International Journal of Engineering and Management Sciences (IJEMS) Vol. 10. (2025). No. 4.
DOI: 10.21791/IJEMS.2025.19

Research article

Thermoelastic Analysis of Functionally Graded Spherical Bodies Using Deep Neural Networks

DÁVID GÖNCZI

University of Miskolc, Institute of Applied Mechanics, Hungary. david.gonczi@uni-miskolc.hu

Abstract. This paper deals with the numerical analysis of functionally graded spherical bodies subjected to combined thermal and mechanical loads. A method is presented to train deep neural networks to approximate the important solutions. We outline two approaches for generating the training dataset for a deep neural network, followed by a method for creating the neural network itself. Then, through a numerical example, we investigate the axisymmetric problems of radially graded spherical bodies (e.g., ideal spherical pressure vessels). Based on the results obtained, we evaluate the accuracy of solving the outlined problem using the proposed neural network.

Keywords: FGM Sphere, Neural Networks, FEM, Thermoelasticity.

Introduction

With the rapid progression of technology, there is an increasing demand for advanced materials that exhibit tailored properties and specialized behaviour. Engineers across diverse disciplines are turning to engineered materials as alternatives to conventional metals, which often fall short in meeting the complex performance requirements of modern applications. Many engineering challenges call for materials that simultaneously possess high hardness, thermal resistance, and ductility - properties that are difficult to achieve with traditional materials alone. To enhance such characteristics, metals are frequently alloyed with other elements or combined with non-metals to create composite materials. These advanced composites typically offer improved mechanical performance and reduced weight relative to their constituent phases. However, their broader application is often constrained by issues such as delamination, particularly under high-temperature conditions where mismatched thermal expansion can induce failure. In response to these challenges, Japanese researchers introduced the concept of Functionally Graded Materials (FGMs) in the mid-1980s as part of a hypersonic spacecraft development project [1]. FGMs are characterized by a continuous variation in composition and microstructure, leading to corresponding gradients in material properties. This smooth transition between phases eliminates sharp material interfaces, significantly reducing the risk of failure due to thermal or mechanical mismatch. Since their inception, FGMs have been the subject of extensive research, with numerous studies investigating their mechanical behaviour, thermomechanical responses, and structural performance from various theoretical, numerical, and experimental perspectives.

DOI: 10.21791/IJEMS.2025.19

Numerous studies have investigated the mechanical behaviour of functionally graded materials (FGMs) from a wide range of perspectives. Foundational textbooks have addressed solutions to linear elastic problems in homogeneous and non-homogeneous media, as exemplified in [1–3]. In addition, a substantial body of literature has introduced analytical, semi-analytical, and numerical approaches for solving thermomechanical problems in various structural configurations, including hollow spheres, cylindrical shells, beams, and rotating disks. These include papers, such as [4-6]. There are a lot of works, that deal with the analysis of functionally graded spherical bodies. Papers [7] and [8] used multilayered approach and the transformation of the decoupled thermomechanical system of differential equations to solve the problems of radially graded spherical pressure vessels. In [9] the nonlinear temperature and stress distributions of the functionally graded sphere are determined by using the pseudospectral Chebyshev method, furthermore the material properties vary with temperature as well as thickness according to Halpin–Tsai homogenization scheme.

Nowadays, artificial intelligence methods are gaining increasing ground in engineering sciences, particularly the applications of deep neural networks. In the field of functionally graded materials, an increasing number of studies are being published that explore the use of such methods. Paper [10] introduces a PINN (physics-informed neural networks) framework to solve coupled thermo-mechanics for FGMs, predicting displacements and stresses in simple 1 and 2 dimensional problems with higher accuracy by embedding physics-based constraints into the network. In the work of Wu et al. [11], the analysis of multi-directional functionally graded materials is presented, the effects of an instantaneous thermal shock on the thermo-elastodynamic response of a doubly curved panel resting on elastic components is investigated using data-driven deep learning. Work of [12] uses Element Free Galerkin meshless formulation and higher order shear deformation theory for the static analysis of functionally graded plates. This technique estimates the shape function based on moving least squares method and utilizes neural networks. Paper [13] predicted the deformation of multi-directional functionally graded plates with variable thickness resting on an elastic Winkler foundation using deep neural networks. There are a few studies, that investigate special phenomena in spherical bodies, such as [14], which investigated the contact between spherical layers using neural networks.

