Computer Physics Communications 295 (2024) 109010

Contents lists available at ScienceDirect

Computer Physics Communications

journal homepage: www.elsevier.com/locate/cpc

Computational Physics

SeismicNet: Physics-informed neural networks for seismic wave modeling in semi-infinite domain ☆

Pu Ren [a,1], Chengping Rao [b,1], Su Chen [c], Jian-Xun Wang [d], Hao Sun [e,f,*], Yang Liu [g,**]

[a] Department of Civil and Environmental Engineering, Northeastern University, Boston, MA 02115, USA
[b] Department of Mechanical and Industrial Engineering, Northeastern University, Boston, MA 02115, USA
[c] Key Laboratory of Urban Security and Disaster Engineering of the Ministry of Education, Beijing University of Technology, Beijing, 100124, China
[d] Department of Aerospace and Mechanical Engineering, University of Notre Dame, Notre Dame, IN 46556, USA
[e] Gaoling School of Artificial Intelligence, Renmin University of China, Beijing, 100872, China
[f] Beijing Key Laboratory of Big Data Management and Analysis Methods, Beijing, 100872, China
[g] School of Engineering Science, University of Chinese Academy of Science, Beijing, 101408, China

ARTICLE INFO

Dataset link: https://github.com/paulpuren/seismic_modeling

Keywords:
Physics-informed neural networks
Domain decomposition
Seismic wave modeling
Absorbing boundary conditions
Forward simulation
Semi-infinite domain

ABSTRACT

Recently, there has been an increasing interest in leveraging physics-informed neural networks (PINNs) for modeling dynamical systems. However, limited studies have been conducted along this horizon on seismic wave modeling tasks. A critical challenge is that these geophysical problems are typically defined in large domains (i.e., semi-infinite), which leads to high computational costs. We present a new PINN model for seismic wave modeling in semi-infinite domain without the need for labeled data. Specifically, the absorbing boundary condition is introduced into the network as a soft regularizer for handling truncated boundaries. To scale up, we consider a sequential training strategy via temporal domain decomposition to improve the scalability of the network and solution accuracy. Moreover, we design a novel surrogate modeling strategy to account for parametric loading, which estimates the wave propagation in semi-infinite domain given the seismic loading at different locations. Various numerical experiments are implemented to evaluate the performance of the proposed PINN model in the context of forward modeling of seismic wave propagation. In particular, we use diverse material distributions to test the versatility of this approach. The results demonstrate excellent solution accuracy under distinctive scenarios.

1. Introduction

Understanding seismic wave propagation is crucial for the communities of seismology and earthquake engineering. Typically, we record seismic signals with seismometers and derive various information about the composition of the Earth, e.g., the discovery of Mohorovicic discontinuity which separates the crust and the mantle of the Earth. For decades, we have observed significant advancements in the theoretical investigation of seismic wave physics. Furthermore, the theoretical breakthrough also brings about the rapid development of numerical modeling of seismic wave propagation for applied seismology and engineering practices, such as finite difference [1,2], finite element [3] and spectral element [4,5] methods. In general, these traditional numerical methods employ finite sets of basis functions (e.g., polynomials) and the corresponding parameters to approximate the derivatives of differential equations in computing their solutions. They have achieved high solution accuracy for a range of ordinary or partial differential equations (ODEs/PDEs). However, these methods may be less suitable for surrogate modeling and inverse analysis due to the need for repeated forward model runs, resulting in expensive computations.

Thanks to the great progress in artificial intelligence (AI), in particular machine learning (ML), many scientists started to explore the poten-

☆ The review of this paper was arranged by Prof. Weigel Martin.

* Corresponding author at: Gaoling School of Artificial Intelligence, Renmin University of China, Beijing, 100872, China.
** Corresponding author at: School of Engineering Science, University of Chinese Academy of Sciences, Beijing, 101408, China.
E-mail addresses: ren.pu@northeastern.edu (P. Ren), haosun@ruc.edu.cn (H. Sun), liuyang22@ucas.ac.cn (Y. Liu).

[1] Equal contribution.

https://doi.org/10.1016/j.cpc.2023.109010
Received 1 November 2022; Received in revised form 4 July 2023; Accepted 6 November 2023
Available online 13 November 2023

tial of using ML to address seismic wave propagation problems [6–13]. For example, Fourier Neural Operator (FNO) [14] has been investigated for forward and inverse analysis of 2D acoustic wave equation [15]. Nevertheless, these research efforts rely on large amounts of high-fidelity labeled data, which are usually difficult to compute or collect. It has been a pressing task to develop novel ML methods for dealing with imperfect measurement data (e.g., sparse and potentially noisy).

The recent development of PINNs [16,17] has shed new light on scientific computing with limited labeled data. The general principle of PINNs is to integrate deep neural networks (DNNs) and physical laws to learn the underlying consistent dynamics from small or zero labeled data. The advent of PINNs provides a new perspective for training neural networks, where the introduction of physical laws into the network strengthens the constraint for optimization for small data regimes. For instance, the latest research work [18–23] attempt to solve PDE systems when the full physics knowledge is accessible, i.e., governing equations, initial and boundary conditions (I/BCs). In addition, we observe the explosive research of applying this paradigm to miscellaneous scientific domains, including fluid flows [24–30], solid mechanics [31–34], multi-scale and multi-physics modeling [35–37], heat transfer [38], subsurface transport [39,40] equation discovery [41–44] and data augmentation [45–48]. Although PINNs have shown great success in various scientific problems, it is also noteworthy that they are not competent in comparison with traditional numerical methods for forward simulations in terms of solution accuracy. Their huge potential lies in tackling surrogate modeling and inverse analysis with a tradeoff between accuracy and computational efficiency [49], especially when only sparse data are recorded.

In this paper, we present a PINN approach for forward modeling of seismic wave propagation, as a basis for possible inverse implementations (e.g., full waveform inversion, a.k.a., FWI). Recently, a few studies [50–53] show the efficacy of PINN succeeding in solving FWI problems in the context of acoustic wave equations. Note that the acoustic wave equation is generally used to model wave propagation in the deep subsurface (e.g., at the km scale), which is sometimes inapplicable to near-surface waveform simulation and inversion [54]. Elastic wave equations provide a solution to near sub-surface wave propagation modeling. To this end, we establish a PINN model for solving elastic wave equations (i.e., a more complex PDE system) in semi-infinite domain with inhomogeneous material profiles. We also design a PINN-based surrogate model to directly infer the full waveforms given diverse loading locations, which, to the best of our knowledge, is the first attempt in the context of seismic wave propagation modeling. The main contributions of this paper are three-fold. First, we introduce the absorbing boundary condition (ABC) into PINN to deal with truncated boundaries in semi-infinite domain, which works as a soft regularizer in network optimization, to enable reliable wave propagation. Second, a temporal decomposition training strategy is presented to improve the scalability and solution accuracy of our PINN architecture for large-scale modeling. Lastly, we validate the effectiveness of our approach for elastic wave propagation modeling compared with the reference numerical results under various scenarios (e.g., different material distributions and loading conditions).

The rest of the paper is organized as follows, apart from this Introduction section. Section 2 formulates the scientific problem of seismic wave propagation. Section 3 presents the details of the specific PINN model, an introduction to ABC, the sequential training scheme via domain decomposition, and the strategy for parametric surrogate modeling. In Section 4, we show the results of extensive numerical experiments to evaluate the performance of our proposed method. Section 5 discusses the advantages and limitations of the current method as well as directions for future research. Section 6 concludes the entire paper.

2. Problem formulation

In this paper, we aim to investigate the potential of PINN for seismic wave propagation modeling in 2D elastic media. The specific governing equations of interest are given by [55]

$$\begin{aligned}\rho\frac{\partial^2 u_0}{\partial t^2} &= \frac{\partial}{\partial x}\left((2\mu+\lambda)\frac{\partial u_0}{\partial x}+\lambda\frac{\partial u_1}{\partial y}\right)+\frac{\partial}{\partial y}\left(\mu(\frac{\partial u_1}{\partial x}+\frac{\partial u_0}{\partial y})\right),\\ \rho\frac{\partial^2 u_1}{\partial t^2} &= \frac{\partial}{\partial x}\left(\mu(\frac{\partial u_1}{\partial x}+\frac{\partial u_0}{\partial y})\right)+\frac{\partial}{\partial y}\left(\lambda\frac{\partial u_0}{\partial x}+(2\mu+\lambda)\frac{\partial u_1}{\partial y}\right),\end{aligned} \tag{1}$$

where $\{x, y\}$ and t denote the spatial and temporal coordinates, respectively; $\{u_0, u_1\}$ are the two displacement variables (horizontal and vertical, respectively). ρ is the density and $\{\lambda, \mu\}$ represent Lamé constants of the medium. Regarding the setting of boundary conditions (BCs), we provide a detailed illustration in Section 3.2. In addition, Eq. (1) can also be rewritten in a matrix formulation, which is summarized as:

$$\mathbf{u}_{tt} = \mathbf{D}_1\mathbf{u}_{xx} + \mathbf{H}\mathbf{u}_{xy} + \mathbf{D}_2\mathbf{u}_{yy}, \tag{2}$$

where $\mathbf{u} = [u_0, u_1]^\top$ is the displacement vector and the subscripts denote the partial derivatives, e.g., $\mathbf{u}_{xx} = \partial^2\mathbf{u}/\partial x^2$. Moreover, $\{\mathbf{D}_1, \mathbf{D}_2, \mathbf{H}\}$ refer to the coefficient matrices associated with material properties, which are expressed as

$$\mathbf{D}_1 = \begin{bmatrix}\alpha^2 & 0\\ 0 & \beta^2\end{bmatrix},\ \mathbf{D}_2 = \begin{bmatrix}\beta^2 & 0\\ 0 & \alpha^2\end{bmatrix},\ \mathbf{H} = \begin{bmatrix}0 & \alpha^2-\beta^2\\ \alpha^2-\beta^2 & 0\end{bmatrix}. \tag{3}$$

Herein, α and β represent the longitudinal and transverse wave speeds, respectively, which can be calculated through

$$\alpha = \sqrt{\frac{\lambda+2\mu}{\rho}},\ \beta = \sqrt{\frac{\mu}{\rho}}. \tag{4}$$

We observe that Eq. (1) is characterized by many second-order derivatives, which could result in a large computational graph for PINNs. The previous work [26] has demonstrated that reducing the order of automatic differentiation (i.e., the order of derivatives in the governing equations) can effectively improve the learnability and the solution accuracy of PINNs. Hence, we re-formulate the governing equations in Eq. (1) to an equivalent state-space form, which is given by

$$\begin{aligned}\rho\mathbf{v}_t &= \mathbf{Q}\boldsymbol{\sigma},\\ \boldsymbol{\sigma}_t &= \mathbf{C}\boldsymbol{\epsilon}_t,\\ \mathbf{u}_t &= \mathbf{v},\end{aligned} \tag{5}$$

where $\mathbf{u} = [u_0, u_1]^\top$ and $\mathbf{v} = [v_0, v_1]^\top$ refer to the displacement and velocity vectors, respectively. $\{\boldsymbol{\sigma}, \boldsymbol{\epsilon}_t\}$ denote the corresponding stress and strain rate vectors, which are expressed as

$$\begin{aligned}\boldsymbol{\sigma} &= \left[\sigma_{xx},\ \sigma_{yy},\ \sigma_{xy}\right]^\top,\\ \boldsymbol{\epsilon}_t &= \left[\frac{\partial v_0}{\partial x},\ \frac{\partial v_1}{\partial y},\ \frac{\partial v_0}{\partial y}+\frac{\partial v_1}{\partial x}\right]^\top.\end{aligned} \tag{6}$$

Note that $\{\sigma_{xx}, \sigma_{yy}\}$ represent the normal stresses along x-dimension and y-dimension, respectively; σ_{xy} is the shear stress. In addition, $\mathbf{Q}$ is an operator matrix and $\mathbf{C}$ represents the constitutive tensor, which are formulated as

$$\begin{aligned}\mathbf{Q} &= \begin{bmatrix}\frac{\partial}{\partial x} & 0 & \frac{\partial}{\partial y}\\ 0 & \frac{\partial}{\partial y} & \frac{\partial}{\partial x}\end{bmatrix},\\ \mathbf{C} &= \begin{bmatrix}\lambda+2\mu & \lambda & 0\\ \lambda & \lambda+2\mu & 0\\ 0 & 0 & \mu\end{bmatrix}.\end{aligned} \tag{7}$$

In essence, Eq. (5) represent the governing equations of elastodynamics under the assumption of plane strain [56].

Our objective is to leverage PINN for forward simulation of elastic wave propagation, including solving the specific PDEs and paramet-

ric surrogate modeling. To be more precise, we aim to solve the seismic wave propagation equations in elastic media (as shown in Eq. (1) or (2)) using PINN in a given semin-infinite domain. We only rely on accessible physical principles (i.e., governing equations and I/BCs) to optimize the entire network without any labeled data. In the part of parametric surrogate modeling, we build a PINN architecture to extrapolate the seismic responses with respect to different loading positions. Herein, a new variable $\mathbf{x}_c$ is introduced into PINNs apart from the typical coordination information (e.g., $\{t, x, y\}$). Here, $\mathbf{x}_c$ is used to mark the specific spatial loading location of interest, which serves for both training and extrapolation.

