Physics-Informed Neural Networks for Acoustic Wave Propagation

Harpreet Sethi*, Doris Pan†, Pavel Dimitrov†, Gunter Roth†, Ken Hester† & Jeffrey Shragge*

*Center for Wave Phenomena and Dept. of Geophysics, Colorado School of Mines, Golden CO 80401
†NVIDIA Corporation, Santa Clara, California
Corresponding Author email: hsethi@mines.edu

ABSTRACT
Modeling the temporal evolution of wavefield solutions through models with heterogeneous material properties is of practical interest for many scientific applications. The acoustic wave equation (AWE) is often used for studying wave propagation in both fluids and solids and is crucial for many applications including seismic imaging and inversion as well as non-destructive testing. Analytical AWE solutions rarely exist for complex heterogeneous media, so methods for generating numerical AWE solutions are very desirable. Traditional numerical solvers require discrete model representations with many restrictions placed on the shape and spacing of grid elements. In this work we use a relatively new class of numerical solvers known as physics informed neural networks (PINNs) that provide a mesh-free alternative for generating AWE solutions using a deep neural-network (DNN) framework. We develop a time-domain AWE formulation in a loss function to train the network parameters. The initial conditions are enforced by enforcing hard constraints on neural network instead of including them as separate terms in the loss function. We also use a Fourier neural network (FNN) to alleviate the spectral bias commonly observed when using full connected neural network (FCNN) in the conventional PINN approach. The numerical tests on both 2D homogeneous and heterogeneous velocity models confirms the accuracy of our approach. We also compare PINN-based solutions with those computed by a pseudo-spectral method. Using FNNs also helps in the convergence of AWE solutions especially for heterogeneous model tests.

Key words: Acoustic, wave propagation, neural networks, physics informed

1 INTRODUCTION

Numerical solutions of wave equations are of scientific importance for wide range of physical problems including in the fields of seismology, acoustics, fluid dynamics, medical science, and electromagnetics. Various numerical approaches including finite-difference, finite-element, finite-volume, and spectral methods have been developed to compute wavefield solutions; however, existing numerical methods require explicitly solving the governing wave equation in an step-wise fashion for each time step and can suffer from numerical errors caused by the discrete meshing. A further challenge is the significant computational cost associated with obtaining multi-scale solutions that require discretizing the computational grid very finely, which results in large memory and/or hardware storage requirements.

Recently, the use of deep neural networks (DNNs) has gained significant attention for modeling physical systems due to their ability to learn complex dynamical systems. Raissi et al. (2019) propose a general framework, known as physics-informed neural networks (PINNs), to solve both forward and inverse problems involving systems of nonlinear partial differential equations (PDEs). The PINN framework involves approximating the PDE solution variables with a DNN. The physical laws governing the PDE, and the initial and boundary conditions (ICs and BCs) are then embedded as terms in the loss function to regularize the network. Standard GPU-accelerated computational packages (e.g., TensorFlow, PyTorch, Keras) then can be used to find the set of solution variables that optimally minimizes the loss function and thereby satisfies the PDE, ICs and BCs.

PINNs offer many practical advantages over traditional numerical methods. First, they represent a mesh-free numerical approach and are less prone to discretization errors. The partial derivative operators involved in the PDEs are computed using automatic differentiation (Baydin et al., 2018), which is exact and free of truncation errors as compared to numerical differentiation methods using approximate stencils derived from Taylor-series or other expansion methods. Once the network is trained, the complete spatial-temporal solution domain can be simulated in a single step, which is important for imaging and inversion applications (e.g., subsurface geological characterization and non-destructive testing) that use the time history of propagating wavefield.

The PINN framework has been used for solving numerous PDEs such as the Navier-Stokes equation for modeling fluid flow (Sun et al., 2020; Jin et al., 2021), cardivascular flow (Kissas et al., 2020), the Black-Scholes equation for option pricing in computational finance (Tanios, 2021), the Schrödinger equation for monitoring pulse evolutions in optical fiber (Jiang et al., 2021), the non-linear equations of motion for structural seismic response modeling (Zhang et al., 2020), the eikonal equation for computing activation times for a conduction velocity field (Sahli Costabal et al., 2020), the seismic travel-time equations (Waheed et al., 2021), the Burgers equation for shallow-water modeling (Raissi et al., 2019), and the Allen-Cahn equation for reaction-diffusion systems (Raissi et al., 2019).

The application of PINNs for generating full-wavefield solution for acoustic wave equation (AWE) has been studied by different authors. Moseley et al. (2020) used PINNs to investigate acoustic wave propagation; however, this approach used wavefield snapshots from a finite-difference method as well as physics-based constraints to train their network. Karimpouli and Tahmasebi (2020) studied PINNs for the forward and inverse seismic AWE applications and demonstrated that PINNs can learn wavefield solution with fewer training data and are capable of estimating the acoustic velocity distribution. Song et al. (2021) formulated the problem in frequency domain to study PINNs for studying seismic wavefields in transversely isotropic media with vertical symmetry axis (VTI) using an acoustic approximation. They computed scattered pressure wavefields based on a perturbation from an isotropic reference solution instead of directly simulating the full pressure wavefield itself.

A key challenge for using PINNs for generating AWE solutions is to train the network without using training data. In practice, assuming a known AWE solution at a certain time step largely defeats the purpose of using PINNs for solving AWE. Because of the multi-scale and oscillatory nature of the AWE solution, it becomes challenging to train the network especially the higher frequencies. A further challenge is handling the injection of source energy localized in both space and time using PINNs. In seismic applications, sources with localized energy over a narrow temporal and spatial window are often used to mimic explosives and marine air gun arrays.

To overcome the aforementioned challenges, we propose to use a time-domain PINN formulation to solve the AWE for pressure wavefields. In contrast with conventional PINN approaches, we train a modified NN in which the initial conditions are satisfied explicitly using hard constraints in the same NN that is used to approximate the solution, which is important in terms of solution convergence. We test our approach using both homogeneous and heterogeneous velocity acoustic models. For the latter we advocate the use of Fourier neural networks (Tancik et al., 2020) instead of the more commonly used fully connected NN to alleviate the spectral-bias towards the low-frequency components during the training process.

In this paper, we start by reviewing the theory of wave propagation in acoustic media. Then, we briefly discuss the 2-D PINN approach including the inputs and outputs of fully-connected neural networks and the loss terms involved in the training procedure. Next, we briefly discuss Fourier-neural networks and the differences between these and fully-connected neural networks. We next discuss the procedure of enforcing initial conditions in a hard manner using a single neural network and changes in network output. Finally, we present numerical examples including using Fourier neural network for increasing velocity model complexity.

2 THEORY

The PINN framework uses physical laws that include the PDE and initial and boundary conditions (ICs and BCs) as terms in the form of loss functions that are used to train the NN parameters. In this section, we first discuss the theoretical aspects of AWE including the ICs and BCs. We then present a standard PINN formulation using fully connected neural networks (FCNNs), briefly introduce the Fourier neural networks (FNNs), and describe the differences between FCNNs and FNNs. Finally, we discuss the enforcement of ICs within the NN (i.e., hard enforcement) instead of using separate loss-function terms (i.e., soft enforcement).

