

SECOND-ORDER ANALYSIS OF BEAM-COLUMNS BY MACHINE LEARNING-BASED STRUCTURAL ANALYSIS THROUGH PHYSICS-INFORMED NEURAL NETWORKS

Liang Chen ¹, Hao-Yi Zhang ¹, Si-Wei Liu ^{1, *} and Siu-Lai Chan ²

¹ Department of Civil and Environmental Engineering, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong, China

² NIDA Technology Company Limited, Science Park, Hong Kong, China

* (Corresponding author: E-mail: si-wei.liu@polyu.edu.hk)

ABSTRACT

The second-order analysis of slender steel members could be challenging, especially when large deflection is involved. This paper proposes a novel machine learning-based structural analysis (MLSA) method for second-order analysis of beam-columns, which could be a promising alternative to the prevailing solutions using over-simplified analytical equations or traditional finite-element-based methods. The effectiveness of the conventional machine learning method heavily depends on both the qualitative and the quantitative of the provided data. However, such data are typically scarce and expensive to obtain in structural engineering practices. To address this problem, a new and explainable machine learning-based method, named Physics-informed Neural Networks (PINN), is employed, where the physical information will be utilized to orientate the learning process to create a self-supervised learning procedure, making it possible to train the neural network with few or even no predefined datasets to achieve an accurate approximation. This research extends the PINN method to the problems of second-order analysis of steel beam-columns. Detailed derivations of the governing equations, as well as the essential physical information for the training process, are given. The PINN framework and the training procedure are provided, where an adaptive loss weight control algorithm and the transfer learning technique are adopted to improve numerical efficiency. The practicability and accuracy of which are validated by four sets of verification examples.

ARTICLE HISTORY

Received: 9 November 2023
Revised: 23 November 2023
Accepted: 29 November 2023

KEYWORDS

Beam-columns;
Physics-informed neural networks;
Second-order analysis;
Machine learning

Copyright © 2023 by The Hong Kong Institute of Steel Construction. All rights reserved.

1. Introduction

Numerical solutions using machine learning have seen drastic development in recent years, largely due to the advancement in Artificial Intelligence (AI) chips, especially the advent of highly computationally efficient Graphical Processor Units (GPUs) with extensive parallel processing capacities. The application of machine learning-based (ML) methods for solving various structural engineering problems has been explored recently, such as structural analysis and design [1-5], structural health monitoring [6-10], structural optimization [11-16], and so on. The existing research mainly focuses on adopting the ML technique for regression problems. These methods are data-intensive to establish artificial neural networks (ANNs) to solve structural analysis problems through regressions. Among these methods, ANNs are used as a “black box” and the effectiveness highly relies on the qualitative and quantitative of the provided data. However, in structural engineering practice, data is often relatively scarce and costly to generate [17], which causes certain obstacles to adopting the emerging machine-learning techniques in the field of computational structural engineering.

Recently, a novel machine learning method, namely Physics-Informed Neural Networks (PINN), has been proposed by Raissi et al. [18], which is an unsupervised ML technique with the potentials for solving complex mechanical problems. PINN enriches the neural network with information from underlying physical laws, making it possible to train the neural network with few or even no pre-defined datasets while still achieving accurate approximations [19, 20]. PINN incorporates physical information into the learning process, thereby relaxing the requirements for training data. The physical information is used to guide the learning process to create a self-supervised learning procedure [21]. The features of this method are highlighted by: (1) the absence of a requirement for a pre-defined dataset, which facilitates its application to problems where the governing equations for the solution are known; (2) a potentially very fast and efficient solution process, once the network has been pre-trained using transfer learning [22-24]; and (3) a mesh-free method that does not require the division of domains.

Despite those promising features, there are still some issues with PINN. For instance, PINNs employ physics as a regularizing term in their objective function, which brings forth the challenge of manually adjusting the corresponding hyperparameters. Besides, the absence of validation data or prior knowledge of the solution to the Partial Differential Equation (PDE) can render

PINN impracticable for solving forward problems [25]. To address this, methods such as reducing the order of a given PDE, relaxing stringent smoothness requirements [25, 26], using pre-trained neural networks, and implementing transfer learning [22-24] have been proposed. Employing these techniques, PINN shows promising potential for solving various forward problems, especially in the structural engineering area, a variety of PINN-based models have been recently developed for a few mechanical problems, such as plate and shell structures [17, 27-29], solid mechanics [30, 31], beam-column members [21, 32, 33], and simple structures [34], among others. For example, Li et al. [27] established a PINN framework to characterize the finite deformation of elastic plates, where only a pseudo dataset was used. Haghighat and Juanes [31] illustrate the PINN to solve nonlinear solid mechanics, revealing that the stress distribution and displacement fields can be predicted accurately. Tao et al. [32] introduced the PINN approach for axial compression buckling analysis of thin-walled cylinders, and their research shows that the PINN model can provide satisfactory prediction only using a considerably reduced size of data. Katsikis et al. [21] illustrated the application of the PINN in static beam problems, where linear behaviors of the beams in different loading conditions can be predicted successfully. These research works confirm the feasibility of using PINN for some forward mechanical problems, suggesting that it could be an alternative solution with promising features to be the next-generation structural analysis method.

Second-order analysis is a simulation-based method and includes the effects of large deflection by establishing the equilibrium conditions on deformed shapes, which is originally proposed for stability analysis of steel members and is now used extensively in the design of steel structures [35-37] nowadays. The use of second-order analysis for the stability and large deflection problems has been extensively studied in recent decades, including stability function methods [38-40], Finite Element method (FEM) using shell elements [41-43], Line element method [44-47], Finite strip method [48-50] and so on. These methods are mostly simulation-based numerically and require trial-and-error procedures to obtain the solutions, which are usually time-consuming with extensive computational efforts. In contrast to the traditional matrix structural analysis (MSA) method, involving computational efforts of solving large sparse linear equations, the machine learning-based structural analysis (MLSA) method through PINN is proposed for the second-order analysis of steel members in this research. A schematic illustration of the traditional MSA and the proposed MLSA methods is given in Fig. 1.

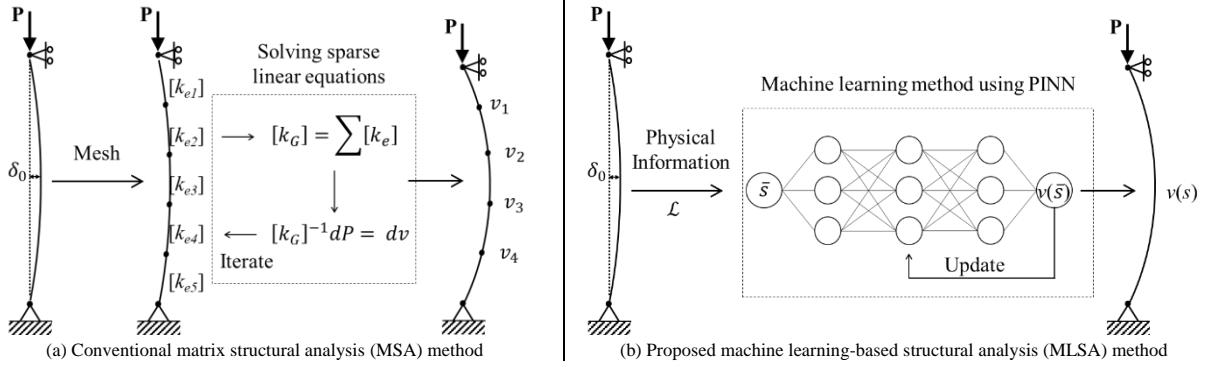


Fig. 1 An illustration of the conventional MSA method and the MLSA method

This paper extends the application of the machine learning-based method to the second-order analysis of steel beam-columns using PINN. The governing equations for the typical beam-columns under large deflection are derived and then incorporated into the physics-informed neural networks. The physical information derived from the governing equations and the boundary conditions is used to guide the training process to establish a self-supervised learning procedure. The derivation of the governing equations, the architecture of the PINN frameworks, and the training procedure are provided in detail. The practicability and accuracy of the proposed MLSA method are validated using four sets of verification examples.

2. Formulation

This paper proposed an innovative machine-learning-based method through PINN for the second-order analysis of steel members, where the accuracy of the embedded physical information in neural networks is the key to an accurate and reliable solution. The detailed derivation of the governing differential equations, which delineates the physical information, is given in the following section.

