



25th ABCM International Congress of Mechanical Engineering
October 20-25, 2019, Uberlândia, MG, Brazil

COB-2019-1377

IMPLEMENTATION OF A USER-CONTROLLED STRUCTURAL ANALYSIS MODULE WITH GEOMETRIC NONLINEARITY

Rafael Lopez Rangel

Luiz Fernando Martha

Department of Civil and Environmental Engineering, Pontifical Catholic University of Rio de Janeiro (PUC-Rio)
Rua Marquês de São Vicente 225 – Gávea, Rio de Janeiro, RJ, 22451-900, Brazil
rafaelrangel@tecgraf.puc-rio.br
lfm@tecgraf.puc-rio.br

Abstract. *This work presents the development and use of a graphical tool to analyze planar framed structure models with the consideration of large displacements and large rotations in the elastic regime of the material behavior. The main objective in the development of this application is to give users the possibility to control the progress of the nonlinear analysis by changing its parameters as the analysis goes forward or backwards. The focus of this controlled analysis is to provide students, engineers, and researchers a better knowledge on the behavior of reticulated structures with geometric nonlinearity and the use of numerical methods to trace equilibrium paths. This nonlinear analysis module was implemented in the Ftool program, a software largely used by the structural analysis community. The results show that the change in the analysis parameters can influence the response of the structure and, therefore, ways of controlling the analysis are necessary.*

Keywords: *structural analysis, geometric nonlinearity, second-order effects, educational tool.*

1. INTRODUCTION

When a structural system is subjected to deflections that are relatively large when compared to the dimensions of its components, the consideration of large displacements and large rotations are required in the formulation of finite element equations. This consideration is necessary to impose the equilibrium of the structural system in its deformed configuration, so the nonlinear response of the structure's geometry can be taken into account. Otherwise, if the equilibrium and kinematic equations of the finite element method were formulated in the undeformed configuration, second-order effects, such as unexpected internal forces and buckling, would not be predicted and could lead the structure to fail.

The identification of second-order effects can be done by studying the history of the equilibrium of the structure with a certain control variable. Hence, one of the main objectives of a geometrically nonlinear analysis is to obtain the equilibrium path of the structural system. This path is traced by an incremental-iterative process, where the nonlinear system must be linearized and solved in a series of iterations within each analysis step, until equilibrium is satisfied according to a convergence criterion. The convergence points represent an equilibrium state of the structure.

Depending on the severity of the nonlinearities and the complexity of the equilibrium path, the problem may present some critical points that can lead to difficulties in tracing the entire path. Many methods and strategies have been developed to overcome these difficulties and efficiently follow the equilibrium path. Sophisticated methods that compute increments of both control variables (load and displacements) are called continuation methods. However, one single method may not be capable of solving any general nonlinear problem, and modifications to the solution algorithms may be necessary to recover the entire equilibrium path. Therefore, as stated by Bergan *et al.* (1978), a computer program for nonlinear analysis should possess several alternative algorithms for the solution of the nonlinear system of equations. These procedures should also allow for the possibility of an extensive control over the solution process by parameters that are input to the analysis.

This work describes the development and use of a geometrically nonlinear analysis tool for planar frame models, considering large displacements and large rotations, but small deformations in the elastic regime of the material behavior. This tool was incorporated as a new feature of the Ftool (Two-dimensional Frame Analysis Tool) program (Martha, 1999). It provides users with a wide range of analysis options and parameters, including the most well-known incremental-iterative methods to solve the nonlinear system of equilibrium equations. It is also possible to perform the analysis process in an interactive-adaptive fashion, by allowing the change of any of these options and parameters between the steps of the analysis. Furthermore, a sophisticated graph-plotting environment was developed to show the equilibrium path, or the behavior of other variables inherent to the analysis.

2. NONLINEAR FINITE ELEMENT FORMULATION

2.1 Kinematic description

In a geometrically nonlinear analysis, a body subjected to external loads assumes different configurations as it moves in space and changes shape. In the context of the continuum mechanics, there are two ways to describe the movement of the body, the *Eulerian* description and the *Lagrangian* description (Malvern, 1969). The latter is the most appropriate for solid mechanics, because it maps the trajectory of all particles of the body, using material coordinates, from the beginning of the movement to the end. To formulate the finite element equations of the body, a reference configuration must be established for all kinematic and static variables. Three formulations, based on the kinematic description, are mainly used to develop the nonlinear system of equations, depending on the reference configuration adopted within the *Lagrangian* description of the movement. They are the *Total Lagrangian* (TL), the *Updated Lagrangian* (UL), and the *Co-Rotational* (CR) formulations. According to Bathe (1996), the only advantage of using one formulation rather than the other lies in its greater numerical efficiency. In this work, the UL and the CR formulations are used, due to their advantages for beam elements with large displacements and small deformations, when compared to the TL formulation, as shown by Bathe and Bolourchi (1979) and stated by Felippa (2017).

In the UL formulation, the reference configuration is periodically updated to the last achieved equilibrium configuration. That is, once the equilibrium is reached, all the static and kinematic variables of the analysis are defined according to the new configuration. In the CR formulation, the reference configuration is divided in two so that rigid body displacements are separated from those that generate deformations. The initial configuration is used to measure rigid body movements, while a co-rotated configuration is used to measure deformations and stresses of the body. Figure 1 illustrates this concept. Throughout this work, the configuration corresponding to the incremental step $i-1$ is the last obtained equilibrium configuration, while the one corresponding to step i is the current configuration, still unknown, in which equilibrium is being sought.

