Research Article

T. N. Nguyen*, Dongsheng Zhang, and Pairod Singhatanadgid*

Fast analysis approach for instability problems of thin shells utilizing ANNs and a Bayesian regularization back-propagation algorithm

https://doi.org/10.1515/nleng-2024-0012
received January 22, 2024; accepted April 22, 2024

Abstract: This research develops a data-driven methodology for structural instability problems with highly nonlinear, difficult, noisy, and small data. A fast analysis and prediction (FAP) approach for instability problems of thin shells is first proposed. This approach contains two phases: the fast numerical analysis and the pure prediction utilizing artificial neural networks (ANNs) incorporated with the Bayesian regularization (B-R) algorithm as follows: (1) in Phase 1 (the fast numerical analysis), post-buckling analysis is conducted utilizing a minor amount of load steps. The load–displacement relation achieved from Phase 1 is not exact because of the small number of load steps utilized; (2) in Phase 2 (the prediction), the loads and deflections achieved from Phase 1 were employed as the data for training ANNs. The trained networks, including the load and displacement networks, were employed to fast predict loads and deflections at any step of the post-buckling analysis. After utilizing Phase 2, a smooth, complete and exact load–displacement curve was achieved. In Phase 1, the available formulation for post-buckling analysis of thin shells in the literature was utilized. Five popular types of instabilities chosen to confirm the effectiveness and exactness of the FAP were snap-through, snap-back, softening–hardening, kink instabilities, and delamination buckling and post-buckling of composites. The high exactness and effectiveness of the FAP were confirmed in the numerical verification section. The present approach saves a huge computation compared to the other ones. It was found that ANNs incorporated with the B-R algorithm have notable advantages compared to numerous neural networks. The proposed approach is applicable to simulations or experiments where data are "expensive", highly nonlinear, difficult, and limited. Utilizing the proposed approach for these fields can dramatically save time and money.

Keywords: fast analysis, post-buckling, Bayesian regularization, artificial neural network, small data

1 Introduction

A structure under a compressive load can present an instability phenomenon. When the compressive load is large enough or reaches the buckling load, the structure can reduce its stiffness, experience a notable change in geometry, and become unstable. When instability occurs, the structure can reduce its load-carrying capacity and is incapable of maintaining a stable equilibrium configuration. Investigations into structural instability and post-buckling behavior are necessary and play an important role in structural engineering, especially when post-buckling behavior is unpredictable. Load–displacement equilibrium paths are usually employed to evaluate structural instability. Several approaches, such as an analytical approach, a semi-analytical approach, and a numerical approach, can be employed to obtain those equilibrium paths. The numerical approach is convenient and powerful. It is applicable to problems with complex loading, boundaries, and geometries. Therefore, it is popularly employed in both academic and industrial problems. Utilizing the numerical approach for solving an instability problem needs a complete combination of an appropriate formulation, a numerical technique, and a nonlinear solver. One of the difficulties of solving instability problems utilizing the numerical approach is high time-consuming because of a huge computation and numerous computational iterations.

* **Corresponding author: T. N. Nguyen,** Department of Mechanical Engineering, Composite Structures Research Unit, Faculty of Engineering, Chulalongkorn University, Patumwan, Bangkok 10330, Thailand, e-mail: tnguyen.chula@gmail.com
* **Corresponding author: Pairod Singhatanadgid,** Department of Mechanical Engineering, Composite Structures Research Unit, Faculty of Engineering, Chulalongkorn University, Patumwan, Bangkok 10330, Thailand, e-mail: pairod.s@chula.ac.th
Dongsheng Zhang: Shanghai Institute of Applied Mathematics and Mechanics, Shanghai Key Laboratory of Mechanics in Energy Engineering, School of Mechanics and Engineering Science, Shanghai University, Shanghai 200444, China, e-mail: donzhang@staff.shu.edu.cn

This reality is clearly seen in large-scale problems utilizing thousands or millions degrees of freedom. Another difficulty in solving instability problems is divergent in some cases, especially when the equilibrium path passes the limit or special points, the inflection points. To overcome the aforementioned difficulties, a lot of approaches, nonlinear solvers, and techniques were proposed to decrease the number of computational iterations and to successfully pass limit or special points on the equilibrium paths including the Newton–Raphson (N-R) technique [1], the modified N-R [2], the modified Riks (M–R) technique [3], the iterative techniques upon an optimization technique [4] and upon residual areas [5], the Koiter (K) technique [6], the accelerated K technique [7], the Koiter-Newton (K-N) technique [8,9], a new technique upon the discretization of governing equations [10], a general path-following methodology [11], the methodology that improves effectiveness of the Newton technique [12], the dynamic relaxation techniques [13,14], a technique without utilizing the predictor [15], and the multi-point techniques [16]. Although numerous nonlinear solvers, approaches, and techniques have been developed and proposed, high time-consuming is still a difficulty in solving instability problems utilizing the numerical approach. To completely overcome this disadvantage, the FAP for instability problems of thin shells is proposed in this work. The present approach includes two phases: the fast analysis utilizing pure isogeometric analysis (IGA) and the prediction utilizing ANNs. In Phase 1, a post-buckling analysis is conducted utilizing *a minor amount of load steps*. In Phase 2, the obtained data from Phase 1 are utilized to train ANNs to fast and completely predict post-buckling behavior of the structure for *many load steps*. Interestingly, the computational cost of the FAP is dramatically lower than that of other ones.

ANNs are computing systems simulated according to the operation of biological neurons in the human brain. An ANN is a group of connected neurons. An ANN can mimic a human brain to fast and simply resolve complex problems that possess nonlinear relationships [17]. Up to now, ANNs have been successfully used in many areas, including bridge deck structure [18], optimal material structure [19], aircraft design [20], prediction of the fracture energy of concrete [21], buckling capacity assessment of steel structures [22], compressive strength prediction of cement [23], an optimization for hydrogen purification performance [24], an optimization for fatigue prediction [25], structural damage identification [26], and damage detection [27]. For nonlinear problems utilizing the numerical methodology, ANNs were employed to improve an algorithm for nonlinear dynamic analysis [28] and the Newton iterative technique [29]. Recently, the FAP for nonlinear bending analysis of structures was proposed [30], in which, N-R technique was applied to solve the nonlinear equation system in Phase 1. In the study by Nguyen *et al.* [30], the approach was proposed and investigated for pure bending problems of plates/shells. As known, loads are priorly known in the bending problems, while displacements are unknown. Thus, ANNs incorporated with the B-R algorithm were only employed to predict series of displacements in Phase 2 in the study by Nguyen *et al.* [30]. In this study, the FAP is proposed for instability problems of thin shells. As known, solving instability problems is much difficult than solving nonlinear bending problems because the equilibrium path can pass the limit or special points, the inflection points. Solving instability problems can be divergent in some cases. Thus, in this study, a powerful nonlinear solver named the modified Riks technique is applied to solve the nonlinear equation system in Phase 1. Besides different from the nonlinear bending problems, loads and displacements are both unknown in the instability problems. Thus, ANNs incorporated with the B-R algorithm are employed to predict the series of displacements and loads in Phase 2. It is emphasized that the present approach employs a minor amount of data points in the training phase. Thus, the time for training a network is very small and can be neglected in evaluating the effectiveness of the proposed approach. The idea of this work is simple in its implementation and can dramatically save computation compared to the another approach. The size of the dataset in this work is relatively small. Both deep neural networks (DNNs) including multiple hidden layers and artificial neural networks (ANNs) including one hidden layer can be used for training. However, we chose ANNs combined with Bayesian regularization back-propagation (BPP) algorithm because of their computational efficiency compared to DNNs. Training a DNN is usually longer than an ANN. This can be explained as follows. DNNs use multiple hidden layers and many neurons on each layer. Thus, there are many computations in each layer, and the weight optimization is more complex and longer. Inversely, ANNs use only one hidden layer, and so time for training is shorter than that for DNNs. However, we recommend using DNNs if the size of dataset is large, for example, 10,000 data points or more. For such big data, DNNs are more accurate than ANNs in predictions. Structure of this article is organized as follows. The next section shows post-buckling analysis of isotropic thin shells utilizing the first-order shear deformation shell theory (FSDT). Section 3 presents the isotropic shell formulation for post-buckling analysis utilizing FSDT and IGA. The FAP for instability problems of thin shells is presented in Section 4. The numerical verification is shown in Section 5. Finally, several notable conclusions are drawn.

2 Post-buckling analysis of isotropic shells utilizing FSDT

The aim of this article is to propose a fast analysis and prediction (FAP) approach for instability problems. Without losing the generality of the proposed approach, shell instabilities were chosen to confirm the effectiveness and exactness of the approach. For the analysis employed in Phase 1, the current formulation was reemployed from a published article on post-buckling analysis of shells. Therefore, a brief formulation of the analysis is presented in this article. The nonlinearity of the shell formulation is established upon the von Karman hypothesis. We now consider an isotropic shell depicted in Figure 1. The shell strain vectors upon the FSDT are

$$\begin{aligned} \boldsymbol{\varepsilon} &= \{\varepsilon_{xx} \quad \varepsilon_{yy} \quad \gamma_{xy}\}^T = \boldsymbol{\varepsilon}_0 + z\boldsymbol{\kappa}_b, \\ \boldsymbol{\gamma} &= \{\gamma_{xz} \quad \gamma_{yz}\}^T = \boldsymbol{\varepsilon}_s, \end{aligned} \tag{1}$$

where

$$\begin{aligned} &\boldsymbol{\varepsilon}_0 = \boldsymbol{\varepsilon}_L + \boldsymbol{\varepsilon}_N; \quad \boldsymbol{\varepsilon}_L = \begin{Bmatrix} u_{0,x} + \dfrac{w_0}{R} \\ v_{0,y} \\ u_{0,y} + v_{0,x} \end{Bmatrix}; \quad \boldsymbol{\varepsilon}_N = \frac{1}{2}\begin{Bmatrix} {w_{0,x}}^2 \\ {w_{0,y}}^2 \\ 2w_{0,xy} \end{Bmatrix}; \\ &\boldsymbol{\kappa}_b = \begin{Bmatrix} \beta_{x,x} \\ \beta_{y,y} \\ \beta_{x,y} + \beta_{y,x} \end{Bmatrix}; \quad \boldsymbol{\varepsilon}_s = \begin{Bmatrix} -\dfrac{u_0}{R} + w_{0,x} + \beta_x \\ w_{0,y} + \beta_y \end{Bmatrix}, \end{aligned} \tag{2}$$

$\boldsymbol{\varepsilon}_N$ is defined as the nonlinear strain vector and can be rewritten as

$$\boldsymbol{\varepsilon}_N = \frac{1}{2}\mathbf{A}\boldsymbol{\theta}; \quad \text{and} \quad \mathbf{A} = \begin{bmatrix} w_{0,x} & 0 \\ 0 & w_{0,y} \\ w_{0,y} & w_{0,x} \end{bmatrix}; \quad \boldsymbol{\theta} = \begin{Bmatrix} w_{0,x} \\ w_{0,y} \end{Bmatrix}, \tag{3}$$

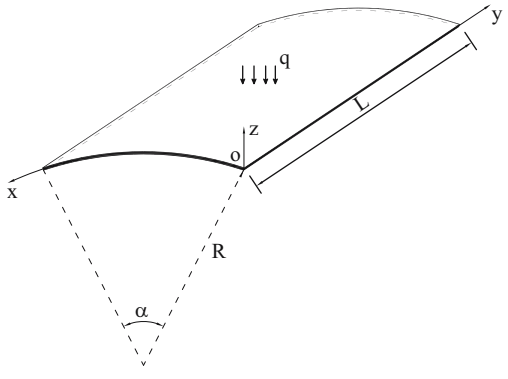


Figure 1: Panel subjected to a radial pressure.

where Ω is the initial shell configuration. In the total Lagrangian methodology, the virtual work equation is expressed as

$$\int_{\Omega} \hat{\boldsymbol{\sigma}}^T \delta\hat{\boldsymbol{\varepsilon}} \mathrm{d}\Omega = \int_{\Omega} \delta\bar{\mathbf{u}}^T \mathbf{f}_s \mathrm{d}\Omega, \tag{4}$$

where $\mathbf{f}_s = \{f_x \; f_y \; f_z\}^T$ is the external load vector. The vector $\bar{\mathbf{u}}^T = \{u_x \; u_y \; u_z\}$, while $u_z = w_0$ and w_0 is the radial displacement. If an isotropic shell is only under a radial load $f_z = \lambda f_0$, the virtual work equation is re-expressed as

$$\int_{\Omega} \hat{\boldsymbol{\sigma}}^T \delta\hat{\boldsymbol{\varepsilon}} \mathrm{d}\Omega = \lambda \int_{\Omega} \delta w_0 f_0 \mathrm{d}\Omega. \tag{5}$$

