VETOR

# A Computational Tool to Determine the Forces Generated by Waves on Piles☆

## Ferramenta Computacional para Determinação de Forças Geradas por Ondas em Estacas

Adilson José Pereira Junior[1†], Márcio Wrague Moura[1]

[1] *Escola de Engenharia, Universidade Federal do Rio Grande, Rio Grande, Brasil*
[†] **Corresponding author:** adilson.ajpj@gmail.com

**Abstract**

This paper presents the development of a computational software tool for the determination of the forces generated by the passage of waves on piles, with the objective of disseminating instructive knowledge, or assisting in the design process of such structure, commonly found in port regions. This tool is built through the use of a free, open-source and easy-to-assimilate computational language. To make this software more intuitive, a graphical interface is designed to illustrate the results obtained. To obtain these results, the present work is based on the methodology adopted by Mason [1], which makes use of Morison's formula that is commonly present in the literature, and also adopted by the Spanish standard (ROM2.0-11). This work differs from the methodology adopted by Mason in what concerns the determination of the incident wave length on the pile. Here, a numerical solver is used for the solution of the dispersion relation, avoiding the need to verify the wave propagation regime. The results found reinforce the importance of using computational tools in the study or practice of engineering, providing greater precision and agility in performing complex calculations.

**Keywords**

Wave on Pile • Morison Equation • Free Software • Python

**Resumo**

Este trabalho apresenta o desenvolvimento de uma ferramenta de software computacional para determinação dos esforços gerados pela passagem de ondas em estacas, com o objetivo de disseminar o conhecimento didático, ou auxiliar o processo de dimensionamento da referida estrutura, comumente encontrada em regiões portuárias. Esta ferramenta é construída através da utilização de uma linguagem computacional gratuita, de código fonte aberto e de fácil assimilação. Para tornar este software mais intuitivo, é elaborada uma interface gráfica, a fim de ilustrar os resultados obtidos. Para obter esses resultados, o presente trabalho se baseia na metodologia adotada por Mason [1], que faz uso da fórmula de Morison, comumente presente na literatura, sendo também adotada pela norma espanhol de engenharia (ROM2.0-11). Este trabalho difere da metodologia adotada por Mason no que diz respeito à determinação do comprimento de onda incidente na estaca. Aqui, um solucionador numérico é utilizado para a solução da relação de dispersão, evitando a necessidade de verificar o regime de propagação das ondas. Os resultados encontrados reforçam a importância da utilização de ferramentas computacionais no estudo ou prática de engenharia, proporcionando maior precisão e agilidade na realização de cálculos complexos.

**Palavras-chave**

Ondas em Estacas • Equação de Morison • Software Livre • Python

☆This article is an extended version of the work presented at the Joint XXV ENMC National Meeting on Computational Modeling, XIII ECTM Meeting on Science and Technology of Materials, 9th MCSul South Conference on Computational Modeling and IX SEMENGO Seminar and Workshop on Ocean Engineering, held in webinar mode, from October 19th to 21th, 2022

# 1 Introduction

The utilization of computers and software is increasingly common in the routine of engineers, scientists, and students. Engineering software facilitates calculations, analysis, and studies of various problems and specific situations in the most varied fields. A significant portion of these software programs is privately owned. Although many companies with proprietary rights provide student licenses for free, most of these tools come with a high acquisition cost. The exorbitant cost of proprietary software often results in reduced utilization or, alternatively, unauthorized use.

An effective solution to this predicament entails the growing adoption of free and open-source software tools. Free software, for clarifying reasons, refers to programs that provide users with the freedom to access, modify, and distribute the source code. This open access to information promotes collaboration through modifications and enhancements to tackle diverse engineering problems [2]. Therefore, it logically follows that the usage of free software and tools has garnered increased attention from the academic community [3].

A noteworthy example to highlight among these software tools is the Python programming language [4], which has become one of the most widely used programming languages in current times. Its relevance stems from the multitude of scientific and mathematical frameworks that are indispensable for implementation in engineering practices. The language is open-source, user-friendly, and possesses a simplified syntax, making it easy to comprehend and work with.