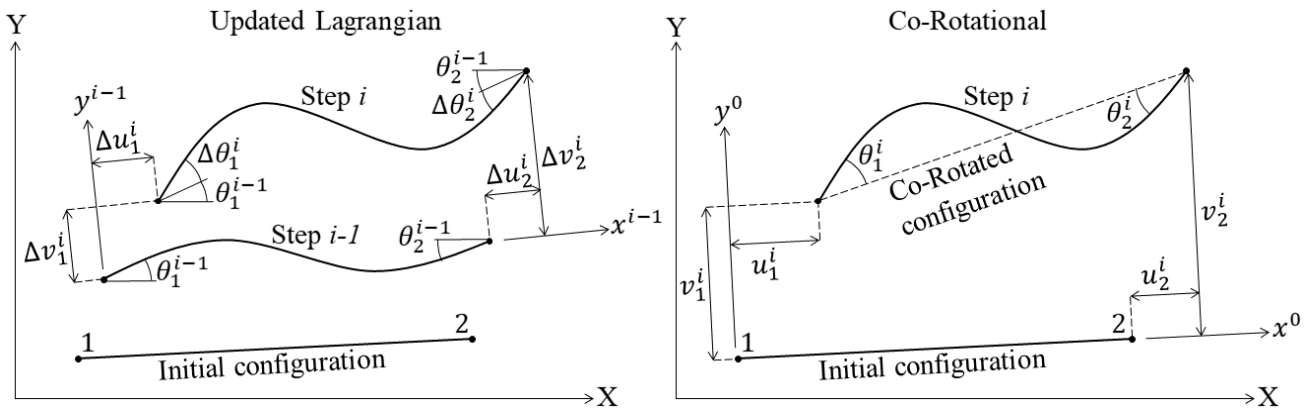


Figure 1. Kinematic descriptions for geometrically nonlinear formulation of beam elements.

2.2 Nonlinear system of equilibrium equations

Common to any kinematic description used to formulate the nonlinear structural problem is that the system of nonlinear equations, which defines the global equilibrium state, considering a finite element discretization, is given by the balance of internal and external forces at nodal points. This balance is expressed in Eq. (1), where \mathbf{F} is the vector of internal nodal forces, which is a function of the nodal displacements vector, \mathbf{u} , and \mathbf{P} is the vector of external nodal forces applied to the structure.

$$\mathbf{F}(\mathbf{u}) = \mathbf{P} \quad (1)$$

The solution of this system is obtained incrementally. For a sequence of external force increments, $\Delta \mathbf{P}_i$, the corresponding increments of nodal displacements, $\Delta \mathbf{u}_i$, are calculated by linearizing the problem, where subscript i indicates the i -th analysis step. The total external forces and nodal displacements of current configuration (step i) are then computed by adding the incremental updates to the previous configuration (step $i-1$):

$$\mathbf{P}_i = \mathbf{P}_{i-1} + \Delta \mathbf{P}_i \quad , \quad \mathbf{u}_i = \mathbf{u}_{i-1} + \Delta \mathbf{u}_i \quad (2)$$

Considering that the structure was in equilibrium at step $i-1$, it is desired to achieve equilibrium in the i -th step. Since the internal forces are a nonlinear function of the displacements, the solution of the linearized incremental problem does not satisfy the equilibrium. A vector of residual forces, \mathbf{R}_{i-1} , then arises as a result of the unbalance between external and internal forces, as shown in Eq. (3) to Eq. (5).

$$\mathbf{F}(\mathbf{u}_{i-1} + \Delta \mathbf{u}_i) = \mathbf{P}_{i-1} + \Delta \mathbf{P}_i \quad (3)$$

$$\mathbf{F}(\Delta \mathbf{u}_i) = \Delta \mathbf{P}_i + \mathbf{P}_{i-1} - \mathbf{F}(\mathbf{u}_{i-1}) \quad (4)$$

$$\mathbf{F}(\Delta \mathbf{u}_i) = \Delta \mathbf{P}_i + \mathbf{R}_{i-1} \quad (5)$$

Corrective iterations of the Newton-Raphson type are performed within each incremental step until the residual forces are numerically null, given a convergence criterion, so a new equilibrium state is established. In this way, the total increments in the i -th step are computed by accumulating the iterative increments of external forces, $\delta \mathbf{P}_i^j$, and displacements, $\delta \mathbf{u}_i^j$. This update process is given in Eq. (6), where superscript j indicates the j -th iteration of the i -th step.

$$\Delta \mathbf{P}_i^j = \Delta \mathbf{P}_i^{j-1} + \delta \mathbf{P}_i^j, \quad \Delta \mathbf{u}_i^j = \Delta \mathbf{u}_i^{j-1} + \delta \mathbf{u}_i^j \quad (6)$$

Substituting Eq. (6) into Eq. (5), it is possible to obtain the incremental-iterative system of equations to achieve equilibrium in the i -th step as follows:

$$\mathbf{F}(\Delta \mathbf{u}_i^{j-1}) + \mathbf{F}(\delta \mathbf{u}_i^j) = \Delta \mathbf{P}_i^{j-1} + \delta \mathbf{P}_i^j + \mathbf{R}_{i-1} \quad (7)$$

$$\mathbf{F}(\delta \mathbf{u}_i^j) = \Delta \mathbf{P}_i^{j-1} + \delta \mathbf{P}_i^j + (\mathbf{P}_{i-1} - \mathbf{F}(\mathbf{u}_{i-1})) - \mathbf{F}(\Delta \mathbf{u}_i^{j-1}) \quad (8)$$

$$\mathbf{F}(\delta \mathbf{u}_i^j) = \delta \mathbf{P}_i^j + (\mathbf{P}_{i-1} + \Delta \mathbf{P}_i^{j-1} - \mathbf{F}(\mathbf{u}_{i-1} + \Delta \mathbf{u}_i^{j-1})) \quad (9)$$

$$\mathbf{F}(\delta \mathbf{u}_i^j) = \delta \mathbf{P}_i^j + (\mathbf{P}_i^{j-1} - \mathbf{F}(\mathbf{u}_i^{j-1})) \quad (10)$$

$$\mathbf{F}(\delta \mathbf{u}_i^j) = \delta \mathbf{P}_i^j + \mathbf{R}_i^{j-1} \quad (11)$$

