Design of Machines and Structures, Vol. 15, No. 2 (2025), pp. 28–38.
https://doi.org/10.32972/dms.2025.011

A STUDY ON THE INVERSE DESIGN OF FUNCTIONALLY GRADED CURVED COMPONENTS WITH FINITE ELEMENT SOFTWARE SCRIPTING

DÁVID GÖNCZI

University of Miskolc, Institute of Applied Mechanics
3515 Miskolc-Egyetemváros,
david.gonczi@uni-miskolc.hu
https://orcid.org/0000-0003-4624-7589

Abstract: Inverse design is an important computational method that determines the optimal material distribution or geometry, particularly in functionally graded materials, to achieve predefined performance targets by tailoring spatially varying properties usually by applying optimisation techniques and other artificial intelligence methods. The purpose of this paper is to study the calculation of basic field variables for curved components (such as curved plates, cylinders) made from functionally graded materials and its implementation into the design process using the scripting environment of a finite element software system. Thermomechanical problems are considered, where the bodies are subjected to combined thermomechanical loading. We examine the characteristics of design optimisation through a simpler, one-dimensional example and explore a few possibilities to facilitate the design process.

Keywords: *FGM, FEM, thermomechanics, scripting*

1. INTRODUCTION

With the rapid advancement in technology, the demand for novel materials exhibiting enhanced properties continues to grow. In numerous engineering fields, researchers are increasingly exploring the application of these advanced materials. To improve mechanical performance, metals are often combined with other metals or non-metallic constituents. One widely used method involves creating solid-state combinations known as composite materials, which consist of two or more distinct phases with different material properties. These inhomogeneous composites offer superior characteristics compared to their individual components. However, their application is limited by certain challenges, most notably, delamination, which can significantly reduce structural integrity. As an alternative, Functionally Graded Materials (FGMs) have emerged. These are advanced materials in which the composition and microstructure vary gradually across the volume, leading to a corresponding gradual change in properties. Unlike traditional composites, FGMs eliminate sharp interfaces between constituents, replacing them with a smooth gradient. This gradual transition reduces stress concentrations and the likelihood of failure initiation at material boundaries.

Inverse design is a computational approach where the desired performance or behaviour of a material or structure is specified first, and then the material distribution or geometry is determined to achieve that target. In the context of functionally graded materials, inverse design is particularly valuable because it allows engineers to tailor the spatial variation of material properties – such as stiffness, thermal conductivity, or strength – to meet specific functional requirements. By applying inverse design techniques, such as optimisation algorithms or machine learning models, the gradation profile of FGMs can be systematically determined to enhance performance under complex loading or thermal conditions, while minimising weight or material usage.

Nowadays, research on functionally graded materials is gaining more and more popularity. Numerous books and articles investigate structures made from these materials from various perspectives. Many studies, for example, focus on the analysis of functionally graded disks (Nayak, Bhowmick, & Saha, 2020), (Gönczi, Thermoelastic analysis of functionally graded anisotropic rotating disks and radially graded spherical pressure vessels, 2024), (Alavi, Nejad, Hadi, & Nikeghbalyan, 2024) or the problems of functionally graded beams, e.g., (Kiss L. P., 2020), (Ecsedi, Baksa, & Habbachi, 2023), (Gönczi, 2019), (Kiss L., 2024) or (Mourad et al., 2024), as well as plates and spherical bodies. Artificial intelligence methods, such as optimisation techniques or surrogate modelling using neural networks, are also gaining increasing prominence. Here we can mention (Khatir et al., 2021) or (Callioglu & Muftu, 2025) etc. In an article by (Rahman & Ali, 2023), an optimisation model is developed to obtain the volume fraction of the functionally graded disk corresponding to the minimum and prescribed stresses using FEM. (Gönczi, Thermomechanical analysis of functionally graded components using Abaqus, 2024) outlined the software structure of Abaqus CAE and the main characteristic of Python scripting environment, furthermore compared the FE modelling of functionally graded beams to a semi-analytical solution. Contributors present an optimisation framework for designing functionally graded materials, such as plates in thermoelastic applications, combining random profile generation, deep learning-based surrogate models, and a genetic algorithm. This field is developing at an accelerating pace with the rise of AI.