2.1 Acoustic wave equation

Two-dimensional (2-D) acoustic wave propagation in a fluid is described by a coupled system of first-order partial differential equations that represent the conservation of linear momentum,

$$\rho \partial_t v_i + \partial_i p = 0, \qquad i = 1, 2, \tag{1}$$

and the conservation of mass,

$$\partial_t p + \rho c^2 \partial_j v_j = f, \qquad j = 1, 2, \tag{2}$$

where ρ is the fluid density, p is the pressure, v_i is the particle velocity, index i represents the two different coordinate axes, c is the medium velocity, and f is the body force per unit volume. We indicate summation notation over repeated indicies. For a constant density medium, we can further substitute equation 1 into equation 2 to obtain a second-order PDE

$$\partial_t \partial_t p - c^2 \partial_j \partial_j p = f, \quad j = 1, 2, \tag{3}$$

which is the form of the PDE solved in the present work.

2.2 Initial and boundary conditions

For the initial conditions, we assume a quiescent medium at $t = 0$ s, which leads to the first boundary condition

$$p(x, z, t = 0) = 0. \tag{4}$$

This assumption also requires that the temporal derivative of pressure at $t = 0$ s is likewise zero,

$$\partial_t p(x, z, t = 0) = 0. \tag{5}$$

For the boundary conditions, we use the absorbing boundary conditions (ABCs) (Engquist and Majda, 1977) to absorb reflections from the boundaries. One advantage of using PINN approach is that incorporating ABCs comes at no additional computation cost, which is in contrast to other numerical methods where one explicitly applies the ABC operators. Thus, there is no need to enforce ABCs in PINNs via an additional loss term.

2.3 Physics Informed Neural Networks (PINNs)

PINN approaches commonly use a fully connected deep neural networks (DNNs) to approximate the solution variable, which herein is the pressure wavefield p given in equation 3. A DNN is a stack of neurons organized in different layers (i.e., an input layer, multiple hidden layers and an output layer; see Figure 1). The input layer includes the spatial (x, z) and temporal t coordinates represented by the tensor $\mathbf{X}(x, z, t)$. For illustration, for a system with L hidden layers with i^{th} hidden layer having N_i neurons, the connection between $(i-1)^{th}$ to i^{th} hidden layer can be represented as

$$\mathbf{h}_i = \sigma\left(\mathbf{b}_i + \mathbf{W}_i \mathbf{h}_{i-1}\right) \quad 1 \leq i \leq l, \tag{6}$$

where $\mathbf{h}_i$ is the output tensor, $\mathbf{h}_{i-1}$ is the input tensor, $\mathbf{W}_i \in \mathbb{R}^{N_i \times N_{i-1}}$ is the trainable weight matrix, $\mathbf{b}_i \in \mathbb{R}^{N_i}$ is the trainable bias vector, and σ is the non-linear activation function. In this case, we also define $\mathbf{h}_0 = \mathbf{X}(x, z, t)$ while the output layer $\mathbf{h}_{L+1}$ represents the AWE pressure field solution. The goal is to find the set of all tunable parameters, $\mathbf{W}_i$ and $\mathbf{b}_i$, for all hidden layers, such that it minimizes the loss function defined by the PDE and the ICs and BCs.

To solve the AWE, the loss function to be minimized incorporates the governing PDE and the two initial conditions

$$\begin{aligned} L_{PDE} &: \quad \partial_t \partial_t p - c^2 \partial_j \partial_j p - f, \quad i = 1, 2 && (7) \\ L_{IC1} &: \quad \partial_t p(x, z, t = 0), && (8) \\ L_{IC2} &: \quad p(x, z, t = 0). && (9) \end{aligned}$$

Thus, the loss function we aim to minimize is

$$L = ||L_{PDE}||_2 + \alpha_1 ||L_{IC1}||_2 + \alpha_2 ||L_{IC2}||_2, \tag{10}$$

where $|| \cdot ||_2$ indicates the L_2 norm, and α_1 and α_2 are scalar weight terms to control the relative contribution of the IC terms.

Figure 2a presents a schematic representation of the PINN workflow using a DNN to approximate acoustic pressure wavefield. The NN outputs the pressure wavefield and then spatial, and temporal derivatives of the output are computed using an automatic differentiation approach to calculate the three loss function terms. The optimization goal is update the NN parameters such that they globally minimize all three loss-term components.

2.3.1 Fourier Neural Network

DNNs are generally biased toward low-frequency solutions (Rahaman et al., 2019) that create challenges for learning high-frequency wavefield components of AWE solutions. This is especially true for heterogeneous velocity models where NNs must

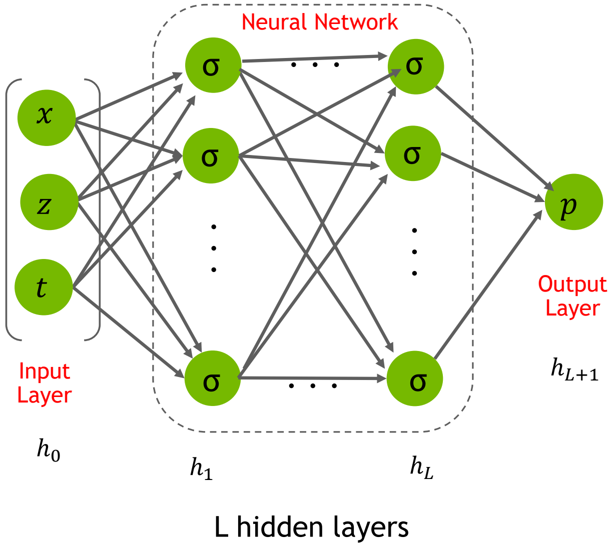
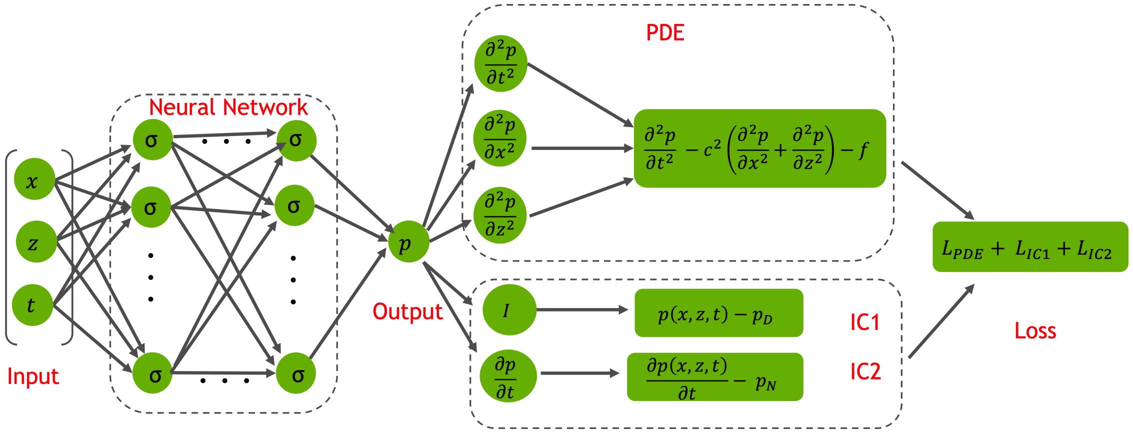
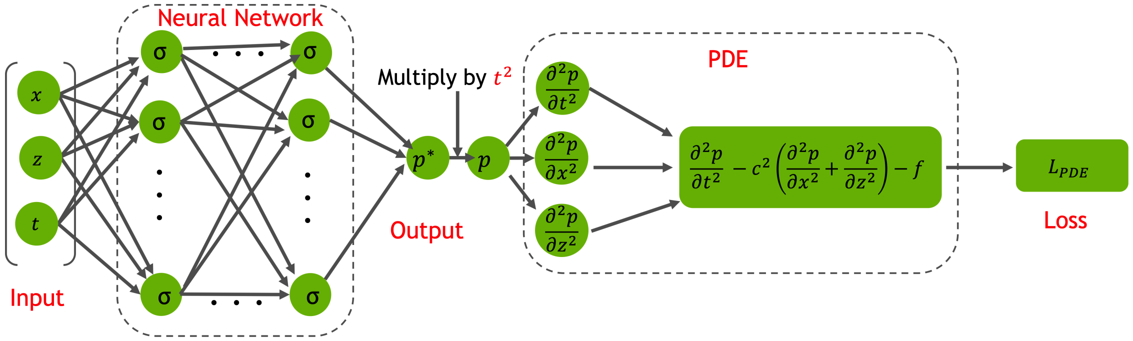


