Physics-Informed Neural Networks for Approximating Dynamic (Hyperbolic) PDEs of Second Order in Time: Error Analysis and Algorithms

Yanxia Qian[a], Yongchao Zhang[b], Yunqing Huang[a,*], Suchuan Dong[c,*]

[a]*School of Mathematics and Computational Science, Xiangtan University, Xiangtan, Hunan, 411105, P.R. China*
[b]*School of Mathematics, Northwest University, Xi'an, Shaanxi 710069, P.R. China*
[c]*Center for Computational and Applied Mathematics, Department of Mathematics, Purdue University, West Lafayette, IN47907, USA*

Abstract

We consider the approximation of a class of dynamic partial differential equations (PDEs) of second order in time by the physics-informed neural network (PINN) approach, and provide an error analysis of PINN for the wave equation, the nonlinear Klein-Gordon equation and the linear elastodynamic equation. Our analyses show that, with feed-forward neural networks having two hidden layers and the tanh activation function, the PINN approximation errors for the solution field, its time derivative and its gradient field can be effectively bounded by the training loss and the number of training data points (quadrature points). Our analyses further suggest new forms for the training loss function, which contain certain residuals that are crucial to the error estimate but would be absent from the canonical PINN loss formulation. Adopting these new forms for the loss function leads to a variant PINN algorithm. We present ample numerical experiments with the new PINN algorithm for the wave equation, the Sine-Gordon equation and the linear elastodynamic equation, which show that the method can capture the solution well.

Keywords: PINN; neural network; error estimate; PDE; scientific machine learning

1. Introduction

Deep neural networks (DNN) have achieved a great success in a number of fields in science and engineering [36] such as natural language processing, robotics, computer vision, speech and image recognition, to name but a few. This has inspired a great deal of research efforts in the past few years to adapt such techniques to scientific computing. DNN-based techniques seem particularly promising for problems in higher dimensions, e.g. high-dimensional partial differential equation (PDE), since traditional numerical methods for high-dimensional problems can quickly become infeasible due to the exponential increase in the computational effort (so-called curse of dimensionality). Under these circumstances deep-learning algorithms can be helpful. In particular, the neural network-based approach for PDE problems provide implicit regularization and can alleviate and perhaps overcome the curse of high dimensions [3, 4].

As deep neural networks are universal function approximators, it is natural to employ them as ansatz spaces for solutions of (ordinary or partial) differential equations. This paves the way for their use in physical modeling and scientific computing and gives rise to the field of scientific machine learning [32, 52, 46, 22, 37]. The physics-informed neural network (PINN) approach was introduced in [46]. It has been successfully applied to a variety of forward and inverse PDE problems and has become one of the most commonly-used methods in scientific machine learning (see e.g. [46, 25, 10, 31, 58, 30, 6, 53, 18, 16, 7, 55, 24, 34, 20, 21, 57, 19, 45, 51, 28, 44], among others). The references [32, 9] provide a comprehensive review of the literature on PINN and about the benefits and drawbacks of this approach.

*Authors of Correspondence.
Email addresses: yxqian0520@xtu.edu.cn (Yanxia Qian), yoczhang@nwu.edu.cn (Yongchao Zhang), huangyq@xtu.edu.cn (Yunqing Huang), sdong@purdue.edu (Suchuan Dong)

The mathematical foundation for PINN aiming at the approximation of PDE solution is currently an active area of research. It is important to account for different components of the neural-network error: optimization error, approximation error, and estimation error [42, 49]. Approximation error refers to the discrepancy between the exact functional map and the neural network mapping function on a given network architecture [8, 23]. Estimation error arises when the network is trained on a finite data set to get a mapping on the target domain. The generalization error is the combination of approximation and estimation errors and defines the accuracy of the neural-network predicted solution trained on the given set of data.

Theoretical understanding of PINN has been advanced by a number of recent works. In [49] Shin et al. rigorously justify why PINN works and shows its consistency for linear elliptic and parabolic PDEs under certain assumptions. These results are extended in [50] to a general abstract framework for analyzing PINN for linear problems with the loss function formulated in terms of the strong or weak forms of the equations. In [40] Mishra and Molinaro provide an abstract framework on PINN for forward PDE problems, and estimate the generalization error by means of the training error and the number of training data points. This framework is extended in [39] to study several inverse PDE problems, including the Poisson, heat, wave and Stokes equations. Bai and Koley [2] investigate the PINN approximation of nonlinear dispersive PDEs such as the KdV-Kawahara, Camassa-Holm and Benjamin-Ono equations. In [5] Biswa et al. provide explicit error estimates (in suitable norms) and stability analyses for the incompressible Navier–Stokes equations. Zerbinati [60] presents PINN as an under-determined point matching collocation method, reveals its connection with Galerkin Least Squares (GALS) method, and establishes an a priori error estimate for elliptic problems.

An important theoretical result on the approximation errors from the recent work [13] establishes that a feed-forward neural network $\hat{u}_\theta$ with a tanh activation function and two hidden layers may approximate a function u with a bound in a Sobolev space, $\|\hat{u}_{\theta N} - u\|_{w^{k,\infty}} \leq C\ln(cN)^k/N^{s-k}$. Here $u \in w^{s,\infty}([0,1]^d)$, d is the dimension of the problem, N is the number of training points, and $c, C > 0$ are explicitly known constants independent of N. Based on this result, De Ryck et al. [12] have studied the PINN for the Navier–Stokes equations and shown that a small training error implies a small generalization error. In particular, Hu et al. [26] provide the higher-order (spatial Sobolev norm) error estimates for the primitive equations, which improve the existing results in the PINN literature that only involve L^2 errors. In [14] it has been shown that, with a sufficient number of randomly chosen training points, the total L^2 error can be bounded by the generalization error for Kolmogorov-type PDEs, which in turn is bounded by the training error. It is proved that the size of the PINN and the number of training samples only increase polynomially with the problem dimension, thus enabling PINN to overcome the curse of dimensionality in this case. In [38] the authors investigate the high-dimensional radiative transfer equation and prove that the generalization error is bounded by the training error and the number of training points, where the upper bound depends on the dimension only through a logarithmic factor. Hence PINN does not suffer from the curse of dimensionality, provided that the training errors do not depend on the underlying dimension. Another interesting study on PINN and extended PINN (XPINN) is [27], in which the authors employ the generalized Barron space to define the function space of DNNs and have provided a prior and a posterior generalization bound on the PDE residual in terms of the complexity of the PDE solution and the posterior weight matrix norm in the neural network, respectively. Their analyses indicate that XPINN induces two opposing effects, with one tending to boost the network's generalization ability and the other tending to cause the network to be less generalizable.

Although PINN has been widely used for approximating PDEs, theoretical investigations on its convergence and errors are still quite limited and are largely confined to elliptic and parabolic PDEs. There seems to be less (or little) theoretical analysis on the convergence of PINN for hyperbolic type PDEs. In this paper, we consider a class of dynamic PDEs of second order in time, which are hyperbolic in nature, and provide an analysis of the convergence and errors of the PINN algorithm applied to such problems. We have focused on the wave equation, the nonlinear Klein-Gordon equation and the linear elastodynamic equation in our analyses. Building upon the result of [13, 12] on tanh neural networks with two hidden layers, we have shown that for these three kinds of PDEs:

- The PINN residuals can be made arbitrarily small with tanh neural networks having two hidden layers.
- The total error of the PINN approximation is bounded by the generalization error of PINN.
- The PINN approximation errors for the solution field, its time derivative and its gradient are bounded by the training error (training loss) and the number of quadrature points (training data points).

Furthermore, our theoretical analyses have suggested PINN training loss functions for these PDEs that are somewhat different in form than from the canonical PINN formulation. These lie in two aspects: (i) Our analyses require certain residual terms (such as the gradient of the initial condition, the time derivative of the boundary condition, or in the case of linear elastodynamic equation the strain and divergence of the initial condition) in the training loss, which would be absent from the canonical PINN formulation of the loss function. (ii) Our analyses may require, depending on the type of boundary conditions, the L^2 norm for certain boundary residuals in the training loss, which is different from the commonly-used L^2 norm squared in the canonical PINN formulation of the loss function.

These new forms for the training loss function suggested by the theoretical analyses lead to a variant PINN algorithm. We have implemented the PINN algorithm based on these new forms of the training loss function for the wave equation, the nonlinear Klein-Gordon equation and the linear elastodynamic equation. Ample numerical experiments based on this algorithm have been presented. The simulation results indicate that the method has captured the solution field reasonably well for these PDEs. The numerical results also to some extent corroborate the theoretical relation between the approximation error and the PINN training loss obtained from the error analysis.

It would be instructive to compare the current analyses with the recent work [27]. We note that the generalization bounds on the PDE residual proved in [27] apply to a general class of second-order linear PDEs (see Assumption 3.1 of [27]). In order to attain an L^2 error bound on the PINN solution, the following assumption (Assumption 3.2 of [27], page A3167) has been made on the PDE $\mathcal{L}u = f$ (with Dirichlet boundary condition) in [27],

$$C_1 \|u\|_{L^2(\Omega)} \leqslant \|\mathcal{L}u\|_{L^2(\Omega)} + \|u\|_{L^2(\partial\Omega)}\,, \tag{1}$$

where Ω is the domain (with boundary $\partial\Omega$), u is the PDE solution, $C_1 > 0$ is a constant independent of u, $\mathcal{L}$ is the linear differential operator in the PDE, and f denotes a prescribed function. By using this assumption and the generalization bounds on the PDE residual, an L^2 approximation error on the PINN solution is given in [27] (Theorem 3.6 therein). We note that similar assumptions (with somewhat different forms) have appeared in several other previous works (see e.g. [40, 39, 50]), and for a few PDEs the proofs for a relation analogous to (1) are available. It is not clear whether the assumption (1) is generally applicable, especially to dynamic PDEs of second-order in time, which are the focus of the current work. At the early stage of this project, we have attempted to develop an L^2 error estimate on the PINN solution using the original form (see Equation (2a) below) of the class of PDEs considered in this paper, which would presumably lead to a form analogous to (1), noting that $\mathcal{L}$ in (1) is the very differential operator appearing in the PDE. However, our attempt was not successful for certain PDEs of second-order in time. This unsuccessful attempt has led us to the reformulation of the original dynamic PDE of second-order in time into a system of two PDEs of first-order in time, as employed in all the current analyses. The L^2 error bounds we have obtained for the PINN solution have a generally different form than (1), especially in two aspects. First, our error bound involves differential operators that are different from the one (or those) appearing in the original PDE. Second, our error bound typically involves the square root of certain boundary norms, as mentioned previously. It should be emphasized that the bounds on the PINN solution error in terms of the PDE residual error for the class of PDE problems in the current paper are proven, not assumed.

The rest of this paper is organized as follows. Section 2 is an overview of PINN. In Sections 3, 4 and 5, we present an error analysis of the PINN algorithm for approximating the wave equation, nonlinear Klein-Gordon equation, and the linear elastodynamic equation. Section 6 summarizes a set of numerical experiments with these three PDEs to supplement and support our theoretical analyses. Section 7 concludes the presentation with some closing remarks. Finally, the appendix (Section 8) recalls some auxiliary results and provides the proofs of the theorems from Sections 4 and 5.

2. Physics Informed Neural Networks (PINN) for Approximating PDEs

2.1. Generic PDE of Second Order in Time

Consider a compact domain $D \subset \mathbb{R}^d$ ($d > 0$ being an integer), and let $\mathcal{D}$ and $\mathcal{B}$ denote the differential and boundary operators. We consider the following general form of an initial boundary value problem with

a generic PDE of second order in time. For any $\boldsymbol{x} \in D$, $\boldsymbol{y} \in \partial D$ and $t \in [0, T]$,

$$\frac{\partial^2 u}{\partial t^2}(\boldsymbol{x}, t) + \mathcal{D}[u](\boldsymbol{x}, t) = 0, \tag{2a}$$

$$\mathcal{B}u(\boldsymbol{y}, t) = u_d(\boldsymbol{y}, t), \tag{2b}$$

$$u(\boldsymbol{x}, 0) = u_{in}(\boldsymbol{x}), \quad \frac{\partial u}{\partial t}(\boldsymbol{x}, 0) = v_{in}(\boldsymbol{x}). \tag{2c}$$

Here, $u(\boldsymbol{x}, t)$ is the unknown field solution, u_d denotes the boundary data, and u_{in} and v_{in} are the initial distributions for u and $\frac{\partial u}{\partial t}$. We assume that in $\mathcal{D}$ the highest derivative with respect to the time variable t, if any, is of first order.

2.2. Neural Network Representation of a Function

Let $\sigma : \mathbb{R} \to \mathbb{R}$ denote an activation function that is at least twice continuously differentiable. For any $n \in \mathbb{N}$ and $z \in \mathbb{R}^n$, we define $\sigma(z) := (\sigma(z_1), \cdots, \sigma(z_n))$, where z_i $(1 \le i \le n)$ are the components of z. We adopt the following formal definition for a feedforward neural network as given in [12].

Definition 2.1 ([12])**.** *Let $R \in (0, \infty]$, $L, W \in \mathbb{N}$ and $l_0, \cdots, l_L \in \mathbb{N}$. Let $\sigma : \mathbb{R} \to \mathbb{R}$ be a twice differentiable function and define*

$$\Theta = \Theta_{L,W,R} := \bigcup_{L' \in \mathbb{N}, L' \le L} \bigcup_{l_0, \cdots, l_L \in \{1, \cdots, W\}} \times_{k=1}^{L'} ([-R, R]^{l_k \times l_{k-1}} \times [-R, R]^{l_k}). \tag{3}$$

For $\theta \in \Theta$, we define $\theta_k := (W_k, b_k)$ and $\mathcal{A}_k^\theta : \mathbb{R}^{l_{k-1}} \to \mathbb{R}^{l_k}$ by $z \mapsto W_k z + b_k$ for $1 \le k \le L$, and we define $f_k^\theta : \mathbb{R}^{l_{k-1}} \to \mathbb{R}^{l_k}$ by $f_k^\theta = \begin{cases} \mathcal{A}_L^\theta(z) & k = L, \\ (\sigma \circ \mathcal{A}_k^\theta)(z) & 1 \le k < L. \end{cases}$ Denote $u_\theta : \mathbb{R}^{l_0} \to \mathbb{R}^{l_L}$ the function that satisfies for all $z \in \mathbb{R}^{l_0}$ that

$$u_\theta(z) = (f_L^\theta \circ f_{L-1}^\theta \circ \cdots \circ f_1^\theta)(z) \qquad z \in \mathbb{R}^{l_0}. \tag{4}$$

We set $z = (\boldsymbol{x}, t)$ and $l_0 = d + 1$ for approximating the PDE problem (2).

u_θ as defined above is the neural-network representation of a parameterized function associated with the parameter θ. This neural network contains $(L + 1)$ layers $(L \ge 2)$, with widths $(l_0, l_1, \cdots, l_L)$ for each layer. The input layer has a width l_0, and the output layer has a width l_L. The $(L - 1)$ layers between the input/output layers are the hidden layers, with widths l_k $(1 \le k \le L - 1)$. W_k and b_k are the weight/bias coefficients corresponding to layer k for $1 \le k \le L$. From layer to layer the network logic represents an affine transform, followed by a function composition with the activation function σ. Note that no activation function is applied to the output layer. We refer to u_θ with $L = 2$ (i.e. single hidden layer) as a shallow neural network, and u_θ with $L \ge 3$ (i.e. multiple hidden layers) as a deeper or deep neural network.

2.3. Physics Informed Neural Network for Initial/Boundary Value Problem

Let $\Omega = D \times [0, T]$ and $\Omega_* = \partial D \times [0, T]$ be the spatial-temporal domain. We approximate the solution u to the problem (2) by a neural network $u_\theta : \Omega \to \mathbb{R}^n$. With PINN we consider the residual function of the initial/boundary value problem (2), defined for any sufficiently smooth function $u : \Omega \to \mathbb{R}^n$ as, for any $\boldsymbol{x} \in D$, $\boldsymbol{y} \in \partial D$ and $t \in [0, T]$,

$$\mathcal{R}_{int}[u](\boldsymbol{x}, t) = \frac{\partial^2 u}{\partial t^2}(\boldsymbol{x}, t) + \mathcal{D}[u](\boldsymbol{x}, t), \quad \mathcal{R}_{sb}[u](\boldsymbol{y}, t) = \mathcal{B}u(\boldsymbol{y}, t) - u_d(\boldsymbol{y}, t), \tag{5a}$$

$$\mathcal{R}_{tb1}[u](\boldsymbol{x}, 0) = u(\boldsymbol{x}, 0) - u_{in}(\boldsymbol{x}), \quad \mathcal{R}_{tb2}[u](\boldsymbol{x}, 0) = \frac{\partial u}{\partial t}(\boldsymbol{x}, 0) - v_{in}(\boldsymbol{x}). \tag{5b}$$

These residuals characterize how well a given function u satisfies the initial/boundary value problem (2). If u is the exact solution, $\mathcal{R}_{int}[u] = \mathcal{R}_{sb}[u] = \mathcal{R}_{tb1}[u] = \mathcal{R}_{tb2}[u] = 0$.

To facilitate the subsequent analyses, we introduce an auxiliary function $v = \frac{\partial u}{\partial t}$ and rewrite $\mathcal{R}_{tb2}$ as

$$\mathcal{R}_{tb2}[v](\boldsymbol{x}, 0) = v(\boldsymbol{x}, 0) - v_{in}(\boldsymbol{x}). \tag{6}$$

We reformulate (2a) into two equations, thus separating the interior residual into the following two components:

$$\mathcal{R}_{int1}[u,v](\boldsymbol{x},t) = \frac{\partial u}{\partial t}(\boldsymbol{x},t) - v(\boldsymbol{x},t), \quad \mathcal{R}_{int2}[u,v](\boldsymbol{x},t) = \frac{\partial v}{\partial t}(\boldsymbol{x},t) + \mathcal{D}[u](\boldsymbol{x},t). \tag{7}$$

With PINN, we seek a neural network (u_θ, v_θ) to minimize the following quantity,

$$\begin{aligned}\mathcal{E}_G(\theta)^2 = &\int_\Omega |R_{int1}[u_\theta,v_\theta](\boldsymbol{x},t)|^2 \,\mathrm{d}\boldsymbol{x} + \int_\Omega |R_{int2}[u_\theta,v_\theta](\boldsymbol{x},t)|^2 \,\mathrm{d}\boldsymbol{x} + \int_D |R_{tb1}[u_\theta](\boldsymbol{x})|^2 \,\mathrm{d}\boldsymbol{x} \\ &+ \int_D |R_{tb2}[v_\theta](\boldsymbol{x})|^2 \,\mathrm{d}\boldsymbol{x} + \int_{\Omega_*} |R_{sb}[u_\theta](\boldsymbol{x},t)|^2 \,\mathrm{d}s(\boldsymbol{x})\,\mathrm{d}t.\end{aligned} \tag{8}$$

The different terms of (8) may be rescaled by different weights (penalty coefficients). For simplicity, we set all these weights to one in the analysis. $\mathcal{E}_G$ as defined above is often referred to as the generalization error. Because of the integrals involved therein, $\mathcal{E}_G$ can be hard to minimize. In practice, one will approximate (8) by an appropriate numerical quadrature rule, as follows

$$\mathcal{E}_T(\theta,\mathcal{S})^2 = \mathcal{E}_T^{int1}(\theta,\mathcal{S}_{int})^2 + \mathcal{E}_T^{int2}(\theta,\mathcal{S}_{int})^2 + \mathcal{E}_T^{tb1}(\theta,\mathcal{S}_{tb})^2 + \mathcal{E}_T^{tb2}(\theta,\mathcal{S}_{tb})^2 + \mathcal{E}_T^{sb}(\theta,\mathcal{S}_{sb})^2, \tag{9}$$

where

$$\mathcal{E}_T^{int1}(\theta,\mathcal{S}_{int})^2 = \sum_{n=1}^{N_{int}} \omega_{int}^n |R_{int1}[u_\theta,v_\theta](\boldsymbol{x}_{int}^n, t_{int}^n)|^2, \tag{10a}$$

$$\mathcal{E}_T^{int2}(\theta,\mathcal{S}_{int})^2 = \sum_{n=1}^{N_{int}} \omega_{int}^n |R_{int2}[u_\theta,v_\theta](\boldsymbol{x}_{int}^n, t_{int}^n)|^2, \quad \mathcal{E}_T^{tb1}(\theta,\mathcal{S}_{tb})^2 = \sum_{n=1}^{N_{tb}} \omega_{tb}^n |R_{tb1}[u_\theta](\boldsymbol{x}_{tb}^n)|^2, \tag{10b}$$

$$\mathcal{E}_T^{tb2}(\theta,\mathcal{S}_{tb})^2 = \sum_{n=1}^{N_{tb}} \omega_{tb}^n |R_{tb2}[v_\theta](\boldsymbol{x}_{tb}^n)|^2, \quad \mathcal{E}_T^{sb}(\theta,\mathcal{S}_{sb})^2 = \sum_{n=1}^{N_{sb}} \omega_{sb}^n |R_{sb}[u_\theta](\boldsymbol{x}_{sb}^n, t_{sb}^n)|^2. \tag{10c}$$

The quadrature points in the spatial-temporal domain and on the spatial and temporal boundaries, $\mathcal{S}_{int} = \{(\boldsymbol{x}_{int}^n, t_{int}^n)\}_{n=1}^{N_{int}}$, $\mathcal{S}_{sb} = \{(\boldsymbol{x}_{sb}^n, t_{sb}^n)\}_{n=1}^{N_{sb}}$ and $\mathcal{S}_{tb} = \{(\boldsymbol{x}_{tb}^n, t_{tb}^n = 0)\}_{n=1}^{N_{tb}}$, constitute the input data sets to the neural network. In the above equations $\mathcal{E}_T(\theta,\mathcal{S})^2$ is referred to as the training error (or training loss), and $\omega_\star^n$ are suitable quadrature weights for $\star = int$, sb and tb. Therefore, PINN attempts to minimize the training error $\mathcal{E}_T(\theta,\mathcal{S})^2$ over the network parameters θ, and upon convergence of optimization the trained u_θ contains the approximation of the solution u to the problem (2).

Remark 2.2. *The generalization error* (8) *(with the corresponding training error* (9)*) is the standard (canonical) PINN form if one introduces* $v = \frac{\partial u}{\partial t}$ *and reformulates* (2a) *into two equations. We would like to emphasize that our analyses below suggest alternative forms for the generalization error, e.g.*

$$\begin{aligned}\mathcal{E}_G(\theta)^2 = &\int_\Omega |R_{int1}[u_\theta,v_\theta](\boldsymbol{x},t)|^2 \,\mathrm{d}\boldsymbol{x} + \int_\Omega |R_{int2}[u_\theta,v_\theta](\boldsymbol{x},t)|^2 \,\mathrm{d}\boldsymbol{x} + \int_\Omega |\nabla R_{int1}[u_\theta,v_\theta](\boldsymbol{x},t)|^2 \,\mathrm{d}\boldsymbol{x} \\ &+ \int_D |R_{tb1}[u_\theta](\boldsymbol{x})|^2 \,\mathrm{d}\boldsymbol{x} + \int_D |R_{tb2}[v_\theta](\boldsymbol{x})|^2 \,\mathrm{d}\boldsymbol{x} + \int_D |\nabla R_{tb1}[u_\theta](\boldsymbol{x})|^2 \,\mathrm{d}\boldsymbol{x} \\ &+ \left(\int_{\Omega_*} |R_{sb}[u_\theta](\boldsymbol{x},t)|^2 \,\mathrm{d}s(\boldsymbol{x})\,\mathrm{d}t\right)^{\frac{1}{2}},\end{aligned} \tag{11}$$

which differs from (8) *in the terms* ∇R_{int1}, ∇R_{tb1} *and the last term. The corresponding training error is,*

$$\begin{aligned}\mathcal{E}_T(\theta,\mathcal{S})^2 = &\mathcal{E}_T^{int1}(\theta,\mathcal{S}_{int})^2 + \mathcal{E}_T^{int2}(\theta,\mathcal{S}_{int})^2 + \mathcal{E}_T^{int3}(\theta,\mathcal{S}_{int})^2 + \mathcal{E}_T^{tb1}(\theta,\mathcal{S}_{tb})^2 \\ &+ \mathcal{E}_T^{tb2}(\theta,\mathcal{S}_{tb})^2 + \mathcal{E}_T^{tb3}(\theta,\mathcal{S}_{tb})^2 + \mathcal{E}_T^{sb}(\theta,\mathcal{S}_{sb}),\end{aligned} \tag{12}$$

where

$$\mathcal{E}_T^{int3}(\theta,\mathcal{S}_{int})^2 = \sum_{n=1}^{N_{int}} \omega_{int}^n |\nabla R_{int1}[u_\theta,v_\theta](\boldsymbol{x}_{int}^n, t_{int}^n)|^2, \quad \mathcal{E}_T^{tb3}(\theta,\mathcal{S}_{tb})^2 = \sum_{n=1}^{N_{tb}} \omega_{tb}^n |\nabla R_{tb1}[u_\theta](\boldsymbol{x}_{tb}^n)|^2. \tag{13}$$

The error analyses also suggest additional terms in the generalization error for different equations.

2.4. Numerical Quadrature Rules

As discussed above, we need to approximate the integrals of functions. The analysis in the subsequent sections requires well-known results on numerical quadrature rules as reviewed below.

Given $\Lambda \subset \mathbb{R}^d$ and a function $f \in L^1(\Lambda)$, we would like to approximate $\int_\Lambda f(z)\,\mathrm{d}z$. A quadrature rule provides an approximation by $\int_\Lambda f(z)\,\mathrm{d}z \approx \frac{1}{M}\sum_{n=1}^{M} \omega_n f(z_n)$, where $z_n \in \Lambda$ $(1 \le n \le M)$ are the quadrature points and ω_n $(1 \le n \le M)$ denote the appropriate quadrature weights. The approximation accuracy is influenced by the type of quadrature rule, the number of quadrature points (M), and the regularity of f. For the mid-point rule, which is assumed in the current work, Λ is partitioned into $M \sim N^d$ cubes with an edge length $\frac{1}{N}$ and the approximation accuracy is given by

$$\left| \int_\Lambda f(z)\,\mathrm{d}z - \mathcal{Q}_M^\Lambda[f] \right| \le C_f M^{-2/d}, \tag{14}$$

where $\mathcal{Q}_M^\Lambda[f] := \frac{1}{M}\sum_{n=1}^{M} \omega_n f(z_n)$, $C_f \lesssim \|f\|_{C^2(\Lambda)}$ ($a \lesssim b$ denotes $a \le Cb$) and $\{z_n\}_{n=1}^M$ denote the midpoints of these cubes [11]. In this paper, we use C to denote a universal constant, which may depend on k, d, T, u and v but not on N. And we use the subscript to emphasize its dependence when necessary, e.g. C_d is a constant depending only on d.

