Pontifícia Universidade Católica
do Rio de Janeiro

Rafael Lopez Rangel

Educational Tool for Structural Analysis of Plane Frame Models with Geometric Nonlinearity

Dissertação de Mestrado

Dissertation presented to the Programa de Pós-graduação em Engenharia Civil of PUC-Rio in partial fulfillment of the requirements for the degree of Mestre em Engenharia Civil.

Advisor: Prof. Luiz Fernando Martha

Rio de Janeiro
September 2019

Pontifícia Universidade Católica
do Rio de Janeiro

Rafael Lopez Rangel

Educational Tool for Structural Analysis of Plane Frame Models with Geometric Nonlinearity

Dissertation presented to the Programa de Pós-graduação em Engenharia Civil of PUC-Rio in partial fulfillment of the requirements for the degree of Mestre em Engenharia Civil. Approved by the Examination Committee.

Prof. Luiz Fernando Campos Ramos Martha
Advisor
Departamento de Engenharia Civil e Ambiental – PUC-Rio

Profa. Deane de Mesquita Roehl
Departamento de Engenharia Civil e Ambiental – PUC-Rio

Prof. Ivan Fabio Mota de Menezes
Departamento de Engenharia Mecânica – PUC-Rio

Prof. Ricardo Azoubel da Mota Silveira
Departamento de Engenharia Civil – UFOP

Prof. Roque Luiz da Silva Pitangueira
Departamento de Engenharia de Estruturas – UFMG

Rio de Janeiro, September 27th, 2019

Rafael Lopez Rangel

The author graduated in Civil Engineering from Pontifical Catholic University of Rio de Janeiro (PUC-Rio) in 2016.

Bibliographic data

Rangel, Rafael Lopez

Educational Tool for Structural Analysis of Plane Frame Models with Geometric Nonlinearity / Rafael Lopez Rangel; advisor: Luiz Fernando Campos Ramos Martha. – Rio de Janeiro: PUC-Rio, Departamento de Engenharia Civil e Ambiental, 2019.

v., 200 f.: il. color. ; 29,7 cm.

Dissertação (mestrado) – Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Engenharia Civil e Ambiental.

Inclui bibliografia

1. Engenharia Civil e Ambiental – Teses. 2. Análise de estruturas. 3. Pórticos planos. 4. Não linearidade geométrica. 5. Ferramenta Educacional. I. Martha, Luiz Fernando Campos Ramos. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Engenharia Civil e Ambiental. III. Título.

CDD: 624

Acknowledgments

I would like to thank my advisor, Luiz Fernando, for all the support, trust, and teachings over the last years of working together.

I’m very grateful to everyone else who contributed to the development of this work by providing valuable knowledge:

Professor Ivan Menezes, for the help with the FRAMOOP code and for patiently receiving me when I had questions.

Professor Carlos Alberto de Almeida, for introducing me to the studies of nonlinear finite element method, and later discussions about the development of my work.

Ricardo Marques, my colleague at the MVGEO group, for teaching me so much about programming in C/C++, as well as the help and support with IUP.

Professor Ricardo Azoubel, for the e-mail exchanges and discussions that contributed enormously to my understanding of the subject.

Murillo Santana, for the time he spent with me sharing his knowledge on nonlinear analysis when he was in Rio, which helped this work to be completed.

Marcos Rodrigues, whose thesis inspired this work. Professor Roque Pitangueira, for the suggestions for improvements. Professor André Pereira for the provided material. Professors Deane Roehl and Paulo Gonçalves for the great courses on the Finite Element Method and Structural Stability. All the other great teachers I´ve had throughout my life, impossible to name everyone, but thank you very much.

I would also like to thank my colleagues at Tecgraf for the everyday convivial, which made my days a lot nicer: Pedro, Fabio, and Antonio.

To my Family: Anna, Paulo, Fernando, Anna Lucia, Euclydes, and Mariana, for supporting me in everything.

To the Tecgraf/PUC-Rio Institute, and all its employees, for providing an excellent working environment.

To CNPq for the financial support.

Abstract

Rangel, Rafael Lopez; Martha, Luiz Fernando Campos Ramos (Advisor). **Educational Tool for Structural Analysis of Plane Frame Models with Geometric Nonlinearity**. Rio de Janeiro, 2019. 200p. Dissertação de Mestrado – Departamento de Engenharia Civil e Ambiental, Pontifícia Universidade Católica do Rio de Janeiro.

Nonlinear analysis of structures is an important task for efficient and safe projects, allowing the saving of material resources and the identification of second-order effects on the behavior of structural models that may have significant consequences. This type of analysis is performed with iterative numerical algorithms, and visualization of graphic results is essential to auxiliary the interpretation of the analyst. For this reason, nonlinear analyses only became common with the advent of graphical-interactive computational applications. However, unlike a linear-elastic analysis, where the results provided by the program depend very little on the users knowledge about the solution methods, a nonlinear analysis requires a series of input parameters related to the numerical methods and thus demands a basic understanding about the solution algorithms and nonlinear structural behavior. With this in mind, this work aims to develop a user-friendly computational tool with a simple graphical interface, but with a robust solver, to assist the learning of geometrically nonlinear analysis of two-dimensional frame models. The structural analysis software Ftool, largely used by the Civil Engineering community and academia, was adopted to receive the new features to perform geometrically nonlinear analyses. In the new version of the Ftool program, students, engineers and researchers have the opportunity to use and test various solution techniques of the nonlinear system of equilibrium equations, which are described in detail throughout this work. The way the nonlinear analysis is performed allows for a full control by users over the progress of the analysis. In addition, graph results can be studied in the new plotting environment of the program.

Keywords

Structural analysis; geometric nonlinearity; two-dimensional frame models; post-buckling behavior; educational software.

Resumo

Rangel, Rafael Lopez; Martha, Luiz Fernando Campos Ramos (Orientador). **Ferramenta Educacional para Análise Estrutural de Modelos de Pórticos Planos**. Rio de Janeiro, 2019. 200p. Dissertação de Mestrado – Departamento de Engenharia Civil e Ambiental, Pontifícia Universidade Católica do Rio de Janeiro.

A análise não linear de estruturas é uma tarefa de grande importância na execução de projetos eficientes e seguros, permitindo a economia de recursos materiais, ao tempo que se identifica efeitos de segunda ordem no comportamento do modelo que podem vir a ter consequências significativas. Esse tipo de análise é realizado através de algoritmos numéricos iterativos, e a visualização de resultados gráficos é essencial para auxiliar a interpretação do analista. Por isso, a análise não linear só se tornou recorrente com o advento de aplicações computacionais gráfico-iterativas. Porém, diferentemente de uma análise linear-elástica, em que os resultados fornecidos pelo programa pouco dependem do conhecimento do usuário sobre os métodos de solução, a análise não linear requer uma série de parâmetros de entrada relacionados aos métodos numéricos e, portanto, exige um conhecimento básico por parte do usuário sobre os algoritmos de solução e comportamento do modelo. Tendo isso em vista, este trabalho busca desenvolver uma ferramenta computacional de fácil uso e com uma interface gráfica simples, porém com um *solver* robusto, para auxiliar a aprendizagem da análise geometricamente não linear de modelos aporticados bidimensionais. Para isso o programa de análise estrutural Ftool, consagrado na comunidade de Engenharia Civil e no meio acadêmico, foi adotado para receber os novos recursos para executar a análise com não linearidade geométrica. Na nova versão do Ftool, os usuários têm a oportunidade de utilizar e testar diversas técnicas de solução do sistema não linear de equilíbrio do modelo, descritas nesse trabalho. A forma como a análise é executada permite um controle total do usuário sobre o progresso da análise. Além disso, resultados em forma de gráficos podem ser estudados no novo ambiente de plotagem do programa.

Palavras-chave

Análise de estruturas; não linearidade geométrica; modelos de pórtico plano; comportamento pós-crítico; software educacional.

Table of contents

List of figures

List of tables

List of abbreviations

ALCM_CYL	Constant Arc-Length Control Method (Cylindrical)
ALCM_FNP	Linear Arc-Length Control Method (Fixed Normal Plane)
ALCM_SPH	Constant Arc-Length Control Method (Spherical)
ALCM_UNP	Linear Arc-Length Control Method (Updated Normal Plane)
CALI	Cylindrical Arc-Length Increment (predictive technique)
CR	Corotational
CSP	Current Stiffness Parameter
EWI	External Work Increment (predictive technique)
FEM	Finite Element Method
GDCM	Generalized Displacement Control Method
GDI	General Displacement Increment (predictive technique)
GL	Green-Lagrange (strain tensor)
GSP	Generalized Stiffness Parameter
LCM	Load Control Method
LI	Load Increment (predictive technique)
LR2O	Large Rotation 2nd Order (geometric stiffness matrix)
LR4O	Large Rotation 4th Order (geometric stiffness matrix)
MNCM	Minimum Norm Control Method
ORCM	Orthogonal Residue Control Method
PKII	Second Piola Kirchhoff (stress tensor)
PMPE	Principle of Minimum Potential Energy
PVD	Principle of Virtual Displacements
PVF	Principle of Virtual Forces
PVW	Principle of Virtual Works
SALI	Spherical Arc-Length Increment (predictive technique)
SR2O	Small Rotation 2nd Order (geometric stiffness matrix)
TL	Total Lagrangian
UL	Updated Lagrangian
WCM	Work Control Method
WRM	Weighted Residual Method

The good thing about science is that it's true
whether or not you believe in it.
Neil deGrasse Tyson

1. Introduction

1.1. Initial Considerations

Mathematical models of real physical systems can be defined as the formulation of equations that express their essential characteristics in order to represent an idealization and simplification of reality. In the case of structural engineering, which deals with solid medium, the conditions that the mathematical model of a structural system must satisfy to properly represent the behavior of the real structure can be divided into three groups: equilibrium conditions, compatibility between displacements and deformations, and the material constitutive law (Martha, 2017).

The equilibrium conditions correspond to the balance of external and internal forces of the system. This balance must ensure the global equilibrium of the structure or any isolated portion. The condition of compatibility between displacements and deformations is expressed by kinematic equations. These are geometric relations that must be satisfied to ensure continuity of the structural model. That is, the deformed configuration of the structure must remain continuous, with no gaps, and compatible with its external links. Finally, the material constitutive law expresses a set of mathematical relations between stresses and strains through parameters that define how the material behaves at a macroscopic level. By considering these conditions, the equations that govern the behavior of the structure can be developed and solved by further imposing displacements (essential) and forces (natural) boundary conditions, which characterize the interaction of the system with the external environment.

In practice, the mathematical modelling of structural systems in the field of computational mechanics makes use of some numerical method of domain discretization. This means that the continuous model is discretized into a finite number of degrees-of-freedom, and the governing equations form an algebraic system whose unknowns are discrete values of the quantities associated to the degrees-of-freedom. The Finite Element Method (FEM) is mostly used for this purpose. In particular, the Direct Stiffness Method, a version of the FEM based on element stiffness to determine nodal displacements and/or rotations, is the most common strategy (Felippa, 2005).

A major concern regarding the development of the finite element equations is that any of the described conditions of the mathematical model can be formulated by linear or nonlinear relations, resulting in a discretized system of the same nature. Therefore, a decision must be made with respect to which formulation should be considered for the mathematical model.

In a linear formulation, several hypothesis are adopted to approximate the structural behavior and simplify the modelling and solution of the problem. The small displacement hypothesis considers that the displacements and rotations that the structure undergoes are negligible when compared to its dimensions. Consequently, equilibrium can be imposed on the initial (undeformed) configuration, and only linear terms are considered in the kinematic equations. The structural analysis with this consideration is called first-order analysis. The hypothesis that the relations of the constitutive law are linear equations with constant parameters results in a linear-elastic behavior of the materials. The material behavior is considered elastic when, after unloading the structure, it returns to its initial configuration with no residual deformations, and considered linear when stresses and strains are proportional. In addition, the natural and essential boundary conditions must be invariable for any level of external load, not depending on the deflections of the structure. If all these simplifying conditions are satisfied, the response of the structure becomes proportional to the applied solicitations, and the system of linear equations of the mathematical model can be solved analytically.

Most civil engineering structures behave in a linear-elastic fashion under service loads. However, prior to reaching their limit of resistance, almost all structures would exhibit significant nonlinear response (McGuire *et al.*, 2000). Therefore, it is not always possible to adopt such a simplified linear formulation, and a nonlinear formulation should be chosen. In structural mechanics, the nonlinear behavior may have a geometric, physical or boundary condition origin, depending on the sources of nonlinearities considered in the conditions of the mathematical model.

The geometric nonlinearity is a result of changes in the geometry of a structural system subjected to deflections that are relatively large when compared to the dimensions of its components. In this case, the consideration of large displacements and large rotations, as well as the inclusion of nonlinear strain terms to the kinematic equations, are required in the formulation of the finite element equations. These considerations are necessary to impose the equilibrium of the structural system on its deformed configuration, so that the nonlinear response of the structure geometry can be taken into account. This type of geometrically nonlinear analysis is also known as second-order analysis. Large deformations can also be considered in a geometrically nonlinear formulation, but it is usually associated with nonlinear material behavior as well.

The physical, or material, nonlinearity occurs when the constitutive relations between stresses and strains are not linear, and the behavior of the material depends on the past states of deformation. When the intensity and/or direction of the externally applied loads depend on the structure displacements, which is the case of follower loads, the problem is said to have nonlinearity in the forces, or natural, boundary conditions. The displacements, or essential, boundary conditions can also exhibit nonlinear behavior when they are displacement dependent, such as the case of contact problems.

A nonlinear analysis, of any type, aims at predicting the actual behavior of structures with more accuracy, as the formulation of the mathematical model involves fewer simplifications when compared to a linear formulation. However, the reason for not always considering the sources of nonlinearity, and when considered, not including all sources, is because it makes the modelling process harder and the solution much more expensive in a computational sense. The simulation of the nonlinear behavior involves numerical techniques to solve the system of equations by performing numerous linear analyses. Therefore, it is an important task for the analyst to decide which types of nonlinearity, if any, are relevant to be considered in the project.

As stated by Bathe (1996), in an actual engineering analysis, it is good practice that a nonlinear analysis of a problem is always preceded by a linear analysis. Based on the linear response, the analyst is able to predict which nonlinearities will be significant and how to account for these nonlinearities most appropriately.

1.2. Motivation

In order to make structural projects more and more economical, engineers have been using materials of higher resistance to reduce the consumption. The result is structures with increasingly slender and flexible elements. Such structures can be subjected to displacements of significant magnitude that, if not considered in the analysis, may give rise to undesirable second-order effects, such as buckling, that compromise the use of the structures and can lead to catastrophic failures. Therefore, to meet the safety and durability criteria, a second-order analysis becomes fundamental in some cases, especially in projects of steel frame models. This type of analysis involves numerical methods, and visualization of graphic results is essential to auxiliary its interpretation. For these reasons, the use of nonlinear analyses was greatly improved with the advent of graphic-interactive computational applications.

The numerical techniques for solving the nonlinear system of equations in a second-order analysis are performed incrementally. This methodology consists of starting the analysis from a known equilibrium solution, normally the undeformed configuration, and follow the behavior of the system until a desired level of solicitation is reached, as the applied actions are incremented in small linear steps to approximate the nonlinear response in that increment. The reason for this incremental approach is because the nonlinear response of the structure may present some critical points, associated with the second-order effects, where iterative numerical methods would not be able to detect with a single step analysis. Thus, the identification of these effects can be done by studying the history of equilibrium solutions, and one of the main goals of a geometrically nonlinear analysis is to obtain the equilibrium paths of the structure. These paths are curves of the variation of the equilibrium configuration of the structure with a given control parameter. A typical curve expresses the relation between the applied load and the displacement associated with a certain degree-of-freedom, and can assume very complex forms depending on the degree of nonlinearity of the solution.

Many incremental solution methods have been developed for capturing the nonlinear behavior of structures and tracing the equilibrium paths. Some of these methods are simpler and computationally more efficient, but limited. Others are more sophisticated and robust, being able to capture the response beyond critical points. However, as stated by Leon *et al*. (2011), one single solution method may not be capable of solving any general nonlinear problem. Dealing with numerical procedures, exposes us to problems of convergence and numerical stability that naturally occur in these analyses. Therefore, according to Bergan *et al*. (1978), a computer program for nonlinear analysis should possess several alternative algorithms for the solution of the nonlinear system of equations.

That said, the use of computer programs to perform nonlinear analyses is not a trivial task. In a linear-elastic analysis, the results provided by the program depend very little on the users knowledge about the solution methods. For frame models, the response is obtained analytically, within the mathematical assumptions of the model, and does not depend on any numerical parameter or mesh refinement of each beam element. On the other hand, the results of a nonlinear analysis depend on a series of input parameters related to the numerical solution techniques, besides the discretization of bars. For each problem, the analyst must select a set of algorithms and analysis parameters that in his/her judgment will perform the analysis without diverging or spending an excessive amount of computer time. The engineer can only base this choice on previous experiences with similar structures and on his/her knowledge of the available algorithms. Even for an experienced analyst this selection is difficult, and the alternative of running several different analyses is inefficient and time consuming. Thus, a nonlinear analysis demands a more in-depth understanding about the solution algorithms and nonlinear structural behavior.

Some techniques of automatic or self-adaptive analysis are available to improve the efficiency and the convergence of nonlinear problems, while minimizing the influence of user decisions, but do not completely overcome the difficulties. Such procedures, even when successful, do not actively promote the understanding of the structural behavior. An appropriate alternative is the use of an interactive-adaptive technique for the nonlinear analysis. Gattass & Abel (1983) presented an investigation of this technique for large-displacement analysis of framed structures. According to these authors, interactive-adaptive methods of analysis are those in which the parameters and algorithms are selected or changed by the user during the analysis itself. These strategies allow for the possibility of an extensive control over the solution process, providing an efficient way to monitor the results and to intervene when problems appear.

The lack of tools that meet the requirements of performing an efficient nonlinear analysis of structures, by offering multiple solution options and allowing for an interactive-adaptive analysis, while focusing on the educational aspect, being accessible to those with little experience, is what motivated this work.

1.3. Literature Review

Two distinct numerical processes are identified in a nonlinear analysis problem. The first is the formulation of the problem which, in the case of the displacement-based FEM, consists of obtaining a system of nonlinear equilibrium equations. The second is the solution of the system of nonlinear equations, mostly done by incremental and/or iterative methods. These processes are independent. That is, the formulation of the system of nonlinear equations is not associated with any solution method, and vice-versa. However, both of them are equally important for obtaining a consistent response of the structural problem. Thus, due attention should be given to each of them.

The period from 1960 to the mid-1980s was emblematic for researches on the formulation and solution of nonlinear structural engineering problems involving the FEM. Later advances, almost entirely, are based on the theories developed during this period. A historical background of some advances and accomplishments made in those decades is presented by Felippa (2004). In the following sections, the main works that contributed to the current state-of-the-art of the analysis of two-dimensional frame structures with geometric nonlinearity will be mentioned. In addition, computational tools with similar purposes of the one developed in this work will be described.

1.3.1. Nonlinear Formulation

The formulation of geometrically nonlinear analysis of structures depends on two aspects, the theory adopted for the mathematical model of structural elements and the kinematic description of motion to derive equilibrium equations. For frame models, made of beam elements, two theories are commonly employed to mathematically describe the flexural behavior of these elements, the Euler-Bernoulli theory and Timoshenko theory.

The Euler-Bernoulli bending theory (also known as Navier theory) originated from the early studies of beam elements by Leonhard-Euler and Jacob Bernoulli in the 18th century. This is the simplest theory for simulating flexural behavior of beam elements, in which element cross-sections remain plane and perpendicular to the longitudinal axis of the element after bending, as a result of disregarding shear deformations. It is the most widely used theory for both linear and nonlinear problems and it is implemented in most structural analysis software. Because of that, this theory will be adopted for the development of nonlinear equilibrium equations in this work.

On the other hand, Timoshenko theory (Timoshenko & Gere, 2009) differs in considering, even though in a simplified fashion, shear deformation of element cross-sections, providing better results for analyzing slender elements. It becomes relevant only for beams with a length to height ratio of less than 5 (Martha, 2018), including analyzes with geometric nonlinearity (Rodrigues, 2019). Since the effects of geometric nonlinearity are generally associated with slender elements, it was chosen not to cover this theory in this work. Due to its greater complexity, this theory has been studied by several researchers. For example, Friedman & Kosmatka (1993) presented a stiffness matrix that can be exactly integrated and is free of ‘shear-locking’. Schramm *et al*. (1994) derived a new general beam stiffness matrix that accounts for both bending and shear deflection. Pilkey *et al*. (1995) provides stiffness, geometric, and mass matrices for straight Timoshenko beams with arbitrarily shaped cross-sections and in an arbitrarily oriented coordinate system. Rangel & Martha (2019) presented unifying coefficients for Euler-Bernoulli and Timoshenko theories. In addition, several researchers have also sought higher order bending theories, such as Levinson (1981), Heyliger & Reddy (1988), Petrolito (1995), among many others.

In a geometrically nonlinear analysis, the formulation of equilibrium equations is generally based on a Total Lagrangian (TL), an Updated Lagrangian (UL) or a Corotational (CR) kinematic description of motion. These kinematic formulations are similar for finite deformation problems in continuum mechanics, with the only difference being the reference configuration adopted to describe the motion of the structural elements as they deform and change shape. The UL and the CR formulations are used in this work.

Total Lagrangian and Updated Lagrangian nonlinear formulations have historically received more attention. Several formulations for analysis with geometric nonlinearity of two-dimensional frame models were developed in this context. Among the first works, we can mention Martin (1966), Jennings (1968), Mallet & Marcal (1968), Powell (1969), and Ebner & Ucciferro (1972), who conducted a comparative study between the previous formulations. Epstein & Murray (1976) developed a formulation for large deformations analogous to that of shell elements, presented by Budiansky (1968).

Posteriorly, Bathe & Bolourchi (1979) compared both Lagrangian formulations for a three-dimensional beam element subjected to large displacements and rotations, but small deformations. In that work, it was verified the equivalence of the results for both formulations and it was concluded that the UL formulation is computationally more efficient for this type of problem than the TL formulation. Bathe (1996) further develops formulations in the Lagrangian framework for various finite elements, continuous and uniaxial, derived from the principles of continuum mechanics and an incremental approach of the Principle of Virtual Works.

Alves (1993a,b) presented and compared different formulations based on the TL and UL descriptions. Pacoste & Eriksson (1997) introduced UL formulations based on improved displacement-deformation relations with nonlinearity expressed by trigonometric functions, including Timoshenko beam theory. Galvão (2000) implemented these proposed geometrically nonlinear formulations for beam elements and made a comparative study of their computational efficiency and results. Neuenhofer & Filippou (1998) presented a force-based element for geometrically nonlinear analysis of plane frame structures. Rodrigues (2019), based on the work of Burgos & Martha (2013), developed geometric stiffness matrices considering high-order terms of the strain tensor and interpolating functions that take into account the axial load acting on the element, formulated for Euler-Bernoulli and Timoshenko theories, with the UL description. Other works include Wen & Rahimzadeh (1983), Chajes & Churchill (1987), Goto & Chen (1987), Wong & Tin-Loi (1990), Yang & Kuo (1994), Torkamani *et al*. (1997), and Nanakorn & Vu (2006).

