Self-updated four-node finite element using deep learning

Jaeho Jung^{1,2} · Hyungmin Jun³ · Phill-Seung Lee¹

Received: 30 March 2021 / Accepted: 29 July 2021 / Published online: 24 August 2021
© The Author(s) 2021

Abstract

This paper introduces a new concept called self-updated finite element (SUFE). The finite element (FE) is activated through an iterative procedure to improve the solution accuracy without mesh refinement. A mode-based finite element formulation is devised for a four-node finite element and the assumed modal strain is employed for bending modes. A search procedure for optimal bending directions is implemented through deep learning for a given element deformation to minimize shear locking. The proposed element is called a self-updated four-node finite element, for which an iterative solution procedure is developed. The element passes the patch and zero-energy mode tests. As the number of iterations increases, the finite element solutions become more and more accurate, resulting in significantly accurate solutions with a few iterations. The SUFE concept is very effective, especially when the meshes are coarse and severely distorted. Its excellent performance is demonstrated through various numerical examples.

Keywords Self-updated element · Finite element method · Four-node element · Shear locking · Kinematic modes · Deep learning

1 Introduction

The finite element method (FEM) has been rapidly developed since the 1950s when it was first introduced and successfully commercialized. Compared with other numerical analysis methods, the finite element method is more widely used in actual engineering fields. A key factor in its success is the use of element meshes, which allow the ease of handling complex geometries and boundary conditions. In addition, as the mesh becomes finer, the finite element solutions become more accurate; hence, this convergence characteristic is a crucial feature for ensuring the reliability of solutions.

Ideally, the convergence of finite element (FE) solutions should only depend on the mesh density. However, actual FE solutions rely on domain geometry and material properties, and under certain conditions of the foregoing, the solution convergence can deteriorate; this is called locking. There are various types such as shear locking, membrane locking, volumetric locking and so on. Over the past 50 years, efforts have been devoted to develop various methods, such as reduced integration [1–4], incompatible mode [5, 6], assumed strain, and enhanced strain [7–9] techniques, for alleviating locking. However, the effectiveness of these methods rapidly diminishes as the element meshes become more distorted. The influence of this distortion becomes even more severe on low-order elements, which are generally preferred in engineering practice.

This study proposes a completely different approach for improving FE solutions. We devise a new type of finite element called “self-updated finite element” (SUFE) and a method that considerably increases the accuracy of FE solutions through iterative analysis without mesh refinement. In conventional FE analysis, the stiffness matrix of an element is determined by element geometry and material properties. In the proposed technique, the key idea is to consider element deformation when calculating the stiffness matrix. First, the deformation is estimated through an

✉ Phill-Seung Lee
phillseung@kaist.edu

¹ Department of Mechanical Engineering, Korea Advanced Institute of Science and Technology, 291 Daehak-ro, Yuseong-gu, Daejeon 34141, Republic of Korea

² Korea Atomic Energy Research Institute, 989 Daedeok-daero, Yuseong-gu, Daejeon 34057, Republic of Korea

³ Division of Mechanical System Engineering, Jeonbuk National University, 567, Baekje-daero, Deokjin-gu, Jeonju-si, Jeollabuk-do 54896, Republic of Korea

initial analysis; then, the improved stiffness matrix reflecting this deformation is calculated. Thereafter, through a second analysis, more accurate deformation is derived. By repeating the foregoing process, the FE solution accuracy improves.

The proposed concept is implemented for a two-dimensional four-node plane-stress finite element. The objective is to improve the solutions by minimizing the shear locking effect on the four-node element through iterative analysis. Accordingly, a mode-based FE formulation is adopted. The four-node element has three rigid-body modes and five deformation modes. Among the deformation modes, two different bending modes can be chosen according to the bending direction. For the bending modes, analytical bending strains are enforced using the assumed strain method. For a given element geometry, material properties, and element deformation, the optimal bending directions are determined such that the element strain energy is minimized. To resolve the time-consuming problem of searching the optimal bending directions, deep learning is employed. The four-node self-updated finite element passes the patch and zero-energy mode tests. The main advantage of this approach is that highly accurate FE solutions can be derived even when the meshes are coarse and severely distorted.

In Sect. 2, the basic formulation of the standard four-node finite element is briefly reviewed. The mode-based finite element formulation used for the SUFE is introduced in Sect. 3. The iterative solution procedure for the SUFE is explained in Sect. 4. In Sect. 5, the use of deep neural network to improve the numerical efficiency of the iterative procedure is described. The results of various numerical tests are elaborated in Sect. 6. Finally, the conclusions are presented in Sect. 7.

2 Isoparametric finite element formulation

The basic formulation of the standard four-node finite is reviewed in this section [10].

Consider a mesh of four-node finite elements in the global Cartesian coordinate system defined by $\mathbf{i}_x$ and $\mathbf{i}_y$, as shown in Fig. 1a. The geometry of a four-node finite element m is interpolated by

$$\mathbf{x} = \sum_{i=1}^4 h_i(r, s) \mathbf{x}_i \text{ with } \mathbf{x} = [x \ y]^T \text{ and } \mathbf{x}_i = [x_i \ y_i]^T, \quad (1)$$

where r and s are the natural coordinates in Fig. 1b, $h_i(r, s)$ is the two-dimensional interpolation function of the standard isoparametric procedure corresponding to node i , and $\mathbf{x}_i$ is the position vector of node i in the global Cartesian coordinate system.

The corresponding displacement interpolation of the element is

$$\begin{bmatrix} u \\ v \end{bmatrix} = \sum_{i=1}^4 h_i(r, s) \begin{bmatrix} u_i \\ v_i \end{bmatrix}, \quad (2)$$

where u and v are the displacements in the directions of $\mathbf{i}_x$ and $\mathbf{i}_y$, respectively, and u_i and v_i are the nodal displacements at node i , respectively.

The derivatives of displacements with respect to the global coordinates are calculated using the Jacobian matrix, $\mathbf{J}$:

$$\begin{bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial u}{\partial y} \end{bmatrix} = \mathbf{J}^{-1} \begin{bmatrix} \frac{\partial u}{\partial r} \\ \frac{\partial u}{\partial s} \end{bmatrix} \text{ and } \begin{bmatrix} \frac{\partial v}{\partial x} \\ \frac{\partial v}{\partial y} \end{bmatrix} = \mathbf{J}^{-1} \begin{bmatrix} \frac{\partial v}{\partial r} \\ \frac{\partial v}{\partial s} \end{bmatrix} \quad (3)$$

with $\mathbf{J} = \begin{bmatrix} \frac{\partial x}{\partial r} & \frac{\partial y}{\partial r} \\ \frac{\partial x}{\partial s} & \frac{\partial y}{\partial s} \end{bmatrix}$.

The derivatives in Eq. (3) are utilized to obtain the strain-displacement matrix, $\bar{\mathbf{B}}_m$:

$$\boldsymbol{\varepsilon} = \bar{\mathbf{B}}_m(r, s) \mathbf{u}, \quad (4)$$

$$\boldsymbol{\varepsilon} = [\varepsilon_{xx} \ \varepsilon_{yy} \ \gamma_{xy}]^T \text{ with } \gamma_{xy} = 2\varepsilon_{xy}, \quad (5)$$

Fig. 1 Four-node finite element in **a** global Cartesian coordinate system and **b** natural coordinate system

$$\mathbf{u} = [u_1 \ u_2 \ u_3 \ u_4 \ v_1 \ v_2 \ v_3 \ v_4]^T, \quad (6)$$

where $\boldsymbol{\varepsilon}$ is the strain vector, and $\mathbf{u}$ is the nodal displacement vector of element m .

The stiffness matrix of the standard four-node finite element m is then obtained as

$$\bar{\mathbf{K}}_m = \int_V \bar{\mathbf{B}}_m^T \mathbf{C} \bar{\mathbf{B}}_m dV, \quad (7)$$

where V is the element volume, and $\mathbf{C}$ is the material law matrix.

