Towards reconstruction of acoustic fields via physics-informed neural networks

Niebler, Korbinian; Bonnaire, Philip; Anh Khoa Doan, Nguyen; Silva, Camilo Fernando

Publication date
2022

Document Version
Final published version

Published in
Internoise 2022 - 51st International Congress and Exposition on Noise Control Engineering

Towards reconstruction of acoustic fields via physics-informed neural networks

Korbinian Niebler[1]
Technical University of Munich, School of Engineering & Design, Department of Engineering Physics and Computation
Boltzmannstr. 15, 85747 Garching, Germany

Philip Bonnaire[2]
Technical University of Munich, School of Engineering & Design, Department of Engineering Physics and Computation
Boltzmannstr. 15, 85747 Garching, Germany

Nguyen Anh Khoa Doan[3]
Faculty of Aerospace Engineering, Delft University of Technology
Kluyverweg 1, 2629HS Delft, Netherlands

Camilo Fernando Silva[4]
Technical University of Munich, School of Engineering & Design, Department of Engineering Physics and Computation
Boltzmannstr. 15, 85747 Garching, Germany

ABSTRACT

Acoustic measurements, obtained by microphones positioned at strategic places, are of great utility for the monitoring of a given acoustic system and for its protection in case large pressure fluctuations are measured. Such strategies are reliable as long as the microphones are properly positioned, which is not evident: in some cases the excited acoustic modes are not known beforehand. In this work, we proposed a method based on physics-informed neural networks (PINN) in order to reconstruct the entire acoustic field of a given acoustic element, when provided with only some acoustic measurements at some few locations. Such a method makes use of a feedforward neural network, where the loss function is taken as the residual of the acoustic wave equation. Such a residual is computed exploiting the automatic differentiation property of neural networks, in order to obtain the corresponding spatial and time derivatives. Additionally, the measurements of the aforementioned microphones are gathered and used also for the calculation of additional terms in the PINN loss function. By doing so, the most adequate acoustic state is obtained, which satisfies

[1]korbinian.niebler@tum.de

[2]philip.bonnaire@tum.de

[3]n.a.k.doan@tudelft.nl

[4]camilo.silva@tum.de

both measurements and the acoustic wave equation. In other words, the acoustic field within the system is reconstructed.

1. INTRODUCTION

After the proof that feedforward neural networks are universal approximators for arbitrary functions [1] and also their derivatives [2], research with neural networks has yielded enormous progress and significant results in a wide range of scientific disciplines. The studies in the fields of image recognition [3], natural language processing [4], cognitive science [5], and genomics [6] are representative of this development. However, when examining complex problems, gaining enough data for the training of the neural network presents a challenge for the state-of-the art machine learning techniques. Therefore the study of Raissi et al. [7] offers a great possibility to counteract the obstacles posed by a small-data regime. They introduced physics-informed neural networks (PINNs), which stand out due to a modification of the loss function. The well-known mean squared error (MSE) minimization of the training data is extended by an error function characterized by the residual of the governing partial differential equation (PDE) of the system. The embedding of the PDE is achieved by a powerful but underestimated computational tool, the automatic differentiation described in the study of Baydin et al. [8]. When dealing with a system, where only a small amount of data is available and where a physical law, especially a PDE, forms a constraint, the concept of physics-informed neural networks offers a promising chance to overcome the challenges posed by the physics and data characterizing these systems.

In our study we aim to analyse a system with the physical constraint defined by the one-dimensional acoustic wave equation:

$$\frac{\partial^2 p}{\partial t^2} - \bar{c}\frac{\partial^2 p}{\partial x^2} = 0. \tag{1}$$

Here, p denotes the acoustic pressure of the system, t the time, x represents the spatial variable and $\bar{c}$ stands for the (constant) speed of sound. We will use the presented concept of PINNs to reconstruct the pressure distribution in a given domain relying on only few training data (measurements). In a second step, we investigate how a given reflection coefficient or wall impedance associated with the system can be calculated in postprocessing.