The present study concerns the development of computational techniques employed for the automation of calculating the forces exerted by waves on piles. As piles are commonly utilized in port areas that are susceptible to wave action [1], the creation of such tools is of significant importance in assisting students or professionals who intend to design these structures. Consequently, this paper aims to foster the democratization of knowledge, given the insufficiency of informative and accessible tools in the application area of port engineering. This objective is set to be achieved through the development of open-source software, which can be openly accessed, shared, scrutinized, and enhanced without any constraints.

# 2 Methodology

Since the present work does not aim at the development or verification of a new method for the determination of the forces caused by waves on piles, its methodology is founded on the application of concepts already established in scientific literature. Thus, this paper shall address the fundamental equations and procedures utilized for the development of the software.

## 2.1 Development Environment

To develop the proposed tool, the first step involved setting up the development environment. To achieve this, the Anaconda distribution (a package of tools for scientific computing) [5] and the Python programming language [4] were utilized. As Python does not provide a built-in library for the manipulation and resolution of equations, the installation of the SymPy [6] library was performed. This library provides a collection of functions and operators for resolving operations that involve symbolic and numeric mathematics.

## 2.2 Wave Parameters

According to the linear wave theory, a progressive wave can be characterized by its period $T$, wavelength $L$ (i.e., the distance between two successive crests), height $H$ (i.e., the distance between the trough and crest), and shape $\eta$. Figure 1 provides an illustration of the characteristics of a progressive wave. Equation (1) describes the shape of the wave as a function of the abscissa $x$ and time $t$, where $\sigma$ represents the angular frequency of the wave (i.e., $\sigma = 2 \cdot \pi / T$) and $k$ represents the wave number (i.e., $k = 2 \cdot \pi / L$):

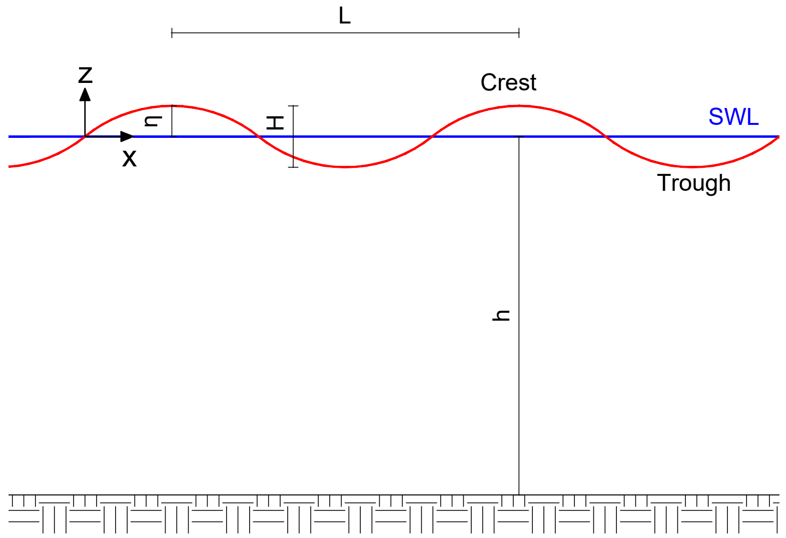


Figure 1: Progressive wave parameters.

$$\eta(t) = \frac{H}{2} \cdot \cos(kx - \sigma t). \tag{1}$$

To obtain the necessary parameters for computing the forces exerted by waves, the wave dispersion equation can be utilized [7]. By applying this equation, the wavelength $L$ can be determined from the wave period $T$ and the depth of the location $h$. The wave dispersion equation, Eq. (2) is presented below, where $g$ represents the acceleration due to gravity:

$$\sigma^2 = g \cdot k \cdot \tanh\left(k \cdot h\right). \tag{2}$$

To obtain the wavelength by analytical solution for Eq. (2) we arrive at Eq. (3), which has no closed-form solution. As a result, the solution requires an iterative process that involves computing the equation repeatedly until both sides converge to the same value. In order to perform this computation, a mathematical solver is employed, which uses the method of secants. The solver utilized in this study is provided by the SymPy library:

$$L = \frac{g \cdot T^2}{2 \cdot \pi} \cdot \tanh\left(\frac{2 \cdot \pi \cdot h}{L}\right). \tag{3}$$

The use of a numerical solver to obtain the wavelength allows for a more precise calculation of the forces exerted by waves on piles without the need to check the boundary conditions for wave depth, such as whether the wave is in shallow, intermediate, or deep waters. The solver also simplifies the data entry process for the software. Once the parameters are obtained, the calculation of the forces exerted by waves on piles can begin.