Figure 1. Deep neural network (DNN) with L hidden layers.


Neural Network
x
z
t
σ
p
Input
Output
PDE
$\frac{\partial^2 p}{\partial t^2}$
$\frac{\partial^2 p}{\partial x^2}$
$\frac{\partial^2 p}{\partial z^2}$
$\frac{\partial^2 p}{\partial t^2} - c^2\left(\frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial z^2}\right) - f$
I
$p(x,z,t) - p_D$
$\frac{\partial p}{\partial t}$
$\frac{\partial p(x,z,t)}{\partial t} - p_N$
IC1
IC2
$L_{PDE} + L_{IC1} + L_{IC2}$
Loss

(a)


Neural Network
x
z
t
σ
Input
Multiply by t^2
p^*
p
Output
PDE
$\frac{\partial^2 p}{\partial t^2}$
$\frac{\partial^2 p}{\partial x^2}$
$\frac{\partial^2 p}{\partial z^2}$
$\frac{\partial^2 p}{\partial t^2} - c^2\left(\frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial z^2}\right) - f$
L_{PDE}
Loss

(b)

Figure 2. (a) Schematic representation of PINNs framework. The neural network with L hidden layers on the left-hand side. (b) Modified PINN framework with hard-enforcement of initial conditions. The neural network with L hidden layers on the left-hand side.

learn sharp-gradient wavefield features. In this work, we use a Fourier-neural networks (FNN) for our PINN framework (Tancik et al., 2020). In FNNs, the input coordinates $\mathbf{X}(x,z,t)$ are mapped before being input to the NN as

$$\mathbf{h_0} = [\sin(2\pi\mathbf{f}\odot\mathbf{X}); \cos(2\pi\mathbf{f}\odot\mathbf{X})], \tag{11}$$

where $\mathbf{f}\in\mathbb{R}^{n_f\times N_0}$ is the trainable frequency matrix, n_f is the number of frequency sets, N_0 are number of neurons in first layer, and the $\odot$ symbol is the Hadamard product (i.e., point-wise element multiplication). The remainder of the layers remain same as in a FCN in equation 6. This transformation allows the network to learn the higher frequencies by shifting the spectrum of the neural tangent kernel (NTK), a kernel regression method used to study NN dynamics using gradient-descent-based learning in the limit of infinite-width layers, controlled by parameter **f**. We refer the reader to Tancik et al. (2020) for further details.

2.4 Enforcing Initial Conditions in PINNs

In the conventional PINN framework, the ICs and BCs are enforced as loss terms in the objective function used for network training. However, this will not necessarily satisfy the ICs because there is no guarantee that the loss terms will be identically zero during the minimization process. Therefore, enforcing the initial conditions in a soft manner as a loss term will not yield a unique solution to the specific PDE system in question.

To avoid this issue, we enforce the ICs using hard constraints by multiplying a function $g(t)$ to the NN output such that $g(t=0) = g'(t=0) = 0$, where the prime indicates a temporal derivative and $t = 0$ s is when the source wavelet is excited (i.e., as if $g(t)$ were a Dirac delta function $\delta(t)$) . The NN input is still $\mathbf{X} = \mathbf{X}(x,z,t)$, but the NN output is a transform variable p^* related to p via

$$p = g\,p^*. \tag{12}$$

Using this expression, equation 7 can be rewritten as

$$\partial_t p = g' p^* + g\,\partial_t p^*. \tag{13}$$

When $t = 0$ s, equations 12 and 13 are exactly zero and thus will satisfy the two ICs defined in equations 7 and 8. Accordingly, there is no need to include terms L_{IC1} and L_{IC2} in our loss function, and thus we explicitly set $\alpha_1 = \alpha_2 = 0$ in equation 10. Figure 2b shows the modified PINN workflow diagram with hard enforcement of the initial conditions, which results in an overall simplification of the NN.

For our numerical experiments we compared two functions that satisfy the two ICs in equations 7 and 8:

$$g_1 = a_1 t^2, \tag{14}$$

and

$$g_2 = 1 - \operatorname{sech}(a_2\,t)\,, \tag{15}$$

where a_1 and a_2 are scaling factors. We did not observe significant differences in the results as long as the chosen f_i function satisfies the ICs. We briefly discuss the numerical results from both functions in the Discussion section below.

