Solution of structural mechanic's problems by machine learning

Himanshu Gaur*

Institute of Structural Mechanics,
Bauhaus-Universität Weimar,
Marienstrasse 15, D-99423 Weimar, Germany
and
Middle East College,
Knowledge Oasis Muscat,
P.B. No. 79, Al Rusayl, Muscat, 124, Sultanate of Oman
Email: himanshu@mec.edu.om
*Corresponding author

Basim Khidhir and Ram Kishore Manchiryal

Middle East College,
Knowledge Oasis Muscat,
P.B. No. 79, Al Rusayl, Muscat, 124, Sultanate of Oman
Email: basim@mec.edu.om
Email: ramkishore@mec.edu.om

Abstract: This article proposes an analysis procedure of structural mechanics' problem as integral formulation. The methodology is novel which can be suitably applied for finding the solution of engineering problems with required accuracy either it is linear or nonlinear range (plastic range) of the material behaviour. This methodology, which was proposed as a stress-based analysis procedure, exploits the unfolded part of the structural analysis problems which were not so easy to solve such as geometric and material nonlinearity together with simple integration technique (Gaur and Srivastav, 2021). In fracture mechanics, it has already unfolded the misery of physically exploiting the plastic behaviour of structures before the start of the crack for elastic materials (Gaur et al., 2021). The formulation is an integral formulation rather than a differential formulation in which whole stress-strain behaviour is utilised in the analysis procedure by using a neural network as a regression tool. In this article, the one-dimensional problem of uniaxial bar, beam bending problem and plane strain axis-symmetric problem of a cylinder subjected to internal pressure is solved. The results are compared with the existing differential formulation or linear theory.

Keywords: computational methods; continuum mechanics; machine learning; plastic analysis; elastic materials.

Reference to this paper should be made as follows: Gaur, H., Khidhir, B. and Manchiryal, R.K. (2022) 'Solution of structural mechanic's problems by machine learning', *Int. J. Hydromechatronics*, Vol. 5, No. 1, pp.22–43.

Biographical notes: Himanshu Gaur is currently pursuing his PhD from Institute of Structural Mechanics, Bauhaus University Weimar, Germany. His current research interests are in computational methods, fraction mechanics, optimisation, solution of partial differential equations by machine learning, etc.

Basim Khidhir is currently working as a Professor and Head of Department of Mechanical Engineering, Middle East College, Muscat, Sultanate of Oman. His current research interests are in aerospace engineering, industrial engineering, manufacturing engineering, etc.

Ram Kishore Manchiryal is currently working as a Professor and Head of Department of Civil Engineering, Middle East College, Muscat, Sultanate of Oman. He completed his PhD from Clarkson University in 2010. His current research interests are in material characterisation using dielectric properties, durability and sustainable materials, applications of artificial neural networks, etc.

1 Introduction

Almost all problems in physics and engineering are derived/dealt based on the differential formulation in which each problem, based on the physics, has its differential formulation. In recent years, considerable development is seen in finding the solution of differential equations by machine learning (Lee and Kang, 1990; Meade and Fernadez, 1994a, 1994b). Applications of machine learning, that are of interest in computational mechanics is based on the solution of differential equations by collocation method and was first developed by Lagaris et al. (1998). Recently, its applications are further extended to solve many other problems in structural mechanics (Raissi et al., 2019; Samaniego et al., 2020; Goswami et al., 2020a, 2020b; Guo et al., 2019; Anitescu et al., 2019; Zhuang et al., 2021).

Machine learning, the name itself, demonstrates that the computer learns from the available data. The artificial neural networks train the model, for the given data and, are known to make very accurate predictions. It has shown wide applications in different fields such as voice and face recognition, business and commercial applications, etc. (Krizhevsky et al., 2012; Lake et al., 2015; Alipanahi et al., 2015). This methodology utilises regression, the ability of the neural network to predict accurate numerical value based on the available data. The importance of the methodology is observed when it solves linear as well nonlinear analysis of the structures with relative ease.

In this article, we propose the solution of physical problems for which the physical response of the system is nonlinear such as problems of structural mechanics. Solution based on these problems covers a wide range of engineering applications in mechanical, civil and infrastructure engineering. In these nonlinear problems, nonlinearity in the physical response comes either from material nonlinearity or geometric nonlinearity (Rabczuk et al., 2008; Areias et al., 2018, 2016; Ghorashi et al., 2015; Rabczuk et al., 2005; Rabczuk and Belytschko, 2007; Talebi et al., 2014). For these kinds of problems, we propose integral formulation rather than differential formulation. For the physical nonlinear problems, the method has been proved as the most efficient approach of analysis (Gaur and Srivastav, 2021).

So far, in formulating the structural problems, all problems are modelled as displacement-based (Nguyen-Thanh et al., 2017; Nguyen-Thanh et al., 2011a, 2011b, 2015; Nguyen Huy et al., 2019; Vu-Bac et al., 2018). In these problems, the material property is inputted as elastic modulus, which is just the ratio of stress over strain. This formulation favours the displacement-based or linear analysis procedure. Hence this structural analysis procedure is basically differential formulation. This type of formulation leads difficulty in finding the solution of problems when material behaviour is nonlinear or enters in the plastic range. To date, this remained unfolded part of the research in which no direct solution is applicable. The difficulty can be observed when complicated, lengthy and time-consuming methods are available for finding the solution of fracture mechanics problems of elastic materials (Kumar, 2014).

For solving the structural mechanic's problems, we utilise the capability/property of the neural network of properly mapping the nonlinear data. This is done with the help of more than one layer (hidden layers) and activation functions. The work of neural networks is essentially to optimise weights and biases such that to give accurate output. For different applications different neural networks are being used and they are named differently in Sharkawy (2020). For our application in continuum mechanics, we wish to predict accurate output in our numerical method, so, we have used a fully connected neural network as a regression tool to make accurate numerical predictions. The architecture and functioning of the neural networks for our application can be understood with the help of Figures 6 and 7.

2 From differential formulation towards integral formulation

The current state of the art of structural mechanics is basically differential formulation in which for solving any typical problem, force balance equations, i.e., equations of equilibrium are written which are then solved analytically or by numerical methods such as finite element method, mesh-free methods and isogeometric analysis (IGA), etc. (Hughes, 2000; Huerta et al., 2018; Nguyen et al., 2008; Nguyen-Thanh et al., 2017). It is evident that this formulation has limited capability of solving the problems in nonlinear range of material behaviour.

In this article, we are proposing an integral formulation of analysis for finding solutions of the problems of structural mechanics. With the integral formulation, strain energy stored in the loaded body is directly found. In the proposed approach, first of all, the true stress-strain behaviour of the material is mapped to some third parameter which is called as reference coordinate system (r). In the present study, it is taken to be varied between –1 to +1. This variation of stress and strain with reference coordinate system can be viewed in Figures 11 and 12 for mild steel. The objective of this mapping is to express stress and strain in terms of this third parameter (r), by which strain energy can be evaluated by simple integration. Once strain energy of the system is found, other desired results such as strain, deformation, etc. can be found suitably. The proposed analysis procedure is novel and is perhaps inspired by the derivation of Piolla Kirchhoffs stress from Green-Lagrange strain (energetic conjugates) of finite element analysis (Steigmann, 2002).

3 Methodology – the basic formulation

As explained previously, the proposed approach is the integral formulation. By integral formulation, we mean to evaluate strain energy of the deformed/stressed body when subjected to loading by the following integration.

$$\textit{Strain energy} = \int_V \boldsymbol{\sigma}\ \boldsymbol{\varepsilon}\ dV \tag{1}$$

Here, $\boldsymbol{\sigma}$ and $\boldsymbol{\varepsilon}$ are the stress and strain tensors generated in a three-dimensional structural body in Cartesian coordinates (Timoshenko and Goodier, 1934). Figure 1 show a typical structure loaded with F at the boundary Γ_F. It has fixed boundaries at Γ_o. Under the application of external loads, the total potential ($\prod$) of the structural body can be expressed as follows.