In this paper functionally graded spherical bodies are investigated which are subjected to combined thermal and mechanical loads. We outline two approaches for generating the training dataset for a deep neural network. One of them is valid for radially graded spherical bodies, in which the material properties are arbitrary functions of the radial coordinate, the other one is finite element method using Abaqus CAE [15, 16]. The temperature field can be a given function or described by thermal boundary conditions which are the thermal loading of the problem. We have constant pressure exerted on the boundary surfaces of the sphere. Our aim is to present a method to solve these problems and generate a date set of the solutions. Then we would like to train neural networks with the results coming from these calculations. Figure 1 shows the sketch of the problem, the mechanical loads are p_1 and p_2 (constant pressure values) while $T(r)$ denotes the temperature field. The modulus of elasticity is denoted by E, the Poisson's ratio is ν, the coefficient of linear thermal expansion is α, and λ is the thermal conductivity. In our paper, we focus on one of the most important field variables, the von Mises equivalent stress distribution. The finite element software Abaqus CAE (and its scripting environment) was used to solve the problems with FEM. To obtain and process the solutions, the Python programming

DOI: 10.21791/IJEMS.2025.19

language was used. In order to create the deep neural network, Tensorflow with Keras packages were utilized.

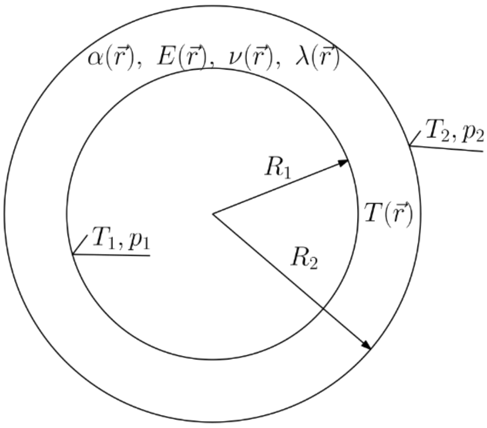

Figure 1. The sketch of the problem.

1. Generating the training dataset

We have multiple approaches to solve the previously discussed thermoelastic problems of spherical bodies. When the material of the axisymmetric problem depends only on the radial coordinate of the spherical coordinate system (r, φ, ϑ), the pressures p_1, p_2 are constants, the equilibrium equation can be expressed as [1-3]

$$\frac{\mathrm{d}\sigma_r}{\mathrm{d}r} + \frac{2(\sigma_r - \sigma_\varphi)}{r} = 0. \tag{1}$$

Considering linearly elastic, isotropic material behaviour, we have Hooke's law as

$$\begin{gathered}
\sigma_r = B\big[(1-\nu)\varepsilon_r + 2\nu\varepsilon_\varphi - \alpha(1+\nu)T(r)\big], \\
\sigma_\varphi = \sigma_\vartheta \ = B\big[\nu\varepsilon_r + \varepsilon_\varphi - \alpha(1+\nu)T(r)\big], \\
B = \frac{E}{(1+\nu)(1-2\nu)},
\end{gathered} \tag{2}$$

where σ_r and σ_φ are the radial- and tangential normal stresses. In case of linearized problems, the kinematic equation of the considered problem has the following nonzero strain components

$$\varepsilon_r = \frac{\mathrm{d}u_r}{\mathrm{d}r}, \qquad \varepsilon_\varphi = \varepsilon_\vartheta = \frac{u_r}{r}. \tag{3}$$