The present study focuses on identifying some of the key characteristics of finite element scripts that assist in the design of structurally simple components made of functionally graded materials. Our chosen software is Abaqus CAE. We investigate how finite element programs for simulating functionally graded materials can be parameterised, how these can be implemented in Abaqus, and how they can be integrated with artificial intelligence methods such as optimisation algorithms. Through illustrative examples, we present several relevant capabilities of Abaqus. We examine how the design process can be facilitated through the use of scripting.

2. The Abaqus environment

Abaqus CAE (Complete Abaqus Environment) is one of the world leading general-purpose finite element software which consists of three main modules. The processor is the main part (or core), the solver of the system, while the preprocessor sets up the problem for the solver and the postprocessor helps us analyse and present the results. FE software are widely used due to their ability to analyse a broad range of engineering problems. Abaqus CAE uses a three-layer modular architecture. The GUI layer, written in C++ with Python hooks, provides the interactive modelling environment where nearly every action corresponds to a Python command. The API layer is a Python-based scripting interface exposing Abaqus objects and constants for automation and customisation. The core solver layer (Standard, Explicit, CFD) consists of compiled executables that process input files and produce output databases (.odb) for post-processing. An embedded Python interpreter links the GUI and solvers, enabling complete automation, customisation, and integration with external workflows. When creating scripts, we use this interpreter.

We can write our script in a .py file, then call Abaqus's Python interpreter from an external Python process (or directly from the OS shell). If we do not need the CAE modelling API (just text I/O, data processing, .odb output datafile reading), we can use batch execution with Abaqus Python. In this case only the Abaqus Python interpreter is running (but without launching CAE), so only limited modules are available. We have a few packages, that allows us to create external programs that communicate with the Python interpreter of Abaqus, but these tools are very limited (a significant amount of the tools is not available). Another option to use sockets for communication between two interpreters. In this advanced case we need a server application and a client program for the interpreters. One of the popular approaches is to use files for communication. In this case the outside program writes a "command file" (JSON, CSV, or text) with parameters. The Abaqus script reads this file, executes tasks, writes results to another file, which is monitored by the external program. It is still not truly interactive, but easier to implement and robust.

Thus, scripting in the software is carried out in Python, which is of particular importance for our purposes, as many advanced AI packages are developed in this language. TensorFlow, for instance, can be cited as one of the most widely used Python libraries. This enables the results obtained with Abaqus to be easily integrated with other programs, such as those performing optimisation. It is, however, recommended to run the custom scripts directly within the Abaqus Python interpreter. This approach allows for rapid and straightforward execution, and the scripts can also be conveniently utilised within the preprocessor of the software in the form of plug-ins. Such plug-ins can be created to link our programs to simple GUI elements in the preprocessor, thereby facilitating the work of engineers who wish to use (i.e., perform calculations with) our programs without having to be familiar with them. We can built plugins, that are capable of running single simulations or specified optimisation algorithms within the preprocessor. Combining simulations with artificial intelligence methods greatly facilitates the design process. It is advisable to implement them into the development workflow prior to perform physical simulations, as illustrated in Figure 1.

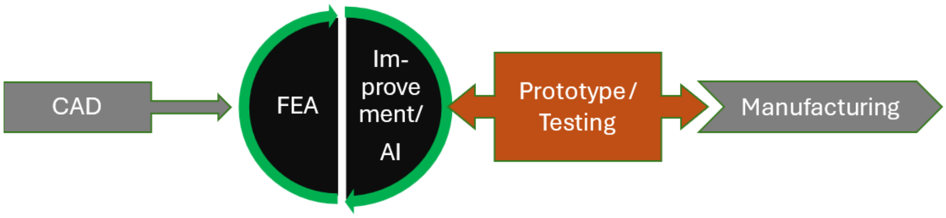

Figure 1. *The recommended implementation of the AI aided simulation into the design process*

