2FRAME eDU: WEB SOFTWARE FOR GEOMETRIC NON-LINEAR STRUCTURAL ANALYSIS OF PLANE FRAMES

NUNES, J.F. de L.[1], BARBIRATO, J.C.C.[2]

[1]*Center of Technology, University of Alagoas - UFAL*
fyllipy@gmail.com
[2]*Center of Technology, University of Alagoas - UFAL*
Av. Lourival Melo Mota, S/N, Tabuleiro do Martins, Maceió - AL, Cep: 57072-970, Alagoas, Brasil
jccb@ctec.ufal.br

Abstract. This work developed a software for nonlinear geometric structural analysis of plane frames. Thus, it seeks to use matrix methods, more precisely the Direct Stiffness Method, the Two Cycle Iterative Method for the geometric nonlinearity and the theory of solid mechanics to calculate from a frame modeled by the user, determining its stiffness matrix from the material parameters entered and its internal forces. In this context, the application will be available for undergraduate and graduate student as for structural professionals in needing of nonlinear analysis via web in the cloud computing model, using the React.JS framework for the front-end side. Its results were compared against data obtained from solved examples in the literature.

Keywords: *Structural Analysis, Matrix Analysis of Structures, Nonlinear Geometric Analysis, Engineering App Development, Cloud Computing*.

1 Introduction

Structural analysis is one of the main activities in the design of civil and mechanical structures. The goal is to determine the forces and stresses acting on the structure and its response to applied loads. With technological advancements, the use of computational tools has become common in this field, enabling designers and engineers to develop more efficient and safer structures. This research aimed to develop a web-based software for the structural analysis of plane frames, considering the geometric nonlinearity of the structure.

In structural analysis, it is crucial for engineers and both undergraduate and graduate students to visualize the behavior of the structure under applied loads. Although there are software programs such as SAP2000, Robot, and the free Ftool, there is always a demand for software that allows for quick and practical nonlinear structural analysis of frames. This development contributes to structural analysis by enabling the design of more efficient and safer structures, considering geometric nonlinear behavior. Furthermore, it opens future possibilities, such as the addition of a design module for steel structures according to NBR8800:2008 [1].

In this context, a web-based software (cloud computing) was developed for geometric nonlinear structural analysis using the Direct Stiffness Method. The software allows user interaction (frontend) using React.JS and the JavaScript programming language, providing graphical visualization of the bar elements in plane frames. Ultimately, it aims to contribute to the dissemination of free-to-use software among undergraduate and graduate students, as well as professors of structural courses, especially those that include second-order analysis in their curricula.

2 Structural analysis formulation

2.1 Direct Stiffness Method

The strain of a material maintains a direct relationship with the applied stress, proportional to an elastic constant E, within the elastic-linear deformation regime. The Direct Stiffness Method employs this strategy by calculating the internal forces in each bar of the structural system, as shown in eq. (1). Through the Principle of Superposition of Effects, it represents the overall behavior of the structure Kassimali [2].

$$\{F\} = [k]\{d\} \tag{1}$$

The matrix [k] contains the so-called stiffness coefficients k of each element and has an order of 3n, where n is the number of nodes. These are proportionality coefficients used to relate unit displacements with actual displacements.

Equation (1) describes the displacement method. Each displacement is calculated at the nodes of the structure, and in the plane case, each node has 3 degrees of freedom and 3 axes of force action. In other words, a structure consists of 3n actions, and an equal number of degrees of freedom.

2.2 Geometric nonlinear analysis

Structural analysis aims to rigorously understand how structures respond to external forces. It is necessary to recognize that there are situations where the structure does not behave according to Hooke's law, i.e., following a linear stress-strain relationship (physical nonlinearity). Additionally, excessive lateral displacements can generate bending moments that were not initially anticipated (geometric nonlinearity). These situations occur in tall buildings, elements with cracks, and material inconsistencies, among others. This analysis is crucial for a better understanding of the structural system's behavior Pereira [3].

Nonlinear analysis is also known as second-order analysis. Due to the complex mathematical processes of rigorous methods, simplified methods are preferred. Despite their simplicity, these methods show good accuracy in the literature. One such method, described by Moura et al. [4], analyzes the changes in Internal Force Demands (IFD) caused by displacements due to external forces. A fictitious horizontal load is applied to displace the structure from its initial condition and analyze the load behavior in this new configuration. This process is repeated iteratively until the displaced structure reaches equilibrium (Silva et al. [5]).

The importance of these simplified methodologies is highlighted by NBR 8800 [1], which presents a similar simplified approach called the Amplification Method of Internal Force Demands (MAES) or Method B1, B2, to reduce the designer's workload. There are methods that implicitly consider the deformed structure in their formulations, such as the Two Iterative Cycles Method (M2CI). This work considers only geometric nonlinearity.