After some manipulation we can reformulate the system of differential equations Eqs. (1-3) into the following equations

$$\frac{\mathrm{d}}{\mathrm{d}r}\begin{bmatrix} S \\ u \end{bmatrix} = \begin{bmatrix} \dfrac{2\nu}{1-\nu} r^{-1} & \dfrac{2E}{1-\nu} \\ \dfrac{B^{-1}}{(1-\nu)} r^{-2} & -\dfrac{2\nu}{(1-\nu)} r^{-1} \end{bmatrix} \begin{bmatrix} S \\ u \end{bmatrix} + \begin{bmatrix} \dfrac{2E}{1-\nu} r \\ \dfrac{1+\nu}{1-\nu} \end{bmatrix} \alpha T, \tag{4}$$

where $S = r^2\sigma_r$ denotes the stress function of the problem [8]. The traction boundary conditions for this case are $\sigma_r(R_1) = -p_1, \sigma_r(R_2) = -p_2$, from which we have $S_1 = R_1^2 p_1,\ S_2 = R_2^2 p_2$. We can get the solutions for this system of equation using the combination of three initial value problem, that can be solved by several numerical methods, such as the Runge-Kutta method. In the first initial value problem we use an arbitrary value for the displacement field $u(R_1)' = u_1{}'$, the initial value of the stress function S_1 and we need the value of $S_2{}'$. For the second initial value problem we need a solution $S_2{}''$ for the same stress function value (as in the first case) and we need a different arbitrary displacement value $u(R_1)'' = u_1{}''$. Due to the linearity of the problem, the actual value of the displacement field can be calculated as

$$u_1 = (u_1' - u_1'')(S_1 + R_2^2 p_2)(S_2'' - S_2')^{-1} + u_1'. \qquad (5)$$

We can create the scripts - based on these equations - to solve a range of problems and save the results of the calculations. We chose the Python programming language to implement the algorithm. An additional advantage of the language is the availability of a wide variety of packages, including those related to artificial intelligence methods.

For a more general approach, we can use commercial finite element software, such as Abaqus CAE. This software provides a wide range of tools for solving the problem, parameterizing it, and evaluating the results [15]. Custom programs (scripts, [16]) can be written to iterate through a user-defined set of problems, solve them, and export selected solution variables to separate files. A control program must also be developed to modify input values, monitor the simulation (handling errors and exceptions if necessary), and save the solutions. In Abaqus, this is done using the Python programming language. The Abaqus CAE preprocessor module includes an integrated Python interpreter, which is suitable for our use case. By adopting this approach, geometry, loads, and boundary conditions can be defined parametrically and generated programmatically. The Abaqus preprocessor provides access to over 500 specialized Python classes, methods, and objects, specifically designed to allow customization and automation of simulations. In our case, the approximation of functionally graded materials can be done multiple ways. We can use user-defined materials and subroutines for the processor module (using Fortran language) or we can approach the problem by using scripts and partitioning. We can divide the geometry into multiple homogeneous subdomains to approximate the material distribution. The more partition we have, the more accurate results we get. For example, we can calculate the material properties in the middle point of the partitions and use them for the homogeneous material of the section. Then the type of simulation must be picked. We can use the coupled thermoelastic solver of Abaqus (coupled temp-displacement step type). From the output database of Abaqus (.odb files) we can extract the displacements, strains or stresses we need and write the values into a csv file. We will pick the equivalent von Mises stress distribution.

2. Creation of the deep neural network

Deep neural networks (DNNs) provide a lot flexibility, when it comes to solving mathematical or engineering problems. Let us consider dense, feed-forward, backpropagating neural networks. In this paper, we aim to examine the accuracy of the distribution of primary variables predicted by the neural networks. To achieve this, the network is designed such that the number of neurons in the output layer corresponds appropriately to the number of target values. When using finite element software, variables

DOI: 10.21791/IJEMS.2025.19

are extracted from specific points along selected element paths (e.g., nodes or integration points), and the number of these points determines the number of output neurons.