The equilibrium equation of an FE model is given by

$$\bar{\mathbf{K}}\mathbf{U} = \mathbf{R} \quad \text{with} \quad \bar{\mathbf{K}} = \mathbf{A}_{m=1}^N \bar{\mathbf{K}}_m, \quad (8)$$

where $\bar{\mathbf{K}}$, $\mathbf{U}$, and $\mathbf{R}$ denote the stiffness matrix, displacement vector, and load vector for the entire FE model, respectively; $\mathbf{A}_{m=1}^N$ is the FE assembly operator; and N is the number of finite elements in the FE model.

3 Mode-based finite element formulation

The four-node finite element has eight nodal degrees of freedom (DOFs). The number of kinematic modes that can be represented by the finite element is equal to the number of DOFs. The kinematic modes shown in Fig. 2 can be categorized into three groups, as follows:

Fig. 2 Kinematic modes of the four-node finite element in Fig. 1: **a** three rigid body modes (Φ_1 , Φ_2 , and Φ_3), **b** three constant strain modes (Φ_4 , Φ_5 , and Φ_6), and **c** two linear strain modes (Φ_7 and Φ_8); solid blue lines indicate kinematic mode shapes

- Three rigid body (called zero strain) modes: two translational ($\boldsymbol{\varphi}_1$ and $\boldsymbol{\varphi}_2$) and one rotational ($\boldsymbol{\varphi}_3$) modes
- Three constant strain modes: two stretching modes ($\boldsymbol{\varphi}_4$ and $\boldsymbol{\varphi}_5$) and one shearing ($\boldsymbol{\varphi}_6$) mode
- Two linear strain modes: two bending modes ($\boldsymbol{\varphi}_7$ and $\boldsymbol{\varphi}_8$)

A kinematic mode-based formulation for the four-node finite element incorporated with the assumed strain method is presented in this section.

3.1 Modal representation of strain–displacement matrix

First, two translational and one rotational rigid body modes can be computed through the following form:

$$[\boldsymbol{\varphi}_1 \ \boldsymbol{\varphi}_2 \ \boldsymbol{\varphi}_3] = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ -y_1 & -y_2 & -y_3 & -y_4 & x_1 & x_2 & x_3 & x_4 \end{bmatrix}^T, \quad (9)$$

where x_i and y_i are the coordinates of node i in the global Cartesian coordinate system. Note that the nodal components of the translational mode $\boldsymbol{\varphi}_1$ are obtained when the displacement fields are $u=1$ and $v=0$. Similarly, the translational mode $\boldsymbol{\varphi}_2$ is obtained when $u=0$ and $v=1$, and the rotational mode $\boldsymbol{\varphi}_3$ is obtained when $u=-y$ and $v=x$.

Three constant strain modes (two stretching modes and one shearing mode) can be similarly obtained as

$$[\boldsymbol{\varphi}_4 \ \boldsymbol{\varphi}_5 \ \boldsymbol{\varphi}_6] = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & y_1 & y_2 & y_3 & y_4 \\ y_1 & y_2 & y_3 & y_4 & 0 & 0 & 0 & 0 \end{bmatrix}^T, \quad (10)$$

where the components of the three constant strain modes are obtained when the displacement field is given by $u=x$ and $v=0$ for the stretching mode $\boldsymbol{\varphi}_4$; $u=0$ and $v=y$ for the stretching mode $\boldsymbol{\varphi}_5$; $u=y$ and $v=0$ for the shearing mode $\boldsymbol{\varphi}_6$.

Two linear strain modes (two bending modes) can be calculated by

$$[\boldsymbol{\varphi}_7 \ \boldsymbol{\varphi}_8] = \begin{bmatrix} x_1 y_1 & \dots & x_4 y_4 & -\frac{x_1^2 + y_1^2}{2} & \dots & -\frac{x_4^2 + y_4^2}{2} \\ -\frac{v x_1^2 + y_1^2}{2} & \dots & -\frac{v x_4^2 + y_4^2}{2} & x_1 y_1 & \dots & x_4 y_4 \end{bmatrix}^T, \quad (11)$$

where v is Poisson's ratio. The bending mode components are obtained when the displacement field is given by

$$u = xy \text{ and } v = -\frac{1}{2}(x^2 + vy^2) \text{ for the bending mode } \boldsymbol{\varphi}_7; \quad (12)$$

$$u = -\frac{1}{2}(vx^2 + y^2) \text{ and } v = xy \text{ for the bending mode } \boldsymbol{\varphi}_8. \quad (13)$$

The nodal displacement vector of the four-node finite element can be expressed using the eight kinematic modes in Eqs. (9)–(11)

$$\mathbf{u} = \boldsymbol{\Phi} \mathbf{q} \quad (14)$$

$$\text{with } \boldsymbol{\Phi} = [\boldsymbol{\varphi}_1 \ \boldsymbol{\varphi}_2 \ \dots \ \boldsymbol{\varphi}_8] \text{ and } \mathbf{q} = [q_1 \ q_2 \ \dots \ q_8]^T, \quad (15)$$

where $\boldsymbol{\Phi}$ is the kinematic mode matrix, and $\mathbf{q}$ is the generalized coordinate vector. By substituting Eqs. (14) into (4), the strain vector can be expressed as $\boldsymbol{\varepsilon}$:

$$\boldsymbol{\varepsilon} = \bar{\mathbf{B}}_m \mathbf{u} = \bar{\mathbf{B}}_m \boldsymbol{\Phi} \mathbf{q} = \hat{\mathbf{B}}_m \mathbf{q}, \quad (16)$$

in which $\hat{\mathbf{B}}$ is the matrix for relating the strains with the kinematic modes, and

$$\hat{\mathbf{B}}_m = [\mathbf{e}_1 \ \mathbf{e}_2 \ \dots \ \mathbf{e}_7 \ \mathbf{e}_8] \text{ with } \mathbf{e}_i = \bar{\mathbf{B}}_m \boldsymbol{\varphi}_i \text{ for } i = 1, 2, \dots, 8. \quad (17)$$

The strain–displacement relationship is given by

$$\boldsymbol{\varepsilon} = \hat{\mathbf{B}}_m \mathbf{q} = \hat{\mathbf{B}}_m \boldsymbol{\Phi}^{-1} \mathbf{u} = \bar{\mathbf{B}}_m \mathbf{u}. \quad (18)$$

3.2 Assumed modal strain

It has been well known that shear locking occurs in bending. To improve the solutions by minimizing shear locking, the strain vectors, $\mathbf{e}^7$ and $\mathbf{e}^8$ in $\hat{\mathbf{B}}_m$, corresponding to the two bending modes ($\boldsymbol{\varphi}_7$ and $\boldsymbol{\varphi}_8$) in the four-node finite element m have to be assumed.

In Eq. (11), two bending modes are calculated in the global Cartesian coordinate system. Noting that the bending modes can be chosen differently according to the bending directions is important.

Consider a Cartesian coordinate system defined by two orthonormal base vectors, $\mathbf{i}'_x$ and $\mathbf{i}'_y$ (called bending directions), whose origin is at the same location as that of the global Cartesian coordinate system, as shown in Fig. 3a. The basis vector, $\mathbf{i}'_x$, has an angle, α_m , from $\mathbf{i}_x$. The corresponding bending modes are obtained as follows:

$$[\boldsymbol{\varphi}'_7 \ \boldsymbol{\varphi}'_8] = \begin{bmatrix} x'_1 y'_1 & \dots & x'_4 y'_4 & -\frac{x'^2_1 + y'^2_1}{2} & \dots & -\frac{x'^2_4 + y'^2_4}{2} \\ -\frac{v x'^2_1 + y'^2_1}{2} & \dots & -\frac{v x'^2_4 + y'^2_4}{2} & x'_1 y'_1 & \dots & x'_4 y'_4 \end{bmatrix}^T, \quad (19)$$

where x' and y' are the coordinates in the new Cartesian coordinate system defined by $\mathbf{i}'_x$ and $\mathbf{i}'_y$, respectively.