The Corotational formulation, despite being an old concept, was the last to gain popularity in the analysis with geometric nonlinearity. Its use is especially widespread for analyses with finite motions but small strains of structural elements such as beams, plates and shells. This approach originated from the theorem of polar decomposition, which states that any general deformation can be uniquely decomposed into a rotation followed by a stretch component, or vice-versa.

Argyris *et al*. (1964) presented the first work applying the Corotational finite element formulation to beam elements. Subsequent works that used the Corotational approach to frame models gave it distinct names. Some of these works include Jennings (1968), Powell (1969), Belytschko & Hsieh (1973), who studied beam elements subjected to large rotations and proposed a method based on a curvilinear coordinate system called *convected coordinates*, and Belytschko & Glaum (1979). Many authors describe the idea by stating that it is based on a local Eulerian reference system attached to elements, such as Oran (1973) and Izzuddin & Elnashai (1993). A good description of the CR formulation and its relation to the more widely used TL and UL formulations is given by Mattiasson & Samuelsson (1984), and Mattiasson *et al*. (1985). Mattiasson & Samuelsson (1984) and Hsiao *et al*. (1999) emphasize that within the corotating system, either a TL or an UL formulation may be employed.

Great attention has been given to Corotational formulation since the 1990s. This formulation for beam elements, including Euler-Bernoulli and Timoshenko theories, is treated in several works, such as Crisfield (1991), who addressed transformation relations between the corotated and global systems. Pacoste & Eriksson (1995, 1997) covered the use of small displacements in the corotated system. Teh & Clarke (1998) compared the CR formulation to the Lagrangian formulations for three-dimensional structures. Souza (2000) presents a force-based formulation for inelastic large displacement analysis of planar and spatial frames. More recently, Kien (2012) developed a plane beam element for large displacements. Oliveira & Silva (2017) used a unified Bernoulli-Timoshenko element. Silva (2016) developed and implemented CR formulations for two-dimensional Euler-Bernoulli and Timoshenko beams. Some other examples of works employing this formulation are Iura (1994), Crisfield and Moita (1996), Krenk *et al*. (1999), Battini (2002), and Baião (2016).

1.3.2. Solution Methodology

In addition to a consistent and well-developed nonlinear formulation, robust and efficient solution methods for the system of equilibrium equations are indispensable. The most efficient methods for solving the nonlinear system of equations are based on an incremental approach, most of the times applied in conjunction with an iterative process.

The pioneer work on incremental nonlinear structural analysis was presented by Turner *et al*. (1960), in which a series of linear analysis steps was applied to a problem with non-uniform heating and large displacements, updating the stiffness matrix at the beginning of each increment to account for changes in the geometric configuration, internal forces, and temperature. In subsequent years, advances were focused on the formulation of the nonlinear finite element equations, and the purely incremental approach to solving these equations continued being used. This technique received the contribution of several researchers at the time, such as Goldberg & Richards (1963), Marcal (1965), and Hibbitt *et al*. (1970).

However, Mallet & Marcal (1968) and Murray & Wilson (1969) observed that the purely incremental methods may lead to anomalies in the solution and started using corrective iterations of the Newton-Raphson type to solve the nonlinear system at each incremental step, giving rise to the incremental-iterative methods. Zienkiewicz (1971) later presented a modified version of the Newton-Raphson iteration scheme, in which the tangent stiffness matrix is updated only at the beginning of each incremental step. By the late 1970s, these incremental-iterative methods enjoyed wide acceptance for geometrically nonlinear analyzes.

One of the greatest challenges in obtaining the complete response of a nonlinear analysis is to reach the solution beyond critical points. The corrective methods based on the conventional Newton-Raphson iteration is not able to do so, as it works only with displacement corrections within each incremental step. Batoz & Dhatt (1979) presented a method capable of passing through load limit points, in which the iterative cycle works only with corrective load increments. Modifications and generalizations of this method for post-buckling and collapse analysis was also presented by Bergan (1980), Powell & Simons (1981), and Bergan & Simons (1984). However this technique cannot capture snap-back (or displacement limit) points. To overcome this problem, incremental-iterative methods that work with corrective increments of both control variables (load and displacement), called continuation methods, have been developed since then.

The main idea of the continuation methods is to restrict the iterative solutions to a hyper-surface that crosses the equilibrium path at one or more points. The equation that defines this hyper-surface is called constraint equation and must be added to the system of finite element equilibrium equations. A method that uses an arc-length to restrict the iterative corrections of load and displacement received great attention from the scientific community and it was recognized by Meek & Tan (1984) as one of the most efficient. The development of this method and its variations is due to the works of Wempner (1971), Riks (1972), Riks (1979), Crisfield (1981), Ramm (1981), Ramm (1982), Crisfield (1991), and Crisfield (1997). A method based on a constant increment of external work to restrict the iterations was studied by Powell & Simons (1981), Bathe & Dvorkin (1983), and Yang & McGuire (1985). Chan (1988) introduced a constraint based on the minimum norm of the residual displacements. Yang & Shieh (1990) and Yang & Kuo (1994) proposed a method based on a generalized displacement to control the iterative corrections, in which a modified version was recently presented by Leon et *al*. (2014). A new solution strategy was later proposed by Krenk (1995) and Krenk & Hededal (1995), with the introduction of an orthogonality condition between the iteration residue and the corresponding increment of displacement.

The techniques for solving the system of nonlinear finite element equations of structural problems with geometric nonlinearity are in a satisfactory state of development nowadays, with robust and efficient methods for capturing the equilibrium path and crossing critical points. Crisfield (1997) discusses some numerical techniques to identify the presence of critical points in the solution. Silveira *et al*. (1999) presented a methodology for the implementation of incremental-iterative solution methods, in which a predicted solution is obtained first from a single linear analysis, and then the iterations of Newton-Raphson type are performed to correct the predicted solution until equilibrium is achieved. Rocha (2000) carried out a comparative study of several incremental-iterative solution methods. Maximiano (2012) introduced an alternative stabilization strategy of the orthogonal residue method proposed by Krenk (1995). Muñoz & Roehl (2017) proposed a continuation method with combined restrictions to obtain the full response in the presence of geometric nonlinearities and elasto-plastic softening.

1.3.3. Computational Tools

Computational programs that use the FEM for structural analysis can follow two lines of development, based on the objectives and target audience. Some are industry oriented while others have a more academic or educational appeal. The first group includes professional programs used in design offices that usually require a paid license. It is natural that these programs are not concerned with teaching structural behavior or the methods they use to solve the problem. Sometimes an educational version is available with a number of limitations, but these programs are mostly robust and hard to use. The second group are programs intended for teaching structural analysis to engineering students. These programs are simpler to use and focus on aspects of the structural behavior that may not be relevant for a professional project. Some of these programs are open-source and explore the implementation of the solution methods. A hybrid approach can also be established as long as it is possible to meet the needs of both audiences. Following are presented some programs, with educational or hybrid approaches, that can perform nonlinear analyzes of reticulated models, which is the case of the tool developed in this work.

The MASTAN2 (McGuire *et al.*, 2000) is a graphical-interactive structural analysis program developed in the MATLAB environment that provides pre-processing, analysis, and post-processing capabilities. The analysis routines provide the user the opportunity to perform first or second-order elastic or inelastic analyses of two or three-dimensional frames and trusses subjected to static loads. To solve the geometrically nonlinear problem, the program offers two single-step methods and two incremental-iterative methods. The first category includes the simple step (or Euler) method and a predictor-corrector method. The incremental-iterative techniques are the load control and the work control methods, with the option to choose between the standard or modified Newton-Raphson iterative schemes. The program also allows for an interactive-adaptive analysis, giving the option to resume the analysis at any step with new input parameters. MASTAN2 is the successor of a family of educational programs developed in Cornell University Computer-Aided Instructional Facility during the 80’s. Unfortunately, there are very few references, if any, available to these programs.

The CS-ASA (Computational System for Advanced Structural Analysis) was initially developed by Silveira (1995), using the Fortran 77 programming language, to investigate the elastic instability of columns and arches with unilateral contact restrictions. Many research projects have used this computational system as a basis for the development and validation of new nonlinear formulations and solution strategies, applied to static and dynamic analysis of steel structures. Among them are the works of Rocha (2000) and Galvão (2000), when several continuation solution methods were implemented, Galvão (2004) also developed routines for vibration analysis, Machado (2005) included inelastic formulations based on the plastic hinge method, and Silva (2009) unified the previous implementations. Prado (2012) built a graphical user interface for pre-processing steps using the IUP (Portable User Interface) system (Levy *et al.*, 1996), which is the same graphical interface development toolkit used in the application implemented for the current work.

The INSANE (Interactive Structural Analysis Environment), as described by Pitangueira & Fonseca (2007), is a platform for scientific computation that applies the FEM to structural analysis problems. The program later extended its applications to different areas, including fluid mechanics and heat transfer, as a result of many research projects. The types of structural models that can be analyzed include trusses, frames, plates, shells, plane stress and strain models, etc. Besides the basic analysis with static and linear-elastic assumptions, several advanced analysis options are available. Germanio (2005) developed a module for dynamic analysis. Fonseca (2008) added geometrically nonlinear analysis to the previous work. Melo (2017) implemented the Corotational formulation for the nonlinear analysis of two-dimensional beam elements. These nonlinear implementations make use of the work of Fuina (2004), who implemented incremental-iterative methods to solve the nonlinear system of equilibrium equations, which include controls of load, displacement, work, arc-length, orthogonal residue, generalized displacement, and deformations. These methods were complemented by Jean (2017), who included the arc-length control method based on the rates of internal and dissipated energy. The program also disposes of a graphical-interactive environment for pre, post, and processing steps of the analysis process. It is developed using the JAVA programming language, and all of its modules are implemented using the Object Oriented Programming (OOP) paradigm.

The AFA-OPSM (Advanced Frame Analysis – Ouro Preto School of Mines), by Santana (2015), is a graphical-interactive program intended to perform geometrically nonlinear static analysis of two-dimensional trusses and frames. This program was developed using the OOP paradigm and the graphical resources of MATLAB. It provides in an integrated way, the phases of modeling, analysis, and results visualization. The nonlinear solution methods include the strategies based on constant load increments, variations of the arc-length control, and the minimum norm of the residual displacements.

It is also worth mentioning the NLS++ library (Leon *et al*., 2011), developed in the C++ programming language, where several nonlinear solution methods are unified into a single space. The methods include load control, displacement control, work control, arc-length control, generalized displacement control, and the orthogonal residue procedure. The unified schemes are formulated and implemented such that additional nonlinear solution methods are readily incorporated and integration into a finite element analysis code is straightforward. Even though it is not a graphical program, the approach follows an educational philosophy. Several other libraries of nonlinear solution algorithms have been previously developed by Mondkar & Powell (1978), Clarke & Hancock (1990), Rezaiee-Pajand *et al*. (2009), among others.

The FEMOOP (Finite Element Method Object Oriented Program) system (Martha & Parente 2002) is a library for numerical analysis of structures develop in the Department of Civil Engineering of PUC-Rio in the early 90's. Its nonlinear analysis capabilities were implemented based on the work of Roehl (1987), which is one of the pioneering work on nonlinear analysis of structures within the mentioned department. This work considers the geometrically nonlinear analysis of three-dimensional frame models with end rotation release and distributed loads, based on the Updated Lagrangian formulation.

Some other interesting projects, whose focus is on the educational aspects of structural mechanics, are the FAST (Finite Element Analysis Tool) (Parente, 2018), SALT (Sistema de Análise de Estruturas) (Lima, 2017), AcadFrame (http://www.set.eesc.usp.br/softwares_depto/acadframe), and LESM (Linear Elements Structure Models) (Rangel & Martha, 2019).

The Ftool (Two-dimensional Frame Analysis Tool) program (Martha, 1999), which served as basis for developing the geometrically nonlinear analysis application in this work, has already received some contributions in the field of nonlinearity. Del Savio (2004) developed a version of the program with semi-rigid connections, where a rotational spring finite element was implemented to perform elastoplastic analyses. Silva (2017) continued previous implementations on the analysis and design of reinforced concrete elements, by introducing the design of concrete columns. In that work, the Two Cycles method (Chen & Lui, 1991) was included to execute a simplified geometrically nonlinear analysis. This method is based on performing a linear-elastic analysis followed by a geometric nonlinear analysis with the updated tangent stiffness matrix to correct the linear results. Silva *et al*. (2016) compared the use of this simple method with the standard Newton-Raphson iterative method and showed that the Two-Cycle scheme is capable of obtaining approximate results at a satisfactory level for designing reinforced-concrete structures with smooth nonlinearities that do not reach load limit points. Gomes (2019) complemented previous works on reinforced concrete design with the insertion of physical nonlinearity by changing the stiffness of element cross-sections to consider the effects of concrete cracking and creep, as well as the nonlinear relation between stresses and strains. Despite these advances, none of the versions of the Ftool program that include some source of nonlinearity was released, remaining as a tool for linear-elastic analysis of plane frame models.

1.4. Main Objectives and Contributions

This work describes the development and use of a graphical-interactive tool for geometrically nonlinear analysis of two-dimensional frame structural models, considering large displacements and large rotations, but small deformations in the elastic regime of the material behavior. Such models consist of beam elements with axial and flexural behavior in a single plane. In the developed tool, Euler-Bernoulli and Timoshenko beam theories are available for the flexural behavior (although, for simplicity, only Euler-Bernoulli formulation is presented in this document). This application was incorporated as a new feature of the Ftool (Two-dimensional Frame Analysis Tool) program (Martha, 1999), which is a largely used software in the Civil Engineering community and has demonstrated to be a valuable program for teaching structural analysis over the last years.

The development of this application aims to meet two main goals. The first one is focused on the academic community and seeks to provide an educational tool to improve users understandings about the second-order effects and the post-buckling behavior of framed structural models. It is also intended to increase their sensitivity on the use of the numerical methods to obtain the nonlinear response, allowing studies on the influence of the input parameters to the converged solution. Considering the difficulties imposed by the numerical solution of a nonlinear analysis, which is described in details throughout this work, it is required an easy-to-use application with intuitive resources to make it accessible not only to graduate students and to researchers, but also to undergraduate students. The Ftool program has the ideal environment for this intuitive nonlinear analysis, since the spread of its use was mainly due to the simplicity of the modeling and analyzing processes, and to the sophisticated data structure for pre and post-processing the analysis information.

The second goal is focused on the industry, by supplying structural engineers with a software that has a robust solver and allows users to have an extensive control of the nonlinear analysis process. Since no algorithm is able to solve all the nonlinear problems, this extensive analysis control brings for the analyst a greater possibility to obtain the full solution of any geometrically nonlinear problem and, therefore, to meet the safety and durability criteria of structural projects. The examples of this work prove the importance and the necessity of having a large number of settable options and parameters to perform and control nonlinear analyses.

To accomplish these goals, the developed tool has a user-friendly graphical interface designed to provide users with a wide range of analysis options, including the most well-known incremental single-step and incremental-iterative methods to solve the nonlinear system of equilibrium equations. An interesting feature of the developed application is the possibility to perform the nonlinear analysis in an interactive-adaptive fashion, by allowing the change of any of the analysis options and parameters between the incremental steps, as the analysis progresses. It is also possible to go back and forward in the analysis steps. These options for driving the analysis can help even experienced users to work with numerical algorithms, because when a non-converging point is found, one can change the parameters or use other solution methods, in the same analysis, to go beyond that point.

Furthermore, a sophisticated graph-plotting environment was developed, where users can create interactive graphs. Several options of information can be plotted in each axis, including the classical curve of displacement versus load, to study the equilibrium path of the structure. The graph data can also be selected to show the results for each incremental step or each iteration.

An auxiliary program, called NLframe2D, was also developed as a parallel implementation for this work using the MATLAB script language. The general structure of the code of this program is the same of the code developed for the nonlinear analysis module of the Ftool program. Since the MATLAB programming language has a very simple syntax, when compared to the C programming language, this auxiliary program is used to present the implementation of the solution process of the nonlinear structural problem considered in this work. In addition, the program is open-source and is available for download in:

https://www.mathworks.com/matlabcentral/fileexchange/73129-nlframe2d.

1.5. Outline

The remainder of work is divided into five chapters. The organization of the subjects follows the order: formulation, solution, implementation, and results. The next two chapters are devoted to the theory of geometrically nonlinear analysis of structures. The last three chapter are intended to present the developed tool, the provided results, and the conclusions.

In Chapter 2, the formulation of the nonlinear structural problem is developed to obtain the tangent stiffness matrix of the system of finite element equilibrium equations. Two types of nonlinear formulation are employed, based on the kinematic description of motion, the Updated Lagrangian formulation and the Corotational formulation. For each formulation, the local tangent stiffness matrix of a two-dimensional beam element that follows Euler-Bernoulli theory is obtained.

Chapter 3 starts by presenting the main reasons for the incremental approach in the solution of the nonlinear system of equilibrium equations, and discussing some important characteristics of the equilibrium path of structures, such as critical points and stability concepts. The overall goal of this chapter is to describe the general steps of the incremental solution process and present in details the theory behind the solution methods implemented in the developed tool. Two classes of methods are presented: incremental single-step and incremental-iterative methods.

Chapter 4 shows the developed tool, and it is divided into two main sections. The first section is intended to demonstrate the computer implementation of the solution process. The auxiliary program NLfram2D is used for this purpose. The MATLAB code of each solution step is provided and described. The second section is dedicated to presenting the new version of the Ftool program, showing the modifications from a user point of view, i.e., the new features in the graphical interface, their purposes and how to use them.

Chapter 5 is intended to validate the results obtained by the new version of the Ftool program, based on the analytical or numerical solutions of four benchmark problems, each one with a distinct nonlinear behavior. The robustness and efficiency of the implemented solution methods and nonlinear formulations are also investigated in the chapter. For this purpose, a study on the performance of the methods and formulations to solve the problems is carried out, taking into account the number of steps and iterations to reach the solution using different analysis options.

Chapter 6 brings the concluding remarks about the numerical results and reinforces some of the points that motivated this work. Suggestions for future works are also presented in the chapter.

Chapter 7 is the bibliography, with all references cited in the text.

2. Geometrically Nonlinear Formulations

2.1. Introduction

The equations of mathematical formulations of physical phenomena are commonly expressed as differential equations. In most cases, these are very complicated equations and/or are applied to complex domains, which makes an analytical solution impossible to obtain. To overcome these difficulties, numerical methods are invoked to discretize the continuous problem into a finite number of degrees-of-freedom, so the differential equations can be written as algebraic equations. The discretized algebraic equations are normally written as a matrix system, in which the coefficient matrix relates state variables to control variables. When the displacement-based Finite Element Method is applied to discretize a structural mechanics problem, the state variables are displacements and rotations of nodal points, while the control variables are external solicitations, such as loads. The system matrix, in this case, is a stiffness matrix that gives information about the necessary forces to impose certain displacements to the structure. In a geometrically nonlinear analysis, this matrix is called a tangent stiffness matrix because of its nonlinear nature, since its coefficients involve internal forces, obtained from the displacements. Therefore, the stiffness coefficients are not constants, varying from point to point in the solution, and the “tangent” term means the linear approximation at any point. The tangent stiffness matrix is composed by a material-dependent portion, called here as elastic stiffness matrix, and a geometry-dependent portion, called geometric stiffness matrix, responsible for accounting for changes in the stiffness of the system due to the large deflections.

This chapter is intended to formulate the geometrically nonlinear structural problem of two-dimensional frame models using the FEM to discretize it into beam elements, considering Euler-Bernoulli flexural behavior. The local tangent stiffness matrices of this type of element are derived. Two types of formulations, based on the kinematic descriptions of motion, are used: Updated Lagrangian formulation and Corotational formulation.

For the Updated Lagrangian formulation, three different types of geometric stiffness matrix are presented for beam elements, considering distinct terms of the strain tensor and degrees of sophistication of the shape functions. For the Corotational formulation, only one tangent stiffness matrix is developed. This matrix is based on the common consideration that the deformations of the beam with respect to a corotated configuration are small, so linear strain measurement is used.

2.2. Kinematic Descriptions of Motion

Consider a continuous body, composed of material points, or particles, that occupy different positions in space described by coordinates on a stationary Cartesian axis system. In a geometrically nonlinear analysis, this body is subjected to large displacements, rotations, and strains that change the position of its particles, as well as its superficial area and volume. Thus, the body 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 former focuses on the analysis of particles that pass through a fixed spatial coordinate, and is best suited for fluid media problems where the origin of the particles is unknown. In contrast, the latter maps the trajectory of all particles of the body, using material coordinates, from the beginning of the movement to the end. In this way, the movement of the body is characterized by the set of positons occupied by the particles, which is more appropriate for solid mechanics.

To formulate the finite element equations of the body, within the Lagrangian description of motion, a reference equilibrium configuration must be established for measuring stresses, strains, and all kinematic and static variables. In principle, any previously obtained equilibrium configuration can be used as the reference configuration. However, in practice, only the initial or the last obtained equilibrium configuration is used to formulate the problem because of the advantages they have over intermediate configurations. Based on the choice of the reference configuration, three kinematic descriptions are commonly used in structural mechanics to formulate the nonlinear system of equilibrium equations: Total Lagrangian (TL), Updated Lagrangian (UL), and Corotational (CR).

In the formulation based on the TL description of motion, the initial (undeformed) configuration of the analysis is taken as the reference configuration, which remains unchanged during the entire analysis process. Displacements, stresses and strains for formulating the equilibrium equations on the deformed (current) configuration are referred to the initial configuration of the analysis, as illustrated in Fig. 2.1.

Initially, the TL formulation was the most widely used in continuum-based FEM codes. Its primary applications are in solid mechanics with finite but moderate displacements and strains, especially for elastic materials, but not reliable for topology changes (Felippa, 2004).

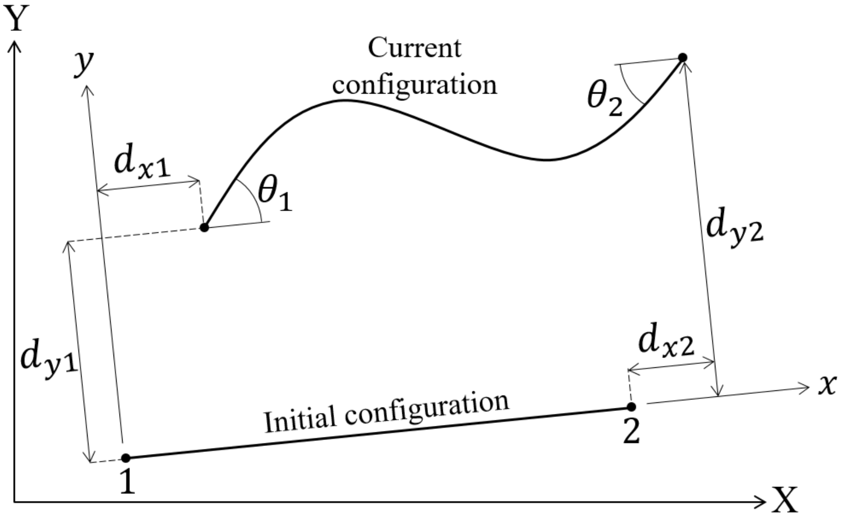


Figure 2.1 – Total Lagrangian reference configuration

In the UL formulation, the reference configuration is periodically updated to the last achieved equilibrium configuration. That is, after an equilibrium configuration is reached, it is desired to establish the equilibrium on a new deformed configuration, so all the static and kinematic variables for formulating the equilibrium equations are defined according to the previous configuration, as illustrated in Fig. 2.2.