Defining the tangent stiffness matrix as the derivatives of the internal forces with respect to the nodal displacements ($\mathbf{K} = \partial \mathbf{F} / \partial \mathbf{u}$), evaluated at the previous known configuration, we finally reach the governing system of nonlinear finite element equations to be solved at the j -th iteration of the i -th step.

$$\mathbf{K}_i^{j-1} \delta \mathbf{u}_i^j = \delta \mathbf{P}_i^j + \mathbf{R}_i^{j-1} \quad (12)$$

The tangent stiffness matrix is composed by a linear portion, which depends only on the elastic properties of the elements, and a geometric portion that depends on the elements internal forces. The development of the local tangent matrix of an element, in the case of the UL formulation, depends on the degree of sophistication of the adopted shape functions and the inclusion of the nonlinear components of the Green strain tensor. Accordingly, different forms of the tangent matrix can be obtained. The tangent matrices of the UL formulation available in the developed tool are presented in Rodrigues (2019). In the CR formulation, only the relative displacements that cause deformations to an element are present in the deformed configuration about the co-rotated referential. Therefore, as stated by Santana (2015), the displacements and rotations measured in the element local coordinate system is usually considered small, and linear deformation measurements can be used. Based on this assumption, only one tangent matrix is available for the CR formulation, and its development can be found in Santana (2015).

Moreover, in the incremental-iterative process, the tangent matrix can be computed in accordance with a standard or modified Newton-Raphson iteration type. In the former method, the tangent matrix is updated in all iterations, considering the last obtained configuration and internal forces, while in the latter, the tangent matrix is only computed at the beginning of each incremental step and held constant for all subsequent iterations, i.e., $\mathbf{K}_i^{j-1} = \mathbf{K}_i^0$ for $j \geq 2$. The modified method has a lower computational cost at each iteration than the standard version, but convergence is usually slower.

2.3 The $N+1$ dimensional space formulation

In order to prepare the system of Eq. (12) to be solved, the vector of external nodal forces is expressed as the product of a load factor, λ , by a vector of reference external forces, \mathbf{P}_{ref} , which is usually taken as the total applied nodal forces to the structure. Therefore, the following expressions are valid:

$$\begin{cases} \mathbf{P}_i^j = \lambda_i^j \mathbf{P}_{\text{ref}} & \rightarrow & \lambda_i^j = \lambda_{i-1} + \Delta\lambda_i^j \\ \Delta\mathbf{P}_i^j = \Delta\lambda_i^j \mathbf{P}_{\text{ref}} & \rightarrow & \Delta\lambda_i^j = \Delta\lambda_i^{j-1} + \delta\lambda_i^j \\ \delta\mathbf{P}_i^j = \delta\lambda_i^j \mathbf{P}_{\text{ref}} & \rightarrow & \delta\lambda_i^j = \text{unknown} \end{cases} \quad (13)$$

The governing system of equations can be rewritten as:

$$\mathbf{K}_i^{j-1} \delta \mathbf{u}_i^j = \delta \lambda_i^j \mathbf{P}_{\text{ref}} + \mathbf{R}_i^{j-1} \quad (14)$$

This system has $N+1$ unknowns, N components of displacement increment in $\delta \mathbf{u}_i^j$ and one load factor increment $\delta \lambda_i^j$, but only N equations. It is necessary to add a constraint equation to the system, given by Eq. (15), where vector \mathbf{a} and scalars b and c are constants that can assume different values depending on the solution method.

$$\mathbf{a}_i^j \cdot \delta \mathbf{u}_i^j + b_i^j \delta \lambda_i^j = c_i^j \quad (15)$$

Equation (14) and Eq. (15) yield an augmented system of $N+1$ equations and unknowns:

$$\begin{bmatrix} \mathbf{K}_i^{j-1} & -\mathbf{P}_{\text{ref}} \\ (\mathbf{a}_i^j)^T & b_i^j \end{bmatrix} \begin{Bmatrix} \delta \mathbf{u}_i^j \\ \delta \lambda_i^j \end{Bmatrix} = \begin{Bmatrix} \mathbf{R}_i^{j-1} \\ c_i^j \end{Bmatrix} \quad (16)$$

The augmented matrix is no longer symmetric and has an increased bandwidth due to the added load factor. The solution of this system would be computationally undesirable with respect to both storage and efficiency. However, Batoz and Dhett (1979) presented a technique to overcome this problem by decomposing the system into two systems that use the original matrix, so the banded and symmetric properties of the original system remain intact:

$$\begin{cases} \mathbf{K}_i^{j-1} \delta \bar{\mathbf{u}}_i^j = \mathbf{P}_{\text{ref}} \\ \mathbf{K}_i^{j-1} \delta \bar{\bar{\mathbf{u}}}_i^j = \mathbf{R}_i^{j-1} \end{cases} \quad (17)$$

The solution for the iterative increment of displacements is the linear combination of a tangent, $\delta \bar{\mathbf{u}}_i^j$, and a residual, $\delta \bar{\bar{\mathbf{u}}}_i^j$, increment of displacements:

$$\delta \mathbf{u}_i^j = \delta \lambda_i^j \delta \bar{\mathbf{u}}_i^j + \delta \bar{\bar{\mathbf{u}}}_i^j \quad (18)$$

The unknown iterative increment of the load factor is given by the constraint equation, which is associated with a particular nonlinear solution method that gives rise to the constraint coefficients \mathbf{a} , b , and c of Eq. (15). The addition of the constraint equation to the system permits an adjustment of both the displacements and the load factor during the iterative cycle and the advance in the solution beyond critical points.