2.1. Assumptions

The following assumptions are adopted: (1) this research focuses on the geometric nonlinearity (second-order effect) of beam-columns; thus, material nonlinearity is not considered; (2) plane sections remain plane after deformation; (3) the applied loads are conservative; (4) all member connections are assumed to be either pinned or rigid, semi-rigid joints are not considered, and (5) member local buckling and distortional buckling are not considered. These assumptions are commonly accounted for in conventional structural engineering practices but can be revised and upgraded using the same concept discussed in this paper.

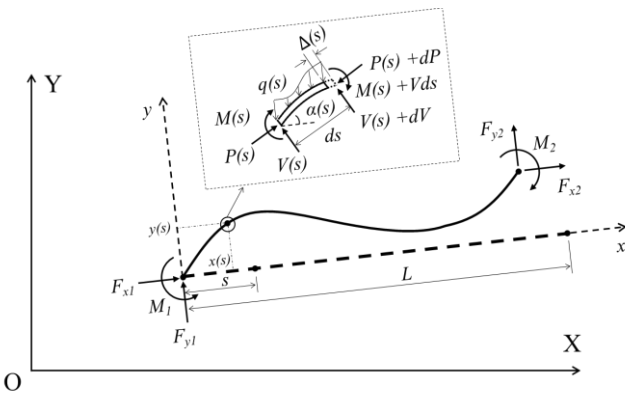


Fig. 2 Forces and deflections of a typical beam-column member

2.2. Mechanical model and problematic description

The basic forces versus displacements relations for a typical steel beam-column member are given in Fig. 2. The forces applied on the member ends can be summarized as follows:

$$\mathbf{F} = [F_{x1} \ F_{y1} \ M_1 \ F_{x2} \ F_{y2} \ M_2]^T \quad (1)$$

where the subscripts 1 and 2 indicate the left and right ends of the member. For a small length of the member with an original length of ds (Fig. 2), the forces acting on it can be calculated based on the equilibrium equations for the forces:

$$P(s) = F_{x1} \cos \alpha(s) + F_{y1} \sin \alpha(s) + \int_0^s q(s) \sin \alpha(s) ds \quad (2)$$

$$M(s) = -F_{x1} y(s) + F_{y1} x(s) - M_1 - \int_0^s q(s) x(s) ds \quad (3)$$

$$V(s) = -F_{x1} \sin \alpha(s) + F_{y1} \cos \alpha(s) - \int_0^s q(s) \cos \alpha(s) ds \quad (4)$$

in which, $y(s)$ and $x(s)$ are the local coordinates along the member after deformation, and $q(s)$ is the distribution load applied on the member.

The displacements of any points on the member after deformation can be generated by,

$$u(s) = x(s) - s = \int_0^s \cos \alpha(s) (ds - \Delta(s)) - s; \quad 0 \leq s \leq L \quad (5)$$

$$v(s) = y(s) = \int_0^s \sin \alpha(s) (ds - \Delta(s)); \quad 0 \leq s \leq L \quad (6)$$

in which, $\Delta(s)$ is the axial deformation as shown in Fig. 2 and $\alpha(s)$ is the slope along the member, which can be obtained by solving the governing differential equations.

2.3. Governing differential equations

The governing differential equations of the beam-columns can be written as given below, based on the Hooke's law and the Euler-Bernoulli moment-curvature relationship,

$$\frac{\Delta(s)}{ds} = \frac{P(s)}{EA}, \quad \frac{d\alpha(s)}{ds} = -\frac{M(s)}{EI}, \quad \frac{d^2\alpha(s)}{ds^2} = -\frac{V(s)}{EI},$$

and

$$\frac{d^3\alpha(s)}{ds^3} = -\frac{q(s)}{EI} \quad (7)$$

where E is the material Young's modulus, A is the cross-section area, and I is the second moment of area.

2.4. Boundary conditions

As shown in Fig. 2, there are totally six boundary cases in a typical beam-column member. The boundary conditions and the corresponding governing equations are given in Table 1.

2.5. Solutions to the problem

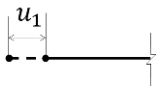
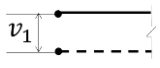
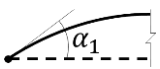
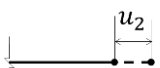
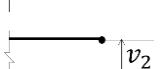
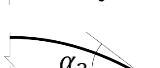
One conventional manner to solve the governing differential equations given above are to use the closed-form solutions like elliptic integral solutions [51-54]. However, in some cases, for example, when general boundary conditions are considered, the governing differential equations are notoriously difficult to solve using closed-form solutions [55]. Therefore, numerical

solutions like the FEM using line elements, also known as the frame analysis method, are often required [56]. However, the conventional FEM is based on variational principles rather than differential equations, where the primary physical law is the extremal principle, and the differential equations is only the

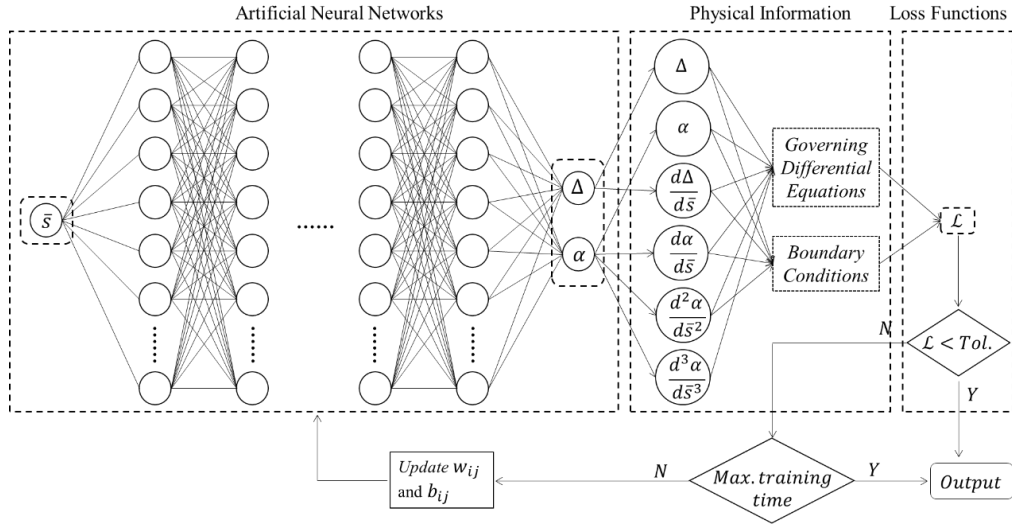
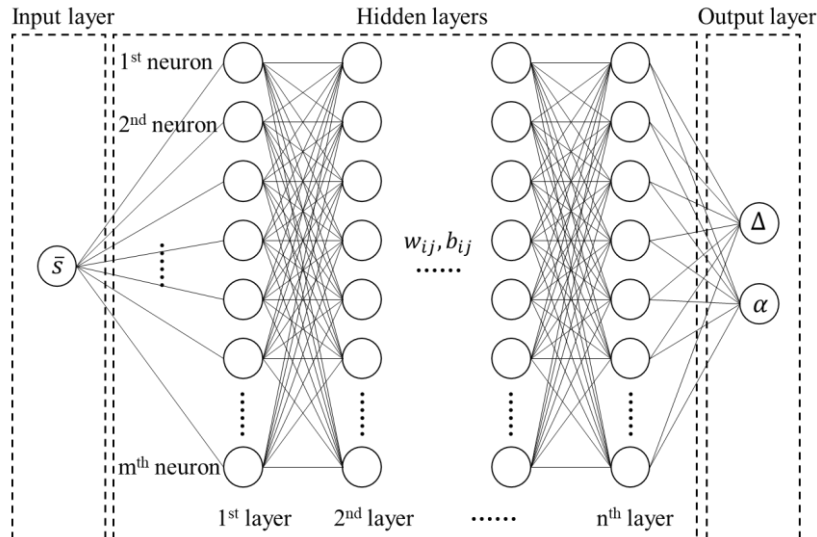
secondary consequence [57]. Therefore, a refined mesh is essentially required to get an accurate approximation, leading to extensive computational time and data-manipulation efforts.