We focus on PDE problems in relatively low dimensions $(d \le 3)$ in this paper and employ the standard quadrature rules. We note that in higher dimensions the standard quadrature rules may not be favorable. In this case the random training points or low-discrepancy training points [41] may be preferred.

In subsequent sections we focus on three representative dynamic equations of second order in time (the wave equation, the nonlinear Klein-Gordon equation, and the linear elastodynamic equation), and provide the error estimate for approximating these equations by PINN. We note that these analyses suggest alternative forms for the training loss function that are somewhat different from the standard PINN forms [46]. The PINN numerical results based on the standard form for the loss function, and based on the alternative forms as suggested by the error estimate, will be provided after the presentation of the theoretical analysis. In what follows, for brevity we adopt the notation of $\mathcal{F}_\Xi = \frac{\partial \mathcal{F}}{\partial \Xi}$, $\mathcal{F}_{\Xi\Upsilon} = \frac{\partial^2 \mathcal{F}}{\partial \Xi \partial \Upsilon}$ $(\Xi, \Upsilon \in \{t, x\})$, for any sufficiently smooth function $\mathcal{F} : \Omega \to \mathbb{R}^n$.

3. Physics Informed Neural Networks for Approximating the Wave Equation

3.1. Wave Equation

Consider the wave equations on the torus $D = [0,1)^d \subset \mathbb{R}^d$ with periodic boundary conditions:

$$u_t - v = 0, \quad v_t - \Delta u = f, \qquad \text{in } D \times [0,T], \tag{15a}$$

$$u(\boldsymbol{x}, 0) = \psi_1(\boldsymbol{x}), \quad v(\boldsymbol{x}, 0) = \psi_2(\boldsymbol{x}), \qquad \text{in } D, \tag{15b}$$

$$u(\boldsymbol{x}, t) = u(\boldsymbol{x} + 1, t), \quad \nabla u(\boldsymbol{x}, t) = \nabla u(\boldsymbol{x} + 1, t), \quad \text{in } \partial D \times [0,T]. \tag{15c}$$

The regularity results for linear evolution equations of the second order in time have been studied in [54]. When the self-adjoint operator $\mathcal{A}$ takes Δ, the linear evolution equations of second order in time from [54] become the classical wave equations, and then we can obtain the following regularity results.

Lemma 3.1. *Let $r \ge 1$, $\psi_1 \in H^r(D)$, $\psi_2 \in H^{r-1}(D)$ and $f \in L^2([0,T]; H^{r-1}(D))$. Then there exists a unique solution u to the classical wave equations such that $u \in C([0,T]; H^r(D))$ and $u_t \in C([0,T]; H^{r-1}(D))$.*

Lemma 3.2. *Let $k \in \mathbb{N}$, $\psi_1 \in H^r(D)$, $\psi_2 \in H^{r-1}(D)$ and $f \in C^{k-1}([0,T]; H^{r-k}(D)) \cap L^2([0,T]; H^{r-1}(D))$ with $r > \frac{d}{2} + k$. Then there exists $T > 0$ and a classical solution u to the wave equations such that $u(\boldsymbol{x}, 0) = \psi_1$, $u_t(\boldsymbol{x}, 0) = \psi_2$, $u \in H^k(D \times [0,T])$ and $v \in H^{k-1}(D \times [0,T])$.*

Proof. By Lemma 3.1, there exists $T > 0$ and the solution (u, v) to the wave equations such that $u(\boldsymbol{x}, 0) = \psi_1$, $v(\boldsymbol{x}, 0) = \psi_2$, $u \in C([0,T]; H^r(D))$ and $v \in C([0,T]; H^{r-1}(D))$. As $r > \frac{d}{2} + k$, $H^{r-k}(D)$ is a Banach algebra.

For $k = 1$, since $u \in C([0,T]; H^r(D))$, $v \in C([0,T]; H^{r-1}(D))$ and $f \in C([0,T]; H^{r-1}(D))$, we have $u_t = v \in C([0,T]; H^{r-1}(D))$ and $v_t = \Delta u + f \in C([0,T]; H^{r-2}(D))$. Then, it implies that $u \in C^1([0,T]; H^{r-1}(D))$ and $v \in C^1([0,T]; H^{r-2}(D))$.

For $k = 2$, by $f \in C^1([0,T]; H^{r-2}(D))$, we have $u_{tt} = v_t \in C([0,T]; H^{r-2}(D))$ and $v_{tt} = \Delta u_t + f_t \in C([0,T]; H^{r-3}(D))$. Then, it implies that $u \in C^2([0,T]; H^{r-2}(D))$ and $v \in C^2([0,T]; H^{r-3}(D))$.

Repeating the same argument, we have $u \in \cap_{l=0}^{k} C^l([0,T]; H^{r-l}(D))$ and $v \in \cap_{l=0}^{k} C^l([0,T]; H^{r-l-1}(D))$. For all $0 \le l \le k$, using $l + r - l \ge k$ with $r > \frac{d}{2} + k$, it holds $u \in H^k(D \times [0,T])$ and $v \in H^{k-1}(D \times [0,T])$. □

3.2. Physics Informed Neural Networks

We would like to approximate the solutions to the problem (15) with PINN. We seek deep neural networks $u_\theta : D \times [0,T] \to \mathbb{R}$ and $v_\theta : D \times [0,T] \to \mathbb{R}$, parameterized by $\theta \in \Theta$, that approximate the solution u and v of (15). Define residuals,

$$R_{int1}[u_\theta, v_\theta](\boldsymbol{x}, t) = u_{\theta t} - v_\theta, \quad R_{int2}[u_\theta, v_\theta](\boldsymbol{x}, t) = v_{\theta t} - \Delta u_\theta - f, \tag{16a}$$

$$R_{tb1}[u_\theta](\boldsymbol{x}) = u_\theta(\boldsymbol{x}, 0) - \psi_1(\boldsymbol{x}), \quad R_{tb2}[v_\theta](\boldsymbol{x}) = v_\theta(\boldsymbol{x}, 0) - \psi_2(\boldsymbol{x}), \tag{16b}$$

$$R_{sb1}[v_\theta](\boldsymbol{x}, t) = v_\theta(\boldsymbol{x}, t) - v_\theta(\boldsymbol{x} + 1, t), \quad R_{sb2}[u_\theta](\boldsymbol{x}, t) = \nabla u_\theta(\boldsymbol{x}, t) - \nabla u_\theta(\boldsymbol{x} + 1, t). \tag{16c}$$

Note that for the exact solution $R_{int1}[u, v] = R_{int2}[u, v] = R_{tb1}[u] = R_{tb2}[v] = R_{sb1}[v] = R_{sb2}[u] = 0$. Let $\Omega = D \times [0,T]$ and $\Omega_* = \partial D \times [0,T]$ be the space-time domain. With PINN, we minimize the the following generalization error,

$$\begin{aligned}
\mathcal{E}_G(\theta)^2 = &\int_\Omega |R_{int1}[u_\theta, v_\theta](\boldsymbol{x}, t)|^2 \,\mathrm{d}\boldsymbol{x} + \int_\Omega |R_{int2}[u_\theta, v_\theta](\boldsymbol{x}, t)|^2 \,\mathrm{d}\boldsymbol{x} + \int_\Omega |\nabla R_{int1}[u_\theta, v_\theta](\boldsymbol{x}, t)|^2 \,\mathrm{d}\boldsymbol{x} \\
&+ \int_D |R_{tb1}[u_\theta](\boldsymbol{x})|^2 \,\mathrm{d}\boldsymbol{x} + \int_D |R_{tb2}[v_\theta](\boldsymbol{x})|^2 \,\mathrm{d}\boldsymbol{x} + \int_D |\nabla R_{tb1}[u_\theta](\boldsymbol{x})|^2 \,\mathrm{d}\boldsymbol{x} \\
&+ \left(\int_{\Omega_*} |R_{sb1}[v_\theta](\boldsymbol{x}, t)|^2 \,\mathrm{d}s(\boldsymbol{x}) \,\mathrm{d}t \right)^{1/2} + \left(\int_{\Omega_*} |R_{sb2}[u_\theta](\boldsymbol{x}, t)|^2 \,\mathrm{d}s(\boldsymbol{x}) \,\mathrm{d}t \right)^{1/2}.
\end{aligned} \tag{17}$$

The form of different terms in this expression will become clearer below.

To complete the PINN formulation, we will choose the training set $\mathcal{S} \subset \overline{D} \times [0,T]$ based on suitable quadrature points. We divide the full training set $\mathcal{S} = \mathcal{S}_{int} \cup \mathcal{S}_{sb} \cup \mathcal{S}_{tb}$ into the following three components:

- Interior training points $\mathcal{S}_{int} = \{z_n\}$ for $1 \le n \le N_{int}$, with each $z_n = (\boldsymbol{x}, t)_n \in D \times (0,T)$.
- Spatial boundary training points $\mathcal{S}_{sb} = \{z_n\}$ for $1 \le n \le N_{sb}$, with each $z_n = (\boldsymbol{x}, t)_n \in \partial D \times (0,T)$.
- Temporal boundary training points $\mathcal{S}_{tb} = \{\boldsymbol{x}_n\}$ for $1 \le n \le N_{tb}$ with each $\boldsymbol{x}_n \in D$.

We define the PINN training loss, $\theta \mapsto \mathcal{E}_T(\theta, \mathcal{S})^2$, as follows,

$$\begin{aligned}
\mathcal{E}_T(\theta, \mathcal{S})^2 = &\mathcal{E}_T^{int1}(\theta, \mathcal{S}_{int})^2 + \mathcal{E}_T^{int2}(\theta, \mathcal{S}_{int})^2 + \mathcal{E}_T^{int3}(\theta, \mathcal{S}_{int})^2 + \mathcal{E}_T^{tb1}(\theta, \mathcal{S}_{tb})^2 + \mathcal{E}_T^{tb2}(\theta, \mathcal{S}_{tb})^2 \\
&+ \mathcal{E}_T^{tb3}(\theta, \mathcal{S}_{tb})^2 + \mathcal{E}_T^{sb1}(\theta, \mathcal{S}_{sb}) + \mathcal{E}_T^{sb2}(\theta, \mathcal{S}_{sb}),
\end{aligned} \tag{18}$$

where

$$\mathcal{E}_T^{int1}(\theta, \mathcal{S}_{int})^2 = \sum_{n=1}^{N_{int}} \omega_{int}^n |R_{int1}[u_\theta, v_\theta](\boldsymbol{x}_{int}^n, t_{int}^n)|^2, \quad \mathcal{E}_T^{tb1}(\theta, \mathcal{S}_{tb})^2 = \sum_{n=1}^{N_{tb}} \omega_{tb}^n |R_{tb1}[u_\theta](\boldsymbol{x}_{tb}^n)|^2, \tag{19a}$$

$$\mathcal{E}_T^{int2}(\theta, \mathcal{S}_{int})^2 = \sum_{n=1}^{N_{int}} \omega_{int}^n |R_{int2}[u_\theta, v_\theta]](\boldsymbol{x}_{int}^n, t_{int}^n)|^2, \quad \mathcal{E}_T^{tb2}(\theta, \mathcal{S}_{tb})^2 = \sum_{n=1}^{N_{tb}} \omega_{tb}^n |R_{tb2}[v_\theta](\boldsymbol{x}_{tb}^n)|^2, \tag{19b}$$

$$\mathcal{E}_T^{int3}(\theta, \mathcal{S}_{int})^2 = \sum_{n=1}^{N_{int}} \omega_{int}^n |\nabla R_{int1}[u_\theta, v_\theta](\boldsymbol{x}_{int}^n, t_{int}^n)|^2, \quad \mathcal{E}_T^{tb3}(\theta, \mathcal{S}_{tb})^2 = \sum_{n=1}^{N_{tb}} \omega_{tb}^n |\nabla R_{tb1}[u_\theta](\boldsymbol{x}_{tb}^n)|^2, \tag{19c}$$

$$\mathcal{E}_T^{sb1}(\theta, \mathcal{S}_{sb})^2 = \sum_{n=1}^{N_{sb}} \omega_{sb}^n |R_{sb1}[v_\theta](\boldsymbol{x}_{sb}^n, t_{sb}^n)|^2, \quad \mathcal{E}_T^{sb2}(\theta, \mathcal{S}_{sb})^2 = \sum_{n=1}^{N_{sb}} \omega_{sb}^n |R_{sb2}[u_\theta](\boldsymbol{x}_{sb}^n, t_{sb}^n)|^2. \tag{19d}$$

Here the quadrature points in space-time constitute the data sets $\mathcal{S}_{int} = \{(\boldsymbol{x}_{int}^n, t_{int}^n)\}_{n=1}^{N_{int}}$, $\mathcal{S}_{tb} = \{\boldsymbol{x}_{tb}^n)\}_{n=1}^{N_{tb}}$ and $\mathcal{S}_{sb} = \{(\boldsymbol{x}_{sb}^n, t_{sb}^n)\}_{n=1}^{N_{sb}}$, and $\omega_\star^n$ are suitable quadrature weights with $\star$ denoting int, tb or sb.

Let $\hat{u} = u_\theta - u$, $\hat{v} = v_\theta - v$ denote the difference between the solution to the wave equations and the PINN approximation of the solution. We define the total error of the PINN approximation by

$$\mathcal{E}(\theta)^2 = \int_\Omega (|\hat{u}(\boldsymbol{x}, t)|^2 + |\nabla \hat{u}(\boldsymbol{x}, t)|^2 + |\hat{v}(\boldsymbol{x}, t)|^2) \,\mathrm{d}\boldsymbol{x} \,\mathrm{d}t. \tag{20}$$

3.3. Error Analysis

In light of the wave equations (15) and the definitions for different residuals (16), we have

$$R_{int1} = \hat{u}_t - \hat{v}, \tag{21a}$$
$$R_{int2} = \hat{v}_t - \Delta\hat{u}, \tag{21b}$$
$$R_{tb1} = \hat{u}(\boldsymbol{x}, 0), \quad R_{tb2} = \hat{v}(\boldsymbol{x}, 0), \quad R_{sb1} = \hat{v}(\boldsymbol{x}, t) - \hat{v}(\boldsymbol{x}+1, t), \quad R_{sb2} = \nabla\hat{u}(\boldsymbol{x}, t) - \nabla\hat{u}(\boldsymbol{x}+1, t). \tag{21c}$$

3.3.1. Bound on the Residuals

Theorem 3.3. *Let $n \geq 2$, d, r, $k \in \mathbb{N}$ with $k \geq 3$. Suppose that $\psi_1 \in H^r(D)$, $\psi_2 \in H^{r-1}(D)$ and $f \in C^{k-1}([0,T]; H^{r-k}(D)) \cap L^2([0,T]; H^{r-1}(D))$ with $r > \frac{d}{2} + k$. For every integer $N > 5$, there exist* tanh *neural networks u_θ and v_θ, each with two hidden layers, of widths at most $3\lceil\frac{k+n-2}{2}\rceil|P_{k-1,d+2}| + \lceil NT\rceil + d(N-1)$ and $3\lceil\frac{d+n+1}{2}\rceil|P_{d+2,d+2}|\lceil NT\rceil N^d$, such that*

$$\|R_{int1}\|_{L^2(\Omega)}, \|R_{tb1}\|_{L^2(D)} \lesssim \ln N N^{-k+1}, \tag{22a}$$
$$\|R_{int2}\|_{L^2(\Omega)}, \|\nabla R_{int1}\|_{L^2(\Omega)}, \|\nabla R_{tb1}\|_{L^2(D)}, \|R_{sb2}\|_{L^2(\partial D\times[0,T])} \lesssim \ln^2 N N^{-k+2}, \tag{22b}$$
$$\|R_{tb2}\|_{L^2(D)}, \|R_{sb1}\|_{L^2(\partial D\times[0,T])} \lesssim \ln N N^{-k+2}. \tag{22c}$$

Proof. Based on Lemma 3.2, it holds that $u \in H^k(\Omega)$ and $v \in H^{k-1}(\Omega)$. In light of Lemma 8.3, there exists neural networks u_θ and v_θ, with the same two hidden layers and widths $3\lceil\frac{k+n-2}{2}\rceil|P_{k-1,d+2}|+\lceil NT\rceil+d(N-1)$ and $3\lceil\frac{d+n+1}{2}\rceil|P_{d+2,d+2}|\lceil NT\rceil N^d$, such that for every $0 \leq l \leq 2$ and $0 \leq s \leq 2$,

$$\|u_\theta - u\|_{H^l(\Omega)} \leq C_{l,k,d+1,u}\lambda_{l,u}(N)N^{-k+l}, \quad \|v_\theta - v\|_{H^s(\Omega)} \leq C_{s,k-1,d+1,v}\lambda_{s,v}(N)N^{-k+1+s}, \tag{23}$$

where $\lambda_{l,u} = 2^l 3^{d+1}(1+\delta)\ln^l\left(\beta_{l,\delta,d+1,u}N^{d+k+3}\right)$, $\delta = \frac{1}{100}$, $\lambda_{s,v} = 2^s 3^{d+1}(1+\delta)\ln^s\left(\beta_{s,\delta,d+1,v}N^{d+k+2}\right)$, and the definition for the other constants can be found in Lemma 8.3.

In light of Lemma 8.1, we can bound the PINN residual terms,

$$\|\hat{u}_t\|_{L^2(\Omega)} \leq \|\hat{u}\|_{H^1(\Omega)}, \qquad \|\hat{v}_t\|_{L^2(\Omega)} \leq \|\hat{v}\|_{H^1(\Omega)},$$
$$\|\Delta\hat{u}\|_{L^2(\Omega)} \leq \|\hat{u}\|_{H^2(\Omega)}, \qquad \|\nabla\hat{u}_t\|_{L^2(\Omega)} \leq \|\hat{u}\|_{H^2(\Omega)}, \qquad \|\nabla\hat{v}\|_{L^2(\Omega)} \leq \|\hat{v}\|_{H^1(\Omega)},$$
$$\|\hat{u}\|_{L^2(D)} \leq \|\hat{u}\|_{L^2(\partial\Omega)} \leq C_{h_\Omega,d+1,\rho_\Omega}\|\hat{u}\|_{H^1(\Omega)}, \qquad \|\hat{v}\|_{L^2(D)} \leq \|\hat{v}\|_{L^2(\partial\Omega)} \leq C_{h_\Omega,d+1,\rho_\Omega}\|\hat{v}\|_{H^1(\Omega)},$$
$$\|\nabla\hat{u}\|_{L^2(D)} \leq \|\nabla\hat{u}\|_{L^2(\partial\Omega)} \leq C_{h_\Omega,d+1,\rho_\Omega}\|\hat{u}\|_{H^2(\Omega)}, \quad \|\hat{v}\|_{L^2(\partial D\times[0,T])} \leq \|\hat{v}\|_{L^2(\partial\Omega)} \leq C_{h_\Omega,d+1,\rho_\Omega}\|\hat{v}\|_{H^1(\Omega)},$$
$$\|\nabla\hat{u}\|_{L^2(\partial D\times[0,T])} \leq \|\nabla\hat{u}\|_{L^2(\partial\Omega)} \leq C_{h_\Omega,d+1,\rho_\Omega}\|\hat{u}\|_{H^2(\Omega)}.$$

By combining these relations with (23), we can obtain

$$\begin{aligned}
\|R_{int1}\|_{L^2(\Omega)} &= \|\hat{u}_t - \hat{v}\|_{L^2(\Omega)} \leq \|\hat{u}\|_{H^1(\Omega)} + \|\hat{v}\|_{L^2(\Omega)} \\
&\leq C_{1,k,d+1,u}\lambda_{1,u}(N)N^{-k+1} + C_{0,k-1,d+1,v}\lambda_{0,v}(N)N^{-k+1} \lesssim \ln N N^{-k+1}, \\
\|R_{int2}\|_{L^2(\Omega)} &= \|\hat{v}_t - \Delta\hat{u}\|_{L^2(\Omega)} \leq \|\hat{v}\|_{H^1(\Omega)} + \|\hat{u}\|_{H^2(\Omega)} \\
&\leq C_{2,k,d+1,u}\lambda_{2,u}(N)N^{-k+2} + C_{1,k-1,d+1,v}\lambda_{1,v}(N)N^{-k+2} \lesssim \ln^2 N N^{-k+2}, \\
\|\nabla R_{int1}\|_{L^2(\Omega)} &= \|\nabla(\hat{u}_t - \hat{v})\|_{L^2(\Omega)} \leq \|\hat{u}\|_{H^2(\Omega)} + \|\hat{v}\|_{H^1(\Omega)} \\
&\leq C_{2,k,d+1,u}\lambda_{2,u}(N)N^{-k+2} + C_{1,k-1,d+1,v}\lambda_{1,v}(N)N^{-k+2} \lesssim \ln^2 N N^{-k+2}, \\
\|R_{tb1}\|_{L^2(D)} &\leq C_{h_\Omega,d+1,\rho_\Omega}\|\hat{u}\|_{H^1(\Omega)} \lesssim \ln N N^{-k+1}, \\
\|R_{tb2}\|_{L^2(D)}, \|R_{sb1}\|_{L^2(\partial D\times[0,T])} &\leq C_{h_\Omega,d+1,\rho_\Omega}\|\hat{v}\|_{H^1(\Omega)} \lesssim \ln N N^{-k+2}, \\
\|\nabla R_{tb1}\|_{L^2(D)}, \|R_{sb2}\|_{L^2(\partial D\times[0,T])} &\leq C_{h_\Omega,d+1,\rho_\Omega}\|\hat{u}\|_{H^2(\Omega)} \lesssim \ln^2 N N^{-k+2},
\end{aligned}$$

which completes the proof of Theorem 3.3. □

Theorem 3.3 implies that one can make the PINN residuals (16) arbitrarily small by choosing N to be sufficiently large. It follows that the generalization error $\mathcal{E}_G(\theta)^2$ in (17) can be made arbitrarily small.

3.3.2. Bounds on the Total Approximation Error

We next show that the total error $\mathcal{E}(\theta)^2$ is also small when the generalization error $\mathcal{E}_G(\theta)^2$ is small with the PINN approximation (u_θ, v_θ). Then we prove that the total error $\mathcal{E}(\theta)^2$ can be arbitrarily small, provided that the training error $\mathcal{E}_T(\theta, \mathcal{S})^2$ is sufficiently small and the sample set is sufficiently large.

Theorem 3.4. *Let $d \in \mathbb{N}$, $u \in C^1(\Omega)$ and $v \in C^0(\Omega)$ be the classical solution to the wave equations (15). Let u_θ and v_θ denote the PINN approximation with parameter θ. Then the following relation holds,*

$$\mathcal{E}(\theta)^2 = \int_\Omega (|\hat{u}(\boldsymbol{x},t)|^2 + |\nabla\hat{u}(\boldsymbol{x},\tau)|^2 + |\hat{v}(\boldsymbol{x},t)|^2)\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t \le C_G T\exp(2T), \tag{24}$$

where C_G is given by (27) in the following proof.

Proof. Taking the L^2 inner product of (21a) and (21b) with $\hat{u}$ and $\hat{v}$ over D, respectively, we have

$$\frac{d}{2dt}\int_D |\hat{u}|^2\,\mathrm{d}\boldsymbol{x} = \int_D \hat{u}\hat{v}\,\mathrm{d}\boldsymbol{x} + \int_D R_{int1}\hat{u}\,\mathrm{d}\boldsymbol{x} \le \int_D |\hat{u}|^2\,\mathrm{d}\boldsymbol{x} + \frac{1}{2}\int_D |R_{int1}|^2\,\mathrm{d}\boldsymbol{x} + \frac{1}{2}\int_D |\hat{v}|^2\,\mathrm{d}\boldsymbol{x}, \tag{25}$$

$$\begin{aligned}
\frac{d}{2dt}\int_D |\hat{v}|^2\,\mathrm{d}\boldsymbol{x} &= -\int_D \nabla\hat{u}\cdot\nabla\hat{v}\,\mathrm{d}\boldsymbol{x} + \int_{\partial D}\hat{v}\nabla\hat{u}\cdot\boldsymbol{n}\,\mathrm{d}s(\boldsymbol{x}) + \int_D R_{int2}\hat{v}\,\mathrm{d}\boldsymbol{x}\\
&= -\frac{d}{2dt}\int_D |\nabla\hat{u}|^2\,\mathrm{d}\boldsymbol{x} + \int_D \nabla\hat{u}\cdot\nabla R_{int1}\,\mathrm{d}\boldsymbol{x} + \int_{\partial D}\hat{v}\nabla\hat{u}\cdot\boldsymbol{n}\,\mathrm{d}s(\boldsymbol{x}) + \int_D R_{int2}\hat{v}\,\mathrm{d}\boldsymbol{x}\\
&\le -\frac{d}{2dt}\int_D |\nabla\hat{u}|^2\,\mathrm{d}\boldsymbol{x} + \frac{1}{2}\int_D |\nabla\hat{u}|^2\,\mathrm{d}\boldsymbol{x} + \frac{1}{2}\int_D |\nabla R_{int1}|^2\,\mathrm{d}\boldsymbol{x} + C_{\partial D_1}\left(\int_{\partial D}|R_{sb1}|^2\,\mathrm{d}s(\boldsymbol{x})\right)^{1/2}\\
&\quad + C_{\partial D_2}\left(\int_{\partial D}|R_{sb2}|^2\,\mathrm{d}s(\boldsymbol{x})\right)^{1/2} + \frac{1}{2}\int_D |\hat{v}|^2\,\mathrm{d}\boldsymbol{x} + \frac{1}{2}\int_D |R_{int2}|^2\,\mathrm{d}\boldsymbol{x}.
\end{aligned} \tag{26}$$

Here, we have used $\hat{v} = \hat{u}_t - R_{int1}$, $C_{\partial D_1} = |\partial D|^{\frac{1}{2}}(\|u\|_{C^1(\partial D\times[0,T])} + \|u_\theta\|_{C^1(\partial D\times[0,T])})$ and $C_{\partial D_2} = |\partial D|^{\frac{1}{2}}(\|v\|_{C(\partial D\times[0,T])} + \|v_\theta\|_{C(\partial D\times[0,T])})$.