3. SOLUTION STRATEGIES

3.1 General solution algorithm

The introduction of the constraint equation to the incremental-iterative solution process of Eq. (17) is usually divided into two phases, a *predictor* and a *corrector* phase. The former ($j = 1$) calculates a predicted solution with a single linear analysis, and the latter ($j > 1$) tries to null the residual forces generated by the predicted solution through a cycle of iterative corrections. The general solution algorithm is illustrated in the diagram of Fig. 2. The difference between the UL and the CR formulations lies in the computation of the tangent matrix and the vector of internal forces. These steps are highlighted with bold boxes. The adopted convergence criterion is based on the ratio between the norms of the residual force vector and the reference force vector, which must be lower than a given tolerance, ε .

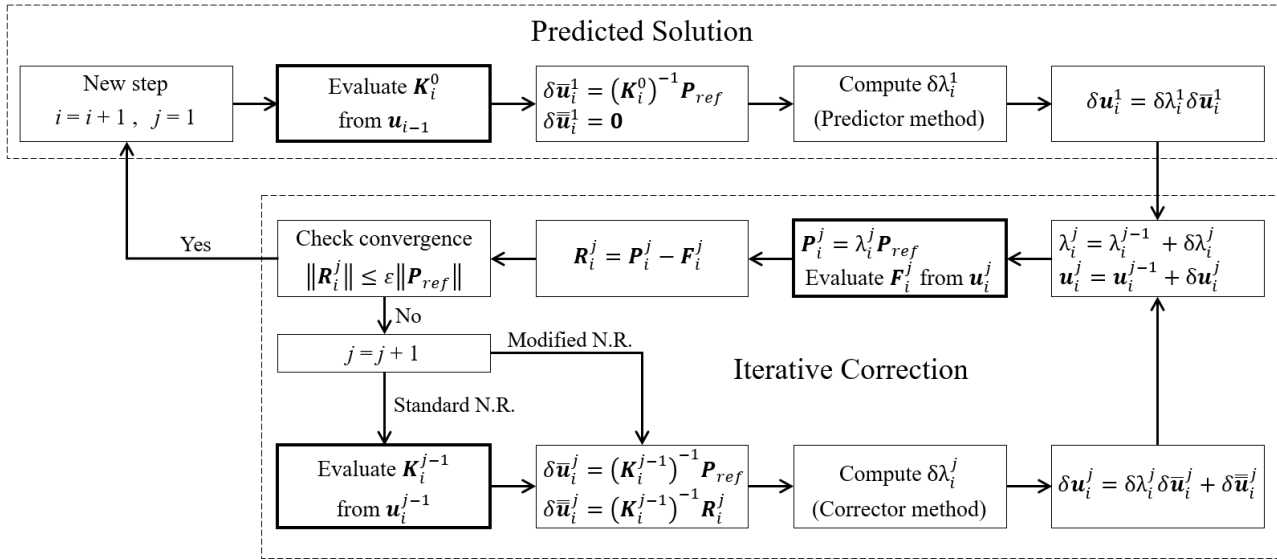


Figure 2. General solution algorithm of the incremental-iterative nonlinear system.

3.2 Predictive techniques

Obtaining the predicted solution has as fundamental task the calculation of the initial increment of the load factor, $\delta\lambda_i^1$. The automatic selection of this increment should reflect the degree of nonlinearity of the system, that is, it should provide large increments when the response is almost linear and lead to small increments when the response is strongly nonlinear. In addition, the algorithm must be able to choose the correct sign for the increment, being able to go beyond limit points.

In the first incremental step of the analysis ($i = 1$), the predicted increment of the load factor must be a prescribed value by the analyst. In the remaining steps ($i > 1$), it is computed according to the selected technique. The differences between the predictive techniques are the constraint equation to obtain the value of the increment, and the strategy to determine its size adjustment factor and correct sign. Table 1 shows the formulas to obtain the predicted increment of load factor for the techniques implemented in this work. In those expressions, I and J are the adjustment factors of the increment size, based on the nonlinearity of the solution. The strategies for obtaining these factors, as well as the appropriate increment sign, are presented posteriorly. The details of these formulations can be found in Silva [24], Santana [19], and Leon et al. [11].

Table 1. Constraint equation for the predictive techniques.

Predictive Technique	Load Factor Increment ($i > 1$)
Direct Increment (DI)	$\delta\lambda_i^1 = \pm I \cdot \delta\lambda_{i-1}^1$
Cylindrical Arc-Length Increment (CALI)	$\delta\lambda_i^1 = \pm I \sqrt{\frac{\Delta\mathbf{u}_{i-1} \cdot \Delta\mathbf{u}_{i-1}}{\delta\bar{\mathbf{u}}_i^1 \cdot \delta\bar{\mathbf{u}}_i^1}}$
Spherical Arc-Length Increment (SALI)	$\delta\lambda_i^1 = \pm I \sqrt{\frac{\Delta\mathbf{u}_{i-1} \cdot \Delta\mathbf{u}_{i-1} + (\Delta\lambda_{i-1})^2 \mathbf{P}_{ref} \cdot \mathbf{P}_{ref}}{\delta\bar{\mathbf{u}}_i^1 \cdot \delta\bar{\mathbf{u}}_i^1 + \mathbf{P}_{ref} \cdot \mathbf{P}_{ref}}}$
External Work Increment (EWI)	$\delta\lambda_i^1 = \pm \sqrt{I \left \frac{\Delta\lambda_{i-1} \mathbf{P}_{ref} \cdot \Delta\mathbf{u}_{i-1}}{\mathbf{P}_{ref} \cdot \delta\bar{\mathbf{u}}_i^1} \right }$
GSP-Based Increment (GSPI)	$\delta\lambda_i^1 = \pm J \cdot \delta\lambda_1^1$