3 NUMERICAL EXAMPLES

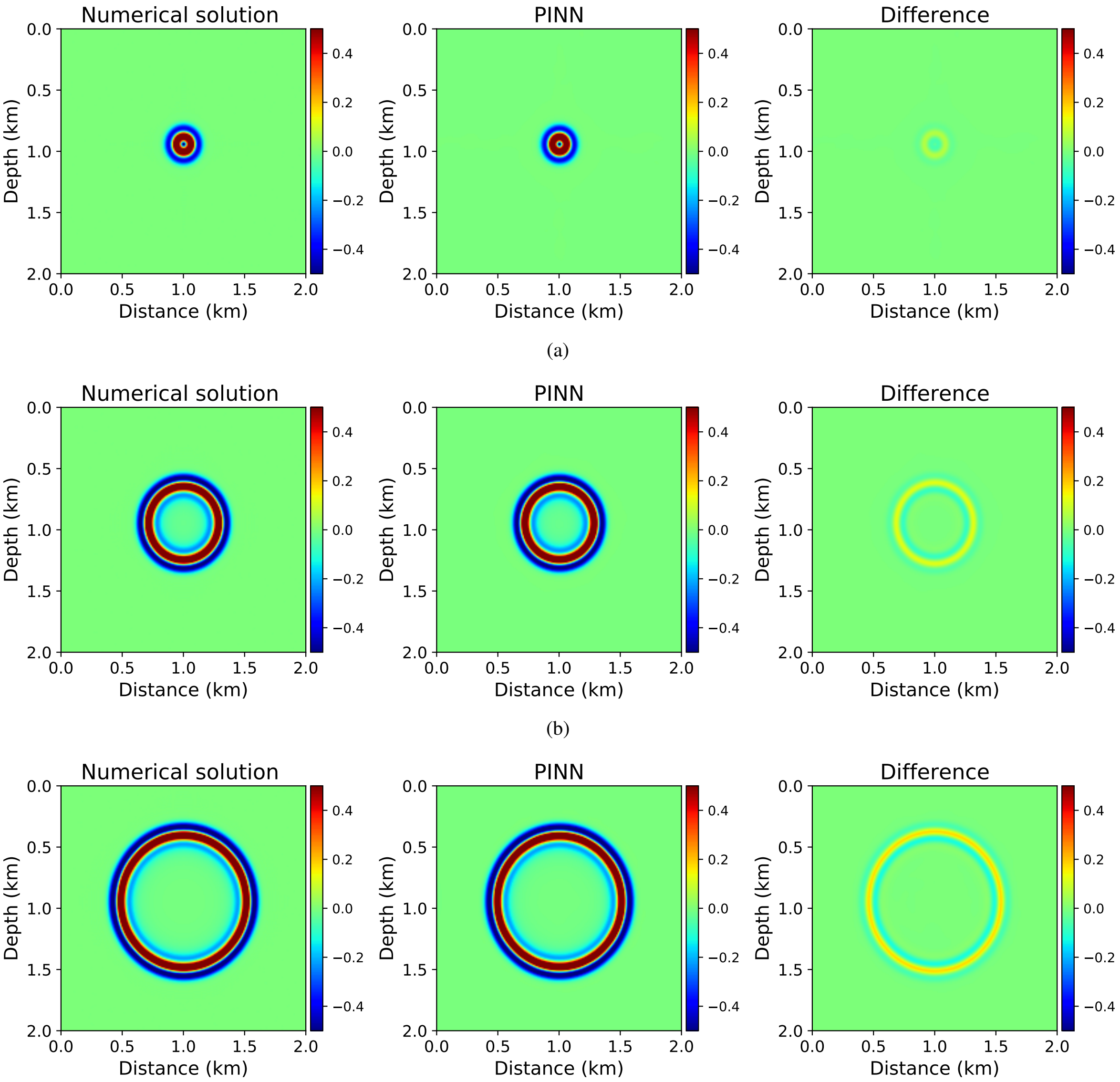
We test our PINN framework on both homogeneous and heterogeneous velocity models and employ FNNs for all numerical tests. We first consider a $N_z \times N_x$ = 200 × 200 homogeneous velocity model (c=1.5 km/s) with $dx = dz = 10$ m spacing. We use a source term in equation 3 given by

$$f(x,z,t) = s(x,z)\,r(t), \tag{16}$$

which is the product of a 20 Hz Ricker wavelet $r(t)$ that is injected as a spatially distributed Gaussian source $s(x,z)$ centered at $[s_x, s_z] = [1.0, 1.0]$ km with weights given by

$$s(x,z) = \mathrm{e}^{-10000\left((x-s_x)^2+(y-s_y)^2\right)}. \tag{17}$$

We run training for 60,000 iterations and compare the computed PINN pressure field to a solution simulated using a highly accurate pseudo-spectral method (Carcione, 2010).



(a)

(b)

(c)

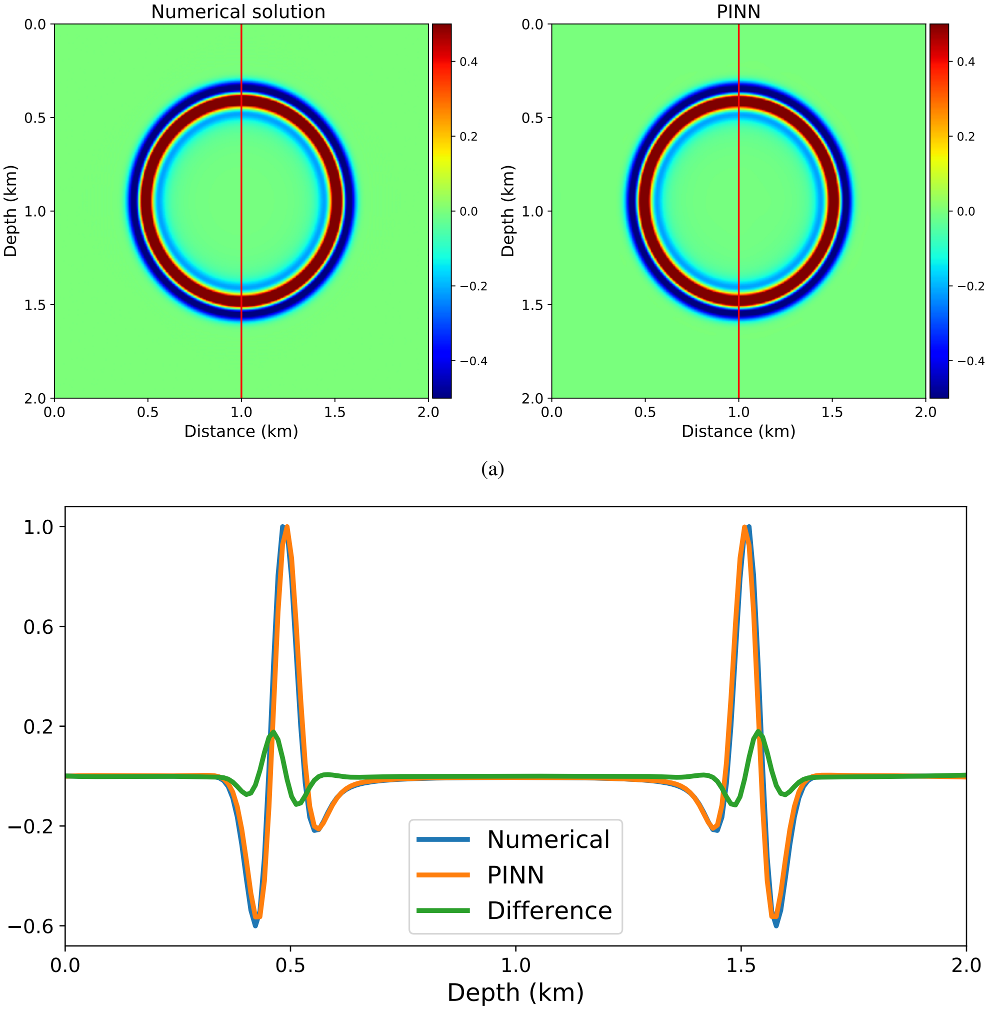
Figure 3. Snapshots for homogeneous velocity model. The left and center columns present the pseudo-spectral and PINN solutions while the right column shows the direct differences. Rows (a)-(c) show the wavefield (differences) at times 0.15 s, 0.30 s, and 0.45 s, respectively.