Adding (25) to (26), integrating it over $[0,\tau]$ for any $\tau \le T$ and applying the Cauchy–Schwarz inequality, we obtain

$$\begin{aligned}
&\int_D |\hat{u}(\boldsymbol{x},\tau)|^2\,\mathrm{d}\boldsymbol{x} + \int_D |\nabla\hat{u}(\boldsymbol{x},\tau)|^2\,\mathrm{d}\boldsymbol{x} + \int_D |\hat{v}(\boldsymbol{x},\tau)|^2\,\mathrm{d}\boldsymbol{x}\\
&\quad\le \int_D |R_{tb1}|^2\,\mathrm{d}\boldsymbol{x} + \int_D |R_{tb2}|^2\,\mathrm{d}\boldsymbol{x} + \int_D |\nabla R_{tb1}|^2\,\mathrm{d}\boldsymbol{x} + 2\int_0^\tau\int_D \left(|\hat{u}|^2 + |\nabla\hat{u}|^2 + |\hat{v}|^2\right)\mathrm{d}\boldsymbol{x}\,\mathrm{d}t\\
&\quad + \int_\Omega \left(|R_{int1}|^2 + |R_{int2}|^2 + |\nabla R_{int1}|^2\right)\mathrm{d}\boldsymbol{x}\,\mathrm{d}t + 2C_{\partial D_1}|T|^{1/2}\left(\int_0^T\int_{\partial D}|R_{sb1}|^2\,\mathrm{d}s(\boldsymbol{x})\,\mathrm{d}t\right)^{1/2}\\
&\quad + 2C_{\partial D_2}|T|^{1/2}\left(\int_0^T\int_{\partial D}|R_{sb2}|^2\,\mathrm{d}s(\boldsymbol{x})\,\mathrm{d}t\right)^{1/2}.
\end{aligned}$$

We apply the integral form of the Grönwall inequality to the above inequality to get

$$\int_D \left(|\hat{u}(\boldsymbol{x},\tau)|^2 + |\nabla\hat{u}(\boldsymbol{x},\tau)|^2 + |\hat{v}(\boldsymbol{x},\tau)|^2\right)\mathrm{d}\boldsymbol{x} \le C_G\exp(2T),$$

where

$$\begin{aligned}
C_G &= \int_D (|R_{tb1}|^2 + |R_{tb2}|^2 + |\nabla R_{tb1}|^2)\,\mathrm{d}\boldsymbol{x} + \int_\Omega (|R_{int1}|^2 + |R_{int2}|^2 + |\nabla R_{int1}|^2)\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t\\
&\quad + 2C_{\partial D_1}|T|^{1/2}\left(\int_0^T\int_{\partial D}|R_{sb1}|^2\,\mathrm{d}s(\boldsymbol{x})\,\mathrm{d}t\right)^{1/2} + 2C_{\partial D_2}|T|^{1/2}\left(\int_0^T\int_{\partial D}|R_{sb2}|^2\,\mathrm{d}s(\boldsymbol{x})\,\mathrm{d}t\right)^{1/2}.
\end{aligned} \tag{27}$$

Then, we integrate the above inequality over $[0,T]$ to yield (24). □

Theorem 3.5. *Let $d \in \mathbb{N}$ and $T > 0$. Let $u \in C^4(\Omega)$ and $v \in C^3(\Omega)$ be the classical solution of the wave equations* (15)*, and let (u_θ, v_θ) denote the PINN approximation with parameter $\theta \in \Theta$. Then the total error satisfies*

$$\begin{aligned}\int_\Omega (|\hat{u}(\boldsymbol{x},t)|^2 + |\nabla\hat{u}(\boldsymbol{x},t)|^2 + |\hat{v}(\boldsymbol{x},t)|^2)\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t &\le C_T T\exp(2T)\\ &= \mathcal{O}(\mathcal{E}_T(\theta,\mathcal{S})^2 + M_{int}^{-\frac{2}{d+1}} + M_{tb}^{-\frac{2}{d}} + M_{sb}^{-\frac{1}{d}}).\end{aligned} \tag{28}$$

The constant C_T is defined by (30) *below.*

Proof. By combining Theorem 3.4 with the quadrature error formula (14), we have

$$\begin{aligned}
&\int_D |R_{tb1}|^2\,\mathrm{d}\boldsymbol{x} = \int_D |R_{tb1}|^2\,\mathrm{d}\boldsymbol{x} - \mathcal{Q}^D_{M_{tb}}(R^2_{tb1}) + \mathcal{Q}^D_{M_{tb}}(R^2_{tb1}) \le C_{(R^2_{tb1})} M_{tb}^{-\frac{2}{d}} + \mathcal{Q}^D_{M_{tb}}(R^2_{tb1}),\\
&\int_D |R_{tb2}|^2\,\mathrm{d}\boldsymbol{x} = \int_D |R_{tb2}|^2\,\mathrm{d}\boldsymbol{x} - \mathcal{Q}^D_{M_{tb}}(R^2_{tb2}) + \mathcal{Q}^D_{M_{tb}}(R^2_{tb2}) \le C_{(R^2_{tb2})} M_{tb}^{-\frac{2}{d}} + \mathcal{Q}^D_{M_{tb}}(R^2_{tb2}),\\
&\int_D |\nabla R_{tb1}|^2\,\mathrm{d}\boldsymbol{x} = \int_D |\nabla R_{tb1}|^2\,\mathrm{d}\boldsymbol{x} - \mathcal{Q}^D_{M_{tb}}(|\nabla R_{tb1}|^2) + \mathcal{Q}^D_{M_{tb}}(|\nabla R_{tb1}|^2) \le C_{(|\nabla R_{tb1}|^2)} M_{tb}^{-\frac{2}{d}} + \mathcal{Q}^D_{M_{tb}}(|\nabla R_{tb1}|^2),\\
&\int_\Omega |R_{int1}|^2\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t = \int_\Omega |R_{int1}|^2\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t - \mathcal{Q}^\Omega_{M_{int}}(R^2_{int1}) + \mathcal{Q}^\Omega_{M_{int}}(R^2_{int1}) \le C_{(R^2_{int1})} M_{int}^{-\frac{2}{d+1}} + \mathcal{Q}^\Omega_{M_{int}}(R^2_{int1}),\\
&\int_\Omega |R_{int2}|^2\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t = \int_\Omega |R_{int2}|^2\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t - \mathcal{Q}^\Omega_{M_{int}}(R^2_{int2}) + \mathcal{Q}^\Omega_{M_{int}}(R^2_{int2}) \le C_{(R^2_{int2})} M_{int}^{-\frac{2}{d+1}} + \mathcal{Q}^\Omega_{M_{int}}(R^2_{int2}),\\
&\int_\Omega |\nabla R_{int1}|^2\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t = \int_\Omega |\nabla R_{int1}|^2\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t - \mathcal{Q}^\Omega_{M_{int}}(|\nabla R_{int1}|^2) + \mathcal{Q}^\Omega_{M_{int}}(|\nabla R_{int1}|^2)\\
&\qquad\qquad \le C_{(|\nabla R_{int1}|^2)} M_{int}^{-\frac{2}{d+1}} + \mathcal{Q}^\Omega_{M_{int}}(|\nabla R_{int1}|^2),\\
&\int_{\Omega_*} |R_{sb1}|^2\,\mathrm{d}s(\boldsymbol{x})\,\mathrm{d}t = \int_{\Omega_*} |R_{sb1}|^2\,\mathrm{d}s(\boldsymbol{x})\,\mathrm{d}t - \mathcal{Q}^{\Omega_*}_{M_{sb}}(R^2_{sb1}) + \mathcal{Q}^{\Omega_*}_{M_{sb}}(R^2_{sb1}) \le C_{(R^2_{sb1})} M_{sb}^{-\frac{2}{d}} + \mathcal{Q}^{\Omega_*}_{M_{sb}}(R^2_{sb1}),\\
&\int_{\Omega_*} |R_{sb2}|^2\,\mathrm{d}s(\boldsymbol{x})\,\mathrm{d}t = \int_{\Omega_*} |R_{sb2}|^2\,\mathrm{d}s(\boldsymbol{x})\,\mathrm{d}t - \mathcal{Q}^{\Omega_*}_{M_{sb}}(R^2_{sb2}) + \mathcal{Q}^{\Omega_*}_{M_{sb}}(R^2_{sb2}) \le C_{(R^2_{sb2})} M_{sb}^{-\frac{2}{d}} + \mathcal{Q}^{\Omega_*}_{M_{sb}}(R^2_{sb2}).
\end{aligned}$$

Combining the fact that $C_{(R^2_{tb1})} \lesssim \|R^2_{tb1}\|_{C^n}$ and $\|R^2_{tb1}\|_{C^n} \le 2^n \|R^2_{tb1}\|^2_{C^n}$ with Lemma 8.2, it holds

$$C_{(R^2_{tb1})} \lesssim \|\hat{u}\|^2_{C^2} \le 2(\|u\|^2_{C^2} + \|u_\theta\|^2_{C^2}) \lesssim \|u\|^2_{C^2} + 16^{2L}(d+1)^8(e^2 2^4 W^3 R^2 \|\sigma\|_{C^2})^{4L}. \tag{29}$$

In a similar way, we can estimate the terms $\int_D |R_{tb2}|^2\,\mathrm{d}\boldsymbol{x}$, $\int_D |\nabla R_{tb1}|^2\,\mathrm{d}\boldsymbol{x}$, $\int_\Omega |R_{int1}|^2\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t$, $\int_\Omega |R_{int2}|^2\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t$ and $\int_\Omega |\nabla R_{int1}|^2\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t$.

By the above inequalities and (24), it holds that

$$\int_\Omega (|\hat{u}(\boldsymbol{x},t)|^2 + |\nabla\hat{u}(\boldsymbol{x},t)|^2 + |\hat{v}(\boldsymbol{x},t)|^2)\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t \le C_T T\exp(2T),$$

where

$$\begin{aligned}
C_T =& C_{(R^2_{tb1})} M_{tb}^{-\frac{2}{d}} + \mathcal{Q}^D_{M_{tb}}(R^2_{tb1}) + C_{(R^2_{tb2})} M_{tb}^{-\frac{2}{d}} + \mathcal{Q}^D_{M_{tb}}(R^2_{tb2}) + C_{(|\nabla R_{tb1}|^2)} M_{tb}^{-\frac{2}{d}} + \mathcal{Q}^D_{M_{tb}}(|\nabla R_{tb1}|^2)\\
&+ C_{(R^2_{int1})} M_{int}^{-\frac{2}{d+1}} + \mathcal{Q}^\Omega_{M_{int}}(R^2_{int1}) + C_{(R^2_{int2})} M_{int}^{-\frac{2}{d+1}} + \mathcal{Q}^\Omega_{M_{int}}(R^2_{int2})\\
&+ C_{(|\nabla R_{int1}|^2)} M_{int}^{-\frac{2}{d+1}} + \mathcal{Q}^\Omega_{M_{int}}(|\nabla R_{int1}|^2)\\
&+ 2C_{\partial D_1}|T|^{1/2}\left(C_{(R^2_{sb1})} M_{sb}^{-\frac{2}{d}} + \mathcal{Q}^{\Omega_*}_{M_{sb}}\right)^{1/2} + 2C_{\partial D_2}|T|^{1/2}\left(C_{(R^2_{sb2})} M_{sb}^{-\frac{2}{d}} + \mathcal{Q}^{\Omega_*}_{M_{sb}}\right)^{1/2}.
\end{aligned} \tag{30}$$

□

4. Physics Informed Neural Networks for Approximating the Nonlinear Klein-Gordon Equation

4.1. Nonlinear Klein-Gordon Equation

Let $D \subset \mathbb{R}^d$ be an open connected bounded set with a boundary ∂D. We consider the following nonlinear Klein-Gordon equation:

$$u_t - v = 0, \quad \varepsilon^2 v_t = a^2 \Delta u - \varepsilon_1^2 u - g(u) + f, \qquad \text{in } D \times [0,T], \tag{31a}$$
$$u(\boldsymbol{x}, 0) = \psi_1(\boldsymbol{x}), \quad v(\boldsymbol{x}, 0) = \psi_2(\boldsymbol{x}), \qquad \text{in } D, \tag{31b}$$
$$u(\boldsymbol{x}, t)|_{\partial D} = u_d(t) \qquad \text{in } \partial D \times [0,T], \tag{31c}$$

where u and v are the field functions to be solved for, f is a source term, and u_d, ψ_1 and ψ_2 denote the boundary/initial conditions. $\varepsilon > 0$, $a > 0$ and $\varepsilon_1 \geq 0$ are constants. $g(u)$ is a nonlinear term. We assume that g is globally Lipschitz, i.e. there exists a constant L (independent of v and w) such that

$$|g(v) - g(w)| \leq L|v - w|, \qquad \forall v, w \in \mathbb{R}. \tag{32}$$

Notice that the problem (31) is nonlinear because of the nonlinear term $g(u)$. With the nonlinear term given by $g(u) = \sin(u)$, equation (31a) becomes the well-known Sine-Gordon equation [15].

Remark 4.1. *The existence and regularity of the solution to the nonlinear Klein-Gordon equation with different nonlinear terms have been the subject of several studies in the literature; see [56, 35, 47, 48, 54].*

The book [54] provides the existence and regularity result of the following Klein-Gordon equation, $u_{tt} + \alpha u_t - \Delta u + g(u) = f$. Let $\alpha \in \mathbb{R}$, $g(u)$ be a C^2 function from $\mathbb{R}$ to $\mathbb{R}$ and satisfy certain assumptions. If $f \in C([0,T]; L^2(D))$, $\psi_1 \in H^1(D)$ and $\psi_2 \in L^2(D)$, then there exists a unique solution u to this Klein-Gordon equation such that $u \in C([0,T]; H^1(D))$ and $u_t \in C([0,T]; L^2(D))$. Furthermore, $f' \in C([0,T]; L^2(D))$, $\psi_1 \in H^2(D)$ and $\psi_2 \in H^1(D)$, it holds $u \in C([0,T]; H^2(D))$ and $u_t \in C([0,T]; H^1(D))$.

Let g be a smooth function of degree 2. The following equation is studied in [48], $u_{tt} - \Delta u + u + g(u, u_t, u_{tt}) = 0$, where it is reformulated as $\boldsymbol{u}_t = A\boldsymbol{u} + G(\boldsymbol{u})$, in which $\boldsymbol{u} = \begin{pmatrix} u \\ u_t \end{pmatrix}$, $A = \begin{pmatrix} 0 & 1 \\ \Delta - 1 & 0 \end{pmatrix}$ and $G = \begin{pmatrix} 0, \\ -g(u, u_t, u_{tt}) \end{pmatrix}$. Set $X = H^k(\mathbb{R}^n) \bigoplus H^{k-1}(\mathbb{R}^n)$, $k > n + 2 + 2a$ with $a > 1$. Given $\boldsymbol{u}_0 = \begin{pmatrix} \psi_1 \\ \psi_2 \end{pmatrix} \in X$ and $\|\boldsymbol{u}_0\|_X = \sigma$, there exists a $T_0 = T_0(\sigma)$ depending on the size of the initial data σ and a unique solution $\boldsymbol{u} \in C([0, T_0], X)$.

The reference [56] provides the following result. Under certain conditions for the nonlinear term $g(u)$, with $f = 0$, $d \leq 5$, $k \geq \frac{d}{2} + 1$, $\psi_1 \in H^k(D)$ and $\psi_2 \in H^{k-1}(D)$, there exists a unique solution $u \in C((0,\infty); H^k(D))$ of nonlinear Klein–Gordon equation.

The following result is due to [35]. Under certain conditions for the nonlinear term $g(u)$, with $f = 0$, $\psi_1 \in H^k(D)$ and $\psi_2 \in H^{k-1}(D)$ with a positive constant $k \geq 4$, there exists a positive constant T_k and a unique solution $u \in C([0,T_k]; H^k(D)) \cap C^1([0,T_k]; H^{k-1}(D)) \cap C^2([0,T_k]; H^{k-2}(D))$ to the nonlinear wave equations with different speeds of propagation.

A survey of literature indicates that, while several works have touched on the regularity of the solution to the nonlinear Klein-Gordon equations, none of them is comprehensive. To facilitate the subsequent analyses, we make the following assumption in light of Remark 4.1. Let $k \geq 1$, $g(u)$ and f be sufficiently smooth and bounded. Given $\psi_1 \in H^r(D)$ and $\psi_2 \in H^{r-1}(D)$ with $r \geq \frac{d}{2} + k$, we assume that there exists $T > 0$ and a classical solution (u, v) to the nonlinear Klein-Gordon equations (31) such that $u \in C([0,T]; H^k(D))$ and $v \in C([0,T]; H^{k-1}(D))$. Therefore, $u \in H^k(D \times [0,T])$ and $v \in H^{k-1}(D \times [0,T])$.

4.2. Physics Informed Neural Networks

Let $\Omega = D \times [0,T]$ and $\Omega_* = \partial D \times [0,T]$ be the space-time domain. We define the following residuals for the PINN approximation, $u_\theta : \Omega \to \mathbb{R}$ and $v_\theta : \Omega \to \mathbb{R}$, for the nonlinear Klein-Gordon equations (31):

$$R_{int1}[u_\theta, v_\theta](\boldsymbol{x}, t) = u_{\theta t} - v_\theta, \quad R_{int2}[u_\theta, v_\theta](\boldsymbol{x}, t) = \varepsilon^2 v_{\theta t} - a^2 \Delta u_\theta + \varepsilon_1^2 u_\theta + g(u_\theta) - f, \tag{33a}$$
$$R_{tb1}[u_\theta](\boldsymbol{x}) = u_\theta(\boldsymbol{x}, 0) - \psi_1(\boldsymbol{x}), \quad R_{tb2}[v_\theta](\boldsymbol{x}) = v_\theta(\boldsymbol{x}, 0) - \psi_2(\boldsymbol{x}), \tag{33b}$$

$$R_{sb}[v_\theta](\boldsymbol{x},t) = v_\theta(\boldsymbol{x},t)|_{\partial D} - u_{dt}(t), \tag{33c}$$

where $u_{dt} = \frac{\partial u_d}{\partial t}$. Note that for the exact solution (u, v), $R_{int1}[u, v] = R_{int2}[u, v] = R_{tb1}[u] = R_{tb2}[v] = R_{sb}[v] = 0$. With PINN we minimize the following generalization error,

$$\begin{aligned}\mathcal{E}_G(\theta)^2 = &\int_\Omega |R_{int1}[u_\theta, v_\theta](\boldsymbol{x},t)|^2 \,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t + \int_\Omega |R_{int2}[u_\theta, v_\theta](\boldsymbol{x},t)|^2 \,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t + \int_\Omega |\nabla R_{int1}[u_\theta, v_\theta](\boldsymbol{x},t)|^2 \,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t \\ &+ \int_D |R_{tb1}[u_\theta](\boldsymbol{x})|^2 \,\mathrm{d}\boldsymbol{x} + \int_D |R_{tb2}[v_\theta](\boldsymbol{x})|^2 \,\mathrm{d}\boldsymbol{x} + \int_D |\nabla R_{tb1}[u_\theta](\boldsymbol{x})|^2 \,\mathrm{d}\boldsymbol{x} \\ &+ \left(\int_{\Omega_*} |R_{sb}[v_\theta](\boldsymbol{x},t)|^2 \,\mathrm{d}s(\boldsymbol{x})\,\mathrm{d}t\right)^{\frac{1}{2}}.\end{aligned} \tag{34}$$

Let $\hat{u} = u_\theta - u$, $\hat{v} = v_\theta - v$, where (u, v) denotes the exact solution. We define the total error of the PINN approximation of the equations (31) as,

$$\mathcal{E}(\theta)^2 = \int_\Omega (|\hat{u}(\boldsymbol{x},t)|^2 + a^2|\nabla\hat{u}(\boldsymbol{x},t)|^2 + \varepsilon^2|\hat{v}(\boldsymbol{x},t)|^2)\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t. \tag{35}$$

Then we choose the training set $\mathcal{S} \subset \overline{D} \times [0, T]$ with $\mathcal{S} = \mathcal{S}_{int} \cup \mathcal{S}_{sb} \cup \mathcal{S}_{tb}$, based on suitable quadrature points:

- Interior training points $\mathcal{S}_{int} = \{z_n\}$ for $1 \le n \le N_{int}$, with each $z_n = (\boldsymbol{x}, t)_n \in D \times (0, T)$.
- Spatial boundary training points $\mathcal{S}_{sb} = \{z_n\}$ for $1 \le n \le N_{sb}$, with each $z_n = (\boldsymbol{x}, t)_n \in \partial D \times (0, T)$.
- Temporal boundary training points $\mathcal{S}_{tb} = \{\boldsymbol{x}_n\}$ for $1 \le n \le N_{tb}$ with each $\boldsymbol{x}_n \in D$.

The integrals in (34) are approximated by a numerical quadrature rule, resulting in the training loss,

$$\begin{aligned}\mathcal{E}_T(\theta,\mathcal{S})^2 = &\mathcal{E}_T^{int1}(\theta,\mathcal{S}_{int})^2 + \mathcal{E}_T^{int2}(\theta,\mathcal{S}_{int})^2 + \mathcal{E}_T^{int3}(\theta,\mathcal{S}_{int})^2 + \mathcal{E}_T^{tb1}(\theta,\mathcal{S}_{tb})^2 + \mathcal{E}_T^{tb2}(\theta,\mathcal{S}_{tb})^2 \\ &+ \mathcal{E}_T^{tb3}(\theta,\mathcal{S}_{tb})^2 + \mathcal{E}_T^{sb}(\theta,\mathcal{S}_{sb}),\end{aligned} \tag{36}$$

where

$$\mathcal{E}_T^{int1}(\theta,\mathcal{S}_{int})^2 = \sum_{n=1}^{N_{int}} \omega_{int}^n |R_{int1}[u_\theta, v_\theta](\boldsymbol{x}_{int}^n, t_{int}^n)|^2, \quad \mathcal{E}_T^{tb1}(\theta,\mathcal{S}_{tb})^2 = \sum_{n=1}^{N_{tb}} \omega_{tb}^n |R_{tb1}[u_\theta](\boldsymbol{x}_{tb}^n)|^2, \tag{37a}$$

$$\mathcal{E}_T^{int2}(\theta,\mathcal{S}_{int})^2 = \sum_{n=1}^{N_{int}} \omega_{int}^n |R_{int2}[u_\theta, v_\theta](\boldsymbol{x}_{int}^n, t_{int}^n)|^2, \quad \mathcal{E}_T^{tb2}(\theta,\mathcal{S}_{tb})^2 = \sum_{n=1}^{N_{tb}} \omega_{tb}^n |R_{tb2}[v_\theta](\boldsymbol{x}_{tb}^n)|^2, \tag{37b}$$

$$\mathcal{E}_T^{int3}(\theta,\mathcal{S}_{int})^2 = \sum_{n=1}^{N_{int}} \omega_{int}^n |\nabla R_{int1}[u_\theta, v_\theta](\boldsymbol{x}_{int}^n, t_{int}^n)|^2, \quad \mathcal{E}_T^{tb3}(\theta,\mathcal{S}_{tb})^2 = \sum_{n=1}^{N_{tb}} \omega_{tb}^n |\nabla R_{tb1}[u_\theta](\boldsymbol{x}_{tb}^n)|^2, \tag{37c}$$

$$\mathcal{E}_T^{sb}(\theta,\mathcal{S}_{sb})^2 = \sum_{n=1}^{N_{sb}} \omega_{sb}^n |R_{sb}[v_\theta](\boldsymbol{x}_{sb}^n, t_{sb}^n)|^2. \tag{37d}$$

Here the quadrature points in space-time constitute the data sets $\mathcal{S}_{int} = \{(\boldsymbol{x}_{int}^n, t_{int}^n)\}_{n=1}^{N_{int}}$, $\mathcal{S}_{tb} = \{\boldsymbol{x}_{tb}^n)\}_{n=1}^{N_{tb}}$ and $\mathcal{S}_{sb} = \{(\boldsymbol{x}_{sb}^n, t_{sb}^n)\}_{n=1}^{N_{sb}}$, and $\omega_\star^n$ are the quadrature weights with $\star$ being int, tb or sb.

4.3. Error Analysis

By subtracting the equations (31) from the residual equations (33), we get

$$R_{int1} = \hat{u}_t - \hat{v}, \tag{38a}$$

$$R_{int2} = \varepsilon^2\hat{v}_t - a^2\Delta\hat{u} + \varepsilon_1^2\hat{u} + g(u_\theta) - g(u), \tag{38b}$$

$$R_{tb1} = \hat{u}(\boldsymbol{x}, 0), \quad R_{tb2} = \hat{v}(\boldsymbol{x}, 0), \quad R_{sb} = \hat{v}(\boldsymbol{x}, t)|_{\partial D}. \tag{38c}$$

The results on the PINN approximations to the nonlinear Klein-Gordon equations are summarized in the following theorems.

Theorem 4.2. *Let $n \geq 2$, d, r, $k \in \mathbb{N}$ with $k \geq 3$. Assume that $g(u)$ is Lipschitz continuous, $u \in H^k(D \times [0,T])$ and $v \in H^{k-1}(D \times [0,T])$. Then for every integer $N > 5$, there exist* tanh *neural networks u_θ and v_θ, each with two hidden layers, of widths at most $3\lceil \frac{k+n-2}{2} \rceil |P_{k-1,d+2}| + \lceil NT \rceil + d(N-1)$ and $3\lceil \frac{d+n+1}{2} \rceil |P_{d+2,d+2}| \lceil NT \rceil N^d$, such that*

$$\|R_{int1}\|_{L^2(\Omega)}, \|R_{tb1}\|_{L^2(D)} \lesssim \ln N N^{-k+1}, \tag{39a}$$

$$\|R_{int2}\|_{L^2(\Omega)}, \|\nabla R_{int1}\|_{L^2(\Omega)}, \|\nabla R_{tb1}\|_{L^2(D)} \lesssim \ln^2 N N^{-k+2}, \tag{39b}$$

$$\|R_{tb2}\|_{L^2(D)}, \|R_{sb}\|_{L^2(\partial D \times [0,T])} \lesssim \ln N N^{-k+2}. \tag{39c}$$

Proof. Being similar to the proof of Theorem 3.3, we can end the proof by noting $u \in H^k(D \times [0,T])$, $v \in H^{k-1}(D \times [0,T])$ and Lemma 8.3. □

Theorem 4.2 implies that the PINN residuals in (33) can be made arbitrarily small by choosing a sufficiently large N. Therefore, the generalization error $\mathcal{E}_G(\theta)^2$ can be made arbitrarily small.