3. THE SIMULATION OF THE PROBLEM

Let us consider simple curved structural components. Due to the simplicity of the geometry, creating the parametric script is not particularly complex. One of the challenges arises from the approximation of the material behaviour. The functionally graded behaviour of the material can be described in several ways. One such approach involves the use of user-defined material subroutines (together with the appropriate programming environment, which in the case of Abaqus is Fortran). However, both composites and functionally graded materials can also be modelled using homogeneous layers. In the latter case, the geometry is divided into sufficiently small subdomains, to which homogeneous material properties are assigned according to the material distribution. The values of these parameters may be determined, for instance, by evaluating the governing function at the centre of the subdomain, or by taking the average of the values at the centre and at the vertices of the subdomain. Based on this partitioning, the subsequent stages of the modelling process (such as section definitions, boundary conditions, and load-related features) must also be generated accordingly. In Abaqus, we can use special methods and object for these.

For these curved components, the geometry is defined by the $[R_1, R_2, \vartheta, h]$ with the main parameters of the partitioning $[number_{radial\ layers}, number_{tangetial\ layers}]$. Let R_1, R_2 denote the inner and outer radius of the body, ϑ is the angle and h is the thickness of the component. For these curved components we need sweep technique in order to automatically generate the mesh. We can create separate files to define the material behaviour. When we use object-oriented paradigms, we can create an object for the material with appropriate interfaces, and we can pass this to the program of the simulation.

In the constructor of the simulation object, it is advisable to define the main parameters of the simulation, while less frequently used parameters can be set as needed through setter methods. The flexibility of the Python language allows for a looser design of interfaces (although enforcing these is rather limited). This makes it possible to generalise the creation of optimisation algorithm objects by using a fixed number of arrays (with variable length) tailored to the task at hand. In other words, the optimisation algorithm receives the simulation function, the list of parameters to be optimised, and the fixed parameters in a single array, along with an optional array specifying the initial parameter values. A main program can then handle not only the execution of the optimisation procedure but also the recording of results, error handling, and the possibility of restarting the process.

Another essential question in our case is how to address the relationship between meshing and material composition. This can cause problems if the material composition changes rapidly within a given area, which a coarser mesh cannot capture, thus creating a large stress gradient. In this case, it is possible to program an adaptive mesh controller that, in these zones, after the initial evaluation, refines the mesh. The initial coarser mesh facilitates a faster initial search phase. This will also be relevant in the one-dimensional problem we have examined, for one of the commonly used distribution functions (the power-law based functions).

In this paper, the efficiency of modern FE software is investigated through the optimisation of curved plates and cylindrical bodies. Modified hill climbing algorithms and genetic algorithms will be used.