3.1 Homogeneous Model

Figure 3 shows the wavefield snapshots for the pseudo-spectral method (left column), the PINN approach (center column), as well as their direct wavefield difference (right column). The top, center and bottom rows show the evolution of the wavefield at 0.15 s, 0.30 s and 0.45 s, respectively. The PINN algorithm has accurately learned the phases of the direct arrivals; however, there are minor observable amplitude mismatches. Figure 3c shows the wavefield snapshots at 0.45 s and the amplitude variation with depth extracted at the red line.

Figure 5 shows the energy difference between the FNN and pseudo-spectral solutions for the homogeneous model using g_1 and g_2 functions with different scaling parameters computed as

$$E(t) = \frac{\sum\limits_{x,z} \left(p(x,z,t) - p_0(x,z,t)\right)^2}{\sum\limits_{x,z} p_0^2(x,z,t)}, \tag{18}$$



(a)

(b)

Figure 4. Snapshots for homogeneous velocity model at t=0.45 s.(a) Pseudo-spectral wavefield solution. (b) PINN wavefield solution. (c) Wavefield difference between solutions in (a) and (b) computed along the vertical red line in (a) and (b).

where $p(x, z, t)$ and $p_0(x, z, t)$ are NN and pseudo-spectral solution, respectively. We observe the energy differences for $g_1 = a_1 t^2$ and $g_2 = 1 - \text{sech}(a_2 t)$ to be minimum and the solutions from other scaling parameters for both functions were comparable.

3.2 Layered Model

Next, we consider a 200 × 200 heterogeneous velocity model with three layers shown in Figure 6. We again inject a 20 Hz Ricker wavelet as a spatially distributed source centered at $[s_x, s_z] = [1.0, 1.0]$ km and use the same FNN for training, but now run a total of 100,000 training iterations. The left and center columns of Figure 7 respectively show wavefield snapshots for the pseudo-spectral and PINN solvers at different time steps, while the right column presents the direct wavefield differences. From top to bottom, the three rows show the wavefields at 0.15 s, 0.30 s and 0.45 s. We note that the PINN solver has learned the internal reflections generated by the layers. Figure 7c shows the snapshot at t=0.45 s and the amplitudes extracted at red vertical line.

3.3 Marmousi Model

Finally, we test our algorithm on modified Marmousi model (see Figure 9). A 20 Hz Ricker wavelet is injected as a spatially distributed source centered at $[s_x, s_z]$=[1.0,0.75] km. Figure 10 shows the snapshots at different time steps and the PINN and

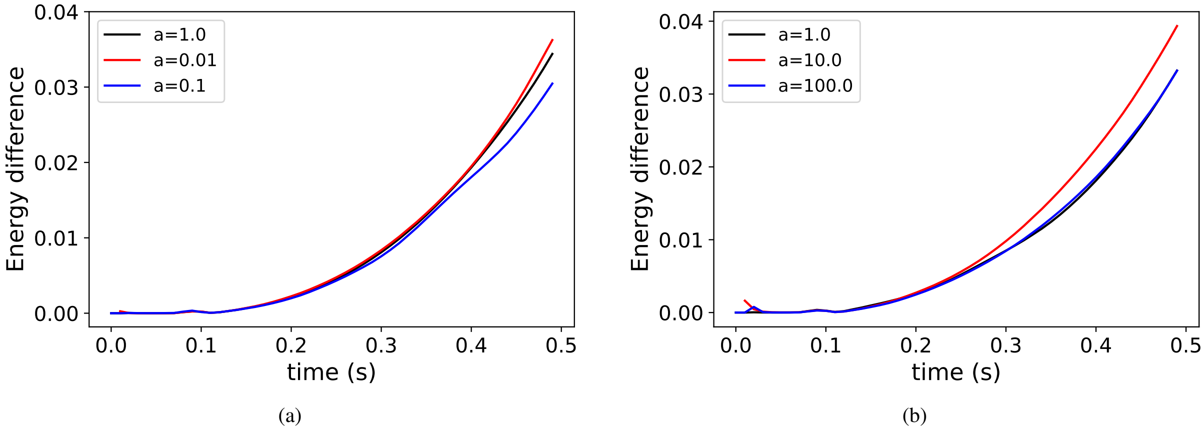


Figure 5. Energy plots for functions with different scaling factors: (a) $g_1 = a_1 t$ and (b) $g_2 = 1 - \mathrm{sech}\, \mathrm{a}_2\mathrm{t}$.

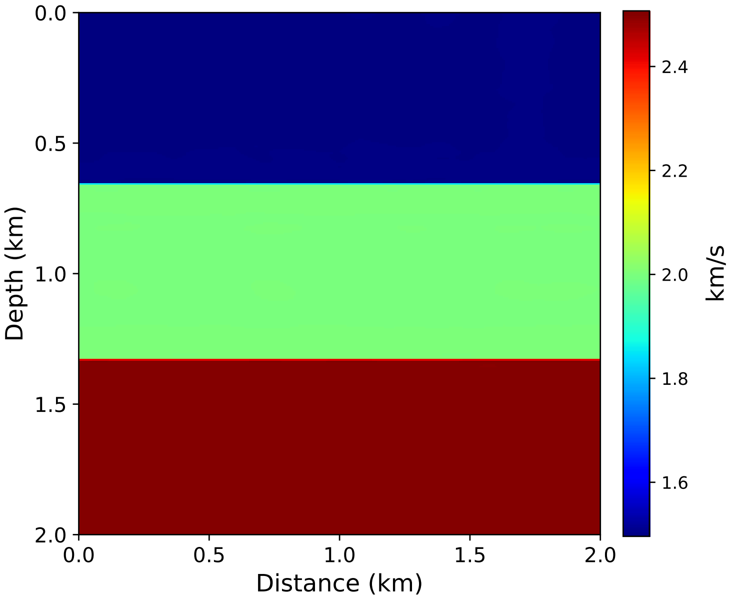
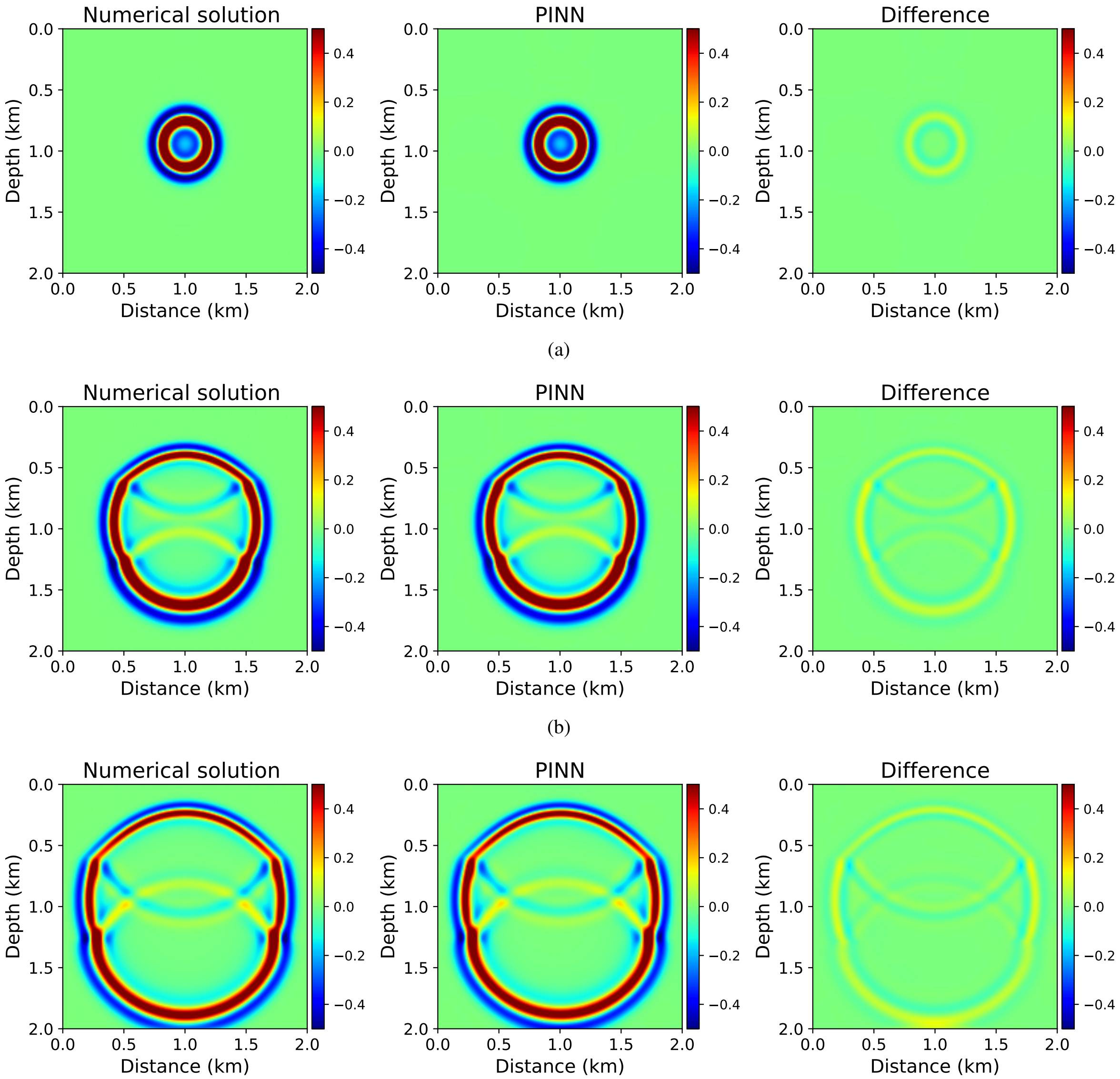