We next show that the PINN total approximation error $\mathcal{E}(\theta)^2$ can be controlled by the generalization error $\mathcal{E}_G(\theta)^2$ (Theorem 4.3 below), and by the training error $\mathcal{E}_T(\theta, \mathcal{S})^2$ (Theorem 4.4 below).

Theorem 4.3. *Let $d \in \mathbb{N}$, $u \in C^1(\Omega)$ and $v \in C^0(\Omega)$ be the classical solution of the nonlinear Klein-Gordon equation* (31)*. Let (u_θ, v_θ) denote the PINN approximation with parameter θ. Then the following relation holds,*

$$\mathcal{E}(\theta)^2 = \int_\Omega (|\hat{u}(\boldsymbol{x},t)|^2 + a^2|\nabla \hat{u}(\boldsymbol{x},t)|^2 + \varepsilon^2 |\hat{v}(\boldsymbol{x},t)|^2)\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t \leq C_G T \exp\left((2 + \varepsilon_1^2 + L + a^2)T\right), \tag{40}$$

where C_G is defined by (74) *in Appendix 8.2.*

The proof for Theorem 4.3 is provided in the Appendix 8.2.

Theorem 4.4. *Let $d \in \mathbb{N}$ and $T > 0$, and let $u \in C^4(\Omega)$ and $v \in C^3(\Omega)$ be the classical solution to the nonlinear Klein-Gordon equation* (31)*. Let (u_θ, v_θ) denote the PINN approximation with parameter $\theta \in \Theta$. Then the following relation holds,*

$$\begin{aligned}
&\int_\Omega (|\hat{u}(\boldsymbol{x},t)|^2 + a^2|\nabla \hat{u}(\boldsymbol{x},t)|^2 + \varepsilon^2 |\hat{v}(\boldsymbol{x},t)|^2)\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t \leq C_T T \exp\left((2 + \varepsilon_1^2 + L + a^2)T\right) \\
&\quad = \mathcal{O}(\mathcal{E}_T(\theta, \mathcal{S})^2 + M_{int}^{-\frac{2}{d+1}} + M_{tb}^{-\frac{2}{d}} + M_{sb}^{-\frac{1}{d}}),
\end{aligned} \tag{41}$$

where the constant C_T is defined by

$$\begin{aligned}
C_T =& C_{(R_{tb1}^2)} M_{tb}^{-\frac{2}{d}} + \mathcal{Q}_{M_{tb}}^D(R_{tb1}^2) + \varepsilon^2 \left(C_{(R_{tb2}^2)} M_{tb}^{-\frac{2}{d}} + \mathcal{Q}_{M_{tb}}^D(R_{tb2}^2) \right) \\
&+ a^2 \left(C_{(|\nabla R_{tb1}|^2)} M_{tb}^{-\frac{2}{d}} + \mathcal{Q}_{M_{tb}}^D(|\nabla R_{tb1}|^2) \right) + C_{(R_{int1}^2)} M_{int}^{-\frac{2}{d+1}} + \mathcal{Q}_{M_{int}}^\Omega(R_{int1}^2) \\
&+ C_{(R_{int2}^2)} M_{int}^{-\frac{2}{d+1}} + \mathcal{Q}_{M_{int}}^\Omega(R_{int2}^2) + a^2 \left(C_{(|\nabla R_{int1}|^2)} M_{int}^{-\frac{2}{d+1}} + \mathcal{Q}_{M_{int}}^\Omega(|\nabla R_{int1}|^2) \right), \\
&+ 2C_{\partial D} |T|^{\frac{1}{2}} \left(C_{(R_{sb}^2)} M_{sb}^{-\frac{2}{d}} + \mathcal{Q}_{M_{sb}}^{\Omega_*}(R_{sb}^2) \right)^{\frac{1}{2}}.
\end{aligned}$$

Proof. Using Lemma 8.2, Theorem 4.3 and the quadrature error formula (14) leads to this result. □

It follows from Theorem 4.4 that the PINN approximation error $\mathcal{E}(\theta)^2$ can be arbitrarily small, provided that the training error $\mathcal{E}_T(\theta, \mathcal{S})^2$ is sufficiently small and the sample set is sufficiently large.

5. Physics Informed Neural Networks for Approximating Linear Elastodynamic Equation

5.1. Linear Elastodynamic Equation

Consider an elastic body occupying an open, bounded convex polyhedral domain $D \subset \mathbb{R}^d$. The boundary $\partial D = \Gamma_D \cup \Gamma_N$, with the outward unit normal vector $\boldsymbol{n}$, is assumed to be composed of two disjoint portions $\Gamma_D \neq \emptyset$ and Γ_N, with $\Gamma_D \cap \Gamma_N = \emptyset$. Given a suitable external load $\boldsymbol{f} \in L^2((0,T];\boldsymbol{L}^2(D))$, and suitable initial/boundary data $\boldsymbol{g} \in C^1((0,T];\boldsymbol{H}^{\frac{1}{2}}(\Gamma_N))$, $\boldsymbol{\psi}_1 \in \boldsymbol{H}^{\frac{1}{2}}_{0,\Gamma_D}(D)$ and $\boldsymbol{\psi}_2 \in \boldsymbol{L}^2(D)$, we consider the linear elastodynamic equations,

$$\boldsymbol{u}_t - \boldsymbol{v} = 0, \quad \rho \boldsymbol{v}_t - 2\mu\nabla\cdot(\underline{\boldsymbol{\varepsilon}}(\boldsymbol{u})) - \lambda\nabla(\nabla\cdot\boldsymbol{u}) = \boldsymbol{f}, \qquad \text{in } D\times[0,T], \tag{42a}$$

$$\boldsymbol{u} = \boldsymbol{u}_d \qquad \text{in } \Gamma_D\times[0,T], \tag{42b}$$

$$2\mu\underline{\boldsymbol{\varepsilon}}(\boldsymbol{u})\boldsymbol{n} + \lambda(\nabla\cdot\boldsymbol{u})\boldsymbol{n} = \boldsymbol{g} \qquad \text{in } \Gamma_N\times[0,T], \tag{42c}$$

$$\boldsymbol{u}(\boldsymbol{x},0) = \boldsymbol{\psi}_1(\boldsymbol{x}), \quad \boldsymbol{v}(\boldsymbol{x},0) = \boldsymbol{\psi}_2(\boldsymbol{x}), \qquad \text{in } D. \tag{42d}$$

In the above system, $\boldsymbol{u} = (u_1, u_2, \cdots, u_d)$ and $\boldsymbol{v} = (v_1, v_2, \cdots, v_d)$ denote the displacement and the velocity, respectively, and $[0,T]$ (with $T>0$) denotes the time domain. $\underline{\boldsymbol{\varepsilon}}(\boldsymbol{u})$ is the strain tensor, $\underline{\boldsymbol{\varepsilon}}(\boldsymbol{u}) = \frac{1}{2}(\nabla\boldsymbol{u}+\nabla\boldsymbol{u}^T)$. The constants λ and μ are the first and the second Lamé parameters, respectively.

Combining the two equations in (42a), we can recover the classical linear elastodynamics equation:

$$\rho\boldsymbol{u}_{tt} - 2\mu\nabla\cdot(\underline{\boldsymbol{\varepsilon}}(\boldsymbol{u})) - \lambda\nabla(\nabla\cdot\boldsymbol{u}) = \boldsymbol{f} \qquad \text{in } D\times[0,T]. \tag{43}$$

The well-posedness of this equation is established in [29].

Lemma 5.1 ([29, 59])**.** *Let $\boldsymbol{\psi}_1 \in H^r(D)$, $\boldsymbol{\psi}_2 \in H^{r-1}(D)$ and $\boldsymbol{f} \in H^{r-1}(D\times[0,T])$ with $r \geq 1$. Then there exists a unique solution $\boldsymbol{u}$ to the classical linear elastodynamic equation* (43) *such that $\boldsymbol{u}(\boldsymbol{x},0) = \boldsymbol{\psi}_1(\boldsymbol{x})$, $\boldsymbol{u}_t(\boldsymbol{x},0) = \boldsymbol{\psi}_2(\boldsymbol{x})$ and $\boldsymbol{u} \in C^l([0,T];H^{r-l}(D))$ with $0 \leq l \leq r$.*

Lemma 5.2. *Let $k \in \mathbb{N}$, $\boldsymbol{\psi}_1 \in H^r(D)$, $\boldsymbol{\psi}_2 \in H^{r-1}(D)$ and $\boldsymbol{f} \in H^{r-1}(D\times[0,T])$ with $r > \frac{d}{2}+k$, then there exists $T>0$ and a classical solution $(\boldsymbol{u},\boldsymbol{v})$ to the elastodynamic equations* (42) *such that $\boldsymbol{u}(\boldsymbol{x},0) = \boldsymbol{\psi}_1(\boldsymbol{x})$, $\boldsymbol{u}_t(\boldsymbol{x},0) = \boldsymbol{\psi}_2(\boldsymbol{x})$, $\boldsymbol{u} \in H^k(D\times[0,T])$ and $\boldsymbol{v} \in H^{k-1}(D\times[0,T])$.*

Proof. As $r > \frac{d}{2}+k$, $H^{r-k}(D)$ is a Banach algebra. By Lemma 5.1, there exists $T>0$ and the solution $(\boldsymbol{u},\boldsymbol{v})$ to the linear elastodynamics equations such that $\boldsymbol{u}(\boldsymbol{x},0) = \boldsymbol{\psi}_1(\boldsymbol{x})$, $\boldsymbol{v}(\boldsymbol{x},0) = \boldsymbol{\psi}_2(\boldsymbol{x})$, $\boldsymbol{u} \in C^l([0,T];H^{r-l}(D))$ with $0 \leq l \leq r$ and $\boldsymbol{v} \in C^l([0,T];H^{r-1-l}(D))$ with $0 \leq l \leq r-1$.

Since $\boldsymbol{u} \in \cap_{l=0}^k C^l([0,T];H^{r-l}(D))$ and $r-l+l \geq k$ with $r > \frac{d}{2}+k$, it holds that $\boldsymbol{u} \in H^k(D\times[0,T])$. Similarly, we obtain $\boldsymbol{v} \in H^{k-1}(D\times[0,T])$. □

5.2. Physics Informed Neural Networks

We now consider the PINN approximation of the linear elastodynamic equations (42). Let $\Omega = D\times[0,T]$, $\Omega_D = \Gamma_D\times[0,T]$ and $\Omega_N = \Gamma_N\times[0,T]$ denote the space-time domain. Define the following residuals for the PINN approximation $\boldsymbol{u}_\theta : \Omega \to \mathbb{R}$ and $\boldsymbol{v}_\theta : \Omega \to \mathbb{R}$ for the elastodynamic equations (42):

$$\boldsymbol{R}_{int1}[\boldsymbol{u}_\theta,\boldsymbol{v}_\theta](\boldsymbol{x},t) = \boldsymbol{u}_{\theta t} - \boldsymbol{v}_\theta, \quad \boldsymbol{R}_{int2}[\boldsymbol{u}_\theta,\boldsymbol{v}_\theta](\boldsymbol{x},t) = \rho\boldsymbol{v}_{\theta t} - 2\mu\nabla\cdot(\underline{\boldsymbol{\varepsilon}}(\boldsymbol{u}_\theta)) - \lambda\nabla(\nabla\cdot\boldsymbol{u}_\theta) - \boldsymbol{f}, \tag{44a}$$

$$\boldsymbol{R}_{tb1}[\boldsymbol{u}_\theta](\boldsymbol{x}) = \boldsymbol{u}_\theta(\boldsymbol{x},0) - \boldsymbol{\psi}_1(\boldsymbol{x}), \quad \boldsymbol{R}_{tb2}[\boldsymbol{v}_\theta](\boldsymbol{x}) = \boldsymbol{v}_\theta(\boldsymbol{x},0) - \boldsymbol{\psi}_2(\boldsymbol{x}), \tag{44b}$$

$$\boldsymbol{R}_{sb1}[\boldsymbol{v}_\theta](\boldsymbol{x},t) = \boldsymbol{v}_\theta|_{\Gamma_D} - \boldsymbol{u}_{dt}, \quad \boldsymbol{R}_{sb2}[\boldsymbol{u}_\theta](\boldsymbol{x},t) = (2\mu\underline{\boldsymbol{\varepsilon}}(\boldsymbol{u}_\theta)\boldsymbol{n} + \lambda(\nabla\cdot\boldsymbol{u}_\theta)\boldsymbol{n})|_{\Gamma_N} - \boldsymbol{g}. \tag{44c}$$

Note that for the exact solution $(\boldsymbol{u},\boldsymbol{v})$, we have $\boldsymbol{R}_{int1}[\boldsymbol{u},\boldsymbol{v}] = \boldsymbol{R}_{int2}[\boldsymbol{u},\boldsymbol{v}] = \boldsymbol{R}_{tb1}[\boldsymbol{u}] = \boldsymbol{R}_{tb2}[\boldsymbol{v}] = \boldsymbol{R}_{sb1}[\boldsymbol{v}] = \boldsymbol{R}_{sb2}[\boldsymbol{u}] = 0$. With PINN we minimize the the following generalization error,

$$\begin{aligned}\mathcal{E}_G(\theta)^2 = &\int_\Omega |\boldsymbol{R}_{int1}[\boldsymbol{u}_\theta,\boldsymbol{v}_\theta](\boldsymbol{x},t)|^2\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t + \int_\Omega |\boldsymbol{R}_{int2}[\boldsymbol{u}_\theta,\boldsymbol{v}_\theta](\boldsymbol{x},t)|^2\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t + \int_\Omega |\underline{\boldsymbol{\varepsilon}}(\boldsymbol{R}_{int1}[\boldsymbol{u}_\theta,\boldsymbol{v}_\theta](\boldsymbol{x},t))|^2\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t\\ &+ \int_\Omega |\nabla\cdot(\boldsymbol{R}_{int1}[\boldsymbol{u}_\theta,\boldsymbol{v}_\theta](\boldsymbol{x},t))|^2\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t + \int_D |\boldsymbol{R}_{tb1}[\boldsymbol{u}_\theta](\boldsymbol{x})|^2\,\mathrm{d}\boldsymbol{x} + \int_D |\boldsymbol{R}_{tb2}[\boldsymbol{v}_\theta](\boldsymbol{x})|^2\,\mathrm{d}\boldsymbol{x}\\ &+ \int_D |\underline{\boldsymbol{\varepsilon}}(\boldsymbol{R}_{tb1}[\boldsymbol{u}_\theta](\boldsymbol{x}))|^2\,\mathrm{d}\boldsymbol{x} + \int_D |\nabla\cdot\boldsymbol{R}_{tb1}[\boldsymbol{u}_\theta](\boldsymbol{x})|^2\,\mathrm{d}\boldsymbol{x}\end{aligned}$$

$$+\left(\int_{\Omega_D}|\boldsymbol{R}_{sb1}[\boldsymbol{v}_\theta](\boldsymbol{x},t)|^2\,\mathrm{d}s(\boldsymbol{x})\,\mathrm{d}t\right)^{\frac{1}{2}}+\left(\int_{\Omega_N}|\boldsymbol{R}_{sb2}[\boldsymbol{u}_\theta](\boldsymbol{x},t)|^2\,\mathrm{d}s(\boldsymbol{x})\,\mathrm{d}t\right)^{\frac{1}{2}}. \tag{45}$$

Let $\hat{\boldsymbol{u}}=\boldsymbol{u}_\theta-\boldsymbol{u},\ \hat{\boldsymbol{v}}=\boldsymbol{v}_\theta-\boldsymbol{v}$ denote the difference between the solution to the elastodynamic equations (42) and the PINN approximation with parameter θ. We define the total error of the PINN approximation as,

$$\mathcal{E}(\theta)^2=\int_\Omega(|\hat{\boldsymbol{u}}(\boldsymbol{x},t)|^2+2\mu|\underline{\boldsymbol{\varepsilon}}(\hat{\boldsymbol{u}}(\boldsymbol{x},t))|^2+\lambda|\nabla\cdot\hat{\boldsymbol{u}}(\boldsymbol{x},t)|^2+\rho|\hat{\boldsymbol{v}}(\boldsymbol{x},t)|^2)\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t. \tag{46}$$

We choose the training set $\mathcal{S}\subset\overline{D}\times[0,T]$ based on suitable quadrature points. The full training set is defined by $\mathcal{S}=\mathcal{S}_{int}\cup\mathcal{S}_{sb}\cup\mathcal{S}_{tb}$, and $\mathcal{S}_{sb}=\mathcal{S}_{sb1}\cup\mathcal{S}_{sb2}$:

- Interior training points $\mathcal{S}_{int}=\{z_n\}$ for $1\le n\le N_{int}$, with each $z_n=(\boldsymbol{x},t)_n\in D\times(0,T)$.
- Spatial boundary training points $\mathcal{S}_{sb1}=\{z_n\}$ for $1\le n\le N_{sb1}$, with each $z_n=(\boldsymbol{x},t)_n\in\Gamma_D\times(0,T)$, and $\mathcal{S}_{sb2}=\{z_n\}$ for $1\le n\le N_{sb2}$, with each $z_n=(\boldsymbol{x},t)_n\in\Gamma_N\times(0,T)$.
- Temporal boundary training points $\mathcal{S}_{tb}=\{\boldsymbol{x}_n\}$ for $1\le n\le N_{tb}$ with each $\boldsymbol{x}_n\in D$.

Then, the integrals in (45) can be approximated by a suitable numerical quadrature, resulting in the following training loss,

$$\begin{aligned}\mathcal{E}_T(\theta,\mathcal{S})^2=&\,\mathcal{E}_T^{int1}(\theta,\mathcal{S}_{int})^2+\mathcal{E}_T^{int2}(\theta,\mathcal{S}_{int})^2+\mathcal{E}_T^{int3}(\theta,\mathcal{S}_{int})^2+\mathcal{E}_T^{int4}(\theta,\mathcal{S}_{int})^2+\mathcal{E}_T^{tb1}(\theta,\mathcal{S}_{tb})^2\\&+\mathcal{E}_T^{tb2}(\theta,\mathcal{S}_{tb})^2+\mathcal{E}_T^{tb3}(\theta,\mathcal{S}_{tb})^2+\mathcal{E}_T^{tb4}(\theta,\mathcal{S}_{tb})^2+\mathcal{E}_T^{sb1}(\theta,\mathcal{S}_{sb1})+\mathcal{E}_T^{sb2}(\theta,\mathcal{S}_{sb2}),\end{aligned} \tag{47}$$

where,

$$\mathcal{E}_T^{int1}(\theta,\mathcal{S}_{int})^2=\sum_{n=1}^{N_{int}}\omega_{int}^n|\boldsymbol{R}_{int1}[\boldsymbol{u}_\theta,\boldsymbol{v}_\theta](\boldsymbol{x}_{int}^n,t_{int}^n)|^2,\quad \mathcal{E}_T^{tb1}(\theta,\mathcal{S}_{tb})^2=\sum_{n=1}^{N_{tb}}\omega_{tb}^n|\boldsymbol{R}_{tb1}[\boldsymbol{u}_\theta](\boldsymbol{x}_{tb}^n)|^2, \tag{48a}$$

$$\mathcal{E}_T^{int2}(\theta,\mathcal{S}_{int})^2=\sum_{n=1}^{N_{int}}\omega_{int}^n|\boldsymbol{R}_{int2}[\boldsymbol{u}_\theta,\boldsymbol{v}_\theta](\boldsymbol{x}_{int}^n,t_{int}^n)|^2,\quad \mathcal{E}_T^{tb2}(\theta,\mathcal{S}_{tb})^2=\sum_{n=1}^{N_{tb}}\omega_{tb}^n|\boldsymbol{R}_{tb2}[\boldsymbol{v}_\theta](\boldsymbol{x}_{tb}^n)|^2, \tag{48b}$$

$$\mathcal{E}_T^{int3}(\theta,\mathcal{S}_{int})^2=\sum_{n=1}^{N_{int}}\omega_{int}^n|\underline{\boldsymbol{\varepsilon}}(\boldsymbol{R}_{int1}[\boldsymbol{u}_\theta,\boldsymbol{v}_\theta](\boldsymbol{x}_{int}^n,t_{int}^n))|^2, \tag{48c}$$

$$\mathcal{E}_T^{int4}(\theta,\mathcal{S}_{int})^2=\sum_{n=1}^{N_{int}}\omega_{int}^n|\nabla\cdot\boldsymbol{R}_{int1}[\boldsymbol{u}_\theta,\boldsymbol{v}_\theta](\boldsymbol{x}_{int}^n,t_{int}^n)|^2, \tag{48d}$$

$$\mathcal{E}_T^{tb3}(\theta,\mathcal{S}_{tb})^2=\sum_{n=1}^{N_{tb}}\omega_{tb}^n|\underline{\boldsymbol{\varepsilon}}(\boldsymbol{R}_{tb1}[\boldsymbol{u}_\theta](\boldsymbol{x}_{tb}^n))|^2,\quad \mathcal{E}_T^{tb4}(\theta,\mathcal{S}_{tb})^2=\sum_{n=1}^{N_{tb}}\omega_{tb}^n|\nabla\cdot\boldsymbol{R}_{tb1}[\boldsymbol{u}_\theta](\boldsymbol{x}_{tb}^n)|^2, \tag{48e}$$

$$\mathcal{E}_T^{sb1}(\theta,\mathcal{S}_{sb1})^2=\sum_{n=1}^{N_{sb1}}\omega_{sb1}^n|\boldsymbol{R}_{sb1}[\boldsymbol{v}_\theta](\boldsymbol{x}_{sb}^n,t_{sb}^n)|^2,\quad \mathcal{E}_T^{sb2}(\theta,\mathcal{S}_{sb2})^2=\sum_{n=1}^{N_{sb2}}\omega_{sb2}^n|\boldsymbol{R}_{sb2}[\boldsymbol{u}_\theta](\boldsymbol{x}_{sb}^n,t_{sb}^n)|^2. \tag{48f}$$

Here the quadrature points in space-time constitute the data sets $\mathcal{S}_{int}=\{(\boldsymbol{x}_{int}^n,t_{int}^n)\}_{n=1}^{N_{int}}$, $\mathcal{S}_{tb}=\{\boldsymbol{x}_{tb}^n)\}_{n=1}^{N_{tb}}$, $\mathcal{S}_{sb1}=\{(\boldsymbol{x}_{sb1}^n,t_{sb1}^n)\}_{n=1}^{N_{sb1}}$ and $\mathcal{S}_{sb2}=\{(\boldsymbol{x}_{sb2}^n,t_{sb2}^n)\}_{n=1}^{N_{sb2}}$. $\omega_\star^n$ denote the suitable quadrature weights with $\star$ being *int*, *tb*, *sb*1 and *sb*2.

5.3. Error Analysis

Subtracting the elastodynamic equations (42) from the residual equations (44), we obtain

$$\boldsymbol{R}_{int1}=\hat{\boldsymbol{u}}_t-\hat{\boldsymbol{v}}, \tag{49a}$$

$$\boldsymbol{R}_{int2}=\rho\hat{\boldsymbol{v}}_t-2\mu\nabla\cdot(\underline{\boldsymbol{\varepsilon}}(\hat{\boldsymbol{u}}))-\lambda\nabla(\nabla\cdot\hat{\boldsymbol{u}}), \tag{49b}$$

$$\boldsymbol{R}_{tb1}=\hat{\boldsymbol{u}}|_{t=0},\quad \boldsymbol{R}_{tb2}=\hat{\boldsymbol{v}}|_{t=0},\quad \boldsymbol{R}_{sb1}=\hat{\boldsymbol{v}}|_{\Gamma_D},\quad \boldsymbol{R}_{sb2}=(2\mu\underline{\boldsymbol{\varepsilon}}(\hat{\boldsymbol{u}})\boldsymbol{n}+\lambda(\nabla\cdot\hat{\boldsymbol{u}})\boldsymbol{n})|_{\Gamma_N}. \tag{49c}$$

The PINN approximation results are summarized in the following three theorems.

Theorem 5.3. *Let $n \geq 2$, d, r, $k \in \mathbb{N}$ with $k \geq 3$. Let $\boldsymbol{\psi}_1 \in H^r(D)$, $\boldsymbol{\psi}_2 \in H^{r-1}(D)$ and $\boldsymbol{f} \in H^{r-1}(D\times[0,T])$ with $r > \frac{d}{2} + k$. For every integer $N > 5$, there exist* tanh *neural networks $(\boldsymbol{u}_j)_\theta$ and $(\boldsymbol{v}_j)_\theta$, with $j = 1, 2, \cdots, d$, each with two hidden layers, of widths at most $3\lceil\frac{k+n-2}{2}\rceil|P_{k-1,d+2}| + \lceil NT\rceil + d(N-1)$ and $3\lceil\frac{d+n+1}{2}\rceil|P_{d+2,d+2}|\lceil NT\rceil N^d$, such that*

$$\|\boldsymbol{R}_{int1}\|_{L^2(\Omega)}, \|\boldsymbol{R}_{tb1}\|_{L^2(\Omega)} \lesssim \ln N N^{-k+1}, \tag{50a}$$

$$\|\boldsymbol{R}_{int2}\|_{L^2(\Omega)}, \|\underline{\boldsymbol{\varepsilon}}(\boldsymbol{R}_{int1})\|_{L^2(\Omega)}, \|\nabla\cdot\boldsymbol{R}_{int1}\|_{L^2(\Omega)} \lesssim \ln^2 N N^{-k+2}, \tag{50b}$$

$$\|\underline{\boldsymbol{\varepsilon}}(\boldsymbol{R}_{tb1})\|_{L^2(D)}, \|\nabla\cdot\boldsymbol{R}_{tb1}\|_{L^2(D)}, \|\boldsymbol{R}_{sb2}\|_{L^2(\Gamma_N\times[0,T])} \lesssim \ln^2 N N^{-k+2}, \tag{50c}$$

$$\|\boldsymbol{R}_{tb2}\|_{L^2(D)}, \|\boldsymbol{R}_{sb1}\|_{L^2(\Gamma_D\times[0,T])} \lesssim \ln N N^{-k+2}. \tag{50d}$$

Proof. Similar to the Theorem 3.3, we can complete the proof by applying Lemma 5.2 and Lemma 8.3. □

It follows from Theorem 5.3 that, by choosing a sufficiently large N, one can make the PINN residuals in (44), and thus the generalization error $\mathcal{E}_G(\theta)^2$ in (45), arbitrarily small.