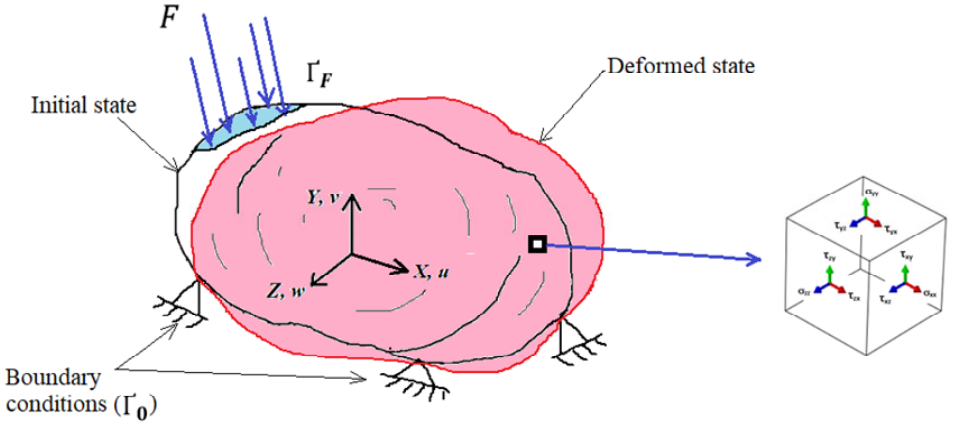
$$\prod = U - W \tag{2}$$

Here, U is strain energy stored in the deformed/stressed body which can be evaluated by equation (1) and W is the work done by the external loads which can be evaluated by the expressions below.

$$W = \int_{\Gamma_F} \boldsymbol{F}\ d\boldsymbol{u} \tag{3}$$

Here, $\boldsymbol{F}$ and $\boldsymbol{u}$ are force and displacement vectors in the three-dimensional coordinate system.

Figure 1 A typical three-dimensional body subjected to loading (see online version for colours)

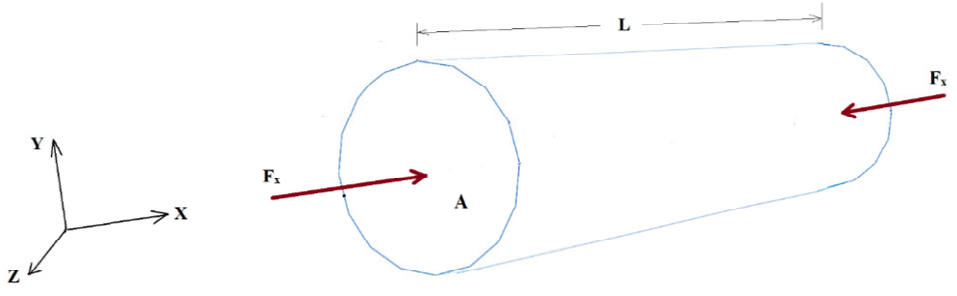


It will be interesting to observe how the integration of equation (1) is performed for finding the solution of any typical structure. For the structure loaded with such a magnitude that it deforms the material in plastic range, such as the bar shown in Figure 13 with the loading condition. In these cases, stress and strain components vary across the volume of the body and hence to evaluate the multiplication, numerical integration is performed in basic Python.

Considering the special case of the uniaxial rod of sectional area A and length L, loaded in the x-direction with the load of F_x as shown in Figure 2. As there will be only axial stress σ_x and strain ε_x for the given loading condition, total potential of the bar from equation (2) can be expressed as,

$$\prod = \int_V \sigma_x \varepsilon_x dV - \int_A F_x du_x \tag{4}$$

Figure 2 A cylindrical bar subjected to axial loading (see online version for colours)



Here u_x refers to the deformation of the bar in x-direction. Total potential of the bar can further be simplified as follows.

$$\prod = A\int_L \sigma_x \varepsilon_x dx - \int_A F_x du_x \tag{5}$$

Applying stationarity condition $\delta\prod = 0$ (Bathe, 2014),

$$0 = A(\sigma_x \varepsilon_x) - F_x \frac{\partial u_x}{\partial x} \tag{6}$$

Since $\dfrac{\partial u_x}{\partial x} - \varepsilon_x$, further simplifying,

$$A\sigma_x = F_x$$

or,

$$\sigma_x = \frac{F_x}{A} \tag{7}$$

This is the force balance equation which is used in the methodology to find reference coordinate system (r), once the stress distribution in the structural body is defined with proper/assumed structural idealisation and modelling assumptions.

3.1 Modelling and structural idealisation

The proposed methodology is regarded as the stress-based analysis procedure as it is required to know the stress distribution within the body of the structure so that strain energy can be calculated (Gaur and Srivastav, 2021). Stress distribution within the loaded

body is determined from the proper structural idealisation and hence the accuracy of the solution depends upon the proper modelling assumptions.

Our assumptions for modelling and structural idealisations are the same as those found in the strength of material theory. For example, Euler-Bernoulli's beam bending theory, torsion theory, etc. Hence, the result obtained from the methodology can suitably be applied and used for engineering applications.

Once the stress distribution within the loaded structural element is determined, strain energy stored within the body can simply be evaluated by the equations as explained in the methodology part.

3.2 Definition of stress and strain

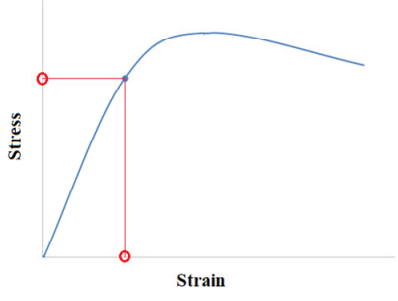
As the proposed integral approach is a complete approach, i.e., the results are valid for linear as well as nonlinear range of the material behaviour. We are using true stress and true strain in order to accommodate the behaviour of large deformation. Therefore, it should be noticed that the results are valid before necking starts in the tension specimen of elastic material used for measuring stress-strain behaviour.

4 Analysis procedure by machine learning as a regression tool

The neural network is used as the nonlinear regression tool in the analysis procedure. For any structural problem, geometric parameters such as area of cross-section, length do not produce nonlinearity in the solution. Nonlinearity essentially comes from the material behaviour. Therefore, in the neural network, the material property is the only parameter that is trained in the neural network.

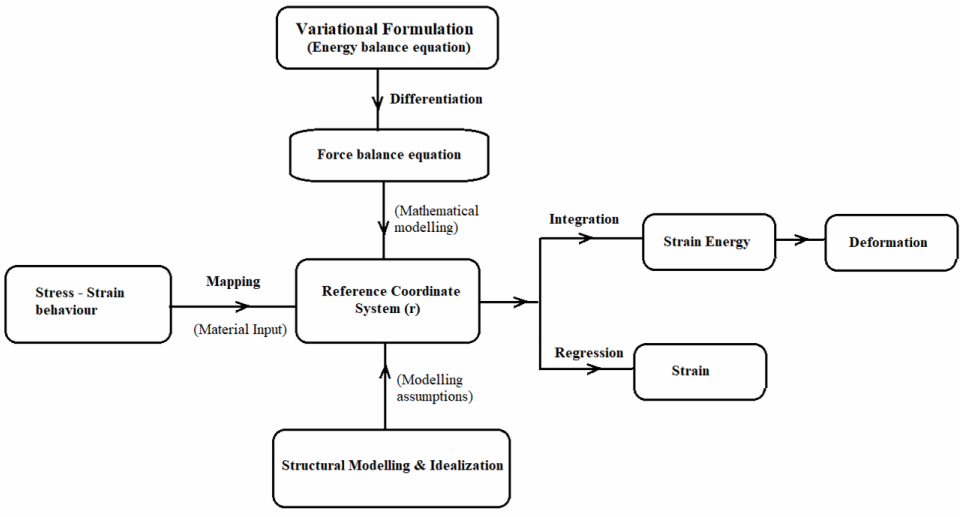
The basic idea behind the analysis procedure is to find reference coordinate system (r), or to say, locating the point on the stress-strain diagram (Figure 3). Force balance equation such as equation (7) is used to find reference coordinate system (r) from the known stresses. Once the reference coordinate is known, strain can also be located by stress-strain graph. Once stress and strain are known, strain energy stored in the structural body can simply be evaluated by integration.