2. DESCRIPTION OF THE PROBLEM

We define the domain of our problem as one-dimensional in space. The inlet of the domain is located at $x = 0$, where a forcing function is also applied. Continuously, this forcing inlet introduces disturbances into the domain. The outlet at $x = L = 1\text{m}$ is characterized by a specific wall reflection coefficient R, which may vary between 0 and 1. The training data for the neural network is generated through a custom numerical simulation. For the set-up of the simulation, instead of the wave equation, we implement the linearized mass and momentum equation as defined in [9]. In this work, we refer to this set of equations as the linearized acoustic equations:

$$\frac{1}{\bar{c}^2}\frac{\partial p}{\partial t} = -\bar{\rho}\frac{\partial v}{\partial x}, \tag{2}$$

$$\bar{\rho}\frac{\partial v}{\partial t} = -\frac{\partial p}{\partial x}. \tag{3}$$

In this set of equations, the constant ambient density $\bar{\rho}$ and speed of sound $\bar{c}$ are introduced. In addition to the acoustic pressure p, the acoustic velocity v is also accounted for. For the simulation we choose the linear acoustic equations over the acoustic wave equation because of the handling of

boundary conditions which is considerably simpler. The boundary conditions are mostly formed by first-order derivatives so their implementation is facilitated by using a system of first-order differential equations. Regarding the discretization and the solving of the equations, we use a centered finite differences scheme with sixth order of accuracy in space and a forward Euler scheme in time.

The specific details of the neural network used in this study regarding its architecture and the training process can be found in the annex, however, as this is an important peculiarity of the network the structure of the loss function will be explained here. For the training process to start, only some of the data obtained by the numerical simulation is transferred to the PINN. In detail, only the boundary condition, the initial condition and few selected probes inside of the domain (microphone locations) are used as training data. Also, for the mentioned locations – by means of a coarser grid and interpolation – the number of data can be further reduced compared to the numerical simulation. We summarize these data as training points, which thus come from the numerical simulation. Furthermore, the acoustic wave equation in the form of Eq. (1) is defined to finish the setup of the custom loss function. Both parts are added with weights of 100:1 for the ratio of the MSE loss for training data to the MSE loss of the PDE. We have found by trial and error that this weight ratio leads in our study to the best results, hence it was chosen empirically via a random search approach. To present the loss function of our study some definitions are necessary. First, we denote MSE as the loss function. The variable $f(x,t) = \frac{\partial^2 p}{\partial t^2} - \bar{c}\frac{\partial^2 p}{\partial x^2}$ is attributed to the acoustic wave equation and $p(x,t)$ denotes the solution of the PDE. Then, we can define:

$$MSE = 100 \cdot MSE_p + MSE_f, \tag{4}$$

where

$$MSE_p = \frac{1}{N_p}\sum_{i=1}^{N_p}\left|p\left(t_p^i, x_p^i\right) - p^i\right|^2, \tag{5}$$

and

$$MSE_f = \frac{1}{N_f}\sum_{i=1}^{N_f}\left|f\left(t_f^i, x_f^i\right)\right|^2. \tag{6}$$

We see that two different sets of data are used for the two different parts of the loss function. Here, $\left\{t_p^i, x_p^i, p^i\right\}_{i=1}^{N_p}$ define the training points resulting from the numerical simulation. Consequently, initial and boundary conditions and the data coming from the microphone locations inside the domain will be respected. $\left\{t_f^i, x_f^i\right\}_{i=1}^{N_f}$ denote the collocations points, where the residual of the PDE is estimated. These collocation points are specified at random locations inside the spatial and temporal domain via Latin Hypercube Sampling.

3. RESULTS

By means of the presented configuration of the PINN, we want to reconstruct the pressure fields based on only few measurements inside of the domain. For realistic and practical circumstances, broadband forcing at the inlet with white or colored noise would emulate the sound radiated by turbulent sources. However, in a first step towards the above mentioned 'realistic case' we focus on three cases, where the principles of the overall idea become apparent. For the first case, the forcing function at the inlet is defined by an overlay of fundamental and first harmonic frequency whilst $R = 1$ is set at the outlet boundary and $R = 0$ at the inlet. In the second, more complex case the harmonic frequency is replaced with a non-harmonic frequency, hence the ratio of both frequencies does not form an integer. The configuration of the first case builds the basis for the third and last case. Again, fundamental and first harmonic frequency are superimposed, while the reflection coefficient at the outlet is reduced to $R = 0.5$. We choose to present the mentioned three cases as they are capable of summarizing the essentials of our study. A graphical representation of the test cases is shown in Fig. 1.

