Gradient enhanced physics-informed neural networks for digital twins of structural vibrations

Tanmoy Chatterjee  
*Faculty of Science & Engineering, Swansea University, United Kingdom.*  
*E-mail: tanmoy.chatterjee@swansea.ac.uk*

Michael I. Friswell  
*Faculty of Science & Engineering, Swansea University, United Kingdom.*  
*E-mail: m.i.friswell@swansea.ac.uk*

Sondipon Adhikari  
*James Watt School of Engineering, University of Glasgow.*  
*E-mail: sondipon.adhikari@glasgow.ac.uk*

Hamed H. Khodaparast  
*Faculty of Science & Engineering, Swansea University, United Kingdom.*  
*E-mail: h.haddadkhodaparast@swansea.ac.uk*

Physics-informed neural networks have received considerable attention for the solution and data-driven discovery of physical systems governed by differential equations. Despite their immense success, it has been observed that the first-generation PINNs suffer from a drawback arising from the regularization of the loss function. The motivation of this work is to address the above inherent drawback and improve their approximation potential for solving forward and inverse structural vibration problems. Thus, a novel second-generation extended approach called gradient enhanced physics-informed neural networks is proposed. The gradient enhanced framework is observed to mitigate the regularization issue effectively by modifying the loss term and hence adequately capture the spatial and temporal response behaviour. The adopted strategy utilizes the same computational cost as that of the conventional approach. Two representative forward and inverse structural vibration problems involving ordinary and partial differential equations are solved to assess the performance of the proposed method. The results are validated with the analytical solutions.

**Keywords:** Deep learning, Physics-informed neural networks, Automatic differentiation, Gradient enhanced, Differential equations, Structural Vibrations, Inverse problem, Digital Twin.

1. **Introduction**

During the past five years (2017-22), deep learning (DL) has proved to be a promising tool for the inference and data-driven discovery of physical systems governed by differential equations. In addition to DL’s lucid, intuitive and general structure as solvers of differential equations, their universal recognition in the scientific computing community should be attributed to the recent developments in automatic differentiation (AD) (Baydin et al., 2017), high-performance computing capabilities and sincere efforts towards democratization of Artificial Intelligence, implemented by open-source software such as TensorFlow (Abadi et al., 2015), PyTorch (Paszke et al., 2019) and Keras (Chollet, 2017). The above developments have significantly contributed to the rise of the next generation neural solvers and data-driven identification tools for ordinary differential equations (ODEs) and partial differential equations (PDEs), referred to as physics-informed neural networks (PINNs) (Lagaris et al., 1998; Raissi et al., 2019). For a comprehensive review of PINNs and their applications to various disciplines, one is referred to a recent article by one of the pioneering research groups (Karniadakis et al., 2021).

The architecture of PINNs can be tailored to obey symmetries, invariance, or conservation principles originating from the governing physi-
Cal laws modelled by time-dependant and nonlinear ODEs and PDEs. This feature makes PINNs an ideal candidate to incorporate the domain knowledge in the form of soft constraints so that this prior information can act as a regularization medium for seamless exploration and exploitation of the solution space. Motivated by the following points that

- time-varying (continuous) systems can be notoriously difficult to capture and,
- PINNs have been mostly applied to systems having spatial variance and their investigation on dynamic systems exhibiting temporal variation and subjected to external forcing are limited (Lai et al., 2021; Yucsesan et al., 2021),

the objective of this paper is to investigate PINNs for solving forward and inverse problems in structural vibrations.

A key highlight of this work involves detecting a regularization issue inherent to first generation PINNs pointed out recently by (Wang et al., 2021) for other applications. Despite the popularity of PINNs, this issue is found to be so prevalent that it leads to significant approximation errors in capturing the behaviour of linear structural dynamic systems. More importantly, it is further demonstrated that with a modified loss function, significant enhancement in approximation accuracy can be achieved leading to a robustly validated Digital Twin for structural vibrations. One of the advantages of the proposed strategy is that no additional computational effort compared to conventional PINNs is required. The solution and identification of two representative structural vibration problems involving both ODE and PDE have been carried out using PINNs.

2. Formulation of gradient enhanced physics-informed neural networks

Two of the crucial concerns which PINNs overcome are

- the overdependence of data-driven deep neural networks (DNN) on training data and,
- lack of interpretability.

This proves to be especially useful as sufficient knowledge/information in the form of data is often not available for physical systems. The basic concept of PINNs is to evaluate hyperparameters of the DNN by making use of the governing physics and encoding this prior information within the neural architecture in the form of the ODE/PDE. This soft constraint ensures the conservation of the physical laws modelled by the governing equation, initial and boundary conditions and available measurements.