3.3 Increment adjustment and sign

Two strategies to automatically adjust the size of the predicted increment of the load factor and determine its sign are used in this work. The strategy presented by Ramm (1981) is given in Eq. (19), where N_d and N_{i-1} are the desired number of iterations and the number of iterations required to achieve convergence in the previous step, respectively. The increment sign, represented by s in Eq. (20), for this strategy is taken as the same sign of the dot product between the vectors of tangent increment of displacements and total increment of displacements in the previous step:

$$I = \left(N_d / N_{i-1} \right)^{1/2} \quad (19)$$

$$s = \text{sign}(\delta \bar{\mathbf{u}}_i^1 \cdot \Delta \mathbf{u}_{i-1}) \quad (20)$$

Yang and Kuo (1994) proposed the use of the *Generalized Displacement Parameter* (GSP), given in Eq. (21), to adjust the predicted increment size in accordance with Eq. (22). When this strategy is used, the stiffness of the structure is measured with respect to the first incremental step, so stiffening and softening behavior are readily identified.

$$\text{GSP} = \frac{\delta \bar{\mathbf{u}}_1^1 \cdot \delta \bar{\mathbf{u}}_1^1}{\delta \bar{\mathbf{u}}_i^1 \cdot \delta \bar{\mathbf{u}}_{i-1}^1} \quad (21)$$

$$J = \sqrt{|\text{GSP}|} \quad (22)$$

The GSP changes sign only immediately after load limit points, as illustrated in Fig. 3. Therefore, as explained in Eq. (23), the sign of the predicted increment is positive in the first step and it must be inverted every time the GSP value is negative.

$$\begin{cases} i = 1 & \rightarrow s = \oplus \\ \text{GSP} < 0 & \rightarrow s = -s \end{cases} \quad (23)$$

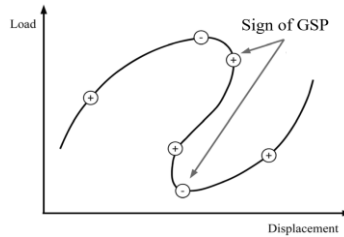


Figure 3. Behavior of the GSP sign.

3.4 Corrective techniques

In the corrective phase, it is sought to restore the structure equilibrium by vanishing the residual forces of the predicted solution through an iterative cycle. The iterative increments of load factor and displacements are restricted to the constraint equation that characterizes the selected technique. If the performed iterations involve not only the displacements, but also the load factor, then it is called a continuation method, since it can continue beyond limit points.

As mentioned earlier, one single strategy may not be capable of solving any general nonlinear problem. Therefore, most of the well-known corrective techniques were implemented, and they are presented in Tab. 2. Some of these iterative strategies are related to a particular predictive technique, and other are not bound to any. Because of that, some computer programs ask users to provide the predictive and corrective techniques separately. In the developed tool, in order to simplify the input data, the solution method refers to the technique for the corrective iterations and it includes the predictive technique that is more appropriate. Among the nine solution methods implemented, only one is not a continuation method, the *Load Control*, which works only with displacement increments in each iteration. The expressions for the *Cylindrical Arc-Length* and *Spherical Arc-Length* methods may result in a complex value for the load factor correction if the incremental steps are not sufficiently small (Krenk, 2009). In this case, the program returns a warning. In addition, it was opted to limit the value of the load factor correction of all methods to 0.5, to avoid extrapolating values.

Table 2. Solution methods with the corresponding predictive and corrective techniques.

Solution Method	Predictive Technique	Load Factor Correction ($j \geq 1$)
Load Control (LC)	DI	$\delta\lambda_i^j = 0$
Linear Arc-Length (Riks)* (LAL_RI)	CALI	$\delta\lambda_i^j = -\frac{\delta\bar{\mathbf{u}}_i^j \cdot \delta\mathbf{u}_i^1}{\delta\bar{\mathbf{u}}_i^j \cdot \delta\mathbf{u}_i^1 + \delta\lambda_i^1 (\mathbf{P}_{\text{ref}} \cdot \mathbf{P}_{\text{ref}})}$
Linear Arc-Length (Ramm)** (LAL_RA)	CALI	$\delta\lambda_i^j = -\frac{\delta\bar{\mathbf{u}}_i^j \cdot \Delta\mathbf{u}_i^{j-1}}{\delta\bar{\mathbf{u}}_i^j \cdot \Delta\mathbf{u}_i^{j-1} + \Delta\lambda_i^{j-1} (\mathbf{P}_{\text{ref}} \cdot \mathbf{P}_{\text{ref}})}$
Cylindrical Arc-Length*** (CAL)	CALI	$\delta\lambda_i^j = -\frac{b}{a} + s\sqrt{\left(\frac{b}{a}\right)^2 - \frac{c}{a}}$
Spherical Arc-Length*** (SAL)	SALI	$\delta\lambda_i^j = -\frac{b}{a} + s\sqrt{\left(\frac{b}{a}\right)^2 - \frac{c}{a}}$
Minimum Norm (MN)	CALI	$\delta\lambda_i^j = -\frac{\delta\bar{\mathbf{u}}_i^j \cdot \delta\bar{\mathbf{u}}_i^j}{\delta\bar{\mathbf{u}}_i^j \cdot \delta\bar{\mathbf{u}}_i^j}$
Orthogonal Residue (OR)	CALI	$\delta\lambda_i^j = -\frac{\mathbf{R}_i^j \cdot \Delta\mathbf{u}_i^{j-1}}{\mathbf{P}_{\text{ref}} \cdot \Delta\mathbf{u}_i^{j-1}}$
Work Control (WC)	EWI	$\delta\lambda_i^j = -\frac{\mathbf{P}_{\text{ref}} \cdot \delta\bar{\mathbf{u}}_i^j}{\mathbf{P}_{\text{ref}} \cdot \delta\bar{\mathbf{u}}_i^j}$
Generalized Displacement (GD)	GSPI	$\delta\lambda_i^j = -\frac{\delta\bar{\mathbf{u}}_i^1 \cdot \delta\bar{\mathbf{u}}_i^j}{\delta\bar{\mathbf{u}}_i^1 \cdot \delta\bar{\mathbf{u}}_i^j}$
* Presented in Riks (1972) and Riks (1979).		
**Presented in Ramm (1981) and Ramm (1982).		
*** Parameters for methods CAL and SAL:		
$a = \delta\bar{\mathbf{u}}_i^j \cdot \delta\bar{\mathbf{u}}_i^j + \beta(\mathbf{P}_{\text{ref}} \cdot \mathbf{P}_{\text{ref}})$ $b = \delta\bar{\mathbf{u}}_i^j \cdot (\delta\bar{\mathbf{u}}_i^j + \Delta\mathbf{u}_i^{j-1}) + \beta\Delta\lambda_i^{j-1} (\mathbf{P}_{\text{ref}} \cdot \mathbf{P}_{\text{ref}})$ $c = \delta\bar{\mathbf{u}}_i^j \cdot (\delta\bar{\mathbf{u}}_i^j + 2\Delta\mathbf{u}_i^{j-1})$		
$\begin{cases} \text{Cylindric Arc-Length} \rightarrow \beta = 0 \\ \text{Spheric Arc-Length} \rightarrow \beta = 1 \end{cases}$ $s = \text{sign}(\Delta\mathbf{u}_i^{j-1} \cdot \delta\bar{\mathbf{u}}_i^j)$		

