Application of the two-dimensional Cartesian Finite-Volume Theory in problems of solid mechanics

Ana F. Albuquerque[1], Laís M. S. Pereira [1], Hícaro R. D. Silva[1], Romildo S. Escarpini Filho[1]

[1]*Production Engineering, Federal University of Alagoas*
Floriano Rosa Street, 57200-000, Penedo/Alagoas, Brazil
ana.fernanda@arapiraca.ufal.br, lais.pereira@arapiraca.ufal.br
hicaro.dionizio@arapiraca.ufal.br, romildo.escarpini@penedo.ufal.br

Abstract. This paper presents the development of a Python software with a simple and intuitive graphical interface that allows for the computational modeling of solid mechanics problems (simple beams) using Two-Dimensional Cartesian Finite-Volume Theory (FVT). The numerical approach based on FVT has gained prominence in the computational modeling of structures due to its mathematically simple formulation compared to other classical methods, as well as its low computational cost. Therefore, the main objective is to demonstrate the importance of FVT in the education of engineering undergraduates, highlighting the advantages of its use as a substitute for traditional methods that have long been used in solid mechanics, such as the Finite Element Method (FEM). Additionally, it is important to emphasize the power of Python programming and its practical applications in the daily life of an engineer. Lastly, the aim is to encourage the inclusion of Finite-Volume Theory as a mandatory subject in engineering curricula, facilitating the understanding and development of simulation programs and structural analysis, enabling comparison with other numerical methods and analytical solutions of material strength. This provides a differentiated education for undergraduate students in the areas of computational modeling and structural mechanics to cope with an increasingly competitive job market.

Keywords: Finite-Volume Theory, Computational Modeling, Solid Mechanics, Python, Graphical Interface.

1 Introduction

Computational methods are becoming fundamental tools for solving problems involving the behavior of solid materials. Among the main techniques, Finite-Volume Theory (FVT) stands out. This method for solving engineering problems enables the modeling of complex issues by discretizing the geometry in a manner similar to the Finite Element Method. In this technique, equilibrium and kinematics equations are computed in terms of averages at the faces of the subvolumes, as described by Cavalcante [1]. The Finite-Volume Theory serves as an alternative to the Finite Element Method. This method was formulated by Bansal and Pindera [2, 3]. This numerical technique is focused on the analysis of elastic, plastic, viscoelastic, and thermal structural behavior. The Finite-Volume Theory, as previously elucidated, is geared towards macroscopic analysis or structural-level examination. According to Gattu [4], the development of this method is rooted in the so-called higher-order theory for materials, originally pioneered by Aboudi, Pindera, and Arnold through a series of articles published in the 1990s and consolidated in a review article by Aboudi et al. [5]. This article is centered around the development of a graphical interface using the Python programming language, which is both easily comprehensible and user-friendly. It aims to efficiently and swiftly generate computational solutions to enhance the visualization and comprehension of mechanics. This initiative is particularly relevant due to the existing high level of difficulty associated with the course curriculum. In other words, the intention is to create an open-source tool that is user-friendly and employs Finite-Volume Theory to address basic problems in solid mechanics. This tool would provide undergraduate and graduate engineering students with access to a graphical interface for study and research purposes, while also facilitating the dissemination of the aforementioned formulation.

2 Methodology

In order to thoroughly comprehend the Finite-Volume Theory, the following steps of mesh discretization were undertaken. This involved dividing the material into smaller fragments, formulating equations to calculate the degrees of freedom of the nodes, populating the stiffness matrix, and subsequently integrating this mesh into the programming code using the Python language.

Initially, an in-depth study of the Finite-Volume Theory was conducted, drawing on foundational works in the field, such as Cavalcante [1], and others(Escarpini Filho and Almeida [6]). After mastering the numerical technique and developing the formulations, the computational implementation phase commenced, employing the Python 3.x programming language. This language was chosen due to its current prominence in engineering courses. Furthermore, it's an open-source language, characterized by ease of use, comprehensibility, and versatility.