## 2.3 Morison's Formula

To determine the forces acting on a pile caused by waves, the Morison equation can be utilized [8]. This equation is widely used in literature and is even incorporated in the Spanish engineering standard (ROM 2.0-11) [9] for designing marine or port structures.

The Morison equation characterizes the forces exerted on a pile through two terms. The first term, presented in Eq. (4), deals with the drag force caused by fluid movement. The second term deals with the acceleration effects due to wave motion. Two coefficients, $CD$ and $CM$ which are experimentally obtained, are used to adjust the experimentally acquired results. In addition to these coefficients, other variables in Eq. (4) include the fluid density, $\rho$, the pile diameter, $D$, and the horizontal component of the velocity of the fluid particles, $u$. Morison's formula, Eq. (4), is presented below:

$$F = C_D \cdot \frac{1}{2} \cdot \rho \cdot D \cdot |u| \cdot u - C_M \cdot \rho \cdot \frac{\pi \cdot D^2}{4} \cdot \frac{du}{dt} \tag{4}$$

The terms can be separated and treated individually. The horizontal component of the particle velocity during the passage of the wave, $u$, will depend on the wave theory used. In this work, the linear theory, which was also adopted by Mason [1], was used. Therefore, the equations that describe this component and its derivative with respect to time are presented in sequence, as Eq. (5) and Eq. (6):

$$u = \frac{H}{2} \cdot \frac{g \cdot T}{L} \cdot \frac{\cosh\left[2 \cdot \pi \cdot (h+z)/L\right]}{\cosh\left(2 \cdot \pi \cdot h/L\right)} \cdot \cos\left(\frac{2 \cdot \pi \cdot t}{T}\right), \tag{5}$$

$$\frac{\mathrm{d}u}{\mathrm{d}t} \simeq \frac{\partial u}{\partial t} = \frac{\pi \cdot g \cdot H}{L} \cdot \frac{\cosh\left[2 \cdot \pi \cdot (h+z)/L\right]}{\cosh\left(2 \cdot \pi \cdot h/L\right)} \cdot \sin\left(-\frac{2 \cdot \pi \cdot t}{T}\right). \tag{6}$$

After performing the substitutions of the appropriate components in Eq. (4), and manipulating algebraically, we have, respectively in Eq. (7) and Eq. (8), the drag and inertial forces per unit length:

$$F_D = C_D \cdot \frac{1}{2} \cdot \rho \cdot g \cdot D \cdot H^2 \left[\frac{g \cdot T^2}{4 \cdot L^2} \cdot \left(\frac{\cosh(k \cdot (z+h))}{\cosh(k \cdot h)}\right)^2\right] \cdot \left|\cos\left(\frac{2 \cdot \pi \cdot t}{T}\right)\right| \cdot \cos\left(\frac{2 \cdot \pi \cdot t}{T}\right), \tag{7}$$

$$F_M = C_M \cdot \rho \cdot g \cdot \frac{\pi \cdot D^2}{4} \cdot H \cdot \left[\frac{\pi}{L} \cdot \frac{\cosh(k \cdot (z+h))}{\cosh(k \cdot h)}\right] \cdot \sin\left(-\frac{2 \cdot \pi \cdot t}{T}\right). \tag{8}$$

To obtain the maximum force caused by the drag force, one can consider the maximum values for the cosine terms, i.e., when $t$ is zero, at which point the wave crest is in contact with the pile. Thus, Eq. (9) is obtained. This equation and Fig. 2 (to illustrate this situation) are presented in sequence:

$$F_D = C_D \cdot \frac{1}{2} \cdot \rho \cdot g \cdot D \cdot H^2 \left[\frac{g \cdot T^2}{4 \cdot L^2} \cdot \left(\frac{\cosh(k \cdot (z+h))}{\cosh(k \cdot h)}\right)^2\right]. \tag{9}$$

