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
rafaelangelo@tecgraf.puc-rio.br
lfm@tecgraf.puc-rio.br

Abstract. This work presents the development and use of a graphical tool to analyze 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 follow its equilibrium path. 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 equilibrium and cinematic equations of the finite element method were written in the initial configuration, second-order effects, such as unexpected internal forces and buckling, would not be predicted and could cause the structure to fail.

The resulting finite element equations constitute a nonlinear system, shown in Eq. (1), where $K_e$ is the elastic stiffness matrix, $K_g$ is the geometric stiffness matrix, $P$ is the vector of external loads acting in the directions of the model’s degrees of freedom, and $U$, the unknowns of the system, is the vector of displacements in the degrees of freedom directions. The nonlinearity of the system resides on the fact that the geometric stiffness matrix depends on the internal forces of the elements. Since the internal forces depends on the displacements of the elements, which are the unknowns, the system must be solved iteratively.

$$([K_e] + [K_g])\{U\} = \{P\}$$

The identification of second-order effects can be done by studying the history of the structure equilibrium 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 a convergence criterion is satisfied. 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 are 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. These procedures should also allow for the possibility of an extensive control over the solution process by parameters that are input to the analysis.
2. INCREMENTAL-ITERATIVE METHODS

To perform the incremental-iterative process, the total load applied to the structure is expressed by a load factor that multiplies a reference load vector. In this work, the reference load is taken as the total load. In each step, indicated by the subscript \( i \), a series of iterations, indicated by the subscript \( j \), are executed to compute iterative increments of the load factor and displacements vector. These iterative increments of load and displacements are restricted to a constraint equation, which characterizes the continuation methods, and are accumulated to give the total increment in the analysis.

2.1 The iterative system of equations

In each iteration, the linear system of Eq. (2) must be solved to compute the vector of iterative displacement increments \( \delta U \). In this equation, \( K_T \) is the tangent stiffness matrix (sum of the elastic and geometric stiffness matrices), computed with the internal forces from the previous iteration, \( \delta \lambda \) is the iterative increment of the load factor, \( P_{\text{ref}} \) is the reference load vector, and \( R \) is the unbalance load vector from the previous iteration. The unbalanced load vector is calculated according to Eq. (3), where \( P \) is the vector of external loads and \( F \) is the vector of internal forces, both referred to the previous iteration.

\[
\begin{align*}
[K_T]^{i-1} \{\delta U\}^i &= \delta \lambda^i \{P_{\text{ref}}\} + \{R\}^{i-1} \\
\{R\}^{i-1} &= \{P\}^{i-1} - \{F\}^{i-1}
\end{align*}
\]

The system in Eq. (2) needs one more equation to determine the load factor increment, which is unknown. This additional equation is called constraint equation, and its insertion into the system leads to Eq. (4), where vector \( a \) and scalars \( b \) and \( c \) are constants that can assume different values depending on the constraint equation that is being used.

\[
\begin{bmatrix}
[K_T]^{i-1} & \{-P_{\text{ref}}\} \\
\{a\}^i & \{b\}^i & \{c\}^i
\end{bmatrix}
\begin{bmatrix}
\{\delta U\}^i \\
\delta \lambda^i
\end{bmatrix}
= 
\begin{bmatrix}
\{R\}^{i-1}
\end{bmatrix}
\]

(4)

The augmented matrix of the new system is no longer symmetric and has an increased bandwidth. To overcome computer efficiency problems, Batoz and Dhatt (1979) proposed the decomposition of the augmented system into the two systems of Eq. (5) that use the original tangent stiffness matrix. The solution for the vector of iterative displacement increments, given in Eq. (6), is obtained by superposition.

\[
\begin{align*}
[K_T]^{i-1} \{\delta U_p\}^i &= \{P_{\text{ref}}\} \\
[K_T]^{i-1} \{\delta U_R\}^i &= \{R\}^{i-1} \\
\{\delta U\}^i &= \delta \lambda^i \{\delta U_p\}^i + \{\delta U_R\}^i
\end{align*}
\]

(5)

(6)

2.2 Constraint equations

The nonlinear solution methods are inherently different in their formulation and therefore feature unique constraint equations for the incremental-iterative procedure by giving rise to the constant parameters of Eq. (4). The combination of the governing finite element equations and the constraint equation parameters into a single matrix equation characterize a unified approach for the solution methods (Leon et. al, 2011). Table 1 gives the expressions for the constraint equations as well as the required input parameter of the solution methods implemented in this work: Load Control Method (LCM), Work Control Method (WCM), Arc-Length Control Method (ALCM), and Generalized Displacement Control Method (GDCM).

In the first iteration, the constraint equation computes the load factor increment that restricts the increment of the remaining iterations. To do so, a prescribed initial control factor is required and must be provided by the analyst. The determination of the direction of the load factor increment in the first iteration (sign of the expression) is based on the stiffness of the structure and follows the criteria of the generalized stiffness parameter (GSP), described by Yang and Kuo (1994). The GSP will be positive for stiffening structural systems and negative for softening structural systems.
The iterative process is performed until equilibrium is satisfied by a convergence criterion and given tolerance. The adopted criteria is based on the ratio of the norm of the unbalanced load vector to the norm of the applied load vector at that incremental step. If it is satisfied, an equilibrium point was reached and the iterative process starts for a new step.