The PDE for the solution \( u(t, x) \) parameterized by parameters \( \xi \) defined in the domain \( \Omega \) can be expressed as

\[
\begin{align*}
F(t, x; \frac{\partial u}{\partial t}, \frac{\partial u}{\partial x_1}, \ldots, \frac{\partial u}{\partial x_d}; \\
\frac{\partial^2 u}{\partial t^2}, \frac{\partial^2 u}{\partial x_1^2}, \ldots, \frac{\partial^2 u}{\partial x_1 \partial x_d}; \xi) = 0
\end{align*}
\]

with the following initial and boundary conditions, respectively, as,

\[
\begin{align*}
I(u, t = 0, x) = 0, & \quad x \in \Omega \\
B(u, t, x) = 0, & \quad t \in [0, T], x \in \partial \Omega
\end{align*}
\]

where \( t \) and \( x \) represent the time and spatial coordinates, respectively and \( \partial \Omega \) is the boundary of \( \Omega \). For solving the PDE via PINNs, the solution \( u(t, x) \) is approximated by constructing a neural network \( \hat{u}(t, x; \theta) \) such that \( u(t, x) \approx \hat{u}(t, x; \theta) = N_u(t, x; \theta) \) with \( \theta \) being the trainable parameters. Next, the training strategy of PINNs to be followed has been illustrated pointwise.

(i) A set of collocation points inside the domain \( (D_F) \) is generated using a suitable experimental design scheme. Another set of points is to be generated individually on the boundary \( (D_B) \) and corresponding to the initial conditions \( (D_I) \).

(ii) The loss function that penalizes the PDE residual is formulated based on the generated interior collocation points as,
The loss functions that ensure the satisfaction of the boundary and initial conditions, respectively, are defined as,

\[ L_B(\theta; D_B) = \frac{1}{|D_B|} \sum_{x \in D_B} \left\| B(\hat{u}, t, x) \right\|_2 \] (4)

\[ L_I(\theta; D_I) = \frac{1}{|D_I|} \sum_{x \in D_I} \left\| I(\hat{u}, t = 0, x) \right\|_2 \] (5)

The composite loss function is defined as the sum of the above individual loss terms.

\[ \mathcal{L} = \mathcal{L}_F + \mathcal{L}_B + \mathcal{L}_I \] (6)

The final goal is to compute the parameters \( \theta \) by minimizing the loss function in Eq. (6) as shown below and construct a DNN representation.

\[ \hat{\theta} = \arg \min_{\theta} \mathcal{L}(\theta) \] (7)

Usually, \( \mathcal{L}(\theta) \) is minimized using the stochastic gradient descent method. Once the PINNs model is constructed, it can be used to predict the system response to an unknown input \((\tilde{t}, \tilde{x})\). It is worth acknowledging that the actual ODE/PDE is not solved (i.e., any response/output data is not required) during the entire training phase of PINNs for capturing the solution and hence, is a simulation free forward ODE/PDE solver.

Despite proving successful in solving multiple problems across diverse disciplines, conventional PINNs (as discussed above) has often been incapable in terms of performance even for simple problems. This is due to the regularization of the composite loss term as defined in Eq. (6). In particular, the individual loss functions \( \mathcal{L}_F, \mathcal{L}_B \) and \( \mathcal{L}_I \) are of widely varying scales leading to gradient imbalances between the loss terms and eventually resulting in an inaccurate representation of the PDE solution. This issue has been recently analyzed in detail (Wang et al., 2021). Since manual tuning to vary the weights of each term can be tedious, numerous studies on multi-objective optimization have been undertaken which allow adaptive scaling. A scaling approach was proposed by (Wang et al., 2021) for PINNs based on balancing the distribution of gradients of each term in the loss function. Although these approaches are effective, they entail additional computational effort.

In this work, we employ an alternative approach to address the scaling issue which requires no extra computational effort. To avoid multiple terms in the composite loss function, the DNN output \( \hat{u} \) is modified to \( \hat{u}_{\text{mod}} \) so that the PDE residual, initial and/or, boundary conditions are satisfied simultaneously (\( \hat{u}_{\text{mod}} = g(\hat{u}, t, x) \)).