3. Methodology

In this section, we introduce a new PINN architecture for forward modeling of elastic wave propagation in semi-infinite domain. Considering the characteristics of seismological tasks, the ABC is applied on the truncated boundaries to eliminate wave reflection due to boundary effect. A domain decomposition training strategy is then introduced to handle large-scale simulations. Furthermore, we also propose a specifically-designed DNN-based framework for parametric surrogate modeling, which is slightly different from vanilla PINN used for solving generic PDEs.

3.1. PINNs

DNNs have received considerable attention in the scientific computing community thanks to their theoretical universal approximation theory [57,58]. Previous implementations of DNNs for scientific tasks generally rely on a large amount of labeled data. Nevertheless, the recent breakthrough of PINNs [16] has enabled learning in a "small data" regime by incorporating physical constraints (e.g., governing equations and I/BCs) into the networks. Typically, a PINN architecture is composed of an input layer, multiple hidden layers and an output layer. Each layer is designated with specific number of neurons. For instance, the mathematical formulation of the connection between the $(i-1)$-th and the i-th hidden layer is given by

$$\mathbf{X}^i = \sigma\left(\mathbf{W}^i \mathbf{X}^{i-1} + \mathbf{b}^i\right), \quad i \in [1, n], \tag{8}$$

where $\{\mathbf{W}^i, \mathbf{b}^i\}$ denote the corresponding weight matrix and bias vector at the i-th layer, respectively; $\sigma(\cdot)$ is a nonlinear activation function; n is the number of hidden layers. Here, $\{\mathbf{X}^{i-1}, \mathbf{X}^i\}$ are the input and output variables at the i-th layer, respectively. Note that there is no activation function for the output layer (i.e., the last layer). The general goal of PINNs is to leverage DNNs to approximate the solutions of PDEs of interest with spatial or spatiotemporal input coordinates (e.g., $\{t, x, y\}$ for 2D time-dependent dynamics). Herein, the difference between general DNN-based solvers and PINNs lies in the introduction of the physics loss apart from data loss. The physics loss is obtained based on the strong form of PDEs and the derivative terms of interest are calculated via automatic differentiation [59]. We discuss the specific network architecture for our tasks in Section 3.3.

3.2. Absorbing boundary condition

As the seismic wave propagation problem commonly features a semi-infinite domain (e.g., in Earth), the computational domain needs to be truncated to avoid unnecessary computation. Nevertheless, one of the main obstacles is to guarantee that the wave propagates out the computational domain without interfering with the upstream field when it passes through truncated boundaries. To overcome this issue in numerical simulations, researchers have developed various mathematical formulations [55,60–62] to describe ABCs on the truncated boundaries. Recent research on PINNs [53,63] has started to explore the integration of ABCs in wave propagation applications. The illustrative diagram of the ABC is presented in Fig. 1. In this paper, we consider one of the most popular ABCs, which is given by

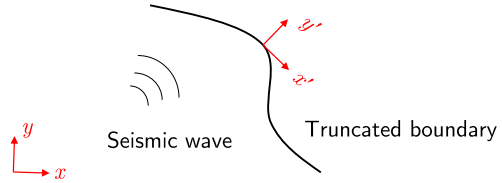


Fig. 1. Illustrative diagram of the absorbing boundary condition. The local coordinate system is determined by x' (tangential direction) and y' (outer normal direction) axes. The x and y axes define the global coordinate.

$$\mathbf{u}_{ty'} + \mathbf{C}_1 \mathbf{u}_{tt} + \mathbf{C}_2 \mathbf{u}_{tx'} + \mathbf{C}_3 \mathbf{u}_{x'x'} = \mathbf{0}, \tag{9}$$

where x' (tangential direction) and y' (outer normal direction) axes determine the local coordinate system. In addition, $\{\mathbf{C}_1, \mathbf{C}_2, \mathbf{C}_3\}$ are coefficients matrices reflecting the material properties of the medium, written as

$$\mathbf{C}_1 = \begin{bmatrix} 1/\beta & 0 \\ 0 & 1/\alpha \end{bmatrix}, \ \mathbf{C}_2 = (\beta - \alpha) \begin{bmatrix} 0 & 1/\beta \\ 1/\alpha & 0 \end{bmatrix}, \ \mathbf{C}_3 = \frac{1}{2} \begin{bmatrix} \beta - 2\alpha & 0 \\ 0 & \alpha - 2\beta \end{bmatrix}. \tag{10}$$

Note that the ABC described in Eq. (9) is defined by using the local coordinate system. Therefore, a proper coordinate transformation is required for automatic differentiation in PINNs. By introducing the ABC into the network, we are free from simulating a large computational domain with PINNs. Moreover, since the PINN deals with the strong form of the governing equation, Eq. (9) can be used readily to impose the corresponding BCs. In contrast, the variational form of the analytical law is required in traditional finite element analysis (FEA), which brings additional complexity to numerical implementation.

3.3. Network architecture

Furthermore, the overall architecture of our designed PINNs for modeling seismic wave propagation is presented in Fig. 2. Three separate DNNs are employed to approximate the variables of displacement $\mathbf{u}$, velocity $\mathbf{v}$, and stress $\boldsymbol{\sigma}$ considering their different numerical scales. After obtaining these field variables, we utilize the automatic differentiation [59] to evaluate the partial derivatives of interests with respect to the input coordinates. Based on the differential terms, we construct a loss function $\mathcal{L}(\mathbf{W}, \mathbf{b})$ of the concerned physical laws in PINN, including the governing equations and I/BCs (including the ABCs). In specific, $\mathcal{L}_e(\mathbf{W}, \mathbf{b})$ is a mean square error (MSE) loss that follows the analytical expressions of elastodynamics as shown in Eq. (5). Such a loss component helps preserve the inherent PDE structure. The I/BC loss $\mathcal{L}_c(\mathbf{W}, \mathbf{b})$ is also in MSE form that measures the misfit in the context of the given I/BCs. It is noteworthy that this type of PINNs does not rely on observation data loss in forward simulation. Namely, only $\mathcal{L}_e$ and $\mathcal{L}_c$ are sufficient to constrain the network and also respect the consistent physical principles. The optimal network parameters $\{\mathbf{W}^*, \mathbf{b}^*\}$ are learned by minimizing the combined loss function, as follows

$$\{\mathbf{W}^*, \mathbf{b}^*\} = \arg\min_{\{\mathbf{W}, \mathbf{b}\}} \{\eta_0 \mathcal{L}_e(\mathbf{W}, \mathbf{b}) + \eta \mathcal{L}_c(\mathbf{W}, \mathbf{b})\}, \tag{11}$$

where η_0 and η are weighting coefficients (i.e., user-defined hyperparameters).

3.4. Sequential training via domain decomposition

In this part, we present the sequential training scheme via domain decomposition. The reasons for applying domain decomposition are

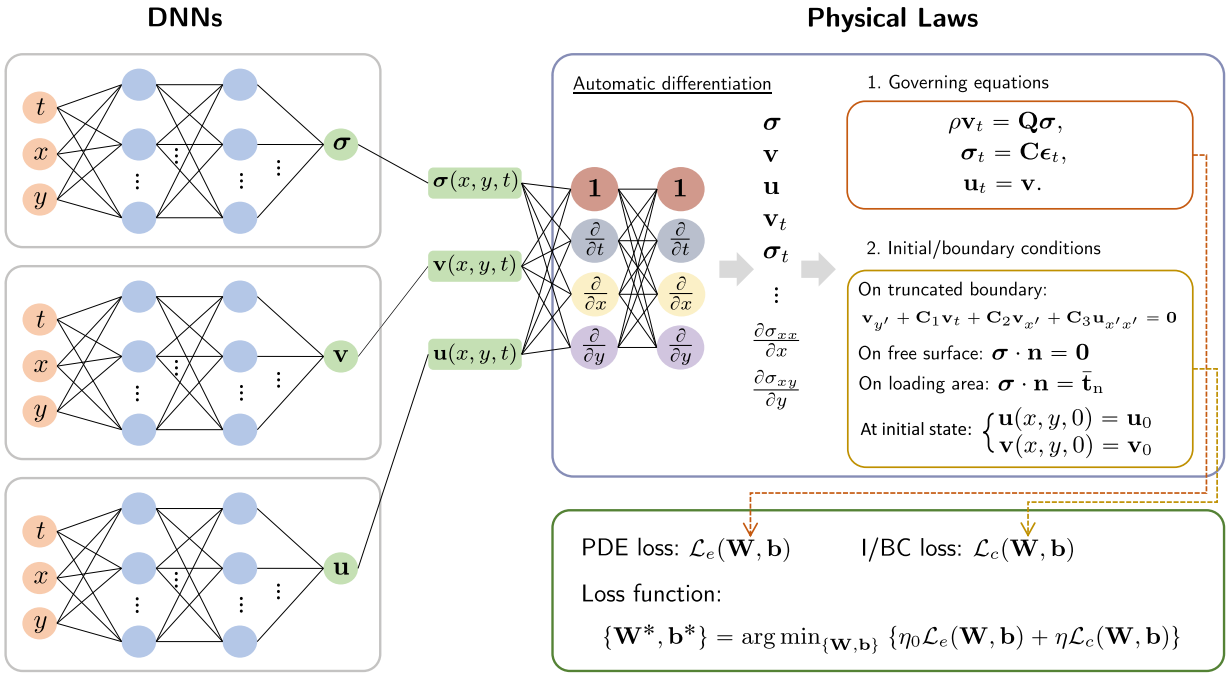


Fig. 2. Overview of the proposed PINN framework for seismic wave propagation, including the network architecture and the physics-informed loss function. Three separate DNNs are used to approximate the displacement $\mathbf{u}$, velocity $\mathbf{v}$, and stress $\boldsymbol{\sigma}$, respectively. Automatic differentiation is exploited for obtaining the derivative terms and further constructing the loss function. Note that the governing equations are formulated in the first-order representation, while the ABC is expressed in the second-order formulation. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

two-fold. Firstly, the scientific problems in geophysics are usually defined in large physical domains, especially seismic wave propagation. Thus, the computational cost of numerically estimating the solutions of such tasks could be extremely high, which poses a major challenge to PINNs. In addition, the global approximation property of PINNs would lead to a heavy burden on the computational memory of computers/servers. To overcome these computation bottlenecks in large-scale scientific problems, we propose a scalable sequential training strategy via domain decomposition, as shown in Fig. 3. The domain decomposition approach has been employed in prior research works [64–66] within spatial domains, where interface points are sampled to constrain the continuity of solutions. It can reduce the computational cost and facilitate the convergence of PINNs. Noteworthy, domain decomposition is a well-established concept in traditional numerical methods for parallel computation of PDEs [67].

Nevertheless, different from the aforementioned strategy, we adopt domain decomposition along the time dimension in this paper. This is because we observe more regularity in temporal interfaces compared with spatial domains. To be more specific, we keep the temporal interfaces between two sub-domains identical, which facilitates the generation of the interface points used for "stitching" PINNs. Furthermore, a particular PINN is employed to approximate the solution of each subdomain, and the final global solution is obtained by stitching each local solution from every PINN together.

As discussed in [65], there exist several benefits of using the domain decomposition scheme, such as better representation capability, improved accuracy of solution due to the local approximation, the flexibility of choosing hyperparameters for each sub-domain, and the convenience for multi-GPU parallelization. However, these benefits are enjoyed at the cost of more coding work. Specifically, an additional loss component $\mathcal{L}_{\mathcal{T}}$, interface loss that computes the discrepancy of solutions on interfaces from adjacent sub-domains, is included to guarantee the continuity of solutions on the interfaces, e.g., $\mathcal{T}_i$ $(i = 1, 2, 3)$ in Fig. 3. Hence, considering m subdomains, the overall loss function can be rewritten as

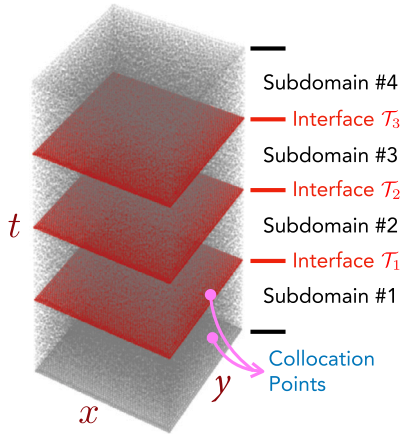


Fig. 3. Schematic of temporal domain decomposition.

$$\mathcal{L}(\mathbf{W},\mathbf{b}) = \eta_0 \underbrace{\sum_{i=1}^{m} \mathcal{L}_e^i(\mathbf{W}_i,\mathbf{b}_i)}_{\text{equation loss}} + \eta_1 \underbrace{\sum_{i=1}^{m} \mathcal{L}_c^i(\mathbf{W}_i,\mathbf{b}_i)}_{\text{I/BC loss}} + \eta_2 \underbrace{\sum_{i=1}^{m-1} \mathcal{L}_{\mathcal{T}}^i(\mathbf{W}_i,\mathbf{W}_{i+1},\mathbf{b}_i,\mathbf{b}_{i+1})}_{\text{interface loss}}, \tag{12}$$

where η's denote the loss weighting coefficients; $\mathcal{L}_e^i$ and $\mathcal{L}_c^i$ represent the residual loss functions of governing equations and I/BCs, respectively, for the i-th sub-domain. The loss function for the i-th interface is expressed as

$$\mathcal{L}_{\mathcal{T}}^i(\mathbf{W}_i,\mathbf{W}_{i+1},\mathbf{b}_i,\mathbf{b}_{i+1}) = \sum_{j=1}^{N_{\mathcal{T}}} \left\| \mathbf{z}_{\mathcal{T}}^{i+1}(\mathbf{W}_{i+1},\mathbf{b}_{i+1};t_j,\mathbf{x}_j) - \mathbf{z}_{\mathcal{T}}^{i}(\mathbf{W}_i,\mathbf{b}_i;t_j,\mathbf{x}_j) \right\|_2^2, \tag{13}$$

