Computers and Structures 330 (2026) 108331

Contents lists available at ScienceDirect

Computers and Structures

journal homepage: www.elsevier.com/locate/compstruc

# The patch tests for verification of machine learning procedures

Klaus-Jürgen Bathe, Sanghee Lee

*Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139, USA*

A R T I C L E I N F O

*Keywords:*
Machine learning
Neural networks
Finite elements
Basic patch tests
Consequences not passing the tests
Convergence

A B S T R A C T

During recent years, a large research effort has focused on the use of machine learning with Neural Networks to solve problems in sciences and engineering, with many procedures proposed. However, in almost no work did the authors perform some basic reliability tests whichever may be applicable. While we exemplify in this paper the use of basic tests in the analysis of solids, the fundamental broad point of reliability we make should be looked at as much more general. In finite element static analyses, hence considering only the spatial discretization, the patch tests including the rigid body mode tests are the basic tests that need to be passed. In direct integration solutions of dynamic equilibrium equations, additional tests need to be passed, like to assure that the integration scheme gives correct results when displacements are imposed, and not just external forces. These basic tests must be passed for any finite element procedure to be accepted in industry. The tests need to also be passed by a machine learning procedure focused on analyses of solids and fluids to verify the scheme and assure its reliability. This is important for the proposed method to be accepted in significant scientific studies and especially in industry. In this paper we describe the basic tests for spatial discretizations, emphasize their importance and illustrate how a machine learning scheme may seemingly work in the solution of some problems, but not pass the patch tests and hence not be sufficiently reliable for general use. Such schemes are not candidates for use in industry, when used on their own or to enhance finite element methods.

## 1. Introduction

As is well-known, the finite element method is very widely used in engineering analyses and scientific studies, see for example Refs. [1–3]. This success is based on various properties of the finite element method including the reliable and efficient use of the technique – the most important property being the reliability of the method [1,2]. Without these features, the finite element method would not have been accepted by engineers and scientists because it is of paramount importance to use reliable analysis techniques that are also efficient for the study of engineering designs and in any scientific investigation.

The reliability of the method also includes its robustness which means that in linear analysis for small changes in geometry, material properties or loading conditions the response changes will not be unduly large. Mathematical analyses have been performed in this regard, notably the inf-sup conditions need to be passed, for example, considering analyses of incompressible media and plates and shells [1,4].

The reliability of the method additionally means that the analysis procedure must never fail provided the mathematical problem is well-posed, that is, reasonable material properties are used, and the natural (force) and essential (displacement) boundary conditions are properly imposed and lead to a physically stable problem definition.

A very important contribution in identifying whether a finite element method and its implementation are reliable was made by Bruce Irons in proposing the patch tests [5]. Finite element procedures must satisfy these tests which is a requirement in all program developments [1,2], and in particular in the development of commercial finite element software like ANSYS, ADINA, NASTRAN, ABAQUS. The tests have been used for many years and certainly in the development of ADINA since the inception of the program [6]. Satisfying the patch tests means that the finite element formulation is sound and valid, the program implementation is error-free, and convergence of the solution scheme is assured. To ensure that the patch tests are satisfied should indeed be the first step in any mathematical analysis of a solution procedure.

While the patch tests were proposed to test the finite element spatial discretization used, for example, whether incompatibilities between elements prevent convergence, the role of the patch tests has been widened to also test direct time integration schemes [2,7]. Here we test whether a proposed solution method gives correct results when displacements are imposed, and not just external tractions.

In recent years, tremendous developments have taken place in Machine Learning with Neural Networks for the solution of partial differential equations, in particular the various Physics-Informed Neural Networks (PINN) have been proposed. While numerous schemes in papers have been published, see for example Refs. [8–28], and give an alternative avenue for solving analysis problems, the reliability of the proposed methods is usually not sufficiently addressed, neither the efficiency. These publications also include proposals on how to enhance

https://doi.org/10.1016/j.compstruc.2026.108331
Received 14 May 2026; Accepted 13 June 2026
Available online 30 June 2026

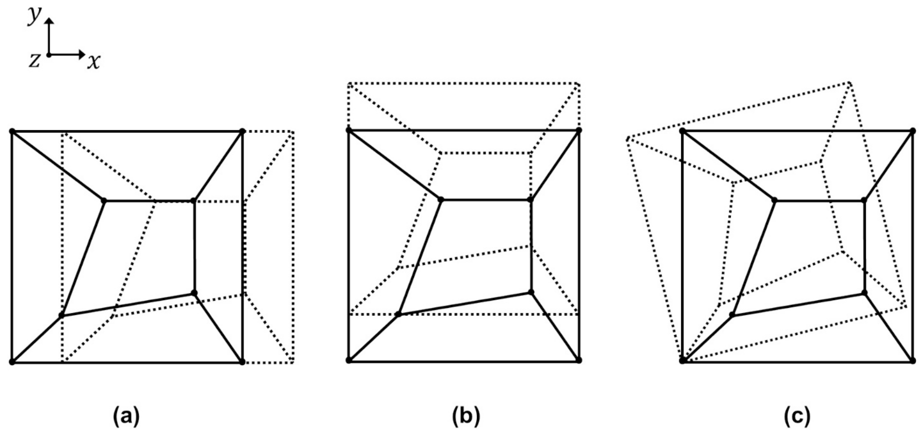


**Fig. 1.** Tests for the three rigid body modes: (a) u-horizontal rigid displacement, (b) v-vertical rigid displacement, and (c) rigid rotation about the z-axis.

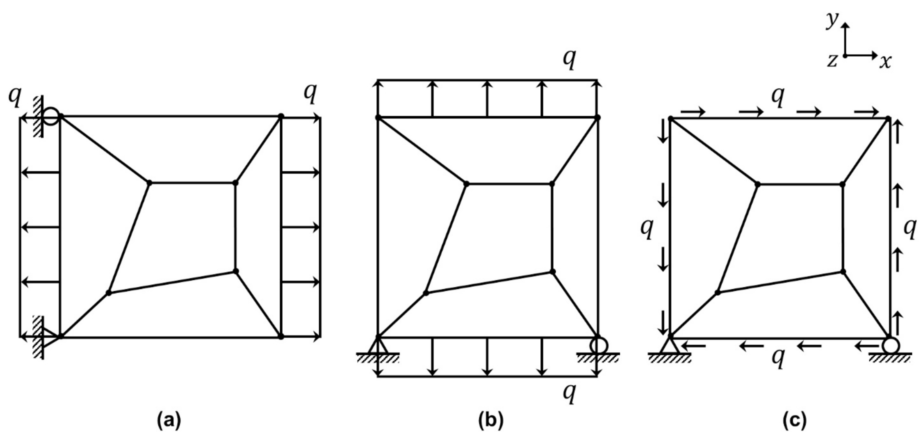


**Fig. 2.** Classical constant stress/ strain patch tests in plane stress/ strain conditions: (a) uniform extension in the x-direction, (b) uniform extension in the y-direction, (c) uniform shear.

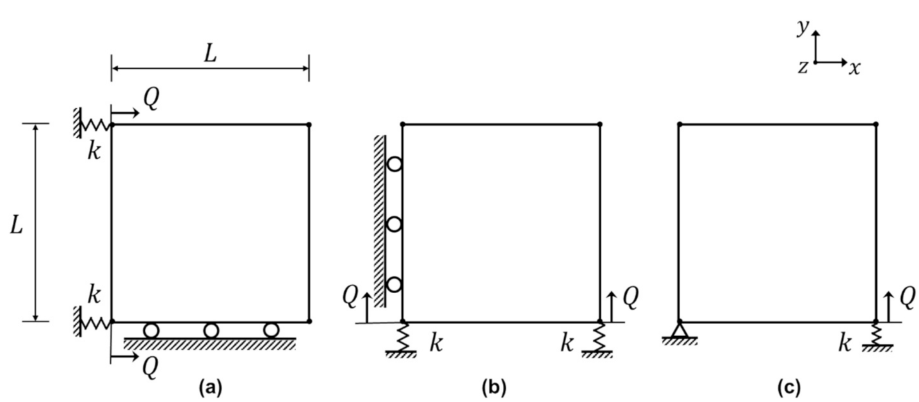


**Fig. 3.** Machine learning models to test for rigid body modes: (a) u-horizontal rigid displacement, (b) v-vertical rigid displacement, and (c) rigid rotation about the z-axis.

finite element methods, see for example Refs. [21–24] but these are also hardly tested for their reliability and efficiency. Only few studies have given focus on using a patch test for the evaluation of the reliability of the proposed scheme [29–31] or mention a patch test [27,28,32]. Furthermore, when applicable, the proposed methods are not evaluated versus the established and newly proposed finite element procedures, like the overlapping finite elements [33,34].

A minimum threshold is to test the proposed solution schemes on the full simple patch tests described in this paper.

The objective in this paper is to review the patch tests for spatial discretizations and show their applicability when using machine learning methods. The importance of performing the basic tests is

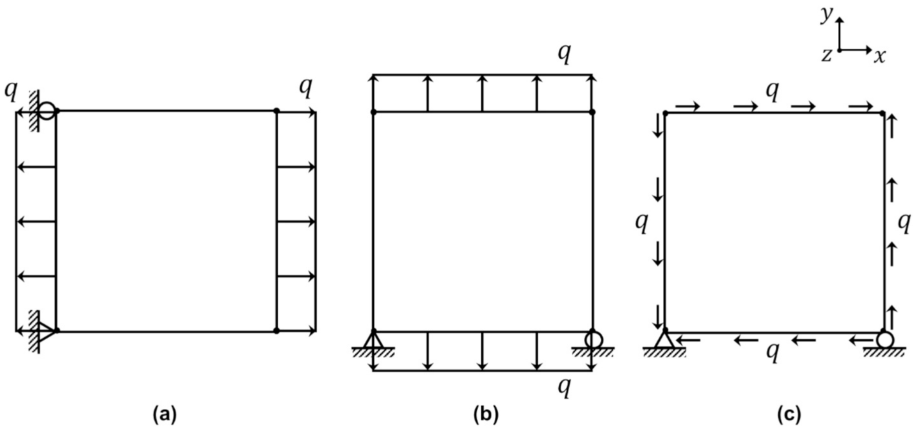


**Fig. 4.** Machine learning models used to test for constant stresses/ strains: (a) uniform extension in the x-direction, (b) uniform extension in the y-direction, (c) uniform shear.

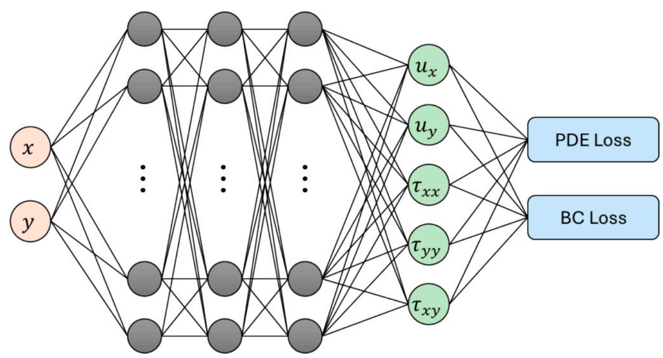


**Fig. 5.** Schematic of the stress–displacement PINN: the network maps spatial coordinates $(x, y)$ to $(u_x, u_y, \tau_{xx}, \tau_{yy}, \tau_{xy})$ and is trained via PDE and boundary-condition (BC) losses.

pointed out and we demonstrate that if the tests are not passed, the machine learning results may *look* acceptable in some analyses but are obviously in general not acceptable solutions. Hence, such machine learning methods are clearly not reliable. The discussion and illustrations underline the importance of performing the patch tests in the verification of machine learning procedures for the solution of static and dynamic responses, and in general when the procedure is designed to solve partial differential equations.

While we exemplify our thoughts here in the use of machine learning procedures operating on the mathematical framework of the principle of virtual work, the fundamental points we make are more general, namely, applicable when proposing machine learning procedures for all areas based on universal approximation theorems. Reliability can only be achieved if the machine learning procedure is correctly designed and verified with appropriate basic tests. These, of course, need to be designed depending on the problems that are addressed.

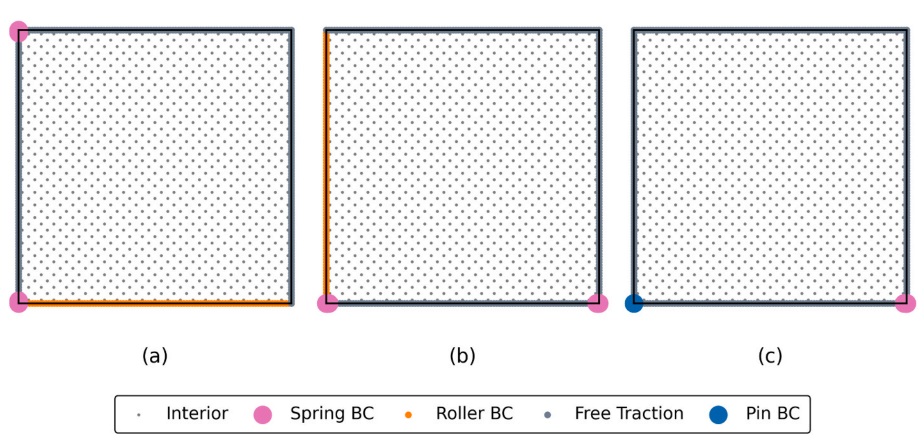


**Fig. 6.** Distribution of interior and boundary collocation points sampled in the tests for rigid body modes: (a) $u$-horizontal rigid displacement, (b) $v$-vertical rigid displacement, and (c) rigid rotation about the $z$-axis. Due to the high density, the boundary points appear as continuous solid lines.