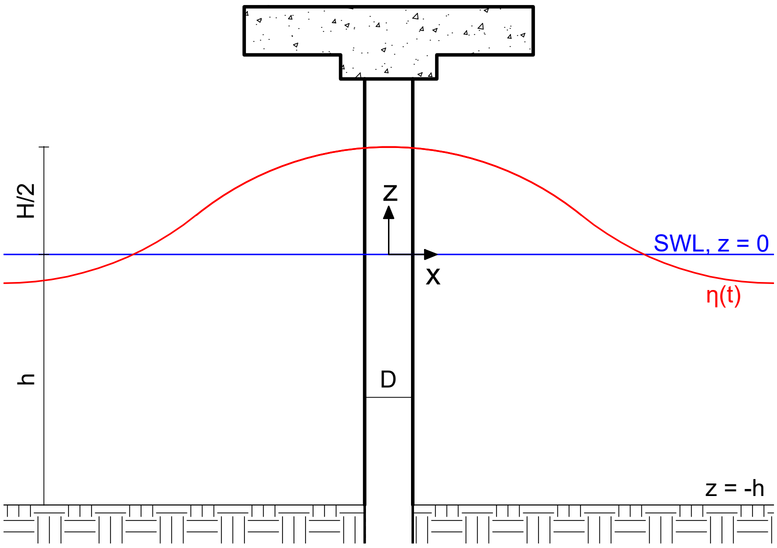


Figure 2: Wave on a pile scheme.

To obtain the maximum force caused by the inertial force, one can consider the maximum value for the sine term, that is, when $t$ is negative by a quarter of the wave period $(-T/4)$. There we have Eq. (10):

$$F_M = C_M \cdot \rho \cdot g \cdot \frac{\pi \cdot D^2}{4} \cdot H \cdot \left[\frac{\pi}{L} \cdot \frac{\cosh(k \cdot (z+h))}{\cosh(k \cdot h)}\right]. \tag{10}$$

To better illustrate the responses of $F_D$ and $F_M$, Fig. 3 is presented containing the approximate profile of the load distribution along the pile. The profile is similar for both components (disregarding the shape and position of the wave). In Figure 3, the pile diameter has been omitted.

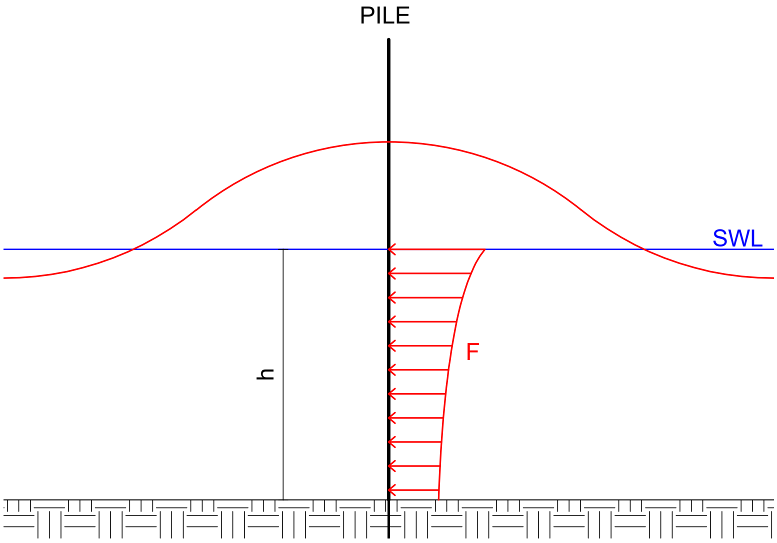


Figure 3: Wave load profile at the pile scheme.

## 2.4 Net Forces

Since Eq. (9) and Eq. (10) give us the force per unit length of the pile, the net force can be obtained by integration. Meanwhile, for integration, the method presented by Mason was adopted, disregarding the wave surface. In other words, the forces are computed from the still water level (SWL) downward. These relations are valid for $D/L < 0.05$. Therefore, we have Eq. (11) and Eq. (12):

$$F_{D_{net}} = C_D \cdot \rho \cdot g \cdot D \cdot H^2 \cdot \frac{1}{8} \cdot \left(1 + \frac{2 \cdot k \cdot h}{\sinh(2 \cdot k \cdot h)}\right), \tag{11}$$

$$F_{M_{net}} = C_M \cdot \rho \cdot g \cdot H \cdot \frac{\pi \cdot D^2}{8} \cdot \tanh(k \cdot h). \tag{12}$$

