energies

MDPI

Article

Numerical Modeling and Analysis of an Electromagnetic Device Using a Weakly Coupled Magnetostatic-Mechanical Formulation and the 2D Finite Element Method

Manuel Pineda-Arciniega [1,*], Marco A. Arjona [1], Concepcion Hernandez [1] and Rafael Escarela-Perez [2]

[1] La Laguna Institute of Technology, TNM, Torreon 27000, Mexico
[2] Energy Department, Metropolitan Autonomous University-Azcapotzalco, Mexico City 02128, Mexico
* Correspondence: pinedaarc@gmail.com; Tel.: +52-8713368073

Abstract: This paper presents a methodology to program the weak coupling between magnetic and structural vector fields in an electromagnetic device modeled in two dimensions. The magneto-mechanical coupling phenomenon is present in electromagnetic devices where magnetic forces cause displacements in metallic materials. This work proposes a numerical solution to this problem by applying the 2D finite element method to the governing equations of this coupled multiphysics phenomenon. The well-known formulation yields accurate results; however, it is often not properly integrated into a computer program. This manuscript proposes a flexible and intuitive methodology for the implementation of the complex mathematics involved in this phenomenon into a computer program. The computer code receives the input parameters, discretizes the geometry by generating a 2D finite mesh, solves the resulting equations using the finite element method, and finally exports the results of the magnetic ang mechanical fields. The modeling is performed using an open-source platform for programming the finite element method in the programming language Python, and afterwards, the results are compared against a commercial software as validation of the proposed numerical approach. The novel magneto-mechanical coupling methodology is used to solve an engineering application, namely an electromagnetic actuator.

Keywords: magnetic; mechanical; multiphysics; coupled problems; finite element method

Citation: Pineda-Arciniega, M.; Arjona, M.A.; Hernandez, C.; Escarela-Perez, R. Numerical Modeling and Analysis of an Electromagnetic Device Using a Weakly Coupled Magnetostatic-Mechanical Formulation and the 2D Finite Element Method. *Energies* **2023**, *16*, 2182. https://doi.org/10.3390/en16052182

Academic Editor: Andrea Mariscotti

Received: 2 January 2023
Revised: 14 February 2023
Accepted: 22 February 2023
Published: 24 February 2023

1. Introduction

The electric current flowing through the coil of an electrical device generates an electromagnetic field. This electromagnetic field generates magnetic forces on metallic materials. When these magnetic forces are present in the air gap of an electrical device, elastic deformations or displacements may occur [1]. Such a method of converting electrical to mechanical energy is widely employed in several applications, some of which include: electrical protection systems [2], sensors [3], pneumatic valves [4], cranes [5], and even artificial hearts [6]. The optimal design and performance of such electromagnetic devices result in low maintenance costs, assurance of safety measures, and an increase in production quality. Therefore, a computer program for simulating the magneto-mechanical phenomenon is proposed in this paper.

Magneto mechanical coupling can be used to analyze the structural behavior, i.e., buckling or bending of a non-magnetic material, such as steel alloys. However, in the presence of ferromagnetic materials, a coupled analysis must include the magnetostrictive and reluctance forces (also called Maxwell forces) [7]. The deformations caused by the two types of forces can be calculated separately or simultaneously [8]. Consideration of these deformations can lead to the calculation of acoustic vibrations in electromagnetic devices [9]. However, it is possible to calculate the displacement caused by magnetic forces alone. Dong [10], et al., carried out vibration analysis on a permanent magnet synchronous motor. In this work, a simulation model capable of predicting the noise in an electric

machine is performed. In the calculation of the magnetic forces, they only considered the radial force obtained by means of Maxwell's stress tensor. In another paper, Lundström and Aidanpää [11], reported the dynamic consequences of magnetic forces in a generator. In this manuscript, the change in the geometry of a generator model is calculated. The displacements are found by the virtual work method, and magnetostriction forces are not considered when obtaining the total forces.

Such magnetic-mechanical phenomenon constitutes a coupled problem that only a coupled model can address in three steps: firstly, the magnetic flux density is derived by solving the equations of the magnetostatic field; next, the Maxwell tensor method is employed for finding the magnetic force; lastly, the obtained magnetic load feeds the formulation for the mechanical field resulting in displacements in the non-ferromagnetic material. Magnetic forces can be determined using the three following methods: the Lorentz force, the principle of virtual work, and Maxwell's stress tensor [12]. In this paper, the Maxwell stress tensor method is implemented for finding the magnetic force resultant, used as a source term in structural field equations to calculate deformations. Such an approach is usually known as weak coupling [13]. A second approach is the strong coupling which involves magnetic and structural governing equations being solved simultaneously [14].

The finite element method has been commonly used to solve magneto-mechanical coupled problems; one example of such a problem was presented by Metsch, et al. [2]. Their results show how the finite element method is used in the solution of magneto-mechanical coupled problems. Li, et al., 2007 [15], studied the relation between electromagnetic forces, vibrations, and eccentricity of a 250 MW hydropower generator. They also applied the strong coupling method by using the finite element method and the Maxwell stress tensor.

In the past, an accurate solution of a multiphysics problem with numerical methods required high computational resources. However, many current commercial software packages are capable of solving a coupled problem within a few minutes; where the initial input parameters are entered into a graphical interface, after which a mesh is generated, and the solving process is executed automatically. Unfortunately, no interaction with the complex background of the mathematical models is involved; however, the finite element open-source software provides options for understanding and manipulating the formulation programming.

Using the open-source computer program FEniCS platform for solving partial differential equations enables researchers to translate complicated mathematical problems into efficient finite elemental method code [16]. The finite element package accepts various coupling methods and implements the finite element method to obtain an automatic solution for partial differential equations with customizable capabilities and shorter processing times [17]. The finite element package comprises four core elements: DOLFIN, the Python finite element library; Unified Form Language (UFL), a finite element symbolic language; The finite element computer software Form Compiler (FFC) [18], that processes UFL language to C code; and Finite Element Automatic Tabulator (FIAT), where the shape and order data of the finite elements are stored [19]. UFL provides a flexible and easy-to-use interface that can encourage investigators to write software code including mathematical expressions that describe the behavior for any given engineering problem, such as deformation due to magnetic field forces [20,21].

