COMPUTACIONAL IMPLEMENTATION OF THE BOUNDARY ELEMENT METHOD APPLIED TO POTENTIAL FLOWS

Diego L. Cutrim[1*], Lívia M.N. Costa[1] and André P. Santana[1]

[1] *Department of Mechanics and Materials - DMM, Federal Institute of Education, Science and Technology of Maranhão (IFMA), São Luís, MA, Brazil.*
cutrimd@acad.ifma.edu.br

ABSTRACT

The present work was carried out with the objective of implementing the boundary element method in a free tool with good applicability within the general field of engineering, highlighting the study of potential flows. For this purpose, Python was chosen for meeting the necessary requirements and for being the most widely used programming language in the world today. The procedure for formulating the method was conducted straightforwardly to present the proposed problem and to emphasize the case arising from the solution, namely the singularity. Comparisons were made regarding the program's processing time and the solution's convergence as a function of the mesh used. Within the addressed topic, the advantages and disadvantages of the chosen tool were highlighted, demonstrating its relevance for innovation in a field of study that is still little explored but has great potential for expansion into various applications.

Keywords: boundary element method, Laplace, python.

INTRODUCTION

Computational analysis is, nowadays, well established in various fields, especially in engineering, due to cost and time reduction, which makes the so-called domain methods essential for solving structural, heat transfer, fluid dynamics problems, etc. The boundary element method (BEM) is one of the most recently developed methods, with its first classical theory of integral equations studied by the mathematician Fredholm (1903), and later, Rizzo (1967) presented the first direct formulation for integral equations. This method is still not as consolidated when compared to, for example, the finite element method (FEM), which includes some commercial tools. However, FEM requires high computational capacity since it solves the problem over the entire domain, while in BEM the discretization is done only along the boundary. This type of approach significantly reduces computational cost and also offers ease of modeling for infinite domains and when modifying the problem mesh.

However, the application of BEM requires the establishment of the integral representation of the solution, in order to deal only with linear problems whose solution can be determined. Problems related to potential fluid flows, heat transfer, and torsion can have their solutions developed through the Laplace equation. Therefore, the application of the boundary element method becomes very feasible, with the treatment carried out through a boundary integral equation derived from the governing equation of the problem. The representation of a typical flow is shown in Figure 1.

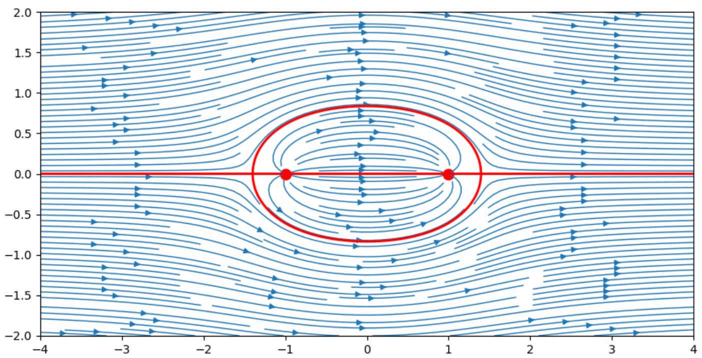

Figure 1: Potential flow of fluid over an elliptical body.

MATERIALS AND METHODS

The development is carried out based on the Laplace equation, Eq. (1), which describes the potential problem. The solution is developed for the values of ϕ and $\partial\phi/\partial n$, with the procedure resulting in unknowns located only on the boundary.

$$\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} = 0 \tag{1}$$

Using the Galerkin method, a function ϕ^*, which is an approximation of the solution, is multiplied on both sides of Eq. (1), and the integration is performed over the domain Ω.

$$\int_\Omega \left(\frac{\partial^2 \Phi}{\partial x^2} + \frac{\partial^2 \Phi}{\partial y^2} \right) \Phi^* \, d\Omega = 0 \tag{2}$$

Performing the integration by parts twice, Gauss's Theorem is used in order to obtain:

$$\int_\Omega \left(\frac{\partial^2 \Phi^*}{\partial x^2} + \frac{\partial^2 \Phi^*}{\partial y^2} \right) \Phi \, d\Omega - \int_\Gamma \Phi \frac{\partial \Phi^*}{\partial n} \, d\Gamma + \int_\Gamma \Phi^* \frac{\partial \Phi}{\partial n} \, d\Gamma = 0 \tag{3}$$

With the transfer property, which makes the integration of the product of the Dirac delta, applied to Eq. (2), and another arbitrary function produce the value of the function at the so-called pole, which is the point $X_i = (x_0, y_0)$ (for more details, see SANTANA, 2021), the development is carried out in order to obtain:

$$\Phi_i + \int_\Gamma \Phi \frac{\partial \Phi^*}{\partial n}\, d\Gamma - \int_\Gamma \Phi^* \frac{\partial \Phi}{\partial n}\, d\Gamma = 0 \tag{4}$$