The UL formulation is useful in treating finite displacements and possibly very large strains as well as in processes involving topology changes, such as fracture (Felippa, 2004). It is also computationally more effective for beam elements with large displacements/rotations and small deformations, when compared to the TL formulation, as shown by Bathe & Bolourchi (1979). Furthermore, the expressions of the formulation based on this kinematic description are simpler than the TL formulation.

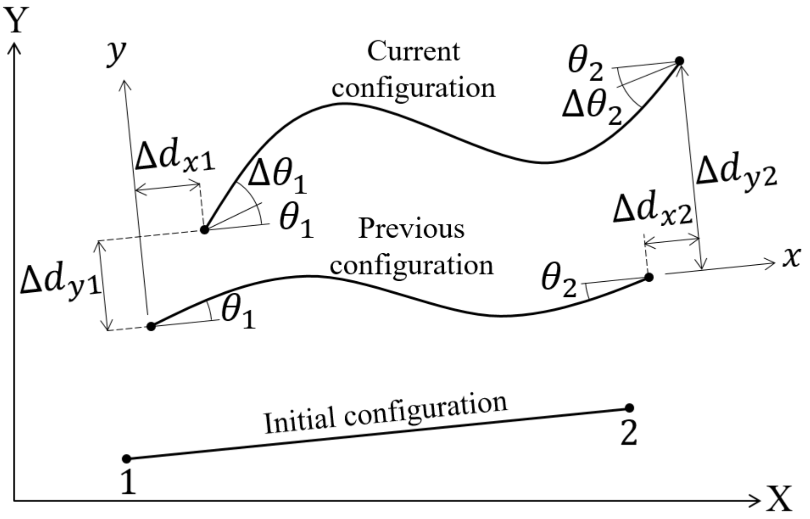


Figure 2.2 – Updated Lagrangian reference configuration

In the CR formulation, the reference configuration is divided into 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 corotated configuration is used to measure strains and stresses in the body. The corotated configuration uses a local coordinate system that moves along with the body, so that, with respect to this system, the rigid body movements are null. Figure 2.3 shows a schematic of this idea. This formulation is the latest among those used in the analysis with geometric nonlinearity and gained popularity for structural elements such as beams, plates and shells, especially for analyses with finite motions but small strains.

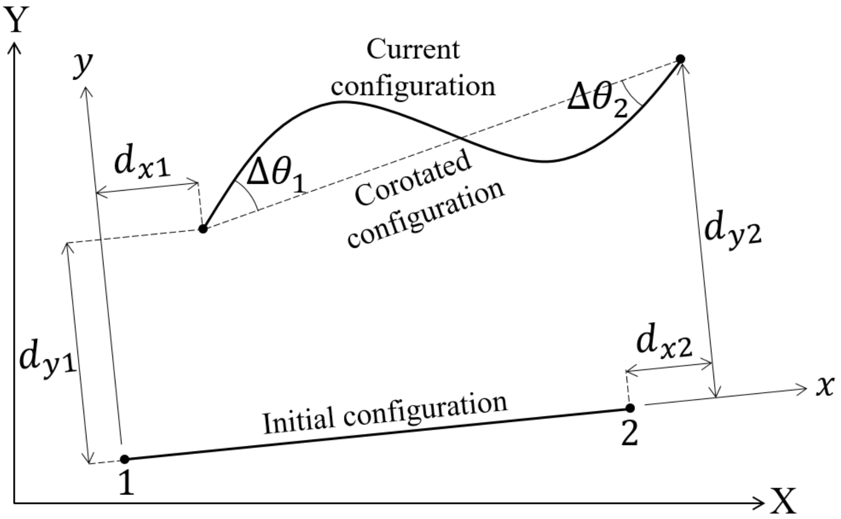


Figure 2.3 – Corotational reference configuration

Despite the differences in the development of the equilibrium equations, the numerical solution should provide identical results using formulations based on any kinematic description. 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 when compared to the TL formulation, as previously mentioned. These formulations are developed next for two-dimensional Euler-Bernoulli beam elements. In the next section, the development of the UL formulation starts from a continuous body to show its generality.

2.3. Updated Lagrangian Formulation

For the UL formulation, a pseudo-time variable, t, is introduced. This variable has nothing to do with a dynamic analysis, since it is considered that the loads are applied very slowly. That is, the load frequencies are much lower than the natural frequencies of the structure, so a static analysis can be assumed. The pseudo-time is used to describe the configurations of a body as it moves under load effects. A configuration of a generic three-dimensional body is defined by its geometry (surface area, S, and volume, V), specific mass, ρ, and position, described by the Cartesian coordinates, x_1, x_2, x_3, of each particle, at any pseudo-time.

The structure response is then defined as a series of equilibrium configurations obtained as the pseudo-time varies in increments, Δt, starting from zero. Given that all equilibrium configurations are known from the beginning of the analysis to time t, it is desired to establish the equilibrium at time $t + \Delta t$, with a system of equations formulated based on the previous configuration (or initial configuration in the case of the TL formulation).

Figure 2.4 illustrates the configurations involved, including the total and the incremental displacements (u_1, u_2, u_3) between them. The time of the configuration in which a quantity occurs is indicated by a superscript on the left side of the variable that represents this quantity. Eventually, a left subscript is also used to indicate the time of the reference configuration for the quantity to be measured, especially for stresses and strains measurements. If the quantity under consideration occurs in the same configuration in which it is also measured, the left subscript may not be used. Indicial notation is also adopted in the next sections, with indexes varying from 1 to 3, and considering the summation convention of repeated index.

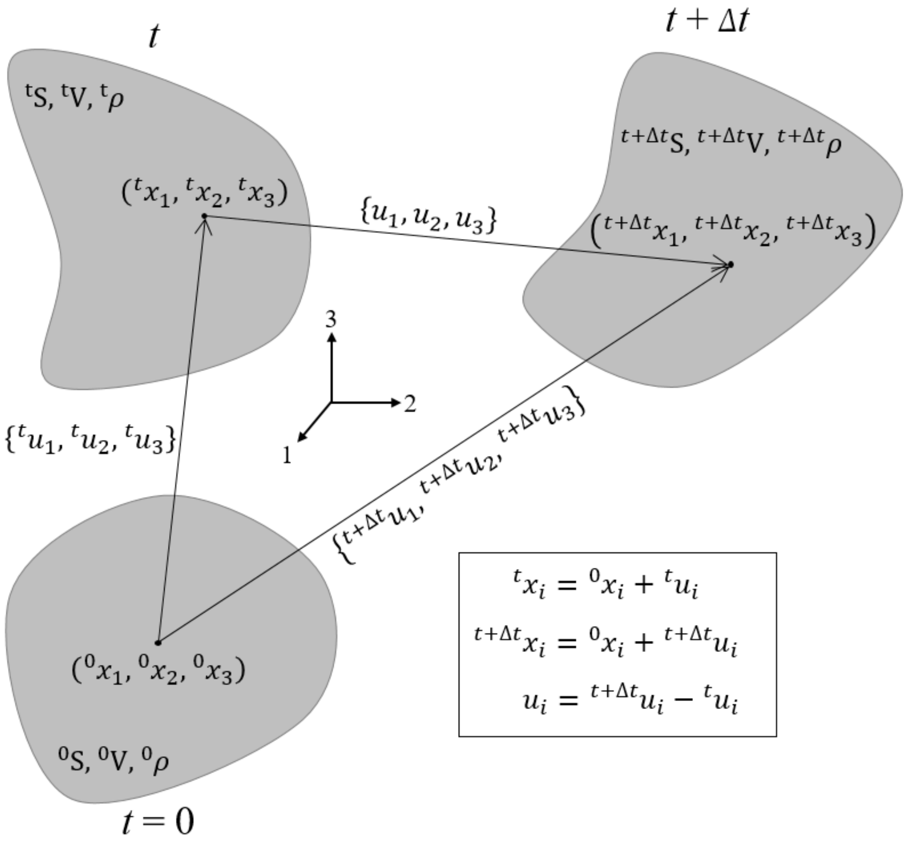

t
$t + \Delta t$
${}^{t}\mathrm{S}, {}^{t}\mathrm{V}, {}^{t}\rho$
$({}^{t}x_1, {}^{t}x_2, {}^{t}x_3)$
$\{u_1, u_2, u_3\}$
${}^{t+\Delta t}\mathrm{S}, {}^{t+\Delta t}\mathrm{V}, {}^{t+\Delta t}\rho$
$({}^{t+\Delta t}x_1, {}^{t+\Delta t}x_2, {}^{t+\Delta t}x_3)$
3
2
1
$\{{}^{t}u_1, {}^{t}u_2, {}^{t}u_3\}$
$\{{}^{t+\Delta t}u_1, {}^{t+\Delta t}u_2, {}^{t+\Delta t}u_3\}$
${}^{t}x_i = {}^{0}x_i + {}^{t}u_i$
${}^{t+\Delta t}x_i = {}^{0}x_i + {}^{t+\Delta t}u_i$
$u_i = {}^{t+\Delta t}u_i - {}^{t}u_i$
$({}^{0}x_1, {}^{0}x_2, {}^{0}x_3)$
${}^{0}\mathrm{S}, {}^{0}\mathrm{V}, {}^{0}\rho$
$t = 0$

Figure 2.4 – Variables for the Updated Lagrangian formulation

The development of the equilibrium equations of the displacement-based FEM lies on the application of the Principle of Virtual Works (PVW), specifically, the Principle of Virtual Displacements (PVD). In the sequence, the application of the PVD to a continuous body follows the work of Bathe (1996). Then the problem is particularized to a beam element and the FEM is used to discretize the beam into nodal degrees-of-freedom, so the local stiffness matrices can be obtained.

2.3.1. Principle of Virtual Works

The displacement-based finite element equations of the discrete mechanical problem can be formulated by different ways. The Principle of Virtual Displacements (PVD), the Principle of Minimum Potential Energy (PMPE), and the Galerkin version of the Weighted Residual Methods (WRM) can be employed. Martha (2018) shows that the energy-based methods of the PVD and the PMPE are both derived from the Galerkin version of the WRM, giving it a physical interpretation. In essence, all these methods are equivalent. The concepts of *virtual displacements* from the PVD, *variation* from the variational calculus used in the PMPE, and *weighted function* from the WRM, are all the same.

The energy-based methods are largely used in structural mechanics to formulate the equations of mathematical models. They are grounded in the Lagrangian mechanics rather than Newton’s classical mechanics that uses vector quantities and is not convenient for general and complex problems. The Lagrangian mechanics is an alternative and more formal manner of imposing equilibrium, compatibility, and constitutive conditions to the mathematical model of a structure. It is based on the Law of Conservation of Energy, a principle that can be applied to formulate mathematical models of physical systems in all scientific areas. This principle expresses the balance of internal and external energy (or work) of a physical system.

For the structural systems studied here, the only type of internal energy stored in the body is the elastic deformation energy, W_I, due to the work produced by internal stresses with the corresponding internal deformations. The external energy, W_E, is the work produced by the externally applied forces with the corresponding displacements. The loads are applied slowly so that no vibration occurs and kinetic energy is zero. In addition, it is also assumed that the material has an elastic behavior and there is no energy dissipation by means of heat, noise, etc. Therefore, according to the principle of energy conservation, the work produced by the externally applied forces to a structure is equal to the internal energy of deformation stored in the structure:

$$W_I = W_E \tag{2.1}$$

Based on this, it is possible to determine the displacement at the point and in the direction of a single force applied to the structure. However, this principle does not allow the calculation of displacements in a generic way. For example, if multiple forces are applied to the structure, or if the objective is to calculate a displacement at another point, a single equation – Eq. (2.1) – is not sufficient to determine more than one unknown displacement. The solution to this is the generalization of this principle to the Principle of Virtual Works (PVW). This is a very useful artifice for various applications in structural mechanics, involving the concept of a virtual work. The methods based on this concept have a generic character and are applicable to problems with elastic or inelastic behavior, structural problems with external solicitation of forces and moments, or thermal solicitations, as well as problems related to structural stability (Tauchert, 2007).

The idea behind the PVW is to work with two independent systems for the same structure (White *et al*., 1976):

- System *A*: System of forces with an external load field in equilibrium with internal stresses.
- System *B*: Deformed configuration with an external displacement field compatible with internal strains.

The generality of the PVW with respect to the principle of energy conservation is that there is no connection (cause-effect relation) between the force system *A* and the deformed configuration *B*. The equality between internal and external virtual works, combining both systems, results in the PVW. These works are called virtual because they are a mere mathematical abstraction, since the two systems are not related in a cause-effect sense. The internal virtual work, δW_I, which corresponds to the internal virtual energy of deformation stored in the structure, is produced by the internal stresses of system *A* with the corresponding strains of system *B*. The external virtual work, δW_E, is produced by the external forces of system *A* with the corresponding external displacements of system *B*:

$$\delta W_I = \delta W_{\mathrm{E}} \tag{2.2}$$

The PVW has two branches: the Principle of Virtual Forces (PVF) and the Principle of Virtual Displacements (PVD). In the PVF, the deformed configuration *B* is taken as the real structure, and a virtual force system *A* that satisfies equilibrium conditions is arbitrarily chosen to impose compatibility conditions to the real deformed configuration.

On the other hand, the PVD is used to impose equilibrium conditions to a real system of external forces and internal stresses *A*, from a virtual deformed configuration *B*, arbitrarily chosen, that satisfies compatibility conditions. Notice that, since the auxiliary virtual system is independent from the real system, the virtual field of external displacements need only to satisfy compatibility of configuration *B*. That is, the virtual displacement field does not have to satisfy compatibility of the real structure (configuration *A*), not even essential boundary conditions. The only restriction is that the virtual displacements must be compatible with the virtual strains. The PVD is the basis of the displacement-based FEM equations.

Alternatively, the symbol δ in Eq. (2.2) can also be interpreted, in the context of variational calculus, as a variation with respect to a perturbation in the displacement field. In this way, the internal virtual work corresponds to the variation of the internal energy of deformation, while the external virtual work is equivalent to the variation of the work produced by external forces. Therefore, the variation of the total potential energy of the structure, δΠ, which is defined as the difference between the variations of the internal energy and the work of external forces, is zero ($\delta\Pi = \delta W_I - \delta W_E = 0$). This interpretation corresponds to the PMPE, which states that the equilibrium configuration is the one that guarantees the stationarity of the total potential energy. This principle is largely used in structural mechanics, but it is restricted to conservative systems, i.e., non-dissipative systems in which the response is not path-dependent and there is no loss of energy (does not work for plasticity problems or dumped dynamical systems, for example). These are not limiting factors for the PVD and thus we rely on this general principle to develop the FEM equations, based on the UL formulation, even though the PMPE could be used for the type of problem dealt within this work.

Using the PVD, the internal virtual work at any configuration of the analysis is given by the product of the real stresses, τ_{ij}, by the virtual strains, δe_{ij}, integrated over the current volume of the body, as shown in Eq. (2.3). The real stresses are unknown and is taken as the Cauchy stress tensor, which represents the forces per unit of area at the current configuration. The virtual strains are the infinitesimal small strain tensor, compatible with a virtual displacement field, δu, imposed to the current configuration.

$$\delta W_I = \int_V \tau_{ij} \delta e_{ij} dV \tag{2.3}$$

The expression of the external virtual work at any configuration of the analysis, considering only loads as external actions, is given in Eq. (2.4). It has terms associated to the work of body forces, f^B, surface forces, f^S, and concentrated forces, f. Each of these terms is obtained with the product of real applied forces by the imposed virtual displacements. In the case of distributed forces, this product is the specific virtual work, which must be integrated over the current volume or surface of the body.

$$\delta W_E = \int_V f_i^B \delta u_i dV + \int_S f_i^S \delta u_i^S dS + f_i \delta u_i \tag{2.4}$$

The PVD states that the internal virtual work must be equal to the external virtual work for any arbitrary variation on the displacement field, i.e., imposed virtual displacements that are compatible with the virtual strains that contribute to the internal virtual work. This equality is used to impose the equilibrium and is valid for any configuration of the analysis. Assuming that all equilibrium configurations are known from the beginning of the analysis to a given time *t*, it is desired to establish the equilibrium at time *t* + Δ*t*. Therefore, the PVD is applied to this unknown configuration. The expressions of the internal and external virtual works at the new configuration can be written, respectively, as:

$$\delta\,{}^{t+\Delta t}W_I = \int_{{}^{t+\Delta t}V} {}^{t+\Delta t}\tau_{ij}\,\delta_{t+\Delta t}e_{ij}\,d\,{}^{t+\Delta t}V \tag{2.5}$$

$$\delta\,{}^{t+\Delta t}W_E = \int_{{}^{t+\Delta t}V} {}^{t+\Delta t}f_i^B\,\delta u_i\,d\,{}^{t+\Delta t}V + \int_{{}^{t+\Delta t}S} {}^{t+\Delta t}f_i^S\,\delta u_i^S\,d\,{}^{t+\Delta t}S + {}^{t+\Delta t}f_i\,\delta u_i \tag{2.6}$$

where the strain tensor components corresponding to the imposed virtual displacements are like the components of the infinitesimal strain tensor, but the derivatives are with respect to the current coordinates at time *t* + Δ*t*:

$$\delta_{t+\Delta t}e_{ij} = \frac{1}{2}\left(\frac{\partial\delta u_i}{\partial\,{}^{t+\Delta t}x_j} + \frac{\partial\delta u_j}{\partial\,{}^{t+\Delta t}x_i}\right) \tag{2.7}$$

We note that this is simply the application of the PVD as in a linear analysis, but considering a deformed configuration of the body, as a result of the large motions that change its position and shape. The Cauchy stresses are the real unknown quantities, and it is desired to determine them (left subscript is not needed because it occurs in the same configuration in which it is also measured). However, a fundamental difficulty in the general application of Eq. (2.2), using Eq. (2.5) and Eq. (2.6), is that the configuration of the body at time *t* + Δ*t* is unknown. This is an important difference compared to the linear analysis, in which it is assumed that the displacements are infinitesimally small so that the original configuration is used to perform the integrations in Eq. (2.5) and Eq. (2.6). Therefore, to apply the PVD to the deformed configuration, it is necessary to use appropriate stresses and strains measurements, as discussed in the next section.

2.3.2. General Formulation of Continuum Mechanics

To properly address the change in the configuration of a body, suitable stresses and strains measurements must be used. These measurements should be able to express the equilibrium of the body in a configuration not yet known in terms of a reference configuration already obtained. The objective is to write the internal virtual work with an integral over a volume that is known and to be able to incrementally decompose the stresses and strains in an effective manner.

The appropriate tensors for working with large displacements and large rotations are the Green-Lagrange (GL) strain tensor and the second Piola-Kirchhoff (PKII) stress tensor, both second-order tensors. To use these tensors, a reference configuration must be established. In the UL formulation, the reference configuration to measure a quantity that occurs at the current unknown configuration (time $t + \Delta t$) is the last configuration at which equilibrium was reached (time t). If the TL formulation were used, the reference configuration should be the beginning of the analysis (time $t = 0$).

The GL strain tensor, ε_{ij}, is a symmetrical tensor that measures only finite deformations of the body, neglecting rigid body displacements. In a linear-elastic analysis, the small strain tensor does not disregard rigid body rotations, which leads, for example, to internal forces in a rotating beam element. This effect is obviously not desired in an analysis with large deflections, so the GL strain tensor treats it well. The indicial expression of the GL tensor to measure finite strains at the current unknown configuration referred to the previous equilibrium configuration is provided by Eq. (2.8). Notice that all derivatives are with respect to the coordinates of the reference configuration of the body.

$$ {}^{t+\Delta t}_{\;\;t}\varepsilon_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial\, {}^{t}x_j} + \frac{\partial u_j}{\partial\, {}^{t}x_i} + \frac{\partial u_k}{\partial\, {}^{t}x_i}\frac{\partial u_k}{\partial\, {}^{t}x_j} \right) \tag{2.8} $$

The GL strain tensor, referred to the previous configuration, can be decomposed into a linear and a nonlinear incremental strain referred to the same configuration, at time t. The linear portion, e_{ij} in Eq. (2.9), corresponds to the infinitesimal strains, used in linear analyses. The nonlinear portion, η_{ij} in Eq. (2.10), is quadratic in displacement increments. Despite only using terms up to the second-order with respect to displacements, the GL strain tensor has no approximations, being valid for any level of deflections and deformations.

$$ {}_{t}e_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial^t x_j} + \frac{\partial u_j}{\partial^t x_i} \right) \tag{2.9} $$

$$ {}_{t}\eta_{ij} = \frac{1}{2}\frac{\partial u_k}{\partial^t x_i}\cdot\frac{\partial u_k}{\partial^t x_j} \tag{2.10} $$

The PKII stress tensor, S_{ij}, is energetically conjugated with the GL strain tensor, which allows both to be used together. Just like the GL tensor, the PKII tensor is symmetrical and invariant for rigid body displacements. It can be obtained from the Cauchy stress tensor, and its indicial expression to measure the stresses at the current unknown configuration referred to the previous equilibrium configuration is given in Eq. (2.11). Although it is used to measure stresses acting on a body, the components of this tensor have no physical meaning. It is only used to formulate the nonlinear problem, and it is not a result of interest in the analysis.

$$ {}^{t+\Delta t}_{t}S_{ij} = \frac{{}^{t}\rho}{{}^{t+\Delta t}\rho}\left(\frac{\partial^t x_i}{\partial^{t+\Delta t} x_m} \right)\left(\frac{\partial^t x_j}{\partial^{t+\Delta t} x_n} \right){}^{t+\Delta t}\tau_{mn} \tag{2.11} $$

Using the GL strain tensor and the PKII stress tensor, both referred to the previous equilibrium configuration of time t, the expression of the internal virtual work at the unknown configuration of time $t + \Delta t$, given in Eq. (2.5), can be rewritten as:

$$ \delta\,{}^{t+\Delta t}W_I = \int_{{}^{t}V} {}^{t+\Delta t}_{t}S_{ij}\,\delta\,{}^{t+\Delta t}_{t}\varepsilon_{ij}\,d\,{}^{t}V \tag{2.12} $$

This expression is equivalent to Eq. (2.5), but integrating over a known volume of the body to calculate the internal virtual work at the configuration of time $t + \Delta t$. In this expression, the stresses , ${}^{t+\Delta t}_{t}S_{ij}$, and strains, ${}^{t+\Delta t}_{t}\varepsilon_{ij}$, are unknown. Since the reference configuration for the GL and PKII tensors is the same, it is possible to decompose them into a known term of strains and stresses that occur at the configuration of time t, plus an unknown increment of these quantities in the interval $[t, t + \Delta t]$, also referred to the configuration of time t:

$$ {}^{t+\Delta t}_{t}\varepsilon_{ij} = {}^{t}_{t}\varepsilon_{ij} + {}_{t}\varepsilon_{ij} \tag{2.13} $$

$$ {}^{t+\Delta t}_{t}S_{ij} = {}^{t}_{t}S_{ij} + {}_{t}S_{ij} \tag{2.14} $$

The left superscript to indicate the configuration of the increments of GL strains, ${}_{t}\varepsilon_{ij}$, and PKII stresses, ${}_{t}S_{ij}$, is not needed and has been dropped.