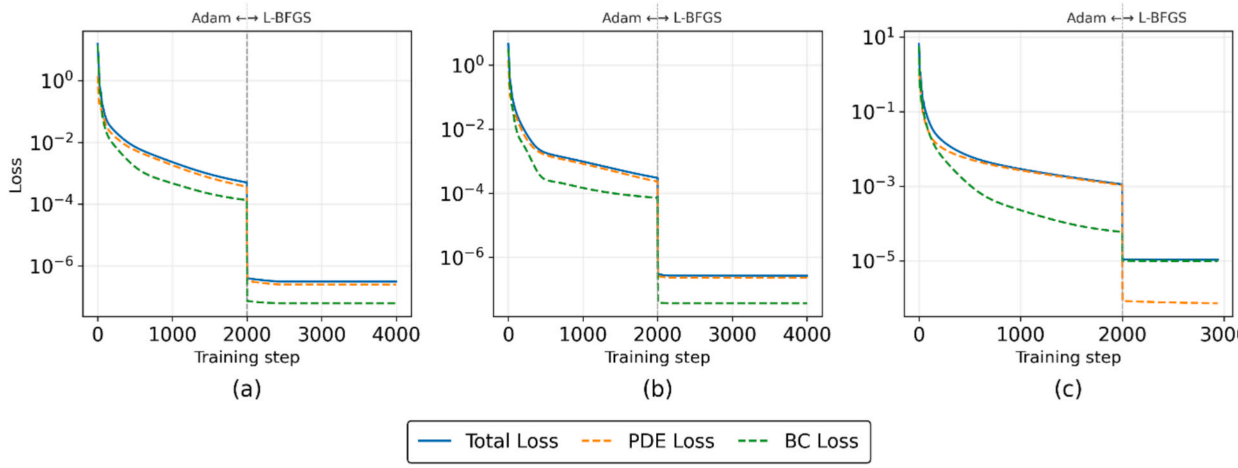


**Fig. 7.** Training loss histories for the test for rigid body modes showing total, PDE, and boundary condition (BC) losses. The vertical dashed line indicates the transition from Adam to L-BFGS optimization: (a) $u$-horizontal rigid displacement, (b) $v$-vertical rigid displacement, and (c) rigid rotation about the $z$-axis.

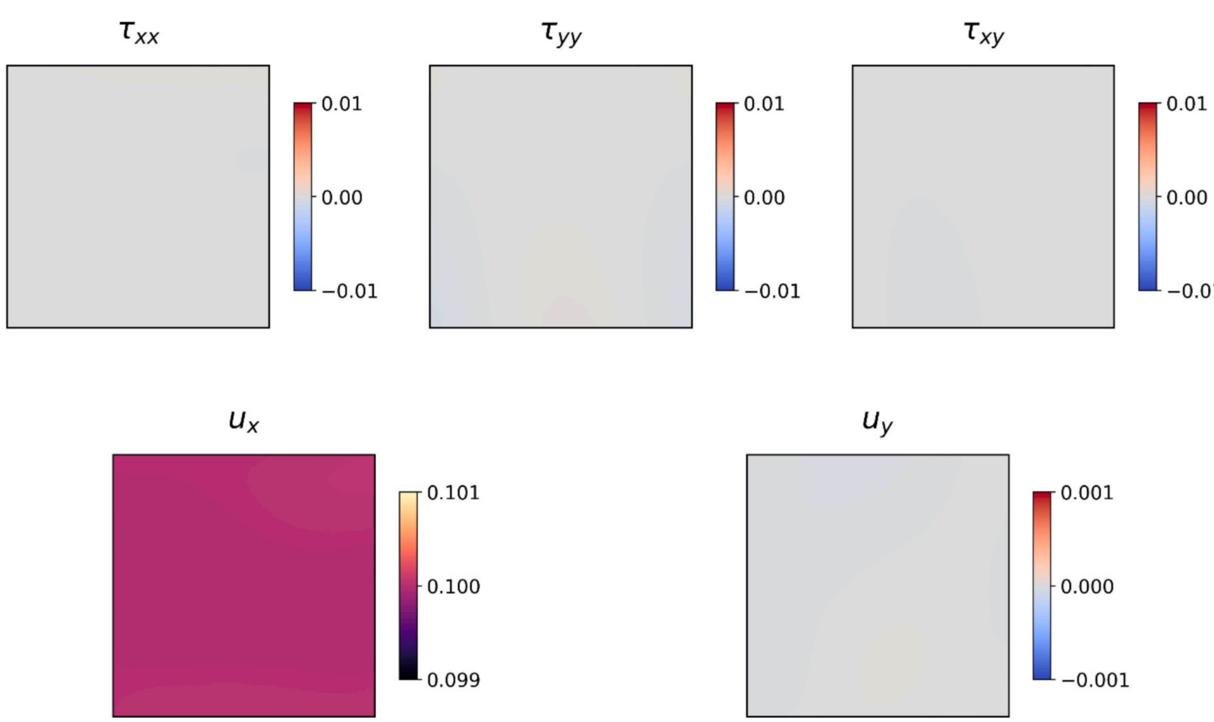


**Fig. 8.** Neural network predictions for $u$-horizontal rigid displacement. The top row shows the stress components $(\tau_{xx}, \tau_{yy}, \tau_{xy})$, and the bottom row shows the displacement components $(u_x, u_y)$.

## 2. The patch tests

We discuss in this section how the patch tests are performed in traditional finite element developments, and then we propose how they can be considered in verifying machine learning procedures. This will show that while patch tests are related to consistency and convergence of finite element methods [1], the proposed tests constitute here a basic verification of machine learning model formulations and implementations.

### 2.1. The traditional patch tests

The objective of the patch tests is to ensure that in the analysis of solids and structures.

- the rigid body modes are properly represented, and
- the constant stress states are also correctly represented.

While in general only small displacement/ small strain conditions are

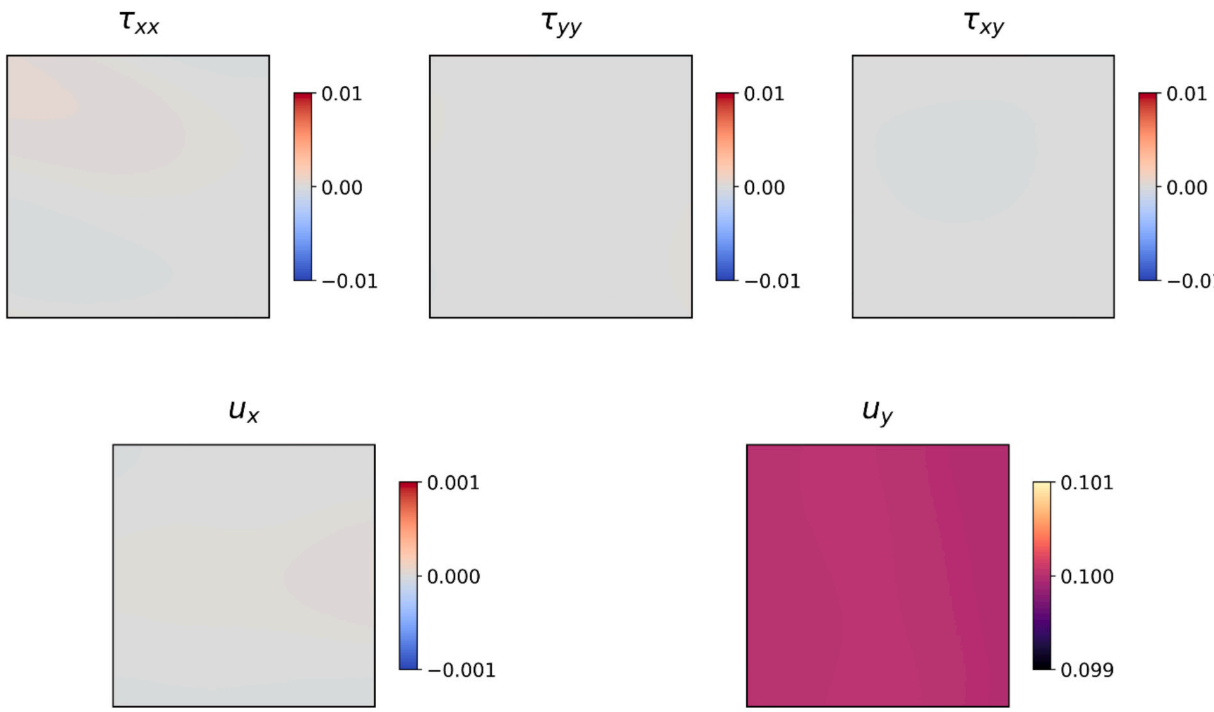


**Fig. 9.** Neural network predictions for $v$-vertical rigid displacement. The top row shows the stress components $(\tau_{xx}, \tau_{yy}, \tau_{xy})$, and the bottom row shows the displacement components $(u_x, u_y)$.

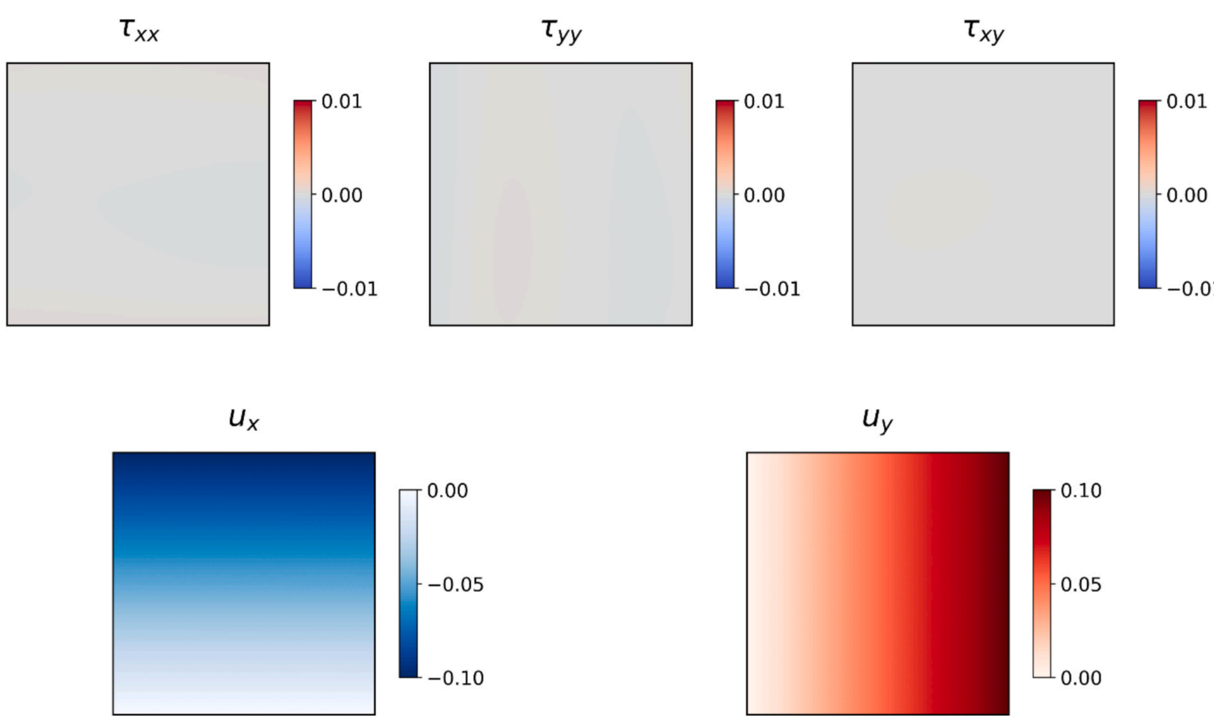


**Fig. 10.** Neural network predictions for rigid body rotation about the $z$-axis.. The top row shows the stress components $(\tau_{xx}, \tau_{yy}, \tau_{xy})$, and the bottom row shows the displacement components $(u_x, u_y)$.

considered, the tests hold also for nonlinear analyses because these are extensions of linear analyses [1]. While we focus on the analysis of solids, of course, similar conditions need also be studied in fluid flow and other analyses. First, we discuss the rigid body mode tests and then the constant strain / stress tests. We should also mention that the rigid body mode tests, while they need to be passed, are frequently not referred to as “patch tests” but we decided to include them in the tests.

Considering a plane stress or strain situation, see Fig. 1, any patch of the domain when discretized by finite elements needs to be able to represent the three rigid body modes, the u- horizontal and v-vertical rigid displacements and the rigid rotation about the z-axis. In Figs. 1 and 2 we use the patch of elements proposed in Ref [1]. In a three-dimensional analysis, three rigid body displacements and three rigid body rotations need to be represented, corresponding to the x-, y- and z-

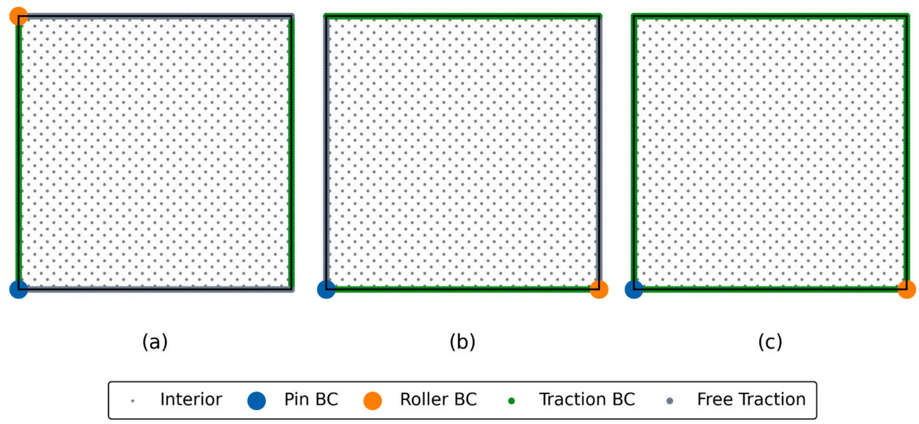


**Fig. 11.** Distribution of interior and boundary collocation points sampled in the constant stress/ strain conditions: (a) uniform extension in the *x*-direction, (b) uniform extension in the *y*-direction, and (c) uniform shear.