The form applied to the set of established functions provides a system that allows the determination of the unknown values within it, so that the boundary values of the problem become known. Shape functions, as in other methods, are used to improve the discretization of the elements, and simplification can be made using constant elements. The fundamental solution for ϕ^* is shown in Eq. (5), as well as its derivative $\partial\phi^*/\partial n$ in Eq. (6).

$$\phi^* = \frac{1}{2\pi} \ln(1/r) \tag{5}$$

$$\frac{\partial \phi^*}{\partial n} = \frac{1}{2\pi r^2}[(x_c - x_0)n_x + (y_c - y_0)n_y] \tag{6}$$

Where,

$$r = \sqrt{(x_c - x_0)^2 + (y_c - y_0)^2} \tag{7}$$

It's worth mentioning that the singularity treated through numerical integration tends toward a linearity of the solution, provided that the approximation is made at an appropriate level, bringing results that are consistent with the imposed theory. The tendency of the solution can be seen in the representation shown in Figure 2.

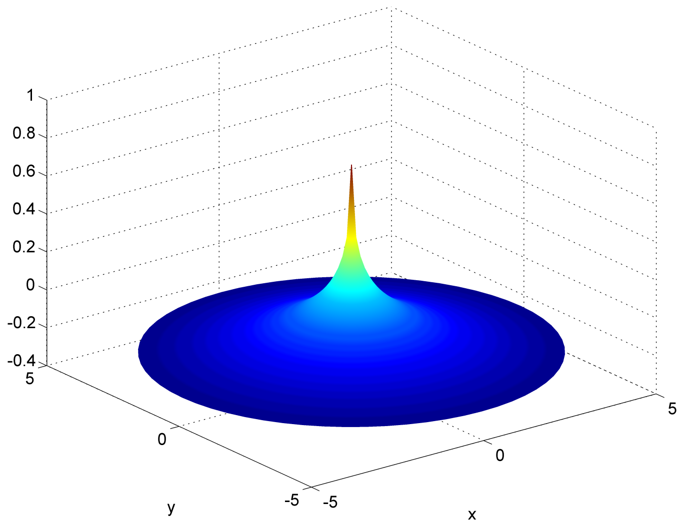

Figure 2: Fundamental solution $1/2\pi \ln(1/r)$.

RESULTS AND DISCUSSION

From the Python implementation, it' s possible to analyze the convergence process of the solution based on the number of Gauss points used for numerical integration. The comparison between the graphs formed by the number of points used and the value of the solution as a function of the number of divisions by element is shown in Figure 3.

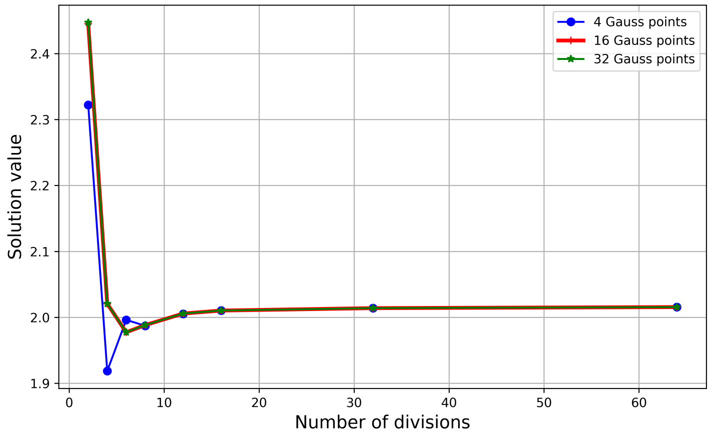

Figure 3: Solution for different number of Gauss points.

A comparison regarding the program's processing time is necessary to highlight the pros and cons of using Python. The curve shown in Figure 4 emphasizes the importance of program optimization, since compared to software like MATLAB, which is entirely dedicated to numerical calculations, Python still shows a certain disadvantage in this aspect. On the other hand, with better use of its functions, Python's interactivity and ease of use can be considered some of its positive points.

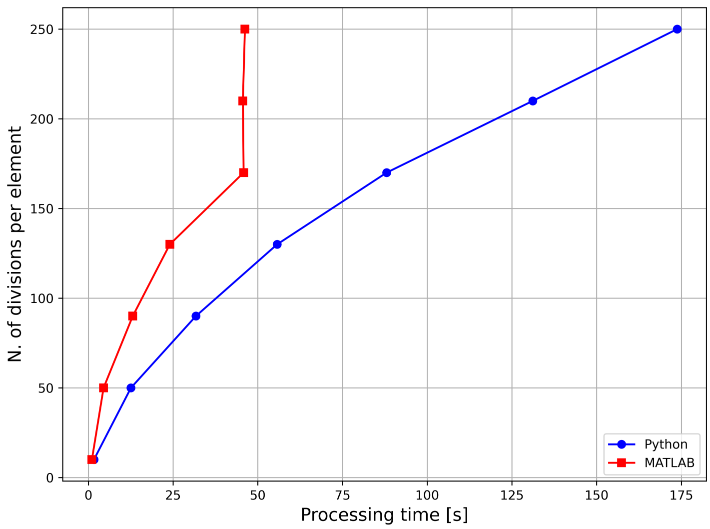

Figure 4: Processing analysis: Python vs MATLAB.