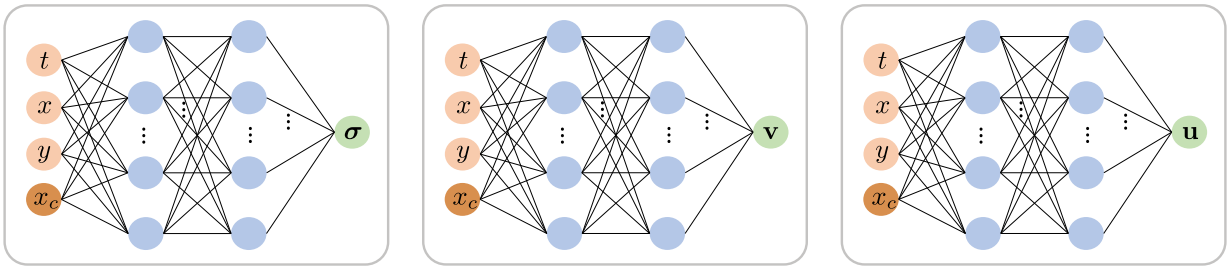


Fig. 4. The DNNs part of our proposed surrogate modeling scheme. Here we only consider the spatial variance in x-dimension of a 2D domain, which means $\mathbf{x}_c = [x_c, y_c]$ and $y_c = 0$. In addition, $\{t, x, y\}$ are spatiotemporal coordinates and $\{\boldsymbol{\sigma}, \mathbf{v}, \mathbf{u}\}$ denote the output variables. There are three different DNNs for learning the corresponding solution variables.

Here, $\{t_j, \mathbf{x}_j\}$ denotes the j-th collocation point on the interface, $j = 1, 2, ..., N_{\mathcal{T}}$; $\mathbf{z}_{\mathcal{T}}^i$ and $\mathbf{z}_{\mathcal{T}}^{i+1}$ represent the physical variables, e.g., $\{\mathbf{u}, \mathbf{v}, \boldsymbol{\sigma}\}$, for the i-th and the $(i+1)$-th sub-domains, respectively. Note that $\mathbf{z}$ can be also augmented to account for the time/spatial derivatives of the physical variables to further enhance the continuity condition at the interface.

In general, our decomposition strategy on the time domain is easy to implement and retains excellent performance in forward seismic wave propagation. More details on numerical experiments and results can be found in Section 4.

3.5. Surrogate modeling for parametric loading

Previously, Sun et al. [19] presented a new direction for surrogate modeling of fluid flows using PINNs. However, the potential of PINNs for surrogate modeling in the field of solid mechanics is still under-explored though the vanilla PINN [16] shows promises for forward and inverse analysis of various PDE systems. To this end, we propose a new PINN strategy for surrogate modeling of elastic wave propagation. Our goal is to directly predict the corresponding seismic responses under different loading scenarios based on the well-trained PINN model. In particular, we consider different source locations while keeping the load profile the same. This is commonly seen in seismic wave inversion at the engineering scale (e.g., collecting waveforms given a pre-defined load profile applied at different locations on the Earth's surface [68]).

An overview of our designed surrogate modeling scheme is illustrated in Fig. 4. Let us define a new variable $\mathbf{x}_c$ to represent the spatial coordinate of the loading position. $\mathbf{x}_c$ is also considered as an input variable apart from the spatiotemporal coordinate $\{x, y, t\}$ used in Section 3.3. Then, PINNs are utilized to approximate the solutions with respect to the loading scenario in the training stage. In the vanilla PINN collocation points are usually sampled in the spatiotemporal domain to impose the governing equations and I/BCs [16]. As the parametric variable (i.e., $\mathbf{x}_c$) is introduced to the surrogate model, we also need to sample different values for the parametric variable (e.g., $\mathbf{x}_c \in \{-5, 0, 5\}$) to ensure the governing equations and I/BCs are enforced for different loading positions. In other words, $\mathbf{x}_c$ serves as an additional input variable to construct the residual loss of the essential BC on the loading area. It is also worthwhile to mention that the computational cost increases drastically as the dimensionality of the input increases. After training the network, we directly infer the specific seismic responses by changing $\mathbf{x}_c$ to other loading positions of interest. The inspiration of this scheme is that the loading positions at training and extrapolation stages share linear relationships and DNNs are capable of capturing the solution variances caused by the change of spatial information [69].

4. Numerical experiments and results

In this section, a set of numerical examples are implemented to evaluate the capability of our designed PINN architectures for modeling elastic wave propagation in semi-infinite domain. We consider the 2D subsurface for all the numerical experiments. In particular, two distinctive material distributions (i.e., Case 1 and Case 2) are designed for solving the elastic wave equations within a truncated domain, and two parametric loading experiments (i.e., Case 3 and Case 4) are displayed for validating the effectiveness of surrogate modeling. Moreover, the ABC described in Section 3.2 is adopted in PINNs to avoid wave reflection near the truncated boundaries. We also present the corresponding finite element solutions simulated in enlarged domains to examine the accuracy of PINN results. All the numerical experiments are programmed in TensorFlow [70] and conducted on an NVIDIA Tesla V100 GPU card (32G) in a standard workstation.

4.1. Domain definition

To simplify, we consider the same physical domain and loading scenario for four numerical cases in this paper. The subsurface is excited by the normal traction of the Ricker wavelet type on the surface, which concentrates near the mid-line of the domain (i.e., $x \in [-8, 8]$, unit: [m]). The analytical expression of the normal stress applied on the surface is

$$\bar{\mathbf{t}}_n(x, 0, t) = T_0 \left[1 - 2\pi^2 (t - t_s)^2 F_r^2\right] \mathrm{e}^{-\pi^2 (t-t_s)^2 F_r^2 - (\frac{x}{L})^2} \text{ [MPa]}, \tag{14}$$

where T_0 is the amplitude of the normal traction, L is the length scale of the normal traction, t_s is the offset of the Ricker wavelet and F_r is the frequency of the excitation. To be more specific, we define $T_0 = 2.0$ [MPa], $L = 1.8$ [m], $t_s = 0.1$ [s] and $F_r = 15$ [Hz]. Moreover, the material density ρ and Poisson's ratio ν are set to be 0.002 [g/mm^3] and 0.25, respectively.

Furthermore, the computational domain is truncated into $x \times y \in [-25, 25] \times [0, 50]$ (unit: [m]) to avoid unnecessary computation. In addition to the surface under normal traction, the remaining three edges are modeled as absorbing boundaries with the ABC as shown in Eq. (9). Note that we adopt the unit system of [mm], [MPa], and [ms] in the simulation to ensure the output variables (i.e., displacement, velocity, and stress) having similar numerical scales. In addition, the time duration for Case 1 and Case 2 is defined as 400 [ms]. For parametric loading, we reduce it to 200 [ms] since our main goal is to evaluate the performance of our framework in the context of surrogate modeling. It is noteworthy that the material distribution for each case is different and described as follows.

4.2. The sampling of collocation points

In this paper, we integrate the traditional LHS method and Gaussian Quadrature [71] in FEA for sampling collocation points. The reason for such a hybrid strategy is that we observe more uniformly distributed collocation points based on Gaussian Quadrature compared with the LHS method. Therefore, we discretize the entire spatiotemporal domain with quadrilateral and then sample Gaussian points. In addition, we consider using the LHS approach as a supplement for adding more collocation points near the source of the wavefield.

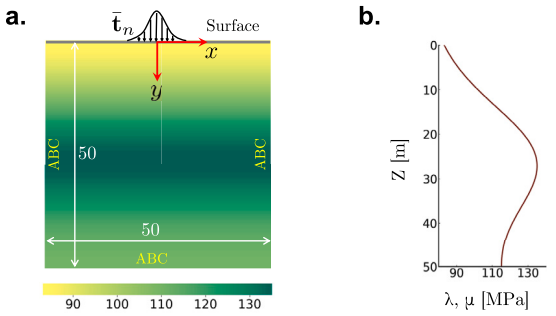
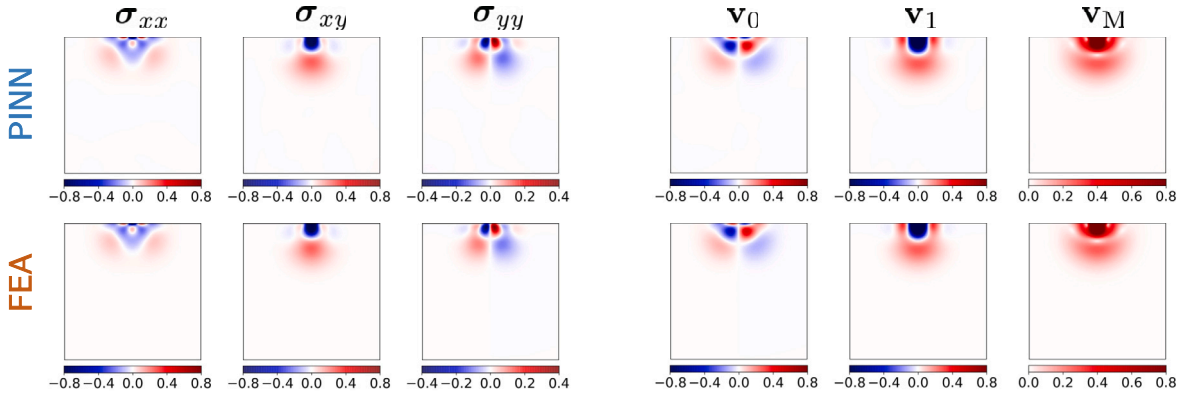
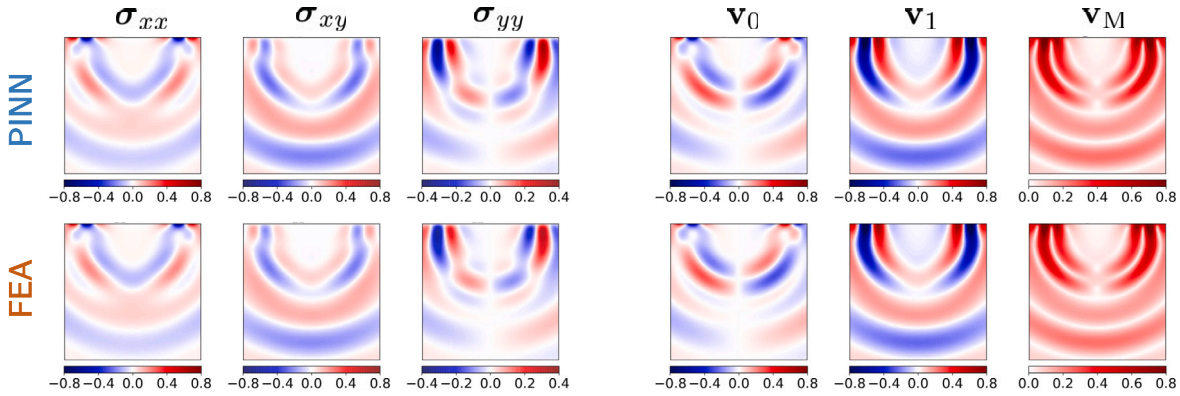


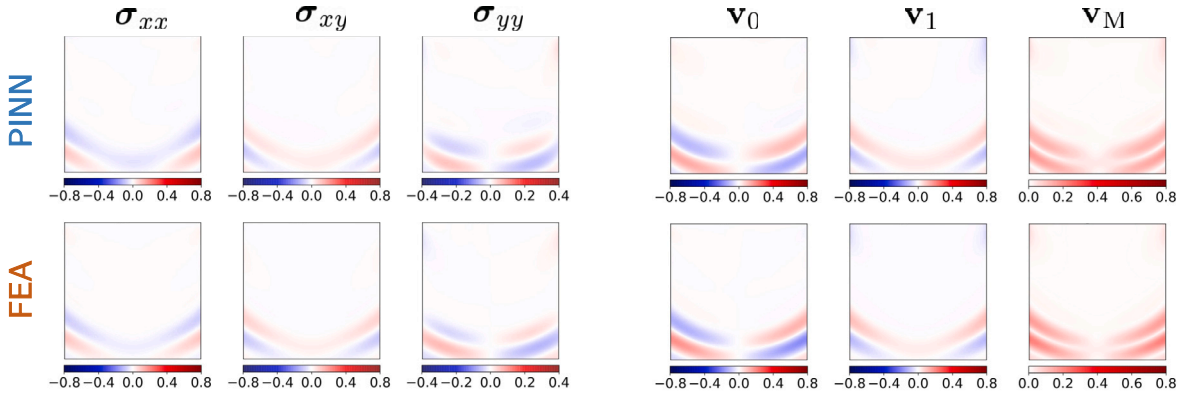
Fig. 5. The computational domain with a mixed material distribution of linear and Gaussian functions. **a**. The spatial domain is defined with $x \times y \in [-25, 25] \times [0, 50]$ (unit: [m]). ABC denotes the absorbing boundary condition. **b**. The material distribution is presented along the depth y.



