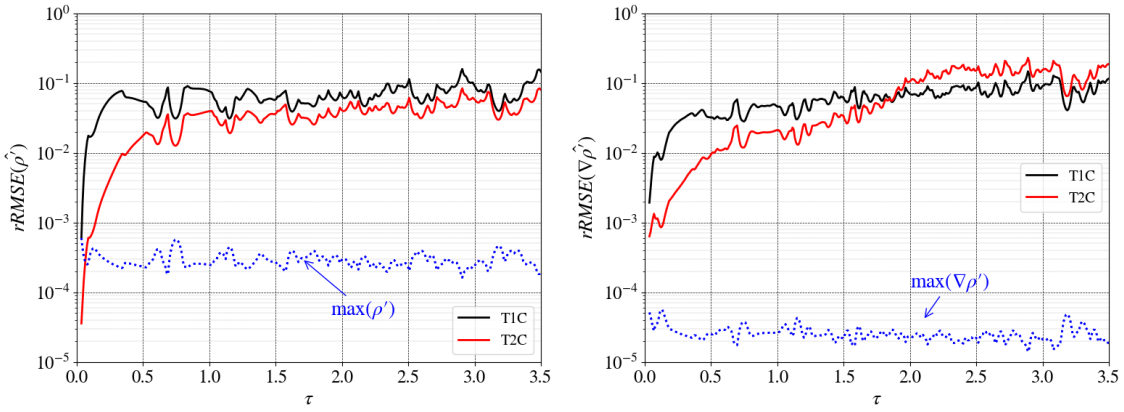


Fig. 11 Comparison of error evolution for trainings $T1C$ and $T2C$ for test Case B (two opposed pulses). (a) Rms error of density relative to the maximum value of density $\max(\rho\prime)$ at each time-step (in blue) and (b) rms error for the sum of density gradients relative to $\max(\nabla_x\rho\prime, \nabla_y\rho\prime)$ at each time-step (in blue).

sharp increase of the maximum value related to wall impingement, while the absolute error maintains its value. T_1^C shows a beginning of mean level drift, preceded by an increase of the gradient error (visible for example for $\tau = 1.83$). Later times show the propagation of accurate gradient predictions, while it is clear that the network tries to maintain good mean density levels.

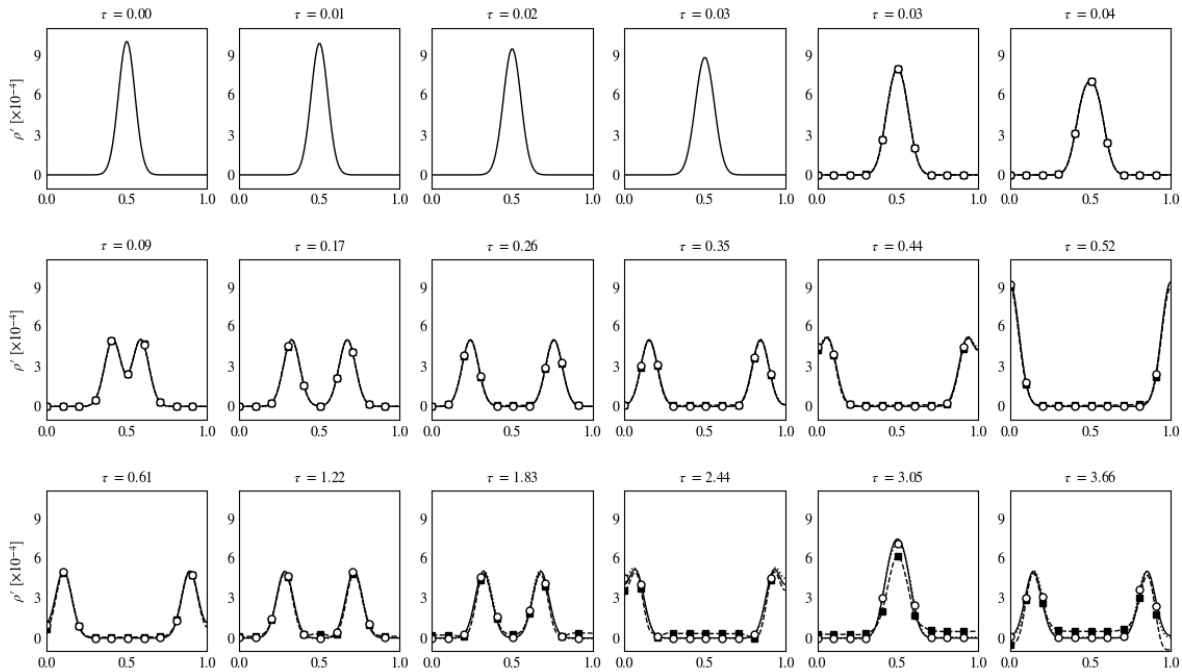


Fig. 12 Slice of density field at $y/D = 0.5$ for Test Case 3A (plane Gaussian pulse). First four images represent input fields, then predictions are performed recursively. (——) Target data (LBM), (-■-) training T_1^C (loss in MSE + correction) and (-o-) training T_2^C (loss in MSE+GDL+correction)