Figure 3 A typical stress-strain behaviour of material (see online version for colours)



The overall analysis procedure of the methodology is summarised with the help of the flowchart shown in Figure 4.

Figure 4 Flowchart of overall analysis procedure employed



4.1 Construction of neural network

Figure 6 shows a fully connected feed-forward neural network that is used in the analysis. Three hidden layers are found to be performing well in comparison to the two, or more than three hidden layers. It is found that changing the number of neurons in the hidden layer did not give any influence on the output. In this study, 64 neurons are used for each hidden layer. Nonlinear mapping is achieved by the ReLU activation function which is used for all hidden layers. For the last hidden layer, which yields the output layer, a linear activation function is used (Figures 6 and 7).

First of all, for each neuron weights are initialised randomly as per standard normal distribution and all biases are set to zero. Python code used for generating a layer can be glanced underneath.

Figure 5 Weights and bias initialisation in a dense layer (see online version for colours)

```
class Dense_layer:
    def __init__(self, n_of_inputs, n_of_neurons):
        self.weights = np.random.randn(n_of_inputs, n_of_neurons)
        self.biases = np.zeros((1, n_of_neurons))

    def forward(self, inputs):
        self.inputs = inputs
        self.output = np.dot(inputs, self.weights) + self.biases
```

Data flow following the code can be glanced with the help of the neural network drawn in Figure 6. In this neural network output from any particular neuron, i in the l^{th} layer can be expressed as follows.

$$\sigma_i^l = \sum_{j=1}^{n_{l-1}} a_{l-1}\left(w_{i,j}^l * \sigma_j^{l-1} + b_i^l\right) \tag{8}$$

Here, σ_j^{l-1} is the input from each neuron of the previous layer (l – 1) to any particular neuron of the current layer (l). b_i^l is the bias used for neuron i in layer l and a_{l-1} is the activation function used for the layer l – 1.

Figure 6 Architecture and functioning of neural network (see online version for colours)

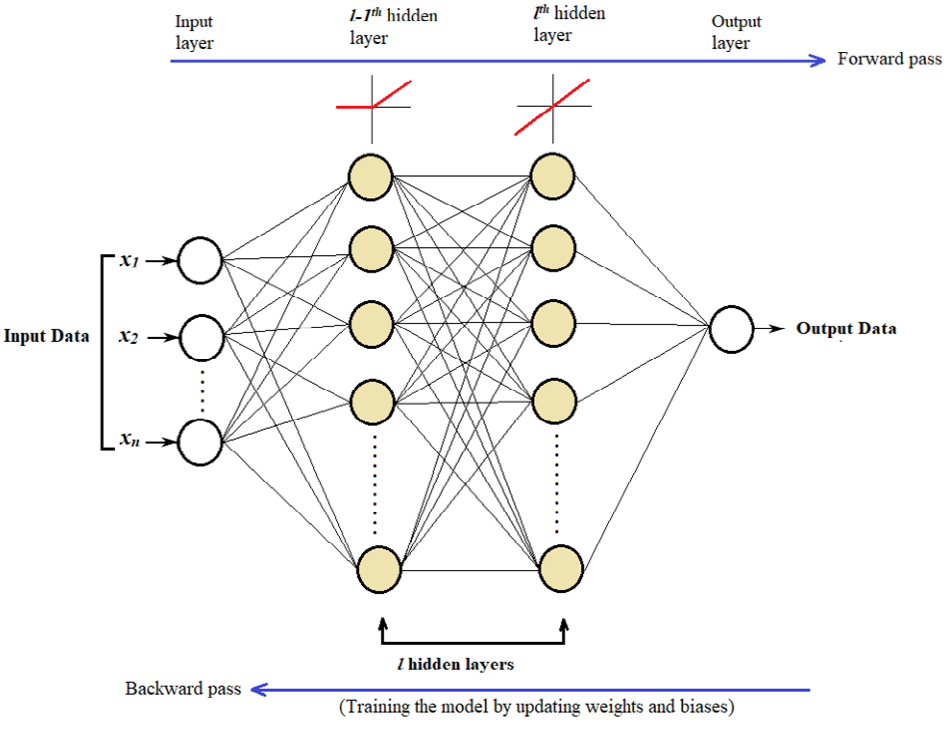
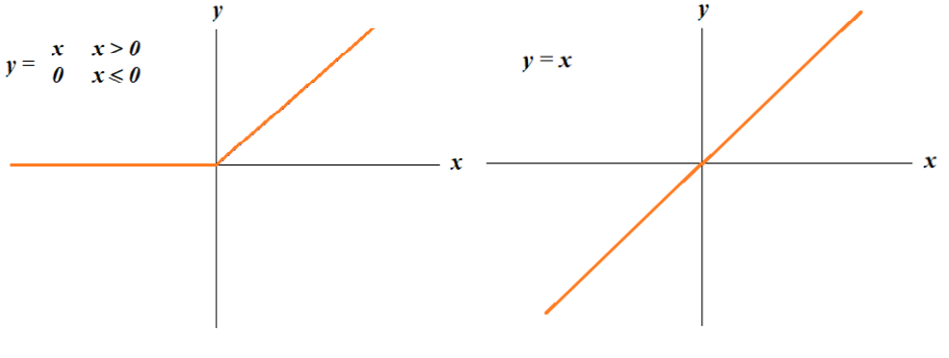


Figure 7 Schematics of (a) ReLU activation function and (b) linear activation functions (see online version for colours)



Because of randomly chosen weights and biases, the output of the forward pass yields a big error. This error has to be minimised by properly tuning weights and biases, which is done in the backward pass. Backward pass is performed for training the neural network, or to say, adjusting/tuning the weights and biases to produce the accurate output. This is done based on the error function (loss or cost function) which is taken as root mean square error (MSE). This root mean squared error is evaluated by the following mathematical formula.

$$Loss\ function\ (MSE) = \mathcal{L}(\hat{y}, w, b) = \frac{1}{n}\sum_{i=1}^{n}\left[Y_i - \mathcal{N}(\hat{y}, w, b)\right]^2 \tag{9}$$

Here, Y_i are the target values which neural network is supposed to predict and $\mathcal{N}(\hat{y}, w, b)$ are the current predicted values by the neural network. MSE of equation (9) is the function of inputs $(\hat{y})$ weights (w) and biases (b). Python code employed for evaluating loss is given in the snippets.

Figure 8 Loss calculation in the forward pass (see online version for colours)

```
class Loss_calculations:
    def evaluate(self, output, y):
        sample_losses = self.forward(output, y)
        data_loss = np.mean(sample_losses)
        return data_loss

class Loss_Mean_Squared_Error(Loss_calculations):
    def forward(self, y_pred, y_true):
        sample_losses = np.mean((y_true - y_pred)**2, axis=-1)
        return sample_losses
```

Weights and biases are updated in the backward pass. For this purpose, gradient $\nabla\mathcal{L}$ of this loss function is evaluated.

$$\nabla\mathcal{L}(\hat{y}, w, b) = \begin{bmatrix} \frac{\partial}{\partial\sigma}\left(\mathcal{L}(\hat{y}, w, b)\right) \\ \frac{\partial}{\partial w}\left(\mathcal{L}(\hat{y}, w, b)\right) \\ \frac{\partial}{\partial b}\left(\mathcal{L}(\hat{y}, w, b)\right) \end{bmatrix} \tag{10}$$

Derivatives of this loss function with respect to weights and biases are used to update weights and biases, and derivatives with respect to inputs are used to chain with the previous layer (Kinsley and Kukieła, 2020). Adaptive momentum (Adam) optimiser is used to train the model (Kingma and Ba, 2014).