λ is known as the load factor. $\hat{\boldsymbol{\sigma}}$ is known as the stress resultant vector determined as

$$\hat{\boldsymbol{\sigma}} = \{\boldsymbol{\sigma}_p \quad \boldsymbol{\sigma}_b \quad \boldsymbol{\sigma}_s\}^T, \tag{6}$$

where the in-plane stress vector is

$$\boldsymbol{\sigma}_p = \{N_x \quad N_y \quad N_{xy}\}^T = \left\{ \int_{-h/2}^{h/2} (\sigma_x \quad \sigma_y \quad \tau_{xy}) \mathrm{d}z \right\}^T. \tag{7}$$

The bending and shear stress vectors, respectively, are

$$\boldsymbol{\sigma}_b = \{M_x \quad M_y \quad M_{xy}\}^T = \left\{ \int_{-h/2}^{h/2} (\sigma_x \quad \sigma_y \quad \tau_{xy}) z \mathrm{d}z \right\}^T, \tag{8}$$

$$\boldsymbol{\sigma}_s = \{Q_x \quad Q_y\}^T = \left\{ \int_{-h/2}^{h/2} (\tau_{xz} \quad \tau_{yz}) \mathrm{d}z \right\}^T, \tag{9}$$

For isotropic shells, the generalized strain vector $\hat{\boldsymbol{\varepsilon}}$ is related to the stress resultant vector $\hat{\boldsymbol{\sigma}}$ through Hooke's law as

$$\begin{aligned} &\hat{\boldsymbol{\sigma}} = \hat{\boldsymbol{D}}\hat{\boldsymbol{\varepsilon}}; \quad \hat{\boldsymbol{D}} = \begin{bmatrix} \mathbf{D}^p & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}^b & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{D}^s \end{bmatrix}; \\ &\hat{\boldsymbol{\varepsilon}} = \begin{Bmatrix} \boldsymbol{\varepsilon}_L \\ \boldsymbol{\kappa}_b \\ \boldsymbol{\varepsilon}_s \end{Bmatrix} + \begin{Bmatrix} \boldsymbol{\varepsilon}_N \\ \mathbf{0} \\ \mathbf{0} \end{Bmatrix}, \end{aligned} \tag{10}$$

where

$$\begin{aligned} &\mathbf{D}^p = \frac{Eh}{1-\upsilon^2}\bar{\mathbf{D}}; \quad \mathbf{D}^b = \frac{Eh^3}{12(1-\upsilon^2)}\bar{\mathbf{D}}; \\ &\mathbf{D}^s = \kappa\frac{Eh}{2(1+\upsilon)}\mathbf{I}, \end{aligned} \tag{11}$$

$$\bar{\mathbf{D}} = \begin{bmatrix} 1 & \upsilon & 0 \\ \upsilon & 1 & 0 \\ 0 & 0 & (1-\upsilon)/2 \end{bmatrix}; \quad \mathbf{I} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \tag{12}$$

where $\kappa = 5/6$ is the shear correction factor, h is the thickness, E is Young's modulus, and ν is Poisson's ratio of shells.

3 Isotropic shell formulation for post-buckling analysis utilizing FSDT and IGA

This part briefly presents the post-buckling analysis of isotropic shells upon FSDT and IGA. As introduced earlier, the detailed formulation can be found in the study for the post-buckling analysis of functionally graded carbon nanotube-reinforced composite (FG-CNTRC) shells utilizing FSDT and IGA. As known, an isotropic shell is a separate case of an FG-CNTRC shell. Therefore, the formulation for isotropic shells is the same as for FG-CNTRC shells, except there is a small difference in determining the matrices for material presented in Eqs (10)–(12). At the mth load increment and the ith iteration, the system of linear incremental equations is

$$\mathbf{K}_T(\mathbf{q}_m)\Delta^i\mathbf{q}_m = {}^i\mathbf{F}_{\text{ext},m} - {}^i\mathbf{F}_{\text{int},m}, \tag{13}$$

where the tangent stiffness matrix is

$$\begin{aligned}\mathbf{K}_T = &\int_\Omega \left[\begin{Bmatrix}\mathbf{B}_A^L\\ \mathbf{B}_A^b\\ \mathbf{B}_A^s\end{Bmatrix} + \begin{Bmatrix}\mathbf{B}_A^N\\ \mathbf{0}\\ \mathbf{0}\end{Bmatrix}\right]^T \begin{bmatrix}\mathbf{D}^p & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{D}^b & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{D}^s\end{bmatrix}\left[\begin{Bmatrix}\mathbf{B}_A^L\\ \mathbf{B}_A^b\\ \mathbf{B}_A^s\end{Bmatrix} + \begin{Bmatrix}\mathbf{B}_A^N\\ \mathbf{0}\\ \mathbf{0}\end{Bmatrix}\right]\mathrm{d}\Omega\\ &+ \int_\Omega (\mathbf{B}_A^g)^T \begin{bmatrix}N_x & N_{xy}\\ N_{xy} & N_y\end{bmatrix}\mathbf{B}_A^g \mathrm{d}\Omega,\end{aligned} \tag{14}$$

and the load vector

$$\begin{aligned}{}^i\mathbf{F}_{\text{ext},m} &= ({}^i\lambda_m + \Delta^i\lambda_m)\int_\Omega f_0\{0 \quad 0 \quad N_A \quad 0 \quad 0\}^T \mathrm{d}\Omega\\ &= ({}^i\lambda_m + \Delta^i\lambda_m)\mathbf{F}_0,\end{aligned} \tag{15}$$

where $\mathbf{F}_0$ is the referenced load vector, and N_A is the nonuniform rational B-spline basic function. Besides, the internal force vector can be computed as

$${}^i\mathbf{F}_{\text{int},m} = {}^i\mathbf{K}_m{}^i\mathbf{q}_m, \tag{16}$$

where

$${}^i\mathbf{K}_m = \int_\Omega \left[\begin{Bmatrix}\mathbf{B}_A^L\\ \mathbf{B}_A^b\\ \mathbf{B}_A^s\end{Bmatrix} + \begin{Bmatrix}\mathbf{B}_A^N\\ \mathbf{0}\\ \mathbf{0}\end{Bmatrix}\right]^T \begin{bmatrix}\mathbf{D}^p & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{D}^b & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{D}^s\end{bmatrix}\left[\begin{Bmatrix}\mathbf{B}_A^L\\ \mathbf{B}_A^b\\ \mathbf{B}_A^s\end{Bmatrix} + 0.5\begin{Bmatrix}\mathbf{B}_A^N\\ \mathbf{0}\\ \mathbf{0}\end{Bmatrix}\right]\mathrm{d}\Omega. \tag{17}$$

In this article, the modified Riks technique [3] is employed to solve the system of linear incremental equations. With an arbitrary load increment, an iterative process is performed until the following convergence criterion is satisfied as

$$e = \frac{\|{}^i\lambda_m\mathbf{F}_0 - {}^i\mathbf{F}_{\text{int},m}\|}{\|({}^i\lambda_m + \Delta^i\lambda_m)\mathbf{F}_0\|} < 10^{-6}. \tag{18}$$

Once the incremental solutions are achieved from Eq. (13), the load factor λ and the displacement vector $\mathbf{q}$ of the iteration are updated as [3]

$$\begin{aligned}{}^{i+1}\lambda_m &= {}^i\lambda_m + \Delta^i\lambda_m,\\ {}^{i+1}\mathbf{q}_m &= {}^i\mathbf{q}_m + \Delta^i\mathbf{q}_m,\\ \Delta^i\mathbf{q}_m &= \Delta^i\mathbf{q}_{R,m} + \Delta^i\lambda_m\mathbf{q}_{F,m},\end{aligned} \tag{19}$$

where $\mathbf{q}_{F,m}$ is the displacement produced by the reference force. $\Delta^i\mathbf{q}_{R,m}$ is another displacement produced by the residual load determined as

$$\begin{aligned}\Delta^i\mathbf{q}_{R,m} &= [\mathbf{K}_T(\mathbf{q}_m)]^{-1}({}^i\lambda_m\mathbf{F}_0 - {}^i\mathbf{F}_{\text{int},m}),\\ \mathbf{q}_{F,m} &= [\mathbf{K}_T(\mathbf{q}_m)]^{-1}\mathbf{F}_0.\end{aligned} \tag{20}$$

4 FAP approach for instabilities of thin shells

4.1 Brief introduction to ANNs

ANNs are systems simulated according to the operation of biological neurons in the human brain [21,31]. An ANN is a group of connected nodes as depicted in Figure 2(a). In this network, signals are spread from these neurons to those neurons via connections. Each neuron in the network has a weight adjusted during the learning process. The weight may decrease and increase the signal strength at the connection. During the learning process, the signals are transmitted from the first layer to the last layer numerous times until optimization of all the weights is achieved. The neuron's output is computed via the function (f), which is sum of the neuron's input, bias, and weight, as depicted in Figure 2(b). The formula is [21]

$$y = f\left(\sum_{i=1}^{n} w_i x_i + b\right), \tag{21}$$

where w_i and x_i, respectively, are the weight and input of the ith neuron connected to the considered one and b is a bias.

ANNs are established upon the feed-forward BPP algorithm. In application, the input data together with the desired output data are first provided for ANNs. Then, the neuron's weight and bias are modified (once or many

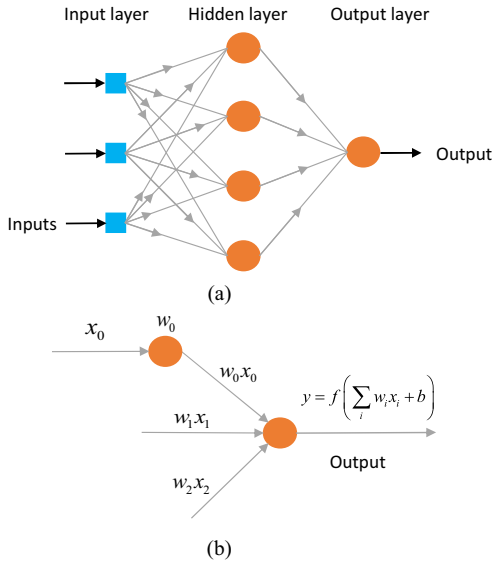


Figure 2: Typical architecture and output calculation of an ANN: (a) architecture of a typical ANN and (b) calculating output for a neuron with three inputs.

times) to decrease the distinction between the actual and desired outputs of the network. The target of training ANNs is to minimize network error (the distinction between the actual output and the desired output of the network) or optimize the performance of the networks in predictions. The training process can end when the stopping criterion upon the mean square error (MSE) is met. The MSE is calculated as [21,31]

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(e_i)^2 = \frac{1}{N}\sum_{i=1}^{N}(t_i - y_i)^2, \tag{22}$$

where N is known as the number of samples. For the ith sample, y_i is the predicted value (the output of the network) and t_i is the target value. In this article, networks are trained utilizing two layers, which are the hidden layer and the output layer. The activation functions f are the PURELIN function for the output layer, while the TANSIG function for the hidden one. These activation functions are written as follows [21,31]:

$$\begin{aligned} f &= \text{tansig}(x) = \frac{2}{1 + e^{-2x}} - 1 \\ f &= \text{purelin}(x) = x. \end{aligned} \tag{23}$$

Several optimization techniques can be employed to speed up the convergence of the BPP algorithm including the scaled conjugate gradient (SCG) technique, the B-R technique, and the Levenberg–Marquardt (L-M) technique. The details of these optimization techniques can be found in previous studies [21,31]. For all the networks in this article, the B-R technique is utilized.