The determination of the mapping function \( g \) involves simple manipulation of the expression of the DNN approximated solution. The idea has been inspired from and is analogous to determination of a consistent displacement function in a classical Lagrange’s approach or selecting a potential shape/basis function in the finite element approach so that they satisfy the given conditions. As the physics will change from problem to problem depending on the given conditions, the mapping function \( g \) will have to be determined intrusively and has been illustrated later in the numerical examples section. In presence of the modified neural network output \( \hat{u}_{\text{mod}} \), the new loss function can be expressed as,

\[ L_{\text{new}}(\theta; D_F) = \frac{1}{|D_F|} \sum_{x \in D_F} \left\| \mathcal{F}(t, x; \hat{u}_{\text{mod}}, \hat{u}_{\text{mod}}, ..., \hat{u}_{\text{mod}}) \right\|_2 \] (8)

Note that the new loss function only involves the PDE residual of the modified output \( \hat{u}_{\text{mod}} \) in the domain \( D_F \) and still satisfies the associated
boundary and/or, initial conditions by avoiding their corresponding loss terms. Hence, generating additional data at \( t = 0 \) and along the boundary is no longer necessary. Likewise, the derivatives \( \frac{\partial u_{\text{mod}}}{\partial t}, \frac{\partial u_{\text{mod}}}{\partial x_1}, \frac{\partial^2 u_{\text{mod}}}{\partial t^2}, \frac{\partial^2 u_{\text{mod}}}{\partial x_1^2} \) in Eq. (8) are computed using AD. The above adopted strategy is referred to as gradient enhanced PINNs.

One useful feature of PINNs is that the same framework can be employed for data-driven discovery (inverse problems) with a slight modification of the loss function. The necessary modification is discussed next. If the parameter \( \xi \) in Eq. (1) is not known, and instead \( D_M \) set of measurements of response \( u^* \) is available, then an additional loss term minimizing the discrepancy between the measurements and the neural network output can be defined as,

\[
L_M(\theta, \xi; D_M) = \frac{1}{|D_M|} \sum_{x \in D_M} \left| \hat{u}(x) - u^*(x) \right|^2
\]

This term \( L_M \) determines the unknown parameters along with the solution. Thus, the combined loss term is expressed as,

\[
L = L_F + L_B + L_I + L_M
\]

Lastly, the parameters \( (\theta, \xi) \) are computed by minimizing the loss function in Eq. (10) as shown below.

\[
\hat{\theta}, \hat{\xi} = \arg\min_{\theta, \xi} L(\theta, \xi)
\]

Identification of the unknown parameters \( \xi \) updates the system (physics) as per the available data and results in a PINNs-based Digital Twin framework for structural vibrations.

### 3. Numerical Examples

In this section, the performance of conventional and gradient enhanced PINNs is accessed for solving two representative structural vibration problems, involving an ODE and a PDE. In doing so, both forward and inverse problems have been addressed. The deep learning library of MATLAB was used to carry out the numerical investigation.

#### 3.1. Forced vibration of a damped spring-mass system

The first example is of a forced vibration of a damped spring-mass system and can be expressed by

\[
\ddot{u} + 2\zeta\omega_n \dot{u} + \omega_n^2 u = f_0 \sin \omega t
\]

The initial conditions are \( u(t = 0) = 0 \) and \( \dot{u}(t = 0) = 0 \). The analytical solution to the above system can be found in (Inman, 2008).

As mentioned previously, the solution space (of the ODE, for this case) can be approximated by DNN such that \( \hat{u} = N_u(t, \theta) \), where the residual of the ODE is evaluated with the help of AD. The resulting optimization problem can be expressed as,

\[
\arg\min_{\theta \in \mathbb{R}^d} L(\theta) := \left\| \frac{\partial^2 \hat{u}}{\partial t^2} + 2\zeta\omega_n \frac{\partial \hat{u}}{\partial t} + \omega_n^2 \hat{u} - f_0 \sin \omega t \right\|_2^2 + \left\| \hat{u}(t = 0) \right\|_2^2 + \left\| \frac{\partial \hat{u}}{\partial t} (t = 0) \right\|_2^2
\]

For the numerical illustration, it is assumed that \( \omega = 3, \zeta = 0.025, r = \omega/\omega_n = 1.5 \) and \( f_0 = 1 \). The displacement \( u \) is approximated using a fully-connected neural network with 4 hidden layers and 20 neurons per layer. A sinusoidal activation function has been used due to the known periodic nature of the data. The neural network is run for 1000 epochs and the mini-batch size is 1000. The initial learning rate is assumed to be 0.01 and the popular ADAM optimizer is employed. The solution \( \hat{u} \) obtained using the PINNs framework has been compared with the actual (analytical) solution \( u \) in Fig. 1a. 20,000 collocation points have been generated for time data \( t \in [0 \ 8\pi] \) with the help of Latin hypercube sampling to obtain the results in Figs. 1a and 1b. For testing the PINNs framework, 5000 points were uniformly generated for time \( t \in [0 \ 8\pi] \) to obtain the results in Figs. 1a and 1b. It can be observed from Fig. 1a that the conventional PINNs framework is not capable of capturing the time response variation satisfactorily. As discussed in the previous sections, the reason is related to the regularization of the loss term in Eq. (13). To automatically satisfy
the initial conditions, the modified output of the neural network $\hat{u}_{\text{mod}}$ is expressed as,