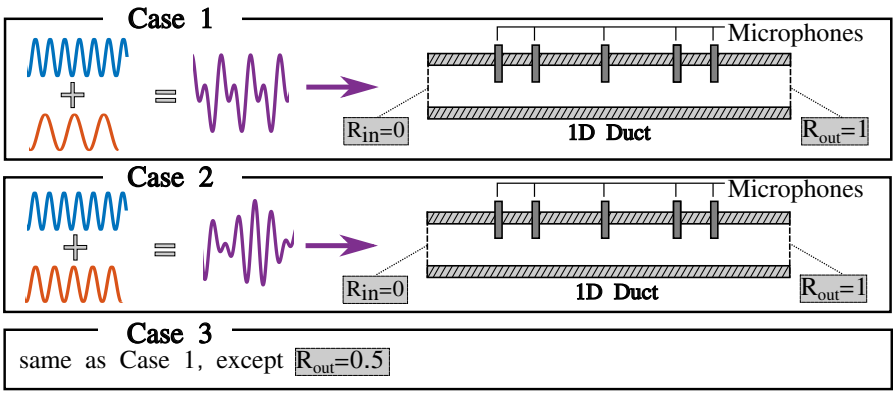

Figure 1: The three test cases that are investigated.

3.1. Case 1: Forcing function with fundamental and first harmonic frequency

We initialize the numerical simulation to provide the necessary training data. The forcing function at the inlet is defined as an overlay of a fundamental frequency (wavelength $\lambda = 2L$) and the first harmonic frequency. The reflection coefficient at the outlet is set to $R = 1$. The pressure domain is fully calculated for a total time of $T = 8$s. We split the obtained data into two subsets of four seconds respectively. Note that the first part (range of 0s – 4s as shown in Fig. 2a) covers the transient build-up of the signal by a travelling pressure wave and a first period of the periodic standing wave pattern. Only few data resulting from the numerical simulation is used as training data for the network. To be specific, the pressure values at the boundaries and at $x = 0.2, x = 0.3, x = 0.5, x = 0.7, x = 0.8$ are implemented in a much coarser temporal grid for boundary conditions, values inside the domain and spatial grid for the initial condition compared to the data from the simulation. Fig. 3 depicts these hard constraints that are transferred to the loss function of the PINN as training data.