4.2 B-R BPP algorithm

The objective function of ANNs utilizing B-R is written as [32]

$$F = \beta E_D + \alpha E_W, \tag{24}$$

where E_D is the sum of squared errors, while E_W is the sum of squared weights. α and β are known as the parameters of the objective function. The network weights are the random variables when the B-R is employed. After collecting data, the density function for the weights is computed as

$$P(\mathbf{w}|D, \alpha, \beta, M) = \frac{P(D|\mathbf{w}, \beta, M)P(\mathbf{w}|\alpha, M)}{P(D|\alpha, \beta, M)}, \tag{25}$$

where $\mathbf{w}$ is the weight vector, M is the employed model while D represents the dataset. $P(\mathbf{w}|\alpha, M)$ is the prior density, and $P(D|\mathbf{w}, \beta, M)$ is the likelihood function. $P(D|\alpha, \beta, M)$ is the normalization factor that ensures that the total probability is one. If we assume that both the noise in the training data and the noise in the prior distribution for the weights are Gaussian, the probability densities are written as

$$\begin{aligned} P(D|\mathbf{w}, \beta, M) &= \frac{1}{Z_D(\beta)}\exp(-\beta E_D), \\ P(\mathbf{w}|\alpha, M) &= \frac{1}{Z_W(\alpha)}\exp(-\alpha E_W), \end{aligned} \tag{26}$$

where $Z_D(\beta) = (\pi/\beta)^{N/2}$ and $Z_W(\alpha) = (\pi/\alpha)^{N/2}$. Substituting Eq. (26) into Eq. (25), we achieve

$$P(\mathbf{w}|D, \alpha, \beta, M) = \frac{1}{Z_F(\alpha, \beta)}\exp(-F(\mathbf{w})). \tag{27}$$

In the B-R BPP algorithm, the optimal weights are found when the posterior probability $P(\mathbf{w}|D, \alpha, \beta, M)$ is maximized. Maximizing the posterior probability $P(\mathbf{w}|D, \alpha, \beta, M)$ is equivalent to minimizing the objective function $F = \beta E_D + \alpha E_W$. Now, we apply Bayes' rule to optimize the parameters α and β of the objective function. We have

$$P(\alpha, \beta|D, M) = \frac{P(D|\alpha, \beta, M)P(\alpha, \beta|M)}{P(D|M)}. \tag{28}$$

We assume that the prior density $P(\alpha, \beta|M)$ is uniform for the parameters α and β of the objective function. Maximizing the posterior probability can be achieved by maximizing the likelihood function $P(D|\alpha, \beta, M)$. As seen, the likelihood function is the normalization factor in Eq. (25). The normalization factor is calculated from Eq. (25) as

$$P(D|\alpha, \beta, M) = \frac{P(D|\mathbf{w}, \beta, M)P(\mathbf{w}|\alpha, M)}{P(\mathbf{w}|D, \alpha, \beta, M)}. \tag{29}$$

Note that all the probabilities have a Gaussian form. Thus, Eq. (29) can be re-expressed as

$$P(D|\alpha, \beta, M) = \frac{Z_F(\alpha, \beta)}{Z_D(\beta)Z_W(\alpha)}, \tag{30}$$

where $Z_D(\beta)$ and $Z_W(\alpha)$ are the constants and defined before. $Z_F(\alpha, \beta)$ can be computed utilizing a Taylor series expansion as

$$Z_F \approx (2\pi)^{N/2}(\det((\mathbf{H})^{-1}))^{1/2} \exp(-F(\mathbf{w})). \tag{31}$$

$\mathbf{H} = \beta\nabla^2 E_D + \alpha\nabla^2 E_W$ is known as the Hessian matrix. Placing the aforementioned result in Eq. (30), we obtain the optimal values for α and β as

$$\alpha = \frac{\gamma}{2E_W(\mathbf{w})} \quad \text{and} \quad \beta = \frac{N-\gamma}{2E_D(\mathbf{w})}, \tag{32}$$

where $\gamma = N - 2\alpha\text{tr}(\mathbf{H})^{-1}$ is the effective number of parameters, while N is the total number of parameters in the network. Finally, optimization of the parameters α and β is performed through some of the following steps [32]:

- (**0**) Initialize the weights w, the parameters α and β.
- (**1**) Employ the L-M algorithm for finding the minimum of the objective function.
- (**2**) Calculate $\gamma = N - 2\alpha\text{tr}(\mathbf{H})^{-1}$. Herein, utilize the Gauss-Newton approximation for the Hessian matrix as follows: $\mathbf{H} = \nabla^2 F(\mathbf{w}) \approx 2\beta\mathbf{J}^T\mathbf{J} + 2\alpha\mathbf{I}_N$.
- (**3**) Compute new values of α and β utilizing Eq. (32).
- (**4**) Steps 1–3 are iterated until we obtain the convergence.

In Step **1**, the weight is updated after each iteration of the L-M algorithm as

$$w_{i+1} = w_i - (\mathbf{J}_i^T\mathbf{J}_i + \mu_i\mathbf{I})^{-1}\mathbf{J}_i^T e_i, \tag{33}$$

where $\mathbf{I}$ is the unit matrix, e is the error factor, and μ represents the learning parameter. The global minimum of the objective function is reached when its value does not dramatically change in subsequent iterations.

4.3 FAP approach for instability problems of thin shells

Some popular types of thin shell instabilities chosen to confirm the effectiveness and exactness of the FAP were

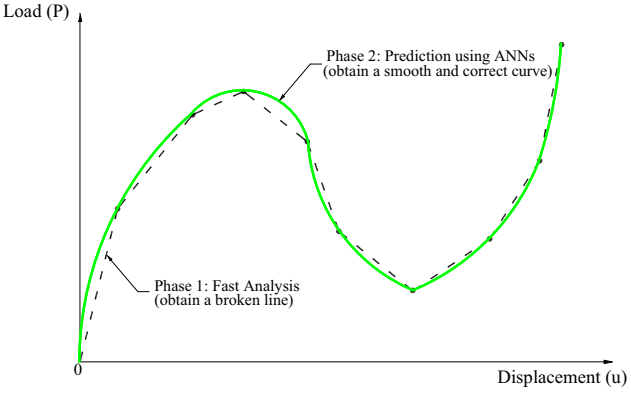


Figure 3: FAP for instability problems utilizing ANNs.

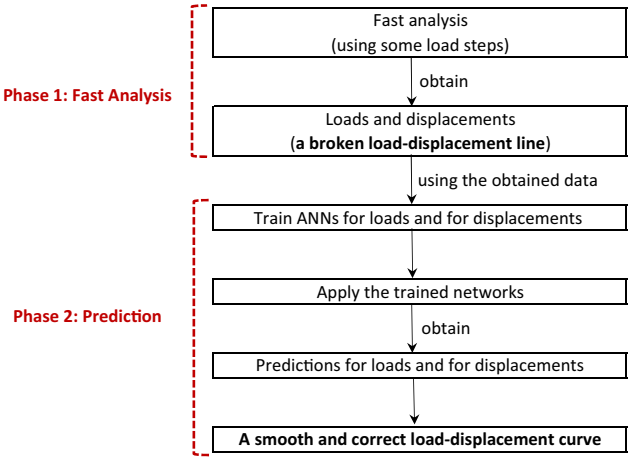


Figure 4: Workflow of the present methodology for instability problems.

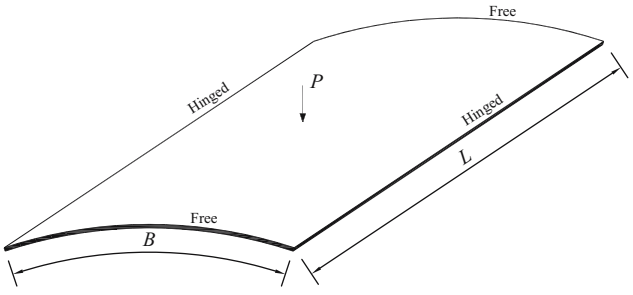


Figure 5: Hinged-hinged panel under a point load at the center.

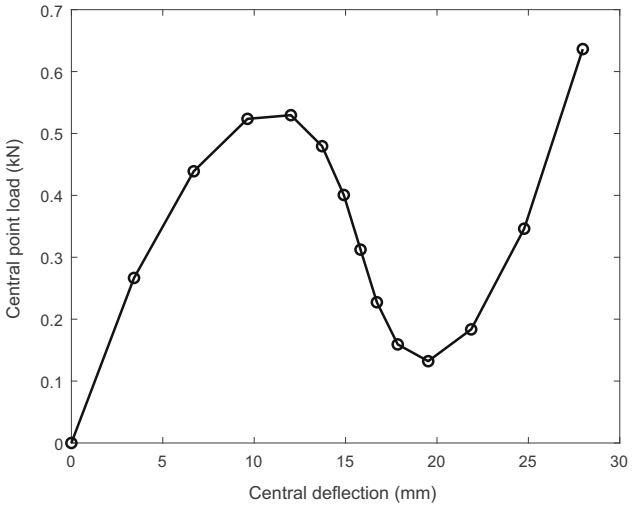


Figure 6: Analysis of the hinged-hinged panel utilizing the FAP: Phase 1 (fast analysis, 13 load steps), h = 12.7 mm.

snap-through, snap-back, kink, and softening–hardening instabilities. One of the great difficulties of solving structural instability problems is highly time-consuming because of numerous computational iterations and a huge computation. This reality is clearly seen in large-scale problems utilizing thousands or millions of degrees of freedom. Another difficulty in solving instability problems is divergence in many cases especially when the equilibrium path passes the limit points, the inflection points, the neighborhoods of turning points, and bifurcations. To overcome the aforementioned difficulties, the FAP for instability problems of thin shells is proposed in this article. It is noted that in instability or post-buckling analysis of structures, loads and displacement vectors of analysis steps are unknown. The present methodology contains two phases: fast numerical analysis and pure prediction utilizing ANNs as follows: (1) for Phase 1, post-buckling analysis is conducted utilizing a minor amount of load steps. The obtained load–displacement relation is not exact because of a minor amount of load steps utilized; (2) for Phase 2, the loads and the deflections obtained from Phase 1 are considered as the data for training networks. Then, the trained networks, including the load network and the displacement network, are utilized to fast predict loads and deflections at any step of the post-buckling analysis. Upon utilizing Phase 2, an exact and smooth load–displacement curve is achieved. Figures 3 and 4, respectively present the description and the flowchart of the FAP. The numerical verification section demonstrates the high exactness and effectiveness of the FAP for instability problems. Besides, the high ability of the FAP to

Phase 1

Fast analysis using 13 load steps
obtain
13 central deflections

Phase 2

Train network

	Input*	Target
Training data	1	0.000
	2	3.427
	3	6.694
	4	9.634
	5	12.009
	6	13.716
	7	14.903
	8	15.818
	9	16.712
	10	17.852
	11	19.521
Testing data	12	21.871
	13	24.771
	14	27.977

Apply the trained network

Input**	Output	Target***
1.000	8.89E-05	0.000
1.419	1.450	1.429
1.839	2.882	2.839
2.258	4.289	4.227
2.677	5.665	5.582
3.097	6.997	6.895
3.516	8.268	8.150
3.935	9.458	9.330
4.355	10.550	10.416
4.774	11.529	11.395
5.194	12.388	12.257
5.613	13.129	13.003
6.032	13.761	13.644
6.452	14.302	14.193
6.871	14.770	14.669
7.290	15.185	15.089
7.710	15.566	15.470
8.129	15.931	15.831
8.548	16.297	16.189
8.968	16.682	16.562
9.387	17.108	16.968
9.806	17.597	17.429
10.226	18.173	17.964
10.645	18.854	18.594
11.065	19.654	19.335
11.484	20.576	20.195
11.903	21.615	21.172
12.323	22.756	22.256
12.742	23.983	23.427
13.161	25.275	24.669
13.581	26.612	25.963
14.000	27.977	27.297

Note:
Fast analysis using 13 load steps and then pure prediction for 31 load steps

* : For convenience and simplicity, we assume input data [1~14] corresponding to 13 load steps

**: Increment: (14-1)/n = 0.419; n=31 corresponding to 31 load steps

***: Obtained from full and pure analysis using 31 load steps

Figure 7: Data structure of the first and second phases for displacement prediction.

detect limit and inflection points is also confirmed in the next section.

5 Numerical verification

The effectiveness and exactness of the present approach for instability problems of thin shells are verified in this section. Without losing the generality of the proposed approach, four popular types of thin shell instabilities chosen to confirm the effectiveness and exactness of the present approach were snap-through, snap-back, kink and softening–hardening instabilities. For convenience and consistency, the following parameters are fixed for the whole manuscript.