Table 1

Governing equations for different boundary conditions

Boundary Condition	Fixed	Free
	$u(0) = 0$	$\frac{\Delta(s)}{ds} \Big _{s=0} = \frac{F_{x1}\cos\alpha(0) + F_{y1}\sin\alpha(0)}{EA}$
	$v(0) = 0$	$\frac{d^2\alpha(s)}{ds^2} \Big _{s=0} = \frac{F_{y1}\cos\alpha(0) - F_{x1}\sin\alpha(0)}{EI}$
	$\alpha(0) = 0$	$\frac{d\alpha(s)}{ds} \Big _{s=0} = \frac{M_1}{EI}$
	$u(L) = 0$	$\frac{\Delta(s)}{ds} \Big _{s=L} = \frac{F_{x2}\cos\alpha(L) + F_{y2}\sin\alpha(L)}{EA}$
	$v(L) = 0$	$\frac{d^2\alpha(s)}{ds^2} \Big _{s=L} = \frac{F_{y2}\cos\alpha(L) - F_{x2}\sin\alpha(L)}{EI}$
	$\alpha(L) = 0$	$\frac{d\alpha(s)}{ds} \Big _{s=L} = \frac{M_2}{EI}$

Note: the subscripts ₁ and ₂ indicates the left and right ends of the member

**Fig. 3** The proposed PINN framework**Fig. 4** Architecture of the neural network

3. Physics-informed neural networks (PINN) framework

A machine learning-based solution via physics-informed neural networks (PINN) for solving and discovering partial differential equations has been adopted in this research to solve the governing differential equations as derived above. The machine learning method using PINN to solve partial differential equations was first proposed by Raissi et al. [18] and later adopted by several researchers for different mechanical problems [19, 58-60].

Fig. 3 illustrates the framework of the PINN proposed for the present problem, which is composed of three parts: the ANNs (the approximation), the physical information (the target and correction for the approximation), and the loss functions (the evaluation of the differences). The details of each component of the PINN framework will be given in the succeeding sections. It should be noted that the proposed PINN is non-data-driven neural network; as such, pre-generated data is not required.

3.1. Neural networks architecture

An essential technique required in the PINN is the architecture of the network that is formed by neurons in several layers. Neurons are expressed in an activation function used for the forward propagation that passes the information in the neural network. The neural network may comprise a large number of neurons with a complicated layer structure, which is crucial for machine-learning-based method. In this research, a fully connected feedforward neural network, as shown in Fig. 4, is used as the core of the training model. The neural network is composed of an input layer, n numbers of hidden layers, and an output layer, where the connection between layers can be written as,

$$\alpha_{ij} = f(w_{ij}\bar{s} + b_{ij}) \text{ for } j=1 \text{ (Input layer)} \quad (8)$$

Table 2

The number of neurons and the activation function for each layer

	Input layer	1 st hidden layer	2 nd hidden layer	3 rd hidden layer	Output layers
Number of neurons	1	10	10	10	2
Activation function	<i>Tanhshrink</i>	<i>Tanh</i>	<i>Tanh</i>	<i>Tanh</i>	<i>Tanhshrink</i>

3.2. Physical information

In the traditional machine-learning methods, the ANNs are used as the “black-box” algorithms, the effectiveness of which is significantly dependent on the quality and quantity of the data provided [17]. The machine-learning methods using PINN integrate the governing equations into the learning process to essentially relax the requirements on training data. The physics-informed governing equations are used to orientate the learning process creating a self-

$$\alpha_{ij} = \sum_{i=1}^m f(w_{ij}\alpha_{ij-1} + b_{ij}) \text{ for } 2 \leq j \leq n \text{ (Hidden layer)} \quad (9)$$

$$\alpha(x) = \sum_{i=1}^m f(w_{ij}\alpha_{ij-1} + b_{ij}) \text{ for } j = n+1 \text{ (Output layer)} \quad (10)$$

in which, $\bar{s} = s/L$ represents the normalized coordinate along the member; n and m are the numbers of hidden layers and the number of neurons at each layer, respectively, and $n=3$ and $m=10$ are used in the present study of being effective for most problems; w_{ij} are the internal weights; b_{ij} are the bias; and f stands for the activation function.

The activation functions were introduced by Candès [61] based on harmonic analysis, which decide whether a neuron should be activated or not based on the importance of the neuron’s input in the process of the prediction. In the present study, two types of activation functions, *Tanh* and *Tanhshrink*, are adopted in the hidden layers and input/output layers, respectively. (See Table 2).

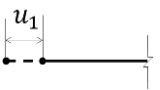
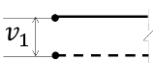
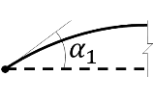
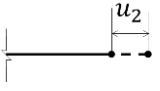
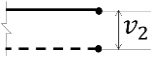
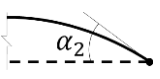
$$\text{Tanh}(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (11)$$

$$\text{Tanhshrink}(x) = x - \text{Tanh}(x) \quad (12)$$

supervised learning procedure, which is considered the unsupervised learning method. In summary, the PINN method is a machine learning-based computational method intended to provide an accurate prediction of problems via the ANNs with incorporation of the physical information. As shown in Fig. 3, the physical information incorporated in the proposed PINN method is provided by two parts: the governing differential equations and the boundary conditions.

Table 3

Governing equations for different boundary conditions

Boundary Conditions	Fixed	Free
	$\mathcal{L}_{BC1} = [0 - u(0)]^2$	$\mathcal{L}_{BC1} = \left[\frac{F_{x1}\cos\theta(0) + F_{y1}\sin\theta(0)}{EA} - \frac{\Delta(s)}{ds} \right]_{s=0}^2$
	$\mathcal{L}_{BC2} = [0 - v(0)]^2$	$\mathcal{L}_{BC2} = \left[\frac{F_{y1}\cos\theta(0) - F_{x1}\sin\theta(0)}{EI} - \frac{d^2\theta(s)}{ds^2} \right]_{s=0}^2$
	$\mathcal{L}_{BC3} = [0 - \theta(0)]^2$	$\mathcal{L}_{BC3} = \left[\frac{M_1}{EI} - \frac{d\theta(s)}{ds} \right]_{s=0}^2$
	$\mathcal{L}_{BC4} = [0 - u(L)]^2$	$\mathcal{L}_{BC4} = \left[\frac{F_{x2}\cos\theta(L) + F_{y2}\sin\theta(L)}{EA} - \frac{\Delta(s)}{ds} \right]_{s=L}^2$
	$\mathcal{L}_{BC5} = [0 - v(L)]^2$	$\mathcal{L}_{BC5} = \left[\frac{F_{y2}\cos\theta(L) - F_{x2}\sin\theta(L)}{EI} - \frac{d^2\theta(s)}{ds^2} \right]_{s=L}^2$
	$\mathcal{L}_{BC6} = [0 - \theta(L)]^2$	$\mathcal{L}_{BC6} = \left[\frac{M_2}{EI} - \frac{d\theta(s)}{ds} \right]_{s=L}^2$

Note: the subscripts 1 and 2 indicates the left and right ends of the member

3.3. Loss functions

The machine-learning solution via PINN requires the problem to be priorly described in a set of governing equations and the boundary conditions in the form of a series of partial differential equations. The training process is subsequently carried out to scrutinize the solution to the problem by assessing the errors, generally articulated in the form of loss functions, to verify its accuracy. The loss functions will provide a beneficial regularization effect on the training process by penalizing simulations that are not respectful of the given physical information. Based on the governing differential equations and the boundary conditions, the total loss function \mathcal{L} can be expressed by,

$$\mathcal{L} = \sum_{i=1}^4 \mathcal{L}_{Gi} + \sum_{j=1}^6 \mathcal{L}_{BCj} \quad (13)$$

where, \mathcal{L}_G is the loss function from the governing differential equations given by Equation (7):

$$\mathcal{L}_{g1} = \frac{1}{Nc} \sum_{i=1}^{Nc} \left[\frac{P(s)}{EA} - \frac{\Delta}{ds} \right]_{s_i}^2 \quad (14)$$

$$\mathcal{L}_{g2} = \frac{1}{Nc} \sum_{i=1}^{Nc} \left[\frac{M(s)}{EI} - \frac{d\theta}{ds} \right]_{s_i}^2 \quad (15)$$

$$\mathcal{L}_{g3} = \frac{1}{Nc} \sum_{i=1}^{Nc} \left[\frac{V(s)}{EI} - \frac{d^2\theta}{ds^2} \right]_{s_i}^2 \quad (16)$$

$$\mathcal{L}_{g4} = \frac{1}{Nc} \sum_{i=1}^{Nc} \left[\frac{q(s)}{EI} - \frac{d^3\theta}{ds^3} \right]_{s_i}^2 \quad (17)$$

in which, Nc is the number of the collocation points, which are randomly aligned along the member. \mathcal{L}_{BC} is the loss function from the boundary conditions that can be defined accordingly to Table 3.