4. DEVELOPED TOOL

A geometrically nonlinear analysis module was implemented and incorporated to the Ftool program (Martha, 1999). Over the last years, Ftool have demonstrated to be a valuable tool for teaching structural engineering. It has been used on solid mechanics, structural analysis, and structural design courses in many universities all over the world as well as in the industry. It consists of a graphical structural analysis program that has, in a single platform, all the necessary tools for efficient modeling, pre and post-processing the results. The internal solver, called FRAMOOOP, is a simplified version of the FEMOOOP (Finite Element Method Object Oriented Program) system (Martha and Parente, 2002), modified to perform only the analysis of framed structure models (models made of beam elements). The FRAMOOOP system is written in the C programming language and adopts a programming philosophy similar to the *Object Oriented Programming* (OOP) paradigm, which is advantageous for a structural analysis code (Rangel and Martha, 2019). The graphical user interface is built using the IUP (Portable User Interface) system (Levy *et al.*, 1996), which is a multi-platform toolkit that offers a simple API for building graphical user interfaces in different programming languages and allows a program source code to be compiled in different systems without any modification. Its main advantage is the high performance, due to the fact that it uses native interface elements. Figure 4 shows the main window of the Ftool program, highlighting the new menus created for this work: *Analysis Menu* and *Plotting Menu*.

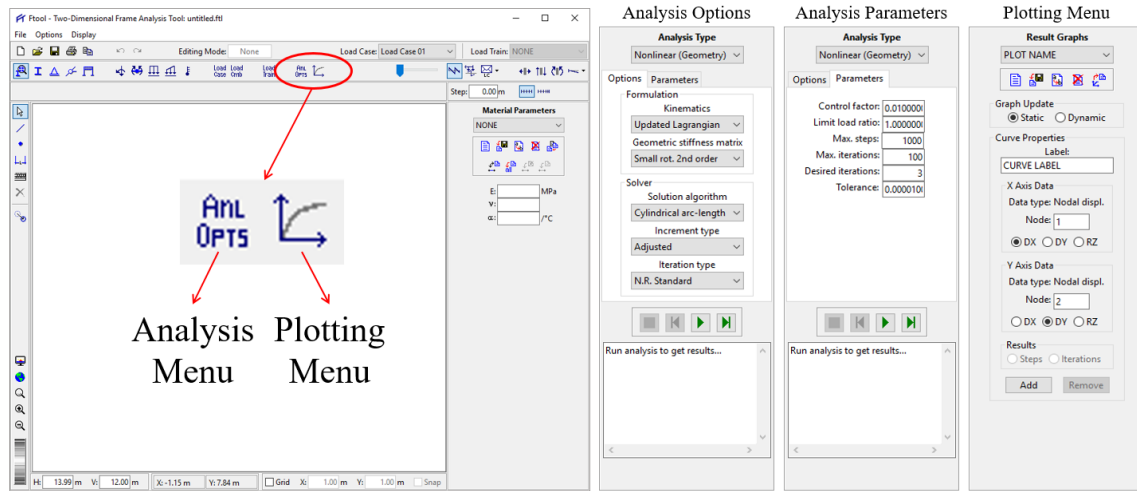


Figure 4. Developed menus for nonlinear analysis.

In the *Analysis Menu*, when the *Analysis Type* option is selected to *Nonlinear (Geometry)*, users are requested to set multiple options and parameters to perform the nonlinear analysis. This nonlinear menu displays two tabs, the *Options* tab and the *Parameters* tab. In the *Options* tab, a *Formulation* panel provides options for the kinematic description used to formulate the problem (Updated Lagrangian or Co-Rotational) and the corresponding geometric stiffness matrices. If the *Updated Lagrangian* option is selected as the kinematics, three options of stiffness matrix are available: *Small rotation 2nd order*, *Large rotation 2nd order*, and *Large rotation 4th order* (Rodrigues, 2019). On the other hand, if *Corotational* option is selected, the geometric matrix option is locked in *Small rotation 2nd order*. All these matrices are implemented in the FRAMOOOP code for Euler-Bernoulli and Timoshenko beam element, and can be found in the previously referred works. A *Solver* panel brings the *Solution algorithm* option to specify which of the incremental-iterative solution methods, listed in Tab. 2, will be used to perform the analysis. The *Increment type* option can be set to *Adjusted* or *Constant*, and defines whether the adjustment factor of the predicted increment will be used or not. In the *Iteration type* option, users can switch between standard or modified Newton-Raphson strategies of updating the tangent stiffness matrix. In the *Parameters* tab, the *Control factor* is the predicted increment of the load factor in the first step. The *Limit load ratio* and the *Max. steps* (maximum number of steps) parameters determine when the analysis will stop. The *Desired iterations* sets the N_d value in Eq. (19). Since this parameter is only used to adjust the predicted increment size, its input field is inactive if the *Increment type* option is set to *Constant*. Finally, the *Max. iterations* (maximum number of iterations) and the *Tolerance* are information for the convergence criterion.