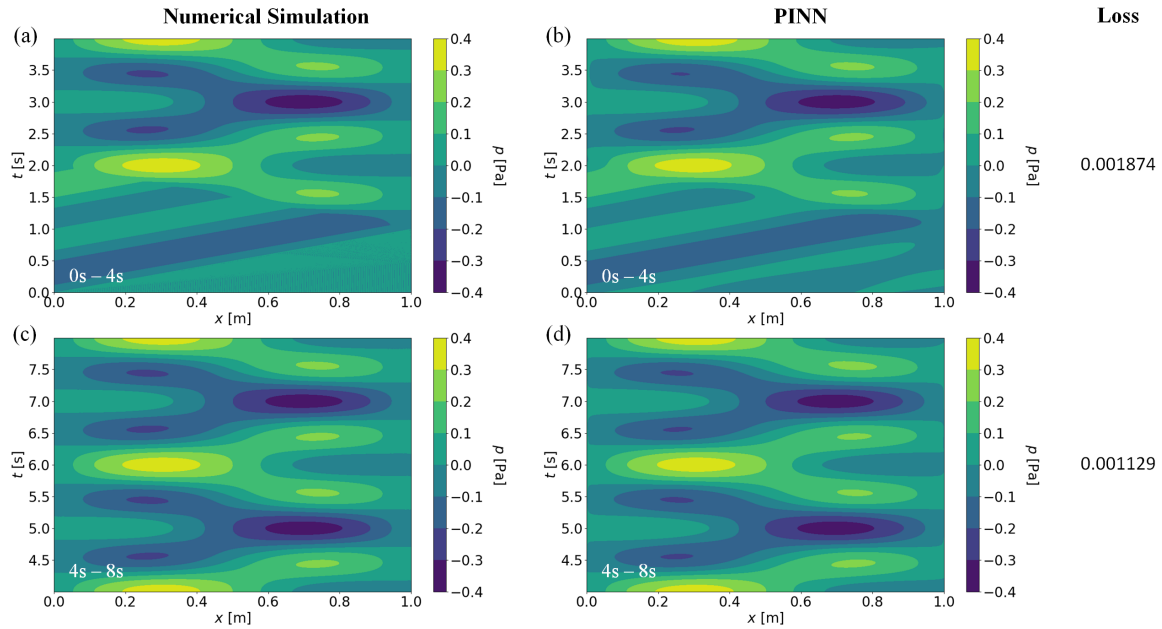

Figure 2: Pressure domain of the Numerical Simulation (a,c) and the PINN (b,d) for case 1 for the time spans of 0s - 4s (a,b) and 4s - 8s (c,d).

The rest of the domain is optimized just by means of minimization of the PDE in the second part of the loss function. The resulting PINN with 60 neurons in each hidden layer predicts the first four seconds of the pressure domain as visible in Fig. 2b. We observe that the main features within the domain are very well captured. The same holds true for the periodicity of the signal after the build-up

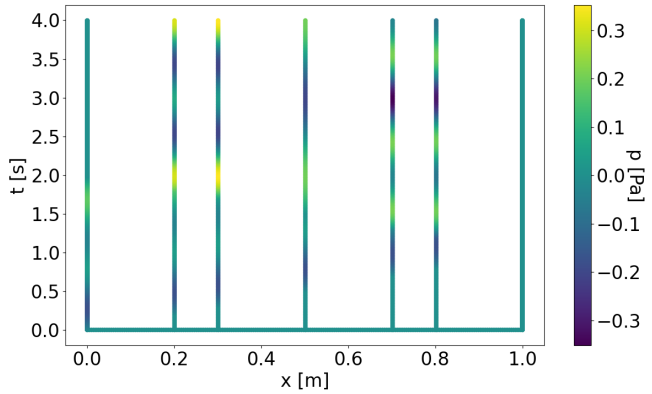

Figure 3: Training points and pressure values respectively transferred to the PINN for the domain of 0s to 4s.

of the signal, hence after two seconds. However, the transient phase can not be predicted with the same high accuracy. Despite a clear representation of the traveling wave peaks during this phase, the regions, where a zero pressure value would be expected according to the numerical simulation, do not show good agreement with the data resulting from the simulation. It also becomes visible that some of the smaller pressure peaks, e.g. at 3.5 seconds, are predicted with a slightly smaller extent compared to the results of the numerical simulation. Eventually, the loss function, which can be interpreted as a non standardized measure of error, amounts to 0.001874. This trained PINN is termed here *c1model1*.

Results improve for the second time window investigated (see Fig. 2c and d). For such a time window of interest, we make use of the values of the already trained PINN related to the first time window (0s to 4s) for initialization (weights and biases of *c1model1*). New data – following the same principle as shown for the first time span – is transferred to the network's loss function as training data. Further training of the model *c1model1* leads to earlier convergence (convergence after 432 epochs instead of 496 epochs for the first time span) of the training algorithm and better accuracy, which can be quantified by the lower loss of 0.001129. For the following sections, this value can be used as a guideline. A detailed look into the predictions by the PINN (Fig. 2d) shows that the pressure levels match very well the simulation data as the pressure oscillations are very well captured in wavelength and amplitude. However, the transitions of the pressure directly at the boundaries present in the field calculated by the simulation are not represented completely accurately by the PINN. The reason for that is the enormous reduction of training data points at the boundaries compared to the simulation, hence the less accurate pressure values are caused by interpolation. This means that an increase in training data would lead to more accurate pressure transitions at the boundaries of the domain. The trained PINN is termed *c1model2*.

3.2. Case 2: Forcing function with fundamental and non-harmonic frequency

The forcing function for the second case is slightly changed: The fundamental frequency f_1 is maintained whereas the harmonic frequency is replaced by a signal with a frequency $f_2 = 1.7 f_1$. Note that case 2 is more challenging than case 1, as the resulting temporal signals are not periodic within the time window analyzed. As the complexity increases, the complexity of the network is adapted as well. The number of hidden layers is raised to 70 and a higher number of training points (finer temporal grid) at the inlet boundary is transferred to the network's loss function. Besides that, the procedure is fully analogue to the first examined case. Again the numerical simulation calculates the pressure domain and the obtained data is split in time spans ranging from 0s to 4s and from 4s to 8s. Fig. 4a depicts the calculated pressure field for the first time span and it becomes visible that no periodicity is present in this case. The network is initialized similarly to case 1. We directly compare