(a) $t = 0.10$ s



(b) $t = 0.20$ s



(c) $t = 0.30$ s

Fig. 6. The results of stress and velocity distributions obtained from our proposed PINN and FEA in Case 1. We select three representative snapshots (i.e., $t = 0.10, 0.20, 0.30$) for comparison. $\mathbf{v}_0$, $\mathbf{v}_1$, and $\mathbf{v}_M$ denote the horizontal velocity, the vertical velocity, and the magnitude respectively.

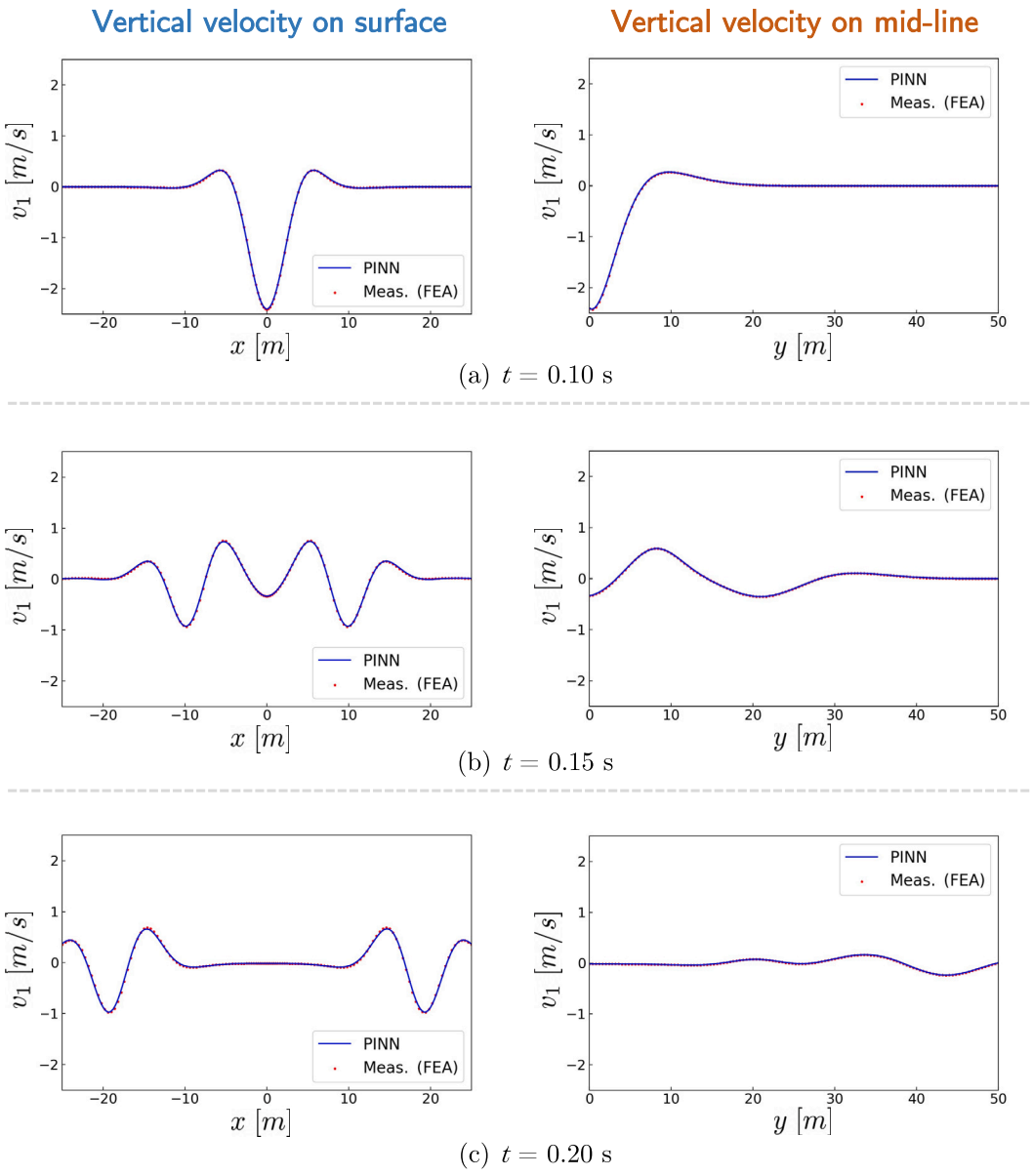


Fig. 7. Vertical velocity distributions (i.e., $\mathbf{v}_1$) at various time steps on the surface ($y = 0$, left) and mid-line ($x = 0$, right) in Case 1. The synthetic measurement is obtained from FEA on an enlarged domain ($x \times y \in [-75, 75] \times [0, 100]$, unit: [m]).

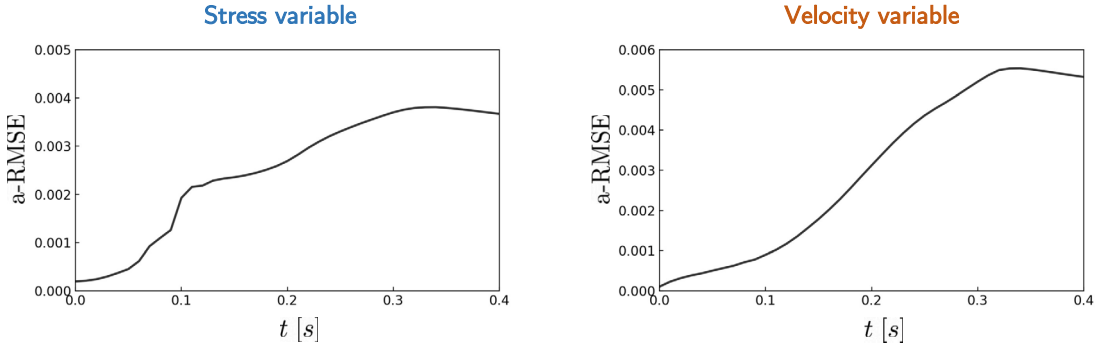


Fig. 8. Error propagation of our proposed PINN framework for stress and velocity variables in Case 1.

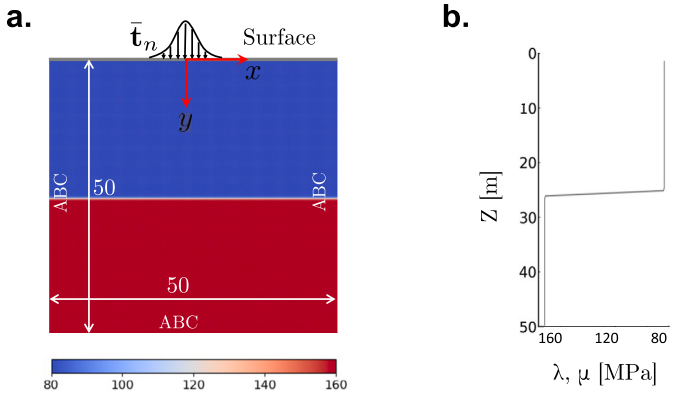
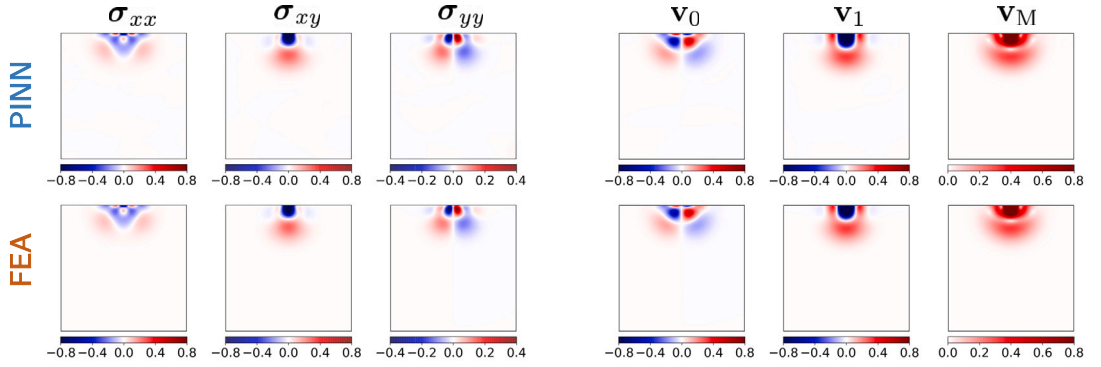
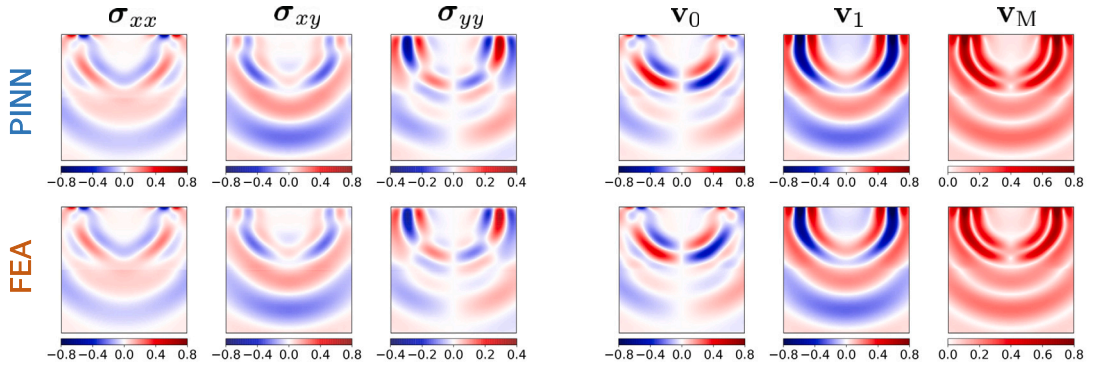


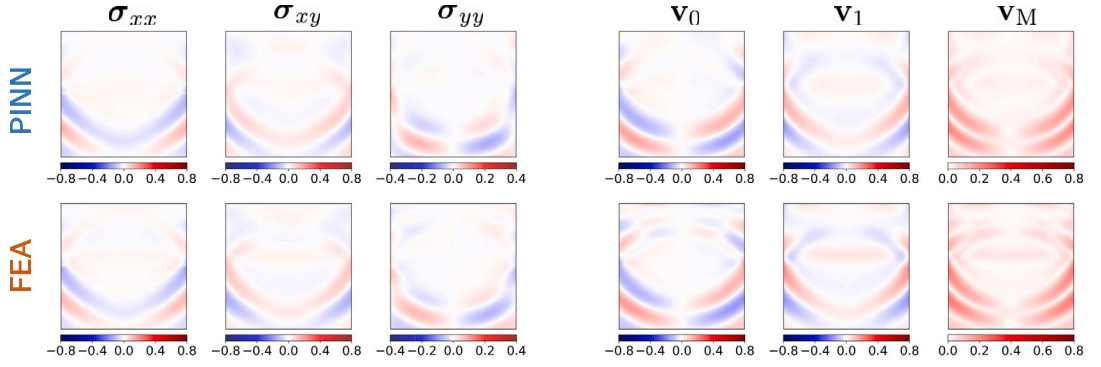
Fig. 9. The illustrative diagram of the 2D subsurface with two layers. **a**. The spatial domain is defined with $x \times y \in [-25, 25] \times [0, 50]$ (units: [m]). ABC denotes the absorbing boundary condition. **b**. The material distribution is presented along the depth y.



(a) $t = 0.10$ s



(b) $t = 0.20$ s



(c) $t = 0.30$ s

Fig. 10. The results of stress and velocity distributions obtained from our proposed PINN and FEA in Case 2. We select three representative snapshots (i.e., $t = 0.10, 0.20, 0.30$ [s]) for comparison. $\mathbf{v}_0$, $\mathbf{v}_1$, and $\mathbf{v}_M$ denote the horizontal velocity, the vertical velocity, and the magnitude respectively.