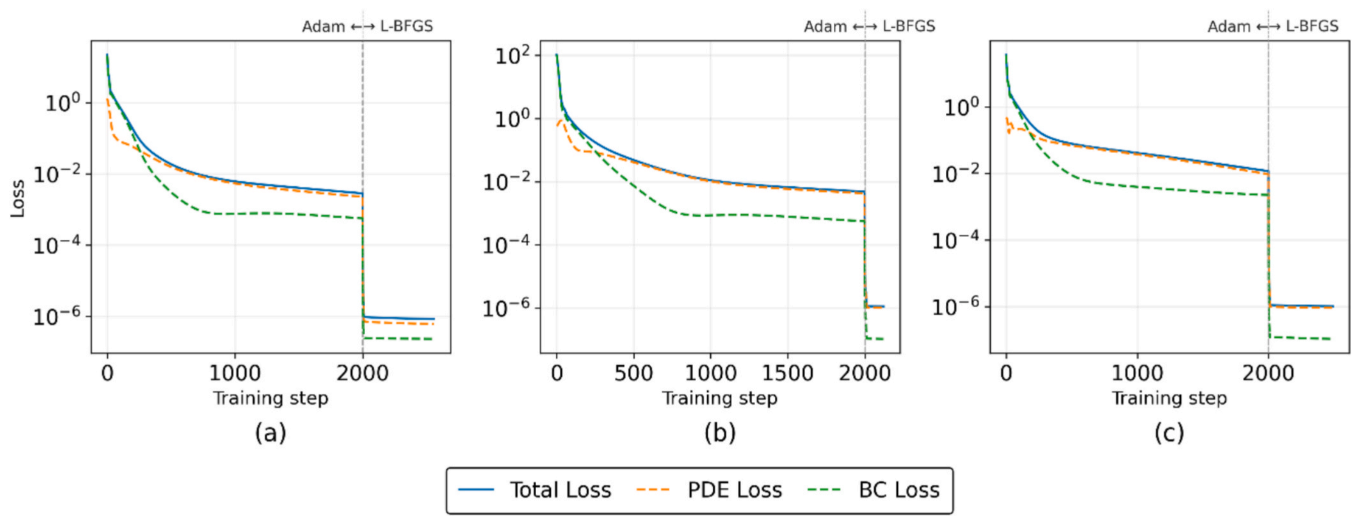


**Fig. 12.** Convergence histories for the tests for constant stresses/ strains. The total, PDE, and boundary condition (BC) losses are shown, and the dashed line marks the switch from Adam to L-BFGS: (a) uniform extension in the *x*-direction, (b) uniform extension in the *y*-direction, and (c) uniform shear.

axes. Each test case must not show any stresses in the elements. The same holds when only a single element is in the patch and for structural elements (beam, plate and shell elements) similar tests are applicable [1].

Of course, the rigid body motions correspond to zero eigenvalues of the stiffness matrix in finite element analysis when the patch of elements is unsupported, that is, free of any boundary conditions. If there are additional zero eigenvalues, these correspond to "spurious zero energy modes" and the solution scheme is unreliable [1].

In addition, any patch of elements must be able to correctly represent the constant stress/ strain states when the mathematical problem is well-posed for the analysis of the state. Fig. 2 shows how these stress/ strain states are imposed for plane stress or plane strain analyses. Of course, in three-dimensional and structural analyses, similar constant stress/ strain conditions are applicable, and in the general solution of partial differential equations also similar states need to be represented, and appropriate patch tests can be designed.

If these tests are passed, the basic conditions of convergence are satisfied, but of course, the rate of convergence is not assessed. The rate of convergence needs to be determined by mathematical analyses and additional numerical tests, like the inf-sup test, see for example [35,36].

Another important point is that the patch tests also show whether the program implementation has been correctly performed. If there is a programming error, it is likely that the patch tests are not passed and show the defect.

The above points are also of much value in the development of machine learning procedures. Here, the developer needs to use any means available to assure that the theoretical foundation and the implementation of the proposed procedure are defect-free and hence reliable for general use. Passing the patch tests – whichever are applicable – is one of the important requirements for methods to be acceptable for general analyses.

### 2.2. Modeling the patch tests for machine learning schemes

We demonstrate in this section how the patch tests can be performed when using a machine learning scheme.

In finite element analysis, the first step in testing an element is to

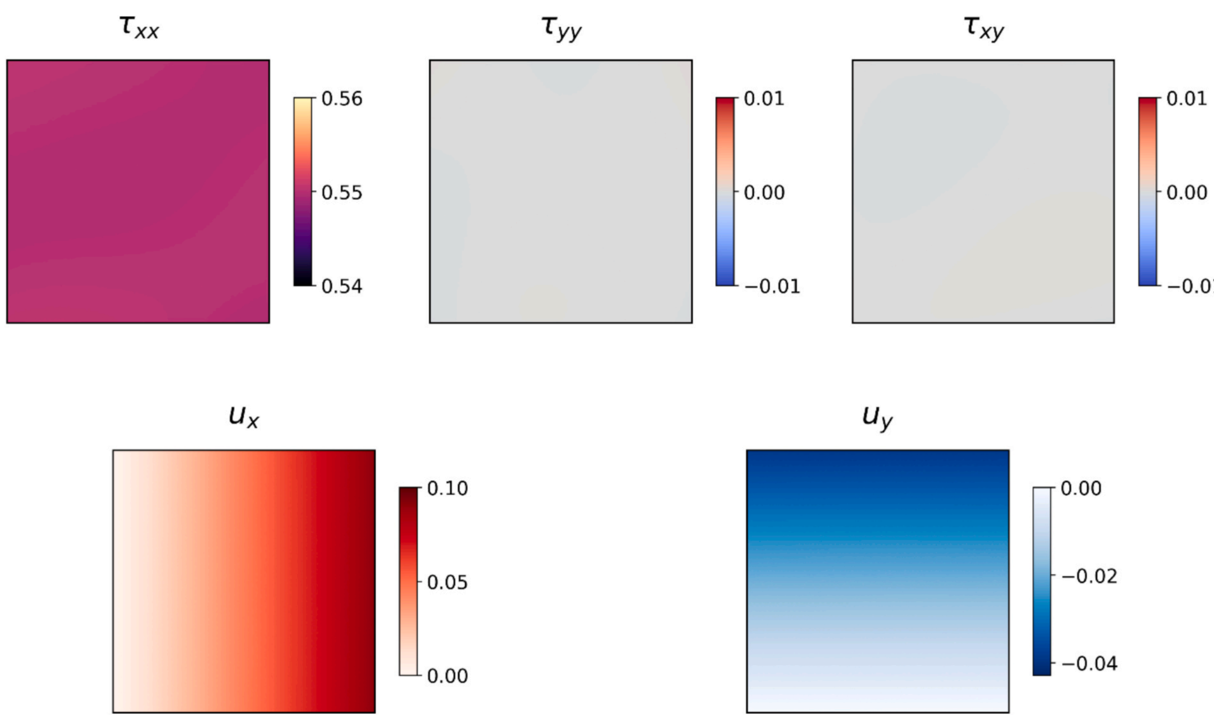


**Fig. 13.** Predicted stress fields and deformed configuration for the x-direction test of constant stresses/ strains. The stress components $(\tau_{xx}, \tau_{yy}, \tau_{xy})$ are shown in the top row, and the displacement components $(u_x, u_y)$ are shown in the bottom row.

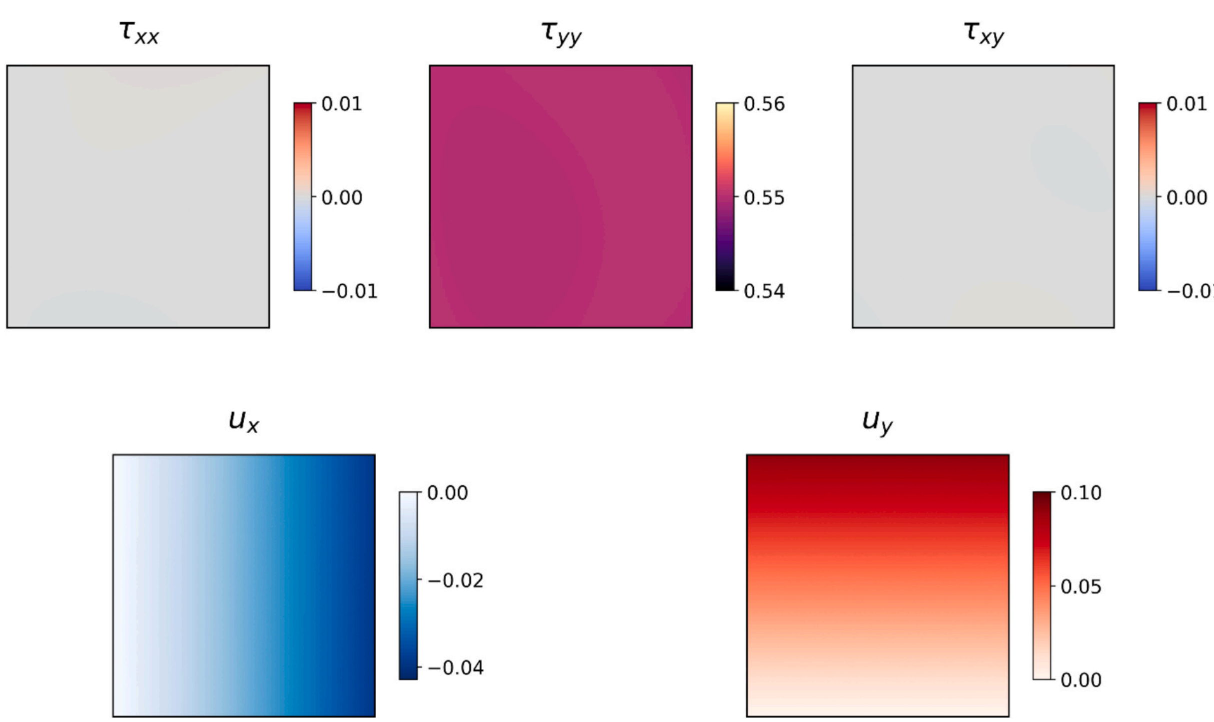


**Fig. 14.** Predicted stress fields and deformed configuration for the y-direction test of constant stresses/ strains. The stress components $(\tau_{xx}, \tau_{yy}, \tau_{xy})$ are shown in the top row, and displacement components $(u_x, u_y)$ are shown in the bottom row.

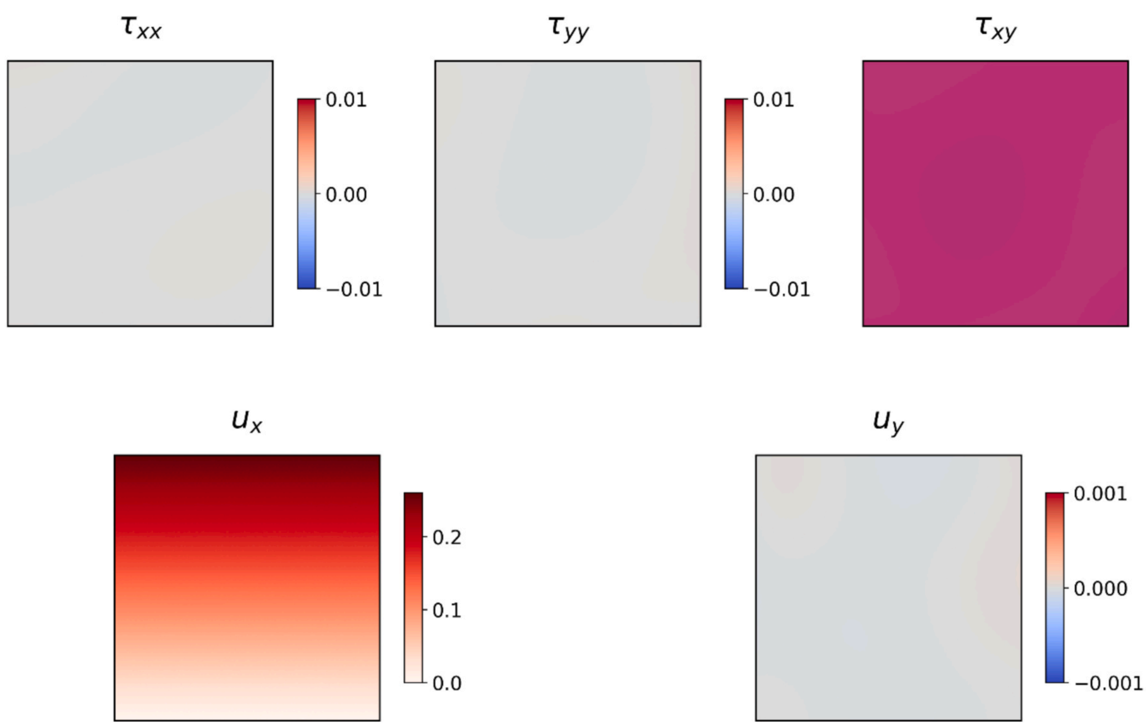


**Fig. 15.** Predicted stress fields and deformed configurations for the shear test of constant stresses/ strains. The stress components $(\tau_{xx}, \tau_{yy}, \tau_{xy})$ are shown in the top row, and the displacement components $(u_x, u_y)$ are shown in the bottom row.

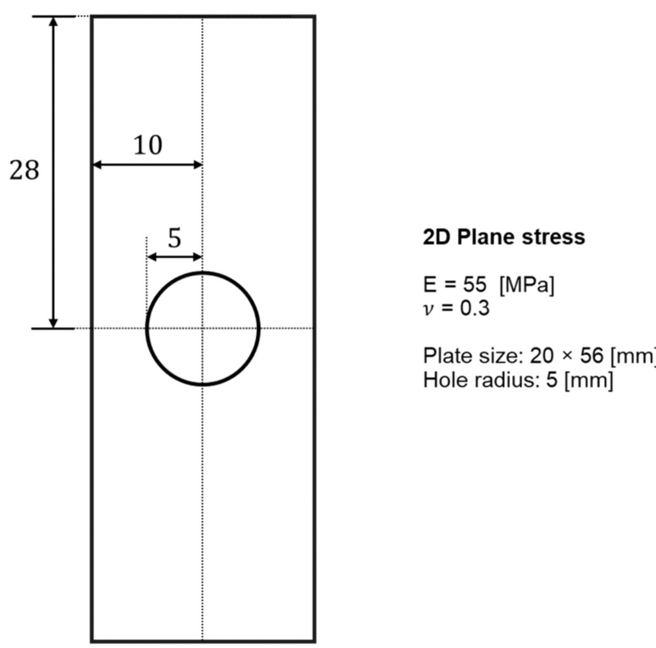


**Fig. 16.** The sheet (or plate in plane stress conditions) with the hole, the complete geometry and material data for the plane stress solution.