A neural network can be trained to predict multiple primary variables simultaneously, or multiple networks can be linked, with each responsible for a different variable. In our selected case study, which involves ideal spherical pressure vessels with axisymmetric geometry and loading (p_1, p_2, T_1, T_2), we investigate the distribution of equivalent (von Mises) stress along the radial coordinate of the spherical body. After the convergence analysis of the finite element mesh, in case of the chosen mesh density, the radial distance between the inner and outer radii is divided into 43 equal segments, resulting in 44 evaluation points. Accordingly, the output layer consists of 44 neurons. The number of neurons in the input layer is determined by the number of input parameters. If we fix the constituent materials of the functionally graded material and we use a one-parameter (m) distribution function, we can have the following parameters: $[R_1, R_2, m, p_1, p_2, T_1, T_2]$. In this case the input layer consists of 7 neurons. If we investigate 10 values for each parameter, we need to perform 10^7 simulation in order to create an input dataset. This number can be significantly smaller, when we use limitations, such as the maximum von Mises stress must be less, than 700 MPa, due to the limitations of Hooke's law. We can investigate the effects of the number of neurons per layer, the number of deep layers, regularisation on the accuracy of the results. To create the neural network, we used Python programming language and Tensorflow (with Keras).

During the training of deep neural networks, two major numerical challenges arise, unstable gradients and overfitting. The unstable gradient problem includes the vanishing gradient, where gradients become very small during backpropagation, hindering weight updates and slowing convergence, and its opposite - exploding gradients, where large weights cause overly large updates. To address these, effective strategies include proper weight initialization and suitable activation functions, such as Glorot initialization with sigmoid, tanh, or softmax activation functions. Another suitable pairings can be ReLU activation with He initialization or SELU (scaled ELU) with LeCun initialization, for self-normalizing behavior. Batch normalization, gradient clipping, and data normalization (e.g., scaling to [0, 1] or [-1, 1]) can also help stabilize training. The second key issue is overfitting, which can be mitigated using a larger training data set, changing the structure of the neural network, regularization techniques like L1, L2, or dropout. Another tool to counter overfitting is early stopping in simple scenarios, though it is generally discouraged as it prevents full training. For the training of the neural networks, the data set is split into training and validation sets (70/30).

In the design of functionally graded spherical pressure vessels, one of the key quantities is the maximum value of the equivalent (von Mises: $\sigma_{eq} = |\sigma_\varphi - \sigma_r|$) stress. This value serves as a critical parameter in stress-peak-based design approaches. If the neural network is to be calibrated to assist in the design process, accurately predicting the maximum stress becomes particularly important, and the choice of loss functions during training plays a crucial role. In this context, we intend to investigate several loss function variations that differ in how they account for such extremums.

DOI: 10.21791/IJEMS.2025.19

3. Investigating radially graded spheres

In our numerical investigation, we consider a radially graded spherical body, in which the constitutent materials (metal and ceramics) are

$$E_1 = 200\text{GPa}, \nu_1 = 0.31, \alpha_1 = 1.2 \cdot 10^{-5}\text{K}^{-1}, \lambda_1 = 44\,\frac{\text{W}}{\text{mK}},$$
$$E_2 = 320\text{GPa}, \nu_2 = 0.24, \alpha_2 = 4.8 \cdot 10^{-6}\text{K}^{-1}, \lambda_2 = 5.5\,\frac{\text{W}}{\text{mK}},$$

furthermore the material distribution P (which can be E, ν, α, λ) is described by the following function

$$P(r) = (P_1 - P_2)(r - R_1)^m t^{-m} + P_2.$$

where t is the wall thickness of the sphere. The dataset we intend to use for the training of the neural network is

$$R_1 = [0.5, 0.8, 1.1, 1.4, 1.3, 1.6, 2];\ c = [1.002, 1.006, 1.01, 1.03, 1.05, 1.75, 1.1, 1.2],$$
$$where\ R_2 = cR_1;\ m = [0.001, 0.01, 0.1, 0.5, 1, 5, 10, 50, 100, 200];$$
$$p_1 = [0, 1, 5, 10, 25, 50, 75, 100, 150, 200];\ p_2 = 0; T_1 = T_2 = [0, 20, 40, 60, 80, 100, 150, 200, 250].$$

We introduced a parameter c, which is the ratio of the internal and external radii. We picked values fo c to investigate thin and thick-walled bodies too. The solutions were obtained using the first method described, namely the direct solution of the differential Eqs. 4, which was then used to train the neural networks using Tensorflow [17]. Since the accuracy of each deep neural network can be noticeably affected by the initial parameter settings, the training process was repeated multiple times, and the network with the best mean absolute error (MAE) for the equivalent stress distribution was selected. We investigated RELU activation with Glorot inicialization and the self-normalizing SELU activation functions with LeCun inicialization. The latter showed better results. To train the network, adam (adaptive moment estimation) and nadam (Nesterov-accelerated) optimizers were used. We rejected solutions where the maximum equivalent stresses were more than 700 MPa.