The Neural Network manages to maintain the straight wavefront, as shown in Fig. 14, although such dynamics were

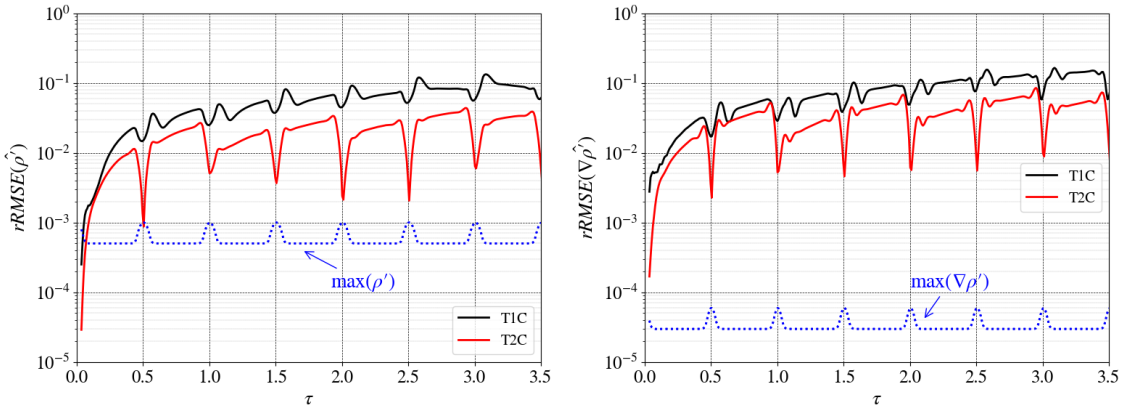


Fig. 13 Comparison of error evolution for trainings $T1C$ and $T2C$ for test Case C (plane wave). (a) Rms error of density relative to the maximum value of density $\max(\rho\prime)$ at each time-step (in blue) and (b) rms error for the sum of density gradients relative to $\max(\nabla_x\rho\prime, \nabla_y\rho\prime)$ at each time-step (in blue).

not present in the training set. This demonstrates the CNN ability to successfully extrapolate to new data and to capture the underlying physics of wave propagation. It should be however noted that other important parameters were kept constant (pulse spatial resolution, lattice resolution), thus the generalization capability of the network has only been studied in the sense of changing the topology of initial conditions.

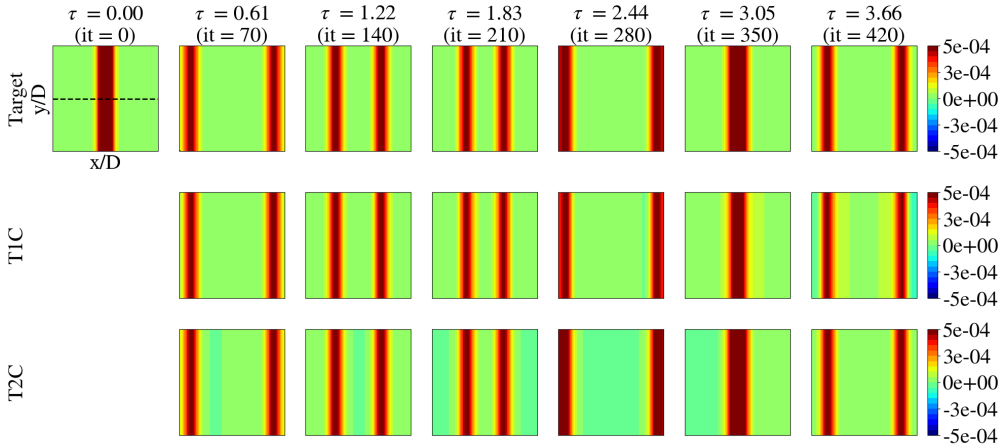


Fig. 14 Results for training T_2^C and test case C (plane pulse) for dimensionless time $\tau = \frac{tc_0}{D}$.