the pressure domain calculated by the numerical simulation to the results predicted by the PINN (Fig. 4b). The problems we encountered for the first case regarding the transient build-up phase can be reduced to very low deviations from the numerical simulation for this case. Hence qualitatively, all the main features of the pressure field can be well captured. The very low loss of 0.0004634 confirms the excellent predictions of the PINN for that case. We can trace the lower loss back to the increase of the number of hidden layer, thus the enhanced complexity of the network. The trained model is termed *c2model1*.

Subsequently, we train *c2model1* with new data ranging from 4s to 8s. When contrasting the results of the numerical simulation (Fig. 4c) with the results of the PINN (Fig. 4d) it becomes visible that the accuracy of the predictions remains very high. All of the pressure peaks are represented correctly compared to the numerical simulation in detail. The loss can even be reduced by a small amount to 0.0004156. We can thus infer that the further training of *c2model1* makes it possible to further increase the accuracy of the prediction capability of our network. On top of that, the further training leads to a significant decrease of the number of epochs, from 619 (0s to 4s) to 316 (4s to 8s). It also becomes apparent that the complexity introduced by non-periodic signals can be easily counteracted by small adjustments to the PINN. The now two-times trained model is termed *c2model2*.

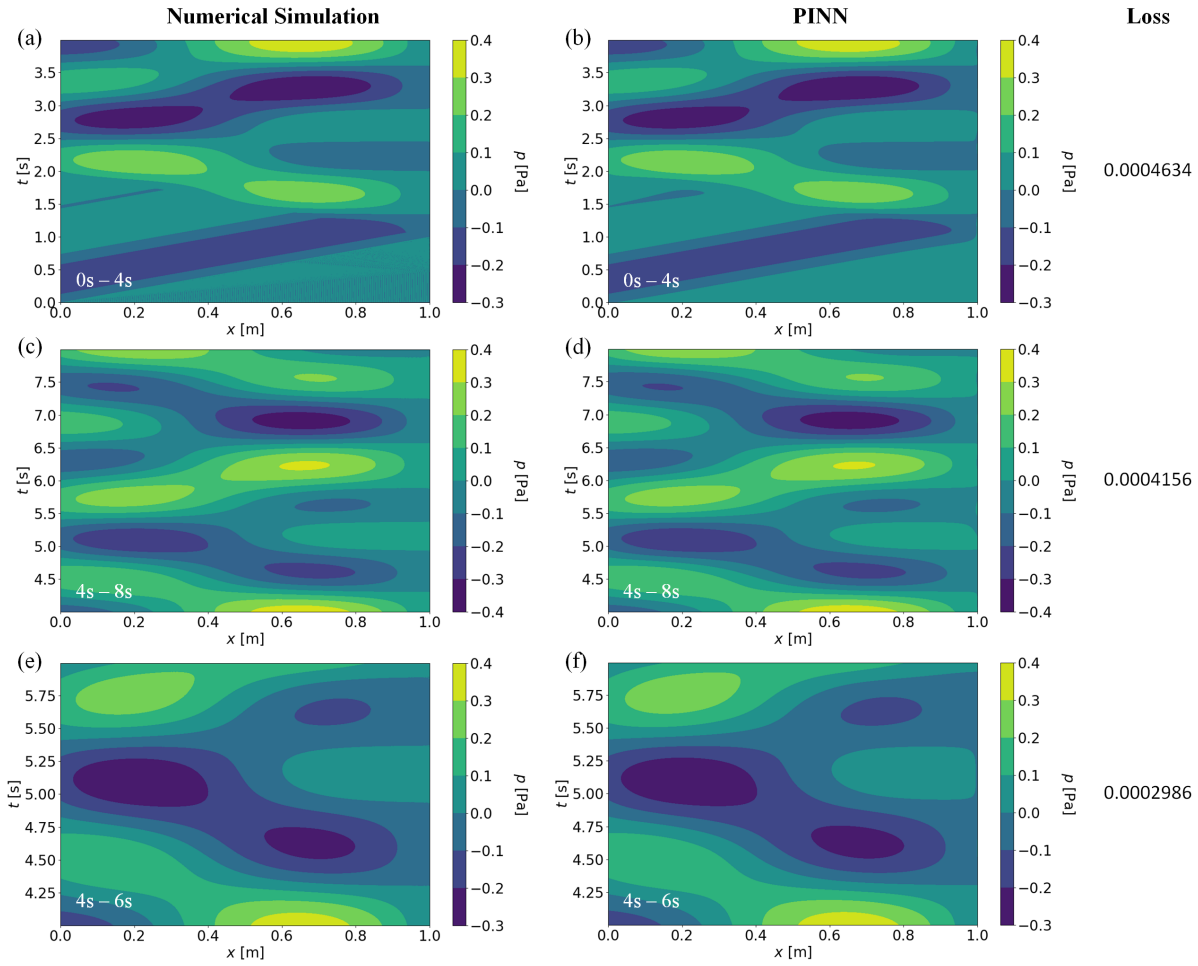

Figure 4: Pressure domain of the Numerical Simulation (a,c,e) and the PINN (b,d,f) for case 2 for the time spans of 0s - 4s (a,b), 4s - 8s (c,d), and 4s - 6s (e,f).

Another approach is made to possibly further hone the results of the network. The time span is decreased to two seconds. Consequently, we use again the trained model *c2model1* for predicting the pressure in the time window from 4s to 6s. Again the contrast between data from the numerical simulation (Fig. 4e) and the prediction of the PINN (Fig. 4f) reveals great congruence with the data resulting from the numerical simulation which can be quantified by a very low loss (0.0002986). This

renewed reduction of the loss can be traced back to the shortened investigated time span.

As a result, we can conclude that with increasing complexity of the case the network may require a more complex structure regarding number of layers or neurons per layer. Alternatively or in addition the time spans of the pressure domains can be decreased to obtain more accurate results. These adaptions, however, show significant success in counteracting the challenges posed by increasing complexity of the signals.