- For the analysis in Phase 1, the computational models are analyzed utilizing 14 × 14 cubic elements. The hinged boundary is described as $u_0 = v_0 = w_0 = \beta_y = 0$, at $x = 0, \alpha R$. The clamped boundary is imposed as $u_0 = v_0 = w_0 = \beta_x = \beta_y = 0$. Besides, the geometric data of shells are provided as the thickness h, and the length $L = B = 508$ mm. The radius is $R = 2{,}540$ mm. Note that the thickness h differs in the first three problems in this article. The material properties of all shells are given as Young's modulus $E = 3.103$ kN/mm^2 and Poisson's ratio $v = 0.3$.
- For the prediction packages in Phase 2, the structure of all ANNs includes the hidden and output layers. The number of neurons in the first and second layers is 15 and 1, respectively. ANNs utilizing the B-R BPP algorithm did not require a validation dataset to evaluate the models. The algorithm has

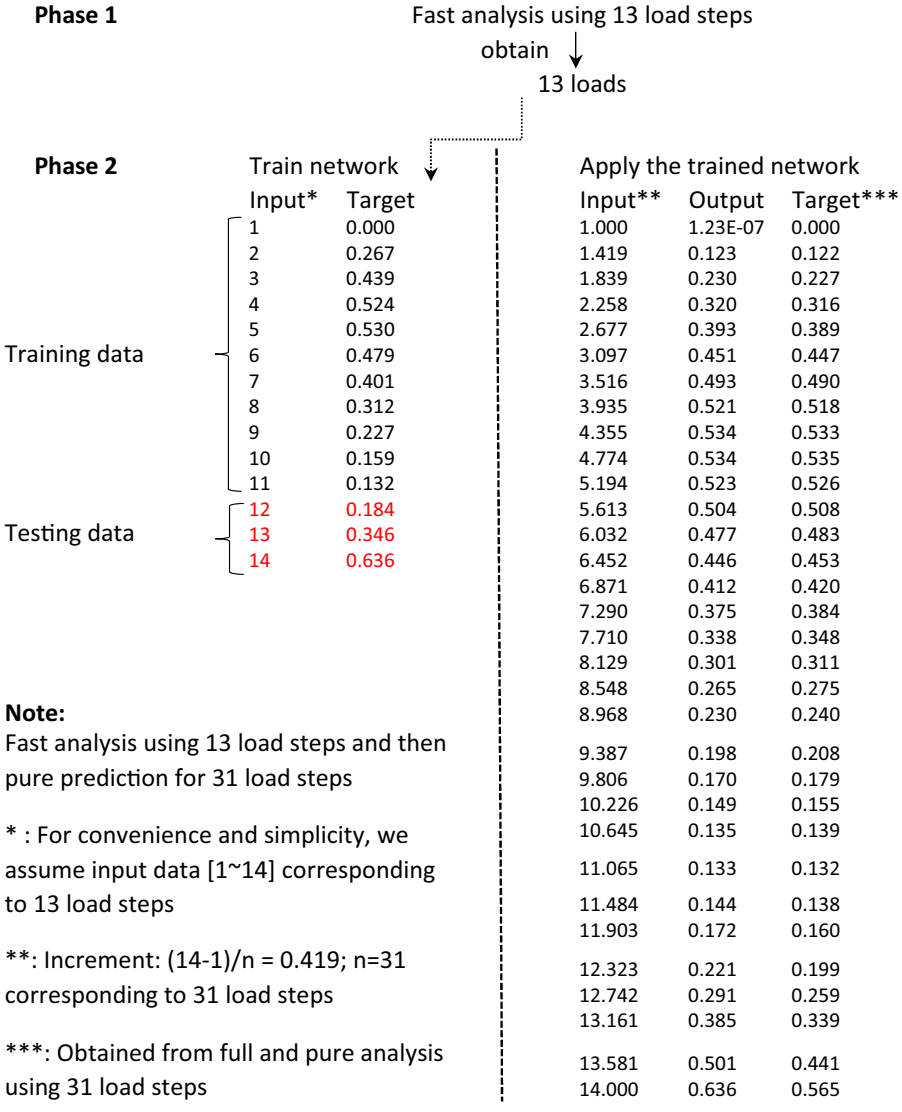


Figure 8: Data structure of the first and second phases for load prediction.

a built-in validation technique. Therefore, in the training phase, the dataset was separated into two data groups: the training data (85%) and the testing data (15%). Utilizing the aforementioned parameters, we proved numerically that all the following networks are well trained and performed in all the problems. Notably, the cost for both training and application of an ANN is very low because of the very small datasets are employed together with the advantages of ANNs. Thus, it can be neglected in the effectiveness computation of the FAP. This fact is mentioned in the first problem.

- All the networks were trained utilizing the NN-fitting application built in MATLAB R2021a. The ANN's parameters and architecture are listed in table 1 in the study by Nguyen *et al.* [30].

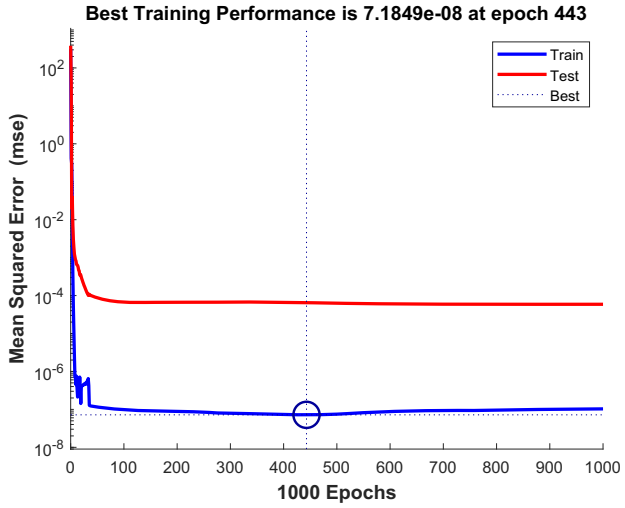


Figure 9: Training performance of ANNs in Phase 2: (a) for displacement network and (b) for load network.

For verification study, the results of the FAP are compared with that of pure IGA and reference solutions. The pure IGA is conducted utilizing the same formulation as the FAP but needs much more load steps.

5.1 FAP approach for snap-through instability problems of thin shells

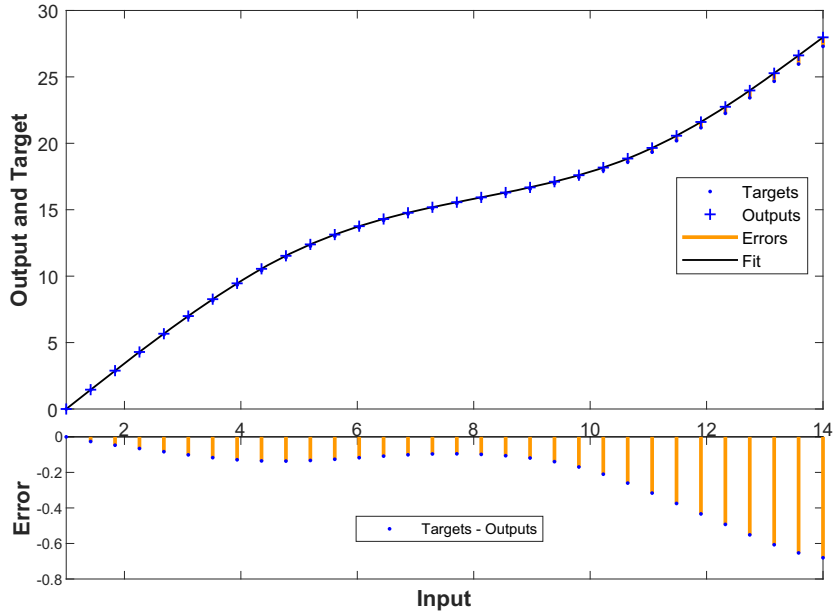
The effectiveness and exactness of the FAP for snap-through instability problems of shells and the performance of ANNs incorporated with the B-R are studied in this subsection. We consider a hinged-hinged isotropic cylindrical panel under a central point load, as in Figure 5. The geometric data and the material properties of the panel are provided at the beginning of this section. Note that in this problem, the panel thickness is h = 12.7 mm. In Phase 1, we perform a fast post-buckling analysis of the thin isotropic panel utilizing 13 equal load steps. The obtained load–displacement relation is shown in Figure 6. This relation is not exact because of a minor amount of load steps utilized. To achieve a smooth, complete, and correct load–displacement curve, we must apply Phase 2 of the FAP utilizing ANNs. We need to build up two ANNs in Phase 2. They are the networks for displacement prediction and load prediction. The central deflections and loads obtained from Phase 1 are considered as *target data* for training ANNs in Phase 2. Data structures of Phases 1 and 2 for displacement and load predictions are, respectively, presented in Figures 7 and 8. In Phase 2, the training process and the application of the displacement network are the same as that of the load network. These processes are explained in detail below.

- For the training process of the displacement network, the data sizes of the input, output, and target are equal. The number of load steps in Phase 1 is 13. Thirteen achieved deflections are utilized as the *target data* for training the displacement network. For training networks, we utilize a typical input data as [1, 2, …,14] in order that the sizes of the input data and target data are equal as in Figure 7 (the left side). Notably, this typical input data is employed for simplicity and convenience without losing of the generality and exactness of the FAP. After that, the displacement network is trained utilizing the network parameters provided at the beginning of Section 5.
- For the application of the displacement network: if we need to predict full deflections of *31 equal load steps* utilizing the trained ANN, the input is still from 1 to 14 but the increment in the data is $(14 - 1)/n = 0.419$. In which, n = 31 corresponds to 31 load steps in order to

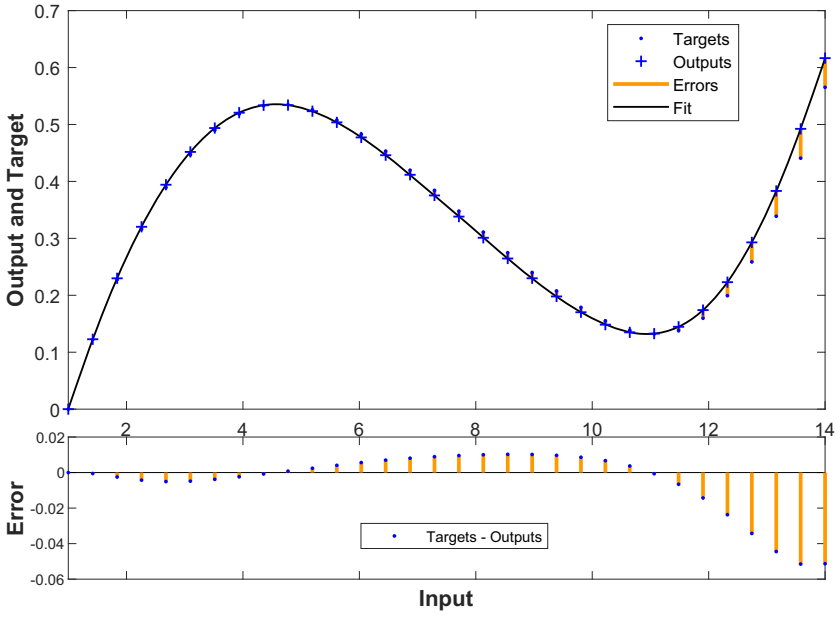
the sizes of the input data, the output data, and the target data are equal as in Figure 7 (the right side). The obtained output is compared to the target achieved from pure IGA utilizing *31 equal load steps*. A satisfactory agreement is reached.

- The training process and the application of the load network are the same as that of the displacement network. The details can be found in Figure 8.

Figure 9 presents the training performance of the displacement and load networks. The training processes of the two networks can be, respectively, stopped after 443 and 99 epochs because the MSEs are very small. The results of applications of the trained ANNs are presented in Figure 10. The very good agreements are found between the obtained outputs (from the networks) and the targets (from pure IGA). The networks are well trained. Interestingly, the cost for training



(a)



(b)

Figure 10: Applying the trained networks in Phase 2: (a) for displacement (mm) network and (b) for load (kN) network.

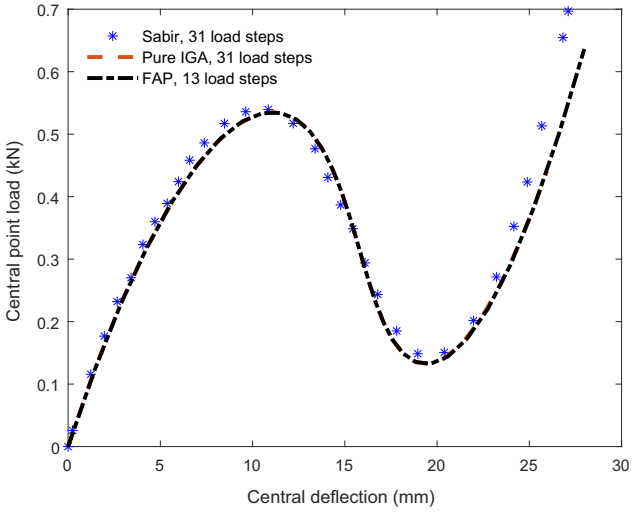


Figure 11: Central deflection of the hinged-hinged panel utilizing various approaches, h = 12.7 mm.

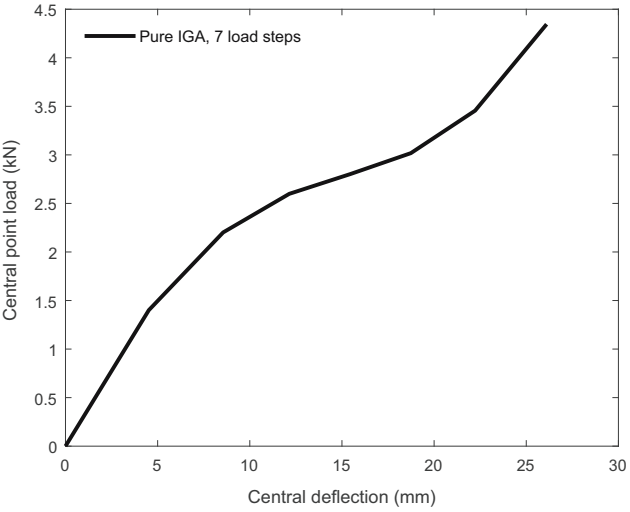


Figure 13: Analysis of the hinged-hinged panel utilizing the FAP: Phase 1 (fast analysis, seven load steps), h = 25.4 mm.