As an illustration of the effectiveness of finite element package in modeling electrical machine problems is the paper by McDonagh, et al., 2022 [17], which presented an electromagnetic model analysis of a three-phase permanent magnet synchronous motor model. The formulation implemented in this paper also involves the strong coupling method. The results are verified by an equivalent model using a commercial software [12]. The implementation of the finite element package demonstrates the scalability, flexibility, and simplicity of the UFL language to define the required formulation.

This paper focuses on calculating and analyzing the displacement in an electromagnetic actuator with a nonmagnetic steel element under the influence of magnetic forces originating from a constant magnetic field produced by a direct current excitation. The

paper introduces a numerical solution of a weakly coupled magnetostatic-mechanical problem that can be solved within the Python framework. The developed computer program finds the solution to the system of equations by obtaining the deformations. This structural response allows to predict the kinetic behavior of actuators. The aim is to engage users firsthand with the numeric solution of partial differential equations using the finite element method and with the mesh generation and refinement process. The second purpose of this work consists of demonstrating the engineering application and demonstrating the validity and accuracy of the weak coupling technique for this type of problem.

2. Magnetostatic-Structural Problem

2.1. Problem Definition

The domain geometry for the proposed problem area is illustrated in Figure 1. The subdomains are identified as: magnetic core (gray color); nonmagnetic actuator (yellow color), coil (orange color), and air (blue color). The air gap between the core and the actuator is 0.1 cm in dimension. The materials are assumed as isotropic and not all of the domain is homogeneous. The relative permeability (μ) for the iron core is 5013, for the steel alloy is 500, and for copper and air the relative permeability is 1. The alloy steel actuator has a Young's modulus (E) of 209 GPa and a Poisson's ratio (Po) of 0.3. A direct current excitation of 1 A is injected into the winding area. The magnetic insulation is placed at the outer boundary of the air domain, thus a Dirichlet condition of $A_z = 0$ at this boundary is imposed. Meanwhile, the lower boundary for the steel alloy element is considered fixed for the mechanical domain, i.e., a $\boldsymbol{u} = 0$ on this boundary. Locations for boundary conditions are indicated by the green lines shown in Figure 1.

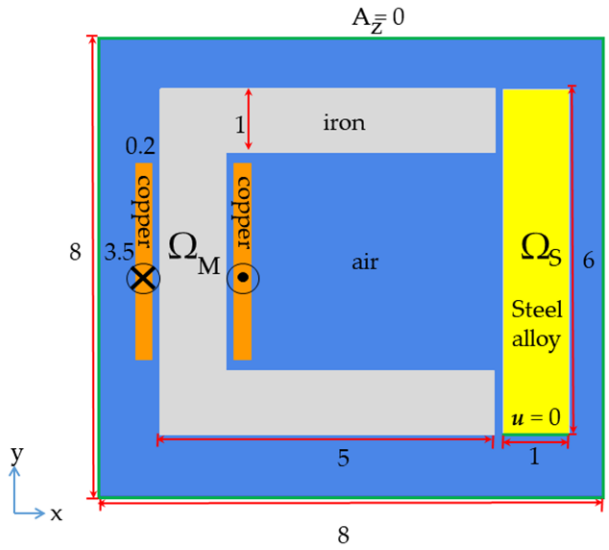

Figure 1. Domains and dimensions of the magneto-static mechanical problem. All dimensions are in cm.

2.2. Two-Dimensional Coupled Magnetostatic-Mechanical Formulation

The magnetostatic-mechanical problem is solved by the weak coupling method, in which the solution of the magnetic field is affected by the elastic deformation effect of the mechanical field, and vice versa. The governing equations for the magnetic domain are obtained from Maxwell's electromagnetic equations. The equations of motion and deformation-displacement equations for linear elastic bodies are considered the governing equations for the mechanical field of the problem. Both sets of equations are expressed in the two-dimensional Cartesian coordinate form. Firstly, the magnetic vector potential is computed from the application of the constitutive equation of the material. Secondly, the magnetic flux density is derived from the calculated magnetic vector potential. Subsequently, the magnetic forces are obtained from applying the Maxwell's stress tensor method

to the magnetic flux density. Finally, the magnetic forces are added to the source term of the mechanical formulation. The solution of the mechanical field consists of finding the displacement of the steel element. Thus, a coupling between the magnetic and mechanical domains is achieved.

The general form of the Ampere's Law for a magnetostatic problem is provided by:

$$\nabla \times \boldsymbol{H} = \boldsymbol{J}, \tag{1}$$

where $\boldsymbol{H}$ is the magnetic field and $\boldsymbol{J}$ is the current density [12]. Furthermore, the constitutive relationship of the material is provided by:

$$\boldsymbol{B} = \mu \boldsymbol{H}, \tag{2}$$

where μ is the permeability of the material and $\boldsymbol{B}$ is the magnetic flux density [22], which can be represented as:

$$\boldsymbol{B} = \nabla \times \boldsymbol{A} \ , \tag{3}$$

where $\boldsymbol{A}$ is the magnetic vector potential.

Equation (1) in terms of $\boldsymbol{A}$ and for a two-dimensional problem, that is, substituting (3) and (2) into (1), can derive the 2D Poisson Equation:

$$-\mu^{-1}\nabla^2 A_z = J_z. \tag{4}$$

Equation (4) is used to solve the magnetostatic problem, and once the magnetic field intensity is obtained, the force components over a differential area can be calculated with the Maxwell stress tensor [12]:

$$\frac{\mathrm{d}f_x}{\mathrm{d}S} = \mu H_x H_y, \tag{5}$$

$$\frac{\mathrm{d}F_y}{\mathrm{d}S} = \frac{\mu}{2}\left(H_x^2 - H_y^2\right), \tag{6}$$

where $\mathrm{d}f_x/\mathrm{d}S$ and $\mathrm{d}f_y/\mathrm{d}S$ are the normal and tangential components of the force over the airgap surface of the nonmagnetic steel element, respectively.

The resultant of the force components integrated over the air gap surface is used in Hooke's Law to obtain the displacement:

$$-\nabla \cdot \boldsymbol{\sigma}(\boldsymbol{u}) = \boldsymbol{f}, \tag{7}$$

where $\boldsymbol{\sigma}$ is the stress tensor, $\boldsymbol{u}$ is the displacement, and $\boldsymbol{f}$ is the magnetic force.