4. Training details

The governing equations and boundary conditions of the beam-columns are defined in the form of a series of partial differential equations as given. Sequentially, the neural networks (see Fig. 4) are then established to provide an approximation solution to the problem. The training process, depicted in Fig. 5, is executed to examine the approximation of the problem by evaluating the errors - usually denoted in the form of loss functions - to validate its accuracy. During the training process, forward and back propagation operations are proceeded, wherein the internal weights w_{ij} and the bias b_{ij} connecting the neurons are updated continuously to minimize errors. The solution could be obtained once the value of the loss function is within the acceptable tolerance (usually 0.1%). This training process can be accelerated dramatically by introducing the adaptive loss function method [62], and the trained network can be inherited and later used for other problems by adopting the transfer learning method [22, 23].

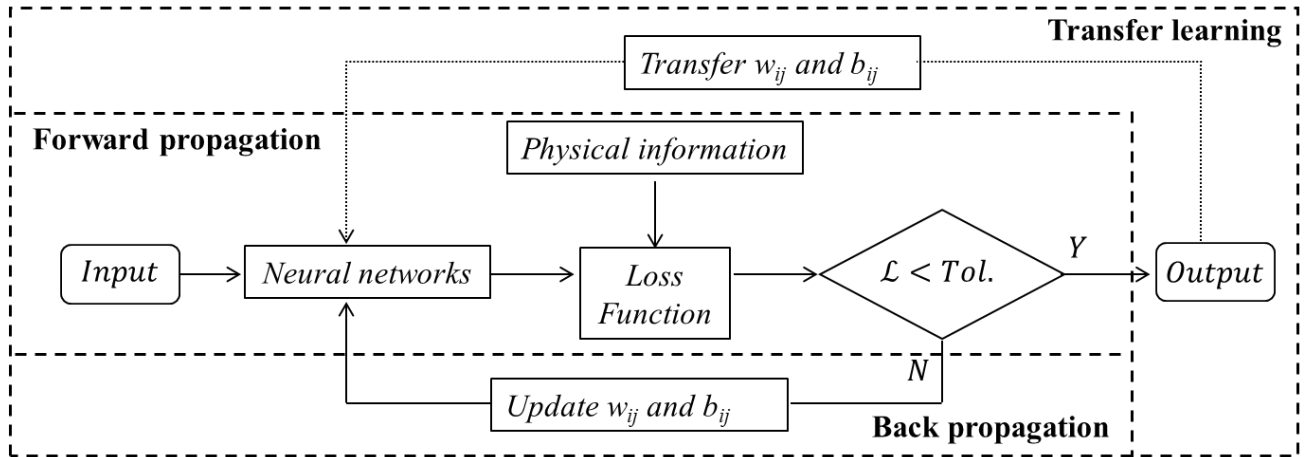


Fig. 5 The training procedure

4.1. Self-supervised learning procedure

In this research, the underlying physical laws underpinning the second-order analysis of beam-columns encompassed within governing differential equations and the boundary conditions, based on which the loss functions are given to evaluate the differences between the approximation from the PINN-derived approximation and the established physical information. As shown in Fig. 5, during the learning procedure, the internal parameters of the neural network (w_{ij} and b_{ij}) will be updated by optimization algorithms to minimize the loss functions given in Equation (13).

Several stochastic optimization algorithms have been proposed to minimize the loss function [63, 64], among which, the Adaptive Moment Estimation introduced by Kingma and Ba [65] is one of the most efficient and widely adopted optimization algorithms [66, 67]. In the present study, the learning rate of the Adam optimizer is set to 0.001 by default, and for the learning rate scheduler, we adjusted the hyperparameters used by the Adam optimizer. The exponential decay rate for the first-moment estimates (mean) is set to 0.9, and the decay rate for the second-moment estimates (uncentered variance) is set to 0.999. Integrating the Adaptive Moment Estimation as the optimization algorithm, the detailed backpropagation training procedure with six steps in total is elaborated as follows:

Step 1: Input the basic information for the structural analysis, such as the section properties, the boundary conditions, the geometric information, the loading conditions, etc.

Step 2: Assemble the neural network and initialize the internal parameters of the neural network.

Step 3: Randomly place Nc numbers of sampling points along the member and get an approximation from the output layer.

Step 4: Calculate the loss function using Equation (13)

Step 5: If the total loss is smaller than tolerance or the maximum training time has been achieved, terminate the training procedure, and output the approximation.

Step 6: Update the internal parameters of the neural network (w_{ij} and b_{ij}) by optimization algorithms to minimize the loss functions and repeat Steps 3-5.

4.2. Adaptive loss weight algorithm

In Equation (13), all the terms in the total loss function have the same weights, suggesting that the physical laws dictated by the governing differential equations and boundary conditions share the same significance. However, in real cases, one or several terms might be much more significant than the others. For example, for a simply supported beam under bending moment, the bending deformations are more significant than the axial shortening, which means that the loss function given by Equation (15) is more significant than the loss function expressed in Equation (14). Consequently, an adaptive loss weight algorithm proposed by Wang and Teng [62] is employed to improve the accuracy and efficiency of the training procedure. Equation (13) will be revised as,

$$\mathcal{L} = \sum_{i=1}^{NL} w_i \mathcal{L}_i \quad (18)$$

in which, NL stands for the number of terms from the governing differential equations and the boundary conditions in the total loss function, and the w_i are the weights for the terms in the loss function. The specific physical laws and boundary conditions, as given in Equation (7) and

Table 1, respectively, contribute to enhancing numerical stability when implementing the adaptive loss weight algorithm, which was previously reported for the instability issue in the literature [68]. Therefore, the potential instability issue is not significant in this research.

All the loss weights are set to be unity at the beginning of the training process. The update step for w_i in the training procedure can be expressed as follows,

$$w_i^t = \beta w_i^{t-1} + (1 - \beta) w_i^t \quad (19)$$

in which,

$$w_i^t = \frac{NL}{w_i^{t-1}} \frac{\max(|\mathcal{L}_i^t|)}{\sum_{i=1}^{NL} |\mathcal{L}_i^t|} \quad (20)$$

where, the superscript t denotes the t^{th} training time; β is a hyper-parameter and taken as 0.9 in this paper; and \mathcal{L}_i^t represents the gradient of the i^{th} loss term in the t^{th} training time.

4.3. Transfer learning

Despite the proposed MLSA is a mesh-free solution with some intriguing features, the training process may be sometimes time-consuming. The neural network composes of enormous numbers of neurons with a complicated layer structure, and the optimization of the internal parameters of the neural network could take a certain long time. To enhance computational efficiency, the transfer learning technic can be adopted. Transfer learning is a well-established technique that has been widely used in machine learning methods [22-24]. The previously established neural network can be directly applied to similar problems, resulting in substantial savings in training time. Therefore, if a neural network can be properly established and well-trained through a range of problems, this network can be reutilized in subsequent applications with significantly enhanced computational efficiency, which may even achieve instant solutions faster than the traditional matrix-analysis-based structural analysis methods.

5. Comparisons between traditional finite-element-based and proposed machine-learning-based analysis procedures

A comparison of the analysis procedures of the conventional finite-element-based analysis and the proposed MLSA methods is provided in Fig. 6. In the conventional FE-based methods [46-47, 69], after acquiring the basic information, for example, the geometric details, the material properties, and the boundary and loading conditions, the models will mesh into a series of fine-meshed elements. Sequentially, the derived element stiffness matrix formulations will be used to form the stiffness matrix, which will be further used for the generation of nodal displacements and forces.