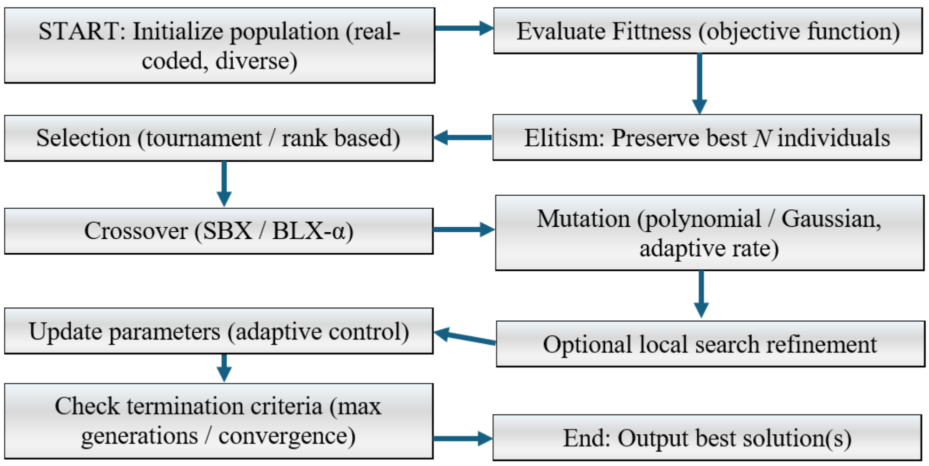

Figure 2. *The flowchart of a genetic algorithm*

In the case of the adaptive hill-climbing algorithm, the parameter neighbours required for the search were initially generated with a larger step size. Once the search failed to identify a better neighbour, the step size was reduced (e.g., halved). The search was then restarted and repeated until it terminated at the smallest prescribed step size. Moreover, we can use random restart to avoid local minimum values. When employing a genetic algorithm (GA), our search was using binary encoding, tournament selection due to the nature of the selected problem. It should be noted here that in structural and engineering optimisation, one of the most effective genetic algorithm approaches is a real-coded, elitist, adaptive GA (Katoch, Chauhan, & Kumar, 2021). Real-coded representation allows direct handling of continuous variables without the precision issues of binary encoding, while elitism ensures the best solutions are preserved from generation to generation. Efficiency comes from combining robust selection methods such as tournament or rank-based selection with powerful crossover operators like simulated binary crossover (SBX) or blend crossover, and adaptive mutation strategies such as polynomial or Gaussian mutation whose rates change during the run to balance exploration and convergence. The algorithm can be further enhanced with adaptive parameter control, where crossover and mutation rates are high early in the search to encourage diversity and reduced later to promote convergence, and with hybridisation, such as adding local search steps or integrating with other metaheuristics. This combination of techniques typically offers strong performance for complex, nonlinear, large-scale optimisation problems.

4. CASE STUDY OF CURVED PLATES AND PANELS

Consider a curved plate with the following material combination and geometry:

$$E_1 = 185\text{GPa}, \nu_1 = 0.29, \alpha_1 = 1.26 \cdot 10^{-5}\text{K}^{-1}, \lambda_1 = 49\frac{\text{W}}{\text{mK}}, \tag{1}$$

$$E_2 = 345\text{GPa}, \nu_2 = 0.21, \alpha_2 = 5.05 \cdot 10^{-6}\text{K}^{-1}, \lambda_2 = 3.5\frac{\text{W}}{\text{mK}}, \tag{2}$$

$$R_1 = 0.7\text{ m}, R_2 = 0.9\text{ m}\;\; t = 0.12\text{ m}, \vartheta = \pi\text{ (rad)}\;\; T_1 = 20\text{ °C}, \tag{3}$$

$$h_{conv} = 30\text{ Wm}^{-2}\text{K}^{-1}, T_{env} = 250\text{ °C}, p_2 = 10^8\text{ Pa}, p_1 = 0\text{ Pa}. \tag{4}$$

The material distribution has the following power-law based function in cylindrical coordinate system:

$$MP(r) = (MP_1 - MP_2)(r - R_1)^m(R_2 - R_1)^{-m} + MP_2, \tag{5}$$

where MP denotes the material properties (E: modulus of elasticity, ν: Poisson ratio, α: coefficient of thermal expansion, λ: thermal conductivity). Our objective is to determine the distribution parameter m while the maximum equivalent stress is minimal. In this planar two-dimensional problem, there are kinematic boundary conditions at the end cross sections, these points can move only the radial direction, while being perfectly insulated. The upper curved surface ($r = R_2$) is subjected to constant pressure (p_2) and third kind thermal boundary conditions (T_{env}, h_{env}), while the lower cylindrical boundary surface ($r = R_1$) has pressure p_1 and T_1. The geometry of the considered problem can be seen in Figure 3.

After running the simulations for m between [30… 0.005], the best solution was at $m = 0.219$, where the von Mises equivalent stress was 451.036 MPa. In the examined simple case, the genetic algorithm, with a small population size and generations (5, 5), performed worse than the hill climbing algorithm ($m = 0.27$), which found the optimal solution very quickly. Here we note that we ran these methods multiple times, and we had a simulation, where the hill climbing method needed 3 restart to find this solution, because after random restarts it got stuck in local minimum positions (e.g. at $m = 0.005$ the stress is 451.8 MPa). The solution depends on a lot of parameters and is random by nature (e.g. depends on the initialisation). By increasing the population size or the number of generations, the solution quickly converged to the optimum. Due to the simplicity of the task, the adaptive hill climbing algorithm also converged rapidly to the solution. However, as the number of parameters to be optimised increases, hill climbing with random restarts (to avoid getting trapped in local extrema) becomes less efficient. In such cases, the genetic algorithm can be a better choice. The stress distribution of the optimal solution can be seen in Figure 4.