Theorem 5.4. *Let $d \in \mathbb{N}$, $\boldsymbol{u} \in C^1(\Omega)$ and $\boldsymbol{v} \in C(\Omega)$ be the classical solution to the linear elastodynamic equation (42). Let $(\boldsymbol{u}_\theta, \boldsymbol{v}_\theta)$ denote the PINN approximation with the parameter θ. Then the following relation holds,*

$$\int_\Omega (|\hat{\boldsymbol{u}}(\boldsymbol{x},t)|^2 + 2\mu|\underline{\boldsymbol{\varepsilon}}(\hat{\boldsymbol{u}}(\boldsymbol{x},t))|^2 + \lambda|\nabla\cdot\hat{\boldsymbol{u}}(\boldsymbol{x},t)|^2 + \rho|\hat{\boldsymbol{v}}(\boldsymbol{x},t)|^2)\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t \leq C_G T\exp\left((2+2\mu+\lambda)T\right),$$

where C_G is given by (79) in Appendix 8.2.

The proof of this theorem is provided in the Appendix 8.2. Theorem 5.4 shows that the total error of the PINN approximation $\mathcal{E}(\theta)^2$ can be controlled by the generalization error $\mathcal{E}_G(\theta)^2$.

Theorem 5.5. *Let $d \in \mathbb{N}$, $\boldsymbol{u} \in C^4(\Omega)$ and $\boldsymbol{v} \in C^3(\Omega)$ be the classical solution to the linear elastodynamic equation (42). Let $(\boldsymbol{u}_\theta, \boldsymbol{v}_\theta)$ denote the PINN approximation with the parameter θ. Then the following relation holds,*

$$\begin{aligned}\int_\Omega (|\hat{\boldsymbol{u}}(\boldsymbol{x},t)|^2 + 2\mu|\underline{\boldsymbol{\varepsilon}}(\hat{\boldsymbol{u}}(\boldsymbol{x},t))|^2 + \lambda|\nabla\cdot\hat{\boldsymbol{u}}(\boldsymbol{x},t)|^2 + \rho|\hat{\boldsymbol{v}}(\boldsymbol{x},t)|^2)\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t &\leq C_T T\exp\left((2+2\mu+\lambda)T\right)\\ &= \mathcal{O}(\mathcal{E}_T(\theta)^2 + M_{int}^{-\frac{2}{d+1}} + M_{tb}^{-\frac{2}{d}} + M_{sb}^{-\frac{1}{d}}),\end{aligned} \tag{51}$$

where

$$\begin{aligned}C_T =& C_{(\boldsymbol{R}_{tb1}^2)}M_{tb}^{-\frac{2}{d}} + \mathcal{Q}_{M_{tb}}^D(\boldsymbol{R}_{tb1}^2) + \rho\left(C_{(\boldsymbol{R}_{tb2}^2)}M_{tb}^{-\frac{2}{d}} + \mathcal{Q}_{M_{tb}}^D(\boldsymbol{R}_{tb2}^2)\right) + 2\mu\left(C_{(|\underline{\boldsymbol{\varepsilon}}(\boldsymbol{R}_{tb1})|^2)}M_{tb}^{-\frac{2}{d}} + \mathcal{Q}_{M_{tb}}^D(|\underline{\boldsymbol{\varepsilon}}(\boldsymbol{R}_{tb1})|^2)\right)\\ &+ \lambda\left(C_{(|\nabla\cdot\boldsymbol{R}_{tb1}|^2)}M_{tb}^{-\frac{2}{d}} + \mathcal{Q}_{M_{tb}}^D(|\nabla\cdot\boldsymbol{R}_{tb1}|^2)\right) + C_{(\boldsymbol{R}_{int1}^2)}M_{int}^{-\frac{2}{d+1}} + \mathcal{Q}_{M_{int}}^\Omega(\boldsymbol{R}_{int1}^2)\\ &+ C_{(\boldsymbol{R}_{int2}^2)}M_{int}^{-\frac{2}{d+1}} + \mathcal{Q}_{M_{int}}^\Omega(\boldsymbol{R}_{int2}^2) + 2\mu\left(C_{(|\underline{\boldsymbol{\varepsilon}}(\boldsymbol{R}_{int1})|^2)}M_{int}^{-\frac{2}{d+1}} + \mathcal{Q}_{M_{int}}^\Omega(|\underline{\boldsymbol{\varepsilon}}(\boldsymbol{R}_{int1})|^2)\right)\\ &+ \lambda\left(C_{(|\nabla\cdot\boldsymbol{R}_{int1}|^2)}M_{int}^{-\frac{2}{d+1}} + \mathcal{Q}_{M_{int}}^\Omega(|\nabla\cdot\boldsymbol{R}_{int1}|^2)\right) + 2|T|^{\frac{1}{2}}C_{\Gamma_D}\left(C_{(\boldsymbol{R}_{sb1}^2)}M_{sb1}^{-\frac{2}{d}} + \mathcal{Q}_{M_{sb1}}^{\Omega_D}(\boldsymbol{R}_{sb1}^2)\right)^{\frac{1}{2}}\\ &+ 2|T|^{\frac{1}{2}}C_{\Gamma_N}\left(C_{(\boldsymbol{R}_{sb2}^2)}M_{sb2}^{-\frac{2}{d}} + \mathcal{Q}_{M_{sb2}}^{\Omega_N}(\boldsymbol{R}_{sb2}^2)\right)^{\frac{1}{2}}.\end{aligned}$$

Proof. Similar to Theorem 3.5, we finish the proof by using Theorem 5.4 and the quadrature error formula (14). The boundedness of the constants $C(\boldsymbol{R}_q^2)$ can be obtained from Lemma 8.2, $\boldsymbol{u} \in C^4(\Omega)$ and $\boldsymbol{v} \in C^3(\Omega)$, where $\boldsymbol{R}_q = \boldsymbol{R}_{tb1}$, $\boldsymbol{R}_{tb2}$, $\underline{\boldsymbol{\varepsilon}}(\boldsymbol{R}_{tb1})$, $\nabla\cdot\boldsymbol{R}_{tb1}$, $\boldsymbol{R}_{int1}$, $\boldsymbol{R}_{int2}$, $\underline{\boldsymbol{\varepsilon}}(\boldsymbol{R}_{int1})$, $\nabla\cdot\boldsymbol{R}_{int1}$, $\boldsymbol{R}_{sb1}$ and $\boldsymbol{R}_{sb2}$. □

Theorem 5.5 shows that the PINN approximation error $\mathcal{E}(\theta)^2$ can be controlled by the training error $\mathcal{E}_T(\theta,\mathcal{S})^2$ with a large enough sample set $\mathcal{S}$.

6. Numerical Examples

The analyses from Sections 3 to 5 suggest several forms for the PINN loss function with the wave, the nonlinear Klein-Gordon, and the linear elastodynamic equations. These forms contain certain non-standard terms, which would be absent from the canonical PINN formulation of the loss function (see Remark 2.2). The presence of such terms is crucial to bounding the PINN approximation errors, as shown in the previous sections.

These non-standard forms of the loss function lead to a variant PINN algorithm. In this section we illustrate the performance of the variant PINN algorithm suggested by the theoretical analysis and the more standard PINN algorithm using numerical examples in one spatial dimension (1D) plus time for the wave equation and the Sine-Gordon equation (i.e. by using $g(u) = \sin(u)$ in the nonlinear Klein-Gordon equation), and in two spatial dimensions (2D) plus time for the linear elastodynamic equation.

Here are some common settings to the numerical simulations in this section. Let $(\boldsymbol{x}, t) \in D \times [0, T]$ denote the spatial and temporal coordinates in the spatial-temporal domain, where $\boldsymbol{x} = x$ and $\boldsymbol{x} = (x, y)$ for 1D and 2D, respectively. For the wave equation and the Sine-Gordon equation, the neural networks contain two input nodes (representing x and t), two hidden layers (number of nodes to be specified below), and two output nodes (representing the solution u and its time derivative $v = \frac{\partial u}{\partial t}$). For the linear elastodynamic equation, three input nodes and four output nodes are employed in the neural network, as will be explained in more detail later. We employ the tanh (hyperbolic tangent) activation function for all the hidden nodes, and no activation function is applied to the output nodes (i.e. linear). For training the neural networks, we employ N collocation points within the spatial-temporal domain drawn from a uniform random distribution, and also N uniform random points on each spatial boundary and on the initial boundary. In the simulations N is varied systematically between 1000 and 3000. After the neural networks are trained, for the wave and Sine-Gordon equations, we compare the PINN solution and the exact solution on a set of $N_{ev} = 3000 \times 3000$ uniform grid points (evaluation points) $(x, t)_n \in D \times [0, T]$ $(n = 1, \cdots, N_{ev})$ that covers the problem domain and the boundaries. For the elastodynamic equation, we compare the PINN solution and the exact solution at different time instants, and at each time instant the corresponding solutions are evaluated at a uniform set of $N_{ev} = 1500 \times 1500$ grid points in the spatial domain, $\boldsymbol{x}_n = (x, y)_n \in D$ $(n = 1, \cdots, N_{ev})$.

The PINN errors reported below are computed as follows. Let $z_n = (\boldsymbol{x}, t)_n$ $((\boldsymbol{x}, t)_n \in D \times [0, T], n = 1, \cdots, N_{ev})$ denote the set of uniform grid points, where N_{ev} denote the number of evaluation points. The errors of PINN are defined by,

$$l_2\text{-error} = \frac{\sqrt{\sum_{n=1}^{N_{ev}} |u(z_n) - u_\theta(z_n)|^2}}{\sqrt{\sum_{n=1}^{N_{ev}} u(z_n)^2}}, \quad l_\infty\text{-error} = \frac{\max\{|u(z_n) - u_\theta(z_n)|\}_{n=1}^{N_{ev}}}{\sqrt{\left(\sum_{n=1}^{N_{ev}} u(z_n)^2\right)/N_{ev}}}, \tag{52a}$$

where u_θ denotes the PINN solution and u denotes the exact solution.

Our implementation of the PINN algorithm is based on the PyTorch library (pytorch.org). We combine the Adam [33] and the L-BFGS [43] optimizers (in batch mode) to train the neural networks. We first employ Adam to train the network for 100 epochs/iterations, and then employ L-BFGS to continue the network training for another 30000 iterations. We employ the default parameter values in Adam, with the learning rate 0.001, $\beta_1 = 0.9$ and $\beta_2 = 0.99$. The initial learning rate 1.0 is adopted in the L-BFGS optimizer.

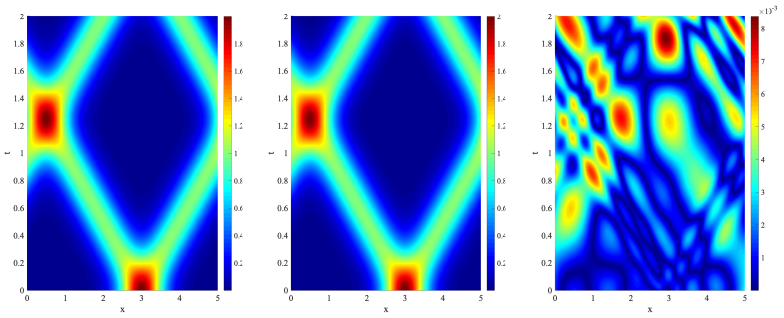
6.1. Wave Equation

We next test the PINN algorithm for solving the wave equation (15) in one spatial dimension (plus time), under a configuration in accordance with that of [17]. Consider the spatial-temporal domain, $(x, t) \in D \times [0, T] = [0, 5] \times [0, 2]$, and the initial-boundary value problem with the wave equation on this domain,

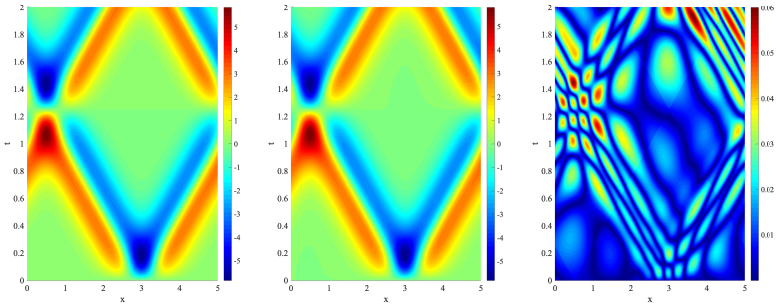
$$\frac{\partial^2 u}{\partial t^2} - c^2 \frac{\partial^2 u}{\partial x^2} = 0, \tag{53a}$$

$$u(0, t) = u(5, t), \quad \frac{\partial u}{\partial x}(0, t) = \frac{\partial u}{\partial x}(5, t), \quad u(x, 0) = 2\,\mathrm{sech}^3\left(\frac{3}{\delta_0}(x - x_0)\right), \quad \frac{\partial u}{\partial t}(x, 0) = 0, \tag{53b}$$

where $u(x, t)$ is the wave field to be solved for, c is the wave speed, x_0 is the initial peak location of the wave, δ_0 is a constant that controls the width of the wave profile, and the periodic boundary conditions are



(a) True solution for u (b) PINN solution u_θ (c) PINN u absolute error



(d) True solution for v (e) PINN solution v_θ (f) PINN v absolute error

Figure 1: Wave equation: Distributions of the True solutions, the PINN solutions and the PINN point-wise absolute errors for u and v in the spatial-temporal domain. $N = 2000$ training points within the domain and on each of the domain boundaries.

imposed on $x = 0$ and 5. In the simulations, we employ $c = 2$, $\delta_0 = 2$, and $x_0 = 3$. Then the above problem has the solution,

$$\begin{cases} u(x,t) = \operatorname{sech}^3\left(\dfrac{3}{\delta_0}\left(-2.5+\xi\right)\right) + \operatorname{sech}^3\left(\dfrac{3}{\delta_0}\left(-2.5+\eta\right)\right), \\ \xi = \operatorname{mod}\left(x - x_0 + ct + 2.5, 5\right), \quad \eta = \operatorname{mod}\left(x - x_0 - ct + 2.5, 5\right), \end{cases}$$

where mod refers to the modulo operation. We reformulate the problem (53) into the following system,

$$u_t - v = 0, \qquad v_t - c^2 u_{xx} = 0, \tag{54a}$$

$$u(0,t) = u(5,t), \quad u_x(0,t) = u_x(5,t), \quad u(x,0) = 2\operatorname{sech}^3\left(\frac{3}{\delta_0}(x-x_0)\right), \quad v(x,0) = 0, \tag{54b}$$

where $v(x,t)$ is an auxiliary field given by the first equation in (54a).

To solve the system (54) with PINN, we employ 90 and 60 neurons in the first and the second hidden layers of neural networks, respectively. We consider the following loss function in PINN,

$$\begin{aligned} \text{Loss} =& \frac{W_1}{N}\sum_{n=1}^{N}\left[u_{\theta t}(x_{int}^n, t_{int}^n) - v_\theta(x_{int}^n, t_{int}^n)\right]^2 + \frac{W_2}{N}\sum_{n=1}^{N}\left[v_{\theta t}(x_{int}^n, t_{int}^n) - u_{\theta xx}(x_{int}^n, t_{int}^n)\right]^2 \\ &+ \frac{W_3}{N}\sum_{n=1}^{N}\left[u_{\theta tx}(x_{int}^n, t_{int}^n) - v_{\theta x}(x_{int}^n, t_{int}^n)\right]^2 + \frac{W_4}{N}\sum_{n=1}^{N}\left[u_\theta(x_{tb}^n, 0) - 2\operatorname{sech}^3\left(\frac{3}{\delta_0}(x_{tb}^n - x_0)\right)\right]^2 \\ &+ \frac{W_5}{N}\sum_{n=1}^{N}\left[v_\theta(x_{tb}^n, 0)\right]^2 + \frac{W_6}{N}\sum_{n=1}^{N}\left[u_{\theta x}(x_{tb}^n, 0) + \frac{18\sinh((3x_{tb}^n - 3x_0)/\delta_0)}{\delta_0\cosh^4((3x_{tb}^n - 3x_0)/\delta_0)}\right]^2 \\ &+ \frac{W_7}{N}\sum_{n=1}^{N}\left[v_\theta(0, t_{sb}^n) - v_\theta(5, t_{sb}^n)\right]^2 + \frac{W_8}{N}\sum_{n=1}^{N}\left[u_{\theta x}(0, t_{sb}^n) - u_{\theta x}(5, t_{sb}^n)\right]^2 \end{aligned}$$

$$=: \sum_{i=1}^{8} \mathfrak{W}_i, \tag{55}$$

where $\mathfrak{W}_i$ $(1 \le i \le 8)$ denote the different terms in the loss expression. Note that in the simulations we have employed the same number of collocation points (N) within the domain and on each of the domain boundaries. The formulation of the loss function here differs from what is used in the error analysis in several aspects. First, we have added a set of penalty coefficients $W_n > 0$ $(1 \le n \le 8)$ for different loss terms in numerical simulations. Second, the collocation points used in simulations (e.g. x_{int}^n, t_{int}^n, x_{sb}^n, t_{sb}^n, x_{tb}^n) are generated randomly within the domain or on the domain boundaries from a uniform distribution. In addition, the averaging used here do not exactly correspond to the numerical quadrature rule (mid-point rule) used in the theoretical analysis.

We also consider the following form for the loss function, as given in (18),

$$\text{Loss} = \sum_{i=1}^{6} \mathfrak{W}_i + W_7 \left(\frac{1}{N} \sum_{n=1}^{N} [v_\theta(0, t_{sb}^n) - v_\theta(5, t_{sb}^n)]^2 \right)^{1/2} + W_8 \left(\frac{1}{N} \sum_{n=1}^{N} [u_{\theta x}(0, t_{sb}^n) - u_{\theta x}(5, t_{sb}^n)]^2 \right)^{1/2}. \tag{56}$$

The difference between this form and the form (55) lies in the last two terms, with the terms here containing a square root.

It should be noted that the loss function defined by (56) is in accordance with our theoretical analysis, while the one in (55) is akin to the more standard PINN formulation. The loss function (55) will be referred to as the loss form #1 in subsequent discussions, and (56) will be referred to as the loss form #2. The PINN schemes that employ these two different loss forms will be referred to as PINN-F1 and PINN-F2, respectively.

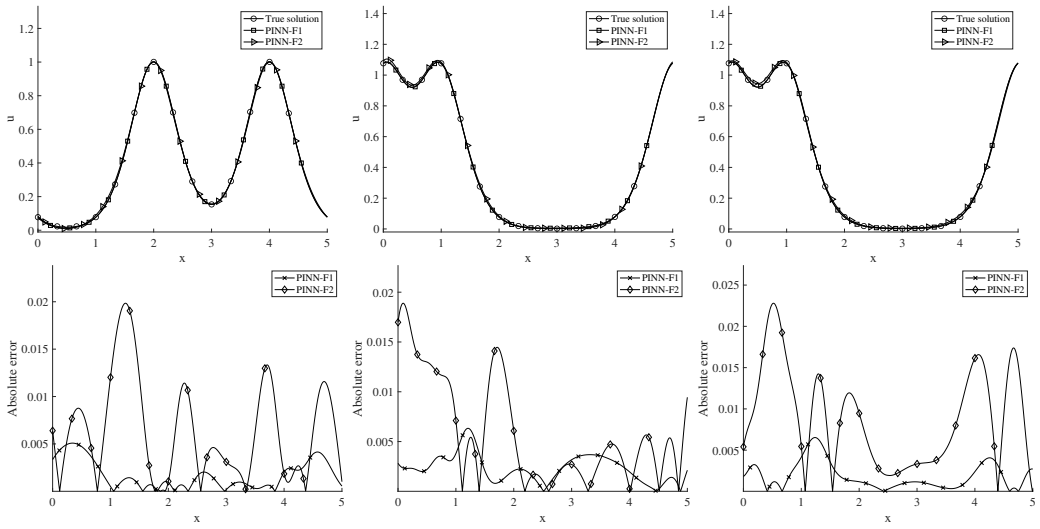
Figure 1 shows distributions of the exact solutions, the PINN solutions, and the PINN point-wise absolute errors for u and $v = \frac{\partial u}{\partial t}$ in the spatial-temporal domain. Here the PINN solution is computed by PINN-F1, in which penalty coefficients are given by $\boldsymbol{W} = (W_1, \ldots, W_8) = (0.8, 0.8, 0.8, 0.5, 0.5, 0.5, 0.9, 0.9)$. One can observe that the method has captured the wave fields for u and v reasonably well, with the error for u notably smaller than that of v.

Figures 2 and 3 provide a comparison of the solutions obtained using the two forms of loss functions. Figure 2 compares profiles of the PINN-F1 and PINN-F2 solutions, and the exact solution, for u (top row) at three time instants ($t = 0.5$, 1.0, and 1.5), as well as the error profiles (bottom row). Figure 3 shows the corresponding results for the field variable v. These results are obtained by using $N = 2000$ training data points in the domain and on each of the domain boundaries. It is observed that both PINN schemes, with the loss functions given by (55) and (56) respectively, have captured the solution reasonably well. We further observe that the PINN-F2 scheme (with the loss form (56)) produces somewhat less accurate results than the PINN-F1 (with loss form (55)), especially for the field v.

We have varied the number of training data points N systematically and studied its effect on the PINN results. Figure 4 shows the loss histories of PINN-F1 and PINN-F2 corresponding to different number of training data points (N) in the simulations, with a total of $30,000$ training iterations. We can make two observations. First, the history curves with the loss function form #1 seem smoother, while fluctuations in the loss history can be observed with the form #2. Second, the eventual loss values produced by the loss form #1 are smaller than those produced by the loss form #2.

Table 1 is another comparison between PINN-F1 and PINN-F2. Here the l_2 and l_∞ errors of u and v computed by PINN-F1 and PINN-F2 corresponding to different training data points (N) have been listed. There appears to be a general trend that the errors tend to decrease with increasing number of training points, but the decrease is not monotonic. It can be observed that the u errors are notably smaller than those for v, as signified earlier in e.g. Figure 1. One again observes that PINN-F1 results appear more accurate than those of PINN-F2 for the wave equation.

Theorem 3.5 suggests the solution errors for u, v, and ∇u approximately scale as the square root of the training loss function. Figure 5 provides some numerical evidence for this point. Here we plot the l^2 errors for u, $\frac{\partial u}{\partial t}$ and $\frac{\partial u}{\partial x}$ from our simulations as a function of the training loss value for PINN-F1 and PINN-F2 in logarithmic scales. It is evident that for PINN-F1 the scaling essentially follows the square root relation. For PINN-F2 the relation between the error and the training loss appears to scale with a power somewhat larger than $\frac{1}{2}$.



(a) $t = 0.5$ (b) $t = 1$ (c) $t = 1.5$

Figure 2: Wave equation: Comparison of profiles of u (top row) and its absolute error (bottom row) between the PINN solutions (loss forms #1 and #2) and the exact solution at time instants (a) $t = 0.5$, (b) $t = 1.0$, and (c) $t = 1.5$. $N = 2000$ training data points within the domain and on each of the domain boundaries ($x = 0$ and 5, and $t = 0$).

Table 1: Wave equation: The u and v errors versus the number of training data points N.

N	l_2-error				l_∞-error			
	PINN-F1		PINN-F2		PINN-F1		PINN-F2	
	u_θ	v_θ	u_θ	v_θ	u_θ	v_θ	u_θ	v_θ
1000	5.7013e-03	1.3531e-02	4.7281e-02	9.2431e-02	1.8821e-02	4.6631e-02	1.4367e-01	3.2764e-01
1500	2.1689e-03	4.1035e-03	4.9087e-02	1.2438e-01	6.7631e-03	1.5109e-02	2.1525e-01	5.0601e-01
2000	4.6896e-03	9.6417e-03	1.8554e-02	4.9224e-02	1.3828e-02	3.3063e-02	6.0780e-02	1.6358e-01
2500	3.7879e-03	9.8574e-03	2.3526e-02	5.4266e-02	1.2868e-02	3.3622e-02	9.8690e-02	1.9467e-01
3000	2.6588e-03	6.0746e-03	1.4164e-02	3.7796e-02	8.1457e-03	1.9860e-02	5.3045e-02	1.4179e-01

6.2. Sine-Gordon Equation

We test the PINN algorithm suggested by the theoretical analysis for the Sine-Gordon equation (i.e. by setting $g(u) = \sin(u)$ in (31)) in this subsection. Consider the spatial-temporal domain $(x,t) \in \Omega = D \times [0,T] = [0,1] \times [0,2]$, and the following initial/boundary value problem on this domain,

$$\frac{\partial^2 u}{\partial t^2} - \frac{\partial^2 u}{\partial x^2} + u + \sin(u) = f(x,t), \tag{57a}$$

$$u(0,t) = \phi_1(t), \quad u(1,t) = \phi_2(t), \quad u(x,0) = \psi_1(x), \quad \frac{\partial u}{\partial t}(x,0) = \psi_2(x). \tag{57b}$$

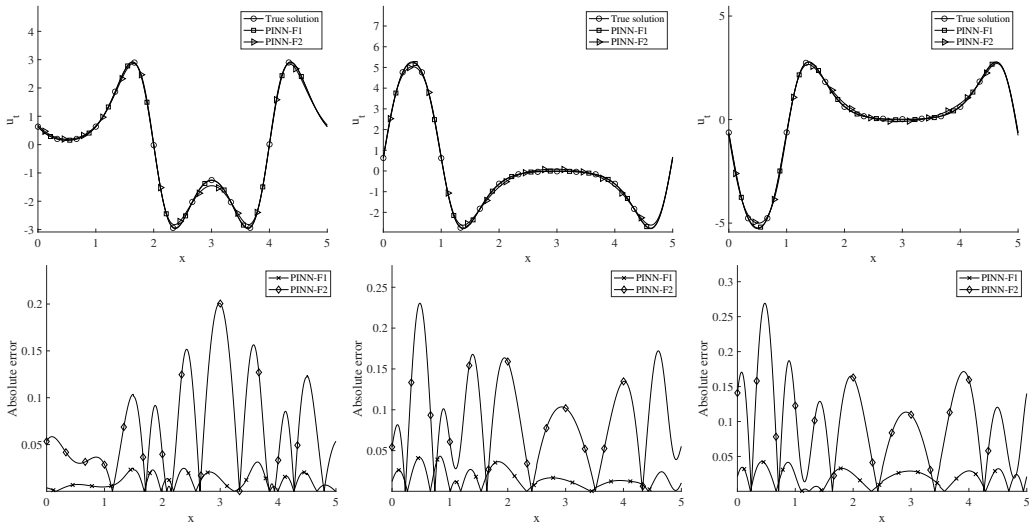
In these equations, $u(x,t)$ is the field function to be solved for, $f(x,t)$ is a source term, ψ_1 and ψ_2 are the initial conditions, and ϕ_1 and ϕ_2 are the boundary conditions. The source term, initial and boundary conditions appropriately are chosen by the following exact solution,

$$u(x,t) = \left[2\cos\left(\pi x + \frac{\pi}{5}\right) + \frac{9}{5}\cos\left(2\pi x + \frac{7\pi}{20}\right)\right]\left[2\cos\left(\pi t + \frac{\pi}{5}\right) + \frac{9}{5}\cos\left(2\pi t + \frac{7\pi}{20}\right)\right]. \tag{58}$$

To simulate this problem with PINN, we reformulate the problem as follows,

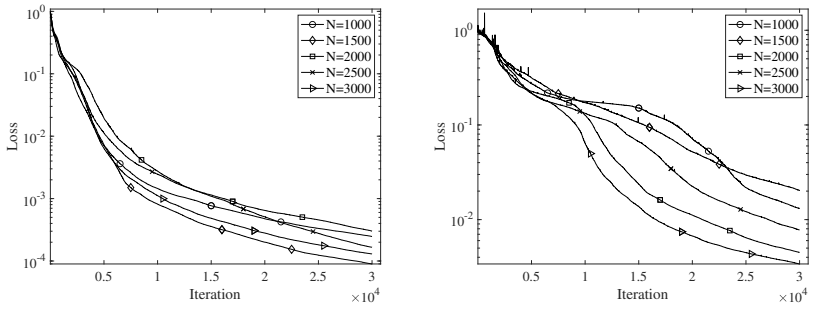
$$u_t - v = 0, \quad v_t - u_{xx} + u + \sin(u) = f(x,t), \tag{59a}$$

$$u(0,t) = \phi_1(t), \quad u(1,t) = \phi_2(t), \quad u(x,0) = \psi_1(x), \quad v(x,0) = \psi_2(x), \tag{59b}$$



(a) $t = 0.5$ (b) $t = 1$ (c) $t = 1.5$

Figure 3: Wave equation: Comparison of the profiles of $v = \frac{\partial u}{\partial t}$ (top row) and its absolute error (bottom row) between the PINN solutions (loss forms #1 and #2) and the exact solution at time instants (a) $t = 0.5$, (b) $t = 1.0$, and (c) $t = 1.5$. $N = 2000$ training data points within the domain and on each of the domain boundaries ($x = 0$ and 5, and $t = 0$).