The nodal components of the bending modes in Eq. (19) are obtained as follows:

Fig. 3 Bending directions and assumed deformed shapes: **a** bending direction angle, α_m , and **b** assumed deformed shapes for bending modes ϕ_7' and ϕ_8'

$$u' = x'y' \text{ and } v' = -\frac{1}{2}(x'^2 + y'^2) \text{ for bending mode } \phi_7'; \quad (20)$$

$$u' = -\frac{1}{2}(vx'^2 + y'^2) \text{ and } v' = x'y' \text{ for bending mode } \phi_8'; \quad (21)$$

where u' and v' are the displacements in the x' and y' directions, respectively. The assumed deformed shape of the element domain obtained using Eqs. (20) and (21) is shown in Fig. 3b. Note that the deformed shape cannot be represented by the displacement interpolation of the standard four-node finite element in Eq. (2).

The corresponding analytical bending strains are employed as the assumed strains ($\mathbf{e}_7'$ and $\mathbf{e}_8'$) for the bending modes (ϕ_7' and ϕ_8'):

$$\mathbf{e}_7' = \mathbf{T} [y' \ -vy' \ 0]^T \text{ and } \mathbf{e}_8' = \mathbf{T} [-vx' \ x' \ 0]^T \quad (22)$$

$$\text{with } \mathbf{T} = \begin{bmatrix} \cos^2 \alpha_m & \sin^2 \alpha_m & -\sin \alpha_m \cos \alpha_m \\ \sin^2 \alpha_m & \cos^2 \alpha_m & \sin \alpha_m \cos \alpha_m \\ 2 \sin \alpha_m \cos \alpha_m & -2 \sin \alpha_m \cos \alpha_m & \cos^2 \alpha_m - \sin^2 \alpha_m \end{bmatrix}, \quad (23)$$

where $\mathbf{T}$ is the strain transformation matrix [11].

Finally, the strain-displacement matrix with the assumed modal strains in Eq. (22) is obtained as follows:

$$\mathbf{B}_m = \hat{\mathbf{B}}_m' \Phi'^{-1} \quad (24)$$

$$\text{with } \hat{\mathbf{B}}_m' = [\mathbf{e}_1 \ \dots \ \mathbf{e}_6 \ \mathbf{e}_7' \ \mathbf{e}_8'] \text{ and } \Phi' = [\phi_1 \ \dots \ \phi_6 \ \phi_7' \ \phi_8']. \quad (25)$$

The stiffness matrix of the four-node finite element adopting the assumed modal strains is obtained as follows:

$$\mathbf{K}_m = \int_V \mathbf{B}_m^T \mathbf{C} \mathbf{B}_m dV. \quad (26)$$

The element stiffness matrix in Eq. (26) depends on the bending directions determined by angle α_m .

The stiffness matrix of the FE model is assembled using the direct stiffness method:

$$\mathbf{K} = \mathbf{A}_{m=1}^N \mathbf{K}_m. \quad (27)$$

4 Iterative solution procedure

The solution accuracy of the standard four-node finite element deteriorates because of shear locking when the displacements contain bending modes. Furthermore, the locking effect becomes more severe when the element mesh is distorted [12–17]. In this section, an iterative procedure to alleviate shear locking is introduced.

The entire solution procedure that adopts the SUFE can be divided into two stages. The preliminary stage determines whether or not the iterative solution procedure proceeds. Once the FE solution does not contain bending modes, the iterative procedure is unnecessary. The determination process is implemented as follows.

The equilibrium equation in Eq. (8) is solved using the standard four-node finite elements to obtain the displacement solution, $\bar{\mathbf{U}}$, and the corresponding strain energy,

$$\bar{E} = 0.5 \cdot \bar{\mathbf{R}}^T \bar{\mathbf{U}}. \quad (28)$$

Using the stiffness matrix in Eq. (27) and the solution ($\bar{\mathbf{U}}$), the strain energy is calculated as

$$E = 0.5 \cdot \bar{\mathbf{U}}^T \mathbf{K} \bar{\mathbf{U}}, \quad (29)$$

Fig. 4 Flowchart of the iterative solution procedure for SUFE

Fig. 5 Random generation of the n th normalized element geometry

Table 1 Cost function values of the training and test datasets according to the depth and width of the neural network and training data size; The batch size is used as half the training data size

Number of weights per layer (network width)	Number of training datasets	Number of layers (network depth)				
		2	4	6	8	10
		(Cost function values of the training / test data)				
160	50,000	3.14/3.23	1.80/2.51	0.84/1.79	0.70/1.50	0.63/1.30
160	100,000	3.09/3.13	1.84/2.28	0.80/1.49	0.62/1.20	0.56/1.02
160	200,000	3.05/3.05	1.88/2.10	0.94/1.35	0.64/1.01	0.57/0.84
320	50,000	3.06/3.18	1.59/2.48	0.72/1.72	0.59/1.40	0.49/1.29
320	100,000	3.03/3.08	1.52/2.16	0.62/1.47	0.47/1.13	0.43/1.01
320	200,000	3.03/3.06	1.62/2.00	0.64/1.22	0.46/0.91	0.44/0.80
640	50,000	2.99/3.16	1.15/2.43	0.42/1.62	0.51/1.38	0.39/1.19
640	100,000	2.97/3.05	1.34/2.15	0.47/1.38	0.36/1.09	0.36/0.96
640	200,000	2.98/3.01	1.49/1.92	0.51/1.20	0.38/0.88	0.37/0.81

Table 2 Cost function values of the training and test datasets according to the activation function type using the network in Fig. 7; The network is trained with a total of 100,000 datasets ($M=100,000$), and the batch size is used as half of the training data size

Type of activation function				
None	ELU	Sigmoid	Tanh	Leaky ReLU
(Cost function values of the training/test data)				
4.39/4.40	0.64/1.48	0.72/1.50	0.46/1.55	0.64/1.50

5 Deep learning for estimating the optimal bending directions

Solving the optimization problem described in Sect. 4 is the most time-consuming step in the iterative solution procedure for SUFE. To considerably reduce the computational time required to solve the optimization problem, a neural network can be used to approximate the result [18]. Here, the neural network is constructed to determine a suitable bending direction angle, α_m . Recently, applications of neural networks to the finite element method have been studied [19–21].

This section introduces the methodology for constructing the neural network model in detail including data generation, network configuration, and training. The optimal bending direction angle is approximated through the trained neural network, and the angle is employed to update the stiffness matrix of SUFE.

5.1 Generation of training datasets

Numerous training datasets, including random geometries, displacements, and material properties, are required to train the neural network to accurately approximate the optimal bending direction angle (α_m) of arbitrary four-node finite elements. Because the processing of all data spaces is extremely difficult, a previously developed methodology is applied [21] to reduce the amount of data required to achieve efficient network training. Note that deep learning is used to generate stiffness matrices of finite elements in Ref. [21].

The geometry of a four-node element is determined by its four nodal positions. A normalized geometry is adopted to represent all similar shapes [21]. Here, the normalized geometry refers to a quadrilateral where the two nodes

Fig. 8 Histograms of α in the **a** training and **b** test datasets

Fig. 9 Preprocessing to obtain network inputs: **a** original and **b** normalized element geometries

(node 1 and 2) at either end of the longest side are fixed at (0, 0) and (1, 0) in the two-dimensional (2D) Cartesian coordinate system. To generate the n th geometry, two other nodes (nodes 3 and 4) are placed randomly, as shown in Fig. 5. In the geometry generation, extremely distorted geometries are excluded. Excluded geometries are quadrilaterals whose interior angle is less than 1° or greater than 179° and those whose ratios between the maximum and minimum side lengths exceed 100. Then, the normalized

geometry is generated by four values (i.e., the coordinates of nodes 3 and 4): x_3 , y_3 , x_4 , and y_4 [21].