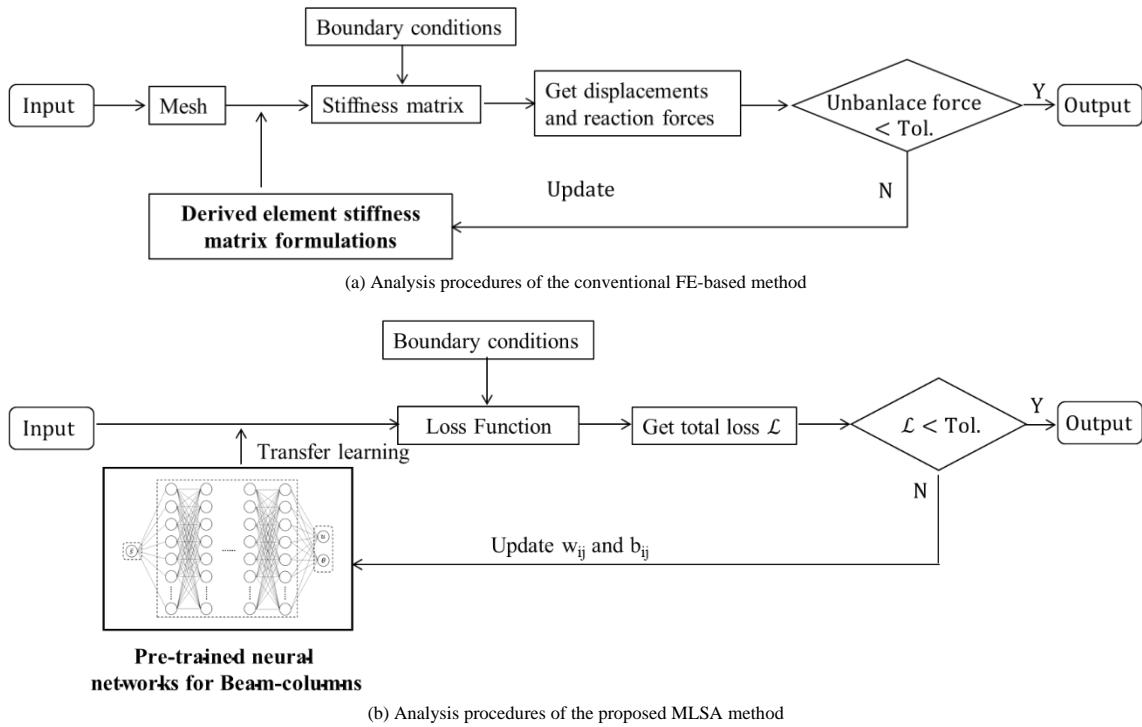


Fig. 6 Analysis procedure

In the proposed MLSA method, the derived element stiffness matrix formulations are not required; instead, the pre-trained neural networks are utilized. The pre-trained neural networks for beam-columns used in this paper were trained using some classic examples provided in the well-recognized textbook [70]. By leveraging the transfer learning method, these pre-trained neural networks will be used to enhance the computational efficiency of the proposed method.

6. Verification examples

Four sets of verification examples are given to validate the practicability and accuracy of the proposed MLSA method. The second-order analysis of members under different boundary and loading conditions is conducted using the proposed method, and the results are compared with those from the closed-form solutions and FEM, where the high-order beam-column elements proposed

by Chan and Zhou [75] are used. The summaries of the analysis parameters and computational expenses for the following examples are provided in Table 4 and Table 5. All training processes and computations are performed on a computer equipped with an i9-12900h CPU and an NVIDIA GeForce 3090 GPU. The number of elements presented in Table 4 specifies the minimum quantity required for accurate analysis of problems exclusively associated with large deflections, as determined through the sensitivity study.

From Table 5, it can be observed that transfer learning is an effective method to reduce the computation time of the MLSA method; but the pre-training of the neural network is time-consuming. Nevertheless, unlike the modeling and mesh generation of the FE model, which is a case-to-case-based procedure, the pre-trained neural network can be a solution for general problems. In this research, the pre-trained neural networks for the second-order analysis of beam-columns are provided, and the practicability and accuracy of which are validated by the following examples.

Table 4

Summary of the analysis parameters for Examples 1 to 4

	FEM [75]			Proposed MLSA Method		
	Number of elements	Number of load steps	Numerical tolerance	Number of collocation points	Number of epochs for pre-training	Numerical tolerance
E.1	5	100	1×10^{-4}	200	10000	1×10^{-4}
E.2	Pin-pinned column	20		100	10000	
	Pin-fixed column	20		100	10000	
E.3	10	1		100	10000	
E.4	10	1		100	10000	

Notes: E. stands for Example

Table 5

Summary of the computational time for Examples 1 to 4

	FEM [75]	Proposed MLSA Method		
	Computational time	Pre-raining time	Computational time (with transfer learning)	Computational time (without transfer learning)
E.1	31.2 s	About 20 minutes	13.32 s	21.66 s
E.2	Pin-pinned column		24.66 s	48.81 s
	Pin-fixed column		13.95 s	41.53 s
E.3	7.54 s		10.26 s	12.46 s
E.4	9.08 s		21.64 s	140.81 s

Notes: E. stands for Example

6.1. Example 1 – Cantilever beams under end moment

A cantilever beam subjected to end moment is studied and loaded to perform large deflection to show the accuracy and capability of the proposed MLSA method. The beam has a length of 4 meters, and the bending stiffness EI of the beam cross-section is assumed to be 1. The bending moments applied on the right end are up to 4π , which can make the member wind twice around itself. This is a classical nonlinear structural analysis problem that has been investigated by several researchers [71-74] during the past decades. Recent studies tried to solve this large-rotation problem using the fine-meshed FEM, where the number of which ranges from three up to a hundred for a single member [71-72], showing the intensive computational expense.


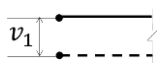
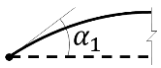
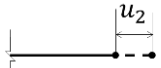

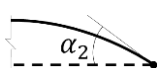
The detailed governing differential equations and boundary equations taken are given in Table 6 for demonstration. The deformed shapes of the members generated from the proposed MLSA method are plotted in Fig. 7, where the closed-form solutions to this problem (circular arcs with radius $\rho = \frac{EI}{M}$) are

also given as the benchmarks. Besides, the results from the FEM using five high-order beam-column elements, proposed by Chan and Zhou [75], are also given for the comparisons.

The results given by the proposed MLSA method using PINN are perfectly in line with those from the closed-form solutions, indicating the proposed method can predict the nonlinear behaviors of the members under pure bending accurately. It should be noted that although the FEM using five beam-column elements can predict the displacements accurately, certain load steps with a number of iterations per step are required (as demonstrated in Table 7) to get accurate predictions leading to a long computational time. Besides, the FEM only gives the displacements at the element nodes, and the displacements between the element nodes need to be calculated approximately based on the assumed element shape functions with some small errors. However, the proposed MLSA method can get accurate member deformations along the whole length by a single attempt of analysis, accurate and efficient.

Table 6

Governing differential equations and boundary equations for Example 1

Governing differential equations	$\frac{\Delta(s)}{ds} = \frac{P(s)}{EA}$, and $\frac{d\alpha(s)}{ds} = -\frac{M(s)}{EI}$		
Governing boundary equations		Fixed	$u(0) = 0$
		Fixed	$v(0) = 0$
		Fixed	$\alpha(0) = 0$
		Free	$\frac{\Delta(s)}{ds} \Big _{s=L} = \frac{F_{x2}\cos\alpha(L) + F_{y2}\sin\alpha(L)}{EA}$
		Free	$\frac{d^2\alpha(s)}{ds^2} \Big _{s=L} = \frac{F_{y2}\cos\alpha(L) - F_{x2}\sin\alpha(L)}{EI}$
		Free	$\frac{d\alpha(s)}{ds} \Big _{s=L} = \frac{M_2}{EI}$

Note: $F_{x2} = 0$, $F_{y2} = 0$, and $M_2 = M$ in this example

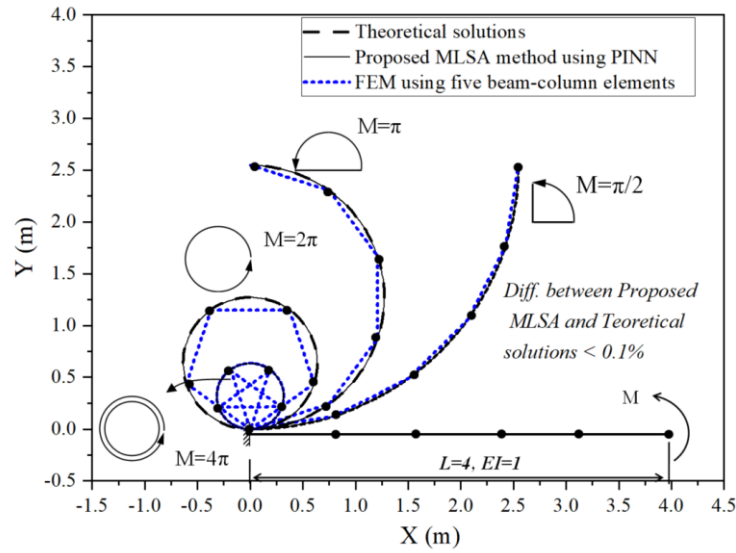


Fig. 7 Deformed shapes of the member under end moments

Table 7

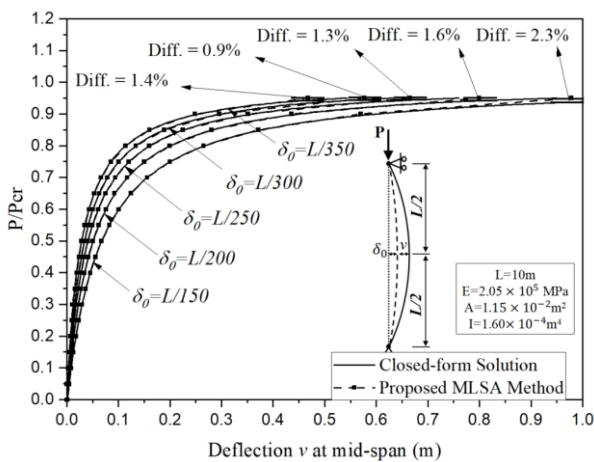
Numerical efficiency of the FEM when analysing the member winding twice around itself