The number of layers was varied between 1 and 7, while the number of neurons per layer was set to either 256 or 512. When 256 neurons were used per layer, the MAE of the predicted von Mises stress distribution was 4.5 MPa for 2 layers, and 3.4 MPa for 5 layers. Beyond 5 layers, the MAE changed only marginally, while training times increased significantly. The training loss history of the 5-layer network is shown in Figure 2, as a loss (in MPa) – number of training epoch graph. The blue line indicates the loss during the training process, while the green line is the validation loss (*val_loss*). The mean average and validation mean average errors showed the same tendencies. In this case, the loss value plateaued after 25 epochs and showed no further improvement.

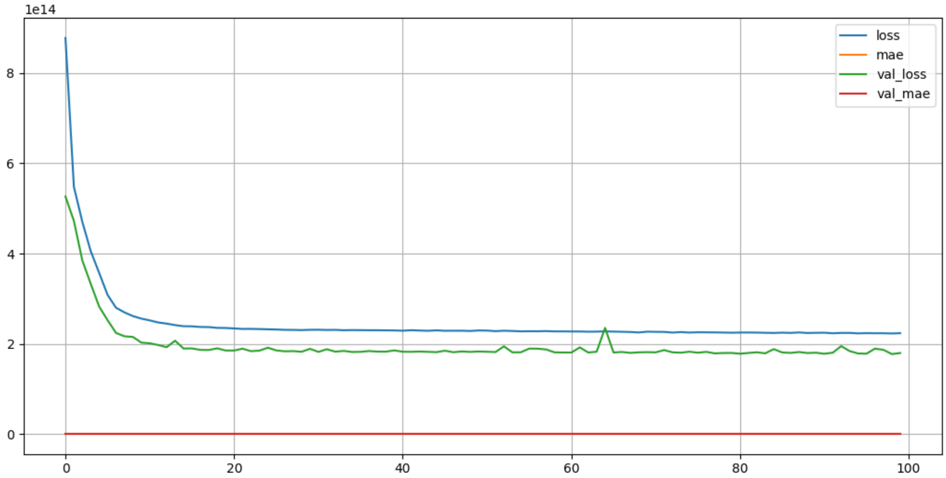

Figure 2. The graph (losses in MPa – number of training epochs) of the training process for the 5 hidden layers.

Table 1 summarizes the results for networks with 512 neurons per layer. When signs of overfitting were observed, dropout regularization and early stopping were applied. In case of 12 layers, the MAE was 1.56 MPa.

number of layers	1	2	3	5	7
mean average error (mae) MPa	28.5	4.1	2.4	1.8	1.65

Table 1. The results of the deep neural networks.

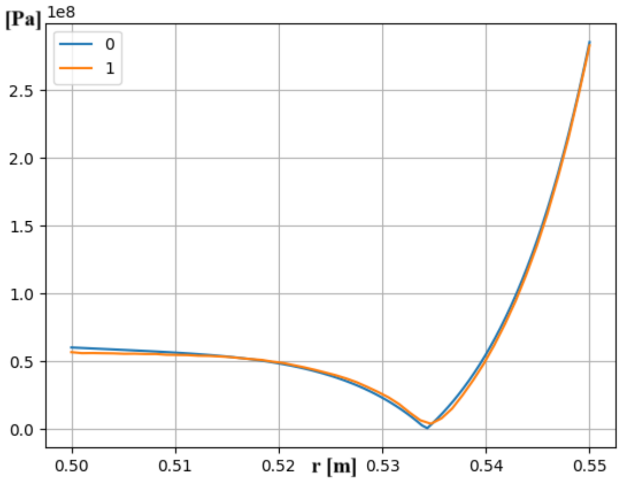

Figure 3. The comparison of results coming from the neural network (1) and the accurate numerical solution (0).