Figure 6. Layered velocity model.

the pseudo-spectral numerical solution difference. We observe a good agreement between the phases of the arrivals originated from sharp gradients in the velocity model; however, there are again fairly minor pressure wavefield amplitude differences (see Figure 10c).

4 DISCUSSION

The time-domain PINN algorithm can learn AWE wavefield solutions for both homogeneous and more complex heterogeneous models. However, there are still challenges in learning amplitude in the absence of training data. FNN training requires choosing a set of frequencies. For homogeneous and layered velocity models, we opted for 20 frequencies that were uniformly distributed between 0-4 Hz in equation 11. However, for Marmousi model we increased this to 50 frequencies between 0-10 Hz. We also experimented with higher frequencies; however, this approach increased the overall training time while minimally improving amplitudes accuracy. Thus, we kept the number of layers to eight, the number of neurons in each layer at 256, and employed the swish activation function (Ramachandran et al., 2017).



(a)

(b)

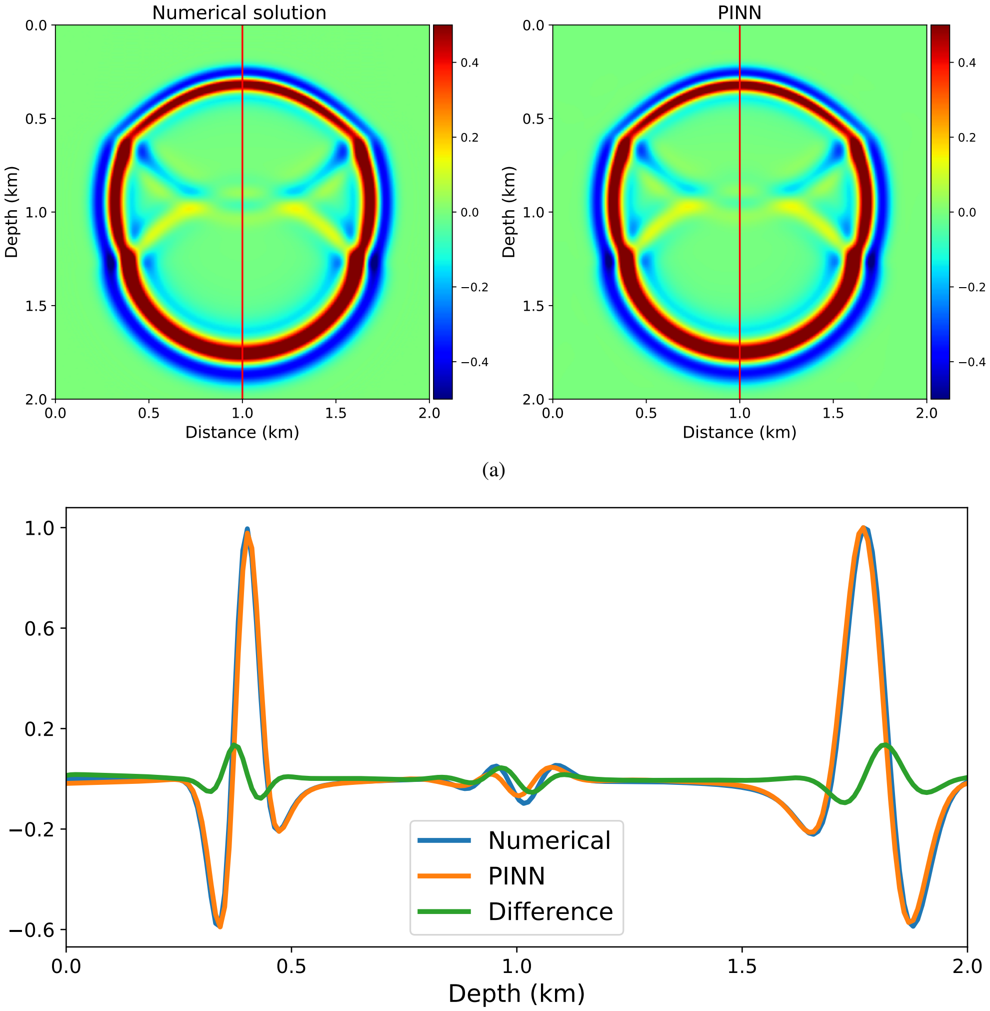
(c)

Figure 7. Snapshots for layered velocity model. The left and center columns present the pseudo-spectral and PINN solutions while the right column shows the direct differences. Rows (a)-(c) show the wavefield (differences) at times 0.20 s, 0.50 s, and 0.50 s, respectively.