The stress tensor is provided by:

$$\boldsymbol{\sigma} = \lambda \ \mathrm{tr}(\boldsymbol{\varepsilon})\boldsymbol{1} + 2G\boldsymbol{\varepsilon}, \tag{8}$$

for a natural condition where tr() is the trace function, $\boldsymbol{\varepsilon}$ is the strain tensor, $\mathbf{1}$ is the identity matrix, and λ and G are the Lamé coefficients provided by [23]:

$$\lambda = \frac{E\nu}{(1+Po)(1-2Po)}, \tag{9}$$

and

$$G = \frac{E}{1+Po}. \tag{10}$$

where E is the Young's modulus and Po is the Poisson's ratio.

3. Variational Formulation

To solve Equations (4) and (7), this system requires to be expressed in a variational formulation. The variational formulation represents the initial stage in applying the finite element method [14]. The finite element software project builds upon the effectiveness and

precision of the finite element method [24]. The governing partial differential equations are expressed in variational formulation with a continuum finite element configuration. The finite element continuous configuration defines the solution for A_z and u, in respective infinitesimal function spaces. The infinitesimal function spaces are substituted by discrete test and trial spaces. This allows the integrations spaces to have finite dimensions. Test and trial spaces are independently specified for the two physical fields.

The finite element package automatically applies the Galerkin method for approximating the solution [25]. To translate the partial differential equations into the variational form, the partial differential equations must be multiplied by a test function v; the resulting formulation is then integrated over the subdomains, then integration by parts is performed for the terms that have second-order derivatives. Unknown variables (A_z, $\boldsymbol{u}$) must be declared as test functions in the variational form. The function properties can be specified by the test and trial functions, which belong to the function spaces [26]. The variational formulation elements, consisting of test and trial functions, function spaces, solutions, and integrals over the domains, can be implemented into the finite element software through a straightforward procedure.

The variational formulation of the magnetic field is obtained by multiplying both sides of the governing equation by the test function (v), then integrating by parts and finally removing the integral boundary term, as the Dirichlet condition ($A_z = 0$) is then imposed:

$$\int_{\Omega_M} -\mu^{-1}\nabla^2 A_z \cdot \nabla v \, dx = \int_{\Omega_M} J_z v dx, \tag{11}$$

where dx denotes the differential element for the integration over the space of magnetic subdomains, Ω_M; and J_z stands for the current density. To incorporate the linear variational problem into the computer program, Equation (11) must take the form:

$$a(A_z, v) = L(v) \tag{12}$$

where

$$a(A_z, v) = \int_{\Omega_M} -\mu^{-1}\nabla^2 A_z \cdot \nabla v \, dx, \tag{13}$$

$$L(v) = \int_{\Omega_M} J_z v dx. \tag{14}$$

where $a(A_z, v)$ is the bilinear form and $L(v)$ is the linear form that specifies the source term. The known variables are grouped in linear form, and the unknowns are collected in bilinear form. The definition of (12), (13) and (14) constitute the linear magnetic system. This is translated directly into Python code, improving understanding of the complicated PDE problem and its relationship to computer programming.

The approach applied to the magnetic field equations is also considered to obtain the variational formulation for the mechanical domain; however, slight changes are made. For this, a continuous interpolation of 2 degrees is considered. The mechanical variational formulation is found with the inner product of (7) and a test vector function and integrating over the mechanical domain, then we obtain:

$$-\int_{\Omega_S} \Big(\nabla \cdot \boldsymbol{\sigma} \Big) \cdot v \, dx = \int_{\Omega} \boldsymbol{f} \cdot v dx, \tag{15}$$

where $\nabla \cdot \boldsymbol{\sigma}$ holds second-order derivatives of the unknown $\boldsymbol{u}$ for the displacements, and then integrated by parts:

$$-\int_{\Omega_S} \Big(\nabla \cdot \boldsymbol{\sigma} \Big) \cdot v \, dx = \int_{\Omega_S} (\boldsymbol{\sigma} \cdot \nabla v) \, dx - \int_{\partial\Omega_S} (\boldsymbol{\sigma} \cdot \boldsymbol{n}) \cdot v dx, \tag{16}$$

where $\boldsymbol{n}$ is the outward unit normal vector over the boundary, and ds is the integration over the surface. The product $\boldsymbol{\sigma}\cdot\boldsymbol{n}$ is the traction vector at the boundary, and is assumed to be zero for this problem. The right-hand term of (16) is then:

$$\int_{\Omega_S} (\boldsymbol{\sigma}\cdot\nabla v)\,\mathrm{dx} = \int_{\Omega} \boldsymbol{f}\cdot v\mathrm{dx}. \tag{17}$$

The summarized variational formulation for the mechanical field is:

$$a(\boldsymbol{u},\ v) = \int_{\Omega_S} \boldsymbol{\sigma}(\boldsymbol{u})\cdot\nabla v\mathrm{dx}, \tag{18}$$

$$L(v) = \int_{\Omega_S} \boldsymbol{f}\cdot v\mathrm{dx}, \tag{19}$$

where $\boldsymbol{\sigma}(\boldsymbol{u})$ is defined by Equation (8) and contains the unknown u. Ω_S represents the mechanical subdomain.

The obtained variational formulation is interpreted directly into Python syntax. This is a key feature of finite element computer software, and is achieved due to the implementation of the UFL [20].

4. Python Implementation

The Python code is started by including the necessary libraries, such as: dolfin, numpy, mshr, math, mat plot lib, and time [25,27]. The geometry and mesh are generated using GMSH [16]. The data generated by GMSH are imported into the program and subsequently defined as variables. Material properties are defined as Python classes and then, by means of functions that label each subdomain, are assigned accordingly. Individual Python functions are declared for the mechanical and magnetic domain.

The mesh is used as an argument by the function dedicated to finding the magnetic vector potential. The function for the magnetic domain contains: the space shape function, which specifies a third-degree polynomial space [28]; the test and trial functions, which are variables and are applied to the space functions; the variational problem formulation; and the boundary conditions.