To check the accuracy in the regression model, a limiting value is defined which is the standard deviation of the target values. This standard deviation is divided by 250 (our choice, this value defines the desired precision in obtaining the accuracy). Following python code is implemented to check the accuracy of the output from the neural network.

It is found that, with this code, even 80% accuracy gave satisfactory results that were required in the analysis. Full Python code used in the study can be accessed from this Github page [online] https://github.com/hgaur0007/Solution-of-Structural-Mechanics-problems-by-ML-.

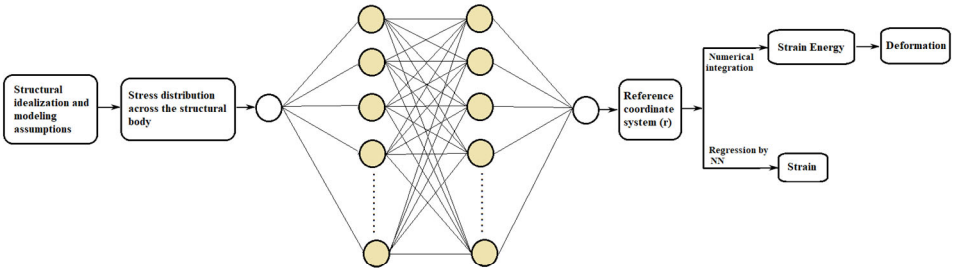
Figure 9 Accuracy adopted in neural network (see online version for colours)

```
# defining the limiting value as standard deviation of the target values
limiting_value = np.std(target_values) / 250
#predicions are the output of the neural network
predictions = activation3.output
# Accuracy is measured with the mean of absolute values obtained from the following inequality.
# False counts to zero and True counts to one
accuracy = np.mean(np.absolute(predictions - target_values) <limiting_value)
```

4.2 Implementation of neural network in the analysis

In the analysis procedure, basically, two neural networks are used. One, which maps true stress to reference coordinate system, and is expressed by equation (11) and another one, which maps reference coordinate system to true strain, and is expressed by equation (12). The flowchart of the analysis procedure using the neural network as regression tool can be viewed in Figure 10. Neural networks are trained with 100% data available from stress-strain behaviour and mapping. Variation of the reference coordinate system (test data which is different from the training data) with true stress can be observed in Figure 10 and the variation of true strain (test data) with reference coordinate system can be observed in Figure 11.

Figure 10 Flowchart of analysis procedure with neural network as regression tool (see online version for colours)



As explained previously, for solving the problem, the methodology requires proper stress distribution across the structural body with proper structural idealisation and modelling assumptions. Once stress distribution in the structural body is defined, corresponding reference coordinate (r) is determined from the output of the neural network, which is expressed by the following equation.

$$r = \mathcal{N}^r\left(\sigma, w^r, b^r\right) \tag{11}$$

Here, $\mathcal{N}^r$ referees the neural network used for regression analysis from stress to reference coordinate (r), σ refers to the true stress (input value), w^r and b^r refers to weights and biases of the neural network obtained after running the optimisation by Adam optimiser. For training this neural network, weights and biases were updated by around 10,000,000 times (iterations). Even 80% accuracy gave satisfactory results. The accuracy of the output values from this neural network could be observed in Figure 11.

Once reference coordinate (r) is obtained, true strain can be obtained with the regression analysis of neural network ($\mathcal{N}^{\varepsilon}$), which maps reference coordinate to strains expressed by the following equation.

$$\varepsilon = \mathcal{N}^{\varepsilon}\left(r, w^{\varepsilon}, b^{\varepsilon}\right) \tag{12}$$

Here w^{ε} and b^{ε} are weights and biases of this neural network after training and, r is the input value of the neural network. This mapping is linear, which could be observed with the help of Figure 12. Around 1,000 iterations sufficiently/satisfactorily tuned the weights and biases, which produced accurate results.

Figure 11 Variation of true stress with reference coordinate system (r) of mid steel (see online version for colours)

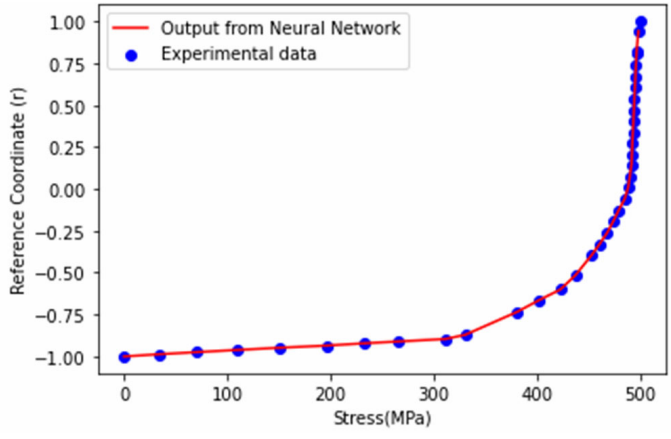
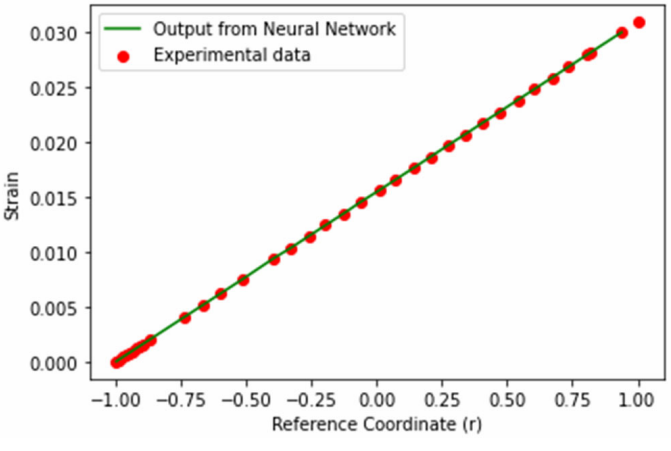


Figure 12 Variation of true strain with reference coordinate system (r) of mid steel (see online version for colours)



5 Numerical problems

In this section, uniaxial bar loaded with point and uniform load acting axially, beam bending problem and plane strain axis-symmetric problem of a long cylinder subjected to internal pressure are solved.

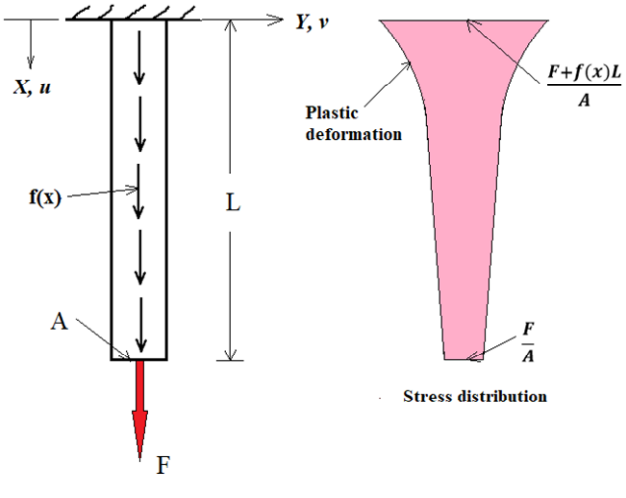
5.1 Uniaxial bar loaded with varying load

Figure 13 shows the uniaxial bar loaded at the free end by the traction force F acting at the free end and a uniform load $f(x)$ acting along the length of the bar downward in X-direction. Considering the magnitude of load combination is such that the material of the bar experiences plastic deformation. One of the possible stress distributions in this condition is shown in Figure 13(b). Stress at the free and fixed end can be evaluated by the following expressions.