FEM using five beam-column elements			FEM using ten beam-column elements		
Number of incremental load steps	Iterations per step	Total number of iterations	Number of incremental load steps	Iterations per step	Total number of iterations
1-23 steps	Divergent		1-23 steps	Divergent	
24 steps	7	168	24 steps	6	144
50 steps	4	200	50 steps	3	150
100 steps	2	200	100 steps	2	200

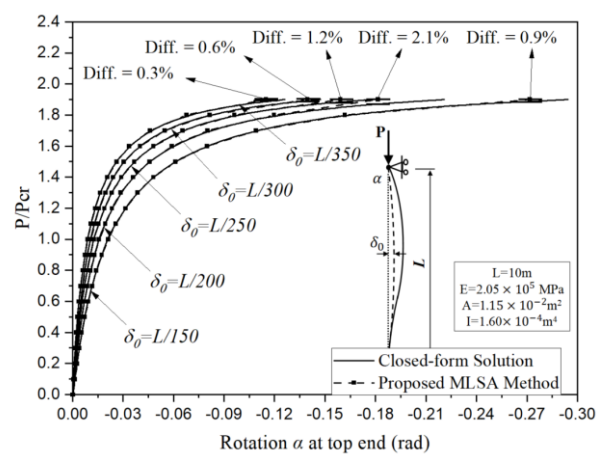
6.2. Example 2 – Imperfect columns under axial compression

This example studies the behaviours of imperfect members under axial compression. The columns with two different boundary conditions, such as the pin-pinned and the pin-fixed boundaries, are investigated. As shown in Fig. 8, the length of the column is 10 meters with the material Young's modulus equals to 205 GPa, and the cross-section area and the second moment of area equal to $1.15 \times 10^{-2} \text{ m}^2$ and $1.60 \times 10^{-4} \text{ m}^4$, respectively. The shapes of member imperfections are all assumed as half sine-wave along the entire member length, and the magnitude of the initial curvature at mid-span δ_0 varies from $L/350$ to $L/150$, as recommended in Eurocode-3 [36].

This example was formerly studied by Liu et al. [76], where an advanced beam-column element is used, and the detailed derivation of the closed-form solutions for this example is available in the reference [76]. The load-displacement curves prior to buckling are generated by the proposed MLSA method, and the comparison results are plotted in Fig. 8, where the results from the closed-form solutions are given as the benchmark. The results given by the proposed MLSA method closely matched those from the closed-form solutions. It confirms the applicability of the proposed MLSA method for accurately analyzing the imperfect column under axial compression.



(a) Pin-pinned column



(b) Pin-fixed column

Fig. 8 Load-displacement curve of the imperfect columns under axial compression

6.3. Example 3 – Cantilever beams under distributed loads

In this example, the behaviours of cantilever beams under distributed loads are investigated. As shown in Fig. 9, four different cases of uniformly and nonuniformly distributed loads are applied to the members to study the large deflection behaviors of the cantilever beams. The length of the beam is 4m, and the bending stiffness EI of the beam cross-section is $1.006 \times 10^3 \text{ kNm}^2$.

The deformed shapes of the cantilever beam under distribution loads are generated and plotted in Fig. 9, where results from the FEM using ten beam-column elements are introduced, serving as the benchmark solutions. From Fig. 9, the member deformations under uniformly and nonuniformly distribution load predicted by the proposed MLSA method using PINN are in line with those from the FEM, showing the accuracy and reliability of the proposed method.

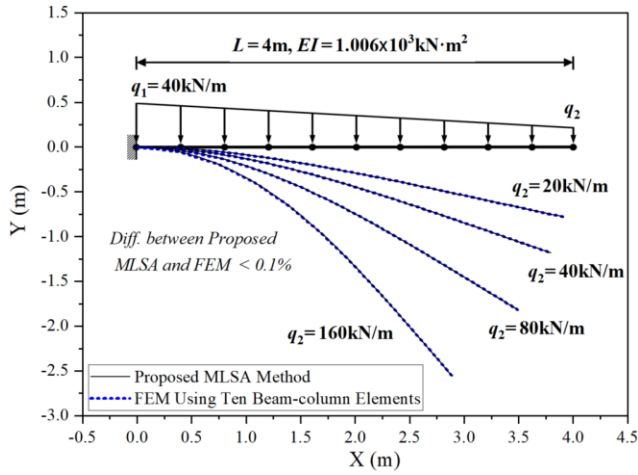


Fig. 9 Deformed shapes of the cantilever beam under distributed loads

6.4. Example 4 - Columns under combined bending and axial loads

In this example, the members with relatively complex loading conditions are investigated, where columns under combined bending and axial loads are analyzed using the proposed MLSA method. Detailed loading and boundary conditions are given in Fig. 10, where the height of the column is 4m, and the bending moment varies from 0.02P to 0.1P. The material is steel with the Young's modulus of 205 GPa. The cross-section area is $5.27 \times 10^{-3} \text{ m}^2$ and the second moment of area about both the principal axes is $4.91 \times 10^{-3} \text{ m}^4$.

The load-displacement curves of the members under combined bending and axial loads generated from the proposed MLSA method are given in Fig. 10, where the results from the FEM using ten beam-column elements are also given for comparison. The overall agreement is very satisfactory, indicating that the proposed MLSA method can predicate the nonlinear behaviors of the members under combined bending and axial loads accurately.

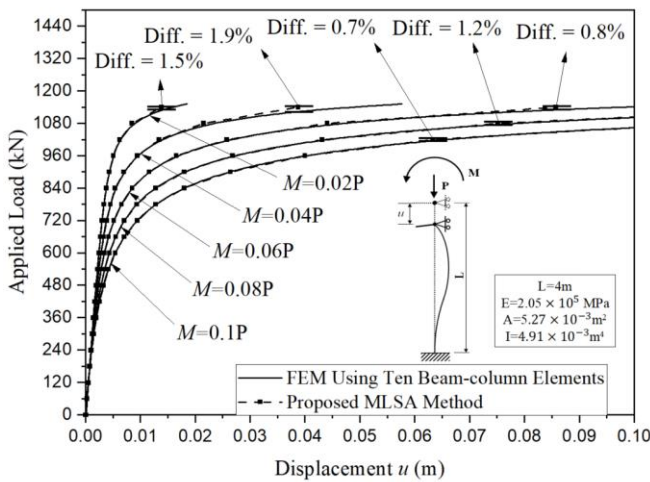


Fig. 10 Load-displacement curves of the column under combined bending and axial loads

References

- [1] Hajela, P. and L. Berke, Neurobiological computational models in structural analysis and design. *Computers & Structures*, 1991. 41(4): p. 657-667.
- [2] Goh, A.T., Prediction of ultimate shear strength of deep beams using neural networks. *Structural Journal*, 1995. 92(1): p. 28-32.
- [3] Kasperkiewicz, J., J. Racz, and A. Dubrawski, HPC strength prediction using artificial neural network. *Journal of Computing in Civil Engineering*, 1995. 9(4): p. 279-284.
- [4] Luo, H. and S.G. Paal, Metaheuristic least squares support vector machine-based lateral strength modelling of reinforced concrete columns subjected to earthquake loads. *Structures*, 2021. 33: p. 748-758.
- [5] El-Kassas, E., R. Mackie, and A. El-Sheikh, Using neural networks in cold-formed steel design. *Computers & Structures*, 2001. 79(18): p. 1687-1696.
- [6] Wu, X., J. Ghaboussi, and J. Garrett Jr, Use of neural networks in detection of structural damage. *Computers & Structures*, 1992. 42(4): p. 649-659.
- [7] Stephens, J.E. and R.D. VanLuchene, Integrated assessment of seismic damage in structures. *Computer-Aided Civil and Infrastructure Engineering*, 1994. 9(2): p. 119-128.
- [8] Hakim, S.J.S. and H.A. Razak, Adaptive neuro fuzzy inference system (ANFIS) and artificial neural networks (ANNs) for structural damage identification. *Structural engineering and mechanics: An international journal*, 2013. 45(6): p. 779-802.
- [9] Huang, H. and H.V. Burton, Classification of in-plane failure modes for reinforced concrete frames with infills using machine learning. *Journal of Building Engineering*, 2019. 25: p. 100767.