For an application subject to zero flow conditions on the walls of the geometry and unitary flow, a section similar to that of a diffuser was simulated, making it possible to evaluate the gradient generated within it, with the representation of internal points distributed throughout the geometry, as shown in Figure 5.

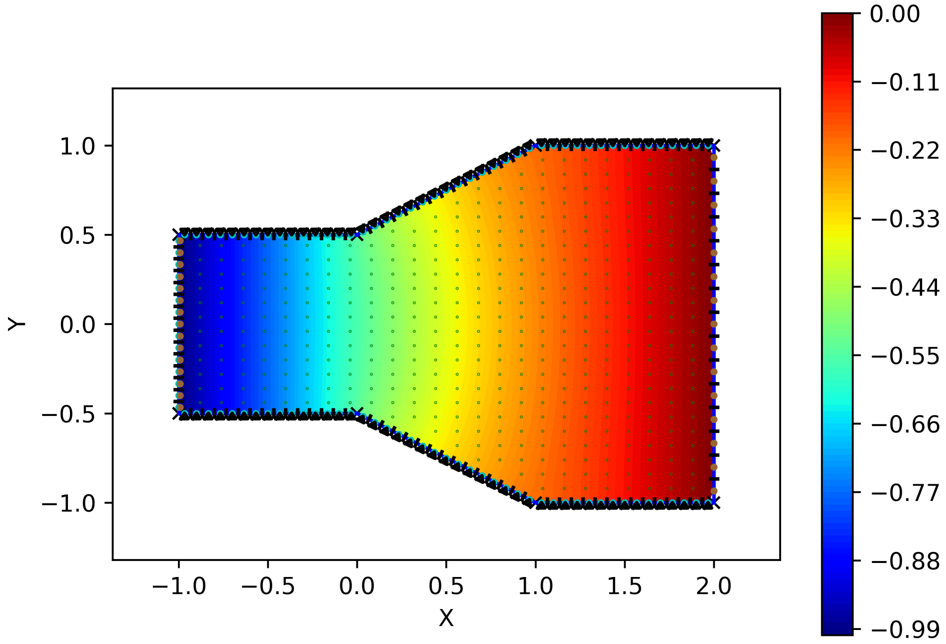

Figure 5: Speed gradient in geometry with section change.

CONCLUSIONS

The computational implementation of methods such as BEM should be approached in a way that correctly utilizes the tools provided — in this case, by Python — which offers a wide range of features, such as a dynamic interface and access to libraries for analyzing the results obtained. The proposed procedure, considering all the aspects discussed, proves to be effective and provides a well-structured foundation for future optimizations. From the present work, the study of two-dimensional potential flows can be further developed, opening possibilities for expansion into structural applications related to torsion in two-dimensional and three-dimensional beams.

REFERENCES

[1] BEER, Gernot; SMITH, Ian; DUENSER, Christian. The boundary element method with programming: for engineers and scientists. Springer Science & Business Media, 2008.
[2] BREBBIA, Carlos Alberto; TELLES, José Claudio Faria; WROBEL, Luiz C. Boundary element techniques: theory and applications in engineering. Springer Science & Business Media, 2012.

[3] CHAPRA, Steven C.; CANALE, Raymond P. Métodos Numéricos para Engenharia-7ª Ediçao. McGraw Hill Brasil, 2016.
[4] FAJARDO, Caetano Soares. Solução numérica bidimensional da equação de Laplace para o escoamento ao redor de um cilindro circular. Doctoral Thesis. Federal Center for Technological Education Celso Suckow da Fonseca, 2015.
[5] FRIEDMAN, Z.; KOSMATKA, J. B. Torsion and flexure of a prismatic isotropic beam using the boundary element method. Computers & Structures, v. 74, p. 479–494, 2000.
[6] HESS, J. L.; SMITH, A. M. O. Calculation of potential flow about arbitrary bodies. Progress in aeronautical sciences, v. 8, p. 1-138, 1967.
[7] KANE, J. A. Boundary Element Analisys in Engineering Continuum Mechanics. Prentice Hall, New Jersey, 1994.
[8] KATSIKADELIS, John T. The Boundary Element Method for Engineers and Scientists: Theory and Applications. Second edition. Academic Press, Elsevier, 2016.
[9] MILLMAN, K. Jarrod; AIVAZIS, Michael. Python for scientists and engineers. Computing in science & engineering, v. 13, n. 2, p. 9-12, 2011.
[10] MUNSON, Bruce Roy et al. Fluid mechanics. Singapore: Wiley, 2013.
[11] SANTANA, André P. Método dos elementos de contorno: Fundamentos e aplicações. First edition. Maranhão: Aluízio de Freitas Carvalho, 2021.
[12] YUNUS, A. Cengel. Fluid Mechanics: Fundamentals And Applications (Si Units). Tata McGraw Hill Education Private Limited, 2010.