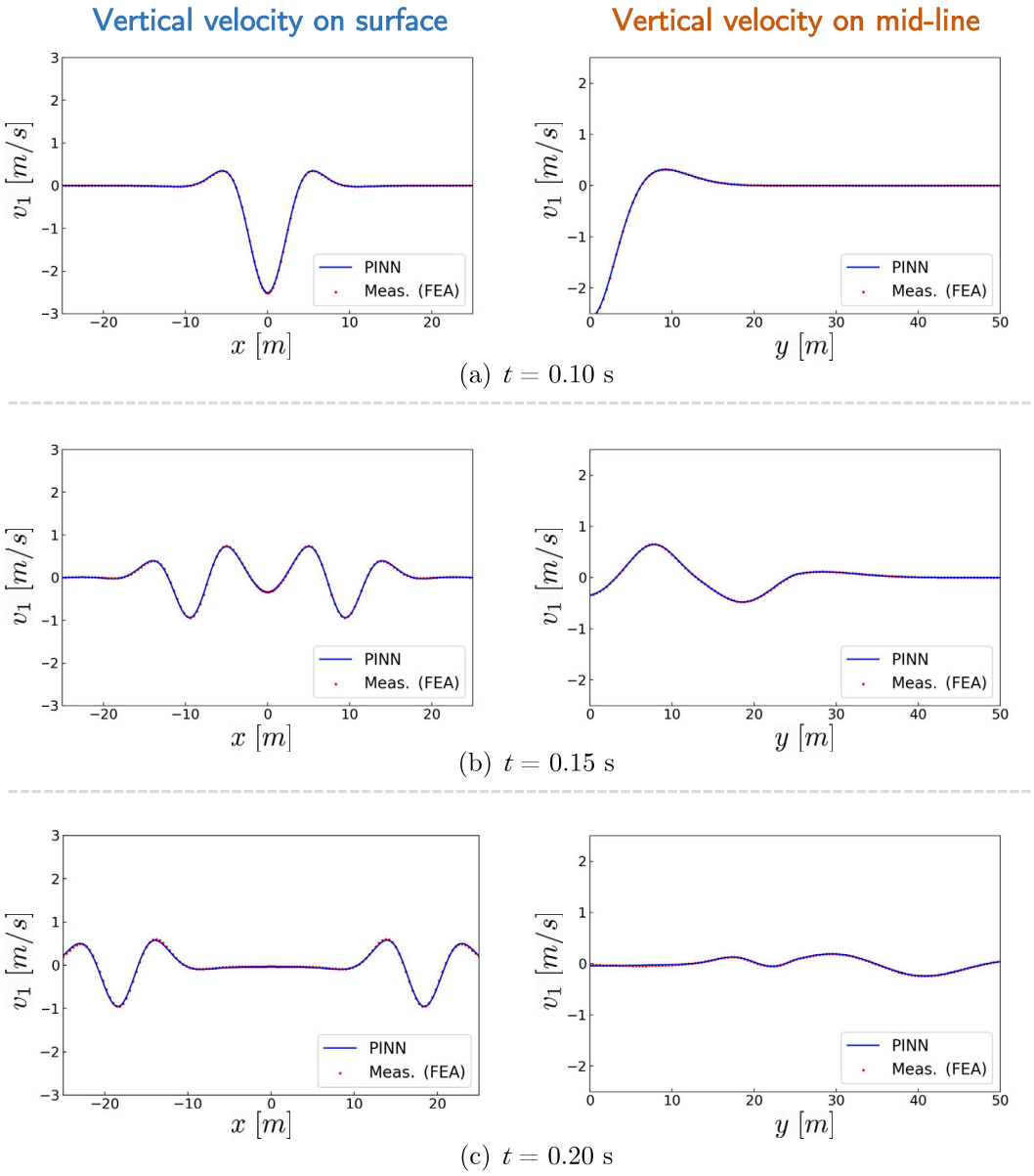


Fig. 11. Vertical velocity distributions (i.e., $\mathbf{v}_1$) at various time steps on the surface ($y = 0$, left) and mid-line ($x = 0$, right) in Case 2. To obtain the PINN prediction, the truncated domain is decomposed into two parts along the time dimension. Each part is modeled with a separate PINN.

4.3. Evaluation metric

We define an accumulative root-mean-square error (a-RMSE) to measure the error propagation between the learned waveforms and the ground truth, expressed as

$$\text{a-RMSE}(t_i) = \sqrt{\text{MSE}\left(\widehat{\boldsymbol{\mathcal{V}}}^{(1:i)} - \boldsymbol{\mathcal{V}}_{\text{ref}}^{(1:i)}\right)} \tag{15}$$

where $\widehat{\boldsymbol{\mathcal{V}}}^{(1:i)}$ and $\boldsymbol{\mathcal{V}}_{\text{ref}}^{(1:i)}$ denote the predicted and the reference solutions, respectively, from the 1st to the i^{th} step. It describes the misfit of all snapshots before the time instance t_i. Note that the ground truth data are simulated with numerical solvers.

4.4. Solving equations

4.4.1. Case 1 – linear and Gaussian material distribution

In the first example, we consider a 2D subsurface with the material property following a combination of linear and Gaussian functions, where the Lamé constants of the media are given by

$$\lambda(x, y) = \mu(x, y) = 80 + 0.64y + 38.4\mathrm{e}^{-\frac{(y-25)^2}{250}}\ [\text{MPa}]. \tag{16}$$

The graphic illustration of the physical domain and the material distribution is exhibited in Fig. 5. Note that this material distribution is relatively simple and smooth. Therefore, domain decomposition is not applied here since there is no harsh requirement for network size and collocation points. In specific, we employ two separate deep networks for learning the displacement $\mathbf{u}$ and the velocity $\mathbf{v}$. Each network consists of four hidden layers with 16, 80, 80, and 80 neurons respectively. Moreover, another independent network is designed for approximating the stress variable $\boldsymbol{\sigma}$, where four hidden layers (24, 120, 120, and 120 neurons) are used. Noteworthy, a larger network is considered for learning $\boldsymbol{\sigma}$ due to the inherent complexity of the stress variable. We sample $218,084$ collocation points within the physical domain to enforce the governing equations, among which $50,000$ points are near the wave source. In addition, $10,000$, $58,431$, and $12,000$ points are sampled on IC, each absorbing boundary and the surface boundary, respectively. The sampling method for collocation points is slightly different from the Latin hypercube sampling (LHS) strategy [72], which is

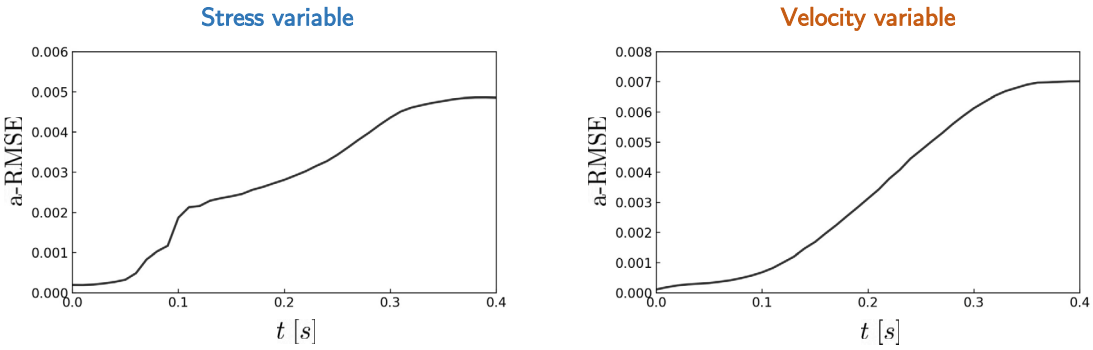


Fig. 12. Error propagation of our proposed PINN framework for stress and velocity variables in Case 2.

typically used in PINN applications. The related details are elaborated in 4.2.

Furthermore, we apply Adam [73] and L-BFGS-B [74] optimizers to sequentially train the networks. In Case 1, 4,500 Adam and 35,000 L-BFGS-B iterations are used. Note that we consider the sine activation function [63,75] in the networks. The selection of hyper-parameters is described in Appendix A. What's more, to generate the reference solution, we also perform the numerical simulation of this problem by FEA using an enlarged computational domain (i.e., $x \times y \in [-75, 75] \times [0, 100]$, unit: [m]), where the waves propagate out the truncated domain naturally.

The comparative results of snapshots are shown in Fig. 6. It is obvious that the solution of PINN matches the ground truth very well. The excellent performance is further validated by comparing the vertical velocity distributions on the surface (i.e., $y = 0$ [m]) and the mid-line (i.e., $x = 0$ [m]) (see Fig. 7). The zoom-in results also exhibit a remarkable agreement between the prediction of PINN and the reference solution. In addition, we showcase the error propagation curves of stress $\boldsymbol{\sigma}$ and velocity $\mathbf{v}$ in Fig. 8. The a-RMSE errors of the snapshots increase mildly and remain generally small (i.e., < 0.005). For Case 1, with a relatively simple material distribution (i.e., linear and Gaussian function), the example shows the excellent solution accuracy of our proposed PINN architecture.

4.4.2. Case 2 – two-layer material distribution

The second numerical case is a two-layer subsurface model as shown in Fig. 9. More precisely, we prescribe the material property of the subsurface as

$$\lambda(x, y) = \mu(x, y) = 80 + \frac{80}{1 + e^{-10(y-25)}}, \tag{17}$$

which is under the assumption of elasticity. In this case, due to the complexity of the material property, we consider using more collocation points than those in Case 1. Hence, in spite of being truncated, the computational domain is still very large and cannot fit into the memory (i.e., 32 GB) of one single Tesla V100 GPU. Therefore, we decompose the spatiotemporal domain into two parts along the time dimension (i.e., $t \in [0, 0.2] \cup [0.2, 0.4]$, units: [s]). The solution of each subdomain is approximated by a separate PINN as described in Sub-section 3.4.

Herein, we utilize the identical network architecture employed in Case 1 to learn the displacement ($\mathbf{u}$), the velocity ($\mathbf{v}$), and the stress ($\boldsymbol{\sigma}$) variables. Besides, 200,188 collocation points are sampled within the domain to evaluate the residual of the governing equations. 4,000 points are sampled on each absorbing boundary while 8,000 points are generated for the surface. In addition, 6,135 interface points are used to guarantee the continuity of the solution. Each PINN is trained with 150,000 iterations of L-BFGS-B. Note that the configuration introduced in this part is specified for each sub-domain.

Fig. 10 presents snapshots of the velocity and stress fields predicted by PINN and FEA at different time steps. We can see that the overall prediction by PINN agrees with the reference solution (i.e., FEA) very well.

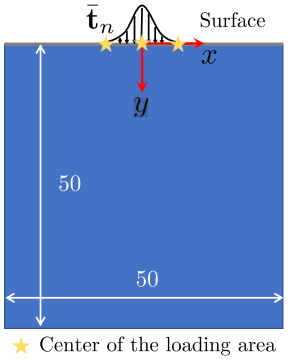


Fig. 13. The illustrative diagram of the 2D subsurface with homogeneous material for parametric loading. The yellow stars denote the centers of the loading areas. The truncated boundaries are regarded as free boundaries due to the homogeneous nature of the medium. The absorbing boundary condition is not required here.

It is interesting to point out that for $t = 0.30$ [s], the reference solution of FEA is characterized by more details on the reflected wave compared with the learned solution by PINN. This is because the reflected wave has a much smaller scale than the incident wave, which poses a major challenge to the optimization of the loss function [49]. This observation implies the limitation of the point-wise global approximation property of PINN. When scientific problems feature fine-scale patterns, the PINN that leverages fully-connected NNs as approximators might be unable to capture some delicate phenomena. The comparisons of the velocity distribution on the surface (i.e., $y = 0$ [m]) and mid-line (i.e., $x = 0$ [m]) are shown in Fig. 11 to quantitatively examine the accuracy of the proposed PINN. We observe that the prediction of PINN matches perfectly with the reference solution, which demonstrates its excellent accuracy. Moreover, the error propagation is displayed in Fig. 12. The error evolution is relatively larger than those of Case 1 due to a more complex material distribution, but the solution performance is still satisfactory thanks to the domain decomposition strategy.

4.5. Parametric loading

4.5.1. Case 3 – homogeneous material

For the parametric loading case, we consider a homogeneous material in the physical domain (see Fig. 13), where Young's modulus E is defined as 200 [MPa]. The spatial domain size is also set as $x \times y \in [-25, 25] \times [0, 50]$ (unit: [m]). The time duration for this case is 0.2 [s]. The domain decomposition is not applied here due to the relatively small domain size. Moreover, the loading follows the formulation of Eq. (14) with $T_0 = -2.0$ [MPa], $L = 5.0$ [m]. However, different from solving seismic wave propagation, we incorporate three excitation positions in $\mathbf{x}_c$ (e.g., $[-5, 0, 5]$, unit: [m]) in addition to the spatiotemporal coordinate information (i.e., $\{x, y, t\}$), which works as loading parame-