It should be noted that the GL strains that occur at any configuration, referred to that same configuration, are null (${}^{t}_{t}\varepsilon_{ij} = 0$). The strain increment can be decomposed into the linear and quadratic portions with respect to the increments of displacements in the interval (${}_{t}\varepsilon_{ij} = {}_{t}e_{ij} + {}_{t}\eta_{ij}$), given in Eq. (2.9) and Eq. (2.10). In addition, the PKII stresses that occur at any configuration, referred to that same configuration, correspond to the Cauchy stresses (${}^{t}_{t}S_{ij} = {}^{t}\tau_{ij}$). Therefore, the incremental decomposition of strains and stresses can be expressed as:

$$ {}^{t+\Delta t}_{t}\varepsilon_{ij} = {}_{t}\varepsilon_{ij} = {}_{t}e_{ij} + {}_{t}\eta_{ij} \tag{2.15}$$

$$ {}^{t+\Delta t}_{t}S_{ij} = {}^{t}\tau_{ij} + {}_{t}S_{ij} \tag{2.16}$$

Replacing the incremental decomposition of strains and stresses into Eq. (2.12), we arrive at the expression for the internal virtual work at the unknown configuration of time *t* + Δ*t* in terms of increments of strains and stresses referred to the known configuration of time *t*, given in Eq. (2.17). Equation (2.18) is obtained by multiplying the terms.

$$ \delta\, {}^{t+\Delta t}W_{I} = \int_{{}^{t}V} \left({}^{t}\tau_{ij} + {}_{t}S_{ij} \right)\left(\delta_{t}e_{ij} + \delta_{t}\eta_{ij} \right) d\,{}^{t}V \tag{2.17}$$

$$ \delta\, {}^{t+\Delta t}W_{I} = \int_{{}^{t}V} {}^{t}\tau_{ij}\delta_{t}e_{ij}\, d\,{}^{t}V + \int_{{}^{t}V} {}^{t}\tau_{ij}\delta_{t}\eta_{ij}\, d\,{}^{t}V + \int_{{}^{t}V} {}_{t}S_{ij}\delta_{t}\varepsilon_{ij}\, d\,{}^{t}V \tag{2.18}$$

The external virtual work at the configuration of time *t* + Δ*t*, given in Eq. (2.6), can be obtained from the integral of the external forces over the volume and surface of any configuration of the analysis, since it is being considered that the loads do not vary with the geometry (linearity in natural boundary conditions). Therefore, this term, given in Eq. (2.19), will be treated the same as in a linear analysis by the FEM.

$$ \delta\, {}^{t+\Delta t}W_{E} = \int_{{}^{t}V} {}^{t+\Delta t}f_{i}^{B}\delta u_{i}\, d\,{}^{t}V + \int_{{}^{t}S} {}^{t+\Delta t}f_{i}^{S}\delta u_{i}^{S}\, d\,{}^{t}S + {}^{t+\Delta t}f_{i}\delta u_{i} \tag{2.19}$$

The general expression of the PVD application is given in Eq. (2.20), by equating the internal and external virtual works at the configuration of time *t* + Δ*t*. This equation is valid for any level of displacements, rotations and deformations that a continuous body undergoes since no approximation has been made to derive it.

$$ \int_{{}^{t}V} {}_{t}S_{ij}\delta_{t}\varepsilon_{ij}\, d\,{}^{t}V + \int_{{}^{t}V} {}^{t}\tau_{ij}\delta_{t}\eta_{ij}\, d\,{}^{t}V = \delta\, {}^{t+\Delta t}W_{E} - \int_{{}^{t}V} {}^{t}\tau_{ij}\delta_{t}e_{ij}\, d\,{}^{t}V \tag{2.20}$$

2.3.3. Linearization of the Principle of Virtual Works

The solution of the general equation of the PVD, presented in Eq. (2.20), cannot be obtained directly, as it is nonlinear with respect to the increments of displacements in the interval [t, $t + \Delta t$]. It must be linearized, disregarding high-order terms of displacement increments, in order to obtain a tangent approximation. In that equation, the variation of the linear and nonlinear portions of the GL strain increment are expressed, respectively, as:

$$\delta_t e_{ij} = \frac{1}{2}\left(\delta \frac{\partial u_i}{\partial\, {}^t x_j} + \delta \frac{\partial u_j}{\partial\, {}^t x_i} \right) \tag{2.21}$$

$$\delta_t \eta_{ij} = \frac{1}{2} \frac{\partial u_k}{\partial\, {}^t x_i} \cdot \delta \frac{\partial u_k}{\partial\, {}^t x_j} + \frac{1}{2} \delta \frac{\partial u_k}{\partial\, {}^t x_i} \cdot \frac{\partial u_k}{\partial\, {}^t x_j} \tag{2.22}$$

Now, it is possible to observe that the second integral on the left hand side of Eq. (2.20) is already linear in displacement increments. In that integral, the Cauchy stresses at time t are known and the variation of the nonlinear portion of the GL strains increment has two components, given in Eq. (2.22). In each component, the virtual term is determined and multiplies a real linear term.

The integral on the right hand side is also linear. Again, the Cauchy stresses at time t are known quantities as well as the variation of the linear portion of the GL strains increment, as seen in Eq. (2.21). Therefore, the only term of Eq. (2.20) that must be linearized is the first integral on the left hand side.

The increment of stresses are obtained from the constitutive law of the material. In this work, it consists of a constant fourth-order tensor, C_{ijrs}, that relates the second-order tensors of stresses and strains (linearity in constitutive relations). When the strain increments are considered very small, the following assumptions may be adopted (Yang & Kuo, 1994):

$${}_t S_{ij} = C_{ijrs}\, {}_t e_{rs} \tag{2.23}$$

$$\delta_t \varepsilon_{ij} = \delta_t e_{ij} \tag{2.24}$$

Replacing these relations into the first integral of Eq. (2.20), we arrive at the linearized equation of the PVD applied to configuration of time $t + \Delta t$ based on the UL formulation (referred to configuration of time t), for continuum mechanics:

$$\int_{{}^tV} C_{ijrs}\, {}_t e_{rs} \delta_t e_{ij}\, d\, {}^tV + \int_{{}^tV} {}^t\tau_{ij} \delta_t \eta_{ij}\, d\, {}^tV = \delta\, {}^{t+\Delta t}W_E - \int_{{}^tV} {}^t\tau_{ij} \delta_t e_{ij}\, d\, {}^tV \tag{2.25}$$

It should be noted that the integral on the right hand side is the internal virtual energy of deformation corresponding to the configuration of time *t*. Since this is an equilibrium configuration, the internal virtual energy is equal to the external virtual work of the same configuration. Therefore, Eq. (2.25) can be rewritten as:

$$\int_{{}^{t}V} C_{ijrs}\,{}_{t}e_{rs}\,\delta\,{}_{t}e_{ij}\,d\,{}^{t}V + \int_{{}^{t}V} {}^{t}\tau_{ij}\,\delta\,{}_{t}\eta_{ij}\,d\,{}^{t}V = \delta\,{}^{t+\Delta t}W_E - \delta\,{}^{t}W_E \tag{2.26}$$

The first integral, obtained by linearization, corresponds to the component of internal virtual energy of deformation that depends on the linear portion of the GL strain tensor, called here as the linear component of internal virtual energy, δW_I^L. The second integral provides the nonlinear component of internal virtual energy, δW_I^{NL}, which depends on the variation of the nonlinear portion of the GL strains.

It is important to mention that the linearized PVD equation corresponds to the linear system of incremental equilibrium equations resulting from the FEM discretization. By applying the FEM, as it will be done in a later section, the internal and external energy components of Eq. (2.26) can be expressed in terms of the vector of nodal displacements of the element, $\{U\}$. It turns that the linear and nonlinear components of internal virtual energy give origin to the elastic stiffness matrix, $[K_E]$, and geometric stiffness matrix, $[K_G]$, respectively. The external virtual work originates the vector of nodal loads, $\{P\}$, in the corresponding configuration. Equations (2.27) to (2.30) show the results of the finite element discretization of each energy component in terms of the nodal displacements of a generic element.

$$\delta W_I^L = \{\delta U\}^T [K_E] \{U\} \tag{2.27}$$

$$\delta W_I^{NL} = \{\delta U\}^T [K_G] \{U\} \tag{2.28}$$

$$\delta\,{}^{t+\Delta t}W_E = \{\delta U\}^T \left\{{}^{t+\Delta t}P\right\} \tag{2.29}$$

$$\delta\,{}^{t}W_E = \{\delta U\}^T \left\{{}^{t}P\right\} \tag{2.30}$$

Substituting these expressions of finite element discretization into the linearized PVD equation of the continuum mechanics, given by Eq. (2.26), we obtain a discrete PVD equation:

$$\{\delta U\}^T \left([K_E]+[K_G]\right)\{U\} = \{\delta U\}^T \left(\left\{{}^{t+\Delta t}P\right\} - \left\{{}^{t}P\right\}\right) \tag{2.31}$$

The virtual displacements are arbitrary and can be cancelled from both sides of the equation. The tangent stiffness matrix of the element, [K_T], is identified as the sum of the elastic and geometric stiffness matrices. Therefore, the linear system of incremental equilibrium equations, resulting from the FEM application to the linearized PVD equation, is given in Eq. (2.32), where {Δ*P*} is the increment of nodal loads between times *t* and *t* + Δ*t*.

$$\left[K_T\right]\left\{U\right\}=\left\{\Delta P\right\} \tag{2.32}$$

This linear system provides the increment of nodal displacements corresponding to an increment of nodal loads, assuming a tangent approximation to the nonlinear solution. The methods to solve the incremental system, including iterative techniques to get rid of the error caused by the linear approximation, are presented in the next chapter. The following sections are dedicated to obtaining the tangent stiffness matrix (elastic and geometric stiffness matrices) of a two-dimensional beam element. This is done by determining the linear and nonlinear components of internal virtual energy in terms of the continuous displacement fields of the beam element, and then applying the FEM to obtain the relations of Eq. (2.27) and Eq. (2.28).

2.3.4. Formulation for Two-Dimensional Beam Elements

The structural element of interest in this work is the beam element, with axial and transverse behavior in a single plane. The *xy* pair of axes will be designated to represent the longitudinal (axial) and transverse directions of the element. The displacement components in these directions, at any point of the element cross-section, are represented by the variables $u(x, y)$ and $v(x, y)$, and variables $u_0(x)$ and $v_0(x)$ for the displacement components at the cross-section center of gravity. The displacements of the center of gravity will be used to define the displacement field of the reticulated element. In addition, the cross-section rotation is represented by $\theta(x)$. Figure 2.5 illustrates the displacement field in the interior of a beam element, where *L* is the element length.

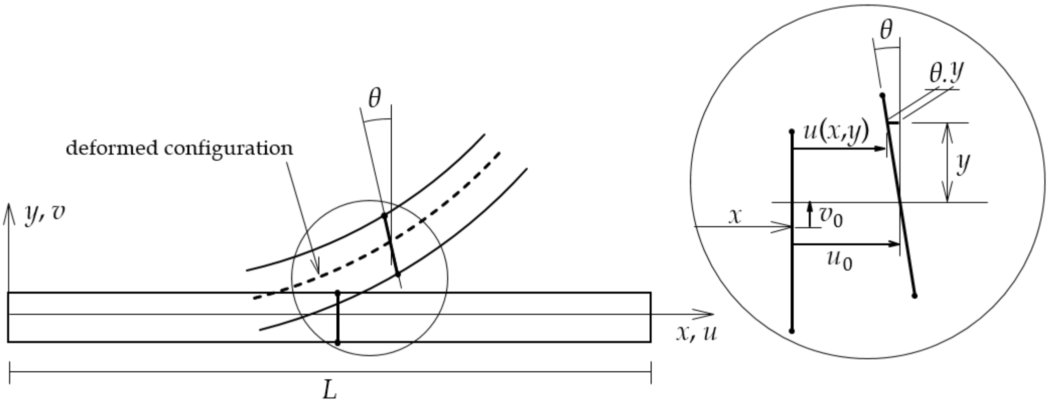


Figure 2.5 – Displacement field in the interior of a beam element (adapted from Martha, 2018)

According to Euler-Bernoulli beam theory, the rotation angle of the cross-section can be approximated by its tangent and it is related to the transverse displacement of the center of gravity as:

$$\theta(x) = \frac{\partial v_0(x)}{\partial x} \tag{2.33}$$

The result of this consideration is that plane cross-sections remain plane and normal to the element longitudinal axis after bending. Unlike Timoshenko beam theory, the shear distortion is neglected, which leads to an approximation that provides good results for slender elements. A comparison of both beam theories is illustrated in Fig. 2.6.

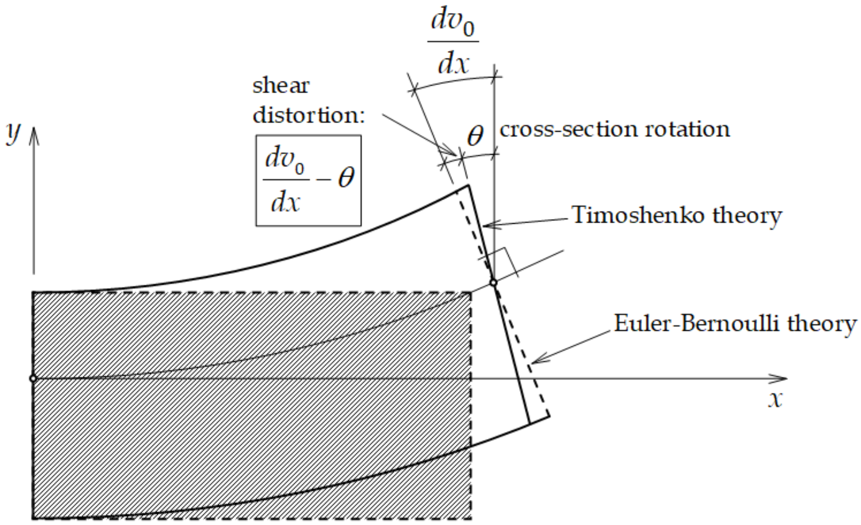


Figure 2.6 – Comparison between Euler-Bernoulli and Timoshenko bending theories (adapted from Martha, 2018)

Considering Euler-Bernoulli beam theory, the displacement field in the interior of the beam element can be expressed in terms of the axial and transverse displacements of the cross-section center of gravity as:

$$u(x,y) = u_0(x) - y \cdot \theta(x) = u_0(x) - y\frac{\partial v_0(x)}{\partial x} \tag{2.34}$$

$$v(x,y) = v_0(x) \tag{2.35}$$

The components of internal virtual energy of the linearized PVD equation, based on the UL formulation, will be determined for this type of element considering the stresses and strains present in an Euler-Bernoulli beam. Since all quantities in Eq. (2.26) are referred to the configuration of time *t*, the left indexes will be dropped to simplify the notation.

In two-dimensional frame elements, there are two components of the Cauchy stress tensor acting on the element cross-section, a normal and a shear component, and two associated components of Green-Lagrange strains. These stresses and strains components are written in vector form as shown below, with the strains decomposed into the linear and nonlinear portions.

$$\begin{Bmatrix} \tau_{xx} \\ \tau_{xy} \end{Bmatrix} \qquad \begin{Bmatrix} \varepsilon_{xx} \\ \varepsilon_{xy} \end{Bmatrix} = \begin{Bmatrix} e_{xx} \\ e_{xy} \end{Bmatrix} + \begin{Bmatrix} \eta_{xx} \\ \eta_{xy} \end{Bmatrix} \tag{2.36}$$

According to Eq. (2.9) and Eq. (2.10), the linear and nonlinear portions of the GL strains related to the axial and transverse displacements are:

$$e_{xx} = \frac{\partial u}{\partial x} \tag{2.37}$$

$$\eta_{xx} = \frac{1}{2}\left(\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial x}\right)^2\right) \tag{2.38}$$

$$e_{xy} = \frac{1}{2}\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) \tag{2.39}$$

$$\eta_{xy} = \frac{1}{2}\left(\frac{\partial u}{\partial x}\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\frac{\partial v}{\partial y}\right) \tag{2.40}$$

Replacing the axial and transverse displacement fields of the beam element, given in Eq. (2.34) and Eq. (2.35), into the previous expressions, the strains can be written in terms of the displacements of the cross-section center of gravity:

$$e_{xx} = \frac{\partial u_0}{\partial x} - y\frac{\partial^2 v_0}{\partial x^2} \tag{2.41}$$

$$\eta_{xx} = \frac{1}{2}\left(\left(\frac{\partial u_0}{\partial x}\right)^2 + \left(\frac{\partial v_0}{\partial x}\right)^2 + y^2\left(\frac{\partial^2 v_0}{\partial x^2}\right)^2 - 2y\frac{\partial u_0}{\partial x}\frac{\partial^2 v_0}{\partial x^2}\right) \tag{2.42}$$

$$e_{xy} = 0 \tag{2.43}$$

$$\eta_{xy} = \frac{1}{2}\left(y\frac{\partial^2 v_0}{\partial x^2}\frac{\partial v_0}{\partial x} - \frac{\partial v_0}{\partial x}\frac{\partial u_0}{\partial x}\right) \tag{2.44}$$

The constitutive tensor is represented by the 2x2 matrix that relates the vectors of stresses and strains components. For the case of a material with linear-elastic behavior, this constitutive matrix is shown in Eq. (2.45), where *E* is the elastic modulus and *G* is the shear modulus, which are constants during the analysis.

$$\begin{bmatrix} E & 0 \\ 0 & G \end{bmatrix} \tag{2.45}$$

The first integral on the left hand side of Eq. (2.26), is the term that provides the linear component of the internal virtual energy of deformation (component that depends on the linear portions of the GL strain tensor). Substituting Equations (2.36), (2.41), (2.42), (2.43), (2.44), and (2.45) into that integral, and recognizing that $\int_A dA = A$, $\int_A y dA = 0$, and $\int_A y^2 dA = I$, where *A* and *I* are respectively the area and moment of inertia of the cross-section, the linear component of internal energy can be developed as follows. A typical assumption of the small strain analysis is that the cross-section properties of area and moment of inertia remain constants.

$$\delta W_I^L = \int_V C_{ijrs} e_{rs} \delta e_{ij} dV = \int_V E e_{xx} \delta e_{xx} dV + \cancel{\int_V 4G e_{xy} \delta e_{xy} dV} \tag{2.46}$$

$$\delta W_I^L = \int_A \int_0^L E \left(\frac{\partial u_0}{\partial x} - y \frac{\partial^2 v_0}{\partial x^2} \right) \left(\delta \frac{\partial u_0}{\partial x} - y \delta \frac{\partial^2 v_0}{\partial x^2} \right) dx dA \tag{2.47}$$

$$\delta W_I^L = \int_0^L EA \frac{\partial u_0}{\partial x} \delta \frac{\partial u_0}{\partial x} dx + \int_0^L EI \frac{\partial^2 v_0}{\partial x^2} \delta \frac{\partial^2 v_0}{\partial x^2} dx \tag{2.48}$$

The nonlinear component of internal virtual energy (component that depends on the variation of the nonlinear portions of the GL strain tensor) is provided by the second integral on the left hand side of Eq. (2.26). By decomposing the stresses and strains of that integral into axial and shear components, as in Eq. (2.36), we obtain:

$$\delta W_I^{NL} = \int_V \tau_{ij} \delta \eta_{ij} dV = \int_V \tau_{xx} \delta \eta_{xx} dV + \int_V \tau_{xy} \delta \eta_{xy} dV \tag{2.49}$$

This expression can be developed by including or not the second-order gradients of the nonlinear portions of the GL strain tensor. The result of this consideration will be reflected in the degree of sophistication of the element geometric stiffness matrix. Disregarding the high-order terms of the nonlinear portions of GL strains, the following approximations are adopted:

$$\eta_{xx} = \frac{1}{2} \left(\left(\frac{\partial u_0}{\partial x} \right)^2 + \left(\frac{\partial v_0}{\partial x} \right)^2 \right) \qquad \eta_{xy} = 0 \tag{2.50}$$

Using these approximations in Eq. (2.49), we obtain a simplified expression for the nonlinear component of internal energy:

$$\delta W_I^{NL} = \int_A \int_0^L \tau_{xx} \delta \left(\frac{1}{2} \left(\left(\frac{\partial u_0}{\partial x} \right)^2 + \left(\frac{\partial v_0}{\partial x} \right)^2 \right) \right) dx dA \tag{2.51}$$

Otherwise, if all terms of the nonlinear portions of GL strains are considered, as in Eq. (2.42) for axial strains and Eq. (2.44) for shear strains, the expression of the nonlinear component of internal energy becomes:

$$\begin{aligned} &\delta W_I^{NL} = \\ &\int_A \int_0^L \tau_{xx} \delta \left(\frac{1}{2} \left(\left(\frac{\partial u_0}{\partial x} \right)^2 + \left(\frac{\partial v_0}{\partial x} \right)^2 + y^2 \left(\frac{\partial^2 v_0}{\partial x^2} \right)^2 - 2y \frac{\partial u_0}{\partial x} \frac{\partial^2 v_0}{\partial x^2} \right) \right) dx dA + \\ &\int_A \int_0^L \tau_{xy} \delta \left(\frac{1}{2} \left(y \frac{\partial^2 v_0}{\partial x^2} \frac{\partial v_0}{\partial x} - \frac{\partial v_0}{\partial x} \frac{\partial u_0}{\partial x} \right) \right) dx dA \end{aligned} \tag{2.52}$$

Consider the equilibrium relations of Eq. (2.53), where *P*, *Q*, and *M* are the resulting axial force, shear force, and bending moment in the element cross-section. Then, the development of the simplified and high-order expressions of the nonlinear component of internal energy results in Eq. (2.54) and Eq. (2.55), respectively.

$$\int_A \tau_{xx} dA = P \qquad \int_A \tau_{xy} dA = Q \qquad \int_A \tau_{xx}(-y)\, dA = M \tag{2.53}$$

$$\delta W_I^{NL} = \frac{1}{2}\int_0^L \left(P \cdot \delta\left(\left(\frac{\partial u_0}{\partial x}\right)^2 + \left(\frac{\partial v_0}{\partial x}\right)^2 \right)\right) dx \tag{2.54}$$

$$\begin{gathered} \delta W_I^{NL} = \\ \frac{1}{2}\int_0^L \left(P \cdot \delta\left(\left(\frac{\partial u_0}{\partial x}\right)^2 + \left(\frac{\partial v_0}{\partial x}\right)^2 \right) + \frac{PI}{A}\cdot\delta\left(\frac{\partial^2 v_0}{\partial x^2}\right)^2 \right) dx + \\ \int_0^L \left(M \cdot \delta\left(\frac{\partial^2 v_0}{\partial x^2}\frac{\partial u_0}{\partial x}\right) - Q\cdot\delta\left(\frac{\partial v_0}{\partial x}\frac{\partial u_0}{\partial x}\right)\right) dx \end{gathered} \tag{2.55}$$

2.3.5. Finite Element Discretization

The use of the Finite Element Method aims to discretize the continuous displacement field of the elements into nodal degrees-of-freedom. Each node of a 2D beam element has three degrees-of-freedom, two translations and one rotation. Figure 2.7 indicates the six nodal displacements of a beam element (translations d'_{x1}, d'_{y1} , d'_{x2}, d'_{y2}, and rotations θ_1, θ_2) that moves and bends in a single plane, in the directions of the degrees-of-freedom in the local axis system of the element. The local displacements are identified by a single quote mark and they are grouped in vector $\{d_l\}$ of Eq. (2.56). Notice that the rotation components about the out-of-plane *z*-axis are the same for local and global axis systems.