The application points of these forces, distant $d_{drag}$ and $d_{inertia}$ with respect to the ground, are used for the determination of the bending moments, which are commonly applied during the design of these structures and can be determined approximately by the equations Eq. (13) and Eq. (14). After that, Fig. 4 is presented.

$$d_{drag} = h \cdot \left( \frac{1}{2} + \frac{1}{\left(1 + \frac{2 \cdot k \cdot h}{\sinh(2 \cdot k \cdot h)}\right)} \cdot \left(\frac{1}{2} + \frac{1 - \cosh(2 \cdot k \cdot h)}{2 \cdot k \cdot h \cdot \sinh(2 \cdot k \cdot h)}\right) \right), \tag{13}$$

$$d_{inertia} = h \cdot \left(1 + \frac{1 - \cosh(k \cdot h)}{k \cdot h \cdot \sinh(k \cdot h)}\right). \tag{14}$$

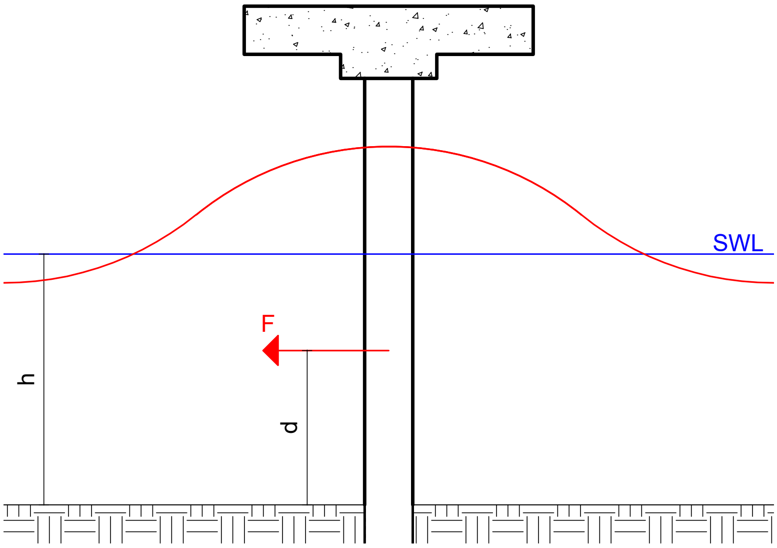


Figure 4: Net force on pile scheme.

### 2.5 Graphical User Interface

After developing solvers for the equations described earlier, a graphical interface was constructed to facilitate data input and subsequent result visualization. For this purpose, the tkinter library, which is naturally present in the Python language, was used. The result obtained is presented in Fig. 5. To enhance the user experience, an export function was implemented to enable the results to be saved in a spreadsheet format. The Pandas framework, which is designed to handle tabular data, was utilized for this purpose. The exported data consists of a spreadsheet that contains information about the position of the pile, as well as the maximum drag and inertia forces.

Once the graphical user interface was completed, the necessary logic for displaying the elements on the screen was developed to make the tool user-friendly and intuitive. Subsequently, tests were conducted using the solved examples proposed by Mason [1] and Morison et al. [8], to verify the accuracy of the software's results.

## 3 Results and Discussion

The software requires only seven input parameters to solve the problem, among them the depth of the site (m), the drag and inertia coefficients (dimensionless), the density of the fluid (kg/m$^3$), the diameter of the pile (m), the period of the incident wave (s) and the height of this wave (m). The final graphical user interface is presented in Fig. 5.