Figure 3 shows a comparison of the different solutions coming from the DNN and the origin numerical solution for a particular parameter combination. The distributions are in good agreement, when the parameter combination is in the range of the training data. Another aspect of the problem is the maximum value of the distribution, which is an important data for the design of functionally graded structures.

One possible approach to modifying the loss function is to increase the weights of data points near the maximum value. To implement this, a threshold value and a weighting factor are defined. If a given variable exceeds the threshold, it is assigned the higher weight during training - meaning the squared error for that data point is multiplied by this weight. In doing so, the network is biased toward more accurate estimation of higher values. An alternative approach (let us call this combined) involves separately computing the error in the predicted maximum values and incorporating this into the overall loss using a weighting factor β. The final loss is then calculated as a weighted sum of the original loss with weight (1-β) and the maximum error with weight β. The comparison of these cases can be seen in Fig. 4, where *NW_ref* denotes the origin squared loss function, *NW_max* is the first modified approach, *NW_combined* is the combined (second) custom loss function.

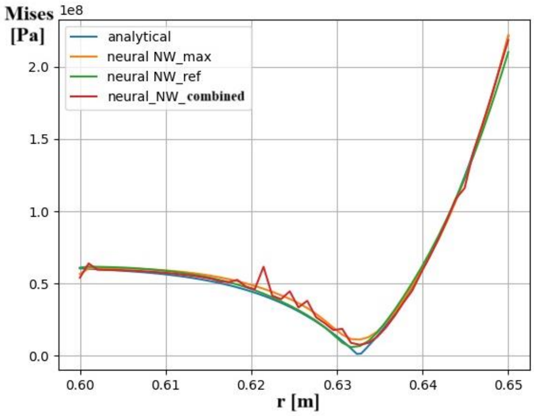

Figure 4. The comparison of the NNs trained with different loss functions.

As expected, in the case of the *NW_max* loss function, the maximum stress values were estimated more accurately. However, the accuracy of the approximation deteriorated for values approaching zero. In the case of the combined loss function (*NW_combined*), spikes appeared in the distribution functions at lower values; nonetheless, the maximum value was still predicted more accurately than in the original neural network configuration. The mean average error on the test dataset did not change significantly, when we trained the neural network with these loss functions. After testing, the solution coming from the neural network were in good agreement with the accurate numerical solution. We got the same conclusion when we trained the neural network for other field variables, such as other stress, strain coordinates or displacement field.

Conclusion

This study focused on the numerical analysis of functionally graded spherical structures subjected to combined thermal and mechanical loading. A methodology is proposed for training deep neural networks to approximate basic field variables. Two approaches for generating the training dataset are presented, followed by the development procedure of the DNN model. A numerical case study involving radially graded spherical bodies (such as idealized spherical pressure vessels) is conducted to demonstrate the approach. The accuracy of the proposed neural network in solving the outlined problem is then assessed based on the obtained results. The results were in good agreement with the accurate solutions, even with a few hidden layers the mean average error were a few percent. A clear drawback of the method (besides its accuracy) is the significant time required to generate the dataset needed for training. Even in the simpler cases, the number of parameters necessitated a large number of simulations. However, once the data is available, the calibration of neural networks can support the design process (particularly in cases where no closed-form solution or simpler methods exist) and one must rely solely on finite element simulations. In addition to the cost of such software tools, the model setup process itself can be time-consuming within these software systems. The use of neural networks can reduce this time demand. We also examined the effect of modified loss functions on the solution, which may be motivated by specific design objectives, such as the critical importance of the maximum stress value. A subsequent question (as future research) may involve identifying the most suitable mesh structure for analysing the problem, as well as examining more complex boundary conditions.

Conflicts of Interest

The author declare no conflict of interest.

References

[1] H-S. Shen, "*Functionally Graded Materials: Nonlinear Analysis of Plates and Shells*" CRC Press, London, 2009.