V. Conclusion

A method for predicting propagation of acoustic waves is presented in this work, based on the implementation of a Multi-Scale convolutional neural network trained on LBM-generated data. Training samples consist of variations around the classical benchmark of the propagation of 1 to 4 2D Gaussian pulses. Two types of training strategies are studied through the variation of loss functions. An *a posteriori* energy-preserving correction (EPC) is proposed to increase the accuracy of the predictions. Both networks are then evaluated with initial conditions unseen during the training process, as a way to test the generalization performance. An increased accuracy is shown by the network trained with a combination of penalizations on the mean square errors of both the density and its spatial gradients, even for the challenging plane wave case. In all cases, the EPC correction yields a significant accuracy gain, at least one order of magnitude compared to pure neural network predictions. This exemplifies the benefits of physics-informed neural networks compared with pure data-driven methods. Here, the EPC allows such an *a posteriori* correction, that is without having to retrain the neural network, which makes this correction flexible, and can be adapted depending on the case and physics to be solved. Finally, the proposed Multiscale net with EPC revealed its ability to predict an unsteady phenomenon well beyond the duration of the training. Moreover, neural networks were predicting flow fields every 3 LBM timesteps. Further investigations should study their ability to predict at even larger time-steps. Such strategies could probably mitigate classical limitations of standard numerical methods such as the CFL constraint, opening the path to fast and efficient tools for acoustic propagation.

Acknowledgments

This work was partly supported by the french « Programme d'Investissements d'avenir » ANR-17-EURE-0005 conducted. We also acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

[1] Lele, S. K., and Nichols, J. W., "A second golden age of aeroacoustics?" *Philos. Trans. R. Soc. A Math. Phys. Eng. Sci.*, Vol. 372, No. 2022, 2014.

[2] Jacob, M. C., Dragna, D., Cahuzac, A., Boudet, J., and Blanc–Benon, P., "Toward Hybrid CAA with Ground Effects," *Int. J. Aeroacoustics*, Vol. 13, No. 3-4, 2014, pp. 235–259.

[3] Schoenwald, N., Panek, L., Richer, C., and Thiele, F., "Investigation of sound radiation from a scarfed intake by CAA-FWH simulations using overset grids," *13th AIAA/CEAS Aeroacoustics Conf. (28th AIAA Aeroacoustics Conf.*, 2007.

[4] Sanjose, M., Towne, A., Jaiswal, P., Moreau, S., Lele, S., and Mann, A., "Modal analysis of the laminar boundary layer instability and tonal noise of an airfoil at Reynolds number 150,000," *Int. J. Aeroacoustics*, Vol. 18, No. 2-3, 2019, pp. 317–350.

[5] Ffowcs Williams, J. E., and Hawkings, D. L., "Sound generation by turbulence and surfaces in arbitrary motion," *Philos. Trans. R. Soc. London. Ser. A, Math. Phys. Sci.*, Vol. 264, No. 1151, 1969, pp. 321, 342.

[6] Singer, B. A., Lockard, D. P., and Lilley, G. M., "Hybrid Acoustic Predictions," *Comput. Math. with Appl.*, Vol. 46, No. 4, 2003, pp. 647–669.

[7] Schönwald, N., Panek, Ł., Richter, C., and Thiele, F., "Mode propagation in bifurcated bypass ducts: Application oriented simulation approach," *16th AIAA/CEAS Aeroacoustics Conf. (31st AIAA Aeroacoustics Conf.*, 2010.

[8] Pérez Arroyo, C., Leonard, T., Sanjosé, M., Moreau, S., and Duchaine, F., "Large Eddy Simulation of a scale-model turbofan for fan noise source diagnostic," *J. Sound Vib.*, Vol. 445, 2019, pp. 64–76.

[9] Goldstein, M. E., *Aeroacoustics*, McGraw-Hill International Book Co New York, 1976.

[10] Hornik, K., Stinchcombe, M., and White, H., "Multilayer feedforward networks are universal approximators," *Neural Networks*, Vol. 2, No. 5, 1989, pp. 359–366.

[11] Brunton, S. L., Noack, B. R., and Koumoutsakos, P., "Machine Learning for Fluid Mechanics," *Annu. Rev. Fluid Mech.*, Vol. 52, No. 1, 2020, pp. 477–508.

[12] Bhatnagar, S., Afshar, Y., Pan, S., Duraisamy, K., and Kaushik, S., "Prediction of aerodynamic flow fields using convolutional neural networks," *Comput. Mech.*, Vol. 64, No. 2, 2019, pp. 525–545.

[13] Tompson, J., Schlachter, K., Sprechmann, P., and Perlin, K., "Accelerating Eulerian Fluid Simulation With Convolutional Networks Jonathan," *Proc. 34th Int. Conf. Mach. Learn.*, 2017.

[14] Yang, C., Yang, X., and Xiao, X., "Data-driven projection method in fluid simulation," *Comput. Animat. Virtual Worlds*, Vol. 27, John Wiley and Sons Ltd, 2016, pp. 415–424.