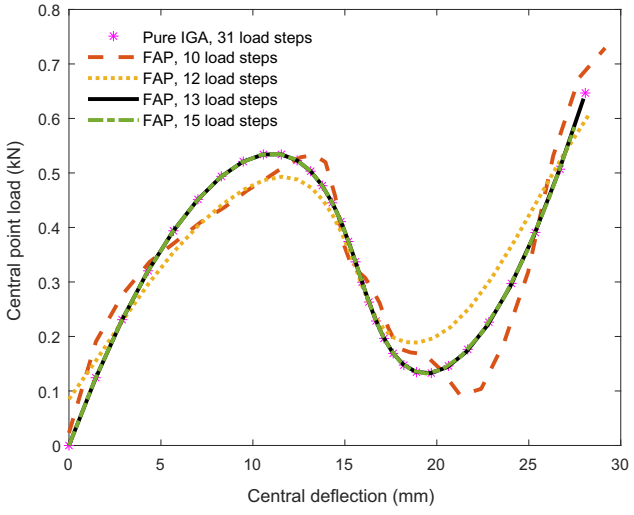


Figure 12: Number of fast analysis steps affects the FAP solution for a hinged-hinged panel, h = 12.7 mm.

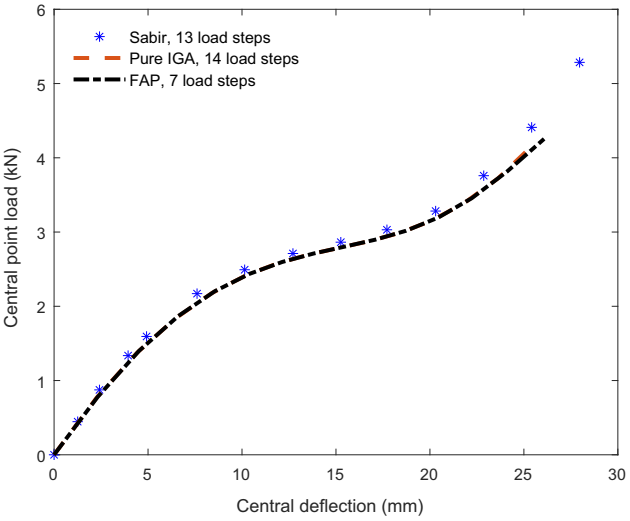


Figure 14: Central deflection of the hinged-hinged panel utilizing various approaches, h = 25.4 mm.

and application of a network is very low (approximately 3 seconds) because the sizes of data are very small together with the advantages of ANNs. Thus, this cost is neglected in evaluation of the effectiveness of the FAP. Besides, Figure 11 presents the obtained results of the FAP utilizing 13 fast analysis steps in the comparison with that of pure IGA utilizing 31 load steps and that of Sabir and Djoudi [33] utilizing pure finite element (FE) analysis with 31 load steps. These results match well. A smooth, complete, and exact curve is achieved after utilizing Phase 2. Also, the upper and lower limit points are detected successfully upon ANNs. The present approach saves 58% of the computational effort compared to other approaches. The high effectiveness and exactness of the present approach are verified. Figure 12 describes the effect of the number of fast analysis steps on the solution. The higher number of fast analysis steps, the higher exactness of the FAP solution. Utilizing 13 analysis steps in Phase 1 can produce a great solution that balances exactness and effectiveness. For snap-through instability problems, we recommend to utilize the present approach with 13 fast analysis steps to ensure exactness and effectiveness in solutions.

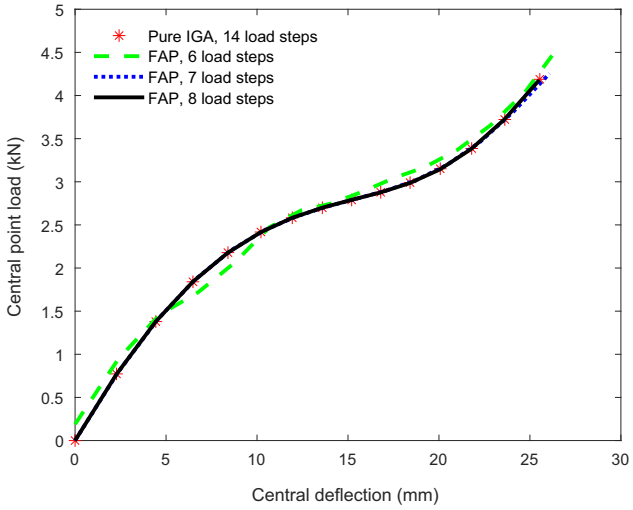


Figure 15: Number of fast analysis steps affects the FAP solution for a hinged-hinged panel, h = 25.4 mm.

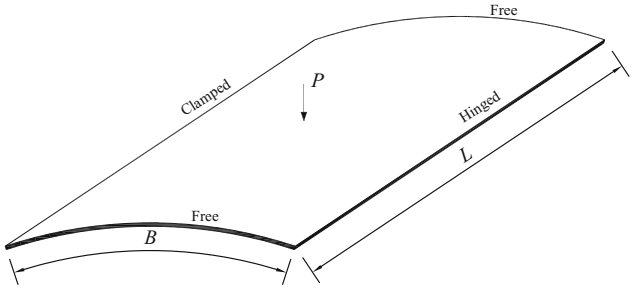


Figure 16: Clamped-hinged panel under a central load, h = 6.35 mm.

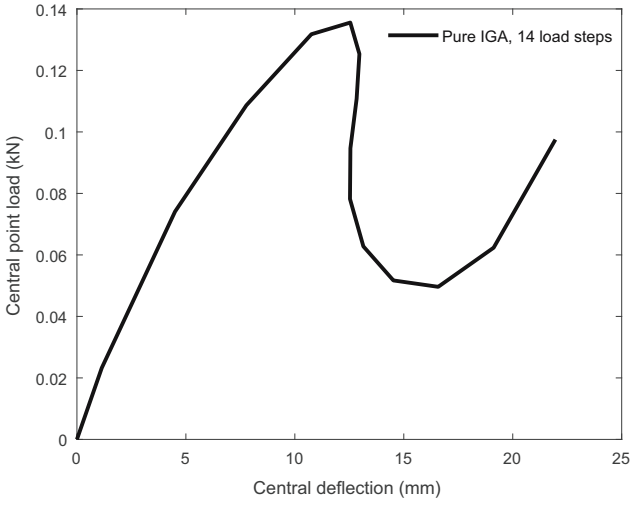


Figure 17: Analysis of the clamped-hinged panel utilizing the FAP: Phase 1 (fast analysis, 14 load steps), h = 6.35 mm.

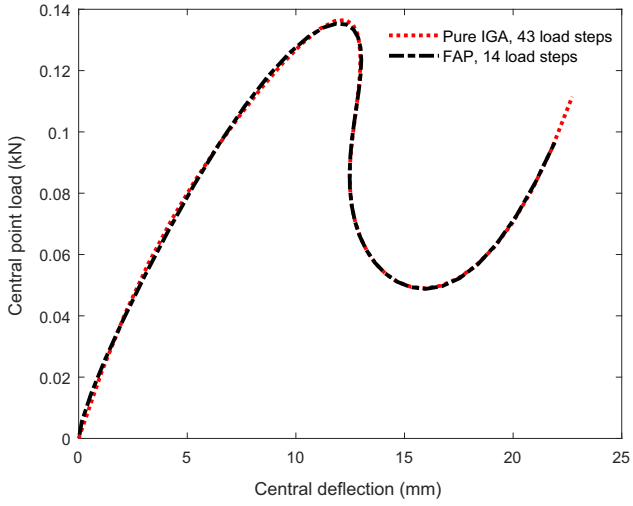


Figure 18: Central deflection of the clamped-hinged panel utilizing various approaches, h = 6.35 mm.

5.2 FAP approach for softening–hardening instability problems of shells

The hinged-hinged panel under a central point load depicted in Figure 5 is continued considering. The geometric data and the material data of the panel are unchanged, except the panel thickness in this problem is h = 25.4 mm. Phase 1 is conducted utilizing seven load steps. The obtained load–displacement relation is shown in Figure 13. As seen, this relation is incomplete and not smooth because of a minor amount of load steps utilized. To achieve a smooth, complete, and correct load–displacement response, we must apply Phase 2 of the FAP utilizing ANNs. The result of Phase 2 is presented in

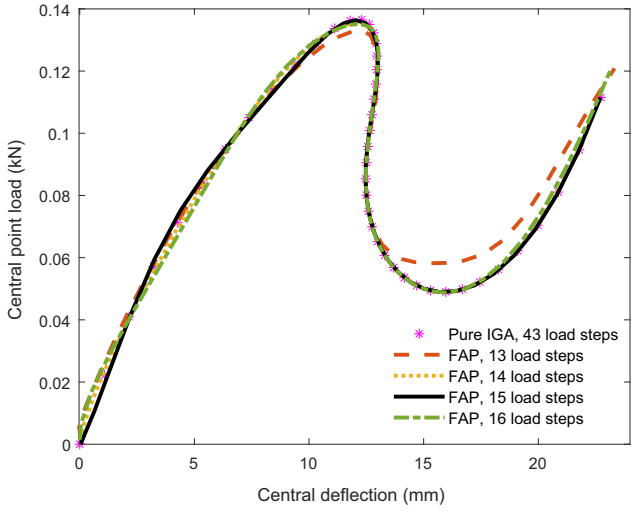
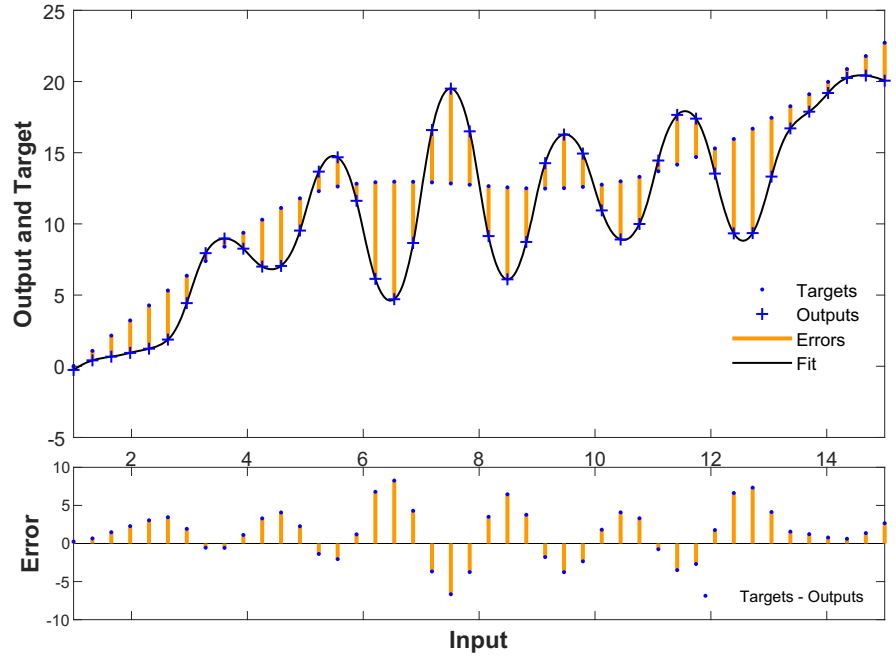
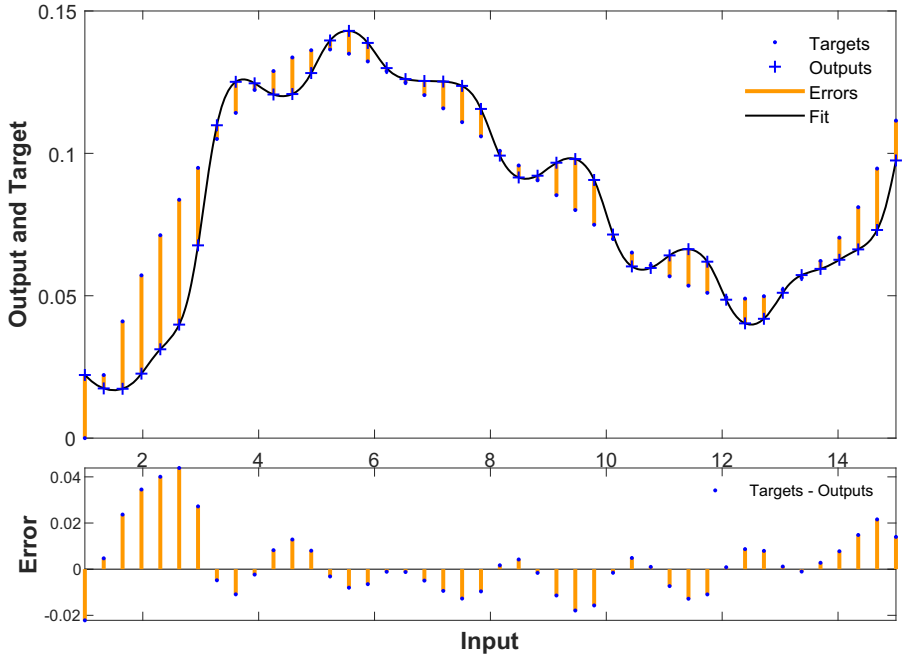


Figure 19: Number of fast analysis steps affects the FAP solution for a clamped-hinged panel, h = 6.35 mm.