The optimal bending directions depend on the nodal displacements representing the deformed shapes of the finite element. Normalized nodal displacements are considered to represent all similar deformed shapes. Consider an element subjected to the minimum number of constraints to prevent rigid body motions, as shown in Fig. 6a. Then only five nodal displacements (i.e., u_2, u_3, v_3 ,

Fig. 10 Preprocessing of nodal displacements for network inputs: **a** initial arbitrary nodal displacements, **b** rotation of element geometry and displacements, **c** elimination of translational rigid body modes,

d elimination of rotational rigid body mode, and **e** normalization of nodal displacements

u_4 , and v_4) can represent the entire deformed shape. The n th normalized nodal displacement is generated by

$$\begin{aligned} u_2 &= \frac{u'_2}{\delta'_{\max}}, u_3 = \frac{u'_3}{\delta'_{\max}}, v_3 = \frac{v'_3}{\delta'_{\max}}, \\ u_4 &= \frac{u'_4}{\delta'_{\max}}, \text{ and } v_4 = \frac{v'_4}{\delta'_{\max}}, \end{aligned} \quad (33)$$

where u'_2, u'_3, v'_3, u'_4 , and v'_4 are uniformly generated random numbers ranging from -1.0 to 1.0 , and $\delta'_{\max}$ is the largest among the absolute values of the five generated random numbers.

Poisson's ratio, ν , is also randomly generated with a uniform distribution in the range $0-0.499999999$, and Young's modulus is arbitrarily chosen because it does not affect the determination of the optimal bending directions.

The optimal bending direction angle, $\alpha^{(n)}$, is required as labeled data for the generated n th geometry, displacements, and material properties. The angle is calculated by solving the optimization problem in Eq. (31). The well-known Nelder–Mead method [22] is employed, and 30 initial seeds evenly distributed in the range $0^\circ-90^\circ$ of angle α are considered to find the global optimum.

Finally, the n th training dataset, $\mathbf{D}^{(n)}$, is configured as

$$\mathbf{D}^{(n)} = [\mathbf{G}^{(n)} \ \alpha^{(n)}] \quad (34)$$

with

$$\mathbf{G}^{(n)} = [x_3 \ y_3 \ x_4 \ y_4 \ u_2 \ u_3 \ v_3 \ u_4 \ v_4 \ \nu]^{(n)}. \quad (35)$$

5.2 Network configuration and training

The neural network model architecture with 10 inputs (four nodal coordinates, five nodal displacements, and one Poisson's ratio), one output ($\alpha(\theta)$), and ten hidden layers is

shown in Fig. 7. The angle $\alpha^{(n)}$ of the training dataset is used for the cost function $C(\theta)$:

$$C(\theta) = \frac{1}{M} \sum_{n=1}^M |\alpha(\theta) - \alpha^{(n)}|^{\frac{1}{2}}, \quad (36)$$

where θ denotes the network weights, and M is the total number of training datasets.

The deep neural network shown in Fig. 7 is modeled using a fully connected neural network with 10 layers; The neural network depth is 320, and batch normalization is applied to each layer. The effect of the network structure and data size on the cost function values of the training and test datasets is shown in Table 1. As the depth and width of the network and training data size increase, the cost function values decrease. In this study, the network structure shown in Fig. 7 was adopted in consideration of efficiency.

To account for the nonlinear behavior of the training datasets, an exponential linear unit used as the activation function is added to each neuron. The effect of the activation function type on the cost function values of the training and test datasets is shown in Table 2. The results show that using the activation function results in smaller cost function values, but there is no significant difference in the results for the activation function type.

To correctly infer the value of α_m , 3,000,000 training datasets and 50,000 test datasets are utilized. Histograms of $\alpha^{(n)}$ in the training and test datasets are shown in Fig. 8. The distribution of the two datasets is similar.

TensorFlow [23] is used as the application programming interface to model and train the neural network. Adam optimization [24] is adopted as the optimizer, and Xavier initializer [25] is applied to randomize the initial weights of the neural network. Training is executed for 30,000 epochs, and a batch size of 50,000 is used. The learning rate linearly converged from 0.01 to 0 as the epoch progressed to approach better results. As a result of training, the average errors of

Table 3 List of finite elements used for comparison. The Q8 element is an eight-node element unlike the others

Element	Description	References
Q4	Standard four-node quadrilateral element	[10]
Q8	Standard eight-node quadrilateral element	[10]
QM6	Four-node quadrilateral element with incompatible modes	[6]
P-S	Four-node stress-hybrid element	[7]
SPS	Four-node hybrid stress elements with adjustable parameters	[30]
B-QE4	Four-node assumed strain element (B-bar)	[37]
QACM4	Four-node incompatible element using QACM-I	[38]
NQ10	Four-node incompatible hybrid stress element	[16]
GC-Q6	Four-node generalized conforming element	[39]
PEAS7	Four-node assumed strain element	[12]
2D-MITC4/1	Four-node assumed strain element based on the MITC method	[31]
SU4	Proposed four-node element	—

Fig. 11 Geometries used for **a** patch test ($q = 1.0$, $E = 3.0 \times 10^7$, $\nu = 0.3$, and thickness = 1) and **b** zero-energy mode test ($E = 1.5 \times 10^3$, $\nu = 0.3$, and thickness = 1)

the training and test datasets are 0.274° and 0.438° , respectively. All training and evaluation calculations are performed using NVIDIA Tesla V100 (5120 CUDA cores and 32 GB memory).

5.3 Estimation of the bending direction angle from the trained network

Normalized geometries, normalized displacements, and Poisson's ratios were employed for efficient network training. To use the trained neural network for the iterative solution procedure proposed in Sect. 4, preprocessing the geometry and displacements of the four-node finite element m is required to obtain network inputs, as shown in Fig. 9.

Fig. 12 Cantilever beam for rotational frame invariance test: **a** problem description ($L_1 = 8$, $L_2 = 2$, $H = 5$, $P = 0.2$, $E = 100$, $\nu = 0.3$, and thickness = 1) and beam geometries rotated by **b** 30° and **c** 60° relative to x -axis

Table 4 Tip displacements at point A of the cantilever beam for rotation dependency test shown in Fig. 11; values in parentheses are final numbers of iterations

Rotation angle (°)	Tip displacements			Normalized
	u_A	v_A	$\sqrt{u_A^2 + v_A^2}$	
0	-2.400E-02	-4.800E-02	5.367E-02	0.9938 (1)
10	-1.530E-02	-5.143E-02	5.366E-02	0.9937 (2)
20	-6.145E-03	-5.328E-02	5.364E-02	0.9933 (3)
30	3.195E-03	-5.353E-02	5.363E-02	0.9931 (4)
40	1.246E-02	-5.218E-02	5.365E-02	0.9934 (5)
50	2.137E-02	-4.926E-02	5.370E-02	0.9944 (6)
60	2.958E-02	-4.479E-02	5.368E-02	0.9940 (4)
70	3.686E-02	-3.896E-02	5.363E-02	0.9932 (3)
80	4.317E-02	-3.198E-02	5.373E-02	0.9950 (3)
90	4.800E-02	-2.400E-02	5.367E-02	0.9938 (1)
Reference solution [27]: $\sqrt{u_A^2 + v_A^2} = 5.400\text{E-}02$				

The geometry normalization shown in the figure is performed for the four-node element used in the FE model. The FE connectivity is assigned such that the side length between nodes 1 and 2 is the largest. Then, the normalized nodal coordinates $\bar{\mathbf{x}}_i$ ($\bar{x}_i$ and $\bar{y}_i$) as inputs of the neural network are obtained by translating, rotating, and resizing the nodal coordinates, $\mathbf{x}_i$, of the finite element m [21]:

$$\bar{\mathbf{x}}_i = \frac{\mathbf{R}^T(\mathbf{x}_i - \mathbf{x}_1)}{l_{\max}} \text{ for } i = 1, 2, 3, 4 \quad (37)$$

with $\mathbf{R} = \begin{bmatrix} \cos \beta & -\sin \beta \\ \sin \beta & \cos \beta \end{bmatrix}$, where $\mathbf{R}$ is the rotation matrix, β is the angle between the longest side and x -axis, and $l_{\max}$ is the largest side length.