7. Conclusion

Modern structural design methods are moving toward the rigorous design of structures through virtual simulations under critical scenarios, complying with physical laws and accounting for the imperfections in reality. The second-order analysis method has been developed in line with the same philosophy, continuously researched in scientific communities, and applied in designing numerous steel structures in engineering practices. The nonlinear finite-element method (FEM) using beam-column elements, also named line finite-elements, has been widely adopted in second-order analysis, which is a matrix analysis-based and computationally intensive method. When adopting the FEM for large deflection problems, numerical convergence is sometimes challenging as well as the long computational time of solving sparse linear equations. Although some research has been conducted during the past decades to improve numerical efficiency, the matrix-based algorithm is unstable when singularity occurs. It may hinder the development of emerging structural analysis methods to be more robust and efficient in satisfying highly demanded requirements in virtual simulations for modern structural design.

The computational efficiency of machine learning has been exceptionally improved in recent years, with the availability of fast AI processors tail-made for massive neural networks. To facilitate the adoption of new AI technology in structural engineering, this research proposes an innovative machine learning-based structural analysis (MLSA) method for the second-order analysis of beam-columns through Physics-Informed Neural Networks (PINN). The governing equations for the problems are discussed and incorporated into the physics-informed neural networks. The physical information provided by the governing equations and boundary conditions is used to orientate the training process creating a self-supervised learning procedure. The PINN framework and the training procedure are provided, where an adaptive loss weight control algorithm and the transfer learning technic are proposed to improve numerical efficiency. The pre-trained neural networks for the second-order analysis of beam-columns are provided.

The new method has been thoughtfully validated by well-established methods in terms of the analytical solutions and the FEM. Four verification examples are given, showing that the proposed MLSA method can obtain the results accurately and robustly. As the potential to be the next-generation structural analysis method, this research innovatively adopts the emerging computational technique based on machine learning to solve second-order analysis problems for single members that could be motivated for other engineering problems.

Data availability

The input and output files for all examples, the computer source codes, and the trained neural networks are available at <https://github.com/zsulsw/mlsa>.

Acknowledgment

The work described in this paper was partially supported by grants from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. PolyU/21E/15203121), and a grant from the National Natural Science Foundation of China (No. 52008410). This work is also partially supported by a grant (BBTH) from Hong Kong Branch of National Rail Transit Electrification and Automation Engineering Technology Research Center.

- [10] Hakim, S.J.S., H.A. Razak, S.A. Ravanfar, and M. Mohammadhassani, Structural damage detection using soft computing method, *Structural Health Monitoring*, Volume 5: Proceedings of the 32nd IMAC, A Conference and Exposition on Structural Dynamics, 2014. Springer International Publishing. p. 143-151.
- [11] Berke, L., S.N. Patnaik, and P.L.N. Murthy, Optimum design of aerospace structural components using neural networks. *Computers & structures*, 1993. 48(6): p. 1001-1010.
- [12] Iranmanesh, A. and A. Kaveh, Structural optimization by gradient-based neural networks. *International journal for numerical methods in engineering*, 1999. 46(2): p. 297-311.
- [13] Kaveh, A., Y. Gholipour, and H. Rahami, Optimal design of transmission towers using genetic algorithm and neural networks. *International Journal of Space Structures*, 2008. 23(1): p. 1-19.
- [14] Tashakor, A. and H. Adeli, Optimum design of cold-formed steel space structures using neural dynamics model. *Journal of Constructional Steel Research*, 2002. 58(12): p. 1545-1566.
- [15] Šipoš, T.K., I. Miličević, and R. Siddique, Model for mix design of brick aggregate concrete based on neural network modelling. *Construction and building materials*, 2017. 148: p. 757-769.
- [16] Jeong, H., J. Bai, C.P. Batuwatta-Gamage, C. Rathnayaka, Y. Zhou, and Y. Gu, A Physics-Informed Neural Network-based Topology Optimization (PINNTO) framework for structural optimization. *Engineering Structures*, 2023. 278, p. 115484.
- [17] Yan, C.A., R. Vescovini, and L. Dozio, A framework based on physics-informed neural networks and extreme learning for the analysis of composite structures. *Computers & Structures*, 2022. 265: p. 106761.