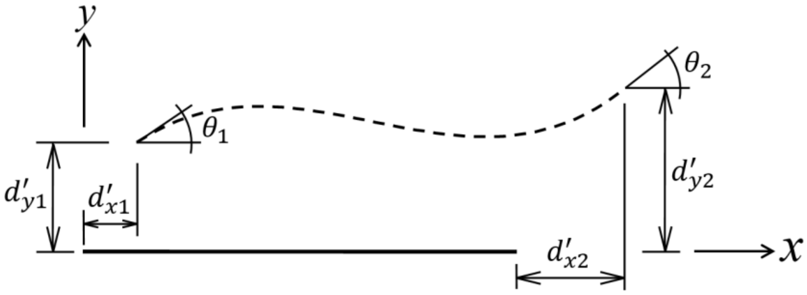


Figure 2.7 – Nodal displacements of a plane beam element in its local axis system

$$\{d_l\} = \{ d'_{x1} \quad d'_{y1} \quad \theta_1 \quad d'_{x2} \quad d'_{y2} \quad \theta_2 \}^T \tag{2.56}$$

These nodal displacements are split into axial and flexural components. Axial displacements act in local *x*-axis direction and they are grouped in vector $\{u\}$ of Eq. (2.57). Flexural displacements are the transverse components that act in local *y*-axis direction and the in-plane rotations, which are grouped in vector $\{v\}$ of Eq. (2.58).

$$\{u\} = \{d'_{x1} \quad d'_{x2}\}^T \tag{2.57}$$

$$\{v\} = \{d'_{y1} \quad \theta_1 \quad d'_{y2} \quad \theta_2\}^T \tag{2.58}$$

Shape functions are used to interpolate the displacements and rotation along the length of the element with the nodal values. Each interpolating shape function, $N_i(x)$, represents the deformed configuration of the element when a unit displacement or rotation is imposed to the corresponding degree-of-freedom. Therefore, the axial and transverse displacements of the cross-section center of gravity can be written in terms of the nodal values and interpolating functions, as illustrated in Fig. 2.8 for each elementary deformed configuration. The nodal values of displacements and rotations are the unknowns of the discrete problem, since the shape functions are pre-defined for each element type.

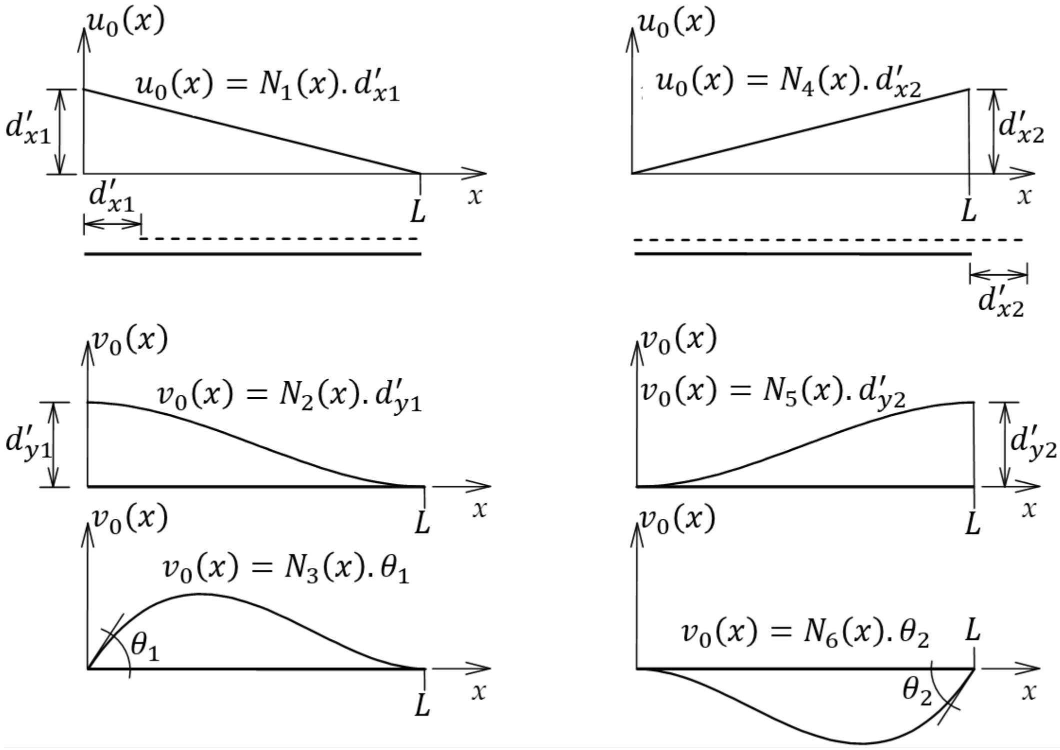


Figure 2.8 – Interpolation of elementary deformed configurations

The interpolation of the displacement fields of the beam element is obtained by superposition of the elementary deformed configurations:

$$u_0(x) = N_1(x) \cdot d'_{x1} + N_4(x) \cdot d'_{x2} \tag{2.59}$$

$$v_0(x) = N_2(x) \cdot d'_{y1} + N_3(x) \cdot \theta_1 + N_5(x) \cdot d'_{y2} + N_6(x) \cdot \theta_2 \tag{2.60}$$

These interpolations are written in vector form as follows:

$$u_0(x) = \{N_1(x) \quad N_4(x)\} \begin{Bmatrix} d'_{x1} \\ d'_{x2} \end{Bmatrix} = \{N_u\}\{u\} \tag{2.61}$$

$$v_0(x) = \{N_2(x) \quad N_3(x) \quad N_5(x) \quad N_6(x)\} \begin{Bmatrix} d'_{y1} \\ \theta_1 \\ d'_{y2} \\ \theta_2 \end{Bmatrix} = \{N_v\}\{v\} \tag{2.62}$$

In the case of beam elements, the shape functions usually employed to interpolate the element displacements are obtained from the homogeneous solution of the differential equations that govern the axial and flexural behaviors of the element. In a first-order analysis, the differential equation of beam elements, formulated for an undeformed infinitesimal element, is a second-order equation for the axial behavior and a fourth-order equation for the flexural behavior. Therefore, the shape functions are linear for axial displacements interpolation and cubic for flexural displacements interpolation (also known as Hermitian functions). The results for the axial shape functions, which do not depend on the adopted bending theory, are:

$$N_1(x) = 1 - \frac{x}{L} \tag{2.63}$$

$$N_4(x) = \frac{x}{L} \tag{2.64}$$

The Hermitian shape functions for interpolating flexural displacements of beam elements that follow Euler-Bernoulli bending theory are:

$$N_2(x) = 1 - \frac{3}{L^2}x^2 + \frac{2}{L^3}x^3 \tag{2.65}$$

$$N_3(x) = x - \frac{2}{L}x^2 + \frac{1}{L^2}x^3 \tag{2.66}$$

$$N_5(x) = \frac{3}{L^2}x^2 - \frac{2}{L^3}x^3 \tag{2.67}$$

$$N_6(x) = -\frac{1}{L}x^2 + \frac{1}{L^2}x^3 \tag{2.68}$$

Since the shape functions represent the exact solution of the differential equations of beam elements, no approximation is made in the interpolation of the displacement fields to the nodal points, providing analytical results for first-order analyses. Therefore, in a linear analysis, the discretization of beam elements into multiple sub-elements is not needed because it does not change the results.

In a geometrically nonlinear analysis, these interpolating functions do not represent the exact deformed configuration, as they are the solution of differential equations formulated for small displacements. In this case, elements refinement are necessary to converge for better results. However, if the differential equation of bending behavior is formulated considering the equilibrium of an infinitesimal element in its deformed configuration, the resulting shape functions can better represent the nonlinear solution by including the internal axial force in their expressions. Rodrigues (2019) developed shape functions based on this assumption for Euler-Bernoulli and Timoshenko bending behaviors. These complete shape functions are not reproduced here due to their complexity and size, but can be found in the referred work.

Based on the presented expressions of internal energy and shape functions, the local tangent stiffness matrix of an Euler-Bernoulli beam element is obtained in the following sections. The tangent stiffness matrix is composed of a linear (or elastic) portion, which depends on the elastic properties of the elements, and a geometric portion that depends on the elements internal forces to account for geometry changes. The elastic stiffness matrix is the same used in a linear-elastic analysis. The development of the geometric stiffness matrix takes into account the degree of sophistication of the adopted shape functions and the inclusion, or not, of high-order terms of the nonlinear portions of the GL strain tensor to the expression of the virtual energy of deformation. Accordingly, different forms of the geometric matrix can be obtained, as shown in the sequence.

The local tangent stiffness matrix of each beam element is used to assemble the global stiffness matrix of the structure. The assembly process is identical to a linear analysis, using the Direct Stiffness Method to rotate each local matrix to the global axis system and insert the stiffness coefficients to the correct position of the global tangent stiffness matrix. This process is slightly covered in Section 4.2.2, but details can be found in almost any book on matrix structural analysis.

2.3.5.1. Elastic Stiffness Matrix

The elastic stiffness matrix is obtained from the linear component of virtual internal energy, given in Eq. (2.48). Applying the interpolation of axial and transverse displacements, as in Eq. (2.61) and Eq. (2.62), it is possible to rewrite the expression of virtual internal energy in terms of the vectors of shape functions and nodal displacements:

$$\begin{aligned} &\delta W_I^L = \\ &\{\delta u\}^T \left[\int_0^L EA \frac{\partial\{N_u\}}{\partial x} \frac{\partial\{N_u\}^T}{\partial x} dx \right] \{u\} + \\ &\{\delta v\}^T \left[\int_0^L EI \frac{\partial^2\{N_v\}}{\partial x^2} \frac{\partial^2\{N_v\}^T}{\partial x^2} dx \right] \{v\} \end{aligned} \tag{2.69}$$

Using the shape functions given in Eq. (2.63) to Eq. (2.68) and integrating the terms inside the brackets of Eq. (2.69), the elastic stiffness coefficients are obtained. The first integral results in the stiffness coefficients related to axial degrees-of-freedom, while the second integral gives the stiffness coefficients of flexural degrees-of-freedom. The complete elastic stiffness matrix of an Euler-Bernoulli beam element is shown next, including all degrees-of-freedom:

$$[k_E] = \begin{bmatrix} \frac{EA}{L} & 0 & 0 & -\frac{EA}{L} & 0 & 0 \\ 0 & \frac{12EI}{L^3} & \frac{6EI}{L^2} & 0 & -\frac{12EI}{L^3} & \frac{6EI}{L^2} \\ 0 & \frac{6EI}{L^2} & \frac{4EI}{L} & 0 & -\frac{6EI}{L^2} & \frac{2EI}{L} \\ -\frac{EA}{L} & 0 & 0 & \frac{EA}{L} & 0 & 0 \\ 0 & -\frac{12EI}{L^3} & -\frac{6EI}{L^2} & 0 & \frac{12EI}{L^3} & -\frac{6EI}{L^2} \\ 0 & \frac{6EI}{L^2} & \frac{2EI}{L} & 0 & -\frac{6EI}{L^2} & \frac{4EI}{L} \end{bmatrix} \tag{2.70}$$

2.3.5.2. Geometric Stiffness Matrices

The geometric stiffness matrix is originated from the nonlinear component of virtual internal energy, using the simplified expression of Eq. (2.54) or the high-order expression, which considers all terms of the Green-Lagrange strains, shown in Eq. (2.55). In both formulations, it is possible to adopt the Hermitian or the complete shape functions to interpolate the field of transverse displacements, giving rise to different geometric matrices.

Considering the simplified expression of internal energy, the interpolation of axial and transverse displacements can be applied to Eq. (2.54), so the nonlinear component of virtual internal energy is expressed in terms of the vectors of shape functions and nodal displacements:

$$
\begin{aligned}
&\delta W_I^{NL} = \\
&\{\delta u\}^T \left[\int_0^L P \frac{\partial \{N_u\}}{\partial x} \frac{\partial \{N_u\}^T}{\partial x} dx \right] \{u\} + \\
&\{\delta v\}^T \left[\int_0^L P \frac{\partial \{N_v\}}{\partial x} \frac{\partial \{N_v\}^T}{\partial x} dx \right] \{v\}
\end{aligned}
\tag{2.71}
$$

Doing the same for the high-order expression of internal energy, Eq. (2.55) can be expressed in terms of shape functions and nodal displacements as:

$$
\begin{aligned}
&\delta W_I^{NL} = \\
&\{\delta u\}^T \left[\int_0^L P \frac{\partial \{N_u\}}{\partial x} \frac{\partial \{N_u\}^T}{\partial x} dx \right] \{u\} + \\
&\{\delta v\}^T \left[\int_0^L P \frac{\partial \{N_v\}}{\partial x} \frac{\partial \{N_v\}^T}{\partial x} dx \right] \{v\} + \\
&\{\delta v\}^T \left[\int_0^L \frac{PI}{A} \frac{\partial^2 \{N_v\}}{\partial x^2} \frac{\partial^2 \{N_v\}^T}{\partial x^2} dx \right] \{v\} + \\
&\{\delta v\}^T \left[\int_0^L M \frac{\partial^2 \{N_v\}}{\partial x^2} \frac{\partial \{N_u\}^T}{\partial x} dx \right] \{u\} + \\
&\{\delta u\}^T \left[\int_0^L M \frac{\partial \{N_u\}}{\partial x} \frac{\partial^2 \{N_v\}^T}{\partial x^2} dx \right] \{v\} + \\
&\{\delta v\}^T \left[\int_0^L Q \frac{\partial \{N_v\}}{\partial x} \frac{\partial \{N_u\}^T}{\partial x} dx \right] \{u\} - \\
&\{\delta u\}^T \left[\int_0^L Q \frac{\partial \{N_u\}}{\partial x} \frac{\partial \{N_v\}^T}{\partial x} dx \right] \{v\}
\end{aligned}
\tag{2.72}
$$

Three types of geometric matrix are used in this work, based on the simplified or high-order expressions of internal energy: Small Rotation 2nd Order (SR2O), Large Rotation 2nd Order (LR2O), and Large Rotation 4th Order (LR4O).

The SR2O geometric stiffness matrix is originated from the simplified expression of internal energy, using the Hermitian shape functions for interpolating flexural displacements. This matrix, resulting from the integration of axial and flexural terms of Eq. (2.71), is given in Eq. (2.73). This is the simplest and most usual geometric stiffness matrix for two-dimensional frame elements with Euler-Bernoulli bending behavior, which considers only axial force effects.

$$[k_G]_{SR2T} = \begin{bmatrix} \frac{P}{L} & 0 & 0 & -\frac{P}{L} & 0 & 0 \\ 0 & \frac{6P}{5L} & \frac{P}{10} & 0 & -\frac{6P}{5L} & \frac{P}{10} \\ 0 & \frac{P}{10} & \frac{2PL}{15} & 0 & -\frac{P}{10} & -\frac{PL}{30} \\ -\frac{P}{L} & 0 & 0 & \frac{P}{L} & 0 & 0 \\ 0 & -\frac{6P}{5L} & -\frac{P}{10} & 0 & \frac{6P}{5L} & -\frac{P}{10} \\ 0 & \frac{P}{10} & -\frac{PL}{30} & 0 & -\frac{P}{10} & \frac{2PL}{15} \end{bmatrix} \tag{2.73}$$

The LR2O matrix, used by Chen (1994), is obtained from the high-order expression of internal energy, using again the Hermitian shape functions for interpolating flexural displacements. The integration of the first two terms of Eq. (2.72) results in the same stiffness coefficients of the SR2O matrix, which account only for axial force effects. The integration of the additional terms results in the stiffness coefficients that correspond to the interaction between axial force and bending behavior, in order to better capture the geometric nonlinearity.

Considering a constant shear force over the length of the element, the internal bending moment and shear force can be expressed as a function of the bending moment values at the ends of the element as:

$$Q = -\frac{M_1 + M_2}{L} \tag{2.74}$$

$$M = -M_1 + \frac{M_1 + M_2}{L} x \tag{2.75}$$

Integrating the terms of Eq. (2.72), with the Hermitian shape functions, and expressing the shear force and bending moment as in Eq. (2.74) and Eq. (2.75), the stiffness coefficients are combined into a single geometric stiffness matrix as follows:

$$[k_G]_{LR2T} = \begin{bmatrix} [k_G]_1 & [k_G]_2 \\ [k_G]_2^T & [k_G]_3 \end{bmatrix} \tag{2.76}$$

$$[k_G]_1 = \begin{bmatrix} \frac{P}{L} & 0 & -\frac{M_1}{L} \\ 0 & \frac{6P}{5L}+\frac{12PI}{AL^3} & \frac{P}{10}+\frac{6PI}{AL^2} \\ -\frac{M_1}{L} & \frac{P}{10}+\frac{6PI}{AL^2} & \frac{2PL}{15}+\frac{4PI}{AL} \end{bmatrix} \tag{2.77}$$

$$[k_G]_2 = \begin{bmatrix} -\frac{P}{L} & 0 & -\frac{M_2}{L} \\ 0 & -\frac{6P}{5L}-\frac{12PI}{AL^3} & \frac{P}{10}+\frac{6PI}{AL^2} \\ \frac{M_1}{L} & -\frac{P}{10}-\frac{6PI}{AL^2} & -\frac{PL}{30}+\frac{2PI}{AL} \end{bmatrix} \tag{2.78}$$

$$[k_G]_3 = \begin{bmatrix} \frac{P}{L} & 0 & \frac{M_2}{L} \\ 0 & \frac{6P}{5L}+\frac{12PI}{AL^3} & -\frac{P}{10}-\frac{6PI}{AL^2} \\ \frac{M_2}{L} & -\frac{P}{10}-\frac{6PI}{AL^2} & \frac{2PL}{15}+\frac{4PI}{AL} \end{bmatrix} \tag{2.79}$$

Finally, to obtain the LR4O geometric stiffness matrix, the complete shape functions for flexural displacements, presented by Rodrigues (2019), must be applied to the high-order expression of internal energy. This matrix can be found in the referred work.

2.4. Corotational Formulation

As a frame structure is loaded and moves from its original configuration, each beam element is subjected to three types of motion: it translates, rotates, and deforms. The rotation and translation are rigid-body motions, which do not generate internal forces and may be removed from the total motion of the beam. Thus, the idea of the Corotational formulation is to separate rigid-body modes from local deformations. The initial configuration is used as reference to measure rigid-body motions in the global axis system, which remains stationary. A corotated configuration that continuously translates and rotates with the element is used as the reference to measure deformations in a local axis system, called natural system. With respect to the corotated configuration, the rigid-body motions are null and only the relative displacements that cause deformations to an element are present.

When using this formulation, the relative displacements, measured in the natural system, can even be considered small. Therefore, the element can be formulated as linear (infinitesimal strains) in the natural system, with the geometric nonlinearity being introduced in the transformation between the reference systems. In this case, the formulation can handle arbitrarily large rigid motions, with small deformations along the element. Such characteristic turned the Corotational formulation popular for analyses with finite motions but small strains, especially for structural elements such as beams, plates and shells.

Based on this approach, the tangent stiffness matrix of a two-dimensional Euler-Bernoulli beam element, subjected to large displacements/rotations but small deformations, is developed in this section. The presented theory follows the works of Crisfield (1991), Souza (2000), and Baião (2017).

2.4.1. Global and Natural Systems

As previously mentioned, the corotated configuration uses a local system to which the deformations of the element are measured. This local system moves with the element and is called natural system. Figure 2.9 shows the initial, deformed, and corotated configurations of a beam element, as well as the displacement components in global and natural systems. The global nodal coordinates of the initial configuration are defined as (x_1, y_1) for the first node, and (x_2, y_2) for the second node.

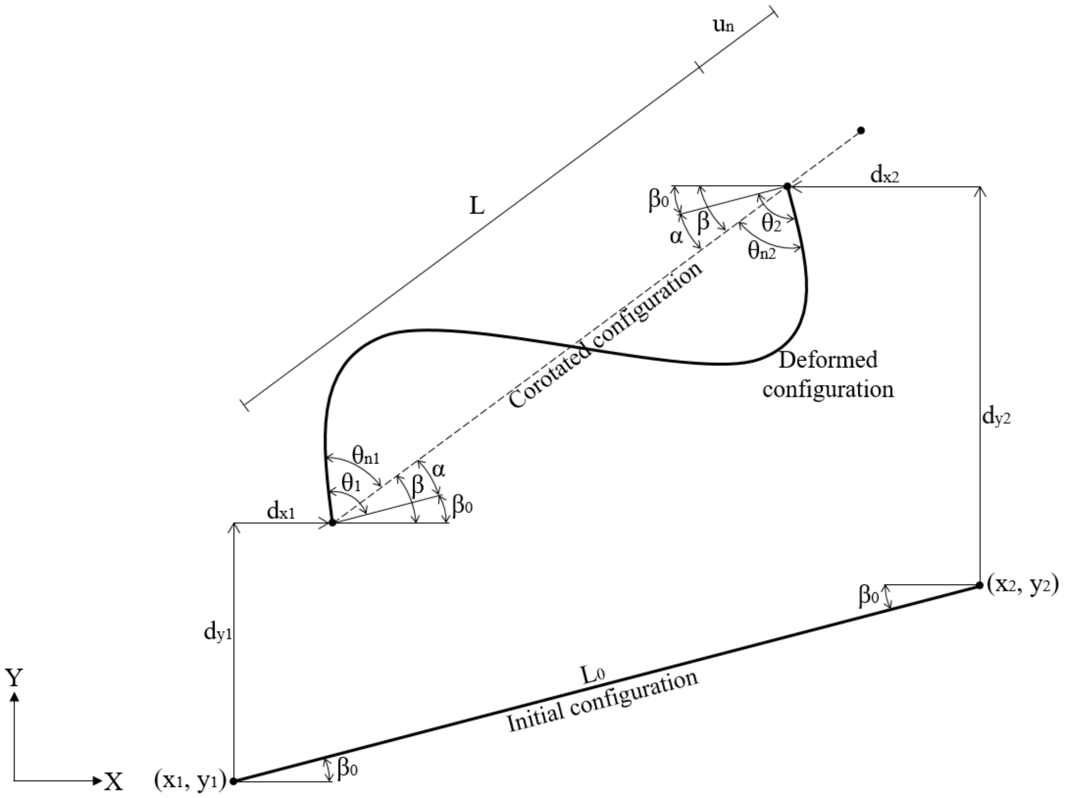


Figure 2.9 – Displacements in global and natural systems

The total displacements of the element degrees-of-freedom, with respect to the initial configuration, are measured in the global axis system *XY*. These displacements (translations d_{x1}, d_{y1} , d_{x2}, d_{y2}, and rotations θ_1, θ_2) are grouped in vector $\{d_g\}$ of Eq. (2.80).