[15] Wiewel, S., Becher, M., and Thuerey, N., "Latent Space Physics: Towards Learning the Temporal Evolution of Fluid Flow," *Comput. Graph. Forum*, Vol. 38, No. 2, 2019, pp. 71–82.

[16] Zhu, W., Sheng, Y., and Sun, Y., "Wave-dynamics simulation using deep neural networks," , 2017.

[17] Sorteberg, W. E., Garasto, S., Pouplin, A. S., Cantwell, C. D., and Bharath, A. A., "Approximating the solution to wave propagation using deep neural networks," *32nd Conf. Neural Inf. Process. Syst. NIPS*, 2018.

[18] LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D., "Backpropagation Applied to Handwritten Zip Code Recognition," *Neural Comput.*, Vol. 1, No. 4, 1989, pp. 541–551.

[19] LeCun, Y., Bottou, L., Bengio, Y., and Haaner, P., "Gradient-Based Learning Applied to Document Recognition," *Proc. IEEE*, Vol. 86, No. 11, 1998, pp. 2278 – 2324.

[20] Krizhevsky, A., Sutskever, I., and Hinton, G. E., "ImageNet Classification with Deep Convolutional Neural Networks," *Commun. ACM*, Vol. 60, No. 6, 2017, pp. 84–90.

[21] LeCun, Y., Kavukcuoglu, K., and Farabet, C., "Convolutional networks and applications in vision," *ISCAS 2010 - 2010 IEEE Int. Symp. Circuits Syst. Nano-Bio Circuit Fabr. Syst.*, 2010, pp. 253–256.

[22] Nair, V., and Hinton, G. E., "Rectified Linear Units Improve Restricted Boltzmann Machines," *Proc. 27th Int. Conf. Mach. Learn.*, 2010, pp. 807–814.

[23] Rumelhart, D. E., Hinton, G. E., and Williams, R. J., "Learning representations by back-propagating errors," *Nature*, Vol. 323, No. 6088, 1986, pp. 533–536.

[24] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., Devito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S., "PyTorch: An Imperative Style, High-Performance Deep Learning Library," *Adv. Neural Inf. Process. Syst. 32*, 2019, pp. 8024–8035.

[25] Mathieu, M., Couprie, C., and LeCun, Y., "Deep multi-scale video prediction beyond mean square error," *6th Int. Conf. Learn. Represent.*, 2016.

[26] Ronneberger, O., Fischer, P., and Brox, T., "U-Net: Convolutional Networks for Biomedical Image Segmentation," *Med. Image Comput. Comput. Interv.*, Vol. 9351, 2015, pp. 234–241.

[27] Xu, H., and Sagaut, P., "Optimal low-dispersion low-dissipation LBM schemes for computational aeroacoustics," *J. Comput. Phys.*, Vol. 230, No. 13, 2011, pp. 5353–5382.

[28] Lallemand, P., and Luo, L. S., "Theory of the lattice Boltzmann method: Dispersion, dissipation, isotropy, Galilean invariance, and stability," *Phys. Rev. E*, Vol. 61, No. 6, 2000, pp. 6546–6562.

[29] Marié, S., Ricot, D., and Sagaut, P., "Comparison between lattice Boltzmann method and Navier-Stokes high order schemes for computational aeroacoustics," *J. Comput. Phys.*, Vol. 228, 2008, pp. 1056–1070.

[30] Brès, G. A., Pérot, F., and Freed, D., "Properties of the Lattice-Boltzmann method for acoustics," *15th AIAA/CEAS Aeroacoustics Conf. (30th AIAA Aeroacoustics Conf.*, 2009.

[31] Brogi, F., Malaspinas, O., Chopard, B., and Bonadonna, C., "Hermite regularization of the lattice Boltzmann method for open source computational aeroacoustics," *J. Acoust. Soc. Am.*, Vol. 142, No. 4, 2017, pp. 2332–2345.

[32] Malaspinas, O., "Increasing stability and accuracy of the lattice Boltzmann scheme: recursivity and regularization," *arXiv:1505.06900*, 2015.

[33] Ladd, A. J., "Numerical Simulations of Particulate Suspensions Via a Discretized Boltzmann Equation. Part 1. Theoretical Foundation," *J. Fluid Mech.*, Vol. 271, 1994, pp. 285–309.

[34] Tam, C. K., and Webb, J. C., "Dispersion-relation-preserving finite difference schemes for computational acoustics," *J. Comput. Phys.*, Vol. 107, No. 2, 1993, pp. 262–281.

[35] Wang, S., Teng, Y., and Perdikaris, P., "Understanding And Mitigating Gradient Pathologies In Physics-informed Neural Networks," *arXiv:2001.04536*, 2020.

[36] Kingma, D. P., and Ba, J. L., "Adam: A method for stochastic optimization," *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, International Conference on Learning Representations, ICLR, 2015.