- [18] Raissi, M., P. Perdikaris, and G.E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 2019. 378: p. 686-707.
- [19] Yuan, L., Y.Q. Ni, X.Y. Deng, and S. Hao, A-PINN: Auxiliary physics informed neural networks for forward and inverse problems of nonlinear integro-differential equations. *Journal of Computational Physics*, 2022. 462: p. 111260.
- [20] Zhang, R., Y. Liu, and H. Sun, Physics-informed multi-LSTM networks for metamodeling of nonlinear structures. *Computer Methods in Applied Mechanics and Engineering*, 2020. 369: p. 113226.
- [21] Katsikis, D., A.D. Muradova, and G.E. Stavroulakis, A Gentle Introduction to Physics Informed Neural Networks, with Applications in Static Rod and Beam Problems. *Journal of Advances in Applied & Computational Mathematics*, 2022. 9: p. 103-128.
- [22] Chen, X., C. Gong, Q. Wan, L. Deng, Y. Wan, Y. Liu, B. Chen, and J. Liu, Transfer learning for deep neural network-based partial differential equations solving. *Advances in Aerodynamics*, 2021. 3(1): p. 1-14.
- [23] Goswami, S., C. Anitescu, S. Chakraborty, and T. Rabczuk, Transfer learning enhanced physics informed neural network for phase-field modeling of fracture. *Theoretical and Applied Fracture Mechanics*, 2020. 106: p. 102447.
- [24] Tang, H., H. Yang, Y. Liao, and L. Xie, A transfer learning enhanced the physics-informed neural network model for vortex-induced vibration. *arXiv preprint arXiv:2112.14448*, 2021.
- [25] Basir, S., Investigating and Mitigating Failure Modes in Physics-Informed Neural Networks (PINNs). *arXiv preprint arXiv:2209.09988*, 2022.
- [26] Basir, S. and I. Senocak, Characterizing and Mitigating the Difficulty in Training Physics-Informed Artificial Neural Networks under Pointwise Constraints. *arXiv preprint arXiv:2206.09321*, 2022.
- [27] Li, W., M.Z. Bazant, and J. Zhu, A physics-guided neural network framework for elastic plates: Comparison of governing equations-based and energy-based approaches. *Computer Methods in Applied Mechanics and Engineering*, 2021. 383: p. 113933.
- [28] Bastek, J.H. and D.M. Kochmann, Physics-Informed Neural Networks for Shell Structures. *European Journal of Mechanics-A/Solids*, 2022. 97: p. 104849.
- [29] Muradova, A.D. and G.E. Stavroulakis, Physics-informed neural networks for elastic plate problems with bending and Winkler-type contact effects. *Journal of the Serbian Society for Computational Mechanics*, 2021. 15(2): p. 45-54.
- [30] Haghighat, E., M. Raissi, A. Moure, H. Gomez, and R. Juanes, A deep learning framework for solution and discovery in solid mechanics. *arXiv preprint arXiv:2003.02751*, 2020.
- [31] Haghighat, E. and R. Juanes, SciANN: A Keras/TensorFlow wrapper for scientific computations and physics-informed deep learning using artificial neural networks. *Computer Methods in Applied Mechanics and Engineering*, 2021. 373: p. 113552.
- [32] Tao, F., X. Liu, H. Du, and W. Yu, Physics-informed artificial neural network approach for axial compression buckling analysis of thin-walled cylinder. *AIAA Journal*, 2020. 58(6): p. 2737-2747.
- [33] Wang, T., W.A. Altabay, M. Noori, and R. Ghiasi, A deep learning based approach for response prediction of beam-like structures. *Structural Durability & Health Monitoring*, 2020. 14(4): p. 315.
- [34] Yan, C., R. Vescovini, and L. Dozio, Analysis of Lightweight Structures Using Physics Informed Neural Networks. *Aerospace Europe Conference 2021 (AEC-21)*, 2021. p. 1-7.
- [35] AISI, ANSI/AISC 360-16: specification for structural steel buildings. 2016.
- [36] Standard, Eurocode 3-Design of steel structures. BS EN 1993-1, 2006. 1: p. 2005.
- [37] CoPHK, Code of practice for the structural use of steel 2011. 2011, Buildings Department Hong Kong SAR Government.
- [38] Chen, W.F., Structural stability: from theory to practice. *Engineering structures*, 2000. 22(2): p. 116-122.
- [39] Liew, J.R., H. Chen, N. E. Shanmugam, and W.F. Chen, Improved nonlinear plastic hinge analysis of space frame structures. *Engineering structures*, 2000. 22(10): p. 1324-1338.
- [40] Chen, W.F., Y. Goto, and J.R. Liew, Stability design of semi-rigid frames. 1995: John Wiley & Sons.
- [41] Schafer, B.W. and T. Peköz, Computational modeling of cold-formed steel: characterizing geometric imperfections and residual stresses. *Journal of constructional steel research*, 1998. 47(3): p. 193-210.
- [42] Yu, C. and B.W. Schafer, Simulation of cold-formed steel beams in local and distortional buckling with applications to the direct strength method. *Journal of Constructional Steel Research*, 2007. 63(5): p. 581-590.
- [43] Tang, Y., Y.P. Liu, and S.L. Chan, A co-rotational framework for quadrilateral shell elements based on the pure deformational method. *Advanced Steel Construction*, 2018. 14(1): p. 90-114.
- [44] Chan, S.L. and S. Cho, Second-order analysis and design of angle trusses Part I: Elastic analysis and design. *Engineering Structures*, 2008. 30(3): p. 616-625.
- [45] Du, Z.L., Y.P. Liu, and S.L. Chan, A second-order flexibility-based beam-column element with member imperfection. *Engineering Structures*, 2017. 143: p. 410-426.
- [46] Liu, S.W., R. D. Ziemian, L. Chen, and S.L. Chan, Bifurcation and large-deflection analyses of thin-walled beam-columns with non-symmetric open-sections. *Thin-Walled Structures*, 2018. 132: p. 287-301.
- [47] Chen, L., A.H.A. Abdelrahman, S.W. Liu, R.D. Ziemian, and S.L. Chan, Gaussian Beam-Column Element Formulation for Large-Deflection Analysis of Steel Members with Open Sections Subjected to Torsion. *Journal of Structural Engineering*, 2021. 147(12): p. 04021206.
- [48] Schafer, B.W., Local, distortional, and Euler buckling of thin-walled columns. *Journal of structural engineering*, 2002. 128(3): p. 289-299.
- [49] Ádány, S. and B.W. Schafer, Generalized constrained finite strip method for thin-walled members with arbitrary cross-section: Primary modes. *Thin-Walled Structures*, 2014. 84: p. 150-169.
- [50] Bian, G., K.D. Peterman, S. Torabian, and B.W. Schafer, Torsion of cold-formed steel lipped channels dominated by warping response. *Thin-Walled Structures*, 2016. 98: p. 565-577.
- [51] Howell, L.L. and A. Midha, Parametric deflection approximations for end-loaded, large-deflection beams in compliant mechanisms. 1995.
- [52] Saxena, A. and S.N. Kramer, A simple and accurate method for determining large deflections in compliant mechanisms subjected to end forces and moments. *Journal of Mechanical Design*, 1998. 120(3): p. 392-400.
- [53] Kimball, C. and L.W. Tsai, Modeling of flexural beams subjected to arbitrary end loads. *Journal of Mechanical Design*, 2002. 124(2): p. 223-235.
- [54] Banerjee, A., B. Bhattacharya, and A.K. Mallik, Large deflection of cantilever beams with geometric non-linearity: Analytical and numerical approaches. *International Journal of Non-Linear Mechanics*, 2008. 43(5): p. 366-376.
- [55] Cuomo, S., V.S. Di Cola, F. Giampaolo, G. Rozza, M. Raissi, and F. Piccialli, Scientific Machine Learning through Physics-Informed Neural Networks: where we are and what's next. *Journal of Scientific Computing*, 2022. 92(3): p. 88.
- [56] Katsikadelis, J.T. and G.C. Tsiatas, Large deflection analysis of beams with variable stiffness. *Acta Mechanica*, 2003. 164(1): p. 1-13.
- [57] Strang, G., Piecewise polynomials and the finite element method. *Bulletin of the American Mathematical Society*, 1973. 79(6): p. 1128-1137.
- [58] Pan, S. and K. Duraisamy, Physics-informed probabilistic learning of linear embeddings of nonlinear dynamics with guaranteed stability. *SIAM Journal on Applied Dynamical Systems*, 2020. 19(1): p. 480-509.
- [59] Nascimento, R.G., K. Fricke, and F.A. Viana, A tutorial on solving ordinary differential equations using Python and hybrid physics-informed neural network. *Engineering Applications of Artificial Intelligence*, 2020. 96: p. 103996.
- [60] Pang, G. and G.E. Karniadakis, Physics-informed learning machines for partial differential equations: Gaussian processes versus neural networks, *Emerging Frontiers in Nonlinear Science*. 2020, Springer. p. 323-343.
- [61] Candès, E.J., Harmonic analysis of neural networks. *Applied and Computational Harmonic Analysis*, 1999. 6(2): p. 197-218.
- [62] Wang, S., Y. Teng, and P. Perdikaris, Understanding and mitigating gradient flow pathologies in physics-informed neural networks. *SIAM Journal on Scientific Computing*, 2021. 43(5): p. A3055-A3081.
- [63] Bottou, L., Stochastic gradient descent tricks, *Neural networks: Tricks of the trade*. 2012, Springer. p. 421-436.
- [64] Johnson, R. and T. Zhang, Accelerating stochastic gradient descent using predictive variance reduction. *Advances in neural information processing systems*, 2013. 26.
- [65] Kingma, D.P. and J. Ba, Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [66] Paszke, A., S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, and A. Desmaison, Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 2019. 32.
- [67] Abadi, M., P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, and M. Kudlur, TensorFlow: a system for Large-Scale machine learning. 12th USENIX symposium on operating systems design and implementation (OSDI 16), 2016. 16: p. 265-283.
- [68] Basir, S. and I. Senocak, Physics and equality constrained artificial neural networks: application to forward and inverse problems with multi-fidelity data fusion. *Journal of Computational Physics*, 2022. 463: p. 111301.
- [69] Chen, L., W.L. Gao, S. W. Liu, R.D. Ziemian, and S.L. Chan, Geometric and material nonlinear analysis of steel members with nonsymmetric sections. *Journal of Constructional Steel Research*, 2022. 198: p. 107537.
- [70] Chen, W.F. and T. Atsuta, Theory of beam-columns, volume 2: space behavior and design. Vol. 2. 2007, J. Ross Publishing.
- [71] Zhang, P., J. Ma, M. Duan, Y. Yuan, and J. Wang, A high-precision curvature constrained Bernoulli-Euler planar beam element for geometrically nonlinear analysis. *Applied Mathematics and Computation*, 2021. 397: p. 125986.
- [72] Dadgar-Rad, F. and S. Sahraee, Large deformation analysis of fully incompressible hyperelastic curved beams. *Applied Mathematical Modelling*, 2021. 93: p. 89-100.
- [73] Schulz, M. and F.C. Filippou, Non-linear spatial Timoshenko beam element with curvature interpolation. *International Journal for Numerical Methods in Engineering*, 2001. 50(4): p. 761-785.
- [74] Jirásek, M., E. La Malfa Ribolla, and M. Horák, Efficient finite difference formulation of a geometrically nonlinear beam element. *International Journal for Numerical Methods in Engineering*, 2021. 122(23): p. 7013-7053.
- [75] Chan, S.L. and Z.H. Zhou, Pointwise equilibrating polynomial element for nonlinear analysis of frames. *Journal of structural engineering*, 1994. 120(6): p. 1703-1717.
- [76] Liu, S.W., Y.P. Liu, and S.L. Chan, Direct analysis by an arbitrarily-located-plastic-hinge element—Part 1: Planar analysis. *Journal of Constructional Steel Research*, 2014. 103: p. 303-315.