Consider the eight nodal displacements (u_i and v_i) of the four-node finite element m . The nodal displacements are rotated through angle β to match the displacements with the normalized geometry, as shown in Fig. 10b. Translational and rotational rigid-body motions are eliminated because the optimal bending directions are not affected by rigid body motions, as shown in Fig. 10c and d. Then the resulting nodal displacements are normalized, as shown in Fig. 10e.

According to the explained procedure, the normalized nodal displacements, $\bar{u}_i$ and $\bar{v}_i$, can be calculated by the following equations:

$$\bar{u}_i = \frac{\hat{u}_i}{\delta_{\max}} \text{ and } \bar{v}_i = \frac{\hat{v}_i}{\delta_{\max}} \text{ for } i = 1, 2, 3, 4 \quad (38)$$

with

$$\hat{u}_i = \tilde{u}_i - \tilde{u}_1 - \bar{y}_i(\tilde{v}_1 - \tilde{v}_2), \quad (39)$$

$$\hat{v}_i = \tilde{v}_i - \tilde{v}_1 + \bar{x}_i(\tilde{v}_1 - \tilde{v}_2), \quad (40)$$

$$\begin{bmatrix} \tilde{u}_i \\ \tilde{v}_i \end{bmatrix} = \mathbf{R}^T \begin{bmatrix} u_i \\ v_i \end{bmatrix}, \quad (41)$$

where x_i and y_i are the nodal coordinates of element m , and $\delta_{\max}$ is the largest among the calculated absolute values of $\hat{u}_i$ and $\hat{v}_i$ for all nodes. As a result, $\bar{u}_1$, $\bar{v}_1$, and $\bar{v}_2$ become zero, and the remaining non-zero displacements, $\bar{u}_2$, $\bar{u}_3$, $\bar{v}_3$, $\bar{u}_4$, and $\bar{v}_4$, are used as network inputs.

Poisson's ratio (ν) is used as an input without any modification into the neural network. By using the network inputs given by

Fig. 13 Cook's skew beam problem: **a** problem description ($P=1$, $E=1$, $\nu=1/3$, and thickness = 1) and **b** FE meshes used

Table 5 Displacements at point A (v_A) for Cook's skew beam shown in Fig. 13a; values in parentheses are final numbers of iterations for the SU4 element

Elements	v_A					Normalized v_A				
	1×1	2×2	4×4	8×8	16×16	1×1	2×2	4×4	8×8	16×16
Q4	5.97	11.85	18.30	22.08	23.43	0.249	0.492	0.763	0.921	0.978
Q8	17.22	21.72	23.71	23.88	23.93	0.718	0.948	0.989	0.997	0.999
QM6	15.92	21.05	23.02	23.69	23.88	0.664	0.878	0.961	0.989	0.996
P-S	–	21.13	23.02	23.69	23.88	–	0.882	0.961	0.989	0.996
B-QE4	–	21.35	23.04	–	23.88	–	0.891	0.961	–	0.996
QACM4	–	20.74	22.99	23.69	–	–	0.865	0.959	0.989	–
GC-Q6	–	27.61	24.31	23.99	–	–	1.152	1.014	1.001	–
2D-MITC4/1	–	19.61	22.64	23.59	23.86	–	0.818	0.945	0.984	0.995
SU4	25.47	22.97	23.90	23.96	23.95	1.063	0.958	0.997	1.000	0.999
($\delta=0.01$)	(5)	(2)	(2)	(2)	(2)	(5)	(2)	(2)	(2)	(2)
SU4	25.47	23.74	23.91	23.96	23.95	1.063	0.991	0.998	1.000	0.999
($\delta=0.001$)	(6)	(12)	(3)	(2)	(2)	(6)	(12)	(3)	(3)	(2)
SU4	25.47	23.78	23.93	23.96	23.95	1.063	0.992	0.998	1.000	0.999
($\delta=0.0001$)	(6)	(21)	(5)	(3)	(2)	(6)	(21)	(5)	(3)	(2)

Reference solution [32]: $v_A = 23.965$ **Fig. 14** Normalized vertical displacement in Cook's skew beam with respect to **a** number of iterations and **b** number of element layers, N

$$\mathbf{G}_{input} = [\bar{x}_3 \ \bar{y}_3 \ \bar{x}_4 \ \bar{y}_4 \ \bar{u}_2 \ \bar{u}_3 \ \bar{v}_3 \ \bar{u}_4 \ \bar{v}_4 \ v], \quad (42)$$

the output, α_{out} , of the trained network is derived. Finally, the bending direction angle, α_m , for element m is calculated by

$$\alpha_m = \beta + \alpha_{out}. \quad (43)$$

6 Numerical results

In this section, three basic tests are performed for the SUFE to verify its performance through several numerical problems: Cook's skew beam, MacNeal's cantilever beam, cantilever beam modeled using two elements with distortion

parameter, cantilever beams modeled using distorted elements, thick and thin curved beams, and strap plate. To thoroughly investigate the performance of SUFE, the results are compared with those of several existing elements listed in Table 3. All compared elements pass the patch test. In this section, the self-updated four-node element is called "SU4." Note that the Q8 element in Table 3 is an eight-node element unlike the others.

The smaller δ , the higher the accuracy, but the more iterations, see Table 5. In the following numerical examples, the iterations are performed using a threshold δ of 0.001 by default.

The computations are performed using Python scripts on an AMD Ryzen 3900X @ 3.80 GHz, 64-GB RAM, 64-bit Windows 10 operating system. The linear equations are

Fig. 15 Convergence curves for Cook's skew beam: **a** relative strain energy error with respect to element size (bold line represents the optimal convergence rate); **b** relative strain energy error with respect to number of iterations

Fig. 16 Shear stress distributions calculated in Cook's skew beam using the SU4 element with $N=8$: **a** iteration=0 and **b** iteration=1; **c** reference solution obtained using standard nine-node quadrilateral elements with $N=64$

solved using a sparse solver in the Python package, NumPy [26].

6.1 Basic numerical tests

In this section, we perform three basic tests: patch test, zero-energy mode test, and rotation dependency test. The patch test is passed if constant stress fields are represented exactly considering shearing and stretching in the finite element model in Fig. 11a [10]. The self-updated four-node finite element (SU4) passes the standard patch tests.

The zero-energy mode test has been widely used to verify that a finite element can correctly represent rigid body motions; the four element geometries shown in Fig. 11b are considered. In calculating the stiffness matrix of an SUFE, the element deformation is used as the input. Accordingly, the zero-energy mode test is performed considering various element deformations. The SU4 element correctly produces only three rigid body modes (two translations and one rotation mode) without any spurious energy mode in all geometry cases regardless of element deformation. That is, the SU4 element passes the zero-energy mode test.