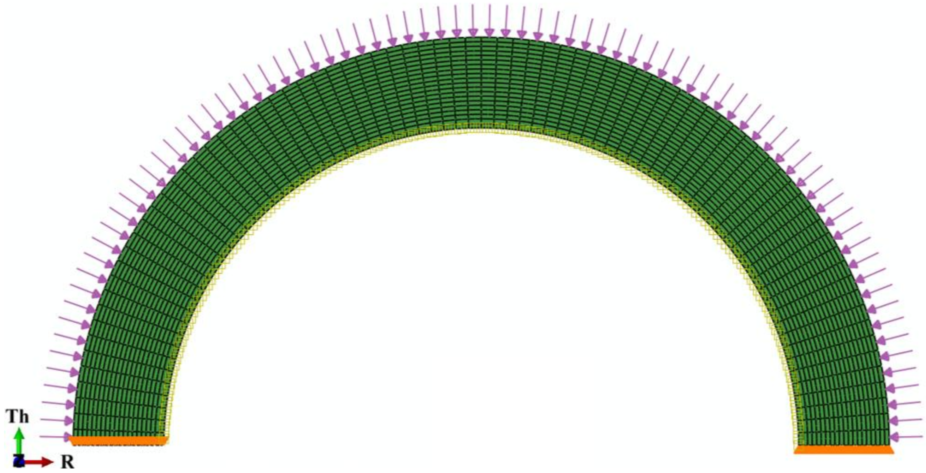

Figure 3. *The sketch of the curved structure*

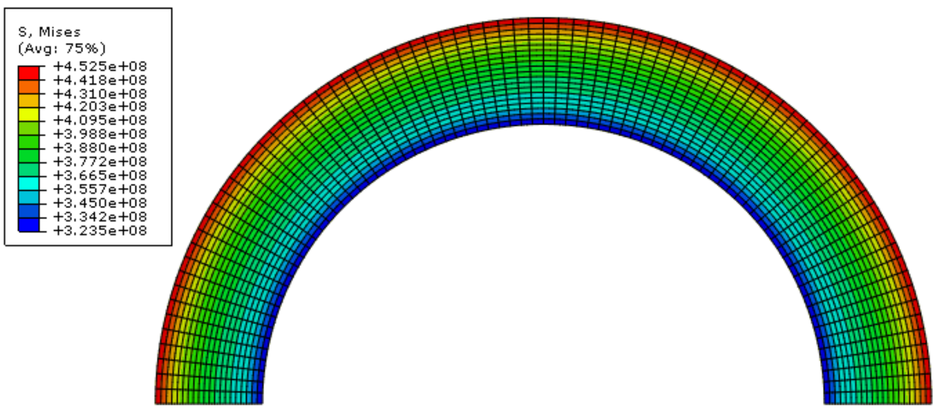

Figure 4. *The von Mises stress distribution of the best solution*

In both algorithms, it is recommended to start with a course mesh, then refine mesh when converging to the best solutions. The difference between the meshes can result in significant errors. The simplest solution is to uniformly refine the global element size. The main disadvantage of this method is the significant increase in computation time. We can use more sophisticated techniques, such as adaptive meshing based on the stress gradient. We can search for big changes of stresses in neighbouring elements and try to introduce a finer local mesh. In three-dimensional problems this can be complicated. In this one-dimensional problem we can segment the radial direction and create multiple ring sections. When we detect large stress gradient (compared to a reference value), we modify the mesh of the segment – in which the problematic area is – and the neighbouring segments and rerun the simulation. This way the time increase is less significant, compared to a uniform mesh refinement. This phenomenon can be seen in Figure 5. Here we started with a course uniform mesh, then we gradually refined the mesh and continued the optimisation with smaller step sizes until we reached a local optimal position (opt).