The finite element package defaults to linear and preconditioning solutions from the PETSc package [19]. This article focuses on the application of the iterative routines Krylov solver, which is widely used to solve sparse linear systems of non-symmetric equations [17,18]. The commercial software employed for validation solves the problem using the MUMPS direct solver. The finite element program allows customization of the parameters of the Krylov solver, in which the maximum number of iterations is 1000. The current density ($\boldsymbol{J}$) is defined as the Python class. The magnetic field function returns the solution, containing the values for the magnetic vector potential (A_z). Another function is defined to obtain the curl of A_z, in order to find $\boldsymbol{B}$.

The boundary conditions presented in Section 2.1 are defined by calling a class of the finite element package, called DirichletBC. This class sets up strong Dirichlet boundary conditions for partial differential equations. The boundary condition class uses the values on the boundary ($A_z = 0$ and $\boldsymbol{u} = 0$) and its respective subdomain as arguments.

The calculation of $\boldsymbol{B}$ allows the application of the Maxwell stress tensor to find the magnetic forces. The magnetic force is integrated over the boundary of the mechanical domain. This is achieved by means of the object "ds", which belongs to the finite element package, and is described as a function that measures and integrates the outer boundary of a domain. Figure 2 shows a graphical interpretation of the Maxwell stress tensor applied to the mechanical domain. The red arrows represent the force vector perpendicular to the boundary edge. The larger arrows are located in the airgap, where the maximum load is generated. The upper air gap is the cause of the moment of greater magnitude, which in turn is the cause of the maximum displacement.

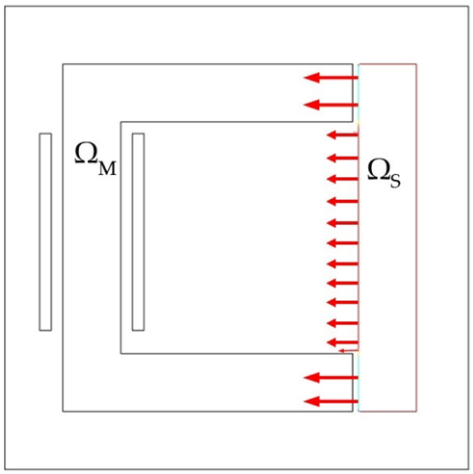

Figure 2. Application of the Maxwell stress tensor on the outer boundary of the domain.

The mechanical domain function uses the mesh, boundary conditions, and the magnetic domain solution as arguments. The spatial shape function is established as a first-degree Lagrangian. The trial and test functions, Lame parameters, and boundary condition values are defined as variables within the mechanical field function. The boundary conditions are set as Dirichlet with a value of zero displacements. The solver is also based on Krylov subspaces and is configured with 1000 maximum iterations. The mechanical field function returns the values of the displacement components (u_x, u_y). The displacement magnitude is found by applying the Euclidean norm.

The nodes of the original mesh are updated when the displacement is obtained, then the new variable obtained is compared with the previous iteration until a previously established tolerance error is reached. The flowchart representing this process is shown in Figure 3.

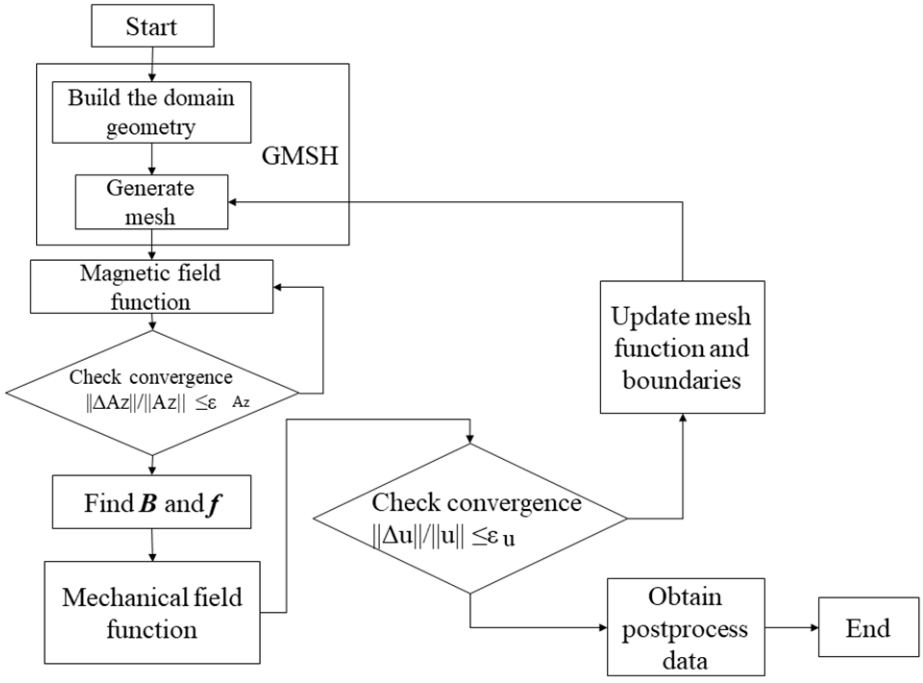

Figure 3. Flowchart for the numerical procedure for solving the weak coupled magnetostatic-mechanical problem.

The simulation results were analyzed using Paraview, an open-source tool for data visualization and post-processing [29].

5. Numerical Simulation Results

The results for $\boldsymbol{B}$, $\boldsymbol{f}$, and $\boldsymbol{u}$ obtained from the simulations performed with the finite element software and with the commercial software are presented in this section.

5.1. Magnetic Field

The magnetic flux density magnitude found by the software is presented in Figure 4; this figure was obtained by projecting B onto a vector space function. In the figure, a white arrow can be seen crossing the air gap parallel to the y-axis. The white arrow represents a probe that extracts the values of B at the position where it is located. Figure 5 shows the comparisons of the values resulting from both programs. An average difference of 1.04% was found.