Second-order effects have a significant impact on tall structures, increasing with the number of floors. The variation of displacements shows a linear relationship with the building's height (Konapure et al. [6]). To reduce the computational load in calculating second-order forces, simplified methodologies can be adopted. In this work, the approach of the Two Iterative Cycles Method, proposed by Chen et al. [7], was chosen due to its reduced computational effort.

The first cycle is performed by a linear analysis of the structure. After, in the second calculation cycle, the stiffness matrix [K_L] is added to the geometric stiffness matrix [K_g], resulting in an updated stiffness matrix, as shown in eq. (2). Finally, a conventional calculation is performed as if it were a linear analysis, using this updated stiffness matrix.

$$[K] = [K_L] + [K_g] \tag{2}$$

Within this method, it is possible to use an even more complex geometric stiffness matrix [k_g], incorporating other internal forces besides the axial force, as suggested by Rodrigues [8]. However, as observed in the obtained results, the mathematical framework presented by Chen et al. [7] provided satisfactory results.

In summary, the Two Iterative Cycles Method is applied as follows:

- First Cycle:
  - Step 1: Perform a linear analysis of the structure;
- Second Cycle:
  - Step 2: Use the forces obtained in Step 1 to calculate the geometric stiffness matrix [Kg];
  - Step 3: Add [Kg] to [KL] to generate the updated stiffness matrix;
  - Step 4: Conduct a new analysis using the updated stiffness matrix.

3 On the software 2frame

The software resulting from this work was designed to meet the needs of engineers and engineering students for quick, easy, and functional calculation of planar truss structures. The user bears full responsibility for the use and interpretation of the results, indemnifying the developer from any implications. The program's function is to serve as a tool for reference rather than the final calculation of the model, requiring the student, researcher, or engineer to review and validate the results. The program is named 2Frame.

Among the various programming options, JavaScript stood out as it is the de facto syntax for web development. However, it needs to be paired with HTML markup language to build web pages. Thus, ECMAScript 6 version of JavaScript will be used.

With the advancement of mobile devices and internet network structure, application developers have turned to web development. This enables the existence of cloud applications, stored on a server and accessible via a web browser from any internet-connected device. This production model in system development is called cloud computing. This software architecture can be understood by analogy to a rope with knots at both ends. One knot allows the user to enter, receive data, and view the application, typically with a more user-friendly UX (User Experience), known as frontend, while the other knot, known as backend, processes requests, performs calculations, among other tasks, to centralize processing and alleviate computational load on the client side. All this communication is facilitated through the "rope," which in this analogy is the internet itself. This approach allows users to interact with the application anytime, anywhere, requiring only internet access. Application architecture involves organizing its different modules, functions, and data in a logical order. Its primary function is to facilitate code development and maintenance. The application resulting from this work is divided into 4 major modules: Post-processor, Processor, Pre-processor, and Canvas.

The Canvas module stores all the logic for the user interface. Therefore, it manages the other modules by exchanging data and supervising user requests. The Pre-processor handles user input data. It converts graphical input into arrays of coordinates and forces, as well as handles geometric and material data manipulation. To solve the matrix system, the processing module, managed by Canvas, receives the data processed by the Processor and performs the necessary calculations. After solving the structure model, results are managed and organized by the post-processor, sending processed results for display by Canvas. All data processing takes advantage of JavaScript's object-oriented nature. Thus, nodes and bars are instantiated as objects, simplifying their manipulation across different modules without the need for complex functions and methods.

To gain greater control over drawing functions, no external graphical framework was used, opting instead to create a proprietary 2D engine named Object Display on Canvas Engine (ODCE). This decision was motivated by the author's desire for greater control over the application's graphical aspects and to avoid dependency on external licenses. A graphic engine is a set of functions that generate graphics, in this case 2D, transforming logical instructions into linear equations to display the necessary shapes on screen. The 2D engine developed here is rudimentary, utilizing JavaScript's canvas methods. It includes a simple meter-based vector space and a scaling transformation. One feature of ODCE is the ability to scale drawings and adjust grid line spacing.

The user interface was designed with the following guidelines in mind: user experience (UX), simplicity, and lightweight design. These guidelines were followed wherever possible, resulting in the standard application interface (Figure 1), which is divided into 4 main areas: sidebar, bottom bar, drawing board, and pop-ups (or windows).