$$\sigma_{free\ end} = \frac{F}{A} \tag{13}$$

$$\sigma_{fixed\ end} = \frac{F + f(x)L}{A} \tag{14}$$

Figure 13 (a) Uniaxial bar with loading condition (b) Stress distribution across the length of the bar is shown (see online version for colours)



(a) (b)

Once the stress distribution across the body is defined, strain energy stored in the body can be evaluated by the volume integral of equation (1). As per the loading, the only stress which is acting in the bar is σ_x and at any particular section stress remains constant throughout the area, this strain energy expression is modified as follows.

$$Strain\ energy = A\int_L \sigma_x \varepsilon_x dx \tag{15}$$

As the stress varies across the length, this volume integral can further be modified as follows.

$$Strain\ energy = AL\int_{r_1}^{r_2} \sigma_x \varepsilon_x dr \tag{16}$$

Here, the multiplication $\sigma_x\varepsilon_x$ is integrated because it is not constant and varies across the length with the lower and upper bound of r_1 and r_2 which can be obtained from the neural network $\mathcal{N}^r$, expressed with equation (11) with known stress values (σ_x) across the length of the bar by expressions (13) and (14). True strain ε_x can also be obtained by the neural network $\mathcal{N}^\varepsilon$ expressed with equation (12) by known values of reference coordinate. Numerical integration is performed with the Python code. Once strain energy stored in the bar is evaluated, its deformation can be evaluated by the Castigliano's Theorem expressed as follows.

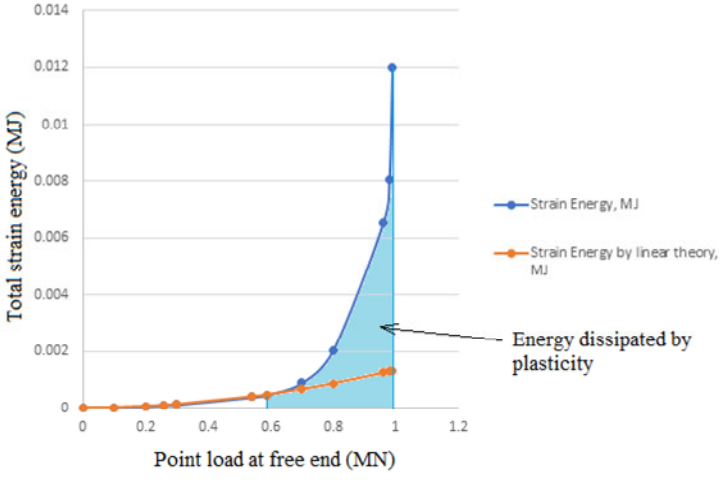
$$D_i = \frac{\partial U}{\partial F_i} \tag{17}$$

Analytical solution (deformation at the free end) by differential formulation (linear theory) of this problem is given by the following equation (Allen, 2013).

$$u = \frac{f(x)}{EA}\left(Lx - \frac{x^2}{2}\right) + \frac{F}{EA}x \tag{18}$$

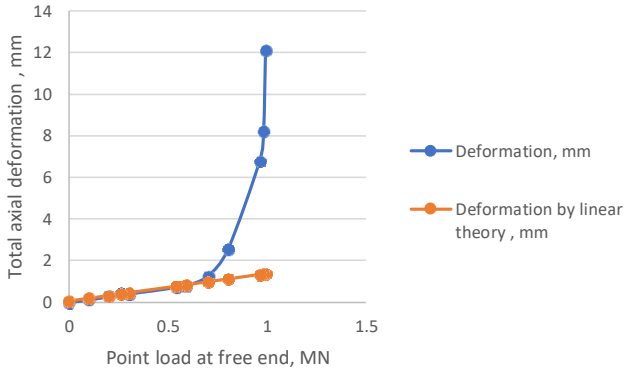
Considering, the area of cross-section of the bar as 20 cm^2 and length as 50 cm. In this analysis, uniform load $f(x)$ is kept constant and taken as 0.4 MN/m. Point load at free end varies in magnitude. Variation of strain energy as well as deformation of the bar at free end, versus the point load F can be glanced in Figures 14 and 15.

Figure 14 Variation of strain energy versus the point load F at free end (see online version for colours)



Note: Static of energy dissipated in plasticity is shown.

Figure 15 Variation of deflection at free end versus the point load *F* at free end (see online version for colours)

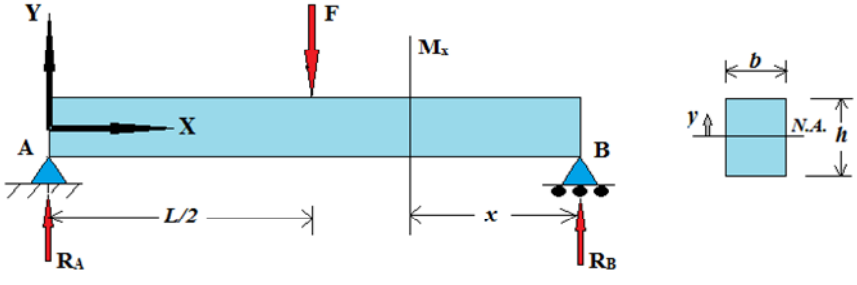


Results of this uniaxial bar are compared with the linear theory in which Young's modulus is taken as 184,128.9683 MPa which is evaluated from the stress-strain curve of the material. It is important to notice the results in linear range, which are very accurately matching with linear theory. In nonlinear range, strain energy, as well as deformation of the bar increases rapidly, this is because of the plastic deformation of the bar as per the material behaviour. Strain energy dissipated because of plasticity is also shown in Figure 14.

5.2 Two dimensional problem of beam bending

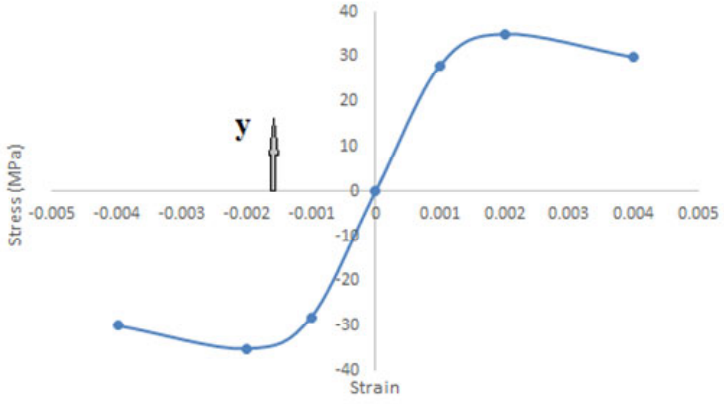
Figure 16 shows a two-dimensional problem of a simply supported beam of length *L*, width *b* and depth *h*, loaded with a concentrated load *F* at the mid span.

Figure 16 Simply supported beam with concentrated load at mid span (see online version for colours)



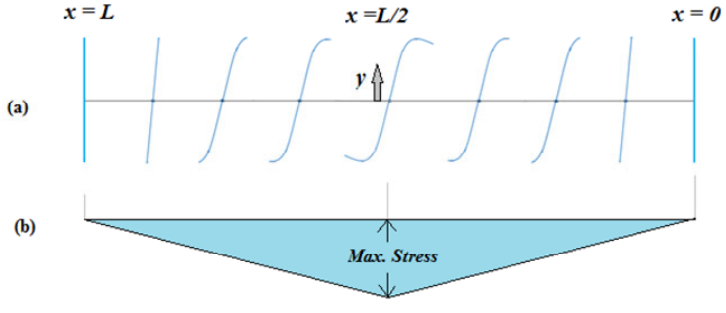
The current state of art of beam bending theory assumes linear variation of the stresses across the depth of the beam. Interestingly, this methodology is capable of analysing structures in the nonlinear range of the material behaviour as well. If the beam is loaded such that it generates stresses corresponding to the plastic/nonlinear range of the material behaviour, it is reasonable to understand that the stress variation across the depth of the beam at any particular section would resemble the variation shown in Figure 17.