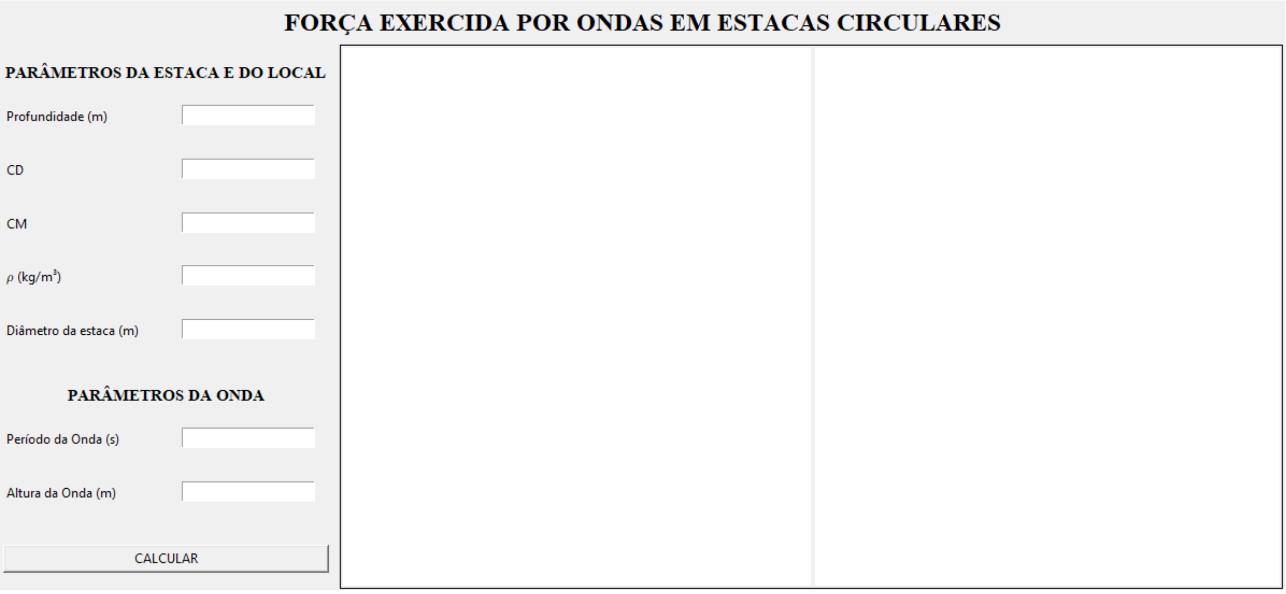


Figure 5: Graphical user interface presented by the software (labels, titles, and parameters in Portuguese).

In order to verify the results of the software, tests were carried out with the resolution of the example presented by Mason [1] and the problem proposed by Morison et al. [8]. Figures 6 and 7 present the results obtained for the maximum possible forces, while Table 1 and Table 2 depict the comparison between the solutions.