The natural system has three components of displacements that effectively cause strains and stresses in the element. They are an elongation, u_n, with respect to the initial configuration, and two relative rotations to the corotated configuration, θ_{n1} and θ_{n2}. The displacement components of the natural system are grouped in vector $\{d_n\}$ of Eq. (2.81).

$$\left\{d_g\right\} = \left\{d_{x1} \quad d_{y1} \quad \theta_1 \quad d_{x2} \quad d_{y2} \quad \theta_2\right\}^T \tag{2.80}$$

$$\left\{d_n\right\} = \left\{u_n \quad \theta_{n1} \quad \theta_{n2}\right\}^T \tag{2.81}$$

The element elongation, responsible for axial deformation, is given by the difference between the lengths of the element in the corotated and initial configurations, according to Eq. (2.82). The element length can be obtained directly from the nodal coordinates in each configuration, according to Eq. (2.83) and Eq. (2.84).

$$u_n = L - L_0 \tag{2.82}$$

$$L_0 = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{2.83}$$

$$L = \sqrt{\left((x_2 + d_{x2}) - (x_1 + d_{x1})\right)^2 + \left((y_2 + d_{y2}) - (y_1 + d_{y1})\right)^2} \tag{2.84}$$

The nodal rotations relative to the corotated configuration, responsible for flexural deformations and the arising of bending moments and shear forces, are obtained from the difference between the total rotation of nodes and the rigid-body rotation of the element, α, according to Eq. (2.85) and Eq. (2.86).

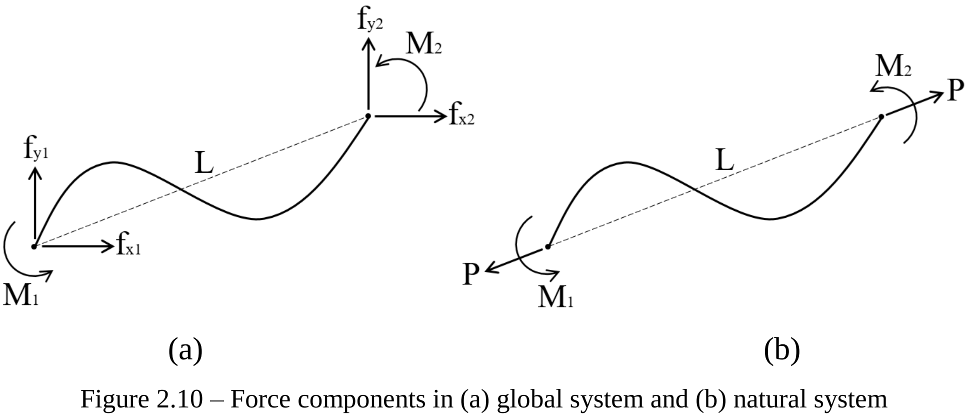
The rigid-body rotation is the angle between corotated and initial configurations. It can be obtained from the initial length of the element and the total displacements, according to Eq. (2.87). However, as explained in Section 4.2.3, in a computer implementation, the use of the arctangent function has some limitations. Hence, it is more appropriate to calculate the difference between the angles of the reference configurations (corotated, β, and initial, β_0), in which the corotated configuration angle is incremented in small values. Souza (2000) also proposed an efficient way of calculating the rigid-body rotation with no limitation to its value.

$$\theta_{n1} = \theta_1 - \alpha \tag{2.85}$$

$$\theta_{n2} = \theta_2 - \alpha \tag{2.86}$$

$$\alpha = \beta - \beta_0 = \tan^{-1}\left(\frac{d_{y2} - d_{y1}}{L_0 + d_{x2} - d_{x1}}\right) \tag{2.87}$$

The nodal forces corresponding to the displacement components of the two reference systems are shown in Fig. 2.10.



(a) (b)

Figure 2.10 – Force components in (a) global system and (b) natural system

The nodal forces in global system (forces f_{x1}, f_{y1}, f_{x2}, f_{y2}, and bending moments M_1, M_2), acting in the same directions of the global displacements, are grouped in vector $\{f_g\}$ of Eq. (2.88). In the natural system, three force components are generated from the natural displacements, an axial force P, and two bending moments, M_1 and M_2. These components are grouped in vector $\{f_n\}$ of Eq. (2.89).

$$\left\{ f_g \right\} = \left\{ f_{x1} \quad f_{y1} \quad M_1 \quad f_{x2} \quad f_{y2} \quad M_2 \right\}^T \tag{2.88}$$

$$\left\{ f_n \right\} = \left\{ P \quad M_1 \quad M_2 \right\}^T \tag{2.89}$$

Using nodal displacements and forces in both reference systems (global and natural), the tangent stiffness matrix of the beam element will be derived. The tangent matrix relates the displacements and forces of the element degrees-of-freedom in global system. In fact, due to its nonlinear character, the tangent matrix relates displacement increments to force increments, hence the term “tangent”. In other words, it involves the variation of these quantities.

The relation between the variations of the global displacements, $\{\delta d_g\}$, and global forces, $\{\delta f_g\}$, is obtained from the relation between the variations of other global and natural quantities, indicated in Fig. 2.11:

- (1): natural displacements, $\{\delta d_n\}$, and natural forces, $\{\delta f_n\}$.
- (2): natural displacements, $\{\delta d_n\}$, and global displacements, $\{\delta d_g\}$.
- (3): natural forces, $\{\delta f_n\}$, and global forces, $\{\delta f_g\}$.

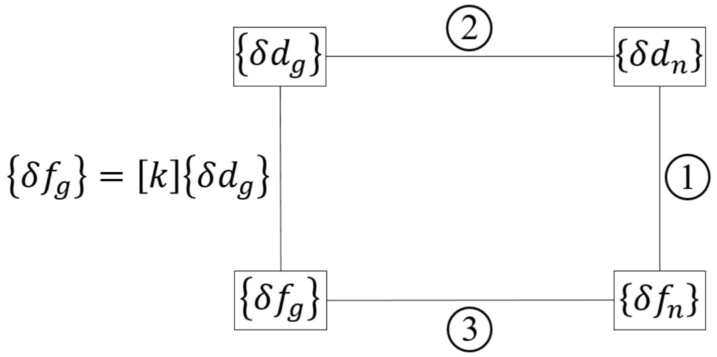


Figure 2.11 – Relations between variations of global and natural quantities

The next sections are dedicated to the development of these relations for obtaining the tangent stiffness matrix of a two-dimensional Euler-Bernoulli beam element in the global system (indicated by $[k]$ in Fig. 2.11).

2.4.2. Displacements and Forces in Natural System

Assuming linear-elastic material behavior, Hooke`s law is used to relate stresses and strains. The displacements and rotations measured with respect to the corotated configuration can be considered small, so engineering strain is used. Furthermore, the cross-section area is considered constant (typical assumption for small strain problems). Therefore, the axial force is related to the element elongation as:

$$P = \frac{EA}{L_0} u_n \tag{2.90}$$

Similarly, using standard structural analysis results it is possible to show that the end moments of the beam are related to the nodal rotations as follows:

$$M_1 = \frac{4EI}{L_0}\theta_{n1} + \frac{2EI}{L_0}\theta_{n2} \tag{2.91}$$

$$M_2 = \frac{2EI}{L_0}\theta_{n1} + \frac{4EI}{L_0}\theta_{n2} \tag{2.92}$$

Equations (2.90), (2.91), and (2.92) can be written in matrix form as in Eq. (2.93), where [C_n] is the matrix of linear-elastic stiffness coefficients that relates displacements and forces in natural system:

$$\begin{Bmatrix} P \\ M_1 \\ M_2 \end{Bmatrix} = \begin{bmatrix} EA/L_0 & 0 & 0 \\ 0 & 4EI/L_0 & 2EI/L_0 \\ 0 & 2EI/L_0 & 4EI/L_0 \end{bmatrix} \begin{Bmatrix} u_n \\ \theta_{n1} \\ \theta_{n2} \end{Bmatrix} \tag{2.93}$$

$$\{f_n\} = [C_n]\{d_n\} \tag{2.94}$$

Finally, the variation of the natural forces are related to the variation of the natural displacements as:

$$\begin{Bmatrix} \delta P \\ \delta M_1 \\ \delta M_2 \end{Bmatrix} = \begin{bmatrix} EA/L_0 & 0 & 0 \\ 0 & 4EI/L_0 & 2EI/L_0 \\ 0 & 2EI/L_0 & 4EI/L_0 \end{bmatrix} \begin{Bmatrix} \delta u_n \\ \delta\theta_{n1} \\ \delta\theta_{n2} \end{Bmatrix} \tag{2.95}$$

$$\{\delta f_n\} = [C_n]\{\delta d_n\} \tag{2.96}$$

2.4.3. Displacements in Natural and Global Systems

To find a relation between nodal displacements in natural and global systems, consider a small movement (variation δ*d*) from the corotated configuration, as shown in Fig. 2.12. In that figure, e_1 is the unit vector from first to second node; e_2 is the unit vector orthogonal to e_1 (such that the cross product is an out-of-plane vector); δu_n and δα are, respectively, the variations of element elongation and rigid-body rotation.

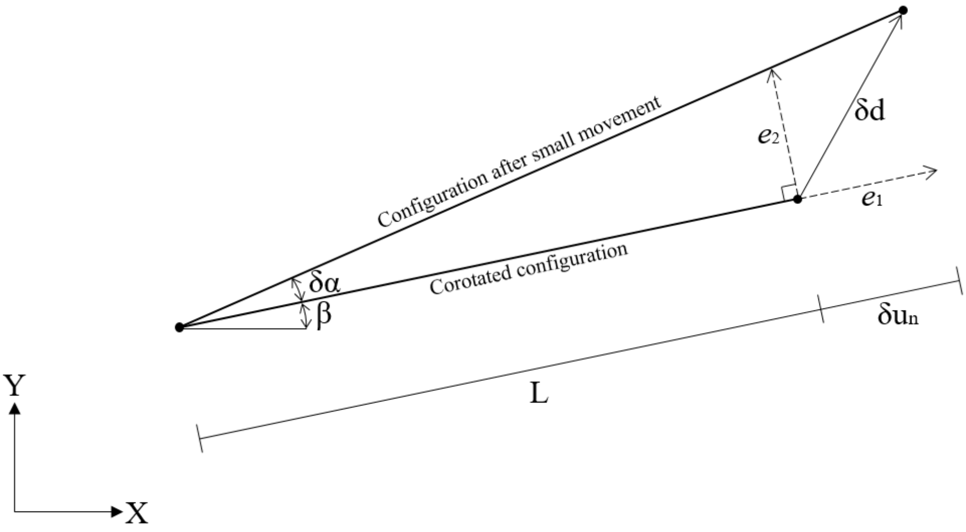


Figure 2.12 – Variation from the corotated configuration

The components of the unit vectors and the movement variation, in the global axis system, are:

$$\{e_1\} = \begin{Bmatrix} \cos\beta \\ \sin\beta \end{Bmatrix} \qquad \{e_2\} = \begin{Bmatrix} -\sin\beta \\ \cos\beta \end{Bmatrix} \qquad \{\delta d\} = \begin{Bmatrix} \delta d_{x2} - \delta d_{x1} \\ \delta d_{y2} - \delta d_{y1} \end{Bmatrix} \tag{2.97}$$

The small movement in the axial direction (variation of element elongation) can be determined by the scalar product between the unit axial vector and the movement variation, which gives the component of δ*d* projected along the direction of e_1:

$$\delta u_n = \{e_1\}^T \{\delta d\} \tag{2.98}$$

By multiplying these vectors and organizing the terms, the relation between the variations of element elongation and global nodal displacements is obtained as follows:

$$\delta u_n = \cos\beta \cdot \delta d_{x2} - \cos\beta \cdot \delta d_{x1} + \sin\beta \cdot \delta d_{y2} - \sin\beta \cdot \delta d_{y1} \tag{2.99}$$

$$\delta u_n = \left\{ -\cos\beta \quad -\sin\beta \quad 0 \quad \cos\beta \quad \sin\beta \quad 0 \right\} \left\{ \begin{matrix} \delta d_{x1} \\ \delta d_{y1} \\ \delta\theta_1 \\ \delta d_{x2} \\ \delta d_{y2} \\ \delta\theta_2 \end{matrix} \right\} \tag{2.100}$$

$$\delta u_n = \left\{ r \right\}^T \left\{ \delta d_g \right\} \tag{2.101}$$

The variation of rigid-body rotation gives, for an infinitesimal angle change, an arc-length change of:

$$L \cdot \delta\alpha = \left\{ e_2 \right\}^T \left\{ \delta d \right\} \tag{2.102}$$

By developing this equation, the relation between the variations of rigid-body rotation and global nodal displacements is obtained as follows:

$$\delta\alpha = \frac{1}{L}\left(-\sin\beta \cdot \delta d_{x2} + \sin\beta \cdot \delta d_{x1} + \cos\beta \cdot \delta d_{y2} - \cos\beta \cdot \delta d_{y1} \right) \tag{2.103}$$

$$\delta\alpha = \frac{1}{L}\left\{ \sin\beta \quad -\cos\beta \quad 0 \quad -\sin\beta \quad \cos\beta \quad 0 \right\} \left\{ \begin{matrix} \delta d_{x1} \\ \delta d_{y1} \\ \delta\theta_1 \\ \delta d_{x2} \\ \delta d_{y2} \\ \delta\theta_2 \end{matrix} \right\} \tag{2.104}$$

$$\delta\alpha = \frac{1}{L}\left\{ z \right\}^T \left\{ \delta d_g \right\} \tag{2.105}$$

The variation of the relative nodal rotations, in which we are interested in, is given by the difference between the variations of total nodal rotations and rigid-body rotation:

$$\delta\theta_{n1} = \delta\theta_1 - \delta\alpha \tag{2.106}$$

$$\delta\theta_{n2} = \delta\theta_2 - \delta\alpha \tag{2.107}$$

Expressing the nodal rotations in terms of the vector of global displacements and using Eq. (2.105), the previous equations can be rewritten as:

$$\delta\theta_{n1} = \{0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0\}\{\delta d_g\} - \frac{1}{L}\{z\}^T\{\delta d_g\} \qquad (2.108)$$

$$\delta\theta_{n2} = \{0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1\}\{\delta d_g\} - \frac{1}{L}\{z\}^T\{\delta d_g\} \qquad (2.109)$$

The relation between variations of relative rotations and global nodal displacements is then:

$$\begin{Bmatrix} \delta\theta_{n1} \\ \delta\theta_{n2} \end{Bmatrix} = \left[\begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} - \frac{1}{L} \begin{bmatrix} \{z\}^T \\ \{z\}^T \end{bmatrix} \right] \{\delta d_g\} \qquad (2.110)$$

$$\begin{Bmatrix} \delta\theta_{n1} \\ \delta\theta_{n2} \end{Bmatrix} = [A]^T \{\delta d_g\} \qquad (2.111)$$

Combining Eq. (2.101) and Eq. (2.111) into a single matrix form, we arrive at the relation between the variation of natural and global displacements:

$$\begin{Bmatrix} \delta u_n \\ \delta\theta_{n1} \\ \delta\theta_{n2} \end{Bmatrix} = \begin{bmatrix} \{r\}^T \\ [A]^T \end{bmatrix} \{\delta d_g\} \qquad (2.112)$$

$$\{\delta d_n\} = [T]^T \{\delta d_g\} \qquad (2.113)$$

The matrix [*T*], whose transpose relates the variation of natural and global displacements, is identified in Eq. (2.114). This transformation matrix is the same that relates natural and global forces, and it consists of an expansion and a rotation from the natural system to the global system, as seen in the next section.

$$[T] = \begin{bmatrix} -\cos\beta & -\sin\beta/L & -\sin\beta/L \\ -\sin\beta & \cos\beta/L & \cos\beta/L \\ 0 & 1 & 0 \\ \cos\beta & \sin\beta/L & \sin\beta/L \\ \sin\beta & -\cos\beta/L & -\cos\beta/L \\ 0 & 0 & 1 \end{bmatrix} \qquad (2.114)$$

2.4.4. Forces in Natural and Global Systems

To relate force components in the natural system to the components of the global system, first a relation between forces in natural and local systems will be established. Figure 2.13 shows the force components in natural system, in red, and the nodal forces in local axis system, in green.

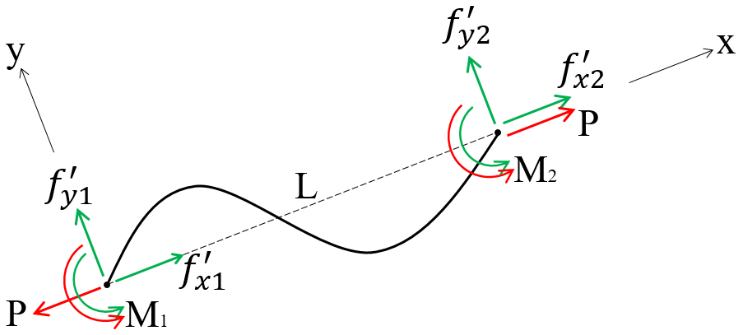


Figure 2.13 – Force components in natural and local systems

It is clear that the local axial force components, f'_{x1} and f'_{x2}, are related to the natural axial force *P* as:

$$f'_{x1} = -P \qquad f'_{x2} = P \tag{2.115}$$

Considering a constant shear force along the element and verifying the equilibrium of moments at each node, the local transverse force components, f'_{y1} and f'_{y2}, can be expressed in terms of the end moments as:

$$f'_{y1} = \frac{M_1 + M_2}{L} \qquad f'_{y2} = -\frac{M_1 + M_2}{L} \tag{2.116}$$

Using Eq. (2.115) and Eq. (2.116) and noticing that the bending moments are common to natural and local systems, the relation between the force components of both systems is given in Eq. (2.117). The matrix [*B*] is identified as an expansion matrix from natural to local system.

$$\begin{Bmatrix} f'_{x1} \\ f'_{y1} \\ M_1 \\ f'_{x2} \\ f'_{y2} \\ M_2 \end{Bmatrix} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1/L & 1/L \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & -1/L & -1/L \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} P \\ M_1 \\ M_2 \end{Bmatrix} \tag{2.117}$$

$$\{f_l\} = [B]\{f_n\} \tag{2.118}$$

Now, a relation between nodal forces in local and global systems will be established. Figure 2.14 shows the nodal forces in local axis system, in green, and in global axis system, in blue.