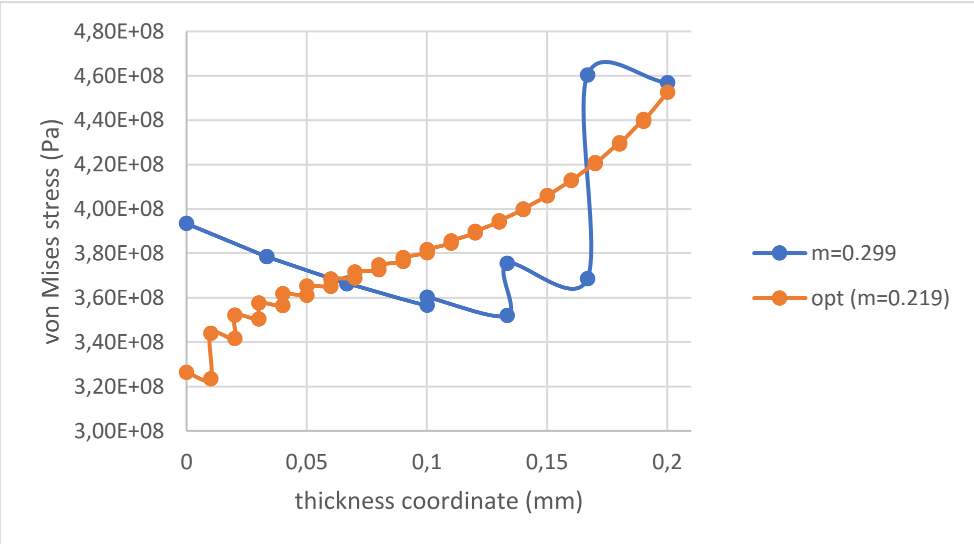

Figure 5. *The effect of mesh refinement on the stress distribution*

In Abaqus, we can introduce additional objective functions for the optimisation. In Python we can define, that we do not need solutions, where the stress gradient reaches a specific value. We can use the stress difference of the neighbouring integration points for this function, or we can use multiple objective functions, such as weight minimisation etc.

In our second numerical example, we modified the model of the problem. This times we had a very long cylindrical body with the same parameters, as in our first example. Due to the nature of the problem, we have plane strain conditions. In this case the optimal value of the material distribution parameter [0…30] was 21.82, where the maximum von Mises stress was 346.83 MPa when we used a uniform greater element size. Upon refining the mesh, the new optimal value was at 30 and the stress 358.7 MPa which highlights the importance of adaptive meshing during the optimisation process.

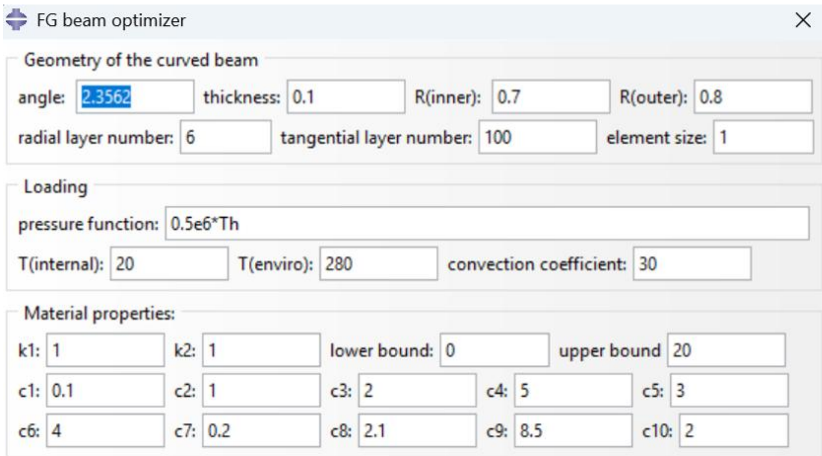

Figure 6. *The plugin of a curved plate with 12 material parameters in Abaqus CAE*

Another useful technique is to create plugins for the completed scripts. Figure 6 illustrates this with an example of a beam or plate, where a program corresponding to a twelve-parameter material distribution can be run through a simple GUI from the Abaqus preprocessor. In this example, the pressure function can be given, as a function of the cylindrical coordinate system. Obviously, more sophisticated plugins can also be developed, in which boundary conditions can be selected for individual edges, and multiple optimisation techniques can be integrated and made selectable within the scripts.