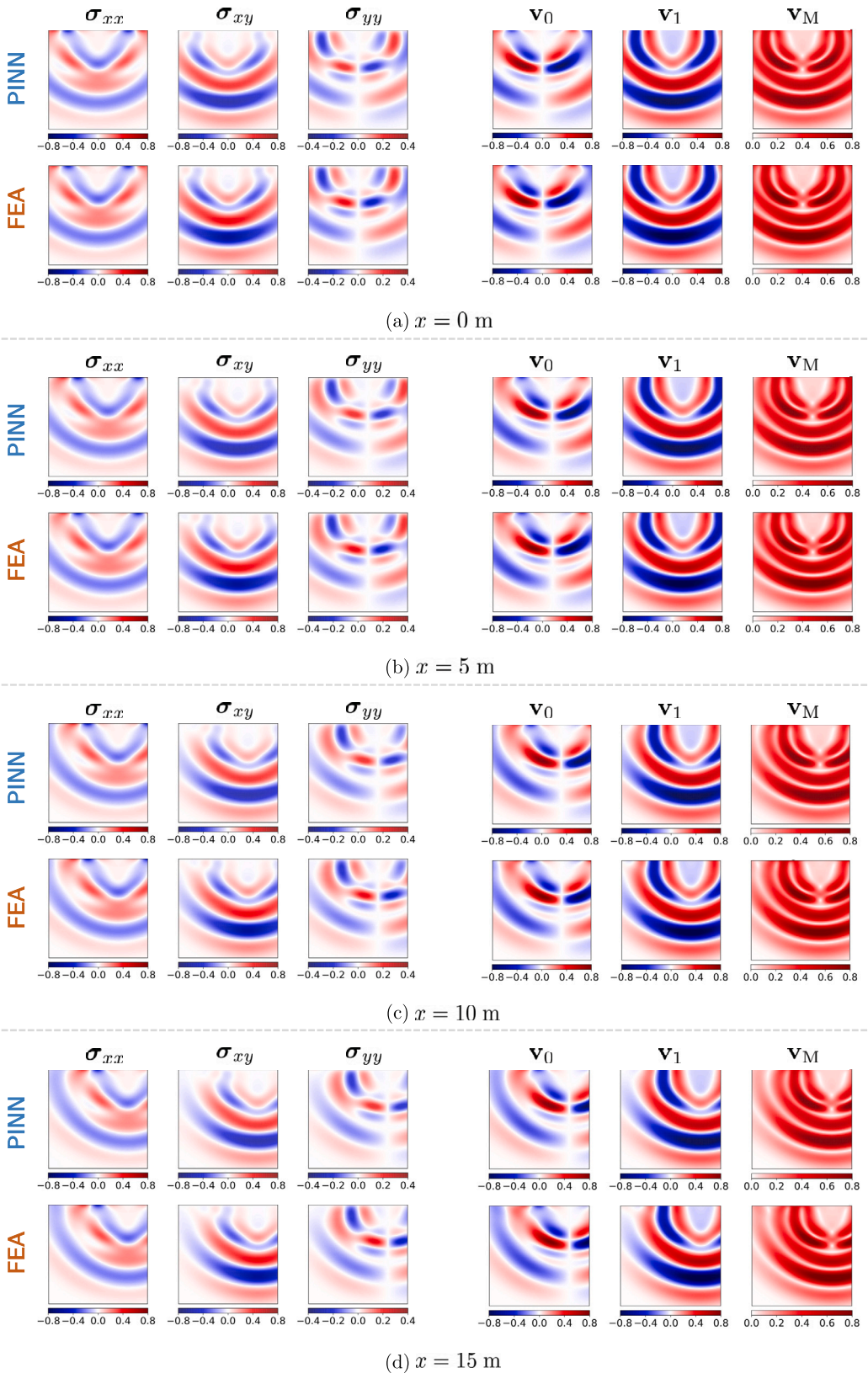


Fig. 14. The snapshots of stress and velocity from our proposed PINN and FEA in Case 3. We present two interpolation ($x = 0, 5$ [m]) and two extrapolation ($x = 10, 15$ [m]) results. The time step is select at $t = 0.2$ [s].

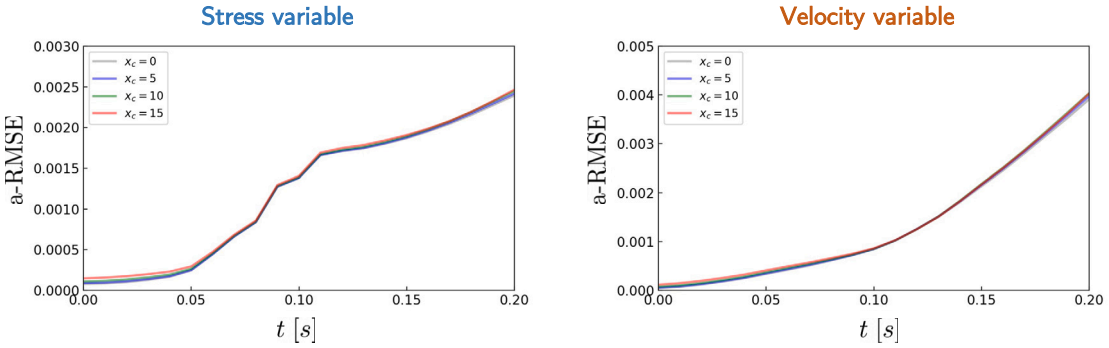


Fig. 15. Error propagation of our proposed PINN framework for stress and velocity variables in Case 3. Four loading scenarios are selected in the diagram, i.e., $x_c = 0, 5, 10, 15$ [m].

ters for the surrogate modeling. Each location is defined as the center of the loading area.

We apply three distinct fully-connected NNs to approximate three variables $\{\mathbf{u}, \mathbf{v}, \boldsymbol{\sigma}\}$ respectively, with five hidden layers per network. Each layer includes 100 neurons. We sample $152,043$ collocation points over the spatiotemporal domain and additional $50,000$ points near the wave source to evaluate the residual of the governing equations. Moreover, $10,500$ and $88,236$ points are sampled for IC and the surface boundary condition, respectively. We train the network with $60,000$ L-BFGS-B epochs until convergence. In specific terms, we define three loading positions of $\mathbf{x}_c = \{-5, 0, 5\}$ (unit: [m]) to train the network. Based on the trained model, we directly infer/extrapolate the dynamics under other loading positions (e.g., $\mathbf{x}_c = 10, 15$ [m]).

The results of parametric loading are shown in Figs. 14 and 15. We show representative snapshots (i.e., $t = 0.2$ [s]) of two training ($x_c = 0, 5$ [m]) and two extrapolation scenarios ($x_c = 10, 15$ [m]). As shown in Fig. 14, both the training and extrapolation results from our proposed PINN match the corresponding ground truth well. It is worthwhile to mention that the extrapolation of $x_c = 15$ [m] maintains excellent solution accuracy, which proves the capability of our method for surrogate modeling. In addition, the propagation error in Fig. 15 further validates the great performance in the context of solving the wave equations under parametric loading. In particular, the error propagations of extrapolation present similar evolving patterns as those of the training scenarios. The errors of the stress variable increase mildly, while the errors of the velocity variable show relatively large increases but are still acceptable.

4.5.2. Case 4 – inhomogeneous material

We further assess the effectiveness of our proposed method on an inhomogeneous material with properties identical to Case 1. The linear and Gaussian distribution of the material is mathematically described in Eq. (16). Other experimental details remain consistent with Case 3.

In Case 4, we employ two separate 4-layer fully-connected neural networks to learn the displacement and velocity variables $(\mathbf{u}, \mathbf{v})$. Each network consists of hidden neurons arranged as 16, 100, 100, and 100. Additionally, another 4-layer network is utilized to estimate the stress variable $(\boldsymbol{\sigma})$, with hidden neurons set to 16, 150, 150, and 150. The configuration of collocation points remains the same as in Case 3. For training, we utilize 60,000 L-BFGS-B epochs, and we consider three excitation locations at $\mathbf{x}_c$ ($[-5, 0, 5]$, unit: [m]) to learn the surrogate model. Furthermore, we extrapolate the seismic dynamics to other loading positions (e.g., $\mathbf{x}_c = 10, 15$ [m]).

Fig. 16 displays the representative snapshots at $t = 0.2$ [s] for two training scenarios ($x_c = 0, 5$ [m]) and two extrapolation scenarios ($x_c = 10, 15$ [m]). The proposed PINN approach demonstrates excellent agreement with the corresponding ground truth in both training and extrapolation. Similarly to Case 3, even at $x_c = 15$ [m], the extrapolation results also match the ground truth well, which further validates the effectiveness of our surrogate modeling strategy. Moreover, Fig. 17 illustrates the error propagation of stress and velocity variables under parametric loading. While the extrapolation shows slightly larger error propagation compared to the training scenarios, the overall performance remains satisfactory.

Table 1
Root mean square error of the PINN prediction against the finite element solution. The prediction error is evaluated on the entire spatiotemporal domain and absorbing boundaries (i.e., left, right and lower edges).

Model	Entire domain		Boundaries	
	Velocity	Stress	Velocity	Stress
w/ ABC	5.2×10^{-3}	3.7×10^{-3}	7.2×10^{-3}	6.2×10^{-3}
w/o ABC	8.1×10^{-3}	6.0×10^{-3}	1.0×10^{-2}	9.3×10^{-3}

4.6. Ablation study

In this part, we present an ablation study to verify the effectiveness of the absorbing boundary condition (ABC) introduced in Section 3.2. Specifically, we train a network without the ABC being enforced[2] using the numerical example of Case 1. Note that the other settings (e.g., collocation points, network size, optimizer, etc.) are kept the same for these two networks. Table 1 compares the prediction error against the finite element solution between the regular and ablated PINN. It can be seen that the proposed PINN (w/ ABC) consistently outperforms the ablated PINN regarding the prediction error on the entire spatiotemporal domain and on the artificial boundaries. In addition, the comparison of the velocity distribution at the lower edge is provided in Fig. 18 where the result with ABC agrees well with the reference solution. To conclude, the introduction of ABC is critical to the accurate prediction of PINN.

5. Discussions

We herein discuss the current limitations of the proposed PINN framework and the outlook of our future work. Generally, the main potential concerns come from (1) the soft imposition scheme of enforcing I/BCs and (2) the issues of slow convergence and scalability due to the setting of fully-connected NNs. More precisely, the typical PINNs consider the physical principles as loss regularizer terms in the optimization. However, it is challenging for such an implementation to guarantee the learned dynamics rigorously following the underlying physical laws. This observation is empirically validated by a recent work [76] which analyzes the failure modes of PINNs. Therefore, instead of using a soft enforcement strategy, we would like to explore the possibility of encoding the known physics (e.g., the PDE structure)

[2] The loss function Eqn. (A.1) would no longer have $\mathcal{L}_{abc}$ term.

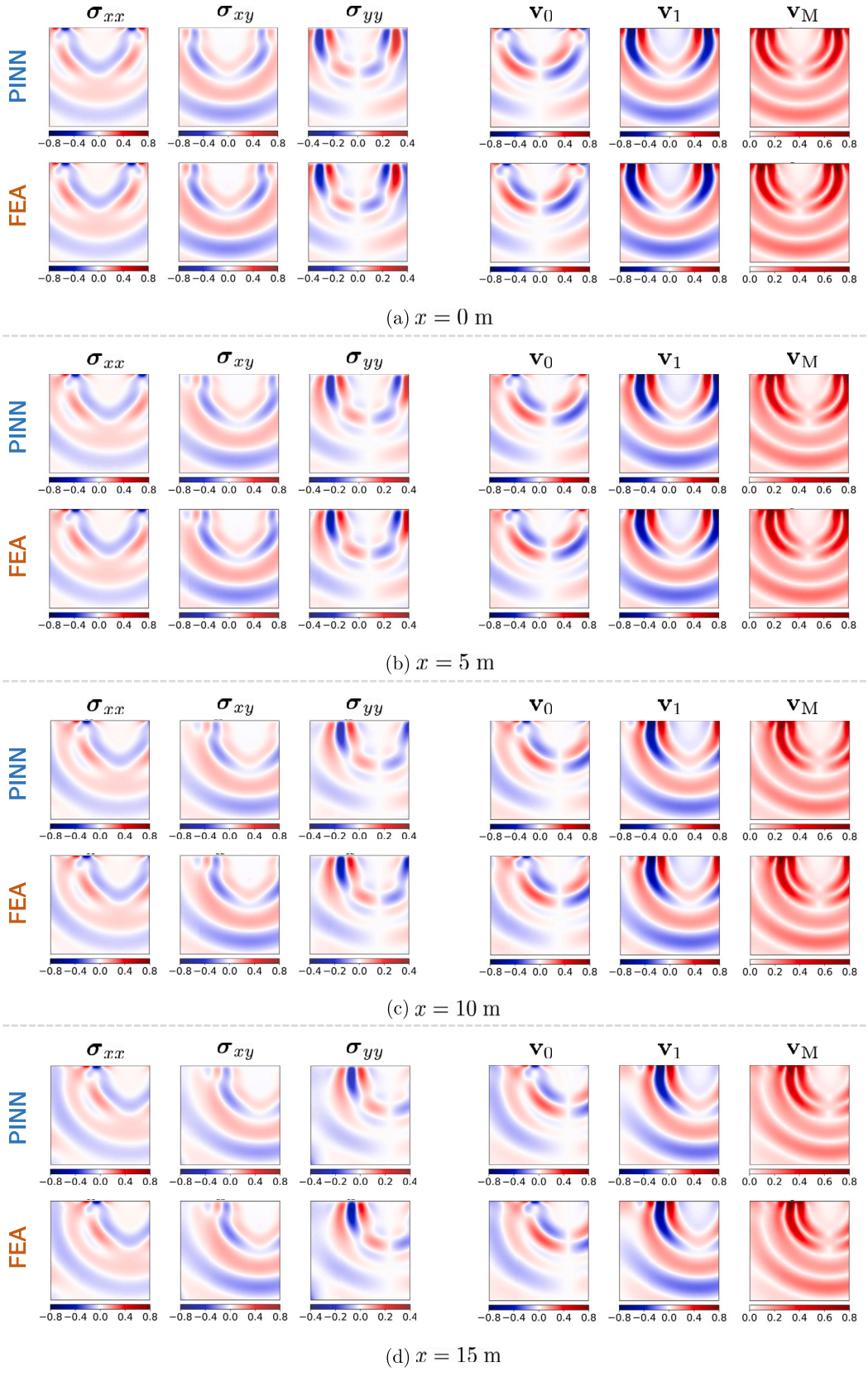


Fig. 16. The snapshots of stress and velocity from our proposed PINN and FEA in Case 4. We present two interpolation ($x = 0, 5$ [m]) and two extrapolation ($x = 10, 15$ [m]) results. The time step is select at $t = 0.2$ [s].