Fig. 17 MacNeal's beam subjected to two load cases: tip shear force P and tip bending moment M ($L=6$, $H=0.2$, $P=1$, $M=0.2$, $E=1 \times 10^7$, $\nu=0.3$, and thickness = 0.1); three different mesh patterns: **a** regular (Mesh I), **b** parallelogram (Mesh II), and **c** trapezoidal (Mesh III) meshes

Table 6 Normalized vertical displacements at point A for MacNeal's cantilever beam subjected to either tip shear force or tip moment; values in parentheses are final numbers of iterations for SU4 element

Elements	Load P			Load M		
	Mesh I	Mesh II	Mesh III	Mesh I	Mesh II	Mesh III
Q4	0.093	0.034	0.027	0.093	0.031	0.022
Q8	0.982	0.899	0.980	0.991	0.919	0.996
QM6	0.993	0.632	0.051	1.000	0.727	0.045
P-S	0.993	0.798	0.221	1.000	0.852	0.167
PEAS7	0.982	0.795	0.217	–	–	–
QACM4	0.995	0.635	0.052	1.000	0.722	0.046
SU4	0.993 (1)	0.994 (1)	0.994 (1)	1.000 (1)	1.000 (1)	1.000 (1)

Reference solutions are -0.1081 for load P and -0.0054 for load M [29]

Fig. 18 Cantilever beam modeled using two elements with distortion parameter, e ($L=10$, $H=2$, $M=2000$, $E=1500$, $\nu=0.25$, and thickness = 1)

Table 7 Normalized vertical displacements at point A in the cantilever modeled using two elements with distortion parameter, e ; values in parentheses are final numbers of iterations for the SU4 element

Elements	Distortion parameter, e					
	0	1	2	3	4	4.9
Q4	0.280	0.144	0.098	0.083	0.071	0.062
Q8	1.000	0.993	0.891	0.597	0.320	0.197
QM6	1.000	0.673	0.624	0.657	0.669	0.648
P-S	1.000	0.679	0.631	0.672	0.700	0.703
SPS	–	1.100	1.208	1.332	1.479	1.639
QACM4	1.000	0.665	0.601	0.614	0.603	0.560
SU4	1.000 (1)	1.000 (1)	1.000 (1)	1.000 (1)	1.000 (1)	1.000 (1)
Reference (exact) solution [30]: $v_A = 100$

Fig. 19 Normalized vertical displacements at point A according to distortion parameter, e , in the cantilever beam modeled using two elements; reference solution is $v_A = 100$ [30]

The rotation test is intended to investigate the angular dependency of the finite element solution. The cantilever beam shown in Fig. 12 is considered [27]. It is discretized into two distorted elements with lengths $L_1 = 8$ and $L_2 = 2$; the width is $H = 5$ and thickness = 1. The plane stress condition is used, and the material properties are Young's modulus (E) = 100 and Poisson's ratio (ν) = 0.3. A moment with $P = 0.2$ is applied at the free end. The beam geometry is rotated counterclockwise by up to 90° at 10° increments with respect to the x -axis. Table 4 lists the tip displacements at point A as the rotation angle varies. The SUFE provides almost the same results regardless of the rotation angle.

6.2 Cook's skew beam

To further examine the SUFE, widely known benchmark problems are solved. The first benchmark test is Cook's skew beam problem [28]. The geometric dimensions and boundary conditions used are shown in Fig. 13a. The plane stress condition is assumed, and the material properties are Young's modulus (E) = 1.0 and Poisson's ratio (ν) = 1/3. A

Fig. 20 Stress distributions calculated using elements Q4, QM6, and SU4 for the cantilever beam shown in Fig. 19 with distortion parameter (e) = 4.9; reference solutions obtained using standard nine-node quadrilateral elements with 4×20 regular mesh

Fig. 21 Cantilever beam modeled using five distorted elements ($P=150$, $M=2000$, $E=1500$, $\nu=0.25$, and thickness = 1)

Table 8 Normalized vertical deflection at point A (v_A) and normalized stress at point B (σ_B) for the cantilever beam modeled using five distorted elements; values in parentheses are final numbers of iterations for the SU4 element

Elements	Load M		Load P	
	v_A	σ_B	v_A	σ_B
Q4	0.457	0.587	0.494	0.604
Q8	0.997	0.994	0.990	1.000
QM6	0.961	0.837	0.956	0.850
P-S	0.962	1.005	0.957	1.021
NQ10	0.960	0.995	0.954	0.993
SPS	1.021	1.000	–	–
GC-Q6	0.950	1.012	0.937	1.033
B-QE4	0.965	1.001	0.958	0.964
QACM4	0.960	1.005	0.955	1.021
SU4	1.000 (1)	1.000 (1)	0.999 (2)	1.030 (2)

Reference solutions [16]:

$v_A = 100$ and $\sigma_B = -3000$ for load M

$v_A = 102.6$ and $\sigma_B = -4050$ for load P

shear force, $P=1$ (1/16 force per unit length), is applied at the right end. The solutions of the skew beam are obtained using $N \times N$ element meshes ($N=2, 4, 8$, and 16). The reference solution of the vertical displacement at point A is 23.965, as given in Ref. [28].

To investigate the predictive capability of SUFE in detail, convergence studies are performed. The solution accuracy is measured using the relative strain energy error, which is given as

$$E_e = \left| \frac{E_{\text{ref}} - E_h}{E_{\text{ref}}} \right|, \quad (44)$$

where E_{ref} and E_h denote the strain energies stored in the entire structure obtained from the reference and FE solutions, respectively. The optimal convergence of the relative strain energy error is estimated as $E_e \cong ch^2$, where c is a constant, and h is the element size. In the convergence curves, $h=1/N$ is used. The reference solution, E_{ref} , is obtained using the 100×100 element mesh of standard nine-node quadrilateral elements.

Table 5 lists the deflections at point A (v_A) and their normalized values. The proposed element yields accurate

Fig. 22 Thick and thin curved beams: **a** thick beam ($R=10$, $H=5$, $P=600$, $E=1000$, $\nu=0$, and thickness = 1) and **b** thin beams ($P=1$, $E=365,010$ (for case $H/R=0.03$) and $E=44,027,109$ (for case $H/R=0.006$), $\nu=0$, and thickness = 1)

Table 9 Normalized vertical deflection at point A in the thick curved beam shown in Fig. 22a; values in parentheses are final numbers of iterations for the SU4 element

Elements	Normalized deflection, v_A
Q4	0.643
Q8	0.990
QM6	0.939
P-S	0.939
PEAS7	0.939
QACM4	0.939
2D-MITC4/1	0.916
SU4	0.963 (3)
Reference solution [38]: $v_A = 90.1$	

Table 10 Vertical displacements at point A in the thin curved beam shown in Fig. 22b; values in parentheses are final numbers of iterations for the SU4 element

Elements	Vertical displacements	
	$H/R=0.03$	$H/R=0.006$
Q4	0.016	0.001
Q8	0.992	0.992
QM6	0.650	0.173
QACM4	0.639	0.026
SU4	1.005 (4)	1.003 (4)
Reference solution [38]: $v_A = 1.0$		

solutions even when the meshes are considerably coarse ($N=1$ and 2); Fig. 14a shows the normalized value of v_A with respect to the number of iterations. Within a few

iterations, the solution rapidly converges to the reference. The normalized value of v_A according to the number of layers (N) are shown Fig. 14b. The SU4 element has an excellent convergence behavior.

The convergence curves are shown in Fig. 15a; even when only one iteration is executed, the SU4 element outperforms the other elements. The convergence behavior of strain energy according to the number of iterations is illustrated in Fig. 15b. At the beginning of the iteration, the strain energy rapidly converges to a certain level. As the mesh becomes finer, the convergence with respect to the iterations becomes faster.

The shear stress contour plots obtained using SU4 elements with $N=8$ layers are shown in Fig. 16; the iterations are 0 and 1. The reference stress distribution shown in Fig. 16c is calculated using a 64×64 mesh of standard nine-node quadrilateral elements [10]. In the first iteration, the shear stress distribution in the SU4 element rapidly improves.