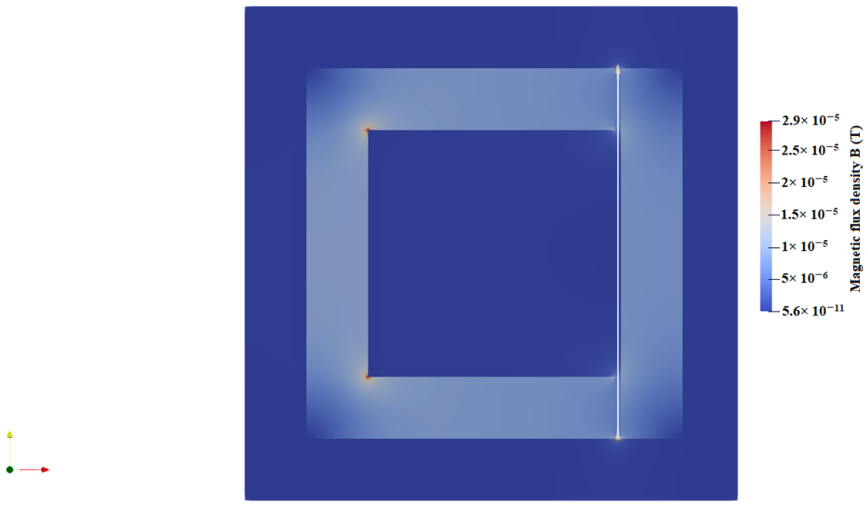

Figure 4. Magnitude of the magnetic flux density obtained by the Python program. The white line represents the probe where the data are extracted.

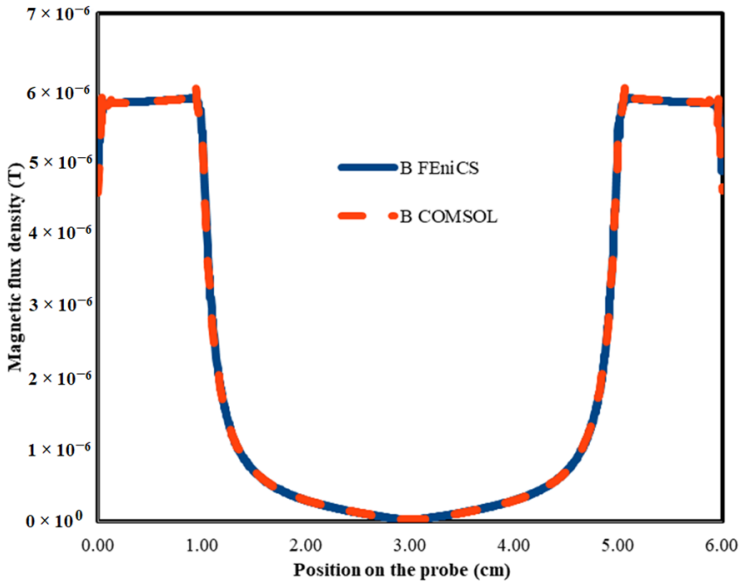

Figure 5. Comparison of the results of B obtained by the Python program proposed versus the commercial software.

As mentioned above, the magnitude of the magnetic force bonds the magnetic domain to the mechanical domain. The x-component of the forces in the air gap is shown in Figure 6. This graph shows the comparison of the forces obtained by the computer program and by the commercial program. Since the magnetic forces depend directly on the magnetic flux density, and their tangential component is close to zero, the values of the y-component of the forces are negligible [30]. The comparison of the resultants of the magnetic forces obtained by both programs is presented in Figure 7, where an average difference of 1.37% is found. The average magnitude of the magnetic force found in the upper air gap by the computer program is 1.529×10^{-5} N/m^2, while it is 1.503×10^{-5} N/m^2 in the commercial program.

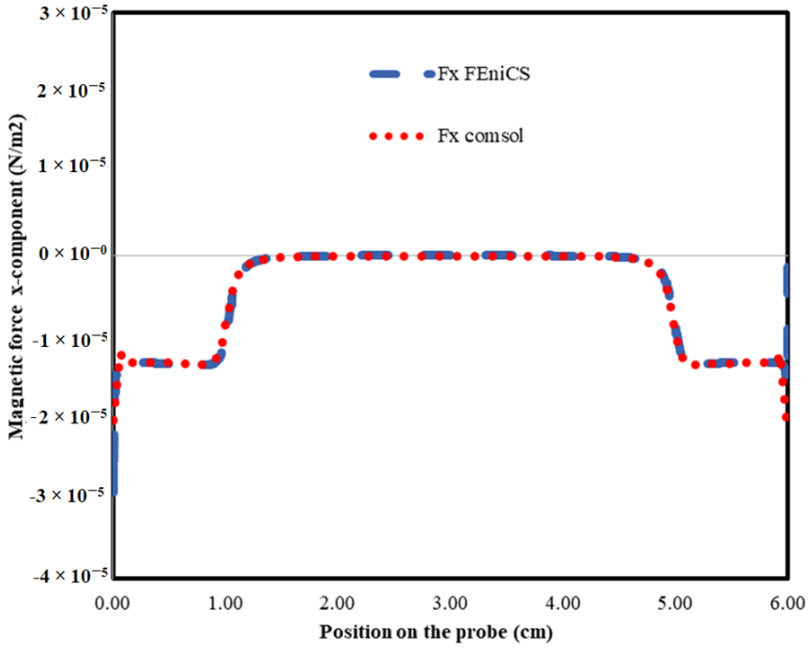

Figure 6. Comparison of the x-component magnetic force versus the commercial software.

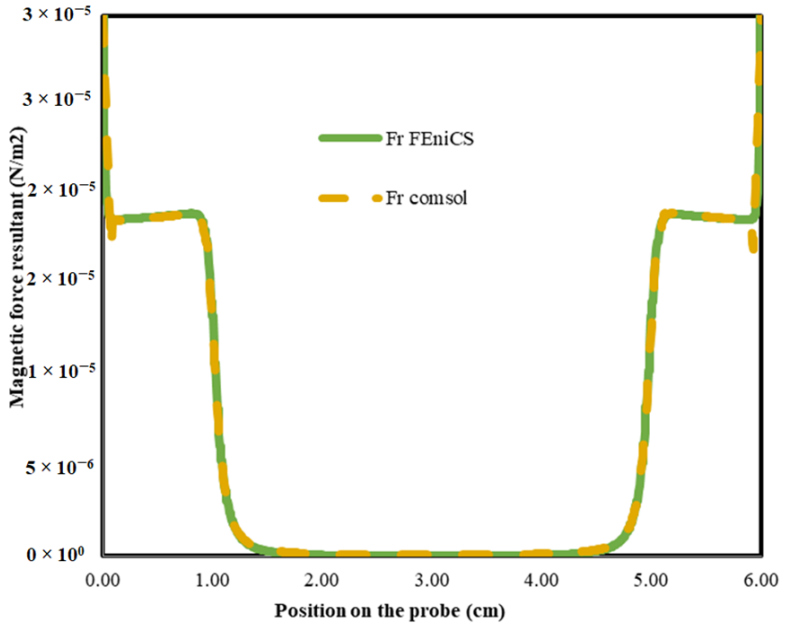

Figure 7. Comparison of the magnetic force resultant on the mechanical domain boundary.