$$\hat{u}_{\text{mod}} = t\hat{u}$$

Therefore, the new loss function can be expressed as,

$$\arg\min_{\theta \in \mathbb{R}^d} \mathcal{L}_{\text{new}}(\theta) := \left\| \frac{\partial^2 \hat{u}_{\text{mod}}}{\partial t^2} + 2\zeta\omega_n \frac{\partial \hat{u}_{\text{mod}}}{\partial t} + \omega_n^2 \hat{u}_{\text{mod}} - f_0 \sin \omega t \right\|_2$$

Following this approach, significant improvement in approximation of the displacement response has been achieved as shown in Fig. 1b. The approximation by PINNs is found to be excellent in terms of capturing the response trends.

Next, the implementation of PINNs has been illustrated for an inverse setting. For doing so, the same problem as defined by Eq. (12) is reformulated such that the displacement time history is given in the form of measurements and both natural frequency $\omega_n$ and damping ratio $\zeta$ have to be identified simultaneously. The optimization problem can be expressed as,

$$\arg\min_{\theta \in \mathbb{R}^d, \omega, \zeta \in \mathbb{R}} \mathcal{L}(\theta, \omega, \zeta) := \left\| \frac{\partial^2 \hat{u}}{\partial t^2} + 2\zeta\omega_n \frac{\partial \hat{u}}{\partial t} + \omega_n^2 \right\|_2 + \|\hat{u} - u^*\|_2$$

where, $u^*$ represents the measured displacement data. 10,000 collocation points have been generated for time data $t \in [0, 4\pi]$ with the help of Latin hypercube sampling. 1000 displacement data points were used to artificially simulate the measurement data and 1% uniform random noise was added.

The results have been presented in the form of convergence of the identified parameters (natural frequency and damping ratio) in Fig. 2. The converged value of $\omega_n = 2.9985$ and $\zeta = 0.0097$ demonstrate a close match with the actual values of 3 and 0.01, respectively.

3.2. Free vibration of a rectangular membrane

A rectangular membrane with unit dimensions excited by an initial displacement $u = \sin \pi x \sin \pi y$ has been considered in this example. The governing PDE, initial and boundary conditions are expressed by
Fig. 2.: Identification results for the damped forced spring-mass system in terms of convergence of the identified natural frequency (left axis) and identified damping ratio (right axis).

\[ c\nabla^2 u = c \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) = \frac{\partial^2 u}{\partial t^2} \quad \forall x, y \in [0,1], t > 0 \]  
(17)

\[ u = 0 \quad \forall x, y \in \Gamma_u \]  
(18)

\[ u = \sin \pi x \sin \pi y \quad \forall t = 0, x, y \in [0,1] \]  
(19)

\[ \frac{\partial u}{\partial t} = 0 \quad \forall t = 0, x, y \in [0,1] \]  
(20)

where, \( u \) is the displacement, \( \Gamma_u \) represents the boundary and \( c \) is the velocity of wave propagation. The analytical solution to the governing PDE is \( u(x, y, t) = \sin \pi x \sin \pi y \cos \sqrt{2} \pi t \).

Using the PINNs framework, the solution of the PDE is approximated by a DNN such that \( \hat{u} = \mathcal{N}_u(x, y, t, \theta) \) and the optimization problem can be expressed as,

\[
\arg \min_{\theta \in \mathbb{R}^d} \mathcal{L}_\text{new}(\theta) := \frac{1}{2} \left[ c \left( \frac{\partial^2 \hat{u}}{\partial x^2} + \frac{\partial^2 \hat{u}}{\partial y^2} \right) - \frac{\partial^2 \hat{u}}{\partial t^2} \right]^2 + \frac{1}{2} \left[ \hat{u}(x, y \in \Gamma_u) \right]^2 + \frac{1}{2} \left[ \hat{u}(t = 0) - \sin \pi x \sin \pi y \right]^2 + \frac{1}{2} \left[ \frac{\partial \hat{u}}{\partial t} (t = 0) \right]^2
\]  
(21)

The architecture and the parameters of the neural network are the same as that adopted in the previous example. 5000 collocation points are generated for the spatial data \( x, y \in [0,1] \) and temporal data \( t \in [0,1/(2\sqrt{2})] \) with the help of Latin hypercube sampling. For testing the PINNs framework, 1000 points were uniformly generated for \( x, y \) and \( t \). The solution in space obtained using the PINNs framework \( \hat{u} \) (Fig. 3b) has been compared with the actual (analytical) solution \( u \) (Fig. 3a) for four different time instants \( t = 0.1, 0.15, 0.2 \) and 0.25.