It is interesting to mention that we continue the domain to 12s with the associated numerical simulation (not shown). After that, we let *c2model2* predict the time span of 8s to 12s with the same initialization as before. Once more, the predictions of the PINN agree very well with the results of the numerical simulation. The loss can even be reduced compared to the time span of 4s to 8s to 0.0003003. However, the number of epochs needed for the training process does not further decrease, au contrair the number increases from 316 (4s to 8s) to 536 (8s to 12s). A possible explanation for that enlargement can be that the weights and biases of *c2model2* are already at their local minima so the adaption to the new data becomes more time consuming. It again should be noted that for this case periodicity is not present.

3.3. Case 3: Forcing function with fundamental and harmonic frequency and $R = 0.5$

The procedure of the third case is again analogue to the one of the first case. However, the wall reflection coefficient of the outlet in the numerical simulation is set to $R = 0.5$. The results of the simulation are depicted in Fig. 5a (0s – 4s) and Fig. 5c (4s – 8s). Similarly to case 1, a periodic signal evolves after the transient build-up phase.

Similarly as cases 1 and 2, the data is split into subsets of four seconds and transferred to the loss function of the neural network. The PINN for this case consists of 60 neurons in each hidden layer and the configuration of the network is fully analogue to the first case. It is important to mention at this point that no information regarding the reflection coefficient R is given to the PINN. The first four seconds of the domain can be predicted by the PINN (Fig. 5b) with a loss of 0.001528. Tab. 1 presents a summary of all the mentioned values to provide a quick comparison of the losses for all cases. The overall accuracy for case 3 is comparable to case 1, hence high, except for the same little inaccuracies detected like in case 1: The transient build-up phase is not predicted as accurately as the periodic part of the domain. The trained model is termed *c3model1*.

A new PINNs is initialized with the values of *c3model1* and the training process is now performed with new data from 4s until 8s. By means of this method it is possible for us to achieve an enormous reduction of the loss to 0.0001624. This increase in accuracy can also be made visible by plotting the pressure field predicted by the PINN (Fig. 5d). These results emphasize that further training of an already trained model can have a huge impact on the enhancement of the results of our PINNs.