During this stage, libraries were utilized for user interface generation (Tkinter), mathematical operations involving matrices, solution of linear equation systems (Numpy), and mesh plotting, both undeformed and deformed (Matplotlib).

3 Finite-Volume Theory - FVT

For this study, in the literature review, the work by Cavalcante [1] titled: "Modeling Transient Thermo-Mechanical Behavior of Composite Material Structures using Finite-Volume Theory" (Text in Portuguese) was utilized as a foundational source, supporting the formulation provided below. In Cartesian Finite-Volume Theory, for each k-th subvolume, the displacement field components are approximated by a second-degree polynomial using the Legendre polynomial expansion in the reference coordinates, given by:

$$u_i{}^{(k)}(\bar{X}_1, \bar{X}_2) = U^{(k)}_{i(00)} + \bar{X}_1 U^{(k)}_{i(10)} + \bar{X}_2 U^{(k)}_{i(01)} + \frac{1}{2}(3\bar{X}_1^2 - \frac{d^2}{4})U^{(k)}_{i(20)} + \frac{1}{2}(3\bar{X}_2^2 - \frac{h^2}{4})U^{(k)}_{i(02)} \tag{1}$$

where the values $U_{i(mn)}$ represent the unknown coefficients of the polynomial shift fields u_i for i = 1, 2.

Given the displacement fields given in Equation 1, the infinitesimal strain components for a subvolume are given by:

$$\varepsilon_{ij} = \frac{1}{2}\left(\frac{\partial \tilde{u}_i}{\partial \bar{X}_j} + \frac{\partial \tilde{u}_j}{\partial \bar{X}_i}\right). \tag{2}$$

By following a series of steps, it is possible to derive the equation that relates the average tractions on the subvolume faces with the average displacements on those faces. To achieve this, we start from the Cauchy formula, Hooke's law, and Equation 2 to establish the relationship between tractions and the coefficients of the displacement fields, resulting in 10 unknowns. To determine these coefficients, it is necessary to apply the equilibrium equations within the subvolume, given by:

$$\sum_{i=1}^{2} F_i^f = 0 \tag{3}$$

Where F_i^f is the force in the i direction of face f. And calculating the average displacements on each face for each direction, ensuring compatibility with adjacent subvolumes. This leads to:

$$\langle \mathbf{t} \rangle = \mathbf{K} \langle \mathbf{u} \rangle \tag{4}$$

Where the parameters within $\langle . \rangle$ represent average terms on the subvolume faces, and K is the stiffness matrix of the subvolume. For assembling the global stiffness matrix, the same strategy used in the Finite Element Method is followed, allocating the local stiffness matrix into the global matrix according to the global degrees of freedom of each subvolume. It's worth noting that in the FVT, degrees of freedom are located along the edges/faces.

4 Graphical Interface

As previously explained, the graphical interface for the application of Finite-Volume Theory was developed using the Python programming language in its 3.x version Lane [7] and Bauer [8]. Essentially, the graphical interface or computational tool allows the user to choose from three simple mechanics of materials problems to visualize mesh discretization and displacement fields or deformed structures. The primary goal was to create a user-friendly and intuitive interface that could be utilized by any undergraduate student.

In this interface (Fig. 1), the user can modify various parameters, such as selecting the model to analyze (Cantilever beam, Double Supported beam, and Tensioned beam), the length and height of the beam, applied load, discretization in each direction, and material properties like Young's Modulus and Poisson's Ratio.