(a)

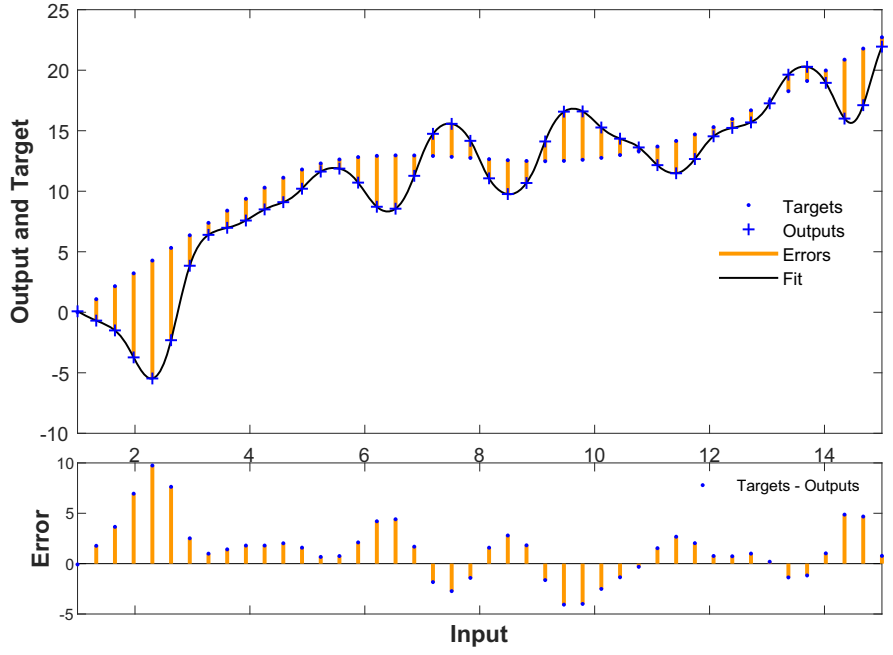


(b)

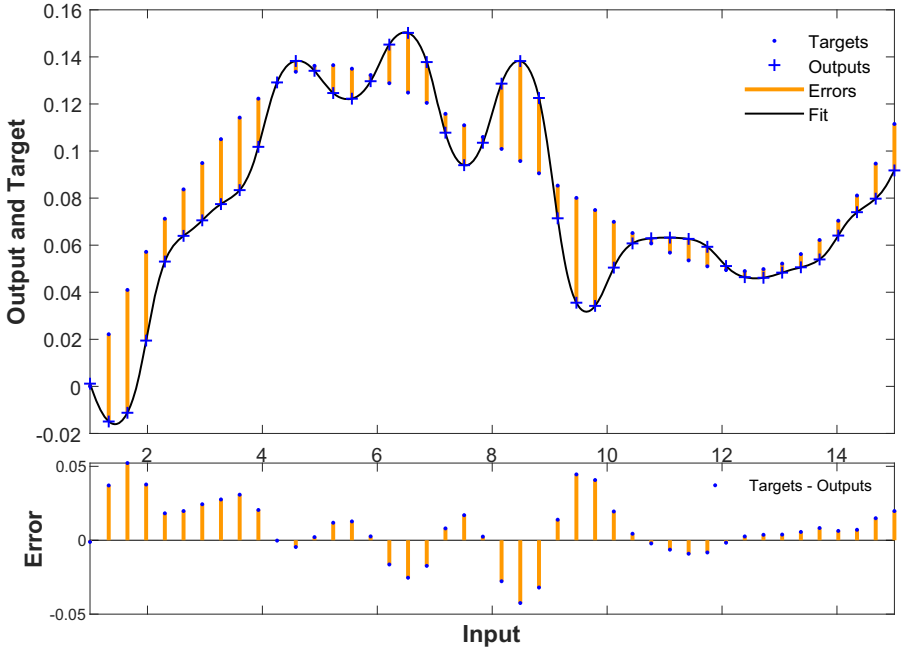
Figure 20: Applications of the networks utilizing ANNs and the L-M algorithm: (a) network for displacement (mm) prediction and (b) network for load (kN) prediction.

Figure 14 in the comparison with that of pure IGA with 14 load steps and that of Sabir and Djoudi [33] utilizing FE analysis with 13 load steps. These results match well. As expected, a smooth, complete, and exact response was achieved after utilizing Phase 2. Besides, the inflection point is detected successfully upon ANNs. The proposed approach can save about 50% of computational effort compared to other ones. The high effectiveness and exactness of the FAP was confirmed. Figure 15 shows the effect of the number of fast analysis steps on the FAP solution. Again,

higher number of fast analysis steps, the higher exactness of the FAP solution. Utilizing seven fast analysis steps produces the great solution that balances exactness and effectiveness. For softening–hardening instability problems, we recommend to utilize the FAP with seven fast analysis steps to ensure exactness and effectiveness in solutions.



(a)



(b)

Figure 21: Applications of the networks utilizing ANNs and the SCG algorithm: (a) network for displacement (mm) prediction and (b) network for load (kN) prediction.

5.3 FAP approach for snap-back instability problems of shells

In the this problem, the clamped-hinged panel under a central point load depicted in Figure 16 is considered. The geometric data and material data of the panel are the same as those in the first problem, except the panel thickness, in this case, is $h = 6.35$ mm. Phase 1 is performed utilizing 14 load steps. The load–displacement relation achieved from Phase 1 is shown in Figure 17. Similar to the previous observations in the first and second problems, the load–displacement relation is a broken line and incomplete, especially in the areas around the limit points. However after application of Phase 2, the load–displacement response becomes smooth, complete, and exact. This fact can be seen in Figure 18 in a comparison of the response of fully pure IGA with 43 load steps. It is concluded that the limit points and the snap-back instability of the panel are detected successfully upon ANNs. Note that to trace successfully the snap-back response of the fully pure IGA, we need to employ a smaller arc length compared to that in the two previous problems. As a result, the number of load steps (43) in the pure analysis of this problem is higher than that of the two previous problems. Figure 19 shows the effect of the number of fast analysis steps on the FAP solution. Utilizing 14 fast analysis steps can produce the good solution with the balance between exactness and effectiveness. For snap-back instability problems, we recommend to utilize the FAP with 14 fast analysis steps to ensure exactness and effectiveness in solutions.

ANNs incorporated with the B-R algorithm recommended for the FAP have some notable advantages

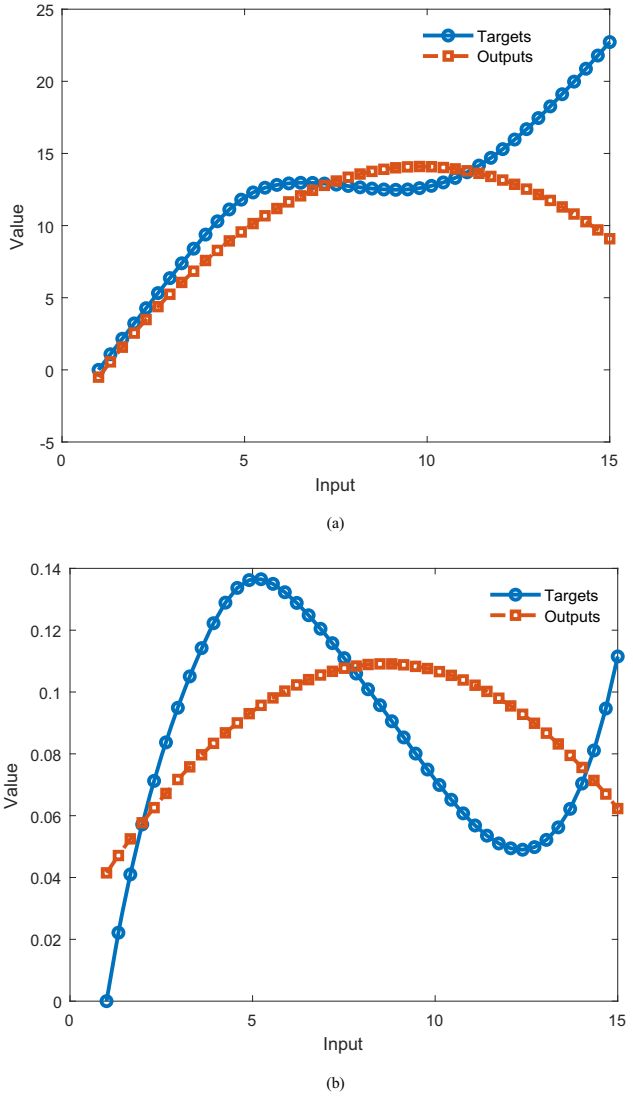


Figure 22: Applications of the networks utilizing the GMDH: (a) network for displacement (mm) prediction and (b) network for load (kN) prediction.

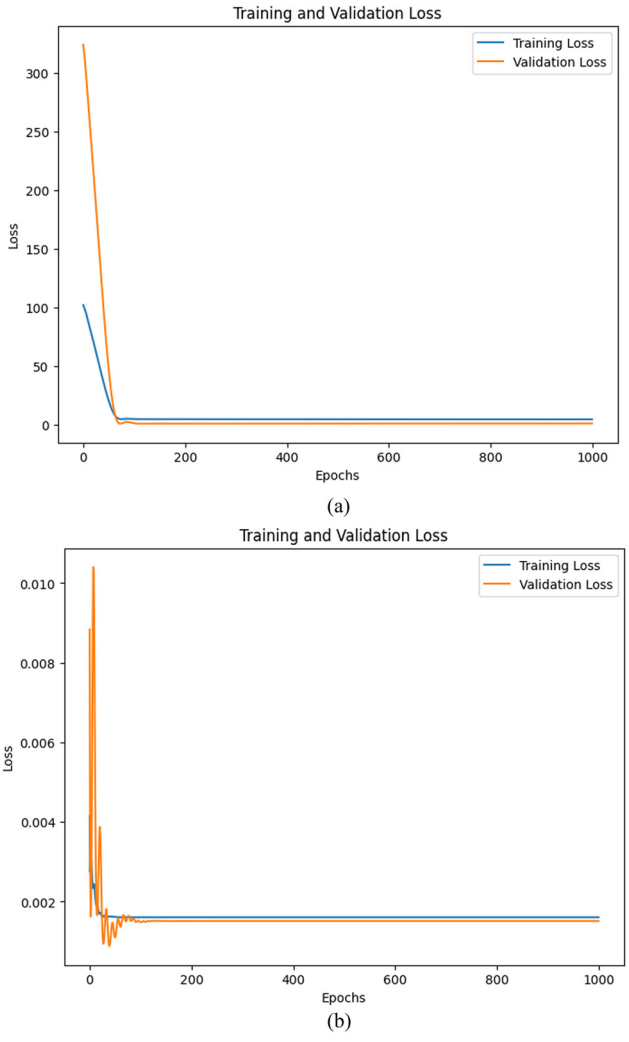


Figure 23: Training performances of DNNs: (a) network for displacement prediction and (b) network for load prediction.

compared to numerous existing networks. To confirm this, we solved this problem utilizing different types of NNs in Phase 2 including ANNs incorporated with the L-M algorithm [21,31], ANNs incorporated with the SCG algorithm [21,31], DNNs [34], and the group method of data handling (GMDH) [35]. For this problem, the data obtained in Phase 1 were re-utilized for training networks in Phase 2. Notably, the ANN's structure, which includes number of neurons and layers, is unchanged. The change is the SCG algorithm, and the L-M algorithm is utilized in this investigation. All the ANNs were trained utilizing MATLAB R2021a as mentioned earlier. For all types of networks in this investigation and in this article, data splitting is performed as follows: the training set (70%), the testing set (15%), and the validation set (15%). Group method of data handling is utilized in many fields including forecasting, optimization, and pattern recognition. In this article, a GMDH-based NN [35] includes 15 neurons per layer and 4 hidden layers. We choose the selection pressure factor α = 0.6. The DNN [34,36,37] is a network utilizing many hidden layers. It can be employed to solve complex nonlinear problems and identify complicated nonlinear systems. DNNs have been investigated for numerous fields including speech recognition, translation, and computer vision. Similar to the GMDH-based network, a DNN includes 15 neurons per layer and 4 hidden layers. The rectified-linear unit function is utilized as an activation function. Besides, the adaptive moment estimation is utilized as an optimizer. MSE is chosen as a loss function. We fix the learning rate at 0.001 and the number of epochs at 1,000. The aforementioned optimizer and functions are well known and presented in detail in the study by Kim *et al.* [34]. Figures 20–24 present the applications of the mentioned networks. In the aforementioned figures, the ground truth or the targets are the values obtained from pure IGA utilizing many load steps. It is found that utilizing ANNs incorporated with the L-M algorithm, ANNs incorporated with the SCG algorithm, GMDH, and DNNs produces incorrect predictions. It should be noted that for each type of network, we need to build up and train two networks to predict displacements and loads. Figure 25 presents the results of the present approach utilizing different types of NNs. These results are compared to that of pure IGA utilizing many load steps. The solution of the present approach utilizing ANNs incorporated with B-R agrees very well with that of the pure IGA, while the remaining solutions are completely incorrect and unstable. It should be noted that