5. CONCLUSION

The study addressed an implementation of inverse design, which refers to a computational strategy in which the desired performance or functional response of a material or structure is defined in advance, and the corresponding material distribution or geometry is subsequently determined to fulfil those requirements. In the case of functionally graded materials, this approach is especially advantageous, as it enables the precise tailoring of spatially varying properties, such as stiffness, thermal conductivity, and mechanical strength, to meet specific design objectives, in many cases by applying artificial intelligence methods. In the case of a simple, unidirectional distribution, we examined the possibilities for optimising curved structural elements. We outlined the main features of realising the optimisation problem using Abaqus CAE and scripting. Through the presented problem, we outlined some of the challenges associated with solving such problems.

REFERENCES

Alavi, N., Nejad, M., Hadi, A., & Nikeghbalyan, A. (2024). Exact thermoelastoplastic analysis of FGM rotating hollow disks in a linear elastic-fully plastic condition. *Steel and Composite Structures*, 51 (4), 377–389.
https://doi.org/10.12989/scs.2024.51.4.377

Callioglu, H., & Muftu, S. (2025). Damped Vibration Responses of Functionally Graded Rotating Discs with Variable Geometry and Modeling with Deep Neural Networks. *Journal of Vibration Engineering & Technologies*, 13 (5), 331.
https://doi.org/10.1007/s42417-025-01900-y

Ecsedi, I., Baksa, A., & Habbachi, M. (2023). Bending of curved beam with rectangular cross section made of double modulus functionally gradient material. *GÉP*, 74 (4), 65–68.

Gönczi, D. (2019). Analysis of a curved bimetallic beam. *Journal of Computational and Applied Mechanics*, 14 (1–2), 41–51. https://doi.org/10.32973/jcam.2019.003

Gönczi, D. (2024). Thermoelastic analysis of functionally graded anisotropic rotating disks and radially graded spherical pressure vessels. *Journal of Computational and Applied Mechanics*, 19 (2), 85–104.
https://doi.org/10.32973/jcam.2024.004

Gönczi, D. (2024). Thermomechanical analysis of functionally graded components using Abaqus. *Multidiszciplináris Tudományok*, 14 (4), 116–124.
https://doi.org/10.35925/j.multi.2024.4.10

Katoch, S., Chauhan, S., & Kumar, V. (2021). A review on genetic algorithm: past, present, and future. *Multimedia Tools and Applications*, 80 (5), 8091–8126.
https://doi.org/10.1007/s11042-020-10139-6

Khatir, S., Tiachacht, S., Le Thanh, C., Ghandourah, E., Mirjalili, S., & Abdel Wahab, M. (2021). An improved Artificial Neural Network using Arithmetic Optimization Algorithm for damage assessment in FGM composite plates. *Composite Structures*, 273, 114287.
https://doi.org/10.1016/j.compstruct.2021.114287

Kiss, L. (2020). Nonlinear stability analysis of FGM shallow arches under an arbitrary concentrated radial force. *International Journal of Mechanics and Materials in Design*, 16 (1), 91–108. https://doi.org/10.1007/s10999-019-09460-2

Kiss, L. (2024). Stability of arches with internal hinge. *Mathematics and Mechanics of Solids*, 29 (10), 1947–1957. https://doi.org/10.1177/10812865241245338

Mourad, C., Rebai, B., Mansouri, K., Khadraoui, F., Berkia, A., & Messas, T. (2024). Investigating the Influence of Material Composition on Bending Analysis of Functionally Graded Beams Using a 2D Refined Theory. *Journal of Computational Applied Mechanics*, 55 (1), 62–76. ű
https://doi.org/10.22059/jcamech.2024.368866.909

Nayak, P., Bhowmick, S., & Saha, K. (2020). Elasto-plastic analysis of thermo-mechanically loaded functionally graded disks by an iterative variational method. *Engineering Science and Technology, an International Journal*, 23 (1), 42–64.
https://doi.org/10.1016/j.jestch.2019.04.007

Rahman, S., & Ali, M. (2023). A novel approach to optimize material distributions of rotating functionally graded circular disk under minimum and prescribed stresses. *Materials Today Communications*, 36, 106620.
https://doi.org/10.1016/j.mtcomm.2023.106620