5.2. Mechanical Field

The distribution of the displacements along the actuator is shown in Figures 8–10. Figure 8 shows the x-component of the displacement, it can be seen that the largest contribution to deformation occurs in this direction. The resulting displacement magnitude is shown in Figure 9. Figure 10 represents the spatial distribution of the displacements over the mechanical domain. The white probe extracts values in the center of the domain. The Paraview program performs this function. The maximum displacements found are 4.659×10^{-14} m at the upper edge. The lower edge remains fixed with zero displacement. The actuator subdomain can be interpreted as an isolated system and is analyzed as a separate domain.

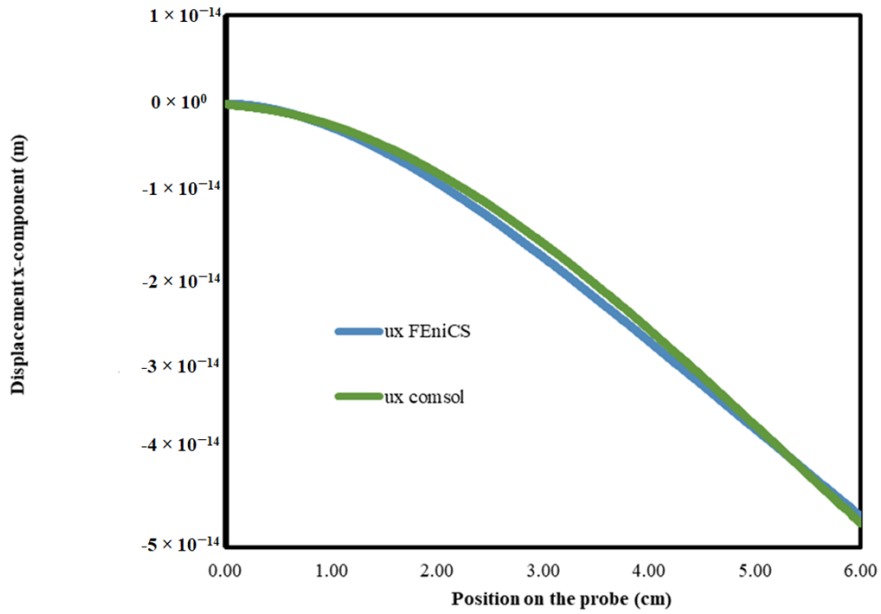

Figure 8. Displacement × component comparison obtained by the Python program and the commercial software.

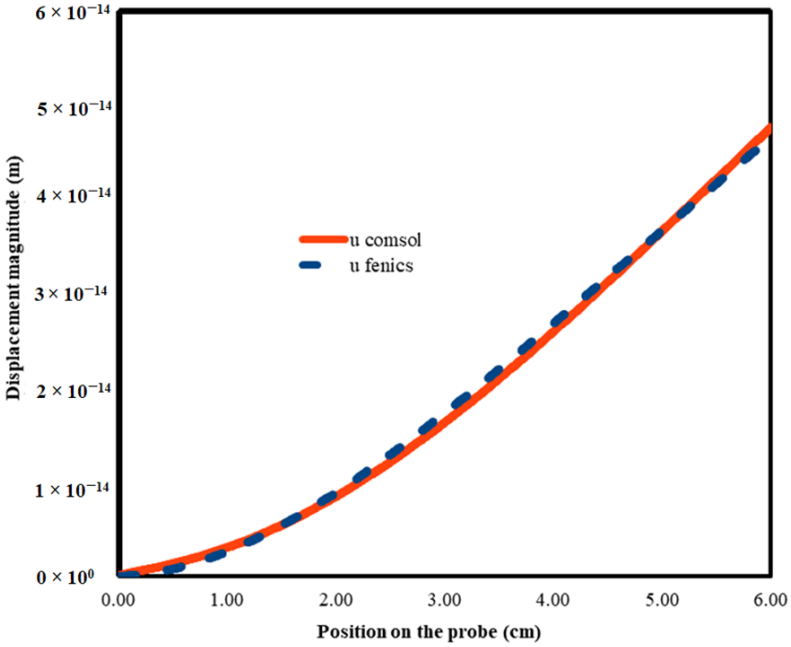

Figure 9. Comparison of the displacements obtained from the Python program and the commercial software.

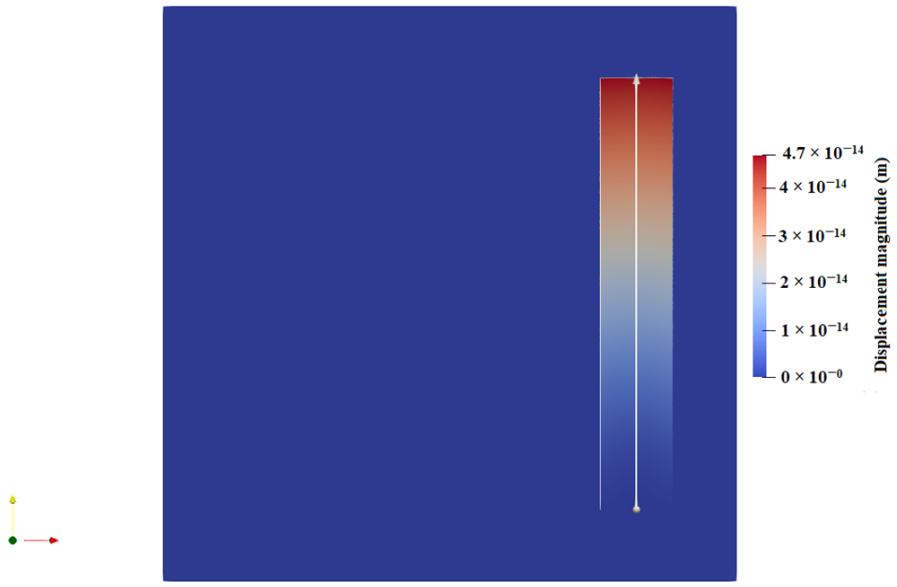

Figure 10. Displacements obtained by the Python code along the mechanical domain.