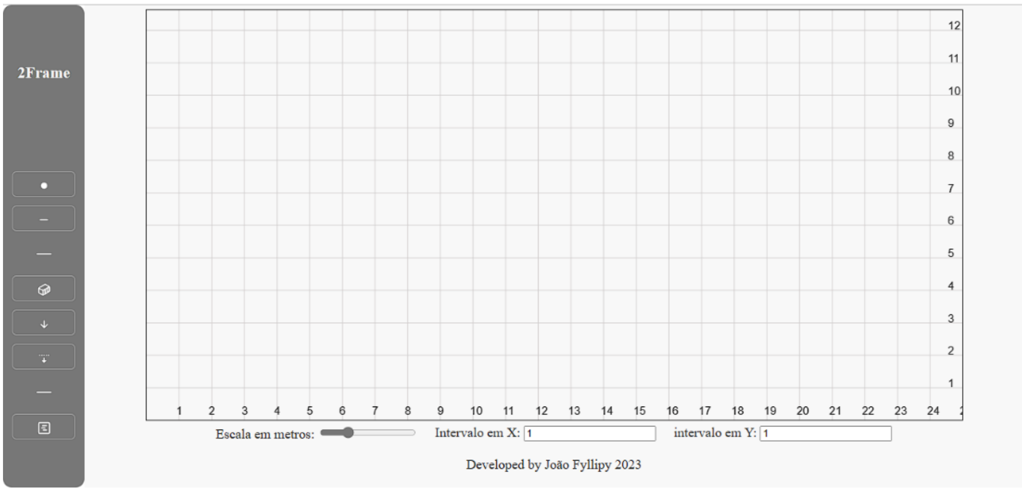


Figure 1 – Overall Application Interface
Source: Authors

The Table 1 presents the function of each of these areas, referred to here as UI (User Interface) elements.

Table 1 – User Interface Elements

Element	Description
Sidebar	Stores tools for data input, such as node insertion, bar insertion, insertion of geometric and material properties, force insertion, and the review and analysis tool.
Bottom Bar	Controls the display of the drawing area, such as grid spacing and scale.
Drawing Area	Area composed of a grid where the user draws the structure using nodes and bars.
Windows	Pop-ups that appear when selecting certain tools with a form for manipulation.

Source: Authors

The concept of windows was adopted (Figure 2) to facilitate user interaction with data, functioning as assistants by displaying data entry fields, their descriptions, and units.

The review and analysis window also introduces the concept of tabs, where each element of the structure can be reviewed in its respective tab. It is also in this window that the analysis and results section is located (shown only after analysis has been performed). The results section provides a comprehensive report, including reaction tables and internal force diagrams for each bar (Figure 3).

Reações

Nó	X	Y	Mz
1	-10.00 kN	50.00 kN	0 kN.m
2	-10.00 kN	50.00 kN	-0.00 kN.m

Deslocamentos

Nó	u	v	Theta
1	0.00000 m	0.00000 m	-0.13810 rad
2	0.00000 m	0.00000 m	0.13810 rad

Esforços Internos Solicitantes

Barra	N1	V1	M1	N2	V2	M2
1	0.00 kN	50.99 kN	-0.00 kN.m	0.00 kN	-50.99 kN	-0.00 kN.m

Figure 3 – Tables from the analysis report of a generic example
Source: Authors

Lista de Nós   Lista de Barras   Análise

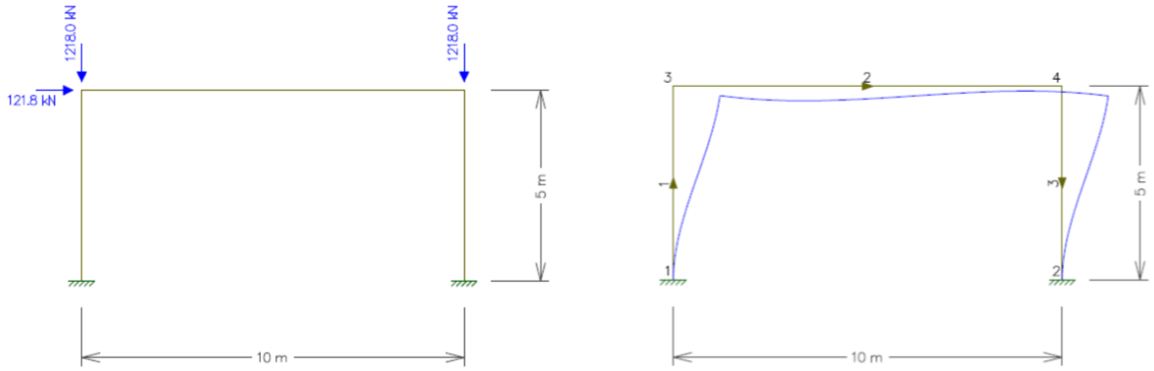
Nó	Apoio	x (m)	y (m)	deletar
1	Livre	7	3	X
2	Livre	10	5	X
3	Livre	6	6	X
4	Livre	16	6	X
5	Livre	17	3	X
6	Livre	13	6	X