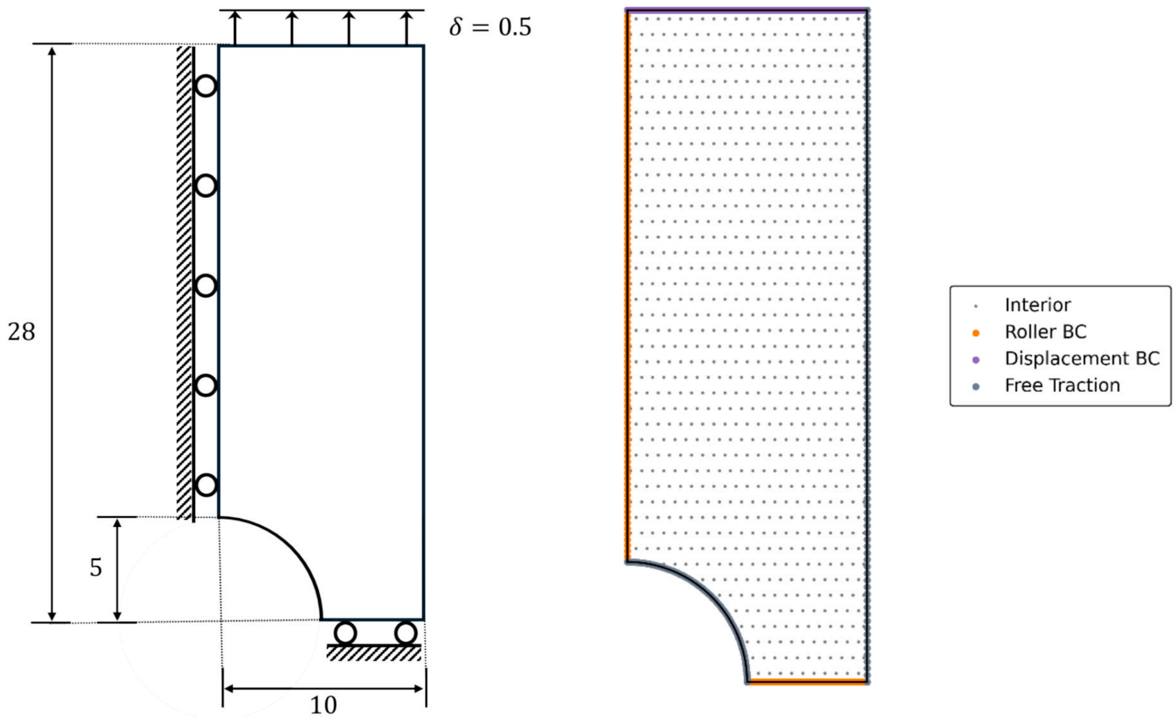


**Fig. 17.** The model of one quarter of the sheet subjected to an end-displacement (left) and the PDE collocation interior and boundary condition points (right). As in the case of the patch tests, the boundary collocation points are very close to each other.

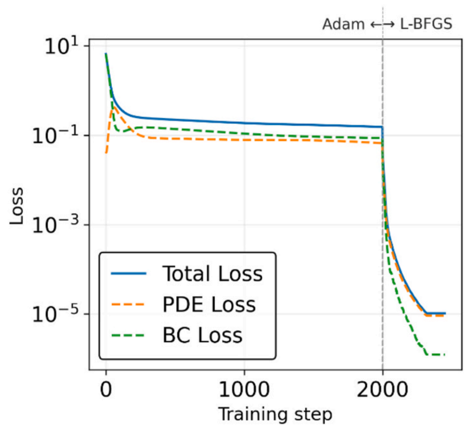


**Fig. 18.** Convergence data for the analysis of the sheet with the hole using the procedure of Fig. 5.

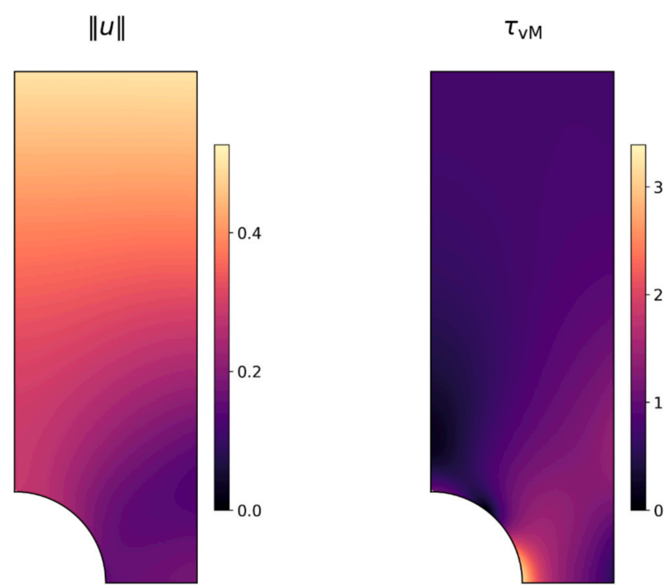


**Fig. 19.** Predicted displacement magnitude $\|u\|$ and von Mises stress ($\tau_{\text{vM}}$) for the quarter plate.

calculate its eigenvalues. There will be as many eigenvalues as the number of element degrees of freedom and in plane stress / strain analyses, exactly only three eigenvalues must be equal to zero (within round-off errors). Hence the fourth eigenvalue must be orders of magnitude larger than the third eigenvalue (when considering the round-off errors).

In machine learning usually the eigenvalues are not solved for, and only displacements and stresses, or fluid velocities and stresses are calculated. Hence, we propose for a plane stress/ strain solution capability, the analysis models shown in Fig. 3 to test whether the rigid body modes are properly represented in the machine learning procedure. Also, in case spurious zero energy modes are present, even if only one such mode is present, the machine learning scheme can be expected to not converge.

The constant stress/ strain situations can then be tested for using the models in Fig. 4 and in these solutions also difficulties will arise if a spurious zero energy mode is present.

### *2.3. A machine learning procedure and patch test results obtained*

The machine learning scheme that we employ is shown in Fig. 5. We use a standard physics-informed neural network (PINN) based on a fully connected multi-layer perceptron (MLP) with three hidden layers, each containing 64 neurons and *tanh* activation functions. Assuming a two-

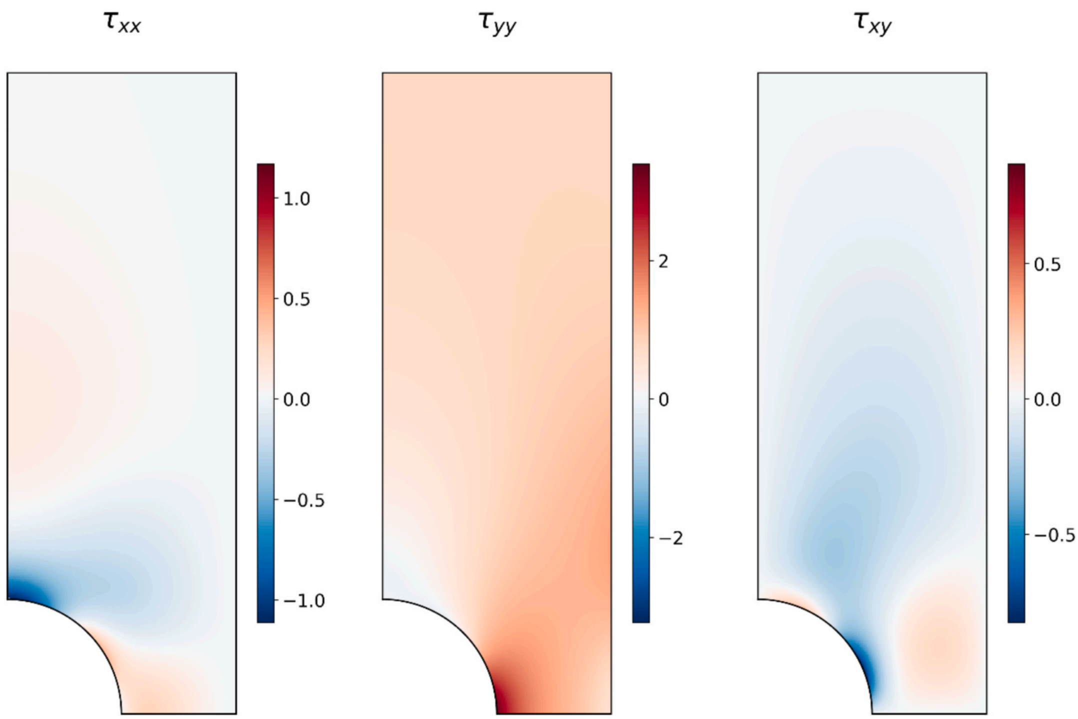


**Fig. 20.** Predicted stress components $(\tau_{xx}, \tau_{yy}, \tau_{xy})$ for the quarter plate.

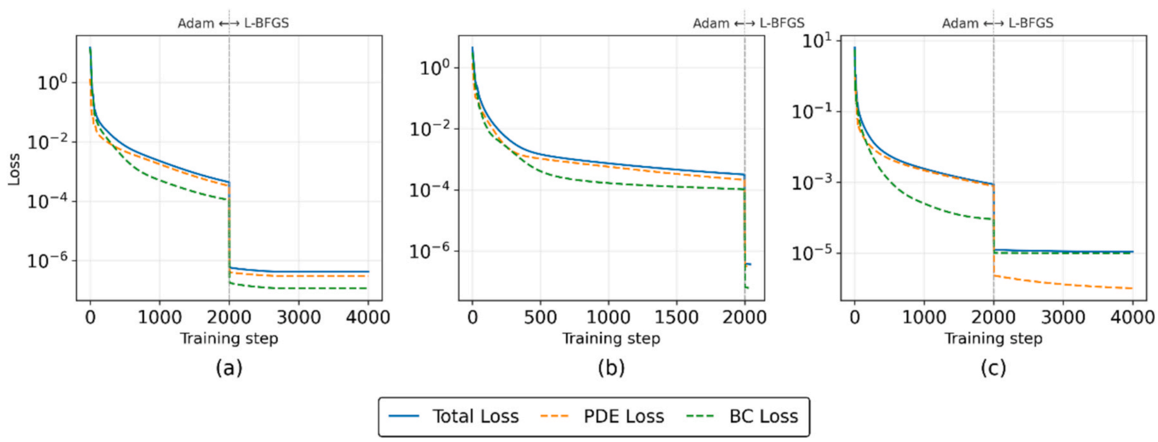


**Fig. 21.** Training loss histories for the test for rigid body modes using the machine learning scheme with an error, showing total, PDE, and boundary condition losses. The vertical dashed line indicates the transition from Adam to L-BFGS optimization: (a) $u$-horizontal rigid displacement, (b) $v$-vertical rigid displacement, and (c) rigid rotation about the $z$-axis.

dimensional elasticity problem, the model approximates the displacement and stress fields with a neural network $\mathscr{N}_\theta$,

$$\left(\widehat{u}_x, \widehat{u}_y, \widehat{\tau}_{xx}, \widehat{\tau}_{yy}, \widehat{\tau}_{xy}\right)(x,y) = \mathscr{N}_\theta(x,y), (x,y) \in \Omega, \tag{1}$$

where $\mathscr{N}_\theta$ denotes the neural network parameterized by $\theta$, and $(\widehat{u}_x, \widehat{u}_y)$ and $(\widehat{\tau}_{xx}, \widehat{\tau}_{yy}, \widehat{\tau}_{xy})$ represent the predicted displacement and stress components, respectively. The network takes the spatial coordinates (x, y) as input and outputs the corresponding field variables over the domain Ω. The model is trained by minimizing a physics-informed loss function defined as

$$\mathscr{L} = \lambda_{pde}\mathscr{L}_{\text{pde}} + \lambda_{BC}\mathscr{L}_{\text{BC}} \tag{2}$$

where $\mathscr{L}_{\text{pde}}$ enforces the equilibrium equations and the constitutive relations in the domain (PDE loss), $\mathscr{L}_{\text{BC}}$ imposes the boundary conditions on $\partial\Omega$ (BC loss) and $\lambda_{\text{pde}}$ and $\lambda_{\text{BC}}$ are weighting coefficients. The losses are evaluated at interior and boundary collocation points. For the two plate-with-a-hole analyses cases the network uses 1000 interior collocation points, and for the remaining cases 250 interior collocation points are employed. To enforce the boundary conditions, 100 collocation points were sampled on each line or arc segment corresponding to an individual boundary-condition term. For further details regarding the formulation, see Appendix A.