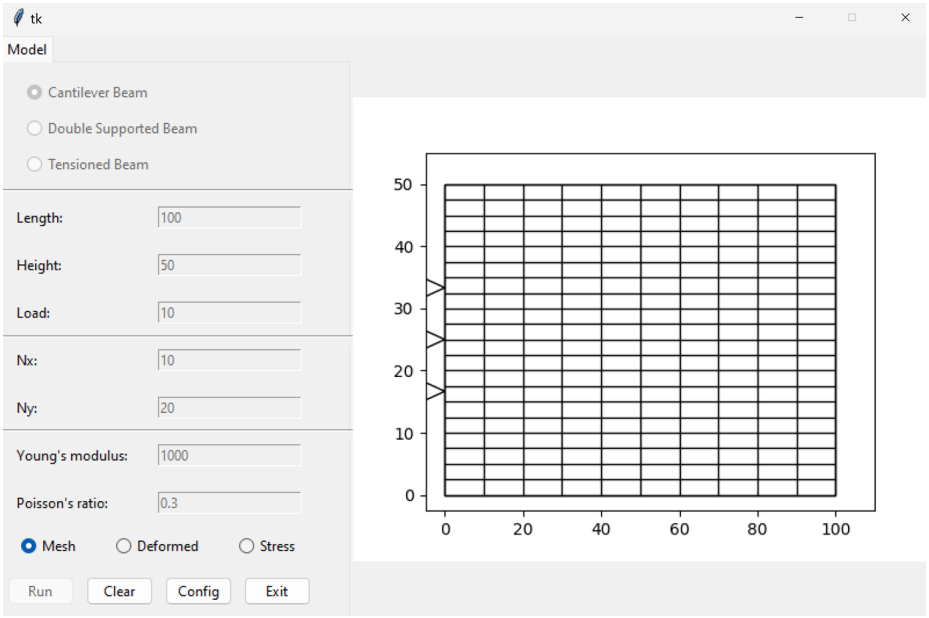

Figure 1. Graphical interface with the initial mesh visualization

In addition to geometric properties (dimensions and mesh discretization) and physical properties (E and ν), the user can also choose to visualize the mesh, displacement field, or stress fields. The latter is still in development. Furthermore, users can clear drawings for new analysis, adjust settings (line colors), or exit the program. It's worth noting that the applied supports are present on all faces of the left vertical side of the beam. For simplification purposes, only three supports have been plotted. In the case of loading, the applied bending forces are uniformly distributed downwards at the free right end.

5 Results

Following the implementation of the Finite-Volume Theory formulation and the computational development of the graphical tool (Fig. 1), it became possible to visualize the results in terms of displacements, as shown in Figure 2. In this example, a generic structure is subjected to bending, and the deformed mesh is depicted in blue. In Table 1 corresponds to the physical and geometric properties that were used.

Properties	**Values (compatible units)**
Length	100
Height	50
Load	10
Discretization along the x-axis	10
Discretization along the y-axis	20
Young's Modulus	1000
Poisson's ratio	0.3