Analysis control buttons are positioned below the *Options* and *Parameters* tabs. These buttons allow users to run the complete analysis, until the maximum number of steps or limit load ratio is reached, or advance and rewind a certain number of steps. Changes in any of the analysis options and parameters are also allowed in between steps, so the analysis will continue with the new input data. The implementation of these control options is done by saving in a linked list, all the necessary data to start the analysis in any given step, based on the history of the results. This possibility to perform the analysis in an incremental-adaptive way is an important feature of the developed tool. Some problems may not converge in a specific step of the solution with a set of parameters, but changing it can make the analysis to go beyond that problematic point. A step-by-step feedback of the analysis progress is given in a text field in the bottom of the *Analysis Menu*.

The *Plotting Menu* is where users can create graphs and add curves to them. Different data options can be plotted in the X and Y-axes. These options include nodal displacement, load ratio, step number, displacement increment, and load factor increment. For example, one can create a graph that shows the relation between the displacements of two degrees of freedom as the analysis goes on, or study the behavior of the load factor increment for each analysis step. Each graph can be set as static or dynamic. Static graphs never change its data while dynamic graphs are automatically updated when the current analysis step changes. Another interesting feature of the *Plotting Menu* is the option to plot the iteration points rather than only the converged step points. A more in-depth study of the behavior of the solution algorithms can be performed from this feature.

5. EXAMPLE

Figure 5 shows a structural model known as Lee frame (Lee *et al.*, 1968), the considered analysis settings, and the deformed configuration for distinct analysis stages, with load limit points in bold. Figure 6 presents the equilibrium paths of the horizontal displacement, vertical displacement, and rotation of the corner node, as well as the load ratio value for each analysis step. Table 3 brings the number of steps to obtain the solution using the continuation methods of Tab. 2 and varying the formulation and the control factor, where X means that the full response could not be captured.

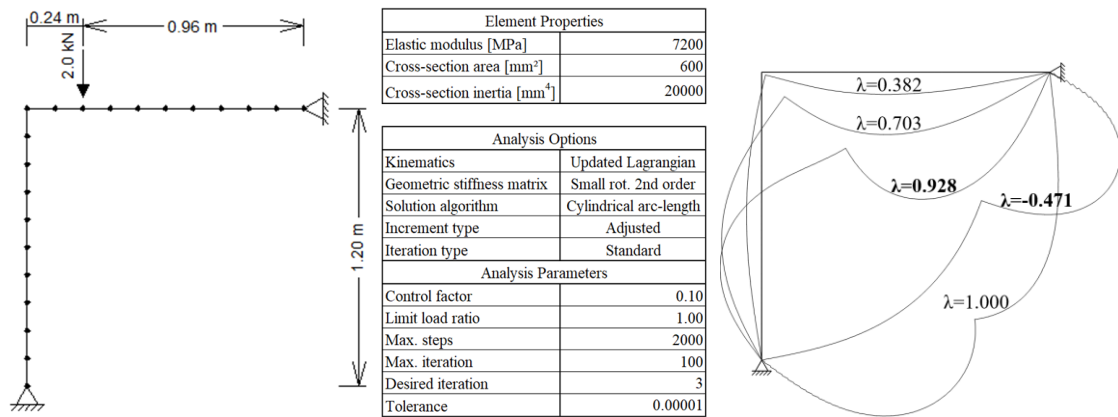


Figure 5. Lee frame analysis.

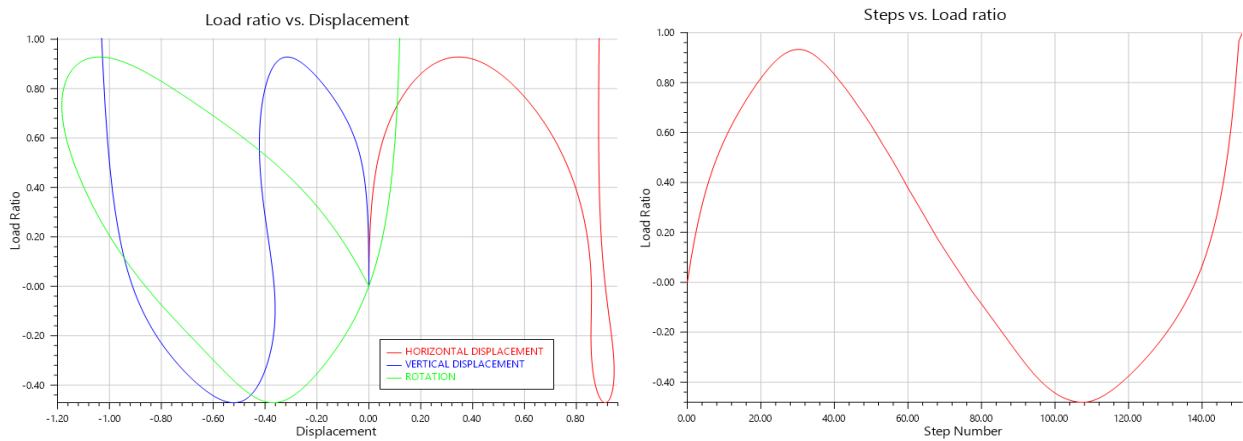


Figure 6. Equilibrium paths and load ratio behavior of the Lee frame.

Table 3. Number of steps to trace full equilibrium path for different analysis settings.