All models were trained using the PyTorch framework [37]. The numerical experiments were conducted on the MIT Engaging cluster

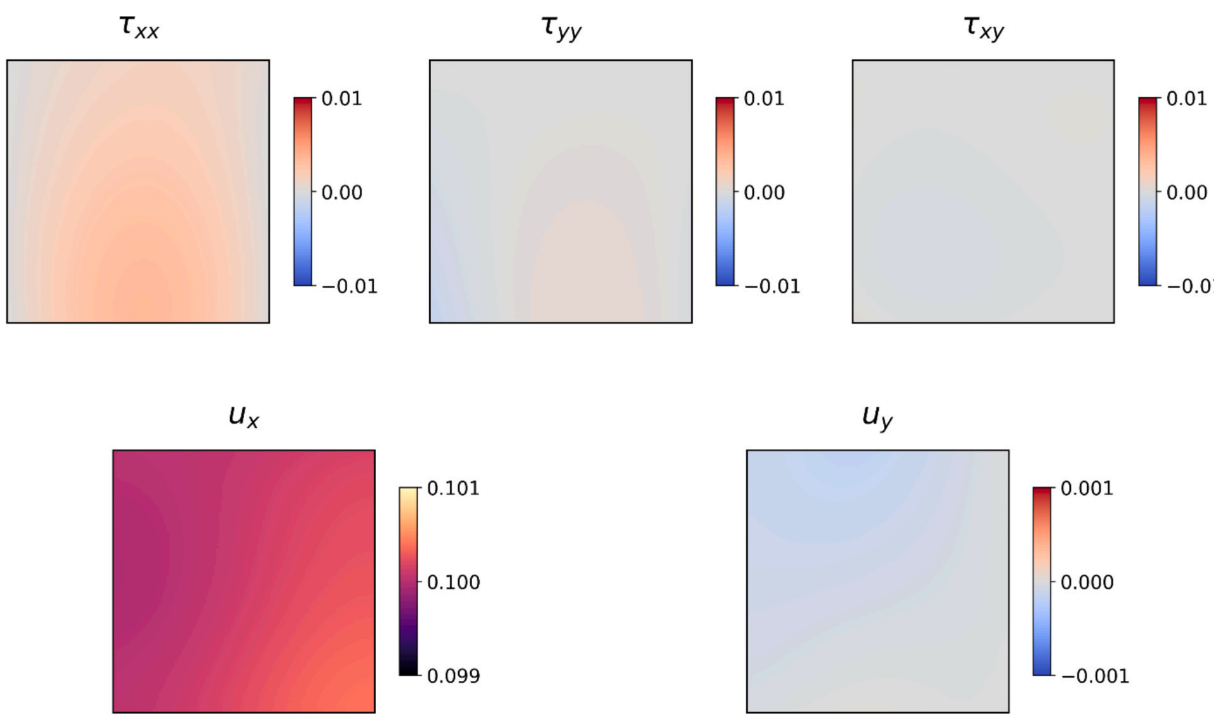


**Fig. 22.** Neural network predictions for the *u*-horizontal rigid displacement using the machine learning scheme with an error. The top row shows the stress components $(\tau_{xx}, \tau_{yy}, \tau_{xy})$, and the bottom row shows the displacement components $(u_x, u_y)$.

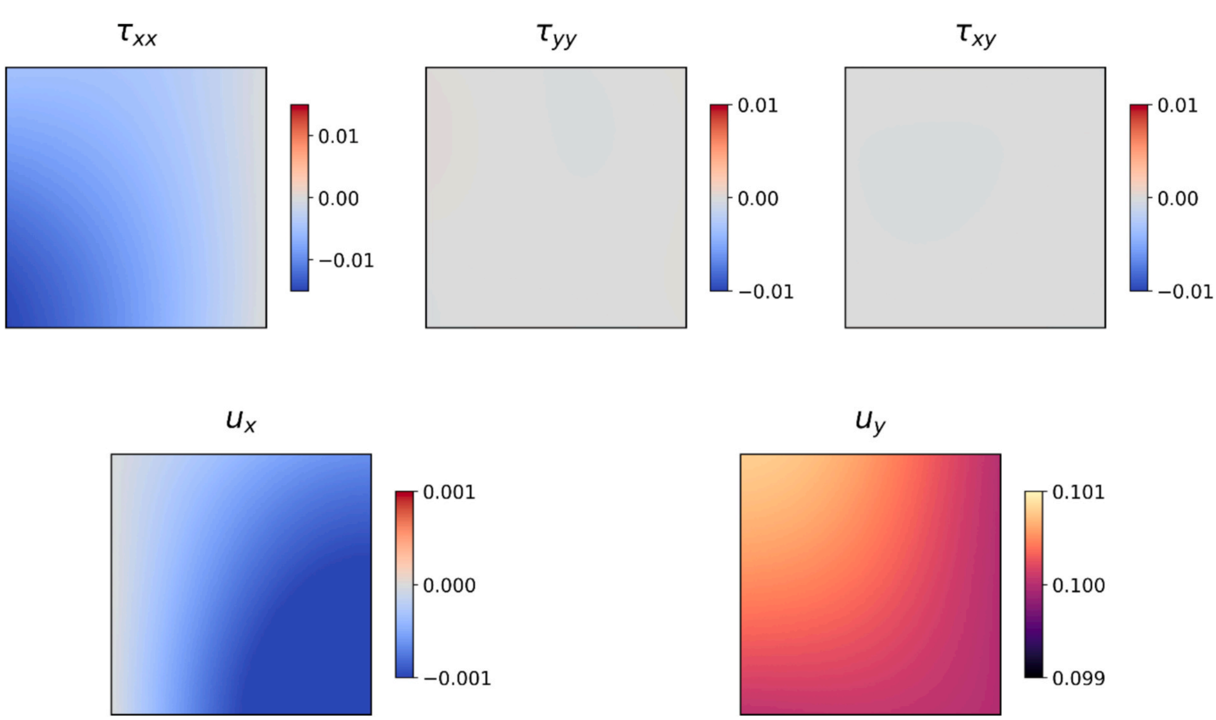


**Fig. 23.** Neural network predictions for the *v*-vertical rigid displacement using the machine learning scheme with an error. The top row shows the stress components $(\tau_{xx}, \tau_{yy}, \tau_{xy})$, and the bottom row shows the displacement components $(u_x, u_y)$.

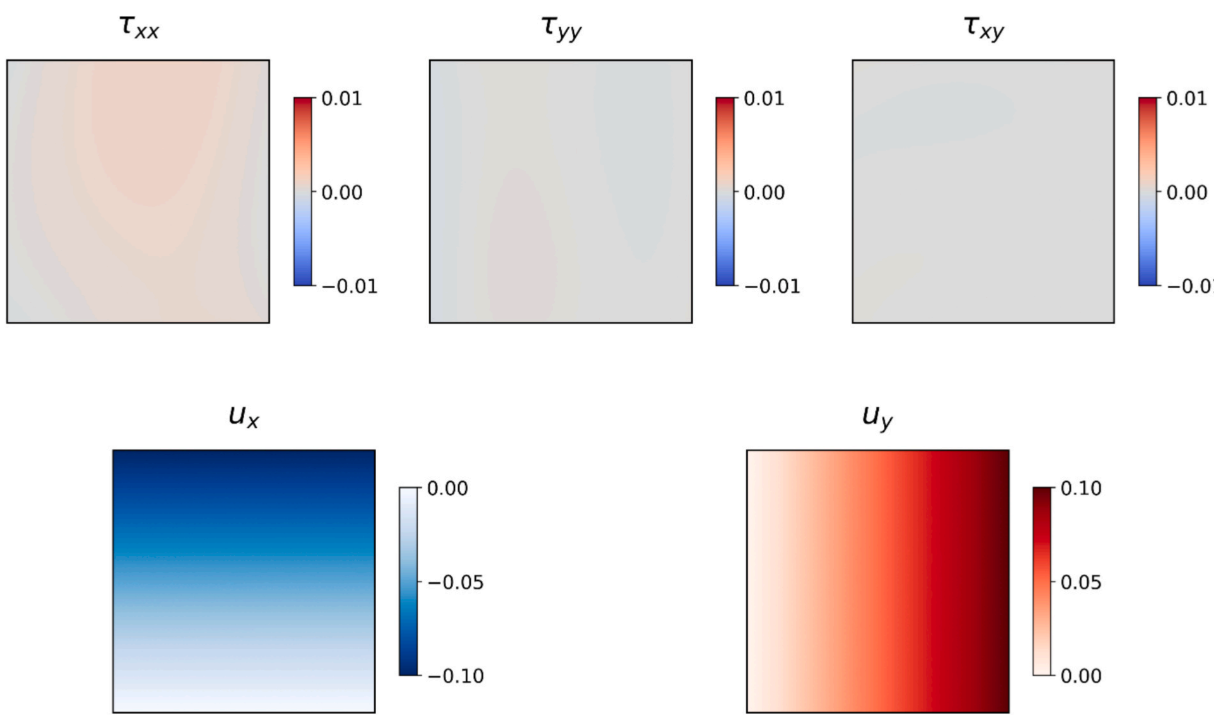


**Fig. 24.** Neural network predictions for the rigid rotation about the z-axis using the machine learning scheme with an error. The top row shows the stress components $(\tau_{xx}, \tau_{yy}, \tau_{xy})$, and the bottom row shows the displacement components $(u_x, u_y)$.

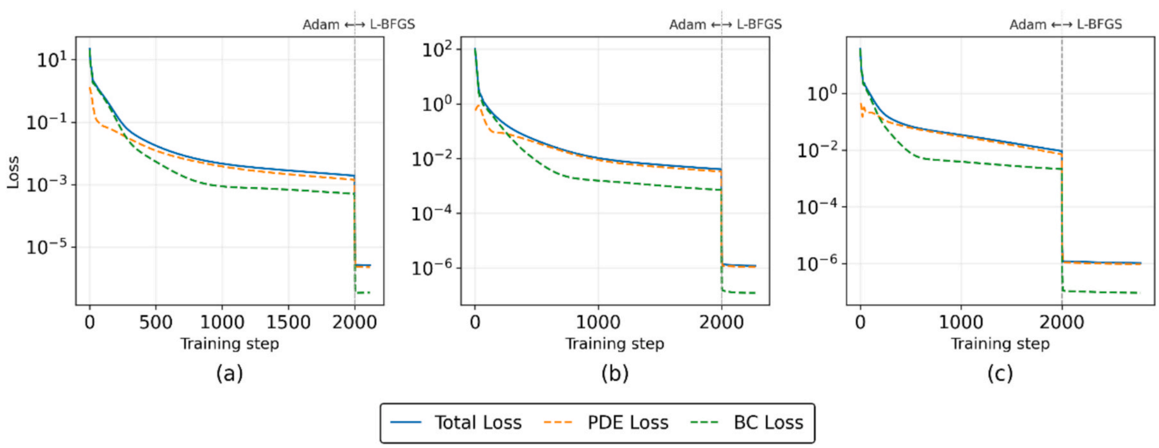


**Fig. 25.** Convergence histories for the test for constant strains/ stresses using the machine learning scheme with an error. The total, PDE, and boundary condition losses are shown, and the dashed line marks the switch from Adam to L-BFGS: (a) uniform extension in the *x*-direction, (b) uniform extension in the *y*-direction, and (c) uniform shear.

[38]. Training used a two-stage optimization strategy (Fig. 7). Adam ([39]) was first applied for 2000 epochs with a learning rate of $1.0 \times 10^{-4}$, followed by L-BFGS ([40]) fine-tuning with strong Wolfe line search for up to 2000 outer steps (max_iter = 200, history_size = 50, tolerance_grad = $1.0\times10^{-8}$, and tolerance_change = $1.0 \times 10^{-12}$). Early stopping was applied during the L-BFGS stage with a patience of 5 steps and a relative improvement threshold of $1.0\times10^{-5}$, and training was terminated early if the total loss dropped below $1.0 \times 10^{-7}$. All experiments were conducted in double precision with fixed random seeds for reproducibility.

We performed all patch tests using plane strain conditions, but of course plane stress conditions could also be used. If the tests are passed for either condition, they are also passed for the other case.

The patch tests for the rigid body modes shown in Fig. 3 were carried out on a square domain with side length $L = 10$ mm, Young's modulus $E = 55$ MPa, Poisson's ratio $\nu = 0.3$, spring stiffness $k = 5.5$N/mm, and applied load $Q = 0.55$N. In the implementation, the action of the spring $k$ and the load $Q$ were distributed over the distance 0.1 mm in order to avoid a numerical instability. The same applicable data was used for the constant strain/stress tests shown in Fig. 4 with the applied traction $q = 0.55$ MPa. In Figs. 6–10 and Figs. 11–15, the results of the rigid body mode tests are shown, including the sampled interior collocation and boundary condition points (Figs. 6 and 11), and the training convergence histories (Figs. 7 and 12).

As shown in the figures, the patch tests are passed when using the proposed machine learning procedure in Fig. 5.

As seen in Figs. 7 and 12, good convergence was reached using the machine learning scheme, which rules out that spurious zero energy modes are present.

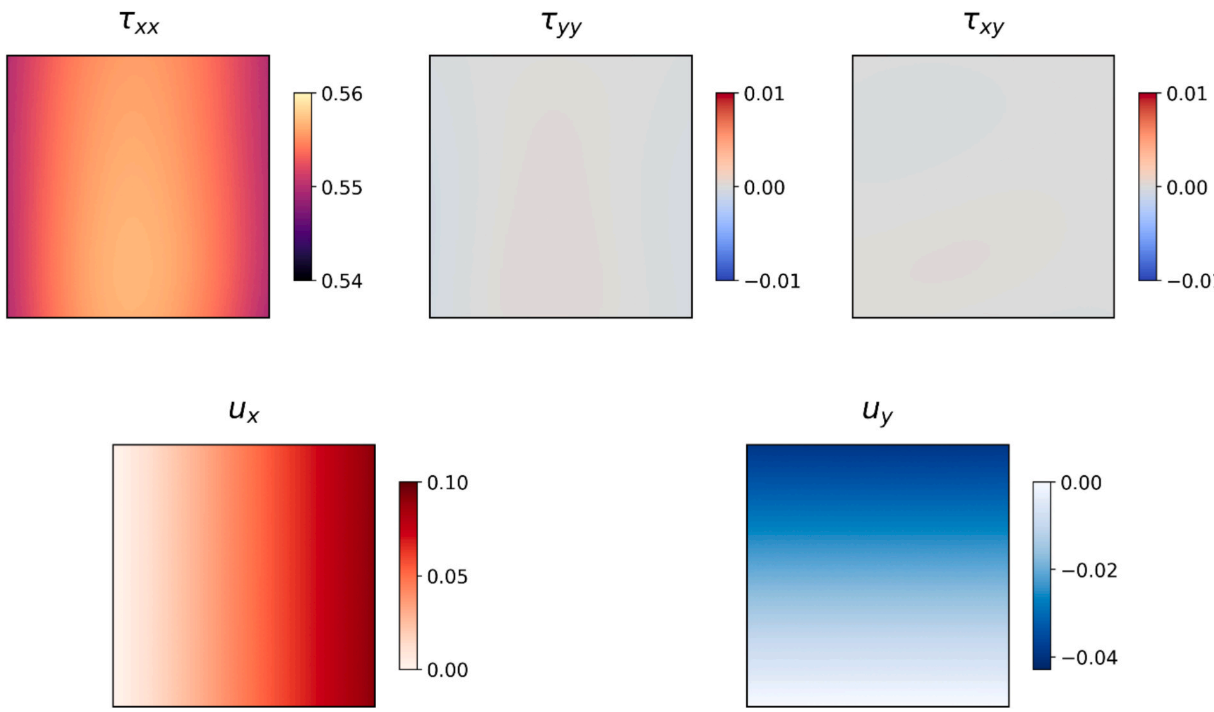


**Fig. 26.** Predicted stress fields and deformed configurations for the x-direction test of constant stresses using the machine learning scheme with an error. The stress components $(\tau_{xx}, \tau_{yy}, \tau_{xy})$ are shown in the top row, and the displacement components $(u_x, u_y)$ are shown in the bottom row.