6.3 MacNeal's cantilever beam

Consider MacNeal's cantilever beam problem [29], which has been frequently employed to test the sensitivity of quadrilateral elements to mesh distortion. The cantilever beam shown in Fig. 17 has a length (L)=6, width (H)=0.2, and thickness=0.1. The plane stress condition is assumed, and the material properties are Young's modulus (E)= 1.0×10^7 and Poisson's ratio (ν)=0.3. The beam is clamped at the left end, and two loading cases are considered: (I) shear force $P=1$ and (II) moment $M=0.2$ at the free end. Three different mesh patterns, including regular (Mesh I),

Fig. 23 Bending directions calculated by the SU4 elements and effective stress distributions for the thin beam case ($H/R=0.03$); green arrows show bending directions, and values right next to arrows indi-

cate differences between the calculated and ideal bending directions; reference stress distribution obtained using standard nine-node quadrilateral elements

Fig. 24 Strap plate problem: **a** problem description ($q=10$ force per unit length, $E=2 \times 10^{11}$, $\nu=0.3$, and thickness=0.01), **b** reference mesh and **c** element meshes used

parallelogram (Mesh II), and trapezoidal (Mesh III) meshes, are considered.

Table 6 summarizes the normalized vertical displacements at point A. The reference solutions are -0.1081 for the shear load case and -0.0054 for the moment load case. Even with only one iteration, the SU4 element exhibits highly accurate solutions in all three mesh patterns regardless of mesh distortion. The SU4 element produces considerably more accurate solutions than the other elements with parallelogram (Mesh II) and trapezoidal (Mesh III) meshes.

6.4 Cantilever beam modeled using two elements with distortion parameter

To further test the mesh distortion sensitivity, modeling the cantilever beam using two elements is considered as

shown in Fig. 18. The geometry of the two elements varies with respect to the distortion parameter, e , which reflects the degree of element distortion. The cantilever beam has a length (L) = 10, width (H) = 2, and thickness = 1. The plane stress condition is considered, and the material properties are Young's modulus (E) = 1500 and Poisson's ratio (ν) = 0.25. The cantilever is subjected to a bending moment $M=2000$ at the right end. The exact solution of the vertical displacement at point A is 100, as given in Ref. [30].

The cantilever is modeled using two four-node elements. As parameter e varies from 0 to 4.9, the two finite elements become more distorted. The normalized displacements at point A are listed in Table 7 and Fig. 19. The SU4 element yields the exact solution regardless of the magnitude of e even with only one iteration, whereas the other elements suffer the effects of mesh distortion. The stress distributions

Fig. 25 Performance of the SUFE: **a** computational efficiency and **b** convergence curves for the strap plate problem; Computation time is measured in seconds, and the number of element layers (N) is displayed near each mark

that are calculated using elements Q4, QM6, and SU4 when $e=4.9$ are shown in Fig. 20.

6.5 Cantilever beams modeled using distorted elements

As shown in Fig. 21, a cantilever beam is modeled with five distorted quadrilateral elements. Two loading cases are considered: (I) moment $M=2000$ and (II) shear force $P=150$ at the free tip. The plane stress condition is considered, and the material properties are Young's modulus (E)=1500 and Poisson's ratio (ν)=0.25. Table 8 lists the vertical displacements at point A and stresses at point B. The SU4 element provides the exact solutions for load case (I) and highly accurate results for load case (II).

6.6 Thick and thin curved beams

Thick and thin curved beams fully clamped at the bottom are shown in Fig. 22. For the thick beam case, the geometry is given by $R=10$ and $H=5$, and the material properties are Young's modulus (E)=1000 and Poisson's ratio (ν)=0. A shear force, $P=600$, is applied to the free tip.

For the thin beam case, two ratios of thickness to radius ($H/R=0.03$ and 0.006) are considered. Poisson's ratio (ν)=0; when $H/R=0.03$, Young's modulus (E)=365,010, and when $H/R=0.006$, $E=44,027,109$. A vertical load, $P=1.0$, is applied at the free tip. The curved beam is modeled using only four or five elements, as shown in Fig. 22. For both

Fig. 26 Effective stress distributions obtained using the SU4 element for the strap plate problem

thick and thin beam problems, the plane stress condition is assumed.

Tables 9 and 10 summarize the normalized vertical displacements at point A for the thick and thin beams, respectively. Regardless of the thickness-to-radius ratio, the SU4 element exhibits excellent solutions. However, the solution accuracies of the other elements except for the Q8 element are extremely inadequate, especially those for the thin beam case.

The bending directions calculated using the SU4 elements for the thin beam case ($H/R=0.03$) are also investigated. The bending directions and effective stress (von Mises stress) distributions are shown in Fig. 23. Ideally, the bending direction should be toward the center of curvature of the thin beam. The differences between the calculated and ideal bending directions are less than 0.4° . In addition, the SU4 elements provide a highly accurate stress distribution.

6.7 Strap plate problem

The strap plate problem is shown in Fig. 24. The strap plate has three holes, two of which are fixed. A uniformly distributed load, $q = 10$ (force per unit length), in the negative y -direction is applied to the right hole of the plate. The plane stress condition is considered with Poisson's ratio (ν) = 0.3 and Young's modulus (E) = 2×10^{11} . The solutions are obtained using coarse, medium, and fine meshes. The reference solutions are calculated using the reference mesh of standard nine-node quadrilateral elements with the number of element layers $N = 50$ (109,582 DOFs), as shown in Fig. 24b.

The computational efficiency of the SU4 element is investigated. Figure 25a shows the relation between the computation time versus the relative strain energy error in Eq. (44) for the strap plate problem. The meshes in Fig. 24c are used. The SU4 element shows better computational efficiency among the tested elements. In other words, the SU4 element requires less computation time compared to other elements when obtaining similar solution accuracy.

The convergence curve is shown in Fig. 25b. The proposed element provides accurate solutions with few DOFs. Figure 26 displays the effective stress (von Mises stress) distributions obtained using the proposed SU4 element.

7 Conclusions

In this paper, an SUFE and an iterative solution procedure were proposed. This new concept affords the capability to improve the accuracy of FE solutions without mesh refinement. The assumed modal strain was employed based on the mode-based FE formulation, leading to the SUFE. To calculate the element stiffness matrix for a certain element deformation, the search for the optimal bending directions is required. To reduce the search time, deep learning is employed. The performance of SUFE was investigated through various numerical examples. The accuracy and computational efficiency of the proposed SU4 element were compared with those of other existing four-node elements. The SUFE produced highly accurate solutions with few iterations, especially when the meshes are coarse and distorted.

This study only examined the four-node finite element. However, the proposed concept can be extended to various types of finite elements, including solid, plate, and shell elements [31–36]. The conduct of research on such elements in the future will certainly be advantageous.

Acknowledgements This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (No. NRF-2018R1A2B3005328). This research

was also supported by a GPU server support program supervised by the NIPA (National IT Industry Promotion Agency).