6. Mesh and Computing Time

The analysis with the finite element method requires the discretization of the problem into small and simple elements, on which a set of equations is solved. The elements comprise the mesh that can be refined by reducing the size of the elements. Mesh refinement is an essential factor for multiphysics modeling [31]. A mesh with smaller elements increases the accuracy of the results; however, additional computing time is required [32]. Thus, a suitable mesh refinement technique must be applied strategically to solve the problem. A mesh refinement for the coupled problem is applied in the area comprising the actuator and the air gaps. Thus, the computational resource is focused on the space where more accurate results are needed, and the simulation time is optimized. The finite elements used by the software are triangles of 2 degrees of freedom, and they are 268,725 elements. The mesh generation performed by GMSH [33] allows the user a quick and simple implementation of geometry discretization. It has a variety of different element types.

The simulations were performed on a computer with AMD Ryzen 5 2600 Six-Core processor and 16 GB of memory. a model developed by Python code can optimize simulation time for engineering problems requiring fine meshes. The solution time for the finite element software was 20.34 s, while for the commercial software, it was 72 s. The process performed by the commercial program required 1248-MB of memory, and the program performed in the Python environment required 320-MB of memory. These differences mean that the Python program has advantages with respect to processing time, memory requirements, meshing, and formulation manipulation.

7. Conclusions

The presented paper proposes a novel methodology to program the Python code with the ability to solve the partial differential equations that describe the magneto-mechanical effect on an electromagnetic device. The deformations were obtained by solving a weak coupled model with the finite element method. This contribution consists of an adaptation of the weak coupling formulation using the Maxwell stress tensor for magneto-mechanical problems. A validation of the results of the Python program against a commercial software was carried out. The validation concluded that the results obtained by the computer program are accurate and with slight differences.

The validation of results clearly suggests that the weak coupled formulation using the Maxwell stress tensor method is suitable for the solution of magneto-static mechanical

problems using the finite element method. Furthermore, the solutions obtained are precise and demonstrated the predicted behavior of the electromagnetic device. The methodology presented has practical applications in mechanical and electrical engineering. This model can be used to calculate deformations in electromagnetic actuators.

Author Contributions: Conceptualization, M.P.-A. and M.A.A.; methodology, M.P.-A.; software, M.P.-A.; validation, C.H., R.E.-P. and M.P.-A.; resources, C.H.; writing—original draft preparation, M.P.-A.; writing—review and editing, R.E.-P., C.H., M.A.A. and M.P.-A.; visualization, M.P.-A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

Acknowledgments: The authors thank the La Laguna Institute of Technology (TNM), in Torreon, Coahuila, Mexico; and CONACYT for the financial support in carrying out this research.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Holopainen, T.P.; Tenhunen, A.; Arkkio, A. Electromechanical Interaction in Rotordynamics of Cage Induction Motors. *J. Sound Vib.* **2005**, *284*, 733–755. [CrossRef]
2. Tüysüz, A.; Breisch, S.; Molter, T. Linear Actuator Utilizing Magnetic Shape Memory Material. In Proceedings of the 2019 12th International Symposium on Linear Drives for Industry Applications (LDIA), Neuchatel, Switzerland, 1–3 July 2019; pp. 1–4.
3. Stephan, J.M.; Pagounis, E.; Laufenberg, M.; Paul, O.; Ruther, P. A Novel Concept for Strain Sensing Based on the Ferromagnetic Shape Memory Alloy NiMnGa. *IEEE Sens. J.* **2011**, *11*, 2683–2689. [CrossRef]
4. Schautzgy, M.; Schnetzler, R.; Laufenberg, M. Ultrafast Actuators with Magnetic Shape Memory Alloys. In Proceedings of the ACTUATOR 2018, 16th International Conference on New Actuators, Bremen, Germany, 25–27 June 2018; pp. 1–4.
5. Wu, Y.; Deng, M. Operator-Based Vibration Control for an L-Type Arm of Crane Systems Using Piezoelectric Actuator. In Proceedings of the 2015 IEEE 12th International Conference on Networking, Sensing and Control, Seattle, DC, USA, 22–25 June 2015; pp. 197–201.
6. Andriollo, M.; Fanton, E.; Tortella, A. A Review of Innovative Electromagnetic Technologies for a Totally Artificial Heart. *Appl. Sci.* **2023**, *13*, 1870. [CrossRef]
7. Delaere, K.; Heylen, W.; Belmans, R.; Hameyer, K. Comparison of Induction Machine Stator Vibration Spectra Induced by Reluctance Forces and Magnetostriction. *IEEE Trans. Magn.* **2002**, *38*, 969–972. [CrossRef]
8. Vandevelde, L.; Melkebeek, J.A.A. Magnetic Forces and Magnetostriction in Electrical Machines and Transformer Cores. *IEEE Trans. Magn.* **2003**, *39*, 1618–1621. [CrossRef]
9. Delaere, K.; Belmans, R.; Hameyer, K.; Heylen, W.; Sas, P. Coupling of Magnetic Analysis and Vibrational Modal Analysis Using Local Forces. In Proceedings of the Xth Int. Symposium on Theoretical Electrical Engineering ISTET, Sofia, Bulgaria, 21–24 July 2019.
10. Dong, Q.; Liu, X.; Qi, H.; Sun, C.; Wang, Y. Analysis and Evaluation of Electromagnetic Vibration and Noise in Permanent Magnet Synchronous Motor with Rotor Step Skewing. *Sci. China Technol. Sci.* **2019**, *62*, 839–848. [CrossRef]
11. Lundström, N.L.P.; Aidanpää, J.-O. Dynamic Consequences of Electromagnetic Pull Due to Deviations in Generator Shape. *J. Sound Vib.* **2007**, *301*, 207–225. [CrossRef]
12. Bastos, J.P.A.; Sadowski, N. *Electromagnetic Modeling by Finite Element Methods*; CRC Press: Boca Raton, FL, USA, 2003; Volume 117, ISBN 978-0-203-91117-4.
13. Kiarasi, F.; Babaei, M.; Asemi, K.; Dimitri, R.; Tornabene, F. Free Vibration Analysis of Thick Annular Functionally Graded Plate Integrated with Piezo-Magneto-Electro-Elastic Layers in a Hygrothermal Environment. *Appl. Sci.* **2022**, *12*, 10682. [CrossRef]
14. Belytschko, T. The Finite Element Method: Linear Static and Dynamic Finite Element Analysis: Thomas J. R. Hughes. *Comput.-Aided Civ. Infrastruct. Eng.* **2008**, *4*, 245–246. [CrossRef]
15. Li, R.; Li, C.; Peng, X.; Wei, W. Electromagnetic Vibration Simulation of a 250-MW Large Hydropower Generator with Rotor Eccentricity and Rotor Deformation. *Energies* **2017**, *10*, 2155. [CrossRef]
16. Langtangen, H.P.; Logg, A. *Solving PDEs in Python*; Springer International Publishing: Cham, Switzerland, 2016; ISBN 978-3-319-52461-0.
17. McDonagh, J.; Palumbo, N.; Cherukunnath, N.; Dimov, N.; Yousif, N. Modelling a Permanent Magnet Synchronous Motor in FEniCSx for Parallel High-Performance Simulations. *Finite Elem. Anal. Des.* **2022**, *204*, 103755. [CrossRef]
18. Logg, A.; Ølgaard, K.B.; Rognes, M.E.; Wells, G.N. FFC: The FEniCS Form Compiler. In *Automated Solution of Differential Equations by the Finite Element Method: The FEniCS Book*; Logg, A., Mardal, K.-A., Wells, G., Eds.; Lecture Notes in Computational Science and Engineering; Springer: Berlin/Heidelberg, Germany, 2012; pp. 227–238. ISBN 978-3-642-23099-8.