Figure 17 Variation of stress across depth of the beam at a particular section (see online version for colours)



Stress variation across the length (for the simply supported beam of Figure 16) is depicted with the help of Figure 18. Its variation, simply depends upon the variation of bending moment across the length.

Figure 18 Variation of stress along the depth (a) and length (b) at different sections of the beam (see online version for colours)



Since the simply supported beam shown in Figure 16 is loaded with the concentrated load at the mid-span, its bending moment at any particular section within the range $0 \le x \le \frac{L}{2}$ can be calculated as,

$$M_x = R_B x \tag{19}$$

where R_B is the reaction at support. Let M_x be the bending moment ant any particular section. Considering stress σ_x in a fibre at any particular depth y from the neutral axis, bending moment at that section can be calculated as,

$$M_x = \int_A \sigma_x y \, dA \tag{20}$$

or,

$$M_x = \int_A \sigma_x y \, dz \, dy$$

or,

$$M_x = b\sigma_x \int_{-h/2}^{+h/2} y dy = b\sigma_x \left[\frac{y^2}{2} \right]_{-h/2}^{+h/2} = \frac{bh^2}{4} \sigma_x$$

Rearranging,

$$\sigma_x = \frac{4M_x}{bh^2} \tag{21}$$

It is to be noted that $\frac{bh^2}{4}$ resembles with the moment of inertia of the section $\left(\frac{bh^3}{12}\right)$ which is found in the derivation of beam bending equations with linear theory of elasticity. Here, $\frac{bh^2}{4}$ avoids any linearity assumption in derivation.

Now, bending energy stored in the beam can be calculated as (Timoshenko and Goodier, 1934),

$$Bending\ energy = \int_V \sigma \varepsilon dV \tag{22}$$

As the stress distribution across the with b remains constant at any section, bending energy can be evaluated as,

$$Bending\ energy = b \int_A \sigma \varepsilon dx \, dy$$

Now, consider stress variation across the length of the beam as depicted in Figure 18.

$$Bending\ energy = b \frac{L}{2} \int_y \sigma \varepsilon dy$$

Now, consider stress variation across the depth of the beam as portrayed with the help of Figure 17. Stress varies from zero at the neutral axis ($r = -1$) and it becomes maximum at the outer layer. Hence, for this variation of stress, the integral modifies as follows.

$$Bending\ energy = 2 * b \frac{L}{2} * \frac{h}{2} \int_{-1}^{r} \sigma_x \varepsilon_x dr \tag{23}$$

Here, the integral is multiplied by $\frac{h}{2}$ as it is calculating strain energy starting from neutral axis to the extreme fibre, either in compression or tension zone. The expression is multiplied again by two because the same energy will be stored, either it is compression or tension zone.

Considering length of the beam as $L = 2$ m, $b = 300$ mm and $h = 400$ mm. If the beam is loaded with a concentrated load of magnitude 2 MN, maximum stress generated (σ_x) at the extreme fibre of the middle span ($x = 1$ m) can be evaluated by equations (19) and (21). For the known stress, corresponding value of reference coordinate system (r) could

be obtained by neural network $\mathcal{N}^r$, expressed by equation (11). Once the value of reference coordinate value (r) is known, corresponding value of strain (ε_x) could be evaluated by the neural network $\mathcal{N}^\varepsilon$ expressed by equation (12). Finally, strain energy stored in the beam bending is evaluated by expression (23) for different magnitudes of load. Deformation in the beam at the mid-span is evaluated by Castigliano's theorem expressed with equation (17). The results are compared with the linear theory and can be glanced Figures 19 and 20. Deformation at mid span for this beam with linear theory is evaluated by following expression of equation (24)

$$y = \frac{FL^3}{48EI} \tag{24}$$

Figure 19 Bending energy stored in the beam for different magnitudes of loads (see online version for colours)

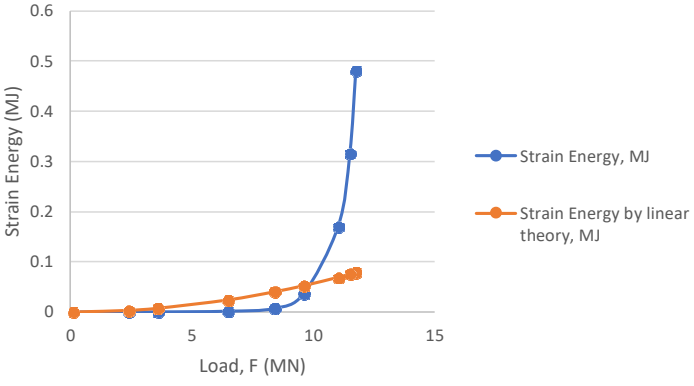
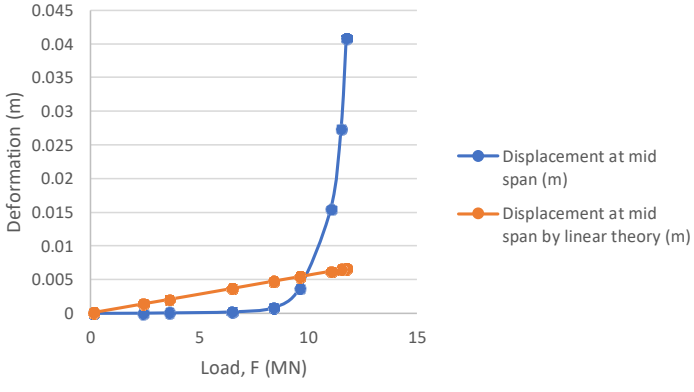


Figure 20 Deflection at the mid span of the beam for different magnitudes of loads (see online version for colours)



It is worthy to note that this approach of structural analysis has given very accurate results, which resembles material behaviour (stress-strain curve). In comparison to linear theory, which is valid until the linear range of the material behaviour, this method is giving accurate results not only in linear range, but in nonlinear range as well.

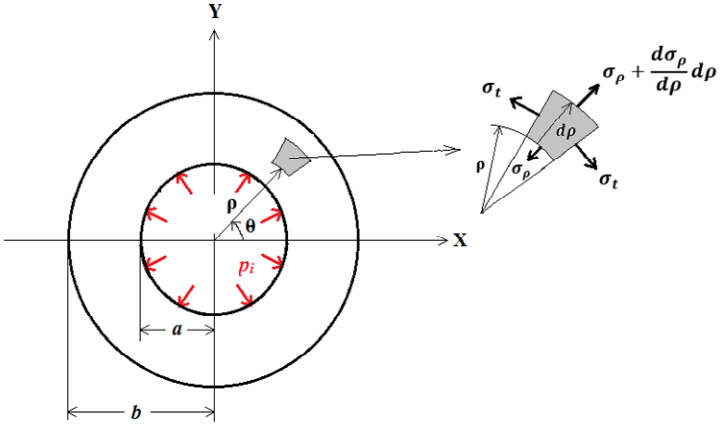
5.3 Plane strain axis-symmetric problem

Consider a long cylinder of inner and outer radius as a and b respectively, which is subjected to the internal pressure of p_i. Figure 21 shows a cross-section of the cylinder in XY-plane in which length of the cylinder is along Z-axis. This is quite a practical problem simulating an infinitely long pipeline subjected to internal pressure from the liquid. This is a plane strain axis-symmetric problem in which deformation is in radial direction only and is constant along the circumferential direction. From the force equilibrium equations, variation of radial (σ_ρ) and hoop ($\sigma_\ominus$) stresses along the thickness of the cylinder are derived as follows (Timoshenko, 1940).

$$\sigma_\rho = \frac{a^2 p_i}{b^2 - a^2}\left(1 - \frac{b^2}{\rho^2}\right) \tag{25}$$

$$\sigma_\ominus = \frac{a^2 p_i}{b^2 - a^2}\left(1 + \frac{b^2}{\rho^2}\right) \tag{26}$$