Table 1. Expressions for the constraint equations of the implemented nonlinear solution methods.

<table>
<thead>
<tr>
<th>Method</th>
<th>Initial control factor</th>
<th>First iteration ((j = 1))</th>
<th>Remaining iterations ((j &gt; 1))</th>
</tr>
</thead>
<tbody>
<tr>
<td>LCM</td>
<td>Load factor increment ((\Delta \lambda))</td>
<td>(\delta \lambda_{i}^{1} = \Delta \lambda)</td>
<td>(\delta \lambda_{i}^{j} = 0)</td>
</tr>
<tr>
<td>WCM</td>
<td>Work increment ((\Delta W))</td>
<td>(\delta \lambda_{i}^{1} = \pm \frac{\Delta W}{\left{ P_{ref} \right}<em>{i} \cdot \left{ \delta U</em>{P} \right}_{i}^{1}})</td>
<td>(\delta \lambda_{i}^{j} = -\frac{\left{ P_{ref} \right}<em>{i} \cdot \left{ \delta U</em>{R} \right}<em>{i}^{j}}{\left{ \delta U</em>{P} \right}_{i}^{j} + 1})</td>
</tr>
<tr>
<td>ALCM</td>
<td>Arc length increment ((\Delta S))</td>
<td>(\delta \lambda_{i}^{1} = \pm \frac{\Delta S^{2}}{\left{ \delta U_{P} \right}<em>{i}^{1} \cdot \left{ \delta U</em>{P} \right}_{i-1} + 1})</td>
<td>(\delta \lambda_{i}^{j} = -\frac{\left{ \delta U_{P} \right}<em>{i}^{j} \cdot \left{ \delta U</em>{R} \right}<em>{i}^{j}}{\left{ \delta U</em>{P} \right}_{i}^{j} + 1})</td>
</tr>
<tr>
<td>GDCM</td>
<td>Load parameter ((\delta \lambda))</td>
<td>(\delta \lambda_{i}^{1} = \pm \delta \lambda)</td>
<td>(\delta \lambda_{i}^{j} = -\frac{\left{ \delta U_{P} \right}<em>{i}^{j} \cdot \left{ \delta U</em>{R} \right}<em>{i}^{j}}{\left{ \delta U</em>{P} \right}_{i}^{j} + 1})</td>
</tr>
</tbody>
</table>

3. DEVELOPED TOOL

This nonlinear analysis module was implemented in the FTOOL (Two-dimensional Frame Analysis Tool) 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. 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 of the results, and a fast solving strategy. The internal solver, called FRAMOOP, is a simplified version of the FEMOOP (Finite Element Method Object Oriented Program) system (Martha and Parente, 2002), modified to perform only the analysis of framed structure models (models made of bar elements). The FRAMOOP system is written in the C programming language and adopts a programming philosophy similar to the Object Oriented Programming (OOP) paradigm. 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 1a shows the developed menu for nonlinear analysis and Fig. 1b shows the new menu for creating and plotting graphs of the equilibrium path and other response options.

Figure 1. a) Nonlinear analysis menu. b) Graph menu with equilibrium path plotted.
The analysis menu allows users to set multiple parameters and to change them in any stage of the analysis. The solution algorithm option specifies which of the mentioned method will be used to perform the analysis. The geometric stiffness matrix option specifies the matrix formulation, based on Rodrigues et. al (2019), that is included in the FRAMOOP code for Euler-Bernoulli or Timoshenko beam element. The matrix update options tells if the tangent matrix is updated in every iteration (standard update) or only in the first iteration of each step (modified update). The control factor is the prescribed value used by the constraint equation in the first iteration of each step. The limit load ratio and the maximum number of steps tell the program when to stop the analysis. Finally, the maximum number of iterations and the tolerance are information about the convergence criterion. A step-by-step feedback of the analysis progress is given in a text field.

User also have the possibility to advance or rewind a certain number of steps in the analysis. Changes in the analysis options and parameters are allowed between steps. The implementation of theses control options was done by saving in a linked list, all the necessary data to start the analysis in any given step. This option is an important feature of the developed tool, since some problems may not converge in a specific point of the equilibrium path with a set of parameters, but changing it can make the analysis to go beyond that problematic point. The manuscript of this work brings some examples of the influence of changing some parameters during the analysis process.

The graph menu is where users can create graphs and add curves to them. Different data options can be selected to be plotted in the X and Y-axes. These options include nodal displacement, load ratio, step number, displacement increment, and load ratio increment. For example, one can create a graph that shows the relation of the displacements between 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 graph 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.

4. ACKNOWLEDGEMENTS

The authors would like to thank the National Council for Scientific and Technological Development (CNPq), the Pontifical Catholic University of Rio de Janeiro (PUC-Rio), and the Technical-Scientific Software Development Institute of PUC-Rio (Tecgraf/TUC-Rio) for the financial support and for providing the necessary space and resources used during the development of this work.

5. REFERENCES


6. RESPONSIBILITY NOTICE

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