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Zienkiewicz O, Taylor R, Too J (1971) Reduced integration technique in general analysis of plates and shells. *Int J Numer Methods Eng* 3:275–290. <https://doi.org/10.1002/nme.1620030211>
2. Pawsey SF, Clough RW (1971) Improved numerical integration of thick shell finite elements. *Int J Numer Methods Eng* 3:575–586. <https://doi.org/10.1002/nme.1620030411>
3. Naylor D (1974) Stresses in nearly incompressible materials by finite elements with application to the calculation of excess pore pressures. *Int J Numer Methods Eng* 8:443–460. <https://doi.org/10.1002/nme.1620080302>
4. Barlow J (1976) Optimal stress locations in finite element models. *Int J Numer Methods Eng* 10:243–251. <https://doi.org/10.1002/nme.1620100202>
5. Wilson E, Taylor R, Doherty W, Ghaboussi J (1973) Incompatible displacement models. *Numerical and computer methods in structural mechanics*. Elsevier, Amsterdam, pp 43–57
6. Taylor RL, Beresford PJ, Wilson EL (1976) A non-conforming element for stress analysis. *Int J Numer Methods Eng* 10:1211–1219. <https://doi.org/10.1002/nme.1620100602>
7. Pian TH, Sumihara K (1984) Rational approach for assumed stress finite elements. *Int J Numer Methods Eng* 20:1685–1695. <https://doi.org/10.1002/nme.1620200911>
8. Macneal RH (1982) Derivation of element stiffness matrices by assumed strain distributions. *Nucl Eng Des* 70:3–12. [https://doi.org/10.1016/0029-5493\(82\)90262-x](https://doi.org/10.1016/0029-5493(82)90262-x)
9. Simo J-C, Armero F (1992) Geometrically non-linear enhanced strain mixed methods and the method of incompatible modes. *Int J Numer Methods Eng* 33:1413–1449. <https://doi.org/10.1002/nme.1620330705>
10. Bathe KJ (2014) *Finite element procedures*, second edition, Klaus-Jürgen Bathe, Watertown, MA
11. Royston D (2001) Transformation of stresses and strains. *Lecture Notes for Mechanics of Materials*, Massachusetts Institute of Technology, Cambridge
12. Andelfinger U, Ramm E (1993) EAS-elements for two-dimensional, three-dimensional, plate and shell structures and their equivalence to HR-elements. *Int J Numer Methods Eng* 36:1311–1337. <https://doi.org/10.1002/nme.1620360805>
13. Piltner R, Taylor R (1995) A quadrilateral mixed finite element with two enhanced strain modes. *Int J Numer Methods Eng* 38:1783–1808. <https://doi.org/10.1002/nme.1620381102>

14. Rajendran S, Zhang B (2007) A “FE-meshfree” QUAD4 element based on partition of unity. *Comput Methods Appl Mech Eng* 197:128–147. <https://doi.org/10.1016/j.cma.2007.07.010>
15. Xu J, Rajendran S (2011) A partition-of-unity based ‘FE-Meshfree’ QUAD4 element with radial-polynomial basis functions for static analyses. *Comput Methods Appl Mech Eng* 200:3309–3323. <https://doi.org/10.1016/j.cma.2011.08.005>
16. Wu C-C, Huang M-G, Pian TH (1987) Consistency condition and convergence criteria of incompatible elements: general formulation of incompatible functions and its application. *Comput Struct* 27:639–644. [https://doi.org/10.1016/0045-7949\(87\)90080-0](https://doi.org/10.1016/0045-7949(87)90080-0)
17. Jun H, Yoon K, Lee PS, Bathe KJ (2018) The MITC3+ shell element enriched in membrane displacements by interpolation covers. *Comput Methods Appl Mech Eng* 337:458–480. <https://doi.org/10.1016/j.cma.2018.04.007>
18. Yu Y, Hur T, Jung J, Jang IG (2019) Deep learning for determining a near-optimal topological design without any iteration. *Struct Multidiscipl Optim* 59:787–799. <https://doi.org/10.1007/s00158-018-2101-5>
19. Baiges J, Codina R, Castañar I, Castillo E (2020) A finite element reduced-order model based on adaptive mesh refinement and artificial neural networks. *Int J Numer Methods Eng* 121:588–601. <https://doi.org/10.1002/nme.6235>
20. Oishi A, Yagawa G (2020) Finite elements using neural networks and a posteriori error. *Arch Comput Methods Eng*. <https://doi.org/10.1007/s11831-020-09507-0>
21. Jung J, Yoon K, Lee PS (2020) Deep learned finite elements. *Comput Methods Appl Mech Eng* 372:113401. <https://doi.org/10.1016/j.cma.2020.113401>
22. Gao F, Han L (2012) Implementing the Nelder-Mead simplex algorithm with adaptive parameters. *Comput Optim Appl* 51:259–277. <https://doi.org/10.1007/s10589-010-9329-3>
23. Abadi M, Barham P, Chen J, Chen Z, Davis A, Dean J, Devin M, Ghemawat S, Irving G, Isard M et al (2016) TensorFlow: a system for large-scale machine learning. In: 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16), 265–283
24. Kingma DP, Ba J (2017) Adam: a method for stochastic optimization, ArXiv:1412.6980 [Cs]. <http://arxiv.org/abs/1412.6980> Accessed 13 Nov 2020
25. Glorot X, Bengio Y (2010) Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the thirteenth international conference on artificial intelligence and statistics, 249–256
26. Van Der Walt S, Colbert SC, Varoquaux G (2011) The NumPy array: a structure for efficient numerical computation. *Comput Sci Eng* 13:22–30. <https://doi.org/10.1109/MCSE.2011.37>
27. Cen S, Zhou PL, Li CF, Wu CJ (2015) An unsymmetric 4-node, 8-DOF plane membrane element perfectly breaking through MacNeal’s theorem. *Int J Numer Methods Eng* 103:469–500. <https://doi.org/10.1002/nme.4899>
28. Cook RD et al (2001) Concepts and applications of finite element analysis, 4th edn. John Wiley & sons, New York
29. Macneal RH, Harder RL (1985) A proposed standard set of problems to test finite element accuracy. *Finite Elem Anal Des* 1:3–20. [https://doi.org/10.1016/0168-874X\(85\)90003-4](https://doi.org/10.1016/0168-874X(85)90003-4)
30. Sze K (2000) On immunizing five-beta hybrid-stress element models from ‘trapezoidal locking’ in practical analyses. *Int J Numer Methods Eng* 47:907–920. [https://doi.org/10.1002/\(sici\)1097-0207\(20000210\)47:4<907::aid-nme808>3.0.co;2-a](https://doi.org/10.1002/(sici)1097-0207(20000210)47:4<907::aid-nme808>3.0.co;2-a)
31. Ko Y, Lee PS, Bathe KJ (2017) A new 4-node MITC element for analysis of two-dimensional solids and its formulation in a shell element. *Comput Struct* 192:34–49. <https://doi.org/10.1016/j.compstruc.2017.07.003>
32. Jun H, Mukai P, Kim S (2018) Benchmark tests of MITC triangular shell elements. *Struct Eng Mech* 68:17–38. <https://doi.org/10.12989/sem.2018.68.1.017>
33. Lee C, Lee PS (2018) A new strain smoothing method for triangular and tetrahedral finite elements. *Comput Methods Appl Mech Eng* 341:939–955. <https://doi.org/10.1016/j.cma.2018.07.022>
34. Lee C, Lee PS (2019) The strain-smoothed MITC3+ shell finite element. *Comput Struct* 223:106096. <https://doi.org/10.1016/j.compstruc.2019.07.005>
35. Kim S, Lee PS (2018) A new enriched 4-node 2D solid finite element free from the linear dependence problem. *Comput Struct* 202:25–43. <https://doi.org/10.1016/j.compstruc.2018.03.001>
36. Ko Y, Lee PS (2017) A 6-node triangular solid-shell element for linear and nonlinear analysis. *Int J Numer Methods Eng* 111:1203–1230. <https://doi.org/10.1002/nme.5498>
37. Piltner R, Taylor R (1999) A systematic construction of B-bar functions for linear and non-linear mixed-enhanced finite elements for plane elasticity problems. *Int J Numer Methods Eng* 44:615–639. [https://doi.org/10.1002/\(sici\)1097-0207\(19990220\)44:5<615::aid-nme518>3.0.co;2-u](https://doi.org/10.1002/(sici)1097-0207(19990220)44:5<615::aid-nme518>3.0.co;2-u)
38. Cen S, Chen X-M, Fu X-R (2007) Quadrilateral membrane element family formulated by the quadrilateral area coordinate method. *Comput Methods Appl Mech Eng* 196:4337–4353. <https://doi.org/10.1016/j.cma.2007.05.004>
39. Long Y-Q, Cen S, Long Z-F (2009) Advanced finite element method in structural engineering. Springer-Verlag, Berlin Heidelberg

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.