into the networks [44]. In addition, PINNs employ fully-connected NNs for approximating the solution of PDEs, which can be essentially regarded as a continuous learning scheme. They are excellent in capturing global patterns but show inferior performance in learning local details compared with discrete learning methods (e.g., convolution-based NNs) [23]. Moreover, applying discrete learning has the potential to

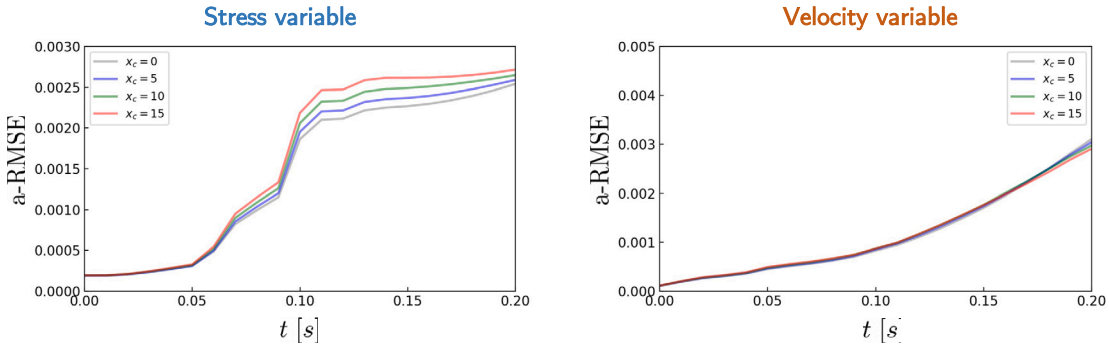
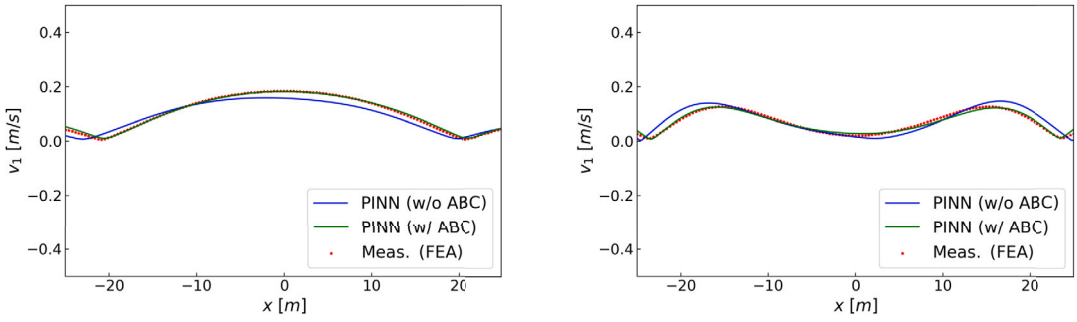


Fig. 17. Error propagation of our proposed PINN framework for stress and velocity variables in Case 4. Four loading scenarios are selected in the diagram, i.e., $x_c = 0, 5, 10, 15$ [m].



(a) 200 ms (b) 300 ms

Fig. 18. Vertical velocity distribution on the lower edge.

mitigate the computation burden of learning seismic wave propagation in large domains.

Noteworthy, it is also challenging for the current PINN architectures to simulate high-frequency seismic wave propagation. For instance, when considering a near-surface model with a depth of 50 m, the common peak frequency is around 100 Hz. This limitation arises due to the low-frequency bias of existing PINNs [77]. To address this issue, the utilization of physics-informed discrete learning shows promise, as it enables the learning of multi-scale features and information across different frequencies. In the future, we will put more research efforts into building discrete learning frameworks (e.g., convolutional and graph NNs) for forward analysis of spatiotemporal dynamics. In addition, our future work will also be placed on extending the proposed model to solve full-wave inversion problems.

6. Conclusions

We proposed a new PINN architecture for forward modeling of seismic wave propagation in a semi-infinite domain. It is capable of both solving the elastic wave equations and parametric surrogate modeling within truncated domains. There are three characteristics highlighted: (1) the introduction of the ABC into the network as a soft constraint to eliminate boundary effect and avoid expensive computation in semi-infinite domains; (2) a new sequential training scheme via temporal domain decomposition to improve scalability and solution accuracy; (3) a parametric surrogate modeling scheme to predict the seismic responses under different loading scenarios. Note that the entire network is trained without any labeled data. Furthermore, we evaluate the performance of our proposed PINN architectures on various numerical cases with different material distributions. The results demonstrate the effectiveness of our approach in the context of solution accuracy and extrapolation capability.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Yang Liu reports financial support was provided by Fundamental Research Funds for the Central Universities.

Data availability

All the datasets and source codes to reproduce the results in this study are available on GitHub at https://github.com/paulpuren/seismic_modeling upon final publication.

Acknowledgements

The work is supported by the National Natural Science Foundation of China (No. 62276269) and the Beijing Natural Science Foundation (No. 1232009). P. Ren would like to acknowledge the support by the Vilas Mujumdar Fellowship at Northeastern University and the constructive discussion with Dr. Jin-Yun Xie. Y. Liu would like to thank the support by the Fundamental Research Funds for the Central Universities at University of Chinese Academy of Sciences (No. E2EG2202X2). H. Sun would like to acknowledge the support by the Fundamental Research Funds for the Central Universities at Renmin University of China (No. 202230265).

Appendix A. The selection of hyper-parameters

The general principle of selecting hyper-parameters (i.e., the weighting coefficients for loss terms) is to ensure that each weighted loss term shares a similar numerical scale. In our implementation, we first initialize the entire network with Xavier's method [78] and then train the specific PINN with one iteration in order to evaluate the magnitudes of loss terms. Next, the weighting coefficients are defined by making the

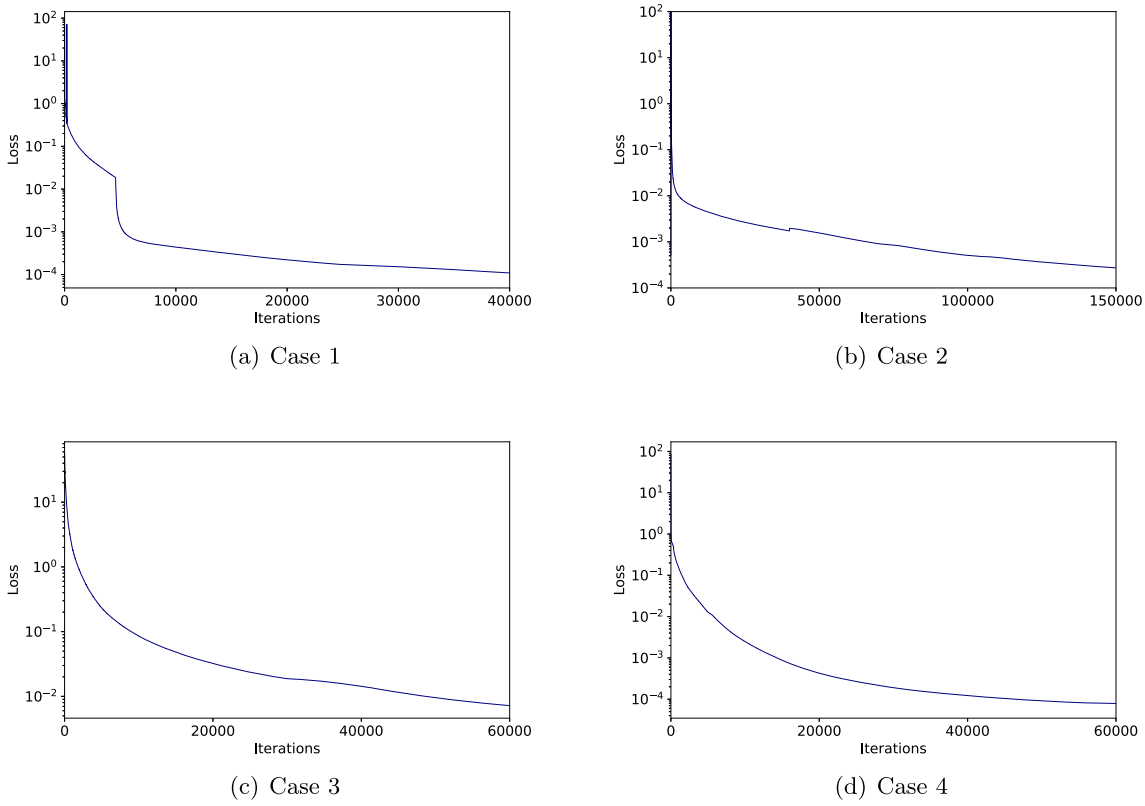


Fig. B.1. Convergence curves of loss functions.

Table A.1
The details of weighting hyper-parameters for four numerical cases. "N/A" represents being inapplicable to the specific index.

	η_0	η_1	η_2	η_3	η_4	η_5	η_6
Case 1	10000	1	10000	5	10	10	10000000
Case 2	10000	1	10000	5	10	10	10000000
Case 3	4000000	200	1000000	200	400	400	N/A
Case 4	4000000	200	1000000	200	400	400	10000000

products of the coefficients and the corresponding loss terms close to unity. The hyper-parameters used in this study are presented in Table A.1. We extend the loss function in Eq. (11) to a more specific formulation, which is defined as

$$\mathcal{L} = \eta_0\mathcal{L}_e + \eta_1\mathcal{L}_{ic} + \eta_2\mathcal{L}_s + \eta_3\mathcal{L}_v + \eta_4\mathcal{L}_{nb} + \eta_5\mathcal{L}_{src} + \eta_6\mathcal{L}_{abc}, \tag{A.1}$$

where $\{\mathcal{L}_{ic}, \mathcal{L}_s, \mathcal{L}_v, \mathcal{L}_{nb}, \mathcal{L}_{src}, \mathcal{L}_{abc}\}$ denote the loss terms for IC, displacement-stress equation, velocity-displacement equation, natural boundary condition, wave source condition and the absorbing boundary condition, respectively. $\{\eta_1, \eta_2, \eta_3, \eta_4, \eta_5, \eta_6\}$ are their corresponding hyper-parameters. Note that absorbing boundary condition is not required for homogeneous material distribution [22] in Case 3 (i.e., parametric loading experiment).

Appendix B. Convergence history

Fig. B.1 displays the convergence history of the four numerical cases. In Case 1, the ADAM optimizer is used for the first 4,500 iterations. For Cases 2 to 4, L-BFGS-B is used throughout the training to minimize the loss function.

References

[1] K.R. Kelly, R.W. Ward, S. Treitel, R.M. Alford, Synthetic seismograms: a finite-difference approach, Geophysics 41 (1) (1976) 2–27.
[2] J. Virieux, P-sv wave propagation in heterogeneous media: velocity-stress finite-difference method, Geophysics 51 (4) (1986) 889–901.
[3] H. Bao, J. Bielak, O. Ghattas, L.F. Kallivokas, D.R. O'Hallaron, J.R. Shewchuk, J. Xu, Large-scale simulation of elastic wave propagation in heterogeneous media on parallel computers, Comput. Methods Appl. Mech. Eng. 152 (1–2) (1998) 85–102.
[4] D. Komatitsch, J.-P. Vilotte, The spectral element method: an efficient tool to simulate the seismic response of 2d and 3d geological structures, Bull. Seismol. Soc. Am. 88 (2) (1998) 368–392.
[5] D. Komatitsch, J. Tromp, Introduction to the spectral element method for three-dimensional seismic wave propagation, Geophys. J. Int. 139 (3) (1999) 806–822.
[6] Z.E. Ross, M.-A. Meier, E. Hauksson, P wave arrival picking and first-motion polarity determination with deep learning, J. Geophys. Res., Solid Earth 123 (6) (2018) 5120–5129.
[7] W. Zhu, G.C. Beroza, Phasenet: a deep-neural-network-based seismic arrival-time picking method, Geophys. J. Int. 216 (1) (2019) 261–273.
[8] K.J. Bergen, P.A. Johnson, M.V. de Hoop, G.C. Beroza, Machine learning for data-driven discovery in solid earth geoscience, Science 363 (6433) (2019) eaau0323.
[9] Q. Kong, D.T. Trugman, Z.E. Ross, M.J. Bianco, B.J. Meade, P. Gerstoft, Machine learning in seismology: turning data into insights, Seismol. Res. Lett. 90 (1) (2019) 3–14.
[10] S.M. Mousavi, W.L. Ellsworth, W. Zhu, L.Y. Chuang, G.C. Beroza, Earthquake transformer—an attentive deep-learning model for simultaneous earthquake detection and phase picking, Nat. Commun. 11 (1) (2020) 1–12.
[11] S.M. Mousavi, G.C. Beroza, Deep-learning seismology, Science 377 (6607) (2022) eabm4470.
[12] Oleg Ovcharenko, Vladimir Kazei, Mahesh Kalita, Daniel Peter, Tariq Alkhalifah, Deep learning for low-frequency extrapolation from multioffset seismic data, Geophysics 84 (6) (2019) R989–R1001.
[13] Bingbing Sun, Tariq Alkhalifah, ML-descent: an optimization algorithm for full-waveform inversion using machine learning, Geophysics 85 (6) (2020) R477–R492.
[14] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, A. Anandkumar, Fourier neural operator for parametric partial differential equations, arXiv preprint, arXiv:2010.08895, 2020.
[15] Y. Yang, A.F. Gao, J.C. Castellanos, Z.E. Ross, K. Azizzadenesheli, R.W. Clayton, Seismic wave propagation and inversion with neural operators, Seismic Rec. 1 (3) (2021) 126–134.
[16] M. Raissi, P. Perdikaris, G.E. Karniadakis, Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, J. Comput. Phys. 378 (2019) 686–707.
[17] G.E. Karniadakis, I.G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, L. Yang, Physics-informed machine learning, Nat. Rev. Phys. 3 (6) (2021) 422–440.