We also observe an accuracy improvement with increasing the number of layers from eight to ten and increasing the number of neurons in each layer from 256 to 1024; however, this drastically increased the total training time. From the numerical experiments, we can see PINN approach can match the phases of different arrivals accurately, but further work is required to learn wavefields amplitudes.

5 CONCLUSIONS

We developed a time-domain PINN approach to learn the wavefield solutions for acoustic wave equation without using training data. The enforcement of initial conditions in hard manner by multiplying a temporal function to the network output, which proved to be crucial for training convergence. The comparison with pseudo-spectral method demonstrates that PINN approach is able to produce accurate phases of different arrivals; however, wavefield amplitudes had minor inaccuracies even for heterogeneous



(a)

(b)

Figure 8. Snapshots for layered velocity model at t=0.45 s. (a) Pseudo-spectral and (b) PINN wavefield solutions. (c) Wavefield differences computed at the vertical red line indicated in (a) and (b).

velocity models. Thus, further refinement of this approach is deemed necessary for applications where a high-degree of amplitude fidelity is required.

6 ACKNOWLEDGMENTS

This work was supported by the sponsors of the Consortium Project on Seismic Inverse Methods for Complex Structures at Center for Wave Phenomena at Colorado School of Mines. We would also like to thank NVIDIA for providing the computational resources for running the numerical experiments.

REFERENCES

Baydin, A. G., B. A. Pearlmutter, A. A. Radul, and J. M. Siskind, 2018, Automatic differentiation in machine learning: a survey: Journal of Machine Learning Research, **18 (153)**, no. 153, 1–43.

Carcione, J. M., 2010, A generalization of the fourier pseudospectral method: Geophysics, **75**, no.6, A53–A56.

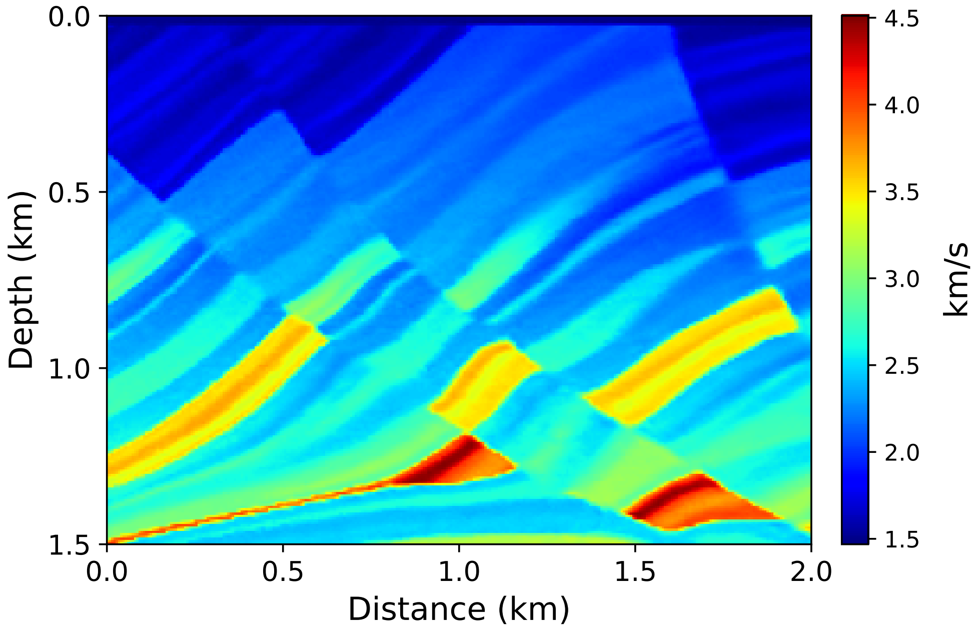


Figure 9. Modified Marmousi velocity model.

Engquist, B., and A. Majda, 1977, Absorbing boundary conditions for numerical simulation of waves: Proceedings of the National Academy of Sciences, **74**, 1765–1766.

Jiang, X., D. Wang, Q. Fan, M. Zhang, C. Lu, and A. P. T. Lau, 2021, Solving the Nonlinear Schrödinger Equation in Optical Fibers Using Physics-informed Neural Network: 2021 Optical Fiber Communications Conference and Exhibition (OFC), IEEE, 1–3.

Jin, X., S. Cai, H. Li, and G. E. Karniadakis, 2021, NSFnets (Navier-Stokes flow nets): Physics-informed neural networks for the incompressible Navier-Stokes equations: Journal of Computational Physics, **426**, 109951.

Karimpouli, S., and P. Tahmasebi, 2020, Physics informed machine learning: Seismic wave equation: Geoscience Frontiers, **11**, 1993–2001.

Kissas, G., Y. Yang, E. Hwuang, W. R. Witschey, J. A. Detre, and P. Perdikaris, 2020, Machine learning in cardiovascular flows modeling: Predicting arterial blood pressure from non-invasive 4D flow MRI data using physics-informed neural networks: Computer Methods in Applied Mechanics and Engineering, **358**, 112623.

Moseley, B., A. Markham, and T. Nissen-Meyer, 2020, Solving the wave equation with physics-informed deep learning.

Rahaman, N., A. Baratin, D. Arpit, F. Draxler, M. Lin, F. Hamprecht, Y. Bengio, and A. Courville, 2019, On the spectral bias of neural networks: International Conference on Machine Learning, PMLR, 5301–5310.

Raissi, M., P. Perdikaris, and G. E. Karniadakis, 2019, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations: Journal of Computational Physics, **378**, 686–707.

Ramachandran, P., B. Zoph, and Q. V. Le, 2017, Searching for activation functions: arXiv preprint arXiv:1710.05941.

Sahli Costabal, F., Y. Yang, P. Perdikaris, D. E. Hurtado, and E. Kuhl, 2020, Physics-informed neural networks for cardiac activation mapping: Frontiers in Physics, **8**, 42.