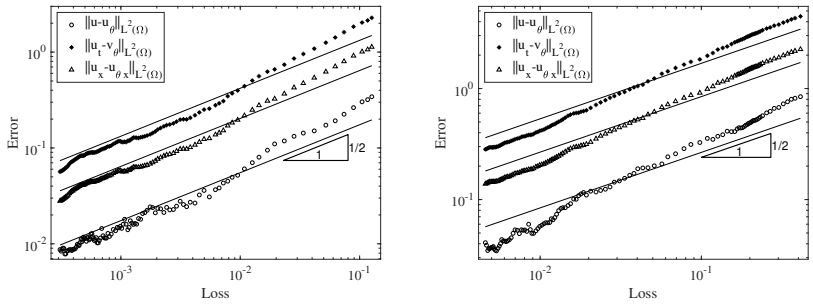


(a) PINN-F1 (b) PINN-F2

Figure 4: Wave equation: Histories of the loss function versus the training iteration with PINN-F1 and PINN-F2, corresponding to different number of training data points (N).

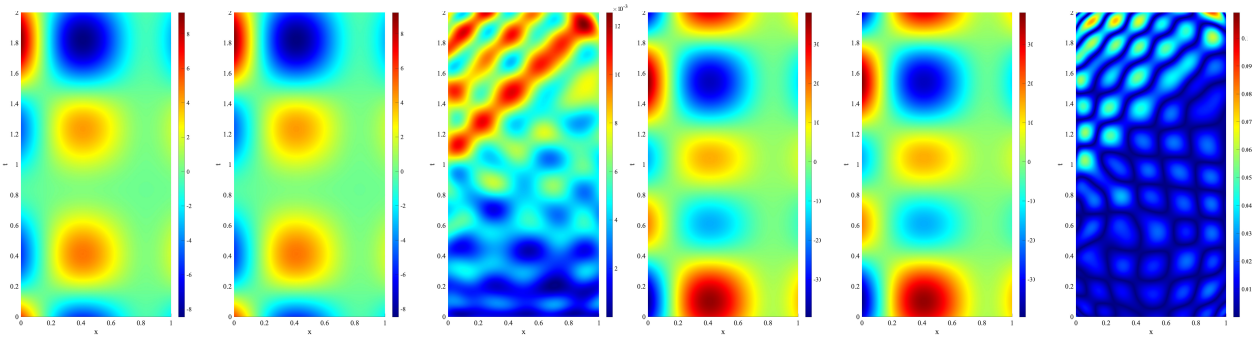
where v is defined by equation (59a). In light of (36), we employ the following loss function in PINN,

$$
\begin{aligned}
\text{Loss} = & \frac{W_1}{N}\sum_{n=1}^{N}\left[u_{\theta t}(x_{int}^n, t_{int}^n) - v_\theta(x_{int}^n, t_{int}^n)\right]^2 \\
& + \frac{W_2}{N}\sum_{n=1}^{N}\left[v_{\theta t}(x_{int}^n, t_{int}^n) - u_{\theta xx}(x_{int}^n, t_{int}^n) + u_\theta(x_{int}^n, t_{int}^n) + \sin(u_\theta(x_{int}^n, t_{int}^n)) - f(x_{int}^n, t_{int}^n)\right]^2 \\
& + \frac{W_3}{N}\sum_{n=1}^{N}\left[u_{\theta tx}(x_{int}^n, t_{int}^n) - v_{\theta x}(x_{int}^n, t_{int}^n)\right]^2 + \frac{W_4}{N}\sum_{n=1}^{N}\left[u_\theta(x_{tb}^n, 0) - \psi_1(x_{tb}^n)\right]^2 \\
& + \frac{W_5}{N}\sum_{n=1}^{N}\left[v_\theta(x_{tb}^n, 0) - \psi_2(x_{tb}^n)\right]^2 + \frac{W_6}{N}\sum_{n=1}^{N}\left[u_{\theta x}(x_{tb}^n, 0) - \psi_{1x}(x_{tb}^n)\right]^2 \\
& + W_7\left(\frac{1}{N}\sum_{n=1}^{N}\left[(v_\theta(0, t_{sb}^n) - \phi_{1t}(t_{sb}^n))^2 + (v_\theta(1, t_{sb}^n) - \phi_{2t}(t_{sb}^n))^2\right]\right)^{1/2}
\end{aligned}
$$



(a) PINN-F1 (b) PINN-F2

Figure 5: Wave equation: The l^2 errors of u, $\frac{\partial u}{\partial t}$, and $\frac{\partial u}{\partial x}$ as a function of the training loss value. $N = 2000$ training data points.



(a) True solution u (b) PINN solution u (c) Solution error u (d) True solution v (e) PINN solution v (f) Solution error v

Figure 6: Sine-Gordon equation: Distributions of the exact solution, the PINN solution and the PINN absolute error for u (left three columns) and for $v = \frac{\partial u}{\partial t}$ (right three columns). $N = 2000$ collocation points within the domain and on the domain boundaries.

$$=: \sum_{i=1}^{7} \mathfrak{S}_i, \tag{60}$$

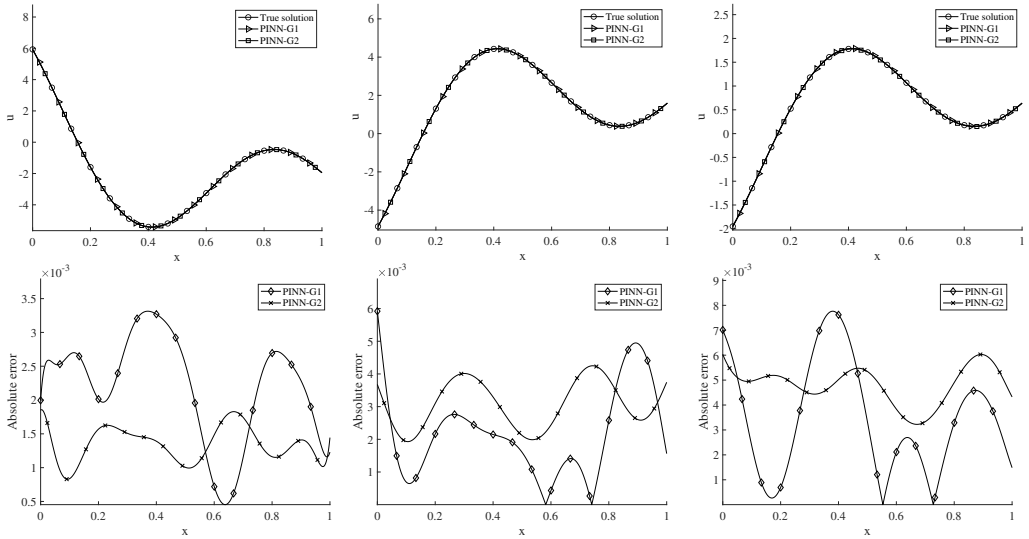
where $\mathfrak{S}_i$ $(1 \le i \le 7)$ represent different terms in the loss expression, and $W_n > 0$ $(1 \le n \le 7)$ are the penalty coefficients for different loss terms added in the PINN implementation. It should be noted that the loss terms $\mathfrak{S}_3$ and $\mathfrak{S}_6$ will be absent from the conventional PINN formulation (see [46]). These terms in the training loss are necessary based on the error analysis in Section 4. It should also be noted that the terms in $\mathfrak{S}_7$ contains a square root, as dictated by the theoretical analysis of Section 4.

We have also implemented a PINN scheme with a variant form for the loss function,

$$\text{Loss} = \sum_{i=1}^{6} \mathfrak{S}_i + \frac{W_7}{N} \sum_{n=1}^{N} \left[(v_\theta(0, t_{sb}^n) - \phi_{1t}(t_{sb}^n))^2 + (v_\theta(1, t_{sb}^n) - \phi_{2t}(t_{sb}^n))^2 \right]. \tag{61}$$

The difference between (61) and (60) lies in the $\mathfrak{S}_7$ terms. These $\mathfrak{S}_7$ terms in (61) are squared, and they are not in (60). We refer to the PINN scheme employing the loss function (60) as PINN-G1 and the scheme employing the loss function (61) as PINN-G2.

In the simulations we employ a feed-forward neural network with two input nodes (representing x and t), two output nodes (representing u and v), and two hidden layers, each having a width of 80 nodes. The tanh activation function has been used for all the hidden nodes. We employ N collocation points generated from a uniform random distribution within the domain, on each of the domain boundary, and also on the initial boundary, where N is varied systematically in the simulations. The penalty coefficients in the loss functions are taken to be $\boldsymbol{W} = (W_1, \ldots, W_7) = (0.5, 0.4, 0.5, 0.6, 0.6, 0.6, 0.8)$.



(a) $t = 0.5$ (b) $t = 1$ (c) $t = 1.5$

Figure 7: Sine-Gordon equation: Top row, comparison of profiles between the exact solution and PINN-G1/PINN-G2 solutions for u at several time instants. Bottom row, profiles of the absolute error of the PINN-G1 and PINN-G2 solutions for u. $N = 2000$ training collocation points.

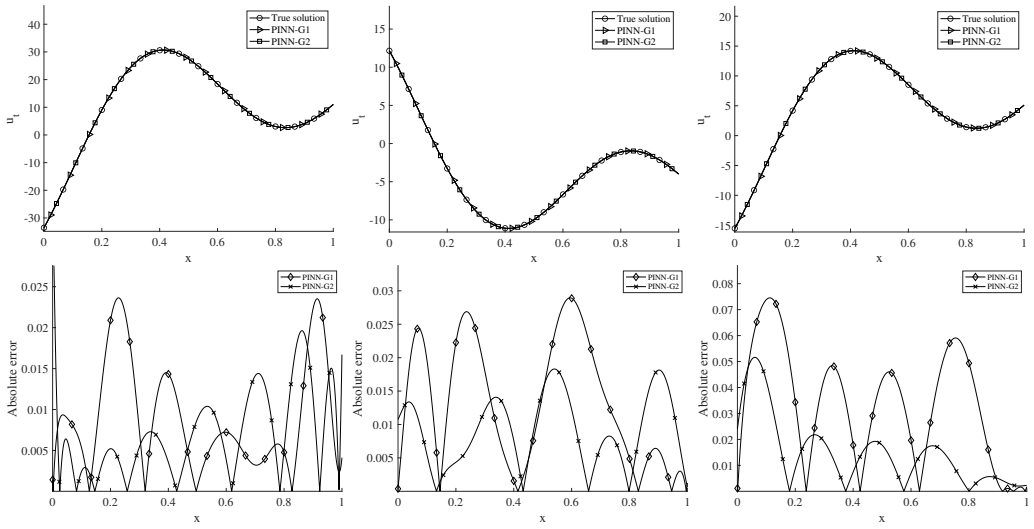
Figure 6 shows distributions of of $u(x,t)$ and $v = \frac{\partial u}{\partial t}$ from the exact solution (left column) and the PINN solution (middle column), as well as the point-wise absolute errors of the PINN solution for these fields (right column). These results are obtained by PINN-G2 with $N = 2000$ random collocation points within the domain and on each of the domain boundaries. The PINN solution is in good agreement with the true solution.

Figures 7 and 8 compare the profiles of u and v between the exact solution, and the solutions obtained by PINN-G1 and PINN-G2, at several time instants ($t = 0.5$, 1 and 1.5). Profiles of the absolute errors of the PINN-G1/PINN-G2 solutions are also shown in these figures. We observe that both PINN-G1 and PINN-G2 have captured the solution for u quite accurately, and to a lesser extent, also for v. Comparison of the error profiles between PINN-G1 and PINN-G2 suggests that the PINN-G2 error in general appears to be somewhat smaller than that of PINN-G1. But this seems not to be true consistently in the entire domain.

The effect of the collocation points on the PINN results has been studied by varying the number of training collocation points systematically between $N = 1000$ and $N = 3000$ within the domain and on each of the domain boundaries. The results are provided in Figure 9 and Table 2. Figure 9 shows histories of the loss function corresponding to different number of collocation points for PINN-G1 and PINN-G2. Table 2 provides the l_2 and l_∞ errors of u and v versus the number of collocation points computed by PINN-G1 and PINN-G2. The PINN errors in general tend to decrease with increasing number of collocation points, but this trend is not monotonic. It can be observed that both PINN-G1 and PINN-G2 have captured the

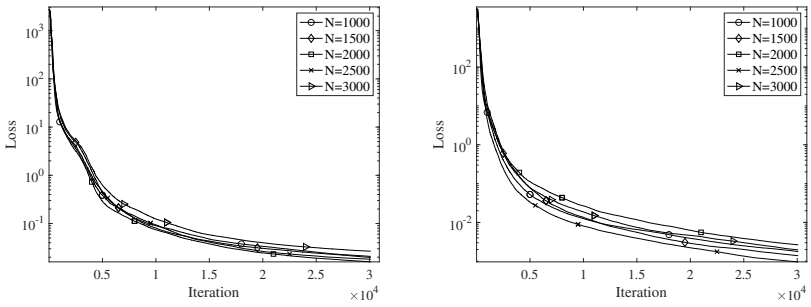
Table 2: Sine-Gordon equation: The l_2 and l_∞ errors for u and v versus the number of training collocation points N corresponding to PINN-G1 and PINN-G2.

N	l_2-error				l_∞-error			
	PINN-G1		PINN-G2		PINN-G1		PINN-G2	
	u_θ	v_θ	u_θ	v_θ	u_θ	v_θ	u_θ	v_θ
1000	3.0818e-03	4.3500e-03	3.0674e-03	2.0581e-03	9.6044e-03	1.8894e-02	7.3413e-03	1.1323e-02
1500	3.4335e-03	4.8035e-03	1.0605e-03	1.4729e-03	1.0566e-02	1.7050e-02	2.2914e-03	6.2831e-03
2000	2.1914e-03	3.0055e-03	2.2469e-03	1.6072e-03	7.5882e-03	1.1099e-02	4.8842e-03	8.8320e-03
2500	3.0172e-03	3.5698e-03	6.6072e-04	6.0509e-04	9.2515e-03	1.4645e-02	1.4099e-03	4.3423e-03
3000	2.5281e-03	4.4858e-03	6.6214e-04	1.0830e-03	7.2785e-03	1.6213e-02	1.9697e-03	7.8866e-03



(a) $t = 0.5$ (b) $t = 1$ (c) $t = 1.5$

Figure 8: Sine-Gordon equation: Top row, comparison of profiles between the exact solution and PINN-G1/PINN-G2 solutions for $v = \frac{\partial u}{\partial t}$ at several time instants. Bottom row, profiles of the absolute error of the PINN-G1 and PINN-G2 solutions for v. $N = 2000$ training collocation points.

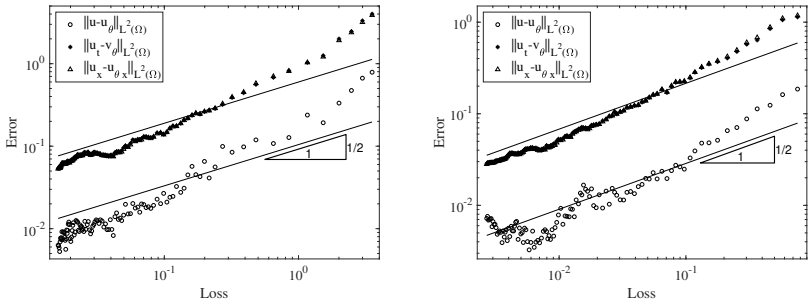


(a) PINN-G1 (b) PINN-G2

Figure 9: Sine-Gordon equation: Loss histories of (a) PINN-G1 and (b) PINN-G2 corresponding to various numbers of training collocation points.

solutions quite accurately, with those errors from PINN-G2 in general slightly better.

Figure 10 provides some numerical evidence for the relation between the total error and the training loss



(a) PINN-G1 (b) PINN-G2

Figure 10: Sine-Gordon equation: The l^2 errors of u, $\frac{\partial u}{\partial t}$, and $\frac{\partial u}{\partial x}$ as a function of the training loss value.

as suggested by Theorem 4.4. Here we plot the l_2 errors for u, $\frac{\partial u}{\partial t}$ and $\frac{\partial u}{\partial x}$ as a function of the training loss value obtained by PINN-G1 and PINN-G2. The results indicate that the total error scales approximately as the square root of the training loss, which in some sense corroborates the error-loss relation as expressed in Theorem 4.4.

6.3. Linear Elastodynamic Equation

In this subsection we look into the linear elastodynamic equation (in two spatial dimensions plus time) and test the PINN algorithm as suggested by the theoretical analysis in Section 5 using this equation. Consider the spatial-temporal domain $(x, y, t) \in \Omega = D \times [0, T] = [0, 1] \times [0, 1] \times [0, 2]$, and the following initial/boundary value problem with the linear elastodynamics equation on Ω:

$$\rho \frac{\partial^2 \boldsymbol{u}}{\partial t^2} - 2\mu \nabla \cdot (\underline{\varepsilon}(\boldsymbol{u})) - \lambda \nabla (\nabla \cdot \boldsymbol{u}) = \boldsymbol{f}(\boldsymbol{x}, t), \tag{62a}$$

$$\boldsymbol{u}|_{\Gamma_d} = \boldsymbol{\phi}_d, \quad \Big(2\mu \underline{\varepsilon}(\boldsymbol{u}) + \lambda (\nabla \cdot \boldsymbol{u})\Big)|_{\Gamma_n} \boldsymbol{n} = \boldsymbol{\phi}_n, \quad \boldsymbol{u}(\boldsymbol{x}, 0) = \boldsymbol{\psi}_1, \quad \frac{\partial \boldsymbol{u}}{\partial t}(\boldsymbol{x}, 0) = \boldsymbol{\psi}_2, \tag{62b}$$

where $\boldsymbol{u} = (u_1(\boldsymbol{x}, t), u_2(\boldsymbol{x}, t))^T$ ($\boldsymbol{x} = (x, y) \in D$, $t \in [0, T]$) is the displacement field to be solved for, $\boldsymbol{f}(\boldsymbol{x}, t)$ is a source term, and ρ, μ and λ are material constants. Γ_d is the Dirichlet boundary and Γ_n is the Neumann boundary, with $\partial D = \Gamma_d \cup \Gamma_n$ and $\Gamma_d \cap \Gamma_n = \emptyset$, where $\boldsymbol{n}$ is the outward-pointing unit normal vector. In our simulations we choose the left boundary ($x = 0$) as the Dirichlet boundary, and the rest are Neumann boundaries. $\boldsymbol{\phi}_d$ and $\boldsymbol{\phi}_n$ are Dirichlet and Neumann boundary conditions, respectively. $\boldsymbol{\psi}_1$ and $\boldsymbol{\psi}_2$ are the initial conditions for the displacement and the velocity. We employ the material parameter values $\mu = \lambda = \rho = 1$, and the following manufactured solution ([1]) to this problem,

$$\boldsymbol{u}(\boldsymbol{x}, t) = \sin(\sqrt{2}\pi t) \begin{bmatrix} -\sin(\pi x)^2 \sin(2\pi y) \\ \sin(2\pi x) \sin(\pi y)^2 \end{bmatrix}. \tag{63}$$

The source term $\boldsymbol{f}(\boldsymbol{x}, t)$, the boundary/initial distributions $\boldsymbol{\phi}_d$, $\boldsymbol{\phi}_n$, $\boldsymbol{\psi}_1$ and $\boldsymbol{\psi}_2$ are chosen by the expression (63).

To simulate this problem using the PINN algorithm suggested by the theoretical analysis from Section 5, we reformulate (62) into the following system

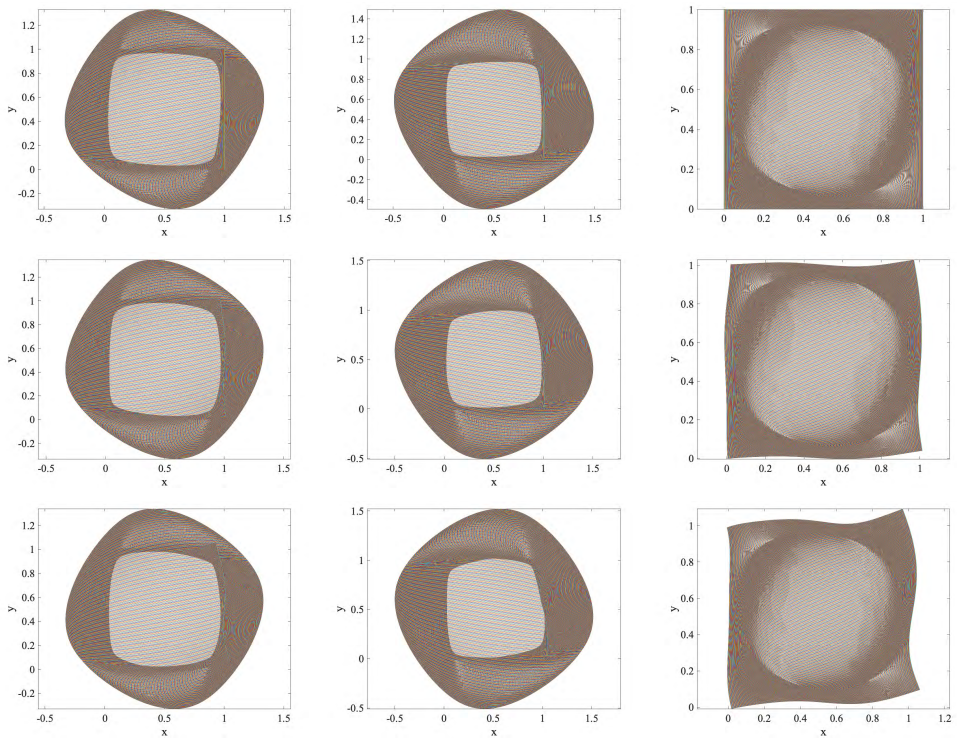
$$\boldsymbol{u}_t - \boldsymbol{v} = \boldsymbol{0}, \qquad \boldsymbol{v}_t - 2\nabla \cdot (\underline{\varepsilon}(\boldsymbol{u})) - \nabla (\nabla \cdot \boldsymbol{u}) = \boldsymbol{f}(\boldsymbol{x}, t), \tag{64a}$$

$$\boldsymbol{u}|_{\Gamma_d} = \boldsymbol{\phi}_d, \quad \Big(2\underline{\varepsilon}(\boldsymbol{u}) + (\nabla \cdot \boldsymbol{u})\Big)|_{\Gamma_n} \boldsymbol{n} = \boldsymbol{\phi}_n, \quad \boldsymbol{u}(\boldsymbol{x}, 0) = \boldsymbol{\psi}_1, \quad \boldsymbol{v}(\boldsymbol{x}, 0) = \boldsymbol{\psi}_2, \tag{64b}$$

where $\boldsymbol{v}(\boldsymbol{x}, t)$ is an intermediate variable (representing the velocity) as given by (64a).