Settings	AL_RI	AL_RA	CAL	SAL	MN	OR	WC	GD
UL ($\lambda_1^0 = 0.10$)	158	158	151	161	162	X	61*	103
CR ($\lambda_1^0 = 0.10$)	X	X	104	X	101	X	46*	103
UL ($\lambda_1^0 = 0.01$)	193	185	178	193	185	X	X	1033
CR ($\lambda_1^0 = 0.01$)	128	124	110	135	108	X	X	1034
* Unable to trace a smooth path								

As observed in Tab. 3, the UL formulation works better for a larger initial increment, while the CR formulation requires less steps when using a smaller increment. The WC method has difficulty to go through snap-back points, as explained by Leon *et al.* (2011). The OR method does not work well for this problem.

6. CONCLUSION

As expected, the results showed that any change to the options and parameters of a nonlinear analysis may lead to a greater or lesser efficiency in obtaining the response, or the inability to obtain it correctly, or even the inability obtain it at all. Therefore, the availability of several methods and options for the analysis, especially for educational purposes, has proved to be fundamental for those who need to analyze models with high nonlinearities, and those who want to study the performance of the solution methods and the influence of each analysis parameter. It is important to keep in mind that the present implementation has some particular considerations and, because of this, the obtained results may vary a little for other implementations. However, similar conclusions should be drawn about the overall behavior of the methods to solve the nonlinear problem.

7. ACKNOWLEDGEMENTS

The authors are grateful to researchers Ricardo Silveira and Murillo Santana for the shared knowledge, CNPq for the financial support, and Tecgraf/PUC-Rio for providing the necessary resources during the development of this work.

8. REFERENCES

- Bathe, K.J., 1996. *Finite Element Procedures*. Prentice-Hall, Englewood Cliffs, NJ.
- Bathe, K.J., Bolourchi, S., 1979. "Large displacement analysis of three-dimensional beam structures". *Int. J. Numer. Methods Eng.*, Vol. 14, pp. 961-986.
- Batoz, J.L., Dhatt, G., 1979. "Incremental displacement algorithms for nonlinear problems", *Int. J. Numer. Methods Eng.*, 14(8), pp. 1262-1267.
- Bergan, P.G., Horigmoe, G., Brakeland, B., Soreide, T.H., 1978. "Solution techniques for non-linear finite element problems". *Int. J. Numer. Methods Eng.*, 12(11), pp. 1677-1696.
- Felippa, C.A., 2017. *Nonlinear Finite Element Methods*. Class notes, University of Colorado, Boulder, USA.
- Krenk, S., 2009. *Non-linear Modeling and Analysis of Solids and Structures*. Cambridge University Press.
- Lee, S., Manuel, F.S., Rossow, E.C., 1968. "Large deflections and stability of elastic frames". *Journal of the Engineering Mechanics Division*, EM2, pp. 521-547.
- Leon, S.E., Paulino, G.H., Pereira, A., Menezes, I.F.M., Lages, E.N., 2001. "A unified library of nonlinear solution schemes". *Applied Mechanics Reviews*, Vol. 64.
- Levy, C.H., Figueiredo, L.H., Gattas, M., Lucena, C., and Cowan, D., 1996. "IUP/LED: A portable user interface development tool". *Software: Practice and Experience*, 26(7), pp. 737-762.
- Malvern, L.E., 1969. *Introduction to the Mechanics of a Continuous Medium*. Prentice-Hall, Englewood Cliffs, NJ.
- Martha, L.F., 1999. "Ftool: A structural analysis educational interactive tool". In *Proceedings of the Workshop in Multimedia Computer Techniques in Engineering Education*. Institute for Structural Analysis, Technical University of Graz, Austria, pp. 51-65.
- Martha, L.F., Parente Jr., E., 2002. "An object-oriented framework for finite element programming". In *Proceedings of the Fifth World Congress on Computational Mechanics – IACM*. Vienna, Austria.
- Ramm, E., 1981. "Strategies for tracing the non-linear response near limit-points". *Nonlinear finite element analysis in structural mechanics*, ed. W. Wunderlich, Springer-Verlag, Berlin, pp. 63-89.
- Ramm, E., 1982. "The Riks/Wempner Approach $\frac{3}{4}$ – An extension of displacement control method in non-linear analysis". *Non-linear Computational Mechanics*, ed. E. Hinton *et al.*, Pineridge, Swansea, pp. 63-86.
- Rangel, R.L., Martha, L.F., 2019. "LESM – An object-oriented MATLAB program for structural analysis of linear element models". *Comput. Appl. Eng. Educ.* Vol. 27, Issue 3, pp. 553-571.
- Riks, E., 1972. "The application of Newton's methods to the problems elastic stability". *Journal of Applied Mechanics*, Vol. 39, pp. 1060-1066.
- Riks, E., 1979. "An incremental approach to the solution of snapping and buckling problems". *International Journal of Solids and Structures*, Vol. 15, pp. 529-551.
- Rodrigues, M.A.C., 2019. *Soluções integradas para as formulações do problema de não linearidade geométrica*. Ph.D. thesis, Pontifícia Universidade Católica do Rio de Janeiro, Brasil.
- Santana, M.V.B., 2015. *Desenvolvimento de sistema computacional via MATLAB/GUI (Graphical User Interface) para análise geometricamente não linear de estruturas*. M.Sc. thesis, Universidade Federal de Ouro Preto, Brasil.
- Silva, A.R.D., 2009. *Sistema computacional para a análise avançada estática e dinâmica de estruturas metálicas*. Ph.D. thesis, Universidade Federal de Ouro Preto, Ouro Preto, Brasil.
- Yang, Y.B., Kuo, S.R., 1994. *Theory and Analysis of Nonlinear Framed Structures*. Prentice-Hall, Englewood Cliffs, NJ.

9. RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.