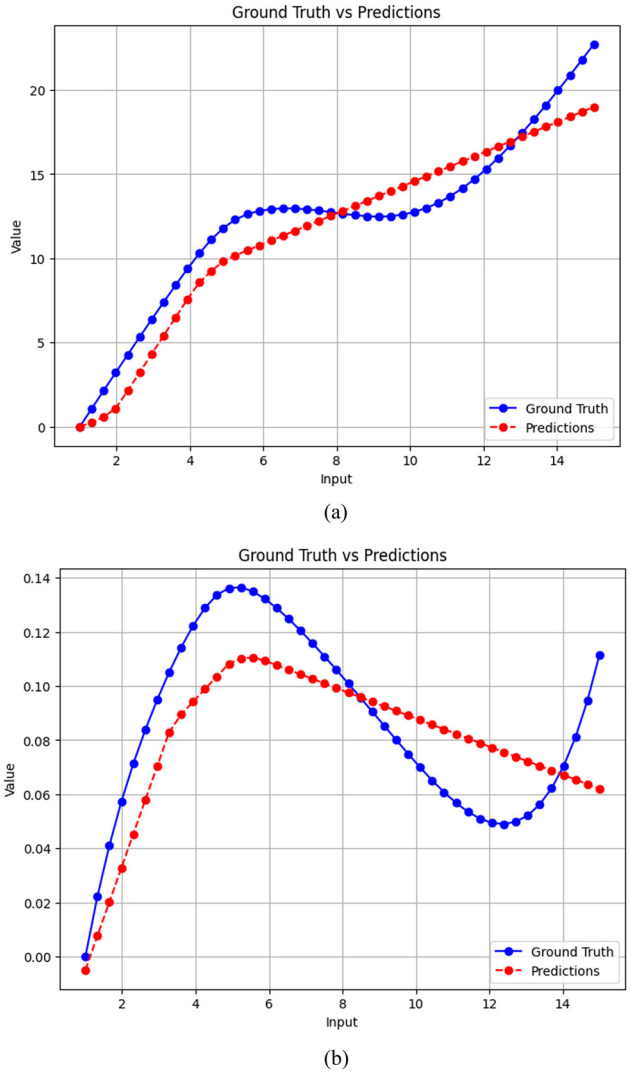


Figure 24: Applications of the trained networks utilizing the DNNs: (a) network for displacement (mm) prediction and (b) network for load (kN) prediction.

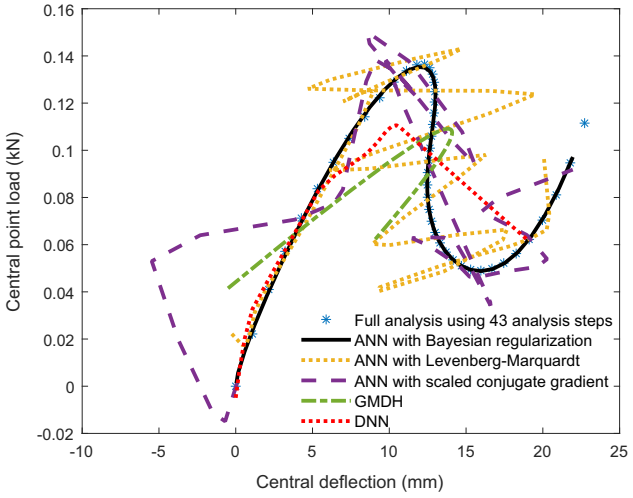


Figure 25: Analysis of a clamped-hinged panel utilizing the FAP and different NNs.

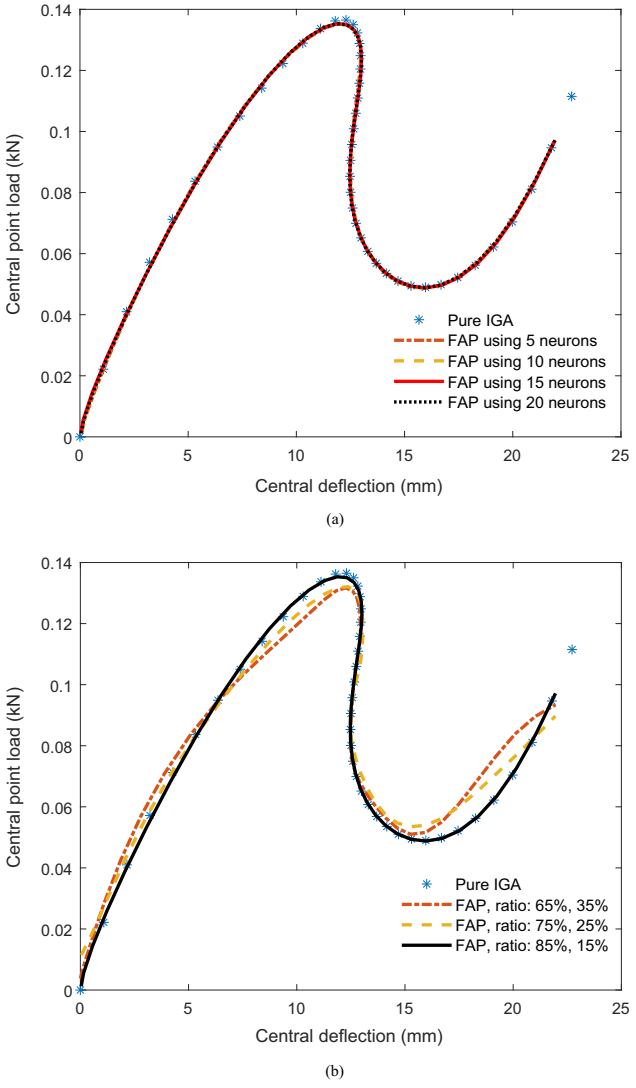


Figure 26: Network's architecture affects the FAP solution: (a) number of neurons, data-splitting ratio: 85%, 15% and (b) data-splitting ratio (training, testing), 15 neurons in the hidden layer.

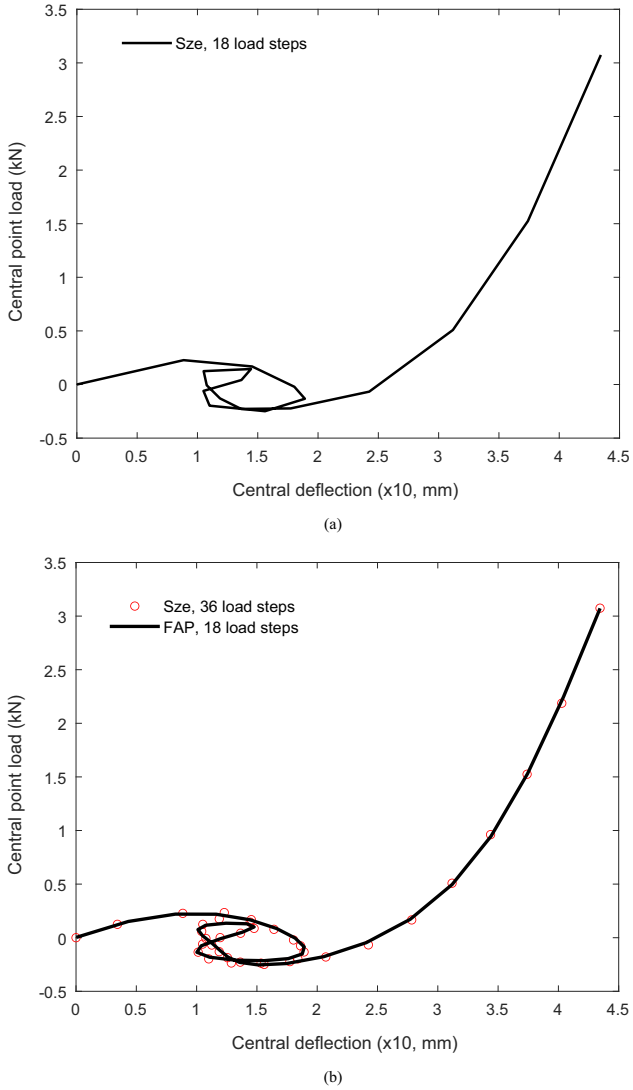
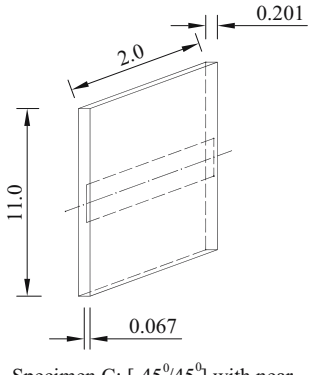


Figure 27: Analysis of the hinged-hinged composite panel with h = 6.35 mm and $[0^0/90^0/0^0]$ stacking sequence utilizing the FAP: (a) Phase 1 (fast analysis) and (b) Phase 2 (prediction) and comparison.

DNNs and GMDH are the improved networks (compared to ANNs) and structured with 15 neurons per layer and 4 hidden layers, while ANNs contain 15 neurons and 1 hidden layer. However, utilizing ANNs incorporated with B-R produces the best solution, as found in Figure 25. This confirms the notable advantages of ANNs incorporated with the B-R algorithm for training with highly nonlinear, difficult, and small data. We recommend to utilize ANNs incorporated with B-R in Phase 2 to achieve exact predictions. Figure 26 shows the effect of the network's architecture on the FAP solution. It can be concluded that the more neurons, the more exact solution; the more data, the more exact solution. To obtain exact predictions, we recommend to utilize the present approach with the network's architecture as: at least 85% data for training and 15 neurons in the hidden layer.



Specimen C: $[-45^0/45^0]_{9S}$ with near surface delamination

Figure 28: An experiment of delamination postbuckling of composite laminates: Geometry of specimen [39].

5.4 FAP approach for kink instability problems of shells

The panel in Figure 5 and Section 5.1 is continued considering. In this problem, the panel thickness is h = 6.35 mm and the laminated composite material is employed with $[0^0/90^0/0^0]$ stacking sequence. The material properties are provided as [38]: E_1 = 3.3 kN/mm^2, E_2 = 1.1 kN/mm^2, $G_{12} = G_{13}$ = 0.66 kN/mm^2, $G_{12} = G_{23}$, ν_{12} = 0.25. A total of 18 data points on the equilibrium path, which has a kink, are extracted from the study by Sze *et al.* [38] as the result of Phase 1. These data points are employed to train networks for load and displacement in Phase 2. The trained networks are then employed to predict loads and displacements for 36 steps. The results of Phases 1 and 2 are presented in Figure 27 in a comparison with that of Sze *et al.* [38] utilizing FEM and 36 steps. A very good agreement is found. It is seen that the result of Phase 1 is not smooth and not completely exact. This is clearly seen when displacement is from 0 to 20. However, after application of the trained networks, the equilibrium path is smooth and exact. Besides, the FAP saves 50% computational effort compared to FEM.

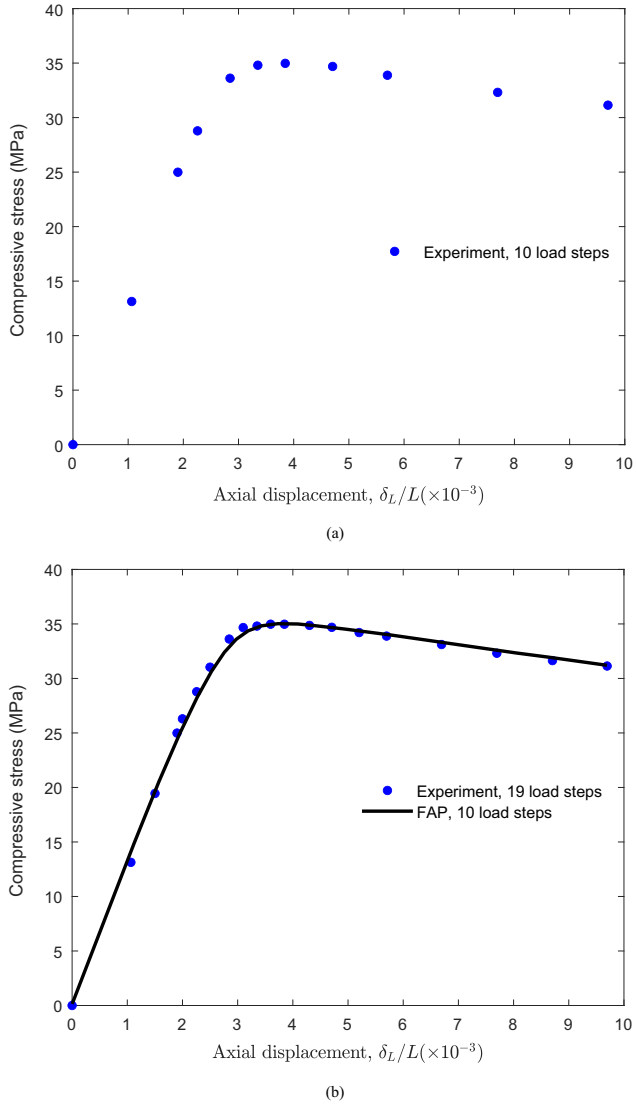


Figure 29: Postbuckling path for the test specimen utilizing the FAP: (a) Phase 1 and (b) Phase 2 (prediction) and comparison.