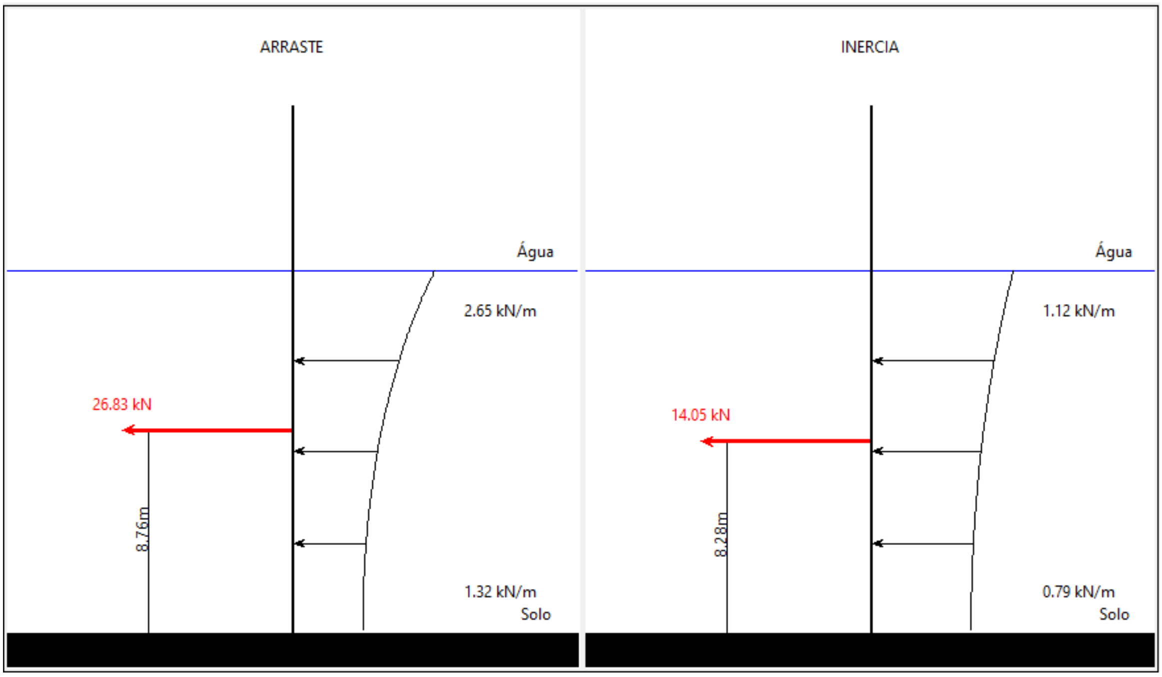


Figure 6: Software graphical result from Mason's example (legends in Portuguese).

Table 1: Comparison between the software result and Mason's example.

| Result | $T$ (s) | $L$ (m) | $H$ (m) | $h$ (m) | $F_D$ (kN) | $F_M$ (kN) |
|---|---|---|---|---|---|---|
| Mason [1] | 10 | 115.71 | 5.60 | 15.61 | 26.09 | 13.73 |
| Software | 10 | 110.74 | 5.60 | 15.61 | 26.83 | 14.05 |
| Relative Error | – | 4.30% | – | – | 2.84% | 2.33% |

It can be observed from Table 1 that the numerical solver used for the dispersion relation found a smaller value for $L$ (by approximately 5 m) compared to that determined by Mason [1], with an error margin of 4.30%. Regarding the resultant forces, a higher value of 0.8 kN was obtained for drag and 0.3 kN for inertia. For both forces, the error margin was below 3%.

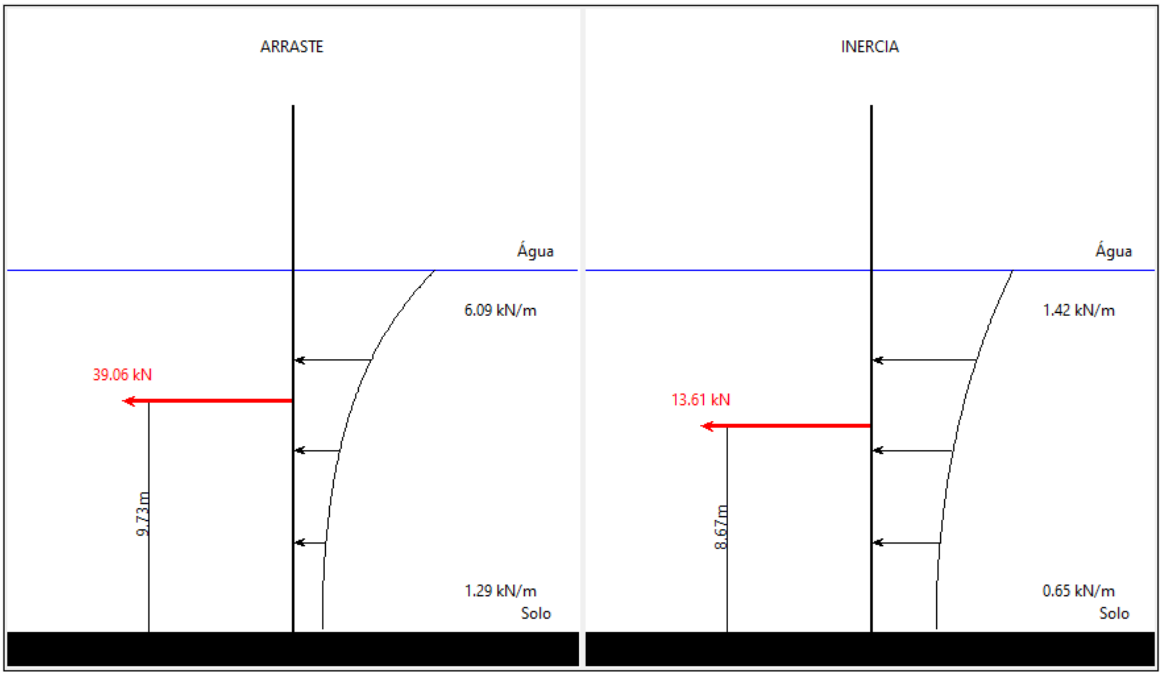


Figure 7: Software graphical result for Morison's example (legends in Portuguese).