[18] Y. Zhu, N. Zabaras, P.-S. Koutsourelakis, P. Perdikaris, Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data, J. Comput. Phys. 394 (2019) 56–81.
[19] L. Sun, H. Gao, S. Pan, J.-X. Wang, Surrogate modeling for fluid flows based on physics-constrained deep learning without simulation data, Comput. Methods Appl. Mech. Eng. 361 (2020) 112732.
[20] N. Geneva, N. Zabaras, Modeling the dynamics of pde systems with physics-constrained deep auto-regressive networks, J. Comput. Phys. 403 (2020) 109056.
[21] H. Gao, L. Sun, J.-X. Wang, Phygeonet: physics-informed geometry-adaptive convolutional neural networks for solving parameterized steady-state pdes on irregular domain, J. Comput. Phys. 428 (2021) 110079.
[22] C. Rao, H. Sun, Y. Liu, Physics-informed deep learning for computational elastodynamics without labeled data, J. Eng. Mech. 147 (8) (2021) 04021043.
[23] P. Ren, C. Rao, Y. Liu, J.-X. Wang, H. Sun, Phycrnet: physics-informed convolutional-recurrent network for solving spatiotemporal pdes, Comput. Methods Appl. Mech. Eng. 389 (2022) 114399.
[24] R. Wang, K. Kashinath, M. Mustafa, A. Albert, R. Yu, Towards physics-informed deep learning for turbulent flow prediction, in: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2020, pp. 1457–1466.
[25] M. Raissi, A. Yazdani, G.E. Karniadakis, Hidden fluid mechanics: learning velocity and pressure fields from flow visualizations, Science 367 (6481) (2020) 1026–1030.
[26] C. Rao, H. Sun, Y. Liu, Physics-informed deep learning for incompressible laminar flows, Theor. Appl. Mech. Lett. 10 (3) (2020) 207–212.
[27] N.B. Erichson, M. Muehlebach, M.W. Mahoney, Physics-informed autoencoders for Lyapunov-stable fluid flow prediction, arXiv preprint, arXiv:1905.10866, 2019.
[28] X. Jin, S. Cai, H. Li, G.E. Karniadakis, Nsfnets (Navier-Stokes flow nets): physics-informed neural networks for the incompressible Navier-Stokes equations, J. Comput. Phys. 426 (2021) 109951.
[29] S. Wang, S. Sankaran, P. Perdikaris, Respecting causality is all you need for training physics-informed neural networks, arXiv preprint, arXiv:2203.07404, 2022.
[30] A.M. Tartakovsky, C.O. Marrero, P. Perdikaris, G.D. Tartakovsky, D. Barajas-Solano, Physics-informed deep neural networks for learning parameters and constitutive relationships in subsurface flow problems, Water Resour. Res. 56 (5) (2020) e2019WR026731.
[31] R. Zhang, Y. Liu, H. Sun, Physics-guided convolutional neural network (phycnn) for data-driven seismic response modeling, Eng. Struct. 215 (2020) 110704.
[32] R. Zhang, Y. Liu, H. Sun, Physics-informed multi-lstm networks for metamodeling of nonlinear structures, Comput. Methods Appl. Mech. Eng. 369 (2020) 113226.
[33] E. Haghighat, M. Raissi, A. Moure, H. Gomez, R. Juanes, A physics-informed deep learning framework for inversion and surrogate modeling in solid mechanics, Comput. Methods Appl. Mech. Eng. 379 (2021) 113741.
[34] C.-T. Chen, G.X. Gu, Learning hidden elasticity with deep neural networks, Proc. Natl. Acad. Sci. 118 (31) (2021) e2102721118.
[35] S. Cai, Z. Wang, L. Lu, T.A. Zaki, G.E. Karniadakis, Deepm&mnet: inferring the electroconvection multiphysics fields based on operator approximation by neural networks, J. Comput. Phys. 436 (2021) 110296.
[36] C. Lin, Z. Li, L. Lu, S. Cai, M. Maxey, G.E. Karniadakis, Operator learning for predicting multiscale bubble growth dynamics, J. Chem. Phys. 154 (10) (2021) 104118.
[37] Z. Mao, L. Lu, O. Marxen, T.A. Zaki, G.E. Karniadakis, Deepm&mnet for hypersonics: predicting the coupled flow and finite-rate chemistry behind a normal shock using neural-network approximation of operators, J. Comput. Phys. 447 (2021) 110698.
[38] S. Cai, Z. Wang, S. Wang, P. Perdikaris, G.E. Karniadakis, Physics-informed neural networks for heat transfer problems, J. Heat Transf. 143 (6) (2021).
[39] Q. He, D. Barajas-Solano, G. Tartakovsky, A.M. Tartakovsky, Physics-informed neural networks for multiphysics data assimilation with application to subsurface transport, Adv. Water Resour. 141 (2020) 103610.
[40] Q. He, A.M. Tartakovsky, Physics-informed neural network method for forward and backward advection-dispersion equations, Water Resour. Res. 57 (7) (2021) e2020WR029479.
[41] M. Raissi, Deep hidden physics models: deep learning of nonlinear partial differential equations, J. Mach. Learn. Res. 19 (1) (2018) 932–955.
[42] F. Sun, Y. Liu, H. Sun, Physics-informed spline learning for nonlinear dynamics discovery, arXiv preprint, arXiv:2105.02368, 2021.
[43] Z. Chen, Y. Liu, H. Sun, Physics-informed learning of governing equations from scarce data, Nat. Commun. 12 (2021) 6136.
[44] C. Rao, P. Ren, Y. Liu, H. Sun, Discovering nonlinear pdes from scarce data with physics-encoded learning, arXiv preprint, arXiv:2201.12354, 2022.
[45] S. Esmaeilzadeh, K. Azizzadenesheli, K. Kashinath, M. Mustafa, H.A. Tchelepi, P. Marcus, M. Prabhat, A. Anandkumar, et al., Meshfreeflownet: a physics-constrained deep continuous space-time super-resolution framework, in: SC20: International Conference for High Performance Computing, Networking, Storage and Analysis, IEEE, 2020, pp. 1–15.
[46] C. Rao, H. Sun, Y. Liu, Embedding physics to learn spatiotemporal dynamics from sparse data, arXiv preprint, arXiv:2106.04781, 2021.
[47] P. Ren, C. Rao, Y. Liu, Z. Ma, Q. Wang, J.-X. Wang, H. Sun, Physics-informed deep super-resolution for spatiotemporal data, arXiv preprint, arXiv:2208.01462, 2022.
[48] H. Gao, L. Sun, J.-X. Wang, Super-resolution and denoising of fluid flow using physics-informed convolutional neural networks without high-resolution labels, Phys. Fluids 33 (7) (2021) 073603.
[49] Tariq Alkhalifah, Chao Song, Umair bin Waheed, Qi Hao, Wavefield solutions from machine learned functions constrained by the Helmholtz equation, Artif. Intell. Geosci. 2 (2021) 11–19.
[50] U. bin Waheed, E. Haghighat, T. Alkhalifah, C. Song, Q. Hao, Eikonal solution using physicsinformed neural networks, arXiv preprint, arXiv:2007.08330, 2020.
[51] J.D. Smith, K. Azizzadenesheli, Z.E. Ross, Eikonet: solving the Eikonal equation with deep neural networks, IEEE Trans. Geosci. Remote Sens. (2020).
[52] C. Song, T. Alkhalifah, U.B. Waheed, Solving the frequency-domain acoustic vti wave equation using physics-informed neural networks, Geophys. J. Int. 225 (2) (2021) 846–859.
[53] M. Rasht-Behesht, C. Huber, K. Shukla, G.E. Karniadakis, Physics-informed neural networks (pinns) for wave propagation and full waveform inversions, J. Geophys. Res., Solid Earth 127 (5) (2022) e2021JB023120.
[54] Zhen-dong Zhang, Mamdoh Alajami, Tariq Alkhalifah, Wave-equation dispersion spectrum inversion for near-surface characterization using fibre-optics acquisition, Geophys. J. Int. 222 (2) (2020) 907–918.
[55] D. Givoli, J.B. Keller, Non-reflecting boundary conditions for elastic waves, Wave Motion 12 (3) (1990) 261–279.
[56] J.N. Reddy, Energy Principles and Variational Methods in Applied Mechanics, John Wiley & Sons, 2017.
[57] G. Cybenko, Approximation by superpositions of a sigmoidal function, Math. Control Signals Syst. 2 (4) (1989) 303–314.
[58] K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators, Neural Netw. 2 (5) (1989) 359–366.
[59] A.G. Baydin, B.A. Pearlmutter, A.A. Radul, J.M. Siskind, Automatic differentiation in machine learning: a survey, J. Mach. Learn. Res. 18 (2018) 1–43.
[60] B. Engquist, A. Majda, Absorbing boundary conditions for numerical simulation of waves, Proc. Natl. Acad. Sci. 74 (5) (1977) 1765–1766.
[61] R. Clayton, B. Engquist, Absorbing boundary conditions for acoustic and elastic wave equations, Bull. Seismol. Soc. Am. 67 (6) (1977) 1529–1540.
[62] B. Engquist, A. Majda, Radiation boundary conditions for acoustic and elastic wave calculations, Commun. Pure Appl. Math. 32 (1979) 313–357.
[63] Chao Song, Tariq Alkhalifah, Umair Bin Waheed, A versatile framework to solve the Helmholtz equation using physics-informed neural networks, Geophys. J. Int. 228 (3) (2022) 1750–1762.
[64] A.D. Jagtap, G.E. Karniadakis, Extended physics-informed neural networks (xpinns): a generalized space-time domain decomposition based deep learning framework for nonlinear partial differential equations, Commun. Comput. Phys. 28 (5) (2020) 2002–2041.
[65] A.D. Jagtap, E. Kharazmi, G.E. Karniadakis, Conservative physics-informed neural networks on discrete domains for conservation laws: applications to forward and inverse problems, Comput. Methods Appl. Mech. Eng. 365 (2020) 113028.
[66] E. Kharazmi, Z. Zhang, G.E. Karniadakis, hp-vpinns: variational physics-informed neural networks with domain decomposition, Comput. Methods Appl. Mech. Eng. 374 (2021) 113547.
[67] Barry F. Smith, Domain decomposition methods for partial differential equations, in: Parallel Numerical Algorithms, Springer, 1997, pp. 225–243.
[68] P.-T. Trinh, R. Brossier, L. Métivier, L. Tavard, J. Virieux, Efficient time-domain 3D elastic and viscoelastic full-waveform inversion using a spectral-element method on flexible Cartesian-based mesh, Geophysics 84 (1) (2019) R75–R97.
[69] Chao Song, Tariq A. Alkhalifah, Wavefield reconstruction inversion via physics-informed neural networks, IEEE Trans. Geosci. Remote Sens. 60 (2021) 1–12.
[70] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al., {TensorFlow}: a system for {Large-Scale} machine learning, in: 12th USENIX Symposium on Operating Systems Design and Implementation, in: OSDI, vol. 16, 2016, pp. 265–283.
[71] T.J. Hughes, The Finite Element Method: Linear Static and Dynamic Finite Element Analysis, Courier Corporation, 2012.
[72] M.D. McKay, R.J. Beckman, W.J. Conover, A comparison of three methods for selecting values of input variables in the analysis of output from a computer code, Technometrics 42 (1) (2000) 55–61.
[73] D.P. Kingma, J. Ba, Adam: a method for stochastic optimization, arXiv preprint, arXiv:1412.6980, 2014.
[74] C. Zhu, R.H. Byrd, P. Lu, J. Nocedal, Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization, ACM Trans. Math. Softw. 23 (4) (1997) 550–560.
[75] V. Sitzmann, J. Martel, A. Bergman, D. Lindell, G. Wetzstein, Implicit neural representations with periodic activation functions, Adv. Neural Inf. Process. Syst. 33 (2020) 7462–7473.
[76] A. Krishnapriyan, A. Gholami, S. Zhe, R. Kirby, M.W. Mahoney, Characterizing possible failure modes in physics-informed neural networks, Adv. Neural Inf. Process. Syst. 34 (2021) 26548–26560.
[77] Xinquan Huang, Tariq Alkhalifah, PINNup: robust neural network wavefield solutions using frequency upscaling and neuron splitting, J. Geophys. Res., Solid Earth 127 (6) (2022) e2021JB023703.
[78] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, JMLR Workshop and Conference Proceedings, 2010, pp. 249–256.