Table 1. Values used in the analysis

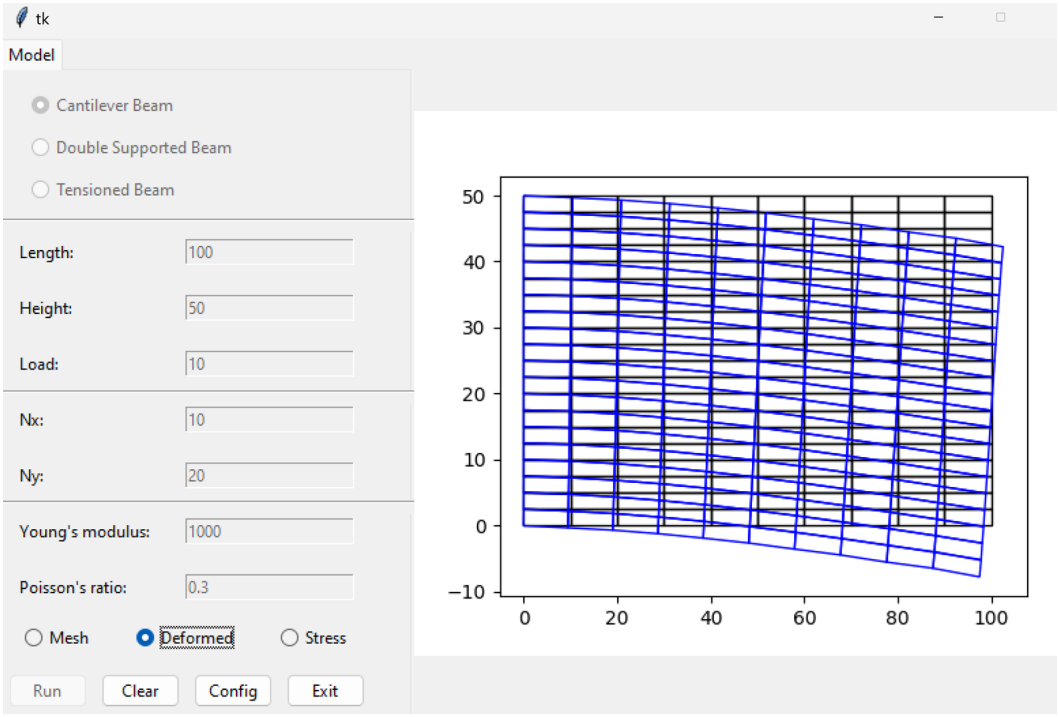

Figure 2. Graphical interface with the visualization of the deformed mesh

6 Conclusions

Ultimately, it is evident that there exists a pronounced correlation between the utilization of Finite Volume Theory within the domain of solid mechanics and the adoption of the Python programming language. This symbiotic relationship yields a robust framework for scrutinizing mechanical behavior, particularly within intricate structural contexts. The computational instantiation thereof yields expeditious and precise outcomes, thereby enhancing the comprehension of intricate mechanical phenomena. Furthermore, the interface offers interactivity and facilitates comprehension through visualizations. Derived from this investigation, it can be inferred that these contributions transcend the confines of conventional engineering paradigms, representing a noteworthy progression in Finite Volume Theory for addressing intricate problem domains. The primary objective lies in the continued advancement of research, with the aim of enriching the educational experiences of both undergraduate and postgraduate students, thereby bolstering their professional development.

Authorship statement. The authors hereby confirm that they are the sole liable persons responsible for the authorship of this work, and that all material that has been herein included as part of the present paper is either the property (and authorship) of the authors, or has the permission of the owners to be included here.

References

[1] M. A. A. Cavalcante, S. P. C. Marques, and M.-J. Pindera. Parametric Formulation of the Finite-Volume Theory for Functionally Graded Materials—Part I: Analysis. *Journal of Applied Mechanics*, vol. 74, n. 5, pp. 935–945, 2006.

[2] Y. Bansal and M. J. Pindera. A second look at the higher-order theory for periodic multiphase materials. *Journal of Applied Mechanics, Transactions ASME*, vol. 72, n. 2, pp. 177–195, 2005.

[3] Y. Bansal and M.-J. Pindera. Efficient reformulation of the thermoelastic higher-order theory for functionally graded materials. *Journal of Thermal Stresses*, vol. 26, pp. 1055–1092, 2003.

[4] M. Gattu, H. Khatam, A. S. Drago, and M.-J. Pindera. Parametric Finite-Volume Micromechanics of Uniaxial Continuously-Reinforced Periodic Materials With Elastic Phases. *Journal of Engineering Materials and Technology*, vol. 130, n. 3, pp. 031015, 2008.

[5] J. Aboudi, M.-J. Pindera, and S. Arnold. Higher-order theory for functionally graded materials. *Composites Part B: Engineering*, vol. 30, n. 8, pp. 777–832, 1999.
[6] R. S. Escarpini Filho and F. P. Almeida. Reinforced Masonry Homogenization by the Finite-Volume Direct Averaging Micromechanics—FVDAM. *Composite Structures*, vol. 320, pp. 117185, 2023.
[7] R. M. Lanes and M. Greco. Aplicação de um método de otimização topológica evolucionária desenvolvido em script python. *Science & Engineering*, vol. 22, n. 1, pp. 1–11, 2013.
[8] J. K. Bauer, P. L. Kinon, J. Hund, L. Latussek, N. Meyer, and T. Böhlke. Mechkit: A continuum mechanics toolkit in python. *Journal of Open Source Software*, vol. 7, n. 78, pp. 4389, 2022.