In light of (47), we employ the following loss function for PINN,

$$\begin{aligned}
\text{Loss} = {} & \frac{W_1}{N} \sum_{n=1}^{N} [\boldsymbol{u}_{\theta t}(\boldsymbol{x}_{int}^n, t_{int}^n) - \boldsymbol{v}_\theta(\boldsymbol{x}_{int}^n, t_{int}^n)]^2 \\
& + \frac{W_2}{N} \sum_{n=1}^{N} [\boldsymbol{v}_{\theta t}(\boldsymbol{x}_{int}^n, t_{int}^n) - 2\nabla \cdot (\underline{\varepsilon}(\boldsymbol{u}_\theta(\boldsymbol{x}_{int}^n, t_{int}^n))) - \nabla (\nabla \cdot \boldsymbol{u}_\theta(\boldsymbol{x}_{int}^n, t_{int}^n)) - \boldsymbol{f}(\boldsymbol{x}_{int}^n, t_{int}^n))]^2 \\
& + \frac{W_3}{N} \sum_{n=1}^{N} [\underline{\varepsilon}(\boldsymbol{u}_{\theta t}(\boldsymbol{x}_{int}^n, t_{int}^n) - \boldsymbol{v}_\theta(\boldsymbol{x}_{int}^n, t_{int}^n))]^2 + \frac{W_4}{N} \sum_{n=1}^{N} [\nabla \cdot (\boldsymbol{u}_{\theta t}(\boldsymbol{x}_{int}^n, t_{int}^n) - \boldsymbol{v}_\theta(\boldsymbol{x}_{int}^n, t_{int}^n))]^2 \\
& + \frac{W_5}{N} \sum_{n=1}^{N} [\boldsymbol{u}_\theta(\boldsymbol{x}_{tb}^n, 0) - \boldsymbol{\psi}_1(\boldsymbol{x}_{tb}^n)]^2 + \frac{W_6}{N} \sum_{n=1}^{N} [\boldsymbol{v}_\theta(\boldsymbol{x}_{tb}^n, 0) - \boldsymbol{\psi}_2(\boldsymbol{x}_{tb}^n)]^2 \\
& + \frac{W_7}{N} \sum_{n=1}^{N} [\underline{\varepsilon}(\boldsymbol{u}_\theta(\boldsymbol{x}_{tb}^n, 0) - \boldsymbol{\psi}_1(\boldsymbol{x}_{tb}^n))]^2 + \frac{W_8}{N} \sum_{n=1}^{N} [\nabla \cdot (\boldsymbol{u}_\theta(\boldsymbol{x}_{tb}^n, 0) - \boldsymbol{\psi}_1(\boldsymbol{x}_{tb}^n))]^2 \\
& + W_9 \left(\frac{1}{N} \sum_{n=1}^{N} [\boldsymbol{v}_\theta(\boldsymbol{x}_{sb1}^n, t_{sb1}^n) - \boldsymbol{\phi}_{dt}(\boldsymbol{x}_{sb1}^n, t_{sb1}^n)]^2 \right)^{1/2}
\end{aligned}$$



(a) $t = 0.5$ (b) $t = 1$ (c) $t = 1.5$

Figure 11: Linear elastodynamic equation: Visualization of the deformed configuration at time instants (a) $t = 0.5$, (b) $t = 1.0$, and (c) $t = 1.5$ from the exact solution (top row), the PINN-H1 solution (middle row) and the PINN-H2 solution (bottom row). Plotted here are the deformed field, $\boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x}, t)$, for a set of grid points $\boldsymbol{x} \in D = [0, 1] \times [0, 1]$. $N = 2000$ training collocation points within domain and on the domain boundaries.

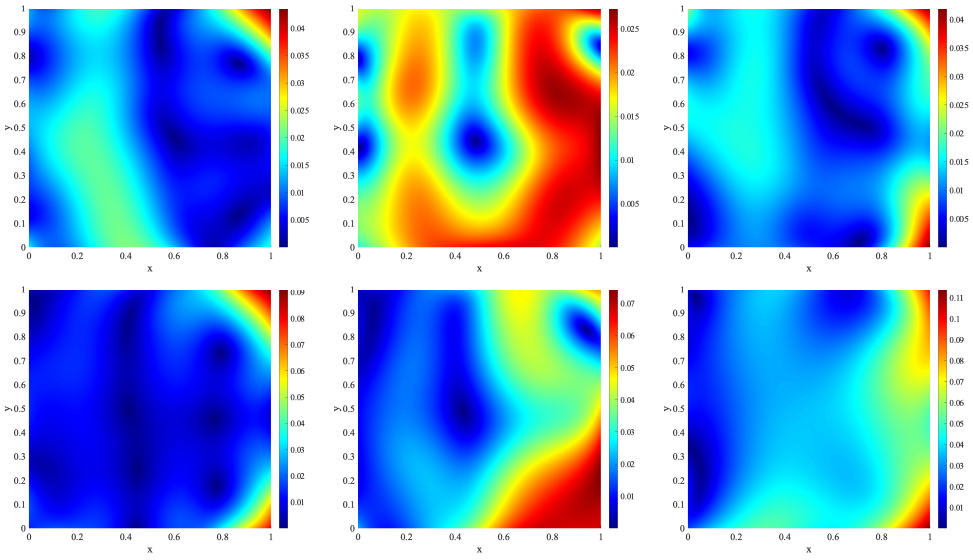
$$+ W_{10} \left(\frac{1}{N} \sum_{n=1}^{N} \left[2\underline{\varepsilon}(\boldsymbol{u}_\theta(\boldsymbol{x}^n_{sb2}, t^n_{sb2}))\boldsymbol{n} + (\nabla \cdot \boldsymbol{u}_\theta(\boldsymbol{x}^n_{sb2}, t^n_{sb2}))\boldsymbol{n} - \boldsymbol{\phi}_n(\boldsymbol{x}^n_{sb2}, t^n_{sb2}) \right]^2 \right)^{1/2}$$
$$=: \sum_{i=1}^{10} \mathfrak{E}_i, \tag{65}$$

where we have added the penalty coefficients, $W_n > 0$ $(1 \le n \le 10)$, for different loss terms in the implementation, and N denotes the number of collocation points within the domain and on the domain boundaries. In the numerical tests we have also implemented another form for the loss function as follows,

$$\text{Loss} = \sum_{i=1}^{8} \mathfrak{E}_i + \frac{W_9}{N} \sum_{n=1}^{N} \left[\boldsymbol{v}_\theta(\boldsymbol{x}^n_{sb1}, t^n_{sb1}) - \boldsymbol{\phi}_{dt}(\boldsymbol{x}^n_{sb1}, t^n_{sb1}) \right]^2$$
$$+ \frac{W_{10}}{N} \sum_{n=1}^{N} \left[2\underline{\varepsilon}(\boldsymbol{u}_\theta(\boldsymbol{x}^n_{sb2}, t^n_{sb2}))\boldsymbol{n} + (\nabla \cdot \boldsymbol{u}_\theta(\boldsymbol{x}^n_{sb2}, t^n_{sb2}))\boldsymbol{n} - \boldsymbol{\phi}_n(\boldsymbol{x}^n_{sb2}, t^n_{sb2}) \right]^2. \tag{66}$$

The difference between these two forms for the loss function lies in the $\mathfrak{E}_9$ and $\mathfrak{E}_{10}$ terms. It should be noted that the $\mathfrak{E}_9$ and $\mathfrak{E}_{10}$ terms in (65) contain a square root, in light of the error terms (48a)−(48f) from the theoretical analysis. In contrast, these terms have no square root in (66). The PINN scheme utilizing the loss function (65) is henceforth referred to as PINN-H1, and the scheme that employs the loss function (66) shall be referred to as PINN-H2.

In the simulations, we employ a feed-forward neural network with three input nodes, which represent $\boldsymbol{x} = (x, y)$ and the time variable t, and four output nodes, which represent $\boldsymbol{u} = (u_1, u_2)$ and $\boldsymbol{v} = (v_1, v_2)$.



(a) $t = 0.5$ (b) $t = 1$ (c) $t = 1.5$

Figure 12: Linear elastodynamic equation: Distributions of the point-wise absolute error, $\|\boldsymbol{u}_\theta - \boldsymbol{u}\|$, of the PINN-H1 solution (top row) and the PINN-H2 solution (bottom row) at three time instants (a) $t = 0.5$, (b) $t = 1.0$, and (c) $t = 1.5$. $N = 2000$ training collocation points within domain and on the domain boundaries.

The neural network has two hidden layers, with widths of 90 and 60 nodes, respectively, and the tanh activation function for all the hidden nodes. For the network training, N collocation points are generated from a uniform random distribution within the domain, on each of the domain boundary, as well as on the initial boundary. N is systematically varied in the simulations. We employ the penalty coefficients $\boldsymbol{W} = (W_1, ..., W_{10}) = (0.9, 0.9, 0.9, 0.9, 0.5, 0.5, 0.5, 0.5, 0.9, 0.9)$ in the simulations.

In Figures 11 and 12 we compare the PINN-H1/PINN-H2 solutions with the exact solution and provide an overview of their errors. Figure 11 is a visualization of the deformed configuration of the domain. Here we have plotted the deformed field, $\boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x}, t)$, for a set of grid points $\boldsymbol{x} \in D$ at three time instants from the exact solution, the PINN-H1 and PINN-H2 solutions. Figure 12 shows distributions of the point-wise absolute error of the PINN-H1/PINN-H2 solutions, $\|\boldsymbol{u}_\theta - \boldsymbol{u}\| = \sqrt{(u_{\theta 1}(\boldsymbol{x}, t) - u_1(\boldsymbol{x}, t))^2 + (u_{\theta 2}(\boldsymbol{x}, t) - u_2(\boldsymbol{x}, t))^2}$, at the same three time instants. Here $\boldsymbol{u}_\theta = (u_{\theta 1}, u_{\theta 2})$ denotes the PINN solution. While both PINN schemes capture the solution fairly well at $t = 0.5$ and 1, at $t = 1.5$ both schemes show larger deviations from the true solution. In general, the PINN-H1 scheme appears to produce a better approximation to the solution than PINN-H2.

The effect of the number of collocation points (N) on the PINN results has been studied in Figure 13 and Table 3, where N is systematically varied in the range $N = 1000$ to $N = 3000$. Figure 13 shows the histories of the loss function for training PINN-H1 and PINN-H2 under different collocation points. Table 3 lists the corresponding l_2 and l_∞ errors of $\boldsymbol{u}$ and $\boldsymbol{v}$ obtained from PINN-H1 and PINN-H2. One can observe that the PINN errors in general tend to improve with increasing number of collocation points. It can also be observed that the PINN-H1 errors in general appear better than those of PINN-H2 for this problem.

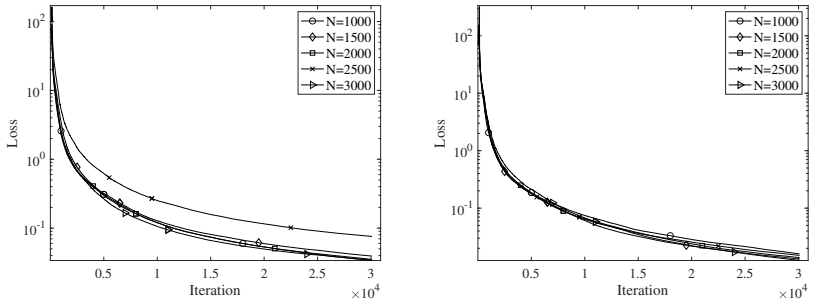
Figure 14 shows the errors of $\boldsymbol{u}$, $\boldsymbol{u}_t$, $\underline{\boldsymbol{\varepsilon}}(\boldsymbol{u})$ and $\nabla \cdot \boldsymbol{u}$ as a function of the loss function value in the network training of PINN-H1 and PINN-H2. The data indicates that these errors approximately scale as the square root of the training loss, which is consistent with the relation as given by Theorem 5.5. This in a sense provides numerical evidence for the theoretical analysis in Section 5.

7. Concluding Remarks

In the present paper we have considered the approximation of a class of dynamic PDEs of second order in time by physics-informed neural networks (PINN). We provide an analysis of the convergence and the error of PINN for approximating the wave equation, the nonlinear Klein-Gordon equation, and the linear

Table 3: Linear elastodynamic equation: The l_2 and l_∞ errors for $\boldsymbol{u} = (u_1, u_2)$ and $\boldsymbol{v} = (v_1, v_2)$ versus the number of training data points N from the PINN-H1 and PINN-H2 solutions.

N	l_2-error				l_∞-error			
	$u_{\theta 1}$	$u_{\theta 2}$	$v_{\theta 1}$	$v_{\theta 2}$	$u_{\theta 1}$	$u_{\theta 2}$	$v_{\theta 1}$	$v_{\theta 2}$
	PINN-H1							
1000	4.8837e-02	6.0673e-02	4.7460e-02	5.1640e-02	1.7189e-01	2.1201e-01	6.9024e-01	6.1540e-01
1500	2.8131e-02	3.1485e-02	4.1104e-02	4.1613e-02	1.9848e-01	2.4670e-01	3.4716e-01	4.0582e-01
2000	2.7796e-02	4.0410e-02	3.5891e-02	4.6334e-02	1.4704e-01	1.7687e-01	4.0678e-01	5.0022e-01
2500	3.0909e-02	4.0215e-02	3.3966e-02	4.4024e-02	1.7589e-01	2.4211e-01	4.1403e-01	3.9570e-01
3000	2.6411e-02	3.5600e-02	4.3209e-02	5.2802e-02	1.4289e-01	1.3625e-01	5.1167e-01	5.3298e-01
	PINN-H2							
1000	4.9869e-02	1.3451e-01	5.6327e-02	5.4796e-02	3.2314e-01	3.4978e-01	6.7624e-01	5.7277e-01
1500	5.4708e-02	1.3987e-01	4.5871e-02	5.1622e-02	2.8609e-01	5.2598e-01	4.9343e-01	2.3518e-01
2000	6.2114e-02	1.0190e-01	6.4477e-02	5.0011e-02	2.5745e-01	3.1642e-01	5.9057e-01	5.8411e-01
2500	3.7887e-02	6.0630e-02	5.4363e-02	5.0659e-02	2.2212e-01	2.4774e-01	5.3681e-01	3.5427e-01
3000	5.4862e-02	6.3407e-02	5.5208e-02	6.0082e-02	3.4102e-01	2.1308e-01	5.1894e-01	4.4995e-01



(a) PINN-H1 (b) PINN-H2

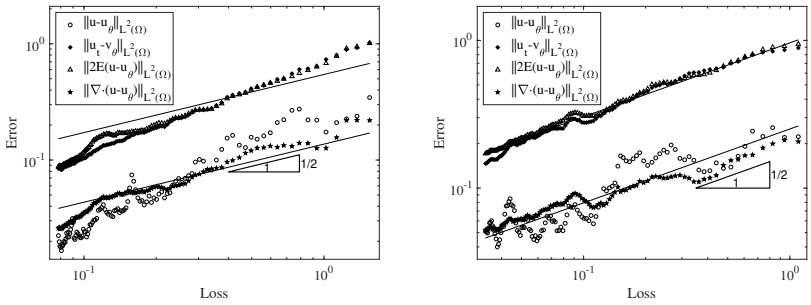
Figure 13: Linear elastodynamic equation: Training loss histories of PINN-H1 and PINN-H2 corresponding to different numbers of collocation points (N) in the simulation.

elastodynamic equation. Our analyses show that, with feed-forward neural networks having two hidden layers and the tanh activation function for all the hidden nodes, the PINN approximation errors for the solution field, its time derivative and its gradient can be bounded by the PINN training loss and the number of training data points (quadrature points).

Our theoretical analyses further suggest new forms for the PINN training loss function, which contain certain residuals that are crucial to the error estimate but would be absent from the canonical PINN formulation of the loss function. These typically include the gradient of the equation residual, the gradient of the initial-condition residual, and the time derivative of the boundary-condition residual. In addition, depending on the type of boundary conditions involved in the problem, our analyses suggest that a norm other than the commonly-used L^2 norm may be more appropriate for the boundary residuals in the loss function. Adopting these new forms of the loss function suggested by the theoretical analyses leads to a variant PINN algorithm. We have implemented the new algorithm and presented a number of numerical experiments on the wave equation, the Sine-Gordon equation and the linear elastodynamic equation. The simulation results demonstrate that the method can capture the solution field well for these PDEs. The numerical data corroborate the theoretical analyses.

Acknowledgments

The work was partially supported by the China Postdoctoral Science Foundation (No.2021M702747), Natural Science Foundation of Hunan Province (No.2022JJ40422), NSF of China (No.11971410 and No.12101495),



(a) PINN-H1 (b) PINN-H2

Figure 14: Linear elastodynamic equation: The errors for $\boldsymbol{u}$, $\boldsymbol{u}_t$, $\underline{\boldsymbol{\varepsilon}}(\boldsymbol{u})$ and $\nabla\cdot\boldsymbol{u}$ versus the training loss value obtained by PINN-H1 and PINN-H2.

General Special Project of Education Department of Shaanxi Provincial Government (No.21JK0943), and the US National Science Foundation (DMS-2012415).

8. Appendix: Auxiliary Results and Proofs of Theorems from Sections 4 and 5

8.1. Some Auxiliary Results

Let a d-tuple of non-negative integers $\alpha \in \mathbb{N}_0^d$ be multi-index with $d \in \mathbb{N}$. For given two multi-indices $\alpha, \beta \in \mathbb{N}_0^d$, we say that $\alpha \le \beta$, if and only if, $\alpha_i \le \beta_i$ for all $i = 1, \cdots, d$. Denote $|\alpha| = \sum_{i=1}^d \alpha_i$, $\alpha! = \prod_{i=1}^d \alpha_i!$, $\binom{\beta}{\alpha} = \frac{\beta!}{\alpha!(\beta-\alpha)!}$. Let $P_{m,n} = \{\alpha \in \mathbb{N}_0^n, |\alpha| = m\}$, for which it holds $|P_{m,n}| = \binom{m+n-1}{m}$.

Lemma 8.1 (Multiplicative trace inequality, e.g. [12])**.** *Let $d \ge 2$, $\Omega \subset \mathbb{R}^d$ be a Lipschitz domain and let $\gamma_0 : H^1(\Omega) \to L^2(\partial\Omega) : u \mapsto u|_{\partial\Omega}$ be the trace operator. Denote by h_Ω the diameter of Ω and by ρ_Ω the radius of the largest d-dimensional ball that can be inscribed into Ω. Then it holds that*

$$\|\gamma_0 u\|_{L^2(\partial\Omega)} \le C_{h_\Omega,d,\rho_\Omega}\|u\|_{H^1(\Omega)}, \quad \text{where } C_{h_\Omega,d,\rho_\Omega} = \sqrt{2\max\{2h_\Omega, d\}/\rho_\Omega}. \tag{67}$$

Lemma 8.2 ([12])**.** *Let $d, n, L, W \in \mathbb{N}$ and let $u_\theta : \mathbb{R}^d \to \mathbb{R}^d$ be a neural network with $\theta \in \Theta$ for $L \ge 2, R, W \ge 1$, c.f. Definition 2.1. Assume that $\|\sigma\|_{C^n} \ge 1$. Then it holds for $1 \le j \le d$ that*

$$\|(u_\theta)_j\|_{C^n(\Omega)} \le 16^L d^{2n}(e^2 n^4 W^3 R^n \|\sigma\|_{C^n(\Omega)})^{nL}. \tag{68}$$

Lemma 8.3 ([12])**.** *Let $d \ge 2, n \ge 2, m \ge 3, \sigma > 0, a_i, b_i \in \mathbb{Z}$ with $a_i < b_i$ for $1 \le i \le d$, $\Omega = \prod_{i=1}^d [a_i, b_i]$ and $f \in H^m(\Omega)$. Then for every $N \in \mathbb{N}$ with $N > 5$ there exists a tanh neural network $\hat{f}^N$ with two hidden layers, one of width at most $3\lceil\frac{m+n-2}{2}\rceil|P_{m-1,d+1}| + \sum_{i=1}^d (b_i - a_i)(N-1)$ and another of width at most $3\lceil\frac{d+n}{2}\rceil|P_{d+1,d+1}|N^d\prod_{i=1}^d(b_i-a_i)$, such that for $k = 0, 1, 2$ it holds that*

$$\|f - \hat{f}^N\|_{H^k(\Omega)} \le 2^k 3^d C_{k,m,d,f}(1+\delta)\ln^k\left(\beta_{k,\delta,d,f} N^{d+m+2}\right) N^{-m+k}, \tag{69}$$

and where

$$C_{k,m,d,f} = \max_{0\le l\le k}\left(\begin{array}{c} d+l-1 \\ l \end{array}\right)^{1/2} \frac{((m-l)!)^{1/2}}{(\lceil\frac{m-l}{d}\rceil!)^{d/2}}\left(\frac{3\sqrt{d}}{\pi}\right)^{m-l}|f|_{H^m(\Omega)},$$

$$\beta_{k,\delta,d,f} = \frac{5\cdot 2^{kd}\max\{\prod_{i=1}^d(b_i-a_i), d\}\max\{\|f\|_{W^{k,\infty}(\Omega)}, 1\}}{3^d\delta\min\{1, C_{k,m,d,f}\}}.$$

Moreover, the weights of $\hat{f}^N$ scale as $O(N^\gamma + N\ln N)$ with $\gamma = \max\{m^2/n, d(2+m+d)/n\}$.

8.2. Proof of Theorem 4.3 and Theorem 5.4

Proof of Theorem 4.3:

Proof. By taking the inner product of (38a) and (38b) with $\hat{u}$ and $\hat{v}$ over D, respectively, we have

$$\frac{d}{2dt}\int_D |\hat{u}|^2\,\mathrm{d}\boldsymbol{x} = \int_D \hat{u}\hat{v}\,\mathrm{d}\boldsymbol{x} + \int_D R_{int1}\hat{u}\,\mathrm{d}\boldsymbol{x} \le \int_D |\hat{u}|^2\,\mathrm{d}\boldsymbol{x} + \frac{1}{2}\int_D |R_{int1}|^2\,\mathrm{d}\boldsymbol{x} + \frac{1}{2}\int_D |\hat{v}|^2\,\mathrm{d}\boldsymbol{x}, \tag{70}$$

$$\begin{aligned}
&\varepsilon^2\frac{d}{2dt}\int_D |\hat{v}|^2\,\mathrm{d}\boldsymbol{x}\\
&= -a^2\int_D \nabla\hat{u}\cdot\nabla\hat{v}\,\mathrm{d}\boldsymbol{x} + a^2\int_{\partial D}\hat{v}\nabla\hat{u}\cdot\boldsymbol{n}\,\mathrm{d}s(\boldsymbol{x}) - \varepsilon_1^2\int_D \hat{u}\hat{v}\,\mathrm{d}\boldsymbol{x} - \int_D (g(u_\theta)-g(u))\hat{v}\,\mathrm{d}\boldsymbol{x} + \int_D R_{int2}\hat{v}\,\mathrm{d}\boldsymbol{x}\\
&= -a^2\frac{d}{2dt}\int_D |\nabla\hat{u}|^2\,\mathrm{d}\boldsymbol{x} + a^2\int_D \nabla\hat{u}\cdot\nabla R_{int1}\,\mathrm{d}\boldsymbol{x} + a^2\int_{\partial D} R_{sb}\nabla\hat{u}\cdot\boldsymbol{n}\,\mathrm{d}s(\boldsymbol{x}) - \varepsilon_1^2\int_D \hat{u}\hat{v}\,\mathrm{d}\boldsymbol{x}\\
&\quad - \int_D (g(u_\theta)-g(u))\hat{v}\,\mathrm{d}\boldsymbol{x} + \int_D R_{int2}\hat{v}\,\mathrm{d}\boldsymbol{x}\\
&\le -a^2\frac{d}{2dt}\int_D |\nabla\hat{u}|^2\,\mathrm{d}\boldsymbol{x} + \frac{a^2}{2}\int_D |\nabla\hat{u}|^2\,\mathrm{d}\boldsymbol{x} + \frac{a^2}{2}\int_D |\nabla R_{int1}|^2\,\mathrm{d}\boldsymbol{x} + C_{\partial D}\left(\int_{\partial D}|R_{sb}|^2\,\mathrm{d}s(\boldsymbol{x})\right)^{\frac{1}{2}}\\
&\quad + \frac{1}{2}(\varepsilon_1^2+L)\int_D |\hat{u}|^2\,\mathrm{d}\boldsymbol{x} + \frac{1}{2}(\varepsilon_1^2+L+1)\int_D |\hat{v}|^2\,\mathrm{d}\boldsymbol{x} + \frac{1}{2}\int_D |R_{int2}|^2\,\mathrm{d}\boldsymbol{x},
\end{aligned} \tag{71}$$

where $C_{\partial D} = a^2|\partial D|^{\frac{1}{2}}(\|u\|_{C^1(\partial D\times[0,T])} + ||u_\theta||_{C^1(\partial D\times[0,T])})$ and $\hat{v} = \hat{u}_t - R_{int1}$ have been used.