5.5 FAP approach for an experiment

In the last problem, the FAP is applied to predict delamination buckling and post-buckling of composite laminates utilizing data from experiment. The compression test was conducted on HYE-3574 OH graphite/epoxy composites with built-in delamination [39]. The material properties are found in table 1 in the study by Gu and Chattopadhyay [39]. Geometry of specimen and compression test setup are shown in Figure 28 and Figure 4 in the study by Gu and Chattopadhyay [39], respectively. The delamination length is 3.0 in. The ply stacking sequence of the test specimen was $[-45^0/45^0]_{9S}$. The delamination is near the surface and at the distance 1.7 mm from the surface. The test specimen was clamped at both ends. The axial displacement and compressive load data were digitally recorded. Ten data points on the compressive stress–axial displacement curve are extracted from figure 5d in the study by Gu and Chattopadhyay [39] as the result of Phase 1. These data points are employed to train networks for stress and displacement in Phase 2. The trained networks are then employed to predict stresses and displacements for any load steps. The results of Phases 1 and 2 are presented in Figure 29 in a comparison with the experimental result [39]. A very good agreement is found. As known, the experimental data are affected by many factors including specimen fabrication technique, test setup, and technique of recording data. Thus, training a network utilizing experimental data is more difficult than that utilizing computational data. Interestingly, the present approach is exact for both the computational and experimental data. Although the aforementioned networks were trained utilizing ten experimental data points, they can be employed to predict stresses and displacements for any load steps. This can save labor, time, and money for the experiment.

Finally, the computational effectiveness of the FAP for various types of instabilities of thin shells is summarized and presented in Tables 1 and 2. It is concluded that utilizing the FAP can save a huge amount of computation compared to the other ones (fully pure isogeometric and

Table 1: Effectiveness of the FAP according to load steps

Instabilities	Total steps per analysis		Saved computation (%) compared with pure analysis approach
	Pure analysis	FAP approach	
Snap-through	31	13	58
Softening–hardening	14	7	50
Snap–back	43	14	67
Kink	36	18	50
Experiment	19	10	47

Table 2: Effectiveness of the FAP according to computational cost

Instabilities	Total cost per analysis (s)		Saved cost (%) compared with pure analysis approach
	Pure analysis	FAP approach	
Snap-through	261	126	52
Softening-hardening	116	64	45
Snap-back	403	132	67

fully pure FE analyses) and the experiment. Notably, the computational costs for problems of kink instability and experiment were not available in the literature [38,39]. Therefore, these costs are not present in Table 2.

6 Conclusions

An FAP approach for instability problems of thin shells has been proposed. The present approach contains two phases: fast analysis and pure prediction. (1) for Phase 1, fast post-buckling analysis is conducted utilizing a minor amount of load steps. The obtained load–displacement relation from Phase 1 is incomplete and not exact because of a minor amount of load steps utilized; (2) for Phase 2, the loads and the deflections obtained from Phase 1 are considered as the data for training ANNs incorporated with Bayesian regularization (B-R). After that, the trained networks, including the load network and the displacement network, are utilized to fast predict load and deflection at any step of the post-buckling analysis. After utilizing Phase 2, an exact, complete, and smooth load–displacement curve is achieved. Interestingly, the cost for training and application of an ANN is very low (approximately 3 s) because the utilized data sizes are very small together with the advantages of ANNs. From the numerical verification, several notable conclusions are drawn as follows:

- Good agreement between the results of the FAP and the reference results was found for all problems in this work. The present approach saves a huge computation compared to other ones. High effectiveness and exactness of the proposed approach for post-buckling analysis of thin shells were demonstrated.
- The FAP solutions for instability problems are stable and exact. The more number of fast analysis steps, the more exact solution but high computational cost is required.
- To produce good solutions for the FAP with a balance between exactness and effectiveness (the lowest computational cost), some conclusions are drawn as follows: For snap-through instability problems, we recommend to utilize the present approach with 13 load steps in Phase 1. For softening–hardening instability problems, we recommend to utilize the present approach with seven load steps in Phase 1. For snap-back instability problems, we recommend to utilize the present approach with 14 load steps in Phase 1.
- ANNs incorporated with B-R, which are recommended for the present approach, have some notable advantages compared to numerous existing networks. We recommend to utilize ANNs incorporated with B-R in Phase 2 to achieve exact predictions. Besides, we recommend to utilize the present approach with the network's architecture as: at least 85% data for training and 15 neurons in the hidden layer.
- The FAP is applicable to experiments and highly nonlinear problems, which can have a kink on their equilibrium paths. This can save labor, time, and money for the experiments.

Funding information: This research project was supported by the Second Century Fund (C2F), Chulalongkorn University.

Author contributions: All authors have accepted responsibility for the entire content of this manuscript and approved its submission.

Conflict of interest: The authors state no conflict of interest.

Data availability statement: The datasets generated during and analyzed during the current study are available from the corresponding author on reasonable request.

References

[1] Zhang LW, Lei ZX, Liew KM, Yu JL. Large deflection geometrically nonlinear analysis of carbon nanotube-reinforced functionally graded cylindrical panels. Comput Meth Appl Mech Eng. 2014;273:1–18.

[2] Crisfield MA. A faster modified Newton-Raphson iteration. Comput Meth Appl Mech Eng. 1979;20(3):267–78.

[3] Crisfield MA. A fast incremental/iterative solution procedure that handles 'snap-through. Comput Struct. 1981;13(1):55–62.

[4] Rezaiee-Pajand M, Naserian R. Geometrical nonlinear analysis based on optimization technique. Appl Math Model. 2018;53:32–48.

[5] Rezaiee-Pajand M, Naserian R. Using residual areas for geometrically nonlinear structural analysis. Ocean Eng. 2015;105:327–35.

[6] Castro SGP, Jansen EL. Displacement-based formulation of Koiter's method: Application to multi-modal post-buckling finite element analysis of plates. Thin-Walled Struct. 2021;159:107217.

[7] Sun Y, Tian K, Li R, Wang B. Accelerated Koiter method for post-buckling analysis of thin-walled shells under axial compression. Thin-Walled Struct. 2020;155:106962.

[8] Liang K, Ruess M, Abdalla M. The Koiter-Newton approach using von xxxx kinematics for buckling analyses of imperfection sensitive structures. Comput Meth Appl Mech Eng. 2014;279:440–68.

[9] Magisano D, Liang K, Garcea G, Leonetti L, Ruess M. An efficient mixed variational reduced-order model formulation for nonlinear analyses of elastic shells. Int J Numer Methods Eng. 2018;113(4):634–55.

[10] Mei Y, Hurtado DE, Pant S, Aggarwal A. On improving the numerical convergence of highly nonlinear elasticity problems. Comput Meth Appl Mech Eng. 2018;337:110–27.

[11] Groh RMJ, Avitabile D, Pirrera A. Generalised path-following for well-behaved nonlinear structures. Comput Meth Appl Mech Eng. 2018;331:394–426.

[12] Magisano D, Leonetti L, Garcea G. How to improve efficiency and robustness of the Newton method in geometrically non-linear structural problem discretized via displacement-based finite elements. Comput Meth Appl Mech Eng. 2017;313:986–1005.

[13] Rezaiee-Pajand M, Estiri H. Geometrically nonlinear analysis of shells by various dynamic relaxation methods. World J Eng. 2017;14(5):381–405.

[14] Rezaiee-Pajand M, Estiri H. Finding equilibrium paths by minimizing external work in dynamic relaxation method. Appl Math Model. 2016;40(23):10300–22.

[15] Rezaiee-Pajand M, Afsharimoghadam H. An incremental iterative solution procedure without predictor step. Eur J Comput Mech. 2018;27(1):58–87.

[16] Maghami A, Shahabian F, Hosseini SM. Path following techniques for geometrically nonlinear structures based on Multi-point methods. Comput Struct. 2018;208:130–42.

[17] Chojaczyk AA, Teixeira AP, Neves LC, Cardoso JB, Guedes Soares C. Review and application of Artificial Neural Networks models in reliability analysis of steel structures. Struct Saf. 2015;52:78–89.

[18] Fahmy AS, El-Madawy MET, Atef Gobran Y. Using artificial neural networks in the design of orthotropic bridge decks. Alex Eng J. 2016;55(4):3195–203.

[19] Sebaaly H, Varma S, Maina JW. Optimizing asphalt mix design process using artificial neural network and genetic algorithm. Constr Build Mater. 2018;168:660–70.

[20] Wang S, Sun G, Chen W, Zhong Y. Database self-expansion based on artificial neural network: An approach in aircraft design. Aerosp Sci Technol. 2018;72:77–83.

[21] Nikbin IM, Rahimi I, Allahyari RS. A new empirical formula for prediction of fracture energy of concrete based on the artificial neural network. Eng Fract Mech. 2017;186:466–82.

[22] Tohidi S, Sharifi Y. Neural networks for inelastic distortional buckling capacity assessment of steel I-beams. Thin-Walled Struct. 2015;94:359–71.

[23] Eskandari-Naddaf H, Kazemi R. ANN prediction of cement mortar compressive strength, influence of cement strength class. Constr Build Mater. 2017;138:1–11.

[24] Ye F, Ma S, Tong L, Xiao J, Baaaanard P, Chahine R. Artificial neural network based optimization for hydrogen purification performance of pressure swing adsorption. The 6th International Conference on Energy, Engineering and Environmental Engineering. 2019. Vol. 44. No. 11. p. 5334–44.

[25] Kong YS, Abdullah S, Schramm D, Omar MZ, Haris SM. Optimization of spring fatigue life prediction model for vehicle ride using hybrid multi-layer perceptron artificial neural networks. Mech Syst Signal Proc. 2019;122:597–621.

[26] Pathirage CSN, Li J, Li L, Hao H, Liu W, Ni P. Structural damage identification based on autoencoder neural networks and deep learning. Eng Struct. 2018;172:13–28.

[27] Tan ZX, Thambiratnam DP, Chan THT, Abdul Razak H. Detecting damage in steel beams using modal strain energy based damage index and Artificial Neural Network. Eng Fail Anal. 2017;79:253–62.

[28] Suk JW, Kim JH, Kim YH. A predictor algorithm for fast geometrically-nonlinear dynamic analysis. Comput Meth Appl Mech Eng. 2003;192(22):2521–38.

[29] Kim JH, Kim YH. A predictor–corrector method for structural nonlinear analysis. Comput Meth Appl Mech Eng. 2001;191(8):959–74.

[30] Nguyen TN, Zhang D, Mirrashid M, Nguyen DK, Singhatanadgid P. Fast analysis and prediction approach for geometrically nonlinear bending analysis of plates and shells using artificial neural networks. Mechanics of Advanced Materials and Structures. Philadelphia: Taylor & Francis. 2023. p. 1–19. doi: 10.1080/15376494.2023.2286626.

[31] Mouloodi S, Rahmanpanah H, Gohari S, Burvill C, Davies HMS. Feedforward backpropagation artificial neural networks for predicting mechanical responses in complex nonlinear structures: A study on a long bone. J Mech Behav Biomed Mater. 2022;128:105079.

[32] Dan Foresee F, Hagan MT. Gauss-Newton approximation to Bayesian learning. In: Proceedings of International Conference on Neural Networks (ICNN'97). vol. 3; 1997. p. 1930–5.

[33] Sabir AB, Djoudi MS. Shallow shell finite element for the large deflection geometrically nonlinear analysis of shells and plates. Thin-Walled Struct. 1995;21(3):253–67.

[34] Kim DJ, Kim GW, Baek Jh, Nam B, Kim HS. Prediction of stress-strain behavior of carbon fabric woven composites by deep neural network. Compos Struct. 2023;318:117073.
[35] Ivakhnenko AG. Polynomial theory of complex systems. IEEE Trans Syst Man Cybern. 1971;SMC-1(4):364–78. Number: 4.
[36] Samaniego E, Anitescu C, Goswami S, Nguyen-Thanh VM, Guo H, Hamdia K, et al. An energy approach to the solution of partial differential equations in computational mechanics via machine learning: Concepts, implementation and applications. Comput Meth Appl Mech Eng. 2020;362:112790.
[37] Zhuang X, Guo H, Alajlan N, Zhu H, Rabczuk T. Deep autoencoder based energy method for the bending, vibration, and buckling analysis of Kirchhoff plates with transfer learning. Eur J Mech A-Solids. 2021;87:104225.
[38] Sze KY, Liu XH, Lo SH. Popular benchmark problems for geometric nonlinear analysis of shells. Finite Elem Anal Des. 2004;40(11):1551–69.
[39] Gu H, Chattopadhyay A. An experimental investigation of delamination buckling and postbuckling of composite laminates. Compos Sci Technol. 1999;59(6):903–10.