19. Habera, M.; Hale, J.S.; Richardson, C.; Ring, J.; Rognes, M.; Sime, N.; Wells, G.N. FEniCSX: A Sustainable Future for the FEniCS Project. In Proceedings of the SIAM PP20 Minisymposium: Improving Productivity and Sustainability for Parallel Computing Software, Seattle, WA, USA, 12–15 February 2020.
20. Alnæs, M.S. UFL: A Finite Element Form Language. In *Automated Solution of Differential Equations by the Finite Element Method: The FEniCS Book*; Logg, A., Mardal, K.-A., Wells, G., Eds.; Lecture Notes in Computational Science and Engineering; Springer: Berlin/Heidelberg, Germany, 2012; pp. 303–338. ISBN 978-3-642-23099-8.
21. Alnæs, M.S.; Logg, A.; Ølgaard, K.B.; Rognes, M.E.; Wells, G.N. Unified Form Language: A Domain-Specific Language for Weak Formulations of Partial Differential Equations. *ACM Trans. Math. Softw.* **2014**, *40*, 1–37. [CrossRef]
22. Metsch, P.; Schiedung, R.; Steinbach, I.; Kästner, M. Benchmark for the Coupled Magneto-Mechanical Boundary Value Problem in Magneto-Active Elastomers. *Materials* **2021**, *14*, 2380. [CrossRef] [PubMed]
23. Feng, K.; Shi, Z.-C. *Mathematical Theory of Elastic Structures*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2013; ISBN 978-3-662-03286-2.
24. Langtangen, H.P. A FEniCS Tutorial. In *Automated Solution of Differential Equations by the Finite Element Method: The FEniCS Book*; Logg, A., Mardal, K.-A., Wells, G., Eds.; Lecture Notes in Computational Science and Engineering; Springer: Berlin/Heidelberg, Germany, 2012; pp. 1–73. ISBN 978-3-642-23099-8.
25. DOLFIN: Automated Finite Element Computing | Semantic Scholar. Available online: https://www.semanticscholar.org/paper/DOLFIN%3A-Automated-finite-element-computing-Logg-Wells/066e817d512bc77109ecd2e2230bfed364ff2ef6 (accessed on 9 December 2022).
26. Langtangen, H.P.; Logg, A. Fundamentals: Solving the Poisson Equation. In *Solving PDEs in Python: The FEniCS Tutorial I*; Langtangen, H.P., Logg, A., Eds.; Simula SpringerBriefs on Computing; Springer International Publishing: Cham, Switzerland, 2016; pp. 11–35. ISBN 978-3-319-52462-7.
27. Logg, A.; Wells, G.N.; Hake, J. DOLFIN: A C++/Python Finite Element Library. In *Automated Solution of Differential Equations by the Finite Element Method: The FEniCS Book*; Logg, A., Mardal, K.-A., Wells, G., Eds.; Lecture Notes in Computational Science and Engineering; Springer: Berlin/Heidelberg, Germany, 2012; pp. 173–225. ISBN 978-3-642-23099-8.
28. Arnold, D.; Logg, A. Periodic Table of the Finite Elements. *Siam News* **2014**, *47*, 212.
29. Ahrens, J.; Geveci, B.; Law, C. ParaView: An End-User Tool for Large Data Visualization. *Vis. Handb.* **2005**, *717*, 50038-1.
30. *The Mechatronics Handbook*; Bishop, R.H.; ISA—The Instrumentation, Systems, and Automation Society (Eds.) The electrical engineering handbook series; CRC Press: Boca Raton, FL, USA, 2002; ISBN 978-0-8493-0066-0.
31. Abali, B.E.; Queiruga, A.F. Theory and Computation of Electromagnetic Fields and Thermomechanical Structure Interaction for Systems Undergoing Large Deformations. *J. Comput. Phys.* **2019**, *394*, 200–231. [CrossRef]
32. Bathe, K.-J. *Finite Element Procedures*; Bathe, K.J., Ed.; Prentice Hall: Upper Saddle River, NJ, USA, 2006; ISBN 978-0-9790049-0-2.
33. Gmsh: A Three-Dimensional Finite Element Mesh Generator with Built-in Pre- and Post-Processing Facilities. Available online: https://gmsh.info/#References (accessed on 9 December 2022).