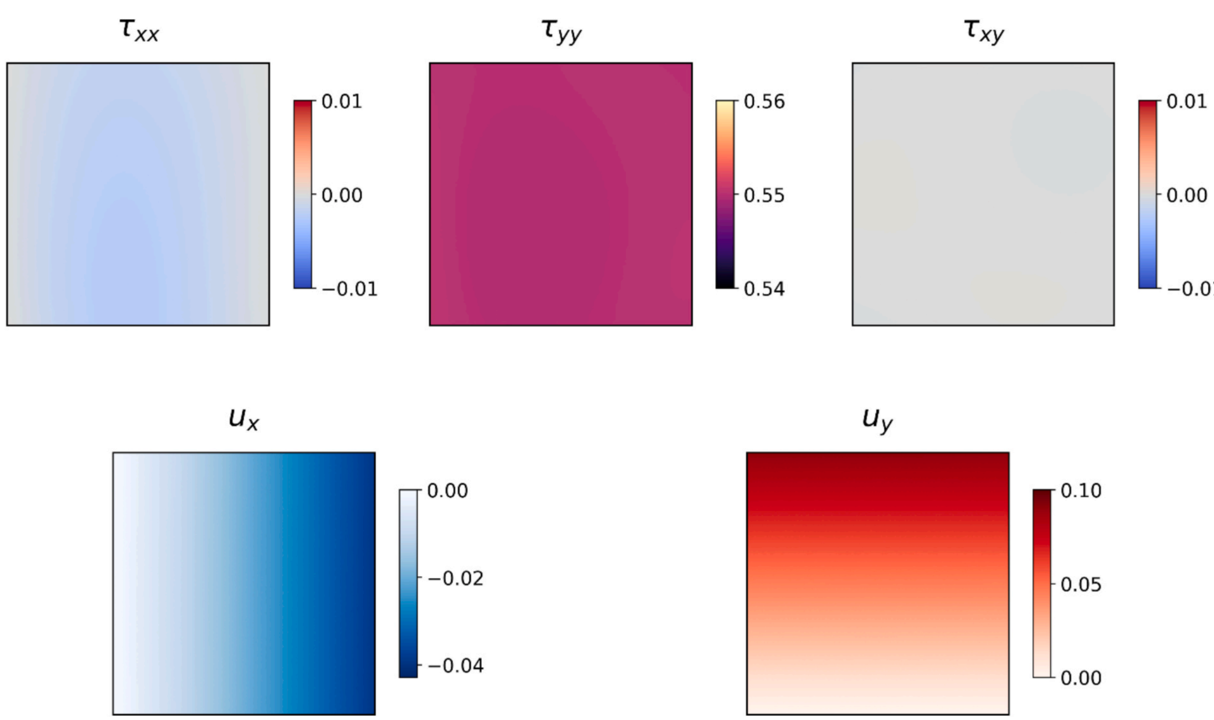


**Fig. 27.** Predicted stress fields and deformed configurations for the y-direction test of constant stresses using the machine learning scheme with an error. The stress components $(\tau_{xx}, \tau_{yy}, \tau_{xy})$ are shown in the top row, and the displacement components $(u_x, u_y)$ are shown in the bottom row.

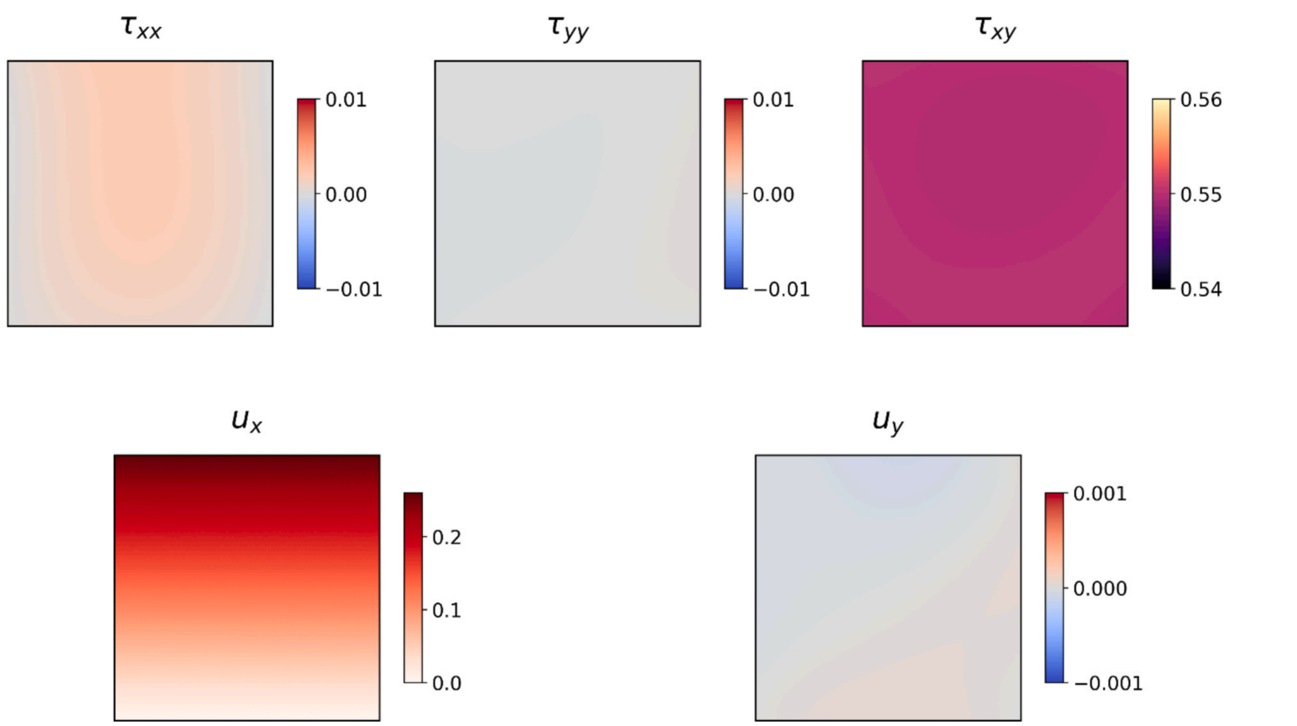


**Fig. 28.** Predicted stress fields and deformed configurations for the shear test using the machine learning scheme with an error. The stress components $(\tau_{xx}, \tau_{yy}, \tau_{xy})$ are shown in the top row, and the displacement components $(u_x, u_y)$ are shown in the bottom row.

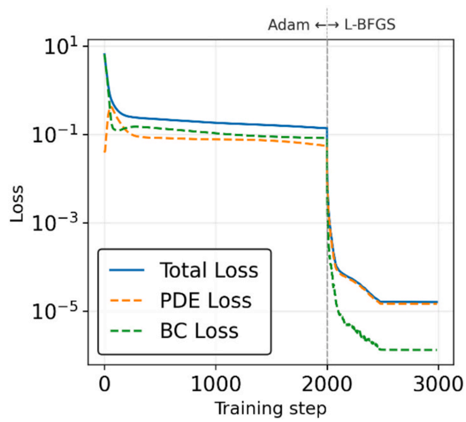


**Fig. 29.** Convergence data for the analysis of the sheet with the hole with the error in the ML calculations.

## 3. Analysis of a sheet with a hole

We next perform the analysis of a sheet with a hole. Fig. 16 describes the problem giving the geometry and material data. First, we consider prescribed end-displacements as the loading and then we solve the problem with prescribed end-tractions.

The objective is to show that if the patch tests are passed, the analysis results obtained are reasonable in both analysis cases, but of course may not be sufficiently accurate because, for example, the discretization used may not be sufficiently fine.

However, if there is a defect in the theoretical formulation of the NN or an error in the implementation, the patch tests are not passed but the results in *"a"* structural analysis may look acceptable whereas in *"another"* analysis the results are clearly not acceptable. The proposed machine learning procedure is then not reliable – and the use of the patch tests reveals this deficiency.

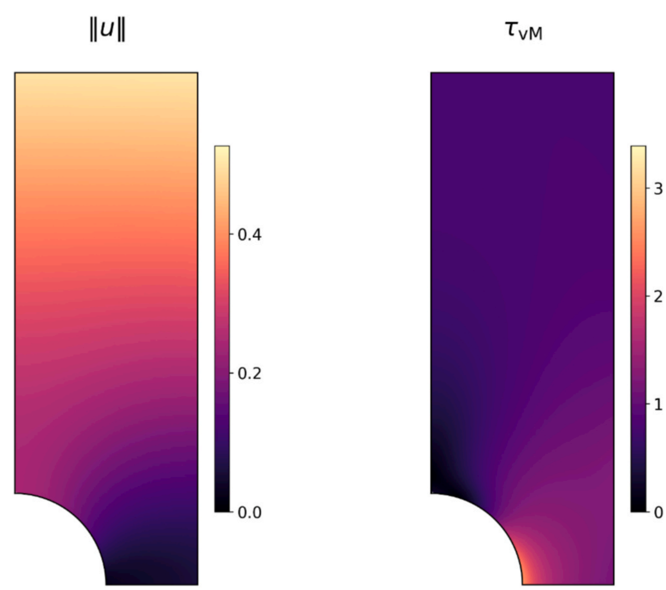


**Fig. 30.** Predicted displacement magnitude $\|u\|$ and von Mises stress $(\tau_{vM})$ for the quarter plate with a circular hole with the error in the machine learning procedure.

### 3.1. *The sheet subjected to prescribed end-displacements*

In this case of prescribed end-displacements, we can use symmetry and consider only one quarter of the sheet, see Fig. 17. As expected, good results with the machine learning procedure of Fig. 5 are obtained, see Figs. 18–20, since the patch tests have been passed, see Figs. 6–15.

As a next step, we intentionally introduce an artificial bias into the loss weighting to mimic a potential programming error. Since the y-directional equilibrium residual is most important in this analysis given that the plate is loaded in the y-direction, we introduce the error in the x-

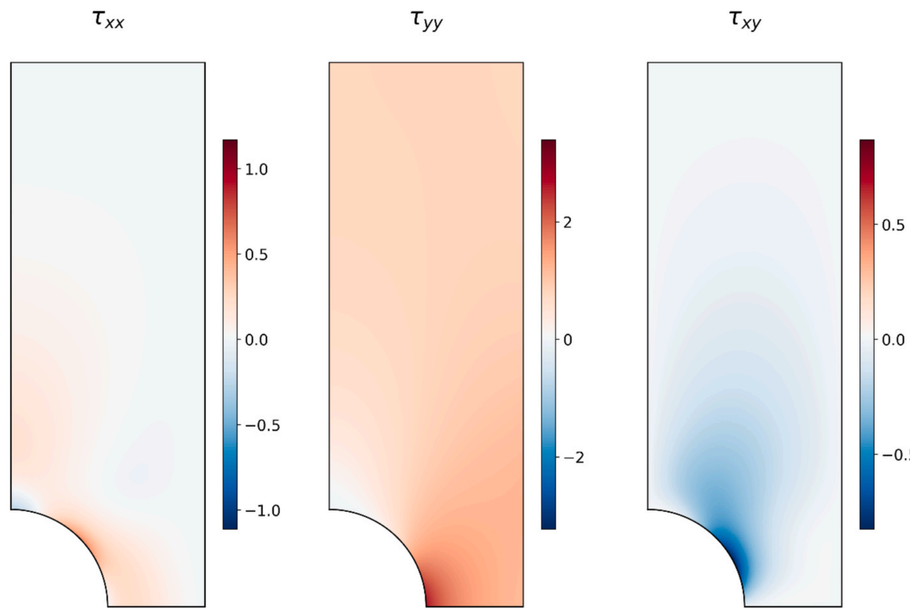


**Fig. 31.** Predicted stress components $(\tau_{xx}, \tau_{yy}, \tau_{xy})$ for the quarter plate with a circular hole with the error in the machine learning procedure.