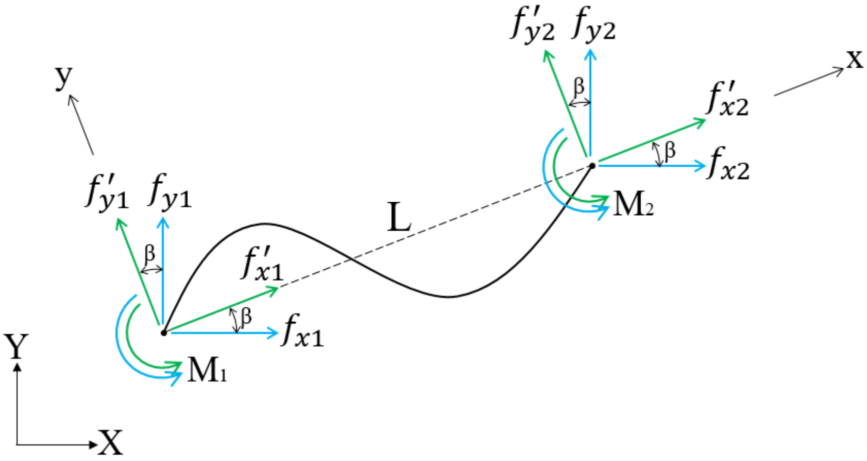


Figure 2.14 – Force components in local and global systems

The equilibrium of forces at each node provides the following relations:

$$f_{x1} = f'_{x1}\cos\beta - f'_{y1}\sin\beta \tag{2.119}$$

$$f_{y1} = f'_{x1}\sin\beta + f'_{y1}\cos\beta \tag{2.120}$$

$$f_{x2} = f'_{x2}\cos\beta - f'_{y2}\sin\beta \tag{2.121}$$

$$f_{y2} = f'_{x2}\sin\beta + f'_{y2}\cos\beta \tag{2.122}$$

Since bending moment components are the same for local and global systems, the relation between local and global nodal forces is given in Eq. (2.123). The matrix [*R*] is identified as a rotation matrix from local to global system.

$$\begin{Bmatrix} f_{x1} \\ f_{y1} \\ M_1 \\ f_{x2} \\ f_{y2} \\ M_2 \end{Bmatrix} = \begin{bmatrix} \cos\beta & -\sin\beta & 0 & 0 & 0 & 0 \\ \sin\beta & \cos\beta & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \cos\beta & -\sin\beta & 0 \\ 0 & 0 & 0 & \sin\beta & \cos\beta & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} f'_{x1} \\ f'_{y1} \\ M_1 \\ f'_{x2} \\ f'_{y2} \\ M_2 \end{Bmatrix} \tag{2.123}$$

$$\{f_g\} = [R]\{f_l\} \tag{2.124}$$

By substituting Eq. (2.118) into Eq. (2.124), the relation between force components in natural and global systems is obtained in Eq. (2.125). The product of the rotation matrix [*R*] by the expansion matrix [*B*] results in the transformation matrix [*T*], previously given in Eq. (2.114).

$$\left\{ f_g \right\} = [R][B]\{ f_n \} \tag{2.125}$$

$$\left\{ f_g \right\} = [T]\{ f_n \} \tag{2.126}$$

The [*T*] matrix is the same whose transpose relates the variation of natural and global displacements. This corresponds to the Principle of Contragradiency of structural analysis. Therefore, the relation of Eq. (2.126) could also be derived from the PVD, considering that the work performed by forces going through virtual displacements in the global and natural systems is equal. This is seen in Eq. (2.127), where virtual work must hold for an arbitrary virtual displacement, which can be canceled from the first and last terms of this equation, resulting in Eq. (2.126).

$$\left\{\delta d_g\right\}^T \left\{ f_g \right\} = \left\{\delta d_n\right\}^T \{ f_n \} = \left([T]^T \left\{\delta d_g\right\}\right)^T \{ f_n \} = \left\{\delta d_g\right\}^T [T]\{ f_n \} \tag{2.127}$$

Once the relation between natural and global forces is known, the variation of these quantities, in which we are interested, is then obtained by the product rule:

$$\left\{\delta f_g\right\} = [T]\{\delta f_n\} + [\delta T]\{ f_n \} \tag{2.128}$$

To use this expression, it is necessary to determine the variation of the transformation matrix, [δ*T*]. The variation of each column of the transformation matrix will be obtained separately. For the first column, {*T*1}, the variation is developed as follows:

$$\left\{\delta T_1\right\} = \delta\left\{-\cos\beta \quad -\sin\beta \quad 0 \quad \cos\beta \quad \sin\beta \quad 0\right\}^T \tag{2.129}$$

$$\left\{\delta T_1\right\} = \left\{\sin\beta \quad -\cos\beta \quad 0 \quad -\sin\beta \quad \cos\beta \quad 0\right\}^T \delta\beta \tag{2.130}$$

$$\left\{\delta T_1\right\} = \{z\}\,\delta\beta \tag{2.131}$$

Observing that δβ = δα and using Eq. (2.105), the variation of the first column of the transformation matrix is:

$$\left\{\delta T_1\right\} = \frac{1}{L}\{z\}\{z\}^T \left\{\delta d_g\right\} \tag{2.132}$$

The second and third columns, $\{T_2\}$ and $\{T_3\}$, can be written as:

$$\{T_2\} = -\frac{1}{L}\{z\} + \{0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0\}^T \tag{2.133}$$

$$\{T_3\} = -\frac{1}{L}\{z\} + \{0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1\}^T \tag{2.134}$$

The variation of the constant vectors in these equations is zero, so the variation of the second and third columns is equal. Using the product rule:

$$\{\delta T_2\} = \{\delta T_3\} = \delta\left(-\frac{1}{L}\{z\}\right) = \delta\left(-\frac{1}{L}\right)\{z\} + \left(-\frac{1}{L}\right)\{\delta z\} \tag{2.135}$$

In the previous expression, it is necessary to find δ(-1/L) and $\{\delta z\}$. The former is developed using the chain rule:

$$\delta\left(-\frac{1}{L}\right) = -\delta L^{-1} = -\left(-L^{-2}\delta L\right) = \frac{\delta L}{L^2} \tag{2.136}$$

Since $\delta L = \delta L_0 + \delta u_n$ and $\delta L_0 = 0$, and using Eq. (2.101) to express δu_n, the following result is obtained:

$$\delta\left(-\frac{1}{L}\right) = \frac{\delta u_n}{L^2} = \frac{1}{L^2}\{r\}^T\{\delta d_g\} \tag{2.137}$$

To find $\{\delta z\}$, it should be noted again that δβ = δα, and Eq. (2.105) is used to express δα:

$$\{\delta z\} = \delta\{\sin\beta \quad -\cos\beta \quad 0 \quad -\sin\beta \quad \cos\beta \quad 0\}^T \tag{2.138}$$

$$\{\delta z\} = \{\cos\beta \quad \sin\beta \quad 0 \quad -\cos\beta \quad -\sin\beta \quad 0\}^T \delta\beta \tag{2.139}$$

$$\{\delta z\} = -\{r\}\delta\alpha = -\frac{1}{L}\{r\}\{z\}^T\{\delta d_g\} \tag{2.140}$$

Replacing Eq. (2.137) and Eq. (2.140) into Eq. (2.135), the following expression is obtained for the variation of the second and third columns of the transformation matrix:

$$\{\delta T_2\} = \{\delta T_3\} = \frac{1}{L^2}\left(\{z\}\{r\}^T + \{r\}\{z\}^T\right)\{\delta d_g\} \tag{2.141}$$

Now that the variation of the transformation matrix is determined, the relation between the variation of natural and global forces, given in Eq. (2.128), is completely defined.

2.4.5. Tangent Stiffness Matrix in Global System

All the necessary relations between global and natural quantities have been established in the previous sections in order to develop a variationally consistent tangent stiffness matrix in the global system. These relations are illustrated in Fig. 2.15, where the bold equation indicates the unknown relation that is being sought. This unknown relation is provided by the tangent matrix, which can be defined as the differentiation of internal global forces with respect to global displacements

$$\begin{tikzcd}
\boxed{\{\delta d_g\}} \arrow[r, dash, "\{\delta d_n\} = [T]^T\{\delta d_g\}"] \arrow[d, dash, "\{\boldsymbol{\delta f_g}\} = [\boldsymbol{k}]\{\boldsymbol{\delta d_g}\}"'] & \boxed{\{\delta d_n\}} \arrow[d, dash, "\{\delta f_n\} = [C_n]\{\delta d_n\}"] \\
\boxed{\{\delta f_g\}} \arrow[r, dash, "\{\delta f_g\} = [T]\{\delta f_n\} + [\delta T]\{f_n\}"'] & \boxed{\{\delta f_n\}}
\end{tikzcd}$$

Figure 2.15 – Established relations between the variations of global and natural quantities

Equation (2.128), at the bottom of Fig. 2.15, is rewritten by separating the columns of [δ*T*]:

$$\left\{\delta f_g\right\} = \left[T\right]\left\{\delta f_n\right\} + P\left\{\delta T_1\right\} + M_1\left\{\delta T_2\right\} + M_2\left\{\delta T_3\right\} \tag{2.142}$$

Equation (2.96), on the right side of Fig. 2.15, is used to replace the natural forces in the previous expression. In addition, Eq. (2.132) and Eq. (2.141) are used to express the variation of the transformation matrix columns in terms of the variation of global displacements. The result is:

$$\begin{aligned}\left\{\delta f_g\right\} = &\left[T\right]\left[C_n\right]\left\{\delta d_n\right\} + \\ &\frac{P}{L}\left\{z\right\}\left\{z\right\}^T\left\{\delta d_g\right\} + \frac{M_1 + M_2}{L^2}\left(\left\{z\right\}\left\{r\right\}^T + \left\{r\right\}\left\{z\right\}^T\right)\left\{\delta d_g\right\}\end{aligned} \tag{2.143}$$

Equation (2.113), at the top of Fig. 2.15, is used to replace the variation of natural displacements in the first term on the right hand side of the previous expression:

$$\begin{aligned}\left\{\delta f_g\right\} = &\left[T\right]\left[C_n\right]\left[T\right]^T\left\{\delta d_g\right\} + \\ &\frac{P}{L}\left\{z\right\}\left\{z\right\}^T\left\{\delta d_g\right\} + \frac{M_1 + M_2}{L^2}\left(\left\{z\right\}\left\{r\right\}^T + \left\{r\right\}\left\{z\right\}^T\right)\left\{\delta d_g\right\}\end{aligned} \tag{2.144}$$

Observing that $\{\delta d_g\}$ is a common factor, the relation between variations of global forces and global displacements is finally obtained:

$$\left\{\delta f_g\right\} = \left[[T][C_n][T]^T + \frac{P}{L}\{z\}\{z\}^T + \frac{M_1+M_2}{L^2}\left(\{z\}\{r\}^T + \{r\}\{z\}^T\right) \right]\left\{\delta d_g\right\} \quad (2.145)$$

where the tangent stiffness matrix is identified as:

$$[k] = [T][C_n][T]^T + \frac{P}{L}\{z\}\{z\}^T + \frac{M_1+M_2}{L^2}\left(\{z\}\{r\}^T + \{r\}\{z\}^T\right) \qquad (2.146)$$

This expression provides the tangent matrix of a beam element already in the global axis system. It does not need to be rotated before it is inserted into the global system matrix. The result of multiplying all terms of the tangent matrix expression is:

$$[k] = \begin{bmatrix} k_1 & -k_3 & -k_5 & -k_1 & k_3 & -k_5 \\ & k_2 & k_4 & k_3 & -k_2 & k_4 \\ & & k_7 & k_5 & -k_4 & k_6 \\ & & & k_1 & -k_3 & k_5 \\ & \text{symm} & & & k_2 & -k_4 \\ & & & & & k_7 \end{bmatrix}$$

$$\begin{aligned}
k_1 &= \left(EAL^2 \left(\cos\beta\right)^2 + NLL_0 \left(\sin\beta\right)^2 - 2\left(M_1 + M_2\right) L_0 \cos\beta \sin\beta + 12EI \left(\sin\beta\right)^2 \right) \Big/ \left(L^2 L_0\right) \\
k_2 &= \left(EAL^2 \left(\sin\beta\right)^2 + NLL_0 \left(\cos\beta\right)^2 + 2\left(M_1 + M_2\right) L_0 \cos\beta \sin\beta + 12EI \left(\cos\beta\right)^2 \right) \Big/ \left(L^2 L_0\right) \\
k_3 &= \left(NLL_0 \cos\beta \sin\beta - \left(M_1 + M_2\right)\left(\left(\cos\beta\right)^2 - \left(\sin\beta\right)^2\right) L_0 + \cos\beta \sin\beta \left(12EI - EAL^2\right)\right) \Big/ \left(L^2 L_0\right) \\
k_4 &= \left(6EI \cos\beta\right) \Big/ \left(LL_0\right) \\
k_5 &= \left(6EI \sin\beta\right) \Big/ \left(LL_0\right) \\
k_6 &= \left(2EI\right) \Big/ L_0 \\
k_7 &= \left(4EI\right) \Big/ L_0
\end{aligned} \qquad (2.147)$$

It is interesting to notice that, in Eq. (2.146), the first term, which depends on the matrix of linear-elastic stiffness coefficients [C_n], corresponds to the elastic stiffness matrix. The remaining terms, which depend on element forces, originate the geometric stiffness matrix. Considering a beam element aligned with the global axes (β = 0), i.e., accounting only for the expansion from natural to local system, the result for the elastic and geometric stiffness matrices are respectively:

$$[k_E]=\begin{bmatrix} \frac{EA}{L_0} & 0 & 0 & -\frac{EA}{L_0} & 0 & 0 \\ & \frac{12EI}{L^2L_0} & \frac{6EI}{LL_0} & 0 & -\frac{12EI}{L^2L_0} & \frac{6EI}{LL_0} \\ & & \frac{4EI}{L_0} & 0 & -\frac{6EI}{LL_0} & \frac{2EI}{L_0} \\ & & & \frac{EA}{L_0} & 0 & 0 \\ & \text{symm} & & & \frac{12EI}{L^2L_0} & -\frac{6EI}{LL_0} \\ & & & & & \frac{4EI}{L_0} \end{bmatrix} \tag{2.148}$$

$$[k_G]=\begin{bmatrix} 0 & \frac{(M_1+M_2)}{L^2} & 0 & 0 & -\frac{(M_1+M_2)}{L^2} & 0 \\ & \frac{P}{L} & 0 & -\frac{(M_1+M_2)}{L^2} & -\frac{P}{L} & 0 \\ & & 0 & 0 & 0 & 0 \\ & & & 0 & \frac{(M_1+M_2)}{L^2} & 0 \\ & \text{symm} & & & \frac{P}{L} & 0 \\ & & & & & 0 \end{bmatrix} \tag{2.149}$$

3. Solution Methods

3.1. Introduction

In the previous chapter, the local stiffness matrices of plane beam elements, subjected to large displacements and large rotations, were deduced. These matrices include elastic and geometric stiffness coefficients, to compose the tangent matrix of the element, which relates the displacements and forces acting on nodal degrees-of-freedom. The global tangent matrix of the model is assembled by a usual direct stiffness process, adding the stiffness coefficients of each local matrix to the corresponding degree-of-freedom of the global matrix, as well as in a linear analysis. However, in this case, a system of nonlinear equilibrium equations is obtained to account for the deformed geometry of the structure.

The system nonlinearity lies in the fact that the evaluation of the tangent stiffness matrix, at any equilibrium configuration, depends on the deformed geometry of the structure and the internal forces of the elements. Such properties are obtained from the nodal displacements, which are the unknowns of the problem. Therefore, it is not possible to solve the system of equilibrium equations analytically, and it needs to be treated numerically.

In addition to a consistent and well-developed nonlinear formulation, a well-implemented solver for the system of equilibrium equations is equally important for obtaining the correct answers to the structural problem. At first, iterative methods for solving systems of nonlinear equations, such as the Newton-Raphson method, could be applied to find the displacements corresponding to the externally applied loads. This is because, considering only the geometric nonlinearity, the equilibrium configuration corresponding to a given load level does not depend on the history of the structure response.

However, directly calculating total displacements from the total applied loads is not a good practice. Firstly, there is no prior guarantee that the structure will withstand the applied loads, and may present critical points in the solution associated with instability phenomena, which would not be detected in a direct analysis. Besides that, iterative algorithms for solving nonlinear systems need an initial solution (initial equilibrium configuration) close to the final answer to be able to converge correctly, without numerical instability problems. Therefore, even for an analysis without dependence on the history of equilibrium configurations, the numerical approach must be such that it is possible to follow the equilibrium path of the structure to the desired load.

The practical way to solve the nonlinear system of equations and follow the equilibrium path is by an incremental analysis. This methodology consists of applying the total load in a series of increments for which the corresponding displacement increments are obtained by direct or iterative solution of an incremental linear system. Thus, a step linearization is performed, approximating the nonlinear relation between applied loads and nodal displacements through one or several linear analyzes in each step.

This chapter presents two categories of incremental methods for solving the nonlinear equilibrium system. The first is the purely incremental, or single-step, methods in which a single stiffness matrix is used to represent the load-displacement relation in each analysis step. The second is the incremental-iterative methods, which perform various linear analyzes to iteratively solve the incremental system at each step, in order to converge to the equilibrium solution within a numerical tolerance. A quick review of concepts about equilibrium paths, critical points and stability is provided first. These concepts are important to the rest of the chapter.

3.2. Equilibrium Path, Critical Points, and Stability

The equilibrium path of a structural model is the graphical representation of the solutions of the equilibrium system, for various levels of the applied loads (or other external solicitations). Normally, the plotted curve gives the value of a representative displacement of the model, associated with one of its degrees-of-freedom, versus a load factor. The representative displacement of a model depends on the study to be performed. This displacement should be able to capture the phenomenon of interest and its choice involves the analyst's prior knowledge of the structural behavior. By studying the equilibrium path, many aspects of the nonlinear response of the structure can be evaluated.

In a linear analysis, the equilibrium path is a straight line with no load limit. On the other hand, nonlinear formulations provide paths that can be complex curves with critical points and multiple responses to a given load level, or even no real response, since nonlinear equations have multiple roots in the complex domain. The curve that passes through the initial configuration of the analysis (the origin of the diagram) is called the fundamental, or primary, path. This path extends to the so-called critical points, where it connects with secondary, or post-critic, paths. Most structures are designed to work on the fundamental path. However, the study of the complete post-critic solution is important to determine how the structure behaves in emergency situations, and to assess its stability conditions.

Analyzing only the equilibrium of a structure is not sufficient for a safe project because an equilibrium configuration can be stable or unstable. Instability phenomena generate sudden changes in geometry and in the distribution of internal forces that can lead to catastrophic failures. Therefore, stability criteria should often be used, in addition to strength criteria, especially for slender elements where instability may occur in the linear-elastic regime. The study of the stability of structures is closely linked with the characteristics presented by their equilibrium paths. It is possible to identify, for example, the sign of the effective stiffness (tangent of the curve) in different equilibrium configurations, which may indicate the stability condition. A negative stiffness is necessarily associated with an unstable equilibrium configuration, while a positive stiffness is necessary, but not sufficient, for a stable equilibrium. Details on stability criteria are beyond the scope of this work. References for this subject are the books of Bazant & Cedolin (2010), Timoshenko & Gere (2009), and Thompson & Hunt (1973).

The equilibrium configuration where the structure loses stability is characterized by some type of critical point. In analyzes with geometric nonlinearity, the types of critical point that may arise in the equilibrium path are load limit points and bifurcation points. Displacement limit points should also receive some attention, although they are not associated with stability conditions. Figure 3.1 shows some examples of limit point occurrences.

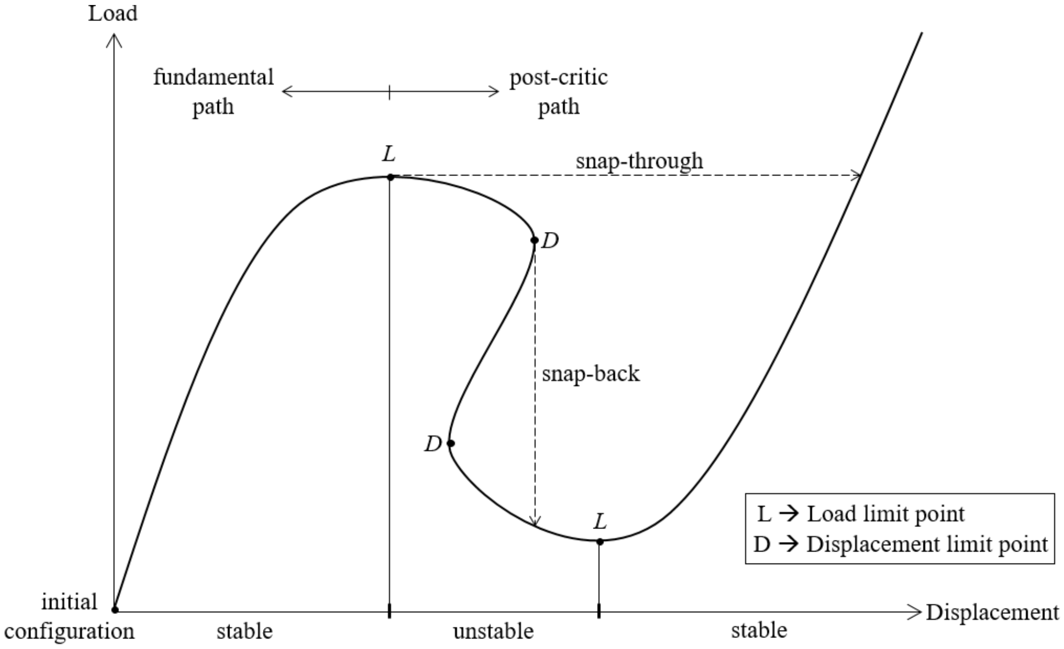


Figure 3.1 – Types of limit points

Load limit points are those at which a system's capacity for resistance to additional load is exhausted and continued deformation results in a decrease in load-resisting capacity (McGuire *et al.*, 2000). Graphically this means a local maximum or minimum of the curve, where the tangent at the point is horizontal (parallel to the axis of displacement). Numerically, these points are characterized by the singularity of the tangent stiffness matrix. Although a zero-stiffness limit point is rarely encountered in the incremental methods commonly employed, the approach to it indicates that the numerical accuracy of the solution may diminish in the neighborhood of a limit point because of ill-conditioning. Special solution techniques are required to detect and pass through a load limit point for continuing the analysis into the post-critic region.

The physical phenomenon associated with load limit points is called snap-through. This phenomenon occurs when the structure undergoes a load increase until, upon reaching the limit point, a dynamic jump occurs to seek another equilibrium position, in a process that releases the internal energy of deformation as kinetic energy. The response between the maximum and minimum load limit points has a negative stiffness and thus is an unstable region. This behavior is usually presented by shallow structures, such as arches and membranes, where concavity inversion occurs.

Displacement limit points, also known as turning points or snap-back points, are not considered as a critical point but have implications for the numerical solution of the path trace. Equilibrium stability after displacement limit points is not necessarily changed. These points do not have much physical meaning. They are graphically identified by vertical tangents of the curve (parallel to the load axis).

Critical points of bifurcation occur at the intersection between two or more equilibrium paths, belonging to different paths. The path followed by the structure is the one with the lowest energy required. The physical phenomenon associated with this type of critical point is called buckling, and commonly occurs in slender structures under compression loads. However, the occurrence of ideal bifurcation points is rare in real structural systems, as the inevitable imperfections of geometry, load application, and materials make the crossing of paths impossible to occur, and only one of them is followed. The detection of bifurcation points is not treated in this work, but it is well explored in Crisfield (1997) and Teh & Clarke (1999).

3.3. Incremental System of Equilibrium Equations

To trace the equilibrium path, the nonlinear system of equations must be solved several times, in an incremental fashion. The nonlinear solution is approximated by linear responses at each increment. The linear approximation generates a residue between the linearized solution and the real nonlinear solution. This residue may, or may not, be corrected by iterative techniques, depending on the solution method selected for the analysis. The following sections aim to present the methodology of incremental solution of the nonlinear system of equilibrium equations. The methods implemented in the developed tool are detailed.

Throughout this chapter, 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. The symbol Δ is used to indicate the increments of the variables in each step, while the symbol δ is used here as the indication of the increments in each iteration. Furthermore, vector and matrix variables are written in bold, and scalar product is represented by a dot. All vectors and matrices correspond to the global structure arrays, assembled with the local arrays of each element.

3.3.1. Incremental Solution

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 as in Eq. (3.1), where ***F*** is the vector of internal nodal forces, which is a function of the nodal displacements vector, ***U***, and ***P*** is the vector of external nodal forces applied to the structure, which is a function of a load factor, λ. Equation (3.2) shows how the vector of external nodal forces is expressed as the product of a load factor by a vector of reference loads, $\bar{\boldsymbol{P}}$. In this work, the reference loads are taken as the total external forces, proportionally applied to the nodal degrees-of-freedom, i.e., all components of the vector of external forces are increased or decreased by the same ratio. Therefore, the load factor is interpreted as a ratio of the externally applied forces, being referred to as load ratio.

$$\boldsymbol{F}\left(\boldsymbol{U}\right)=\boldsymbol{P}\left(\lambda\right) \tag{3.1}$$

$$\boldsymbol{P}=\lambda\bar{\boldsymbol{P}} \tag{3.2}$$

Both force vectors in Eq. (3.1) are developed using the Principle of Virtual Displacements, a kinematic description, and the FEM to discretize each element displacement and rotation fields.