Based on the results of this last case, the calculation of the wall reflection coefficient at the outlet via postprocessing becomes possible, given the highly resolved acoustic field in space and time. Therefore we reconstruct downstream and upstream traveling waves f and g to define the reflection coefficient R of the outlet:

$$R = \frac{g}{f}, \tag{7}$$

with

$$f = \frac{p}{\bar{\rho}\bar{c}} + v, \tag{8}$$

and

$$g = \frac{p}{\bar{\rho}\bar{c}} - v. \tag{9}$$

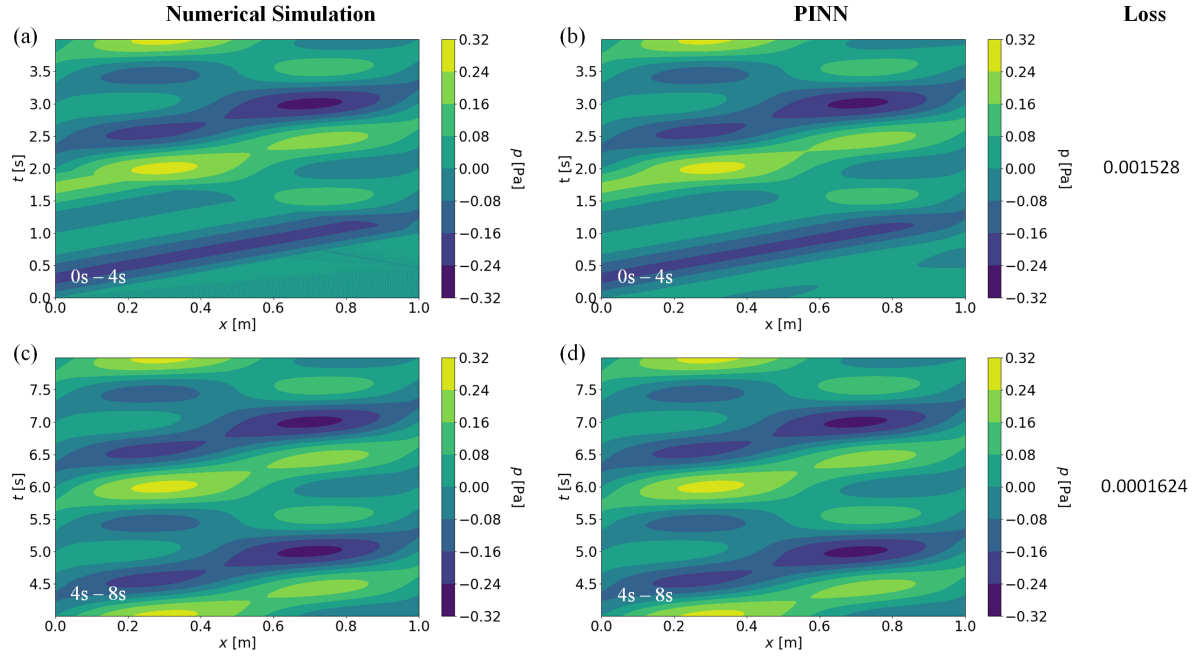

Figure 5: Pressure domain of the Numerical Simulation (a,c) and the PINN (b,d) for case 3 for the time spans of 0s - 4s (a,b) and 4s - 8s (c,d).

Subsequently, we differentiate Eq. (8) and (9) with respect to time to obtain:

$$\frac{\partial f}{\partial t} = \frac{1}{\bar{\rho}\bar{c}}\frac{\partial p}{\partial t} + \frac{\partial v}{\partial t}, \tag{10}$$

$$\frac{\partial g}{\partial t} = \frac{1}{\bar{\rho}\bar{c}}\frac{\partial p}{\partial t} - \frac{\partial v}{\partial t}. \tag{11}$$

In a next step, we substitute the derivative in time of the velocity by means of the Euler equation (3) which leads to:

$$\frac{\partial f}{\partial t} = \frac{1}{\bar{\rho}\bar{c}}\frac{\partial p}{\partial t} - \frac{1}{\bar{\rho}}\frac{\partial p}{\partial x}, \tag{12}$$

$$\frac{\partial g}{\partial t} = \frac{1}{\bar{\rho}\bar{c}}\frac{\partial p}{\partial t} + \frac{1}{\bar{\rho}}\frac{\partial p}{\partial x}. \tag{13}$$