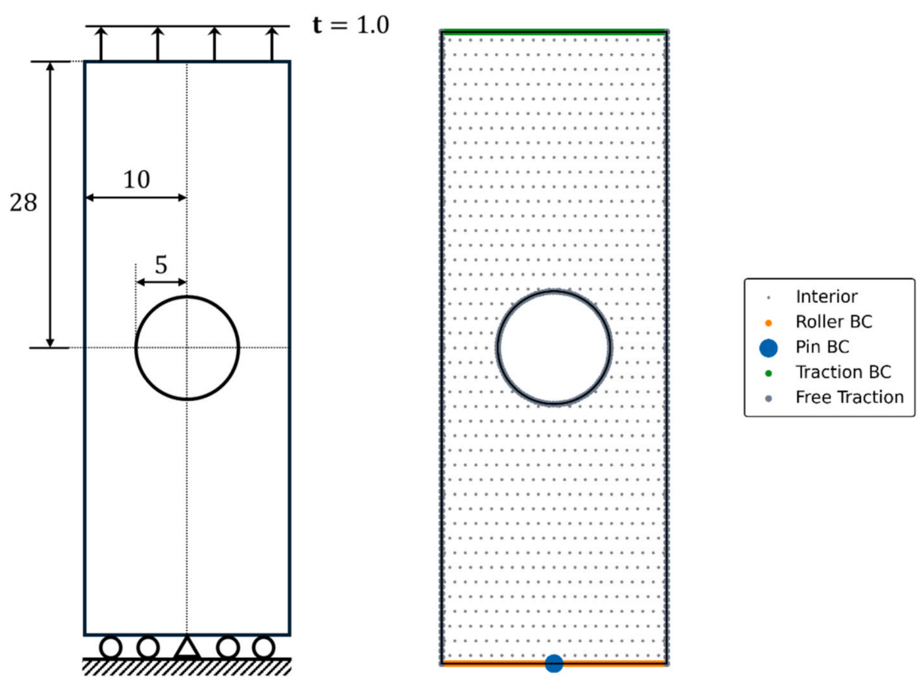


**Fig. 32.** The model of the geometry, applied tractions and boundary conditions used for the solution of the complete sheet (left) and the distribution of the interior and boundary collocation points (right).

directional equilibrium residual by reducing it by the factor $10^{-4}$. This allows us to examine whether the neural network may produce visually plausible yet physically inconsistent solutions, a phenomenon we refer to as "hallucination".

Figs. 21–28 show that the patch tests are not passed in that case, although convergence has been reached in the machine learning solution.

Then solving the problem of Fig. 17 with the solution procedure not passing the patch tests, we see that convergence is reached, as shown in Fig. 29, and that the stress predictions obtained, see Figs. 30 and 31, are somewhat like the calculated results in Figs. 19 and 20, (but some differences can be seen). We might expect this result because the model is highly constrained.

Based on these results, if only the analysis of this sheet is performed with the machine learning procedure containing the error, the solution scheme might be erroneously looked at as a scheme to use in general.

### 3.2. The sheet subjected to prescribed end tractions

We next solve the problem with tractions applied (instead of displacements) and the sheet subjected only to the typical boundary conditions to prevent rigid body modes. In this case we model the complete sheet, see Fig. 32. Hence the model is not highly constrained (as it is in Section 3.1) and we may expect that the machine learning procedure will show different results with the error than without the error. The convergence data and final results for these analyses are shown in

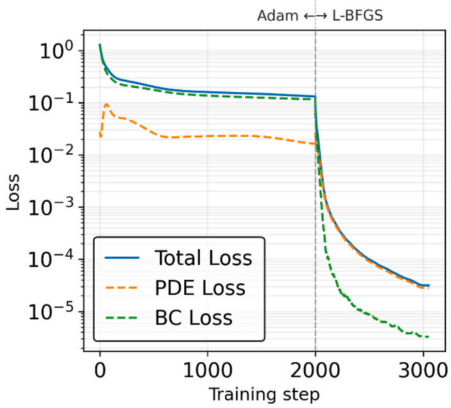


**Fig. 33.** Convergence data in the analysis of the full plate with a circular hole subjected to tractions.

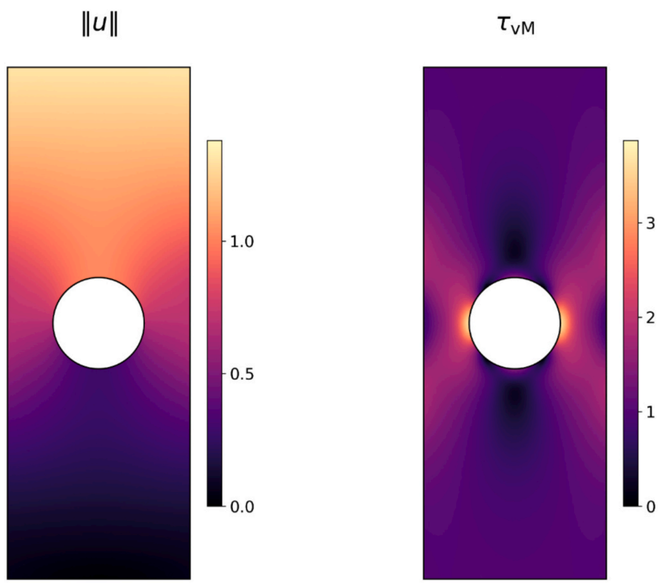


**Fig. 34.** Predicted displacement magnitude $\|u\|$ and von Mises stress ($\tau_{vM}$) for the full plate with a circular hole subjected to tractions.

Figs. 33–36 when the machine learning procedure is used without the error, and in Figs. 37–40 when the error is present.

We see that in this solution, the error in the machine learning calculations results in significant differences, see Figs. 34–36 to compare with Figs. 38–40. We also see that the predicted $\tau_{xx}$ stress is unphysical in Fig. 39 and the deformed shape in Fig. 40 is not quite symmetric. These unacceptable results are clearly a consequence of the patch tests not being passed.

Hence, these solutions show the importance of considering the patch tests using a machine learning scheme and ensuring that these tests are passed before any other analyses are performed. In industry, very

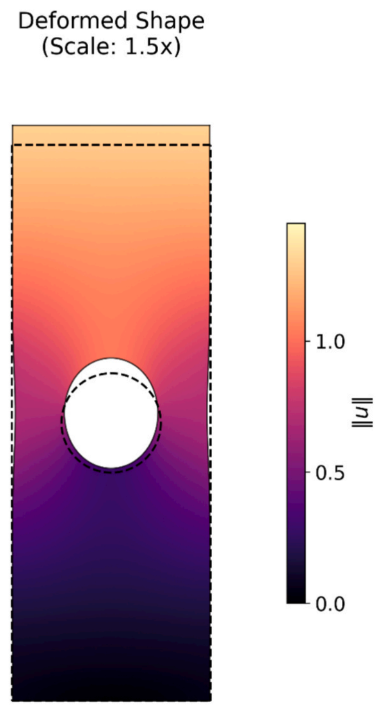


**Fig. 36.** Deformed configuration in the analysis of the sheet with the hole subjected to tractions (Scaled up by factor 1.5).

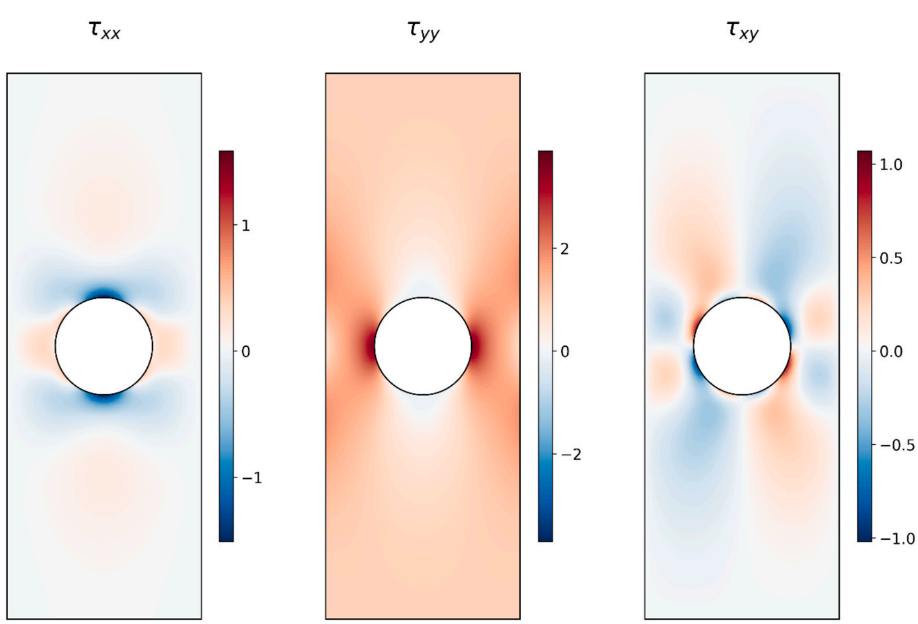


**Fig. 35.** Predicted stress components $(\tau_{xx}, \tau_{yy}, \tau_{xy})$ for the full plate with a circular hole subjected to tractions.

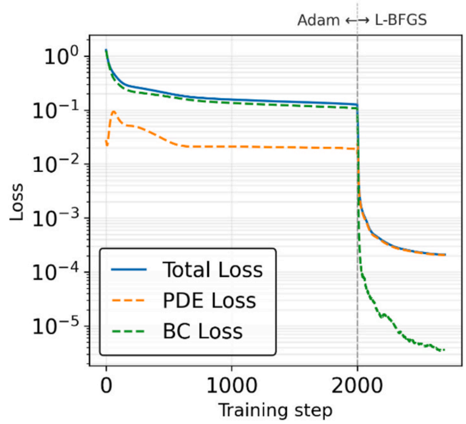


**Fig. 37.** Convergence data for the analysis of the sheet with the hole subjected to tractions with the error in the machine learning calculations.

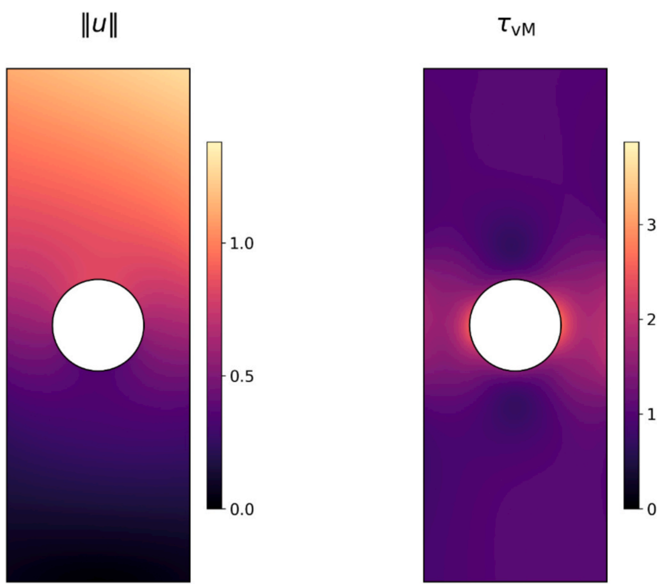


**Fig. 38.** Predicted displacement magnitude $\|u\|$ and von Mises stress ($\tau_{vM}$) for the full plate with a circular hole with the error in the machine learning calculations.

complex geometries subjected to various loading and boundary conditions need to be solved and whether a solution is correct or not may be difficult to discern. Therefore, only reliable solution techniques can be employed.

## 4. Concluding remarks

Our objective in this paper was to focus on the patch tests which should be solved and passed in the testing of machine learning procedures designed to perform displacement, velocity, temperature or stress analyses. If these are not passed, the proposed machine learning

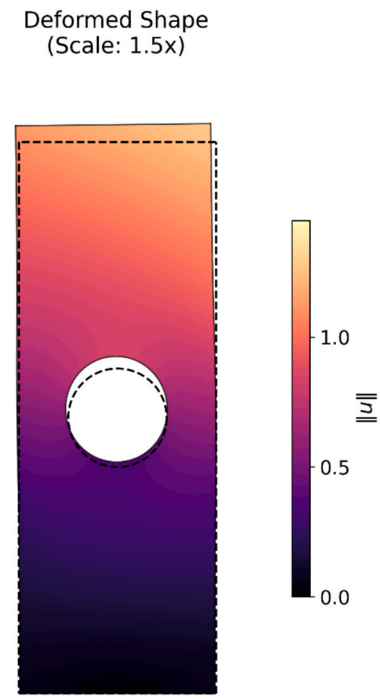


**Fig. 40.** Deformed configuration in the analysis of the sheet with the hole subjected to tractions with the error in the machine learning calculations (scaled up by factor 1.5).

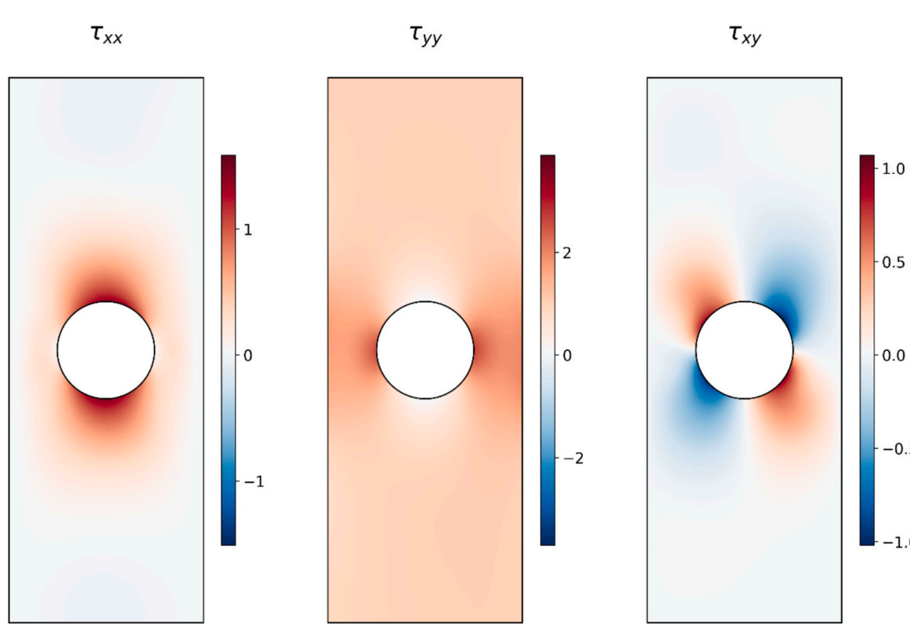


**Fig. 39.** Predicted stress components $(\tau_{xx}, \tau_{yy}, \tau_{xy})$ for the full plate with a circular hole with the error in the machine learning calculations.

scheme is not reliable for general use.