Figure 21 Section of the cylinder across Z-plane (see online version for colours)



From equations (25) and (26), as there are two different stresses acting, strain energy stored in the cylindrical body can be evaluated as follows.

$$U = U_\rho + U_\ominus \tag{27}$$

Here U_ρ is the strain energy stored because of radial stress and $U_\ominus$ is the strain energy stored because of hoop stress. Strain energy stored per unit length of the cylinder because of radial stress only U_ρ, can be evaluated as,

$$U_\rho = \int_A \sigma_\rho \varepsilon_\rho dA \tag{28}$$

As stresses vary from the inner surface to outer surface, this integration can be modified as,

$$U_\rho = \int_A dA \int_{r_1}^{r_2} \sigma_\rho \varepsilon_\rho d\rho \tag{29}$$

Here, r_1 and r_2 are the values of the reference coordinate system at the inner and outer layer of the cylinder which are obtained from the output of the neural network of equation (11) by input values of stress from equation (25) at inner and outer layer respectively. Further simplifying the expression of equation (29).

$$U_\rho = \pi(b^2 - a^2) \int_{r_1}^{r_2} \sigma_\rho \varepsilon_\rho dr \tag{30}$$

Similarly, strain energy stored because of hoop stress, per unit length of the cylinder, can be evaluated by the following expression.

$$U_\ominus = \pi(b^2 - a^2) \int_{r_1}^{r_2} \sigma_\ominus \varepsilon_\ominus dr \tag{31}$$

Here, r_1 and r_2 can be obtained from the output of the neural network of equation (11) by input values of stress from equation (26) at inner and outer layer respectively. Numerical integration is performed in basic Python code. Once total strain energy of the system by equation (27) is evaluated, displacement in radial direction is evaluated by Castigliano's Theorem of equation (17). The exact radial deformation D_ρ of this problem at the inner layer by the linear theory is given by the expression (Timoshenko, 1940).

$$(D)_{\rho=a} = \frac{ap_i}{E}\left(\frac{a^2 + b^2}{b^2 - a^2} + v\right) \tag{32}$$

Here E is Young's modulus of linear theory which is evaluated from the stress-strain curve in the linear range, v is Poisson's ratio which is taken as 0.3 for mild steel. Taking inner radius a as 10 cm and outer radius as 20 cm, total strain energy of the cylinder, as well as displacement in the radial direction at inner layer is evaluated by Castigliano's theorem. Radial deformation at the inner layer of the cylinder is compared with linear theory. These results can be glanced in Figures 22 and 23 with varying internal pressure.

Figure 22 Variation of strain energy versus internal pressure inside the long cylinder (see online version for colours)

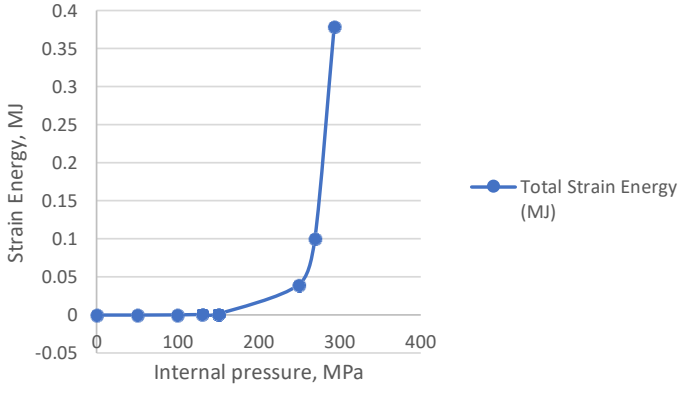
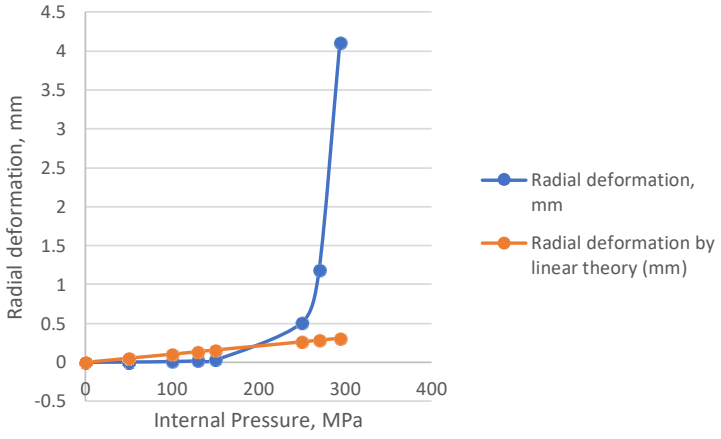


Figure 23 Variation of radial deformation as internal pressure increases inside the cylinder (see online version for colours)



From the results of Figure 23, it is important to notice the accuracy of the results which are matching with linear theory, in the linear range of the material behaviour. In the plastic range, deformation explodes as per the material behaviour.

6 Conclusions

The methodology solves the uniaxial bar, either linear or nonlinear range of the material behaviour, with just one simple integration. It is important to notice the accuracy of the results which are matching with the linear theory in Figures 14 and 15. The beam bending problem of the simply-supported beam is also solved by evaluating simple integral. Results in the linear range are in a good match with the linear theory (Figures 19 and 20). From the results of the cylinder which is subjected to internal pressure only, radial deformation at the inner layer is found to be in accurate match with the results of linear theory (Figure 23).

It should be noticed that finding the solution of these problems in the plastic range is not an easy task. It requires step by step method in which the secant modulus changes with each step, which is a long and cost consuming method. The analysis procedure also evaluates strain energy dissipated in plasticity by evaluating one simple integration (Figure 14). From the results of the uniaxial bar and plane strain axis-symmetric problem, it is important to notice the accuracy of the results, which matched with the linear theory. This proves the validity and accuracy of the method.

References

Alipanahi, B., Delong, A., Weirauch, M.T. and Frey, B.J. (2015) 'Predicting the sequence specificities of DNA-and RNA-binding proteins by deep learning', *Nat. Biotech-nol.*, Vol. 33, No. 2, pp.831–838.

Allen, D.H. (2013) 'Theory of uniaxial bars', in *Introduction to the Mechanics of Deformable Solids*, Springer, New York, NY [online] https://doi.org/10.1007/978-1-4614-4003-1_3.

Anitescu, C., Atroshchenko, E., Alajlan, N. and Rabczuk, T. (2019) 'Artificial neural network methods for the solution of second order boundary value problems', *Computers, Materials and Continua*, Vol. 59, No. 1, pp.345–359.

Areias, P., Rabczuk, T. and Cesar de Sa, J. (2016) 'A novel two-stage discrete crack method based on the screened Poisson equation and local mesh refinement', *Computational Mechanics*, Vol. 58, No. 6, pp.1003–1018.

Areias, P., Reinoso, J., Camanho, P.P., Cesar de Sa, J. and Rabczuk, T. (2018) 'Effective 2D and 3D crack propagation with local mesh refinement and the screened Poisson equation', *Engineering Fracture Mechanics*, Vol. 189, pp.339–360.

Bathe, K.J. (2014) *Finite Element Procedures*, 2nd ed., Massachusetts Institute of Technology, Cambridge, MA 02139, USA.

Gaur, H. and Srivastav, A. (2021) 'A novel formulation of material nonlinear analysis in structural mechanics', *Defence Technology* [online] https://doi.org/10.1016/j.dt.2020.06.018.

Gaur, H., Dakssa, L., Dawood, M. et al. (2021) 'A novel stress-based formulation of finite element analysis', *J. Zhejiang Univ. Sci. A*, Vol. 22, pp.481–491 [online] https://doi.org/10.1631/jzus.A2000397

Ghorashi, S., Valizadeh, N., Mohammadi, S. and Rabczuk, T. (2015) 'T-spline based XIGA for fracture analysis of orthotropic media', *Computers & Structures*, Vol. 147, pp.138–146.