[2] R. B. Hetnarski, M. R. Eslami, "*Thermal Stresses – Advanced Theory and Applications*" Springer, New York, USA, 2010, doi: 10.1007/978-3-030-10436-8

[3] N. Noda, R. B. Hetnarski, Y. Tanigawa, (2000) "*Thermal Stresses*" Lastran Corporation, Rochester, New York, USA, 2000, doi: 10.1115/1.1349549

[4] L. P. Kiss, "Nonlinear stability analysis of FGM shallow arches under an arbitrary concentrated radial force" *International Journal of Mechanics and Materials in Design*, vol. 16, no. 1, pp. 91-108., 2020, doi: 10.1007/s10999-019-09460-2

[5] S. Rahman, Md. Afsar Ali, "A novel approach to optimize material distributions of rotating functionally graded circular disk under minimum and prescribed stresses" *Materials Today Communications*, vol. 36, 2023, doi: 10.1016/j.mtcomm.2023.106620.

[6] M. D. Demirbaş, D. Çakır (sofuoğlu), "Evaluation of the Performance of ANN Algorithms with the Bidirectional Functionally Graded Circular Plate Problem" *International Scientific and Vocational Studies Journal*, vol. 6, no. 2, pp. 103-115., 2022, doi: 10.47897/bilmes.1207256

[7] D. Gönczi, "Thermoelastic Problems of Multilayered Spherical Pressure Vessels Subjected to Axisymmetric Loading", *International Journal of Engineering and Management Sciences*, vol. 5, no. 3, pp. 106-115, 2020, doi: 10.21791/IJEMS.2020.3.11.

[8] D. Gönczi, "Thermoelastic analysis of functionally graded anisotropic rotating disks and radially graded spherical pressure vessels" *Journal of Computational and Applied Mechanics*, vol. 19, no. 2, pp. 85-104., 2024, doi: 10.32973/jcam.2024.004

[9] Yıldırım, D. Yarımpabuç, V. Arikan, M. Eker, K. Celebi, "Nonlinear thermal stress analysis of functionally graded spherical pressure vessels" *International Journal of Pressure Vessels and Piping*, vol. 200, 2022, doi: 10.1016/j.ijpvp.2022.104830.

[10] M. Raj, P. Kumbhar, R. K. Annabattula, "Physics-informed neural networks for solving thermo-mechanics problems of functionally graded material" *arXiv preprint* arXiv:2111.10751., 2021

[11] X. Wu, Q. Chen, W. He, A. Almadhor, M. Abbas, "Unraveling thermoelastic performance of the multi-directional functionally graded composite structure by data-driven deep-learning approach" *Mechanics of Advanced Materials and Structure*s, vol. 31, no. 26, pp. 8465-8487, 2024, doi: 10.1080/15376494.2023.2259902

[12] K. P. Afsal, K. Swaminathan, N. Indu, "A novel EFG meshless-ANN approach for static analysis of FGM plates based on the higher-order theory" *Mechanics of Advanced Materials and Structures*, vol. 31, no. 25, pp. 6501-6517, 2024, doi: 10.1080/15376494.2023.2231459

[13] N. G. Luu, T. T. Banh, "Static, dynamic and stability analysis of multi-dimensional functional graded plate with variable thickness using deep neural network" *arXiv preprint*, 2023, doi: arXiv:2301.05900.

[14] E. Öner, B. Şengül Şabano, E. Uzun Yaylacı, G. Adıyaman, M. Yaylacı, A. Birinci, "On the plane receding contact between two functionally graded layers using computational, finite element and artificial neural network methods" ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik, 102(2), e202100287, 2022, doi: 10.1002/zamm.202100287

[15] Abaqus 6.13 online documentation, Dassault Systems, 2015, Accessed: 5, 5, 2025

[16] Abaqus 2016. Scripting user's guide, Dassault Systems, 2016, Accessed: 5, 5, 2025

[17] TensorFlow v2.16 online documentation, www.tensorflow.org/api_docs, Accessed: 5, 5, 2025