Table 2: Comparison between the software result and Morison's example.

| Result | $T$ (s) | $L$ (m) | $H$ (m) | $h$ (m) | $M_D$ (kN·m) |
|---|---|---|---|---|---|
| Morison et al. [8] | 7 | 68.58 | 6.096 | 15.24 | 397.26 |
| Software | 7 | 67.90 | 6.096 | 15.24 | 380.05 |
| Relative Error | – | 0.99% | – | – | 4.33% |

The example proposed by Morison et al. [8] did not show the resultant inertia and drag forces, but it presented the maximum moment ($M_D$) in the structure. The units used were converted, and the results were calculated. The maximum moment was then compared, which presented a relative error of 4.33%, and the difference between the calculated wavelength was practically zero, close to 0.99%.

The results obtained were satisfactory, as they presented values close to those found in the examples proposed by Mason [1] and Morison et al. [8]. It is worth noting that when solving such exercises, Mason does not make direct use of the dispersion relation for waves, probably due to the impossibility of its analytical solution. Here, the power provided by the computational tools is highlighted, especially the solver used, which does not require the need to verify propagation in deep, intermediate, or shallow waters. Such verifications, when performed manually, increase the time needed to solve the problem, as well as the possibility of error propagation.

## 4 Conclusions

The findings of this study indicate that the absence of numerical integrations to determine resultant forces and points of application limits the accuracy of the current results. Numerical integrations play a crucial role in engineering practice, and their implementation would lead to greater precision. Therefore, future works should focus on incorporating these techniques to enhance the accuracy of the calculations and improve the overall reliability of the software.

Regarding the values of the resultant forces found in Table 1, it can be observed that the software yields higher bending forces. When adopted by structural designers or students, these values would enhance structural safety, constituting another fundamental aspect of engineering practice.

As a consequence, with the development of this work, the dissemination of knowledge is expected, making academic life easier and contributing to the professional development of individuals who make use of the material. Furthermore, since the programming language used in the development of this tool is free and open -source (Python) and its code is hosted in a collaborative system such as GitHub (github.com/adilsonjpj), the maintenance and eventual improvement of the adopted solutions is facilitated.

## References

[1] J. Mason, *Obras Portuárias*, 1st ed. Rio de Janeiro, Brasil: Editora Campus, 1982, in Portuguese.

[2] S. A. dos Santos, M. N. Carvalho, D. Barbosa, R. J. P. de Araújo, and K. F. da Silva, "Desenvolvimento de uma ferramenta computacional para aplicação num estudo de qualidade da energia elétrica associada a geração solar fotovoltaica," *Anais do Simpósio Brasileiro de Sistema Elétricos*, 2021, in Portuguese. Available at: https://www.sba.org.br/open_journal_systems/index.php/sbse/article/view/2343/1996

[3] A. Azevedo, L. M. Michelini, D. N. Pinto, J. R. G. LIMA, and G. A. V. D. Menezes, "Softwares livres no ensino de engenharia," *Anais do Congresso Nacional Universidade, EAD e Software Livre*, 2011, in Portuguese. Available at: http://www.periodicos.letras.ufmg.br/index.php/ueadsl/article/download/2713/2666

[4] G. V. Rossum and F. L. Drake, *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009.

[5] ANACONDA, *Anaconda Software Distribution*, 2020. Available at: https://docs.anaconda.com/

[6] A. Meurer and C. P. Smith, "SymPy: symbolic computing in Python," *PeerJ Computer Science*, vol. 3, p. e103, 2017. Available at: https://doi.org/10.7717/peerj-cs.103

[7] R. G. Dean and R. A. Dalrymple, *Water Wave Mechanics for Engineers and Scientists*, 2nd ed. Singapore: World Scientific Publishing Company, 1991.

[8] J. R. Morison, J. W. Johnson, and S. A. Schaaf, "The Force Exerted by Surface Waves on Piles," *Journal of Petroleum Technology*, vol. 2, no. 05, pp. 149–154, 1950. Available at: https://doi.org/10.2118/950149-G

[9] ROM 2.0-11, *Recomendaciones para el proyecto y ejecución en Obras de Atraque y Amarre*, ser. Recomendaciones para Obras Marítimas. Serie 2, Obras portuárias interiores. Puertos del Estado, 2012, in Spanish.