Add (70) to (71), and we get

$$\begin{aligned}
&\frac{d}{2dt}\int_D |\hat{u}|^2\,\mathrm{d}\boldsymbol{x} + a^2\frac{d}{2dt}\int_D |\nabla\hat{u}|^2\,\mathrm{d}\boldsymbol{x} + \varepsilon^2\frac{d}{2dt}\int_D |\hat{v}|^2\,\mathrm{d}\boldsymbol{x}\\
&\quad \le \frac{1}{2}(\varepsilon_1^2+L+2)\int_D |\hat{u}|^2\,\mathrm{d}\boldsymbol{x} + \frac{a^2}{2}\int_D |\nabla\hat{u}|^2\,\mathrm{d}\boldsymbol{x} + \frac{1}{2}(\varepsilon_1^2+L+2)\int_D |\hat{v}|^2\,\mathrm{d}\boldsymbol{x} + \frac{1}{2}\int_D |R_{int1}|^2\,\mathrm{d}\boldsymbol{x}\\
&\quad + \frac{1}{2}\int_D |R_{int2}|^2\,\mathrm{d}\boldsymbol{x} + \frac{a^2}{2}\int_D |\nabla R_{int1}|^2\,\mathrm{d}\boldsymbol{x} + C_{\partial D}\left(\int_{\partial D}|R_{sb}|^2\,\mathrm{d}s(\boldsymbol{x})\right)^{\frac{1}{2}}.
\end{aligned} \tag{72}$$

Integrating (72) over $[0,\tau]$ for any $\tau \le T$ and applying the Cauchy–Schwarz inequality, we obtain

$$\begin{aligned}
&\int_D |\hat{u}(\boldsymbol{x},\tau)|^2\,\mathrm{d}\boldsymbol{x} + a^2\int_D |\nabla\hat{u}(\boldsymbol{x},\tau)|^2\,\mathrm{d}\boldsymbol{x} + \varepsilon^2\int_D |\hat{v}(\boldsymbol{x},\tau)|^2\,\mathrm{d}\boldsymbol{x}\\
&\le \int_D |R_{tb1}|^2\,\mathrm{d}\boldsymbol{x} + a^2\int_D |\nabla R_{tb1}|^2\,\mathrm{d}\boldsymbol{x} + \varepsilon^2\int_D |R_{tb2}|^2\,\mathrm{d}\boldsymbol{x} + (2+\varepsilon_1^2+L+a^2)\int_0^\tau\int_D \left(|\hat{u}|^2 + |\nabla\hat{u}|^2 + |\hat{v}|^2\right)\mathrm{d}\boldsymbol{x}\,\mathrm{d}t\\
&+ \int_0^T\int_D \left(|R_{int1}|^2 + a^2|\nabla R_{int1}|^2 + |R_{int2}|^2\right)\mathrm{d}\boldsymbol{x}\,\mathrm{d}t + 2C_{\partial D}|T|^{\frac{1}{2}}\left(\int_0^T\int_{\partial D}|R_{sb}|^2\,\mathrm{d}s(\boldsymbol{x})\,\mathrm{d}t\right)^{\frac{1}{2}}.
\end{aligned}$$

Applying the integral form of the Grönwall inequality to the above inequality leads to,

$$\int_D |\hat{u}(\boldsymbol{x},\tau)|^2\,\mathrm{d}\boldsymbol{x} + a^2\int_D |\nabla\hat{u}(\boldsymbol{x},\tau)|^2\,\mathrm{d}\boldsymbol{x} + \varepsilon^2\int_D |\hat{v}(\boldsymbol{x},\tau)|^2\,\mathrm{d}\boldsymbol{x} \le C_G\exp\left((2+\varepsilon_1^2+L+a^2)T\right), \tag{73}$$

where

$$\begin{aligned}
C_G &= \int_D (|R_{tb1}|^2 + a^2|\nabla R_{tb1}|^2 + \varepsilon^2|R_{tb2}|^2)\,\mathrm{d}\boldsymbol{x} + \int_0^T\int_D (|R_{int1}|^2 + |R_{int2}|^2 + a^2|\nabla R_{int1}|^2)\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t\\
&\quad + 2C_{\partial D}|T|^{\frac{1}{2}}\left(\int_0^T\int_{\partial D}|R_{sb}|^2\,\mathrm{d}s(\boldsymbol{x})\,\mathrm{d}t\right)^{\frac{1}{2}}.
\end{aligned} \tag{74}$$

Then, we integrate (73) over $[0,T]$ to end the proof. □

Proof of Theorem 5.4:

Proof. Taking the L^2 inner product of (49a) and (49b) with $\hat{\boldsymbol{u}}$ and $\hat{\boldsymbol{v}}$ over D, respectively, we have

$$\frac{d}{2dt}\int_D |\hat{\boldsymbol{u}}|^2\,\mathrm{d}\boldsymbol{x} = \int_D \hat{\boldsymbol{u}}\hat{\boldsymbol{v}}\,\mathrm{d}\boldsymbol{x} + \int_D \boldsymbol{R}_{int1}\hat{\boldsymbol{u}}\,\mathrm{d}\boldsymbol{x} \le \int_D |\hat{\boldsymbol{u}}|^2\,\mathrm{d}\boldsymbol{x} + \frac{1}{2}\int_D |\boldsymbol{R}_{int1}|^2\,\mathrm{d}\boldsymbol{x} + \frac{1}{2}\int_D |\hat{\boldsymbol{v}}|^2\,\mathrm{d}\boldsymbol{x}, \tag{75}$$

$$\begin{aligned}
\rho\frac{d}{2dt}\int_D |\hat{\boldsymbol{v}}|^2\,\mathrm{d}\boldsymbol{x} &= -2\mu\int_D \underline{\boldsymbol{\varepsilon}}(\hat{\boldsymbol{u}}) : \nabla\hat{\boldsymbol{v}}\,\mathrm{d}\boldsymbol{x} - \lambda\int_D (\nabla\cdot\hat{\boldsymbol{u}})(\nabla\cdot\hat{\boldsymbol{v}})\,\mathrm{d}\boldsymbol{x} + \int_{\partial D}(2\mu\underline{\boldsymbol{\varepsilon}}(\hat{\boldsymbol{u}})\boldsymbol{n} + \lambda(\nabla\cdot\hat{\boldsymbol{u}})\boldsymbol{n})\cdot\hat{\boldsymbol{v}}\,\mathrm{d}s(\boldsymbol{x}) \\
&\quad + \int_D \boldsymbol{R}_{int2}\hat{\boldsymbol{v}}\,\mathrm{d}\boldsymbol{x} \\
&= -\frac{d}{dt}\int_D \mu|\underline{\boldsymbol{\varepsilon}}(\hat{\boldsymbol{u}})|^2\,\mathrm{d}\boldsymbol{x} - \frac{d}{dt}\int_D \frac{\lambda}{2}|\nabla\cdot\hat{\boldsymbol{u}}|^2\,\mathrm{d}\boldsymbol{x} + 2\mu\int_D \underline{\boldsymbol{\varepsilon}}(\hat{\boldsymbol{u}}) : \nabla\boldsymbol{R}_{int1}\,\mathrm{d}\boldsymbol{x} + \lambda\int_D (\nabla\cdot\hat{\boldsymbol{u}})(\nabla\cdot\boldsymbol{R}_{int1})\,\mathrm{d}\boldsymbol{x} \\
&\quad + \int_{\Gamma_D}(2\mu\underline{\boldsymbol{\varepsilon}}(\hat{\boldsymbol{u}})\boldsymbol{n} + \lambda(\nabla\cdot\hat{\boldsymbol{u}})\boldsymbol{n})\cdot\boldsymbol{R}_{sb1}\,\mathrm{d}s(\boldsymbol{x}) + \int_{\Gamma_N}\boldsymbol{R}_{sb2}\cdot\hat{\boldsymbol{v}}\,\mathrm{d}s(\boldsymbol{x}) + \int_D \boldsymbol{R}_{int2}\hat{\boldsymbol{v}}\,\mathrm{d}\boldsymbol{x} \\
&\le -\frac{d}{dt}\int_D \mu|\underline{\boldsymbol{\varepsilon}}(\hat{\boldsymbol{u}})|^2\,\mathrm{d}\boldsymbol{x} - \frac{d}{dt}\int_D \frac{\lambda}{2}|\nabla\cdot\hat{\boldsymbol{u}}|^2\,\mathrm{d}\boldsymbol{x} + \mu\int_D |\underline{\boldsymbol{\varepsilon}}(\hat{\boldsymbol{u}})|^2\,\mathrm{d}\boldsymbol{x} + \mu\int_D |\underline{\boldsymbol{\varepsilon}}(\boldsymbol{R}_{int1})|^2\,\mathrm{d}\boldsymbol{x} \\
&\quad + \frac{\lambda}{2}\int_D |\nabla\cdot\boldsymbol{R}_{int1}|\,\mathrm{d}\boldsymbol{x} + \frac{\lambda}{2}\int_D |\nabla\cdot\hat{\boldsymbol{u}}|\,\mathrm{d}\boldsymbol{x} + \frac{1}{2}\int_D |\hat{\boldsymbol{v}}|^2\,\mathrm{d}\boldsymbol{x} + \frac{1}{2}\int_D |\boldsymbol{R}_{int2}|^2\,\mathrm{d}\boldsymbol{x} \\
&\quad + C_{\Gamma_D}\left(\int_{\Gamma_D}|\boldsymbol{R}_{sb1}|^2\,\mathrm{d}s(\boldsymbol{x})\right)^{\frac{1}{2}} + C_{\Gamma_N}\left(\int_{\Gamma_N}|\boldsymbol{R}_{sb2}|^2\,\mathrm{d}s(\boldsymbol{x})\right)^{\frac{1}{2}}.
\end{aligned} \tag{76}$$

Here we have used $\hat{\boldsymbol{v}} = \hat{\boldsymbol{u}}_t - \boldsymbol{R}_{int1}$, and the constants are given by $C_{\Gamma_D} = (2\mu+\lambda)|\Gamma_D|^{\frac{1}{2}}\|\boldsymbol{u}\|_{C^1(\Gamma_D\times[0,T])} + (2\mu+\lambda)|\Gamma_D|^{\frac{1}{2}}||\boldsymbol{u}_\theta||_{C^1(\Gamma_D\times[0,T])}$ and $C_{\Gamma_N} = |\Gamma_N|^{\frac{1}{2}}(\|\boldsymbol{v}\|_{C(\Gamma_N\times[0,T])} + ||\boldsymbol{v}_\theta||_{C(\Gamma_N\times[0,T])})$.

Add (75) to (76), and we get,

$$\begin{aligned}
&\frac{d}{2dt}\int_D |\hat{\boldsymbol{u}}|^2\,\mathrm{d}\boldsymbol{x} + \frac{d}{dt}\int_D \mu|\underline{\boldsymbol{\varepsilon}}(\hat{\boldsymbol{u}})|^2\,\mathrm{d}\boldsymbol{x} + \frac{d}{2dt}\int_D \lambda|\nabla\cdot\hat{\boldsymbol{u}}|^2\,\mathrm{d}\boldsymbol{x} + \rho\frac{d}{2dt}\int_D |\hat{\boldsymbol{v}}|^2\,\mathrm{d}\boldsymbol{x} \\
&\quad \le \int_D \left(|\hat{\boldsymbol{u}}|^2 + \mu|\underline{\boldsymbol{\varepsilon}}(\hat{\boldsymbol{u}})|^2 + \frac{\lambda}{2}|\nabla\cdot\hat{\boldsymbol{u}}| + |\hat{\boldsymbol{v}}|^2\right)\mathrm{d}\boldsymbol{x} + \frac{1}{2}\int_D (|\boldsymbol{R}_{int1}|^2 + |\boldsymbol{R}_{int2}|^2 + 2\mu|\underline{\boldsymbol{\varepsilon}}(\boldsymbol{R}_{int1})|^2)\,\mathrm{d}\boldsymbol{x} \\
&\qquad + \frac{\lambda}{2}\int_D |\nabla\cdot\boldsymbol{R}_{int1}|\,\mathrm{d}\boldsymbol{x} + C_{\Gamma_D}\left(\int_{\Gamma_D}|\boldsymbol{R}_{sb1}|^2\,\mathrm{d}s(\boldsymbol{x})\right)^{\frac{1}{2}} + C_{\Gamma_N}\left(\int_{\Gamma_N}|\boldsymbol{R}_{sb2}|^2\,\mathrm{d}s(\boldsymbol{x})\right)^{\frac{1}{2}}.
\end{aligned} \tag{77}$$

Integrating (77) over $[0,\tau]$ for any $\tau \le T$ and applying Cauchy–Schwarz inequality, we obtain,

$$\begin{aligned}
&\int_D |\hat{\boldsymbol{u}}(\boldsymbol{x},\tau)|^2\,\mathrm{d}\boldsymbol{x} + \int_D 2\mu|\underline{\boldsymbol{\varepsilon}}(\hat{\boldsymbol{u}}(\boldsymbol{x},\tau))|^2\,\mathrm{d}\boldsymbol{x} + \int_D \lambda|\nabla\cdot\hat{\boldsymbol{u}}(\boldsymbol{x},\tau)|^2\,\mathrm{d}\boldsymbol{x} + \rho\int_D |\hat{\boldsymbol{v}}(\boldsymbol{x},\tau)|^2\,\mathrm{d}\boldsymbol{x} \\
&\quad \le \int_D |\boldsymbol{R}_{tb1}|^2\,\mathrm{d}\boldsymbol{x} + \int_D 2\mu|\underline{\boldsymbol{\varepsilon}}(\boldsymbol{R}_{tb1})|^2\,\mathrm{d}\boldsymbol{x} + \int_D \lambda|\nabla\cdot\boldsymbol{R}_{tb1}|^2\,\mathrm{d}\boldsymbol{x} + \rho\int_D |\boldsymbol{R}_{tb2}|^2\,\mathrm{d}\boldsymbol{x} \\
&\qquad + (2+2\mu+\lambda)\int_0^\tau\int_D \left(|\hat{\boldsymbol{u}}|^2 + |\underline{\boldsymbol{\varepsilon}}(\hat{\boldsymbol{u}})|^2 + |\nabla\cdot\hat{\boldsymbol{u}}|^2 + |\hat{\boldsymbol{v}}|^2\right)\mathrm{d}\boldsymbol{x}\,\mathrm{d}t \\
&\qquad + \int_0^T\int_D \left(|\boldsymbol{R}_{int1}|^2 + 2\mu|\underline{\boldsymbol{\varepsilon}}(\boldsymbol{R}_{int1})|^2 + \lambda|\nabla\cdot\boldsymbol{R}_{int1}|^2 + |\boldsymbol{R}_{int2}|^2\right)\mathrm{d}\boldsymbol{x}\,\mathrm{d}t \\
&\qquad + 2|T|^{\frac{1}{2}}C_{\Gamma_D}\left(\int_0^T\int_{\Gamma_D}|\boldsymbol{R}_{sb1}|^2\,\mathrm{d}s(\boldsymbol{x})\,\mathrm{d}t\right)^{\frac{1}{2}} + 2|T|^{\frac{1}{2}}C_{\Gamma_N}\left(\int_0^T\int_{\Gamma_N}|\boldsymbol{R}_{sb2}|^2\,\mathrm{d}s(\boldsymbol{x})\,\mathrm{d}t\right)^{\frac{1}{2}}.
\end{aligned}$$

By applying the integral form of the Grönwall inequality to the above inequality, we have

$$\int_D (|\hat{\boldsymbol{u}}(\boldsymbol{x},\tau)|^2 + 2\mu|\underline{\boldsymbol{\varepsilon}}(\hat{\boldsymbol{u}}(\boldsymbol{x},\tau))|^2 + \lambda|\nabla\cdot\hat{\boldsymbol{u}}(\boldsymbol{x},\tau)|^2 + \rho\int_D |\hat{\boldsymbol{v}}(\boldsymbol{x},\tau)|^2)\,\mathrm{d}\boldsymbol{x} \le C_G\exp\left((2+2\mu+\lambda)T\right), \tag{78}$$

where

$$C_G = \int_D |\boldsymbol{R}_{tb1}|^2\,\mathrm{d}\boldsymbol{x} + \int_D 2\mu|\underline{\boldsymbol{\varepsilon}}(\boldsymbol{R}_{tb1})|^2\,\mathrm{d}\boldsymbol{x} + \int_D \lambda|\nabla\cdot\boldsymbol{R}_{tb1}|^2\,\mathrm{d}\boldsymbol{x} + \rho\int_D |\boldsymbol{R}_{tb2}|^2\,\mathrm{d}\boldsymbol{x}$$

$$+\int_0^T\int_D \left(|\boldsymbol{R}_{int1}|^2+2\mu|\underline{\boldsymbol{\varepsilon}}(\boldsymbol{R}_{int1})|^2+\lambda|\nabla\cdot\boldsymbol{R}_{int1}|^2+|\boldsymbol{R}_{int2}|^2\right)\mathrm{d}\boldsymbol{x}\,\mathrm{d}t$$
$$+2|T|^{\frac{1}{2}}C_{\Gamma_D}\left(\int_0^T\int_{\Gamma_D}|\boldsymbol{R}_{sb1}|^2\,\mathrm{d}s(\boldsymbol{x})\,\mathrm{d}t\right)^{\frac{1}{2}}+2|T|^{\frac{1}{2}}C_{\Gamma_N}\left(\int_0^T\int_{\Gamma_N}|\boldsymbol{R}_{sb2}|^2\,\mathrm{d}s(\boldsymbol{x})\,\mathrm{d}t\right)^{\frac{1}{2}}. \tag{79}$$

Then, we finish the proof by integrating (78) over $[0,T]$. □

References

[1] P. F. Antonietti, I. Mazzieri, High-order discontinuous Galerkin methods for the elastodynamics equation on polygonal and polyhedral meshes, Comput. Methods Appl. Mech. Engrg. 342 (2018) 414–437.

[2] G. Bai, U. Koley, S. Mishra, R. Molinaro, Physics informed neural networks (PINNs) for approximating nonlinear dispersive PDEs, J. Comput. Math. 39 (6) (2021) 816–847.

[3] C. Beck, W. E, A. Jentzen, Machine learning approximation algorithms for high-dimensional fully nonlinear partial differential equations and second-order backward stochastic differential equations, J. Nonlinear Sci. 29 (4) (2019) 1563–1619.

[4] J. Berner, P. Grohs, A. Jentzen, Analysis of the generalization error: empirical risk minimization over deep artificial neural networks overcomes the curse of dimensionality in the numerical approximation of Black-Scholes partial differential equations, SIAM J. Math. Data Sci. 2 (3) (2020) 631–657.

[5] A. Biswas, J. Tian, S. Ulusoy, Error estimates for deep learning methods in fluid dynamics, Numer. Math. 151 (3) (2022) 753–777.

[6] Z. Cai, J. Chen, M. Liu, X. Liu, Deep least-squares methods: an unsupervised learning-based numerical method for solving elliptic PDEs, J. Comput. Phys. 420 (2020) 109707.

[7] F. Calabro, G. Fabiani, C. Siettos, Extreme learning machine collocation for the numerical solution of elliptic PDEs with sharp gradients, Comput. Methods Appl. Mech. Engrg. 387 (2021) 114188.

[8] O. Calin, Deep learning architectures–a mathematical approach, Springer Series in the Data Sciences, Springer, Cham, 2020.

[9] S. Cuomo, V. Schiano Di Cola, F. Giampaolo, G. Rozza, M. Raissi, F. Piccialli, Scientific machine learning through physics-informed neural networks: where we are and what's next, J. Sci. Comput. 92 (3) (2022) 88.

[10] E. Cyr, M. Gulian, R. Patel, M. Perego, N. Trask, Robust training and initialization of deep neural networks: An adaptive basis viewpoint, Proceedings of Machine Learning Research 107 (2020) 512–536.

[11] P. Davis, P. Rabinowitz, Methods of numerical integration, Dover Publications, Inc, 2007.

[12] T. De Ryck, A. D. Jagtap, S. Mishra, Error estimates for physics-informed neural networks approximating the Navier–Stokes equations, IMA J. Numer. Anal. (2023) drac085(https://doi.org/10.1093/imanum/drac085).

[13] T. De Ryck, S. Lanthaler, S. Mishra, On the approximation of functions by tanh neural networks, Neural Networks 143 (2021) 732–750.

[14] T. De Ryck, S. Mishra, Error analysis for physics-informed neural networks (PINNs) approximating Kolmogorov PDEs, Adv. Comput. Math. 48 (6) (2022) 79.

[15] M. Dehghan, A. Shokri, Numerical solution of the nonlinear Klein-Gordon equation using radial basis functions, Journal of Computational and Applied Mathematics 230 (2009) 400–410.

[16] S. Dong, Z. Li, Local extreme learning machines and domain decomposition for solving linear and nonlinear partial differential equations, Comput. Methods Appl. Mech. Engrg. 387 (2021) 114129, (also arXiv:2012.02895).

[17] S. Dong, Z. Li, A modified batch intrinsic plasticity method for pre-training the random coefficients of extreme learning machines, J. Comput. Phys. 445 (2021) 110585.

[18] S. Dong, N. Ni, A method for representing periodic functions and enforcing exactly periodic boundary conditions with deep neural networks, J. Comput. Phys. 435 (2021) 110242.

[19] S. Dong, Y. Wang, A method for computing inverse parametric PDE problems with randomized neural networks, arXiv:2210.04338 .

[20] S. Dong, J. Yang, Numerical approximation of partial differential equations by a variable projection method with artificial neural networks, Comput. Methods Appl. Mech. Engrg. 398 (2022) 115284, (also arXiv:2201.09989).

[21] S. Dong, J. Yang, On computing the hyperparameter of extreme learning machines: algorithms and applications to computational PDEs, and comparison with classical and high-order finite elements, J. Comput. Phys. 463 (2022) 111290, (also arXiv:2110.14121).

[22] W. E, B. Yu, The deep Ritz method: a deep learning-based numerical algorithm for solving variational problems, Commun. Math. Stat. 6 (2018) 1–12.

[23] D. Elbrächter, D. Perekrestenko, P. Grohs, H. Bölcskei, Deep neural network approximation theory, IEEE Trans. Inform. Theory 67 (5) (2021) 2581–2623.

[24] G. Fabiani, F. Calabro, L. Russo, C. Siettos, Numerical solution and bifurcation analysis of nonlinear partial differential equations with extreme learning machines, J. Sci. Comput. 89 (2021) 44.

[25] J. He, J. Xu, MgNet: A unified framework for multigrid and convolutional neural network, Sci. China Math. 62 (2019) 1331–1354.

[26] R. Hu, Q. Lin, R. Alan, S. Tang, Higher-order error estimates for physics-informed neural networks approximating the primitive equations, arXiv:2209.11929 .

[27] Z. Hu, A. D. Jagtap, G. E. Karniadakis, K. Kawaguchi, When do extended physics-informed neural networks (XPINNs) improve generalization?, SIAM J. Sci. Comput. 44 (5) (2022) A3158–A3182.

[28] Z. Hu, C. Liu, Y. Wang, Z. Xu, Energetic variational neural network discretizations to gradient flows, arXiv:2206.07303 .

[29] T. J. R. Hughes, J. E. Marsden, Classical elastodynamics as a linear symmetric hyperbolic system, J. Elasticity 8 (1) (1978) 97–110.

[30] A. Jagtap, G. Karniadakis, Extended physics-informed neural network (XPINNs): A generalized space-time domain decomposition based deep learning framework for nonlinear partial differential equations, Commun. Comput. Phys. 28 (2020) 2002–2041.

[31] A. Jagtap, E. Kharazmi, G. Karniadakis, Conservative physics-informed neural networks on discrete domains for conservation laws: applications to forward and inverse problems, Comput. Methods Appl. Mech. Engrg. 365 (2020) 113028.

[32] G. Karniadakis, G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, L. Yang, Physics-Informed Machine Learning, Nat. Rev. Phys. 3 (2021) 422–440.

[33] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980 .

[34] A. Krishnapriyan, A. Gholami, S. Zhe, R. Kirby, M. Mahoney, Characterizing possible failure modes in physics-informed neural networks, Advances in Neural Information Processing Systems 34 (2021) 26548–26560.

[35] K. Kubota, K. Yokoyama, Global existence of classical solutions to systems of nonlinear wave equations with different speeds of propagation, Japan. J. Math. 27 (1) (2001) 113–202.

[36] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (2015) 436–444.

[37] L. Lu, X. Meng, Z. Mao, G. E. Karniadakis, DeepXDE: a deep learning library for solving differential equations, SIAM Rev. 63 (1) (2021) 208–228.

[38] S. Mishra, R. Molinaro, Physics informed neural networks for simulating radiative transfer, J. Quant. Spectrosc. Radiat. Transfer 270 (2021) 107705.

[39] S. Mishra, R. Molinaro, Estimates on the generalization error of physics-informed neural networks for approximating a class of inverse problems for PDEs, IMA J. Numer. Anal. 42 (2) (2022) 981–1022.

[40] S. Mishra, R. Molinaro, Estimates on the generalization error of physics-informed neural networks for approximating PDEs, IMA J. Numer. Anal. 43 (1) (2023) 1–43.

[41] S. Mishra, T. K. Rusch, Enhancing accuracy of deep learning algorithms by training with low-discrepancy sequences, SIAM J. Numer. Anal. 59 (3) (2021) 1811–1834.

[42] P. Niyogi, F. Girosi, Generalization bounds for function approximation from scattered noisy data, Adv.

Comput. Math. 10 (1) (1999) 51–80.

[43] J. Nocedal, S. J. Wright, Numerical optimization, Springer, New York, second edn., 2006.

[44] M. Penwarden, A. Jagtap, S. Zhe, G. Karniadakis, R. Kirby, A unified scalable framework for causal sweeping strategies for Physics-Informed Neural Networks (PINNs) and their temporal decompositions, arXiv:2302.14227 .

[45] A. F. Psaros, K. Kawaguchi, G. E. Karniadakis, Meta-learning PINN loss functions, J. Comput. Phys. 458 (2022) 111121.

[46] 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.

[47] J. Shatah, Global existence of small solutions to nonlinear evolution equations, J. Differential Equations 46 (3) (1982) 409–425.

[48] J. Shatah, Normal forms and quadratic nonlinear Klein-Gordon equations, Comm. Pure Appl. Math. 38 (5) (1985) 685–696.

[49] Y. Shin, J. Darbon, G. E. Karniadakis, On the convergence of physics informed neural networks for linear second-order elliptic and parabolic type PDEs, Commun. Comput. Phys. 28 (5) (2020) 2042–2074.

[50] Y. Shin, Z. Zhang, G. E. Karniadakis, Error estimates of residual minimization using neural networks for linear PDEs, arXiv:2010.08019 .

[51] J. Siegel, Q. Hong, X. Jin, W. Hao, J. Xu, Greedy Training Algorithms for Neural Networks and Applications to PDEs, arXiv:2107.04466 .

[52] J. Sirignano, K. Spoliopoulos, DGM: A deep learning algorithm for solving partial differential equations, J. Comput. Phys. 375 (2018) 1339–1364.

[53] A. Tartakovsky, C. Marrero, P. Perdikaris, G. Tartakovsky, D. Barajas-Solano, Physics-informed deep neural networks for learning parameters and constitutive relationships in subsurface flow problems, Water Resour. Res. 56 (2020) e2019WR026731.

[54] R. Temam, Infinite-dimensional dynamical systems in mechanics and physics, Springer-Verlag, New York, second edn., 1997.

[55] X. Wan, S. Wei, VAE-KRnet and its applications to variational Bayes, Commun. Comput. Phys. 31 (2022) 1049–1082.

[56] B. Wang, Classical global solutions for non-linear Klein-Gordon-Schrödinger equations, Math. Methods Appl. Sci. 20 (7) (1997) 599–616.

[57] S. Wang, X. Yu, P. Perdikaris, When and why PINNs fail to train: a neural tangent kernel perspective, J. Comput. Phys. 449 (2022) 110768.

[58] Y. Wang, G. Lin, Efficient deep learning techniques for multiphase flow simulation in heterogeneous porous media, J. Comput. Phys. 401 (2020) 108968.

[59] K. Yosida, Functional analysis, vol. 123, Springer-Verlag, Berlin-New York, sixth edn., 1980.

[60] U. Zerbinati, PINNs and GaLS: A Priori error estimates for shallow physics informed neural networks applied to elliptic problems, IFAC-PapersOnLine 55 (20) (2022) 61–66.