Goswami, S., Anitescu, C., Chakraborty, S. and Rabczuk, T. (2020a) 'Transfer learning enhanced physics informed neural network for phase-field modeling of fracture', *Theoretical and Applied Fracture Mechanics*, Vol. 106, p.102447.

Goswami, S., Anitescu, C. and Rabczuk, T. (2020b) 'Adaptive fourth-order phase field analysis using deep energy minimization', *Theoretical and Applied Fracture Mechanics*, Vol. 107, p.102527.

Guo, H., Zhuang, X. and Rabczuk, T. (2019) 'A deep collocation method for the bending analysis of Kirchhoff plate', *CMC*, Vol. 59, No. 2, pp.433–456.

Huerta, A., Belytschko, T., Fernández-Méndez, S., Rabczuk, T., Zhuang, X. and Arroyo, M. (2018) 'Meshfree methods', *Encyclopedia Comput. Mech.*, Sec. Ed., pp.1–38, John Wiley & Sons, Chichester, UK.

Hughes, T.J. (2000) *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*, Dover Publications, Mineola, NY.

Kingma, D.P. and Ba, J. (2014) 'Adam: a method for stochastic optimization', *International Conference for Learning Representations*, San Diego, 2015, arxiv: 1412.6980Comment.

Kinsley, H. and Kukieła, D. (2020) *Neural Networks from Scratch (NNFS)*, Sentdex, Kinsley Enterprises Inc [online] https://nnfs.io.

Krizhevsky, A., Sutskever, I. and Hinton, G.E. (2012) 'Imagenet classification with deep convolutional neural networks', in *Advances in Neural Information Processing Systems*, pp.1097–1105.

Kumar, P. (2014) *Elements of Fracture Mechanics*, McGraw Hill Education Private Limited, P-24, Green Park Extension, New Delhi 110 016, India.

Lagaris, I.E., Likas, A. and Fotiadis, D.I. (1998) 'Artificial neural networks for solving ordinary and partial differential equations', *IEEE Trans. Neural Netw.*, Vol. 9, No. 5, pp.987–1000.

Lake, B.M., Salakhutdinov, R. and Tenenbaum, J.B. (2015) 'Human-level concept learning through probabilistic program induction', *Science*, Vol. 350, No. 6266, pp.1332–1338.

Lee, H. and Kang, I. (1990) 'Neural algorithms for solving differential equations', *Journal of Computational Physics*, Vol. 91, No. 1, pp.110–117.

Meade Jr., A.J. and Fernadez, A.A. (1994a) 'The numerical solution of linear ordinary differential equations by feedforward neural networks', *Math. Comput. Modelling*, Vol. 19, No. 12, pp.1–25.

Meade Jr., A.J. and Fernadez, A.A. (1994b) 'Solution of nonlinear ordinary differential equations by feedforward neural networks', *Math. Comput. Modelling*, Vol. 20, No. 9, pp.19–44.

Nguyen Huy, B., Zhuang, X. and Rabczuk, T. (2019) 'NURBS-based formulation for nonlinear electro-gradient elasticity in semiconductors', *Computer Methods in Applied Mechanics and Engineering*, April, Vol. 346, No. 1, pp.1074–1095.

Nguyen, V.P., Rabczuk, T., Bordas, S. and Duot, M. (2008) 'Meshless methods: a review and computer implementation aspects', *Mathematics and Computers in Simulation*, Vol. 79, No. 3, pp.763–813.

Nguyen-Thanh, N., Kiendl, J., Nguyen-Xuan, H., Wüchner, R., Bletzinger, K.U., Bazilevs, Y. and Rabczuk, T. (2011a) 'Rotation free isogeometric thin shell analysis using PHT-splines', *Computer Methods in Applied Mechanics and Engineering*, Vol. 200, Nos. 47–48, pp.3410–3424.

Nguyen-Thanh, N., Nguyen-Xuan, H., Bordas, S. and Rabczuk, T. (2011b) 'Isogeometric analysis using polynomial splines over hierarchical T-meshes for two-dimensional elastic solids', *Computer Methods in Applied Mechanics and Engineering*, Vol. 200, Nos. 21–22, pp.1892–1908.

Nguyen-Thanh, N., Valizadeh, N., Nguyen, M.N., Nguyen-Xuan, H., Zhuang, X., Areias, P., Zi, G., Bazilevs, Y., De Lorenzis, L. and Rabczuk, T. (2015) 'An extended isogeometric thin shell analysis based on Kirchhoff-Love theory', *Computer Methods in Applied Mechanics and Engineering*, Vol. 284, pp.265–291.

Nguyen-Thanh, N., Zhou, K., Zhuang, X., Areias, P., Nguyen-Xuan, H., Bazilevs, Y. and Rabczuk, T. (2017) 'Isogeometric analysis of large-deformation thin shells using RHT splines for multiple-patch coupling', *Computer Methods in Applied Mechanics and Engineering*, Vol. 316, pp.1157–1178.

Rabczuk, T. and Belytschko, T. (2007) 'A three-dimensional large deformation meshfree method for arbitrary evolving cracks', *Comput. Methods Appl. Mech. Engrg.*, Vol. 196, No. 29–30, pp.2777–2799.

Rabczuk, T., Akkermann, J. and Eibl, J. (2005) 'A numerical model for reinforced concrete structures', *International Journal of Solids and Structures*, Vol. 42, Nos. 5–6, pp.1327–1354.

Rabczuk, T., Zi, G., Bordas, S. and Nguyen-Xuan, H. (2008) 'A geometrically non-linear three dimensional cohesive crack method for reinforced concrete structures', *Engineering Fracture Mechanics*, Vol. 75, No. 16, pp.4740–4758.

Raissi, M., Perdikaris, P. and Karniadakis, G.E. (2019) 'Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations', *J. Comput. Phys.*, Vol. 378, pp.686–707.

Samaniego, E., Anitescu, C., Goswami, S., Nguyen-Thanh, V.M., Guo, H., Hamdia, K., Zhuang, X. and Rabczuk, T. (2020) 'An energy approach to the solution of partial differential equations in computational mechanics via machine learning: concepts, implementation and applications', *Comput. Methods Appl. Mech. Engrg.*, Vol. 362, p.112790.

Sharkawy, A-N. (2020) 'Principle of neural network and its main types: review', *Journal of Advances in Applied & Computational Mathematics*, Vol. 7, pp.8–19.

Steigmann, D.J. (2002) 'Invariants of the stretch tensors and their application to finite elasticity theory', *Mathematics and Mechanics of Solids*, Vol. 7, No. 4, pp.393–404.

Talebi, H., Silani, M., Bordas, S., Kerfriden, P. and Rabczuk, T. (2014) 'A computational library for multiscale modelling of material failure', *Computational Mechanics*, Vol. 53, No. 5, pp.1047–1071.

Timoshenko, S. (1940) *Strength of Materials*, 2nd ed., D. Van Nostrand Company, Inc, USA.

Timoshenko, S. and Goodier, G.N. (1934) *Theory of Elasticity*, 3rd ed., Mcgraw Hill Education Private Limited, USA.

Vu-Bac, N., Duong, T.X., Lahmer, T., Zhuang, X., Sauer, R.A., Park, H.S. and Rabczuk, T. (2018) 'A NURBS-based inverse analysis for reconstruction of nonlinear deformations in thin shell structures', *Computer Methods in Applied Mechanics and Engineering*, Vol. 331, pp.427–455.

Zhuang, X., Guo, H., Alajlan, N., Zhu, H. and Rabczuk, T. (2021) 'Deep autoencoder based energy method for the bending, vibration, and buckling analysis of Kirchhoff plates with transfer learning', *European Journal of Mechanics – A/Solids*, May–June, Vol. 87, p.104225.