Figure 2 – Example of windows in the application
Source: Authors

A distinctive feature of this work is its online usability facilitation. To achieve this, the form and hosting method for deploying the program were carefully chosen. In computing terms, hosting means "allocating web space for a page," which involves putting the application server online. Hosting can be done internally by the product itself (in this case, software with an in-house server implementation). However, for convenience and cost-effectiveness, it is generally preferable to use hosting services. At the time of developing this work, the program is hosted on a free host, subject to the policies and clauses of the service provider. This proved to be the main challenge during the program's development because functionalities originally intended for 2Frame had to be excluded from the web due to issues related to the limitations of the free service. Therefore, the author of this work cannot guarantee its continuous presence online, with only the source code remaining under their control.

4 Application

The portal developed by Silva et al. [5] was selected for processing in 2FRAME and its implementation. It is a single-story, planar portal frame with no bracing, as shown in Fig. 4(a). This is a metallic frame with the following parameters: longitudinal elastic modulus, E = 205,000 MPa; and Poisson's ratio, ν = 0.3. The frame consists of two columns, each 5 meters in height, and a beam with a clear span of 10 meters (columns CS 400x137 and beam CVS 500x217). The force values are as per the authors, also shown in Fig. 4(a). The deformed shape of the structure is presented in Fig. 4(b). Silva et al. [5] solved this portal using a linear approach and softwares (Robot and STAAD), as presented in the Table 2.



(a) (b)

Figure 4. Planar Portal Frame: (a) Structure definition and applied forces (1st increment); (b) Deformed shape and node numbering.
Source: Silva et al. [5]

The results obtained using 2FRAME are also presented in Table 2, compared with the results provided by Silva et al. [5] and those resolved using SAP2000. The accuracy of the formulation used and the implementation performed are verified.

Table 2. Results obtained in 2FRAME and those from Silva et al. [5] plus.

Analysis	Node 3		Element 3		Element 1		Node 1	
	Displacement (m)		Normal Force (kN)		Shear Force (kN)		Bending Moment(kN.m)	
	Silva	2FRAME	Silva	2FRAME	Silva	2FRAME	Silva	2FRAME
Linear	0.12440	0.12446	18,666.5	18,666.47	917.79	917.78	2,599.20	2,599.19
P-Delta Robot	0.28182		19,153.0		969.90		5,325.18	
P-Delta STAAD	0.28188	0.28199	19,153.0	19,153.44	969.92	939.33	5,325.75	5,312.45
*NL Sap2000	*0.2614		*19,107.6		*960.92		*5,193.06	

5 Conclusions

The development of the web software, named 2Frame, was detailed, describing its functionalities and features available to users, as well as highlighting the main advantages and challenges faced during development. Comparative analyses were then conducted between the results obtained by the developed software and the other academic softwares, using as benchmark those results found by Silva et al. [5]. The results demonstrated that 2Frame yielded satisfactory outcomes, comparable to other calculation methods and softwares. Additionally, it was possible to emphasize the software's advantage in terms of ease of use, accessibility, and convenience, as it can be accessed from any internet-connected device. In conclusion, the developed software is deemed a valuable tool for professionals in the field and undergraduate students, contributing to the understanding of structural behavior in both linear and nonlinear analyses.

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] ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBR 8800: Projeto de estruturas de aço e de estruturas mistas de aço e concreto de edifícios. Rio de Janeiro, 2008. 247 p.
[2] KASSIMALI, A. Matrix Analysis of Structures. 2. ed. [S.l.]: Cengage, 2010
[3] PEREIRA, A. Projeto ótimo de pórticos planos com restrição à flambagem. 99 p. Dissertação (Mestrado em Engenharia Civil) — Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2002.
[4] MOURA, B. C. de A.; BARBIRATO, J. C. C. Contribuição para análise não linear de pórticos planos de edifícios utilizando o p-delta com a técnica das cargas laterais fictícias. XLI CILAMCE, 2020.
[5] SILVA, R. G. L. da; LAVALL, A. C. C. Utilização de análises elásticas estimadas ou aproximadas de 2ª ordem para o dimensionamento de pórticos planos de aço. XXXI Jornadas Sud-Americanas de Ingeniería Estructural, 2004.
[6] KONAPURE, C. G.; DHANSHETTI, P. V. Effect of p-delta action on multi-storey buildings. International Journal of Engineering Research & Technology, v. 4, 2015.
[7] CHEN, W.-F.; LUI, E. M. Stability Design of Steel Frames. [S.l.]: CRC Press, 1991.
[8] RODRIGUES, M. Soluções integradas para as formulações do problema de não linearidade geométrica. Tese (Doutorado) — Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2017.