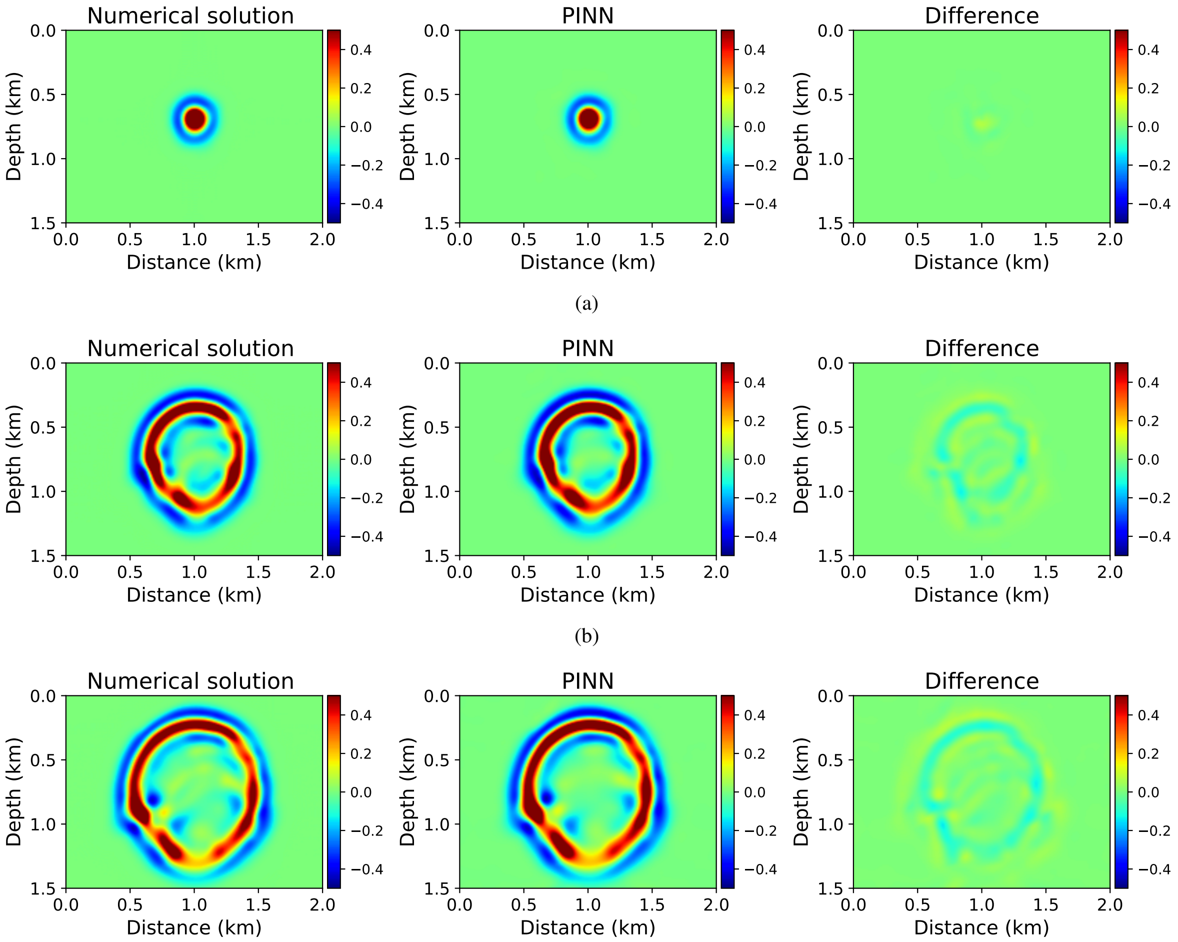
Song, C., T. Alkhalifah, and U. B. Waheed, 2021, Solving the frequency-domain acoustic VTI wave equation using physics-informed neural networks: Geophysical Journal International, **225** **(2)**, no. 2, 846–859.

Sun, L., H. Gao, S. Pan, and J.-X. Wang, 2020, Surrogate modeling for fluid flows based on physics-constrained deep learning without simulation data: Computer Methods in Applied Mechanics and Engineering, **361**, 112732.

Tancik, M., P. P. Srinivasan, B. Mildenhall, S. Fridovich-Keil, N. Raghavan, U. Singhal, R. Ramamoorthi, J. T. Barron, and R. Ng, 2020, Fourier features let networks learn high frequency functions in low dimensional domains: arXiv preprint arXiv:2006.10739.

Tanios, R., 2021, Physics informed neural networks in computational finance: High dimensional forward & inverse option pricing: Master's thesis.

Waheed, U., E. Haghighat, T. Alkhalifah, C. Song, and Q. Hao, 2021, PINNeik: Eikonal solution using physics-informed neural



(a)

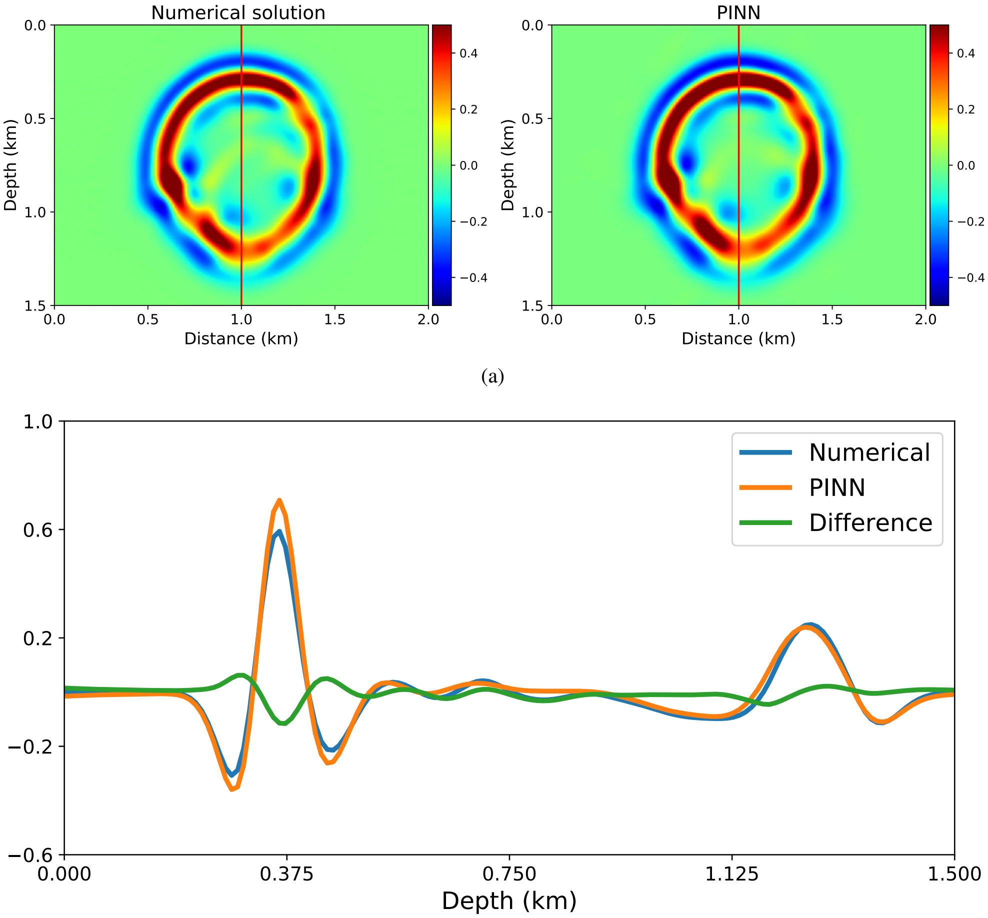
(b)

(c)

Figure 10. Snapshots for modified Marmousi velocity model. The left and center columns present the pseudo-spectral and PINN solutions while the right column shows the direct differences. Rows (a)-(c) show the wavefield (differences) at times 0.30 s, 0.50 s, and 0.60 s, respectively.

networks: Computers & Geosciences, **155**, 104833.

Zhang, R., Y. Liu, and H. Sun, 2020, Physics-guided convolutional neural network (phycnn) for data-driven seismic response modeling: Engineering Structures, **215**, 110704.



(a)

(b)

Figure 11. Snapshots for modified Marmousi model at t=0.55 s. (a) Pseudo-spectral and (b) PINN wavefield solutions. (c) Wavefield differences computed at the vertical red line indicated in (a) and (b).