It can be observed from Fig. 3b that the conventional PINNs framework is not capable of capturing the time response variation satisfactorily. The reason is once again related to the regularization of the loss term in Eq. (21). Specifically, the fact that the condition \( u = 0 \) at the boundary of the domain not being satisfied in the predicted response by conventional PINNs can be visualized from Fig. 3b.

To ensure the satisfaction of residual, initial and boundary conditions and improve upon the approximation accuracy, the neural network output has been modified as,

\[ \hat{u}_\text{mod} = t^2(x-1)y(y-1)\hat{u} + \sin \pi x \sin \pi y \]  
(22)

Since the modified neural network output is \( \hat{u}_\text{mod} \), the new optimization problem can be expressed as,

\[
\arg \min_{\theta \in \mathbb{R}^d} \mathcal{L}_\text{new}(\theta) := c \left( \frac{\partial^2 \hat{u}_\text{mod}}{\partial x^2} + \frac{\partial^2 \hat{u}_\text{mod}}{\partial y^2} \right) - \frac{\partial^2 \hat{u}_\text{mod}}{\partial t^2} \right]^2
\]  
(23)

Following this modified PINNs approach, significant improvement in the spatial distribution of the displacement response has been achieved as shown in Fig. 3c. Next, the implementation of PINNs has been illustrated in solving another inverse problem. For doing so, the same problem as defined by Eqs. (17)-(20) is re-formulated such that the displacement time history is given in the form of measurements and the wave velocity \( c \) has to be identified. The optimization problem can be expressed as,
Fig. 3.: Results of free vibration of the rectangular membrane (a) True forward spatial solution, (b) Predicted forward spatial solution by conventional PINNs, (c) Predicted forward spatial solution by gradient enhanced PINNs, (d) Inverse solution in the form of convergence of the identified parameter $c$. 

$$\arg \min_{\theta \in \mathbb{R}^n, c \in \mathbb{R}} L(\theta, c) := \left\| c \left( \frac{\partial^2 \hat{u}}{\partial x^2} + \frac{\partial^2 \hat{u}}{\partial y^2} \right) - \frac{\partial^2 \hat{u}}{\partial t^2} \right\|_2^2 + \| \hat{u} - u^* \|_2^2$$

(24)

where, $u^*$ represents the measured displacement data. 20,000 collocation points have been generated for spatial coordinates $x, y \in [0, 1]$ and time $t \in [0, 1/(2\sqrt{2})]$ with the help of Latin hypercube sampling. 5000 displacement data points were used to artificially simulate the measurement data and 2% uniform random noise was added. The results have been presented in the form of convergence of the identified parameter $c$ at the end of 10,000 epochs in Fig. 3d. The converged value of $c = 0.9902$ demonstrates good match.
with the actual value \( c = 1.0 \). It is worth noting that the PINNs framework is inherently adapted to also provide the solution to the PDE along with the identified parameter in the inverse setup.

4. Summary and Conclusions

This work presents gradient enhanced PINNs for solving structural vibration problems. This is one of the very few applications of PINNs in structural vibrations to date which makes the work timely in nature. It demonstrates a critical drawback of the first generation PINNs while solving vibration problems, which leads to inaccurate predictions. The proposed gradient enhanced PINNs framework addresses the above drawback with the help of a modification in the loss term without adding any extra computational cost. This has led to significant improvement in the approximation accuracy.

Two representative problems in structural vibrations, involving both an ODE and a PDE have been solved. Forward and inverse problems have been addressed while solving each of the problems. Results of both the problems clearly show that the conventional PINNs is incapable of approximating the response due to the regularisation issue. Gradient enhanced PINNs addresses the above issue and captures the solution of the ODE/PDE adequately. Excellent performance of the proposed digital twin framework obtained for linear structural vibrations makes it an ideal choice to extend their investigation to nonlinear dynamics and systems with higher-order PDEs, such as beam and plate models.

Acknowledgement

The authors gratefully acknowledge the support of the Engineering and Physical Sciences Research Council through the award of a Programme Grant "Digital Twins for Improved Dynamic Design", grant number EP/R006768.

References