Now, we transform Eq. (7) to make possible the calculation of the reflection coefficient of the outlet just with the acoustic pressure field in space and time:

$$R = \frac{g}{f} = \frac{\frac{\partial g}{\partial t}}{\frac{\partial f}{\partial t}} = \frac{\frac{1}{\bar{\rho}\bar{c}}\frac{\partial p}{\partial t} + \frac{1}{\bar{\rho}}\frac{\partial p}{\partial x}}{\frac{1}{\bar{\rho}\bar{c}}\frac{\partial p}{\partial t} - \frac{1}{\bar{\rho}}\frac{\partial p}{\partial x}}. \tag{14}$$

The necessary differentiation of the given acoustic pressure field can be executed numerically and will not be part of this particular study.

4. CONCLUSIONS

In this work we have shown the potential of PINN to reconstruct acoustic fields, when only acoustic measurements at some specific locations are available. Three cases were discussed, where an input

Table 1: Examined cases and their loss.

Case	Loss
1 (0s - 4s)	0.001874
1 (4s - 8s)	0.001129
2 (0s - 4s)	0.0004634
2 (4s - 8s)	0.0004156
2 (4s - 6s)	0.0002986
2 (8s - 12s)	0.0003003
3 (0s - 4s)	0.001528
3 (4s - 8s)	0.0001624

forcing signal – built by the superposition of two sinusoids – is imposed at the inlet in a one-dimensional duct, and five measurements are performed at five discrete locations along the duct. We show that the PINN proposed is able to reconstruct the entire field in space and time with great accuracy for all three presented cases regardless of whether the signal is periodic or not. Note that the PINN does not have the information of the acoustic reflection at the outlet. Accordingly, the method proposed could be used to infer acoustic conditions once the acoustic field is reconstructed. Such a possibility would be of great interest in acoustic studies, where acoustic boundary conditions are unknown. This work also shows that when the incoming signal is not periodic within the time window of interest, the acoustic reconstruction exercise becomes more challenging: we need a more complex network to make the results of the PINN agree well with the reference data. Future studies will focus on this last case, as input forcing signals that exhibit such non-periodic characteristics – such as broadband forcing – would make the method more appealing and robust for industrial purposes. Further studies will also try to compute the outlet's reflection coefficient of the domain via the presented way in the postprocessing.

ANNEX

This annex gives a detailed overview of the neural networks' architecture used in this study as well as the corresponding training process. The PINNs in our study consist of two inputs (space x and time t), nine hidden layers and one output (pressure p) in a classic architecture of a feedforward multilayer perceptron. Depending on the complexity of the problems, 60 or 70 neuron are used in each hidden layer.

The minimization of the loss function (see Eq. (4)) is accomplished by the Adam optimizer presented in [10]. The hyperparameters for the network are chosen empirically via random search. Because of the time-consuming computational effort the hyperparameters are chosen at first for much simpler cases not presented in this report and adapted to the presented cases by random adjustment. Eventually, this resulted in a learning rate of 0.001, a batch size of 150 and a number of epochs of 10000. However, the convergence was monitored such that, if it is reached earlier, the training process automatically stops.

REFERENCES

[1] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2, 1989.

[2] K. Hornik, M. Stinchcombe, and H. White. Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks. *Neural Networks*, 2, 1990.

[3] A. Krizhevsky, I. Sutskever, and G.E. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25, 2012.

[4] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521, 2015.

[5] B.M. Lake, R. Salakhutdinov, and J.B. Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350, 2015.

[6] B. Alipanahi, A. Delong, M.T. Weirauch, and B.J. Frey. Predicting the sequence specificities of dna- and rna-binding proteins by deep learning. *Nature Biotechnology*, 33, 2015.

[7] M. Raissi, P. Perdikaris, and G.E. Karniadakis. Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations. *ArXiv e-prints*, 2018.

[8] A.G. Baydin, B.A. Pearlmutter, A.A. Radul, and J.M. Siskind. Automatic differentiation in machine learning: a survey. *ArXiv e-prints*, 2015.

[9] L.E. Kinsler, A.R. Frey, A.B. Coppens, and J.V. Sanders. *Fundamentals of Acoustic*. John Wiley & Sons, Inc., 4 edition, 2000.

[10] D.P. Kingma and J.L. Ba. Adam: A method for stochastic optimization. *ArXiv e-prints*, 2014.