We first discussed the patch tests used in traditional finite element developments, then described the models we employ to perform the patch tests using a machine learning scheme when focused on analysis and finally exemplified that if the patch tests are not passed erroneous results can be obtained although these may look acceptable in some analyses.

Of course, the patch tests are only one important means to identify the correctness and reliability of a machine learning analysis scheme. However, passing the patch tests is surely a necessary condition for establishing confidence in a proposed machine learning scheme. There are also other tests, like the inf-sup tests and the rate of convergence tests [35,36], which are also important to perform in order to understand the full capabilities of a machine learning procedure.

Apart from presenting and illustrating the usefulness of the patch tests, a major point we want to also convey is that more focus needs to, in general, be directed towards the testing of a proposed machine learning procedure. The aim should be to identify whether the procedure is reliable and efficient for general use in engineering and scientific studies. It is not sufficient to rely on the principle of virtual work (as we did in this paper) or a universal approximation theorem. These mathematical frameworks are very important and fundamental, but when designing a solution scheme operating on these frameworks, we need to show reliability of the scheme, which means verification and validation (V&V), and have strong V&V tests. Significant further research should be directed towards this aim in various areas of machine learning.

**Funding statement**

There was no funding to be acknowledged for the research.

**CRediT authorship contribution statement**

**Klaus-Jürgen Bathe:** Conceptualization, Investigation, Methodology, Supervision, Writing – original draft, Writing – review & editing. **Sanghee Lee:** Conceptualization, Investigation, Software, Validation, Writing – review & editing.

**Declaration of competing interest**

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Appendix

*Appendix A. PINN model and loss function*

Assuming a two-dimensional elasticity problem, we approximate the displacement and stress fields through a neural network $\mathscr{N}_\theta$:

$$\left(\widehat{u}_x, \widehat{u}_y, \widehat{\tau}_{xx}, \widehat{\tau}_{yy}, \widehat{\tau}_{xy}\right)(x,y) = \mathscr{N}_\theta(x,y), (x,y) \in \Omega \tag{A.1}$$

The network adopts a mixed formulation in which both displacement and stress components are predicted. The residuals are derived from the stress–strain constitutive relations and the strong-form equilibrium equations in the domain $\Omega$.

Let $\widehat{\mathbf{u}} = \left(\widehat{u}_x, \widehat{u}_y\right)$ and $\widehat{\boldsymbol{\tau}} \in \mathbb{R}^{2\times 2}$ with components $\left(\widehat{\tau}_{xx}, \widehat{\tau}_{yy}, \widehat{\tau}_{xy}\right)$ and assume small strain and small displacement conditions. The strains and stresses are computed as

$$\widehat{\boldsymbol{\varepsilon}}(\widehat{\mathbf{u}}) = \frac{1}{2}\left(\nabla\widehat{\mathbf{u}} + \nabla\widehat{\mathbf{u}}^\top\right), \boldsymbol{\tau}^\star(\widehat{\mathbf{u}}) = \lambda\,\mathrm{tr}(\widehat{\boldsymbol{\varepsilon}})\boldsymbol{I} + 2\mu\,\widehat{\boldsymbol{\varepsilon}}, \tag{A.2}$$

where $\lambda, \mu$ are Lame's constants.

The residuals derived from the constitutive relations and the equilibrium equation $\nabla\cdot\boldsymbol{\tau} = 0$ are

$$\mathbf{r}_{\text{cons}}(\mathbf{x}) := \frac{\widehat{\boldsymbol{\tau}}(\mathbf{x}) - \boldsymbol{\tau}^\star(\widehat{\mathbf{u}}(\mathbf{x}))}{s_0},\ \mathbf{r}_{\text{eq}}(\mathbf{x}) := \frac{L}{s_0}\ \nabla\cdot\widehat{\boldsymbol{\tau}}(\mathbf{x}), \tag{A.3}$$

where $L$, $s_0$ are scaling factors that make the residuals unitless.

The loss in satisfying the differential equilibrium equations in the domain is defined as

$$\mathscr{L}_{\text{pde}} = \mathbb{E}_{\mathbf{x}\sim\mathscr{U}(\Omega)}\left[\sum_{\alpha\in\{xx,yy,xy\}}\left|r_{\text{cons},\alpha}(\mathbf{x})\right|^2 + \sum_{\beta\in\{x,y\}}\left|r_{\text{eq},\,\beta}(\mathbf{x})\right|^2\right], \tag{A.4}$$

where $\mathbf{r}_{\text{cons}}(\mathbf{x}) = \begin{bmatrix} r_{cons,xx}(\mathbf{x}) & r_{cons,xy}(\mathbf{x}) \\ r_{cons,yx}(\mathbf{x}) & r_{cons,yy}(\mathbf{x}) \end{bmatrix}$, and $\mathbf{r}_{\text{eq}}(\mathbf{x}) = \begin{bmatrix} r_{\text{eq},\,x}(\mathbf{x}) \\ r_{\text{eq},\,y}(\mathbf{x}) \end{bmatrix}$.

and we have

$$\mathscr{L}_{\text{pde}} \approx \frac{1}{|\mathscr{X}_c|}\sum_{\mathbf{x}_i\in\mathscr{X}_c}\left(\sum_{\alpha\in\{xx,yy,xy\}}\left|r_{\text{cons},\alpha}(\mathbf{x}_i)\right|^2 + \sum_{\beta\in\{x,y\}}\left|r_{\text{eq},\,\beta}(\mathbf{x}_i)\right|^2\right), \mathscr{X}_c\subset\Omega. \tag{A.5}$$

Here, $\mathscr{X}_c = \{\mathbf{x}_i\}_{i=1}^{N_c}$ denotes the set of interior collocation points sampled in $\Omega$ that is used to evaluate the PDE residuals.

Likewise, for the boundary conditions on $\Gamma$,

$$\Gamma = \partial\Omega = \bigcup_{m=1}^{M}\Gamma_m, (\Gamma_m \cap \Gamma_{m'} = \varnothing \text{ if } m \neq m') \tag{A.6}$$

Let $\mathscr{B}_m(\hat{\mathbf{u}}(\mathbf{x}), \hat{\boldsymbol{\sigma}}(\mathbf{x}); \mathbf{x}) = 0$ be the equations to be satisfied on $\Gamma_m$. It may be scalar- or vector-valued; in the latter case, the mean square error is computed by summing the squared residuals of its components. For each boundary segment $\Gamma_\alpha$, $X_\alpha$ denotes the corresponding set of sampled collocation points, and $\mathbf{n}$ denotes the outward unit normal vector on $\Gamma_\alpha$. The loss for boundary conditions is defined as

$$\mathscr{L}_{bc} = \sum_{m=1}^{M} \left\| \frac{\mathscr{B}_m(\hat{\mathbf{u}}, \hat{\boldsymbol{\tau}}; \cdot)}{c_m} \right\|^2_{\mathscr{X}_m} \tag{A.7}$$

where $c_m \in \{L, u_0, s_0\}$ is a corresponding scaling factor that renders $\frac{\mathscr{B}_m(\hat{\mathbf{u}}, \hat{\boldsymbol{\tau}}; \cdot)}{c_m}$ to be unitless.

The weighting coefficients $\lambda_{\text{pde}}$ and $\lambda_{\text{BC}}$ are applied to each component of $\mathscr{L}_{\text{pde}}$ and $\mathscr{L}_{BC}$ to balance their contributions during training. In this study, all point boundary conditions are assigned a weight of 10.0.

## Data availability

Data will be made available on request.

## References


[1] Bathe KJ. Finite Element Procedures. 1st Edition, Prentice-Hall, 1996; 2nd Edition, Amazon.com, 2014.
[2] Bathe KJ. Finite Element Procedures – *En Plus*. In press, Springer.
[3] Zienkiewicz OC, Taylor RL. The Finite Element Method. Butterworth-Heinemann; 2000.
[4] Chapelle D, Bathe KJ. The Finite Element Analysis of Shells – Fundamentals. 2nd ed. Springer; 2011.
[5] Irons BM. Engineering Applications of Numerical Integration in Stiffness Method. AIAA J 1966;4:2035–7.
[6] Bathe KJ. To Enrich Life – My journey & the finite element method. Amazon.com 2025.
[7] Noh G, Bathe KJ. Imposing displacements in implicit direct time integration & a patch test. Adv Eng Softw 2023;175:103286.
[8] Raissi M, Perdikaris P, Karniadakis GE. Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J Comput Phys 2019;378:686–707.
[9] Guo H, Zhuang X, Fu X, Zhu Y, Rabczuk T. Physics-informed deep learning for three-dimensional transient heat transfer analysis of functionally graded materials. Comput Mech 2023;72:513–24.
[10] Caforio F, Regazzoni F, Pagani S, Karabelas E, Augustin C, Haase G, Plank G, Quarteroni A. Physics-informed neural network estimation of material properties in soft tissue nonlinear biomechanical models. Comput Mech 2025;75:487–513.
[11] Linghu J, Dong H, Nie Y, Cui J. Higher-order multi-scale deep Ritz method (HOMS-DRM) and its convergence analysis for solving thermal transfer problems of Composite materials. Comput Mech 2025;75:71–95.
[12] Yan CA, Vescovini R, Dozio L. A framework based on physics-informed neural networks and extreme learning for the analysis of composite structures. Comput Struct 2022;265:106761.
[13] Zhang M, Zhang J, Ding J, Li B. An adaptive multi-task learning method for response prediction and optimal sensor placement. Comput Struct 2025;315: 107779.
[14] Li Q, Wang Z, Li L, Hao H, Chen W, Shao Y. Machine learning prediction of structural dynamic responses using graph neural networks. Comput Struct 2023; 289:107188.
[15] Topping BHV, Khan AI, Bahreininejad A. Parallel training of neural networks for finite element mesh decomposition. Comput Struct 1997;63:693–707.
[16] Bai Z, Song S. Physics-informed neural network for first-passage reliability assessment of structural dynamic systems. Comput Struct 2023;289:107189.
[17] Ramesh Babu J, Gopalakrishnan S. Synergistic approach: peridynamics and machine learning regression for efficient pitting corrosion simulation. Comput Struct 2024;305:107588.
[18] Cao H, Li M, Nie L, Xie Y, Kong F. Vertex-based graph neural network classification model considering structural topological features for structural optimization. Comput Struct 2024;305:107542.
[19] Kalina KA, Linden L, Brummund J, Metsch P, Kästner M. Automated constitutive modeling of isotropic hyperelasticity based on artificial neural networks. Comput Mech 2022;69:213–32.
[20] Peirlinck M, Linka K, Hurtado JA, Holzapfel GA, Kuhl E. Democratizing biomedical simulation through automated model discovery and a universal material subroutine. Comput Mech 2025;75:1703–23.
[21] Zhang L, Cheng L, Li H, Gao J, Yu C, Domel R, et al. Hierarchical deep-learning neural networks: finite elements and beyond. Comput Mech 2021;67:207–30.
[22] Markou G, Bakas NP, Chatzichristofis SA, Papadrakakis M. A general framework of high-performance machine learning algorithms: application in structural mechanics. Comput Mech 2024;73:705–29.
[23] Li G, Luo R, Yu D-H. A framework for developing a machine learning-based finite element model for structural analysis. Comput Struct 2025;307:107617.
[24] Jayasinghe SC, Mahmoodian M, Alavi A, Sidiq A, Shahrivar F, Sun Z, et al. A review on the applications of artificial neural network techniques for accelerating finite element analysis in the civil engineering domain. Comput Struct 2025;310:107698.
[25] Jokar M, Semperlotti F. Finite element network analysis: a machine learning based computational framework for the simulation of physical systems. Comput Struct 2021;247:106484.
[26] Rafiq MY, Bugmann G, Easterbrook DJ. Neural network design for engineering applications. Comput Struct 2001;79:1541–52.
[27] Bartel T, Harnisch M, Menzel A, Schweizer B. Aspects of accuracy and uniqueness of solutions in data-driven mechanics. PAMM 2023;22:e202200206.
[28] Sahin T, Wolff D, Popp A. Physics-Informed Neural Networks for Solving Contact Problems in Three Dimensions; 2024. https://doi.org/10.48550/arXiv.2412.09022 .
[29] Jung J, Yoon K, Lee P-S. Deep learned finite elements. Comput Methods Appl Mech Eng 2020;372:113401.
[30] Jung J, Jun H, Lee P-S. Self-updated four-node finite element using deep learning. Comput Mech 2022;69:23–44.
[31] Lee Sh, Bathe KJ. A constitutive neural network for incompressible hyperelastic materials. Mach Learn Comput Sci Eng 2025;1:31.
[32] Wang Y, Hao Z, Eshaghi MS, Anitescu C, Zhuang X, Rabczuk T, Liu Y. Pretrain Finite Element Method: A Pretraining and Warm-start Framework for PDEs via Physics-Informed Neural Operators; 2026. https://doi.org/10.48550/arXiv.2601.03086.
[33] Bathe KJ, Zhang L. The finite element method with overlapping elements – a new paradigm for CAD driven simulations. Comput Struct 2017;182:526–39.
[34] Lee S, Bathe KJ. An enhancement of overlapping finite elements. Comput Struct 2022;260:106704.
[35] Ko Y, Bathe KJ. Inf-sup testing of some three-dimensional low-order finite elements for the analysis of solids. Comput Struct 2018;209:1–13.
[36] Nicomedes WL, Bathe KJ. An effective overlapping finite element for three-dimensional incompressible linear elasticity. Comp Meth Appl. Mech. Eng. 2026; 457:118613.
[37] Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, et al. PyTorch: an imperative style, high-performance deep learning library. Adv Neural Inf Proces Syst 2019;32:8024–35.
[38] MIT Office of Research Computing and Data. ORCD Engaging System. ORCD Docs, 2025.
[39] Kingma DP, Adam JB. A method for stochastic optimization. Proceedings of the 3rd International Conference on Learning Representations. 2015.
[40] Liu DC, Nocedal J. On the limited memory BFGS method for large scale optimization. Math Program 1989;45(1–3):503–28.