The solution of the equilibrium system in Eq. (3.1) is obtained incrementally. For a sequence of external load increments, given by increments of the load ratio, $\Delta\lambda_i$, the corresponding increments of nodal displacements, Δ***Ui***, are calculated by linearizing the problem, where subscript *i* indicates the *i*-th analysis step. The total external forces and nodal displacements of the current (desired) configuration (step *i*) are then computed by adding the incremental updates to the previous (known) configuration (step *i*-1):

$$\boldsymbol{P}_i=\left(\lambda_{i-1}+\Delta\lambda_i\right)\bar{\boldsymbol{P}} \tag{3.3}$$

$$\boldsymbol{U}_i=\boldsymbol{U}_{i-1}+\Delta\boldsymbol{U}_i \tag{3.4}$$

Therefore, starting from a known equilibrium configuration, the next equilibrium state is sought by the balance of internal and external forces, which are incrementally increased:

$$\boldsymbol{F}\left(\boldsymbol{U}_i\right)=\boldsymbol{P}_i \tag{3.5}$$

$$\boldsymbol{F}\left(\boldsymbol{U}_{i-1}+\Delta\boldsymbol{U}_i\right)=\left(\lambda_{i-1}+\Delta\lambda_i\right)\bar{\boldsymbol{P}} \tag{3.6}$$

$$\boldsymbol{F}\left(\Delta\boldsymbol{U}_i\right)=\Delta\lambda_i\bar{\boldsymbol{P}}+\lambda_{i-1}\bar{\boldsymbol{P}}-\boldsymbol{F}\left(\boldsymbol{U}_{i-1}\right) \tag{3.7}$$

Notice that, as a result of linearization, it was possible to write $\boldsymbol{F}(\boldsymbol{U}_{i-1} + \Delta\boldsymbol{U}_i) = \boldsymbol{F}(\boldsymbol{U}_{i-1}) + \boldsymbol{F}(\Delta\boldsymbol{U}_i)$. Moreover, as the structure was in equilibrium in the previous configuration ($\boldsymbol{P}_{i-1} - \boldsymbol{F}(\boldsymbol{U}_{i-1}) = \mathbf{0}$), the incremental system of equilibrium equations is:

$$\boldsymbol{F}\left(\Delta\boldsymbol{U}_i\right)=\Delta\lambda_i\bar{\boldsymbol{P}} \tag{3.8}$$

The tangent stiffness matrix, $\boldsymbol{K}$, is defined in Eq. (3.9), as the derivatives of internal forces with respect to nodal displacements evaluated at the previous known configuration. Thus, the internal forces can be written as in Eq. (3.10), and the incremental system to solve for displacement increments is given in Eq. (3.11).

$$\boldsymbol{K}=\left.\frac{\partial\boldsymbol{F}}{\partial\boldsymbol{U}}\right|_{\boldsymbol{U}_{i-1}} \tag{3.9}$$

$$\boldsymbol{F}\left(\Delta\boldsymbol{U}_i\right)=\boldsymbol{K}_{i-1}\Delta\boldsymbol{U}_i \tag{3.10}$$

$$\boldsymbol{K}_{i-1}\Delta\boldsymbol{U}_i=\Delta\lambda_i\bar{\boldsymbol{P}} \tag{3.11}$$

The tangent matrix of a beam element was developed in Chapter 2 using the Principle of Virtual Displacements, a kinematic description (Updated Lagrangian or Corotational), and the Finite Element Method to discretize each element displacement field. The global system matrix is then assembled by a Direct Stiffness procedure. If the formulation is consistent, this matrix represents the derivatives of the internal forces with respect to nodal displacements (tangent of the solution). However, since the internal forces are a nonlinear function of the displacements, the solution of the linearized incremental system of Eq. (3.11) does not satisfy the equilibrium. A vector of residual forces arises as a result of the unbalance between external and internal forces. An iterative process is often employed within each increment to ensure the equilibrium by eliminating the residual forces.

It should be mentioned that the evaluation of the tangent stiffness matrix and the vector of internal forces, at any configuration, is done in distinct ways depending on the formulation of the problem. Details on this process, for the UL and CR formulations implemented in this work, are left for Chapter 4.

3.3.2. Incremental-Iterative System

Considering that the structure was in equilibrium at step *i*-1, it is desired to achieve equilibrium in the *i*-th step. Therefore, it is necessary to nullify the residual forces originated from the linearized increment. This is done by performing corrective iterations of the Newton-Raphson type within each incremental step until a convergence criterion is satisfied, and a new equilibrium state is established.

In each iteration, indicated by subscript *j* = 1,2,3…, increments of load ratio, $\delta\lambda_i^j$, and nodal displacements, $\delta \boldsymbol{U}_i^j$, are obtained. These iterative increments represent a correction of the load and displacement values in the respective step. Therefore, after the the *j*-th iteration, the total increments of the *i*-th step are updated by accumulating the iterative increments:

$$\Delta\lambda_i^j = \Delta\lambda_i^{j-1} + \delta\lambda_i^j \tag{3.12}$$

$$\Delta\boldsymbol{U}_i^j = \Delta\boldsymbol{U}_i^{j-1} + \delta\boldsymbol{U}_i^j \tag{3.13}$$

The total values of the load ratio and nodal displacements for the entire analysis are updated as follows:

$$\lambda_i^j = \lambda_{i-1} + \Delta\lambda_i^j \tag{3.14}$$

$$\boldsymbol{U}_i^j = \boldsymbol{U}_{i-1} + \Delta\boldsymbol{U}_i^j \tag{3.15}$$

Equation (3.16) brings the residual force vector after the *j*-th iteration, given by the difference between the total values of external and internal nodal forces.

$$\boldsymbol{R}_i^j = \lambda_i^j \bar{\boldsymbol{P}} - \boldsymbol{F}\left(\boldsymbol{U}_i^j\right) \tag{3.16}$$

Assuming that, in the iterative solution, we have evaluated $\boldsymbol{U}_i^{j\text{-}1}$ and the iterative increments are sufficiently small, the residual forces can be obtained by a Taylor series expansion up to the first derivatives about the conditions at step *i*, iteration *j*-1:

$$\boldsymbol{R}_i^j \cong \boldsymbol{R}_i^{j-1} + \left.\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{U}}\right|_{\boldsymbol{U}_i^{j-1}} \delta\boldsymbol{U}_i^j + \left.\frac{\partial \boldsymbol{R}}{\partial \lambda}\right|_{\boldsymbol{U}_i^{j-1}} \delta\lambda_i^j + \text{higher order terms} \tag{3.17}$$

Notice that, considering the externally applied forces to be independent of displacements, the derivatives of the residual forces with respect to nodal displacements result in the tangent stiffness matrix (with negative sign). Similarly, the derivatives of the residual forces with respect to the load ratio result in the reference load vector:

$$\left.\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{U}}\right|_{\boldsymbol{U}_i^{j-1}} = \left.\left(\cancel{\frac{\partial \boldsymbol{P}}{\partial \boldsymbol{U}}} - \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{U}}\right)\right|_{\boldsymbol{U}_i^{j-1}} = -\left.\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{U}}\right|_{\boldsymbol{U}_i^{j-1}} = -\boldsymbol{K}_i^{j-1} \tag{3.18}$$

$$\left.\frac{\partial \boldsymbol{R}}{\partial \lambda}\right|_{\boldsymbol{U}_i^{j-1}} = \left.\left(\frac{\partial \boldsymbol{P}}{\partial \lambda} - \cancel{\frac{\partial \boldsymbol{F}}{\partial \lambda}}\right)\right|_{\boldsymbol{U}_i^{j-1}} = \left.\frac{\partial \boldsymbol{P}}{\partial \lambda}\right|_{\boldsymbol{U}_i^{j-1}} = \bar{\boldsymbol{P}} \tag{3.19}$$

Using Eq. (3.18) and Eq. (3.19) in Eq. (3.17), considering the residual forces in the current iteration to be null ($\boldsymbol{R}_i^j = 0$), as it is desired, and neglecting higher order terms, it is possible to obtain the governing incremental-iterative system of finite element equations to be solved in the *j*-th iteration of the *i*-th step, to achieve equilibrium by means of the Newton-Raphson iteration scheme. The resulting system to compute the iterative increments of displacements is given in Eq. (3.20), whereas the improved, or corrected, displacements solution is given in Eq. (3.21). These equations were obtained by linearizing the response of the finite element system about the conditions at step *i*, iteration *j*-1, by means of a Taylor series expansion. The residual forces are calculated according to Eq. (3.16).

$$\boldsymbol{K}_i^{j-1}\delta\boldsymbol{U}_i^j = \delta\lambda_i^j\bar{\boldsymbol{P}} + \boldsymbol{R}_i^{j-1} \tag{3.20}$$

$$\boldsymbol{U}_i^j = \boldsymbol{U}_i^{j-1} + \delta\boldsymbol{U}_i^j \tag{3.21}$$

The initial conditions to start the iterative process of the *i*-th incremental step are the results obtained at the end of the previous step:

$$\lambda_i^0 = \lambda_{i-1} \tag{3.22}$$

$$\boldsymbol{U}_i^0 = \boldsymbol{U}_{i-1} \tag{3.23}$$

The Newton-Raphson iteration scheme is the most frequently used for the solution of nonlinear finite element equations because of its convergence properties, as presented by Bertsekas (1982). An important property of this scheme is that if the current iterative solution is sufficiently close to the real solution, if the exact tangent stiffness matrix is used in the iterations (evaluated consistently with Eq. (3.9)), and if the tangent matrix does not change abruptly, we can expect a quadratic convergence (Bathe, 1996). This means that if the error after iteration *j* is of order *e*, then the error after iteration *j*+1 will be of order e^2. On the other hand, if these conditions are not satisfied, then the iteration may diverge. The primary procedure for reaching convergence, if convergence difficulties are encountered, is to decrease the magnitude of the load step.

In the standard Newton-Raphson iteration scheme, the tangent stiffness matrix is updated in all iterations, considering the last obtained geometric configuration, $\boldsymbol{U}_i^{j-1}$, and internal forces, $\boldsymbol{F}(\boldsymbol{U}_i^{j-1})$. However, in general, the major computational cost per iteration lies in the calculation and factorization of the tangent stiffness matrix. Since these calculations can be quite expensive when large-order systems are considered, the use of a modification of the Newton-Raphson algorithm can be effective. In the modified Newton-Raphson iteration scheme, the tangent matrix is only evaluated at the beginning of each incremental step and held constant for all subsequent iterations, i.e., $\boldsymbol{K}_i^{j-1} = \boldsymbol{K}_i^1$ for $j > 1$.

The modified scheme has a lower computational cost at each iteration than the standard version, but convergence is usually slower because it typically requires more iterations in each incremental step. However, the additional effort of the extra iterations performed by the modified scheme is often offset by the substantial savings realized as a result of not having to assemble and decompose a new tangent stiffness matrix in all iterations. The use of the modified scheme to update the tangent matrix may result in convergence problems in the analysis of structures that exhibit significant stiffening behavior as a result of large internal tensile forces (McGuire *et al.*, 2000).

The overall behavior of the two types of Newton-Raphson iteration schemes is illustrated in Fig. 3.2 for a single degree-of-freedom, where point A is the start of the incremental step and point B is the converged equilibrium solution.

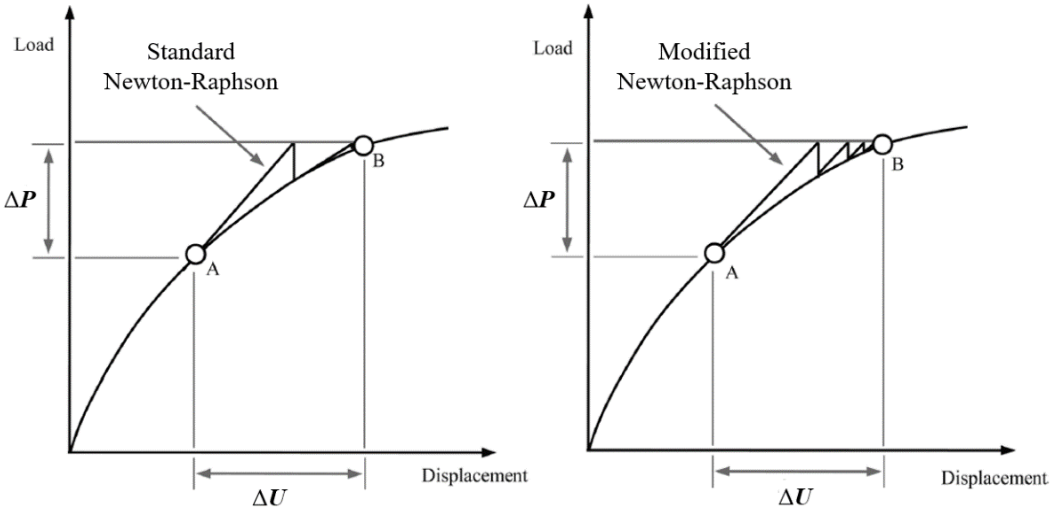


Figure 3.2 – Versions of the Newton-Raphson iteration scheme (adapted from Leon *et al.*, 2011)

3.3.3. Augmented Incremental-Iterative System

The linearized system of Eq. (3.20) has N+1 unknowns: N components of displacement increment in $\delta\boldsymbol{U}_i^j$ and one load ratio increment $\delta\lambda_i^j$; but only N equations. It is necessary to add a constraint equation to that system. The general form of this additional equation is given in Eq. (3.24), where vector $\boldsymbol{A}$ and scalars B and C are constants that can assume different values depending on the solution method to which this equation is associated (Yang & Kuo, 1994).

$$\boldsymbol{A}_i^j \cdot \delta\boldsymbol{U}_i^j + B_i^j \delta\lambda_i^j = C_i^j \tag{3.24}$$

The governing system of Eq. (3.20) and the constraint equation of Eq. (3.24) yield an augmented system of N+1 equations and unknowns:

$$\begin{bmatrix} \boldsymbol{K}_i^{j-1} & -\bar{\boldsymbol{P}} \\ \left(\boldsymbol{A}_i^j\right)^T & B_i^j \end{bmatrix} \begin{Bmatrix} \delta\boldsymbol{U}_i^j \\ \delta\lambda_i^j \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{R}_i^{j-1} \\ C_i^j \end{Bmatrix} \tag{3.25}$$

The augmented system matrix is no longer symmetric and has an increased bandwidth due to the added load ratio. The solution of this system would be computationally undesirable with respect to both storage and efficiency. However, Batoz & Dhatt (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:

$$\begin{cases} \boldsymbol{K}_i^{j-1} \delta\bar{\boldsymbol{U}}_i^j = \bar{\boldsymbol{P}} \\ \boldsymbol{K}_i^{j-1} \delta\bar{\bar{\boldsymbol{U}}}_i^j = \boldsymbol{R}_i^{j-1} \end{cases} \tag{3.26}$$

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

$$\delta\boldsymbol{U}_i^j = \delta\lambda_i^j \delta\bar{\boldsymbol{U}}_i^j + \delta\bar{\bar{\boldsymbol{U}}}_i^j \tag{3.27}$$

The unknown iterative increment of load ratio is given by the constraint equation, which is associated with a particular nonlinear solution method that gives rise to the constraint coefficients $\boldsymbol{A}$, B, and C of Eq. (3.24). Replacing Eq. (3.27) into Eq. (3.24), one can notice, in Eq. (3.28), that the constraint equation is also a function of the tangent and residual components of the iterative increment of displacements. Therefore, the linear systems of Eq. (3.26) must be solved prior to computing the iterative increment of load ratio.

$$\delta\lambda_i^j = \frac{C_i^j - \boldsymbol{A}_i^j \cdot \delta\bar{\bar{\boldsymbol{U}}}_i^j}{B_i^j + \boldsymbol{A}_i^j \cdot \delta\bar{\boldsymbol{U}}_i^j} \tag{3.28}$$

It is also important to mention that the addition of the constraint equation to the system permits an adjustment of both the displacements and the load ratio during the iterative cycle and, therefore, allows the algorithms to advance in the solution beyond load limit and displacement limit points.

The incremental-iterative solution process is summarized with Equations (3.26), (3.28), and (3.27) to calculate the iterative increments of load ratio and nodal displacements. Equations (3.12) to (3.15) are used to update the iterative results with Equations (3.22) and (3.23) as initial conditions for the incremental step. Finally, the residual forces are calculated using Eq. (3.16). Table 3.1 organizes these important equations.

Table 3.1 – Summary of equations for solving the incremental-iterative system

Tangent and residual increments	$\boldsymbol{K}_i^{j-1}\delta\bar{\boldsymbol{U}}_i^j = \bar{\boldsymbol{P}}$
	$\boldsymbol{K}_i^{j-1}\delta\bar{\bar{\boldsymbol{U}}}_i^j = \boldsymbol{R}_i^{j-1}$
Iterative increments	$\delta\lambda_i^j = \left(C_i^j - \boldsymbol{A}_i^j \cdot \delta\bar{\bar{\boldsymbol{U}}}_i^j\right)\Big/\left(B_i^j + \boldsymbol{A}_i^j \cdot \delta\bar{\boldsymbol{U}}_i^j\right)$
	$\delta\boldsymbol{U}_i^j = \delta\lambda_i^j\delta\bar{\boldsymbol{U}}_i^j + \delta\bar{\bar{\boldsymbol{U}}}_i^j$
Step increments	$\Delta\lambda_i^j = \Delta\lambda_i^{j-1} + \delta\lambda_i^j$
	$\Delta\boldsymbol{U}_i^j = \Delta\boldsymbol{U}_i^{j-1} + \delta\boldsymbol{U}_i^j$
Total values	$\lambda_i^j = \lambda_{i-1} + \Delta\lambda_i^j$
	$\boldsymbol{U}_i^j = \boldsymbol{U}_{i-1} + \Delta\boldsymbol{U}_i^j$
Residual forces	$\boldsymbol{R}_i^j = \lambda_i^j\bar{\boldsymbol{P}} - \boldsymbol{F}\left(\boldsymbol{U}_i^j\right)$

The incremental single-step solution process, in turn, works only with the purely incremental system of Eq. (3.11), as explained in Section 3.5.1.

The process of solving the purely incremental or the incremental-iterative system can be split into two phases within each step, called *predictor* phase and *corrector* phase. For the methods with iterative solutions, the *predictor* phase is equivalent to the first iteration, and the remaining iterations correspond to the *corrector* phase. This is an alternative way of presenting and implementing the solution process, rather than performing all iteration in the same loop. The advantage is to make explicit the role of each phase and to combine different increment strategies with correction strategies. For purely incremental methods, this division makes no difference.

Details of the strategies that can be adopted in each solution phase, for both incremental single-step and incremental-iterative methods, are described in the following sections. In Section 3.6, a summary is presented with a flow-chart of all the general steps to solve the nonlinear problem by the mentioned classes of incremental methods. The reader is encouraged to take a look at Fig. 3.15 before proceeding.

3.4. Predicted Solution

In each incremental step, the *predictor* phase is executed first. The purpose is to calculate a predicted solution with a single linear analysis that uses the tangent stiffness matrix evaluated at the beginning of the step.

This phase, for the *i*-th incremental step, starts with the evaluation of the tangent stiffness matrix, $\boldsymbol{K}_i^0$, based on the results obtained at the end of the previous step (nodal displacements, $\boldsymbol{U}_{i\text{-}1}$, and internal forces, $\boldsymbol{F}(\boldsymbol{U}_{i\text{-}1})$). The tangent increment of displacements, $\delta\bar{\boldsymbol{U}}_i^j$, is then calculated with a linear analysis, according to the first line of Eq. (3.26). The residual increment of displacements in this phase is null ($\delta\bar{\bar{\boldsymbol{U}}}_i^j = \boldsymbol{0}$), because residual forces coming from the previous step are disregarded. The predicted increment of displacements, $\delta\boldsymbol{U}_i^1$, is obtained according to Eq. (3.27), but only with the product of the tangent increment of displacements by an increment of load ratio ($\delta\boldsymbol{U}_i^j = \delta\lambda_i^j\delta\bar{\boldsymbol{U}}_i^j$). This load ratio increment is computed with a constraint equation that defines a hyper-surface to restrict the corrective solutions of the incremental-iterative methods. Figure 3.3 illustrates a schematic of the *predictor* phase for a single degree-of-freedom system.

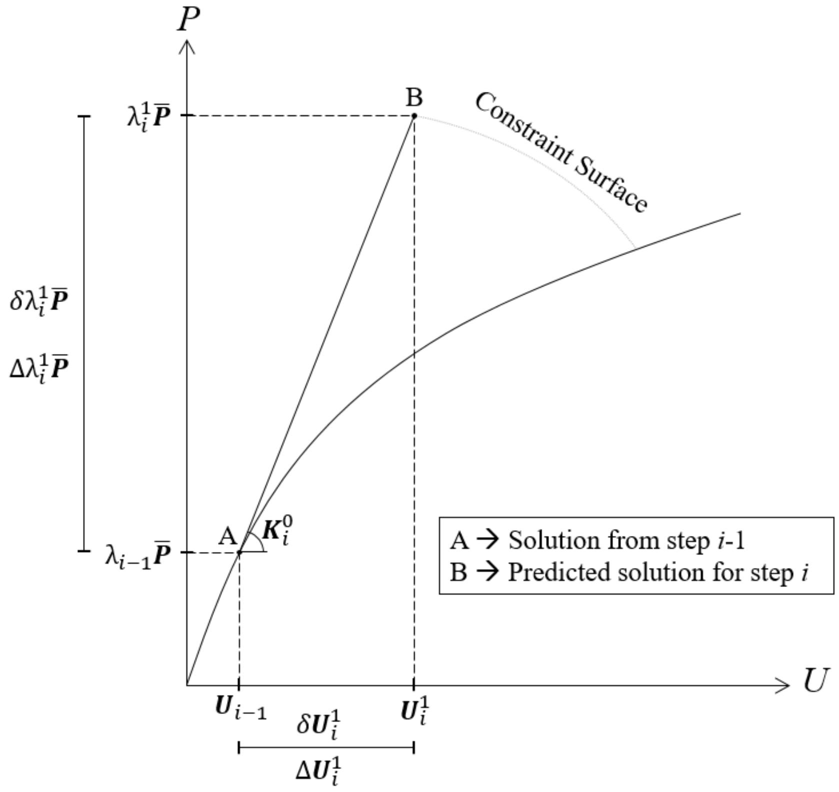

P
B
Constraint Surface
$\lambda_i^1\bar{\boldsymbol{P}}$
$\delta\lambda_i^1\bar{\boldsymbol{P}}$
$\Delta\lambda_i^1\bar{\boldsymbol{P}}$
A
$\boldsymbol{K}_i^0$
$\lambda_{i-1}\bar{\boldsymbol{P}}$
A → Solution from step *i*-1
B → Predicted solution for step *i*
U
$\boldsymbol{U}_{i-1}$
$\delta\boldsymbol{U}_i^1$
$\boldsymbol{U}_i^1$
$\Delta\boldsymbol{U}_i^1$

Figure 3.3 – Schematic of the *predictor* phase

Obtaining the predicted solution has as fundamental task the calculation of the predicted increment of load ratio. In the first incremental step of the analysis (i = 1), the predicted increment of load ratio must be a prescribed value by the analyst. According to McGuire *et al.* (2000), this value should normally be about 10% to 20% of the maximum applied load. In the remaining steps (i > 1), it is computed according to the constraint equation of the selected strategy.

The size of the predicted increment of load ratio employed in each step of the analysis can have a dramatic effect on the solution. In the single-step methods, proper selection of this increment is the only means for controlling drift-off error. In the incremental-iterative methods, a poor definition of this value could result in convergence problems. In both methods, an excessively small increment may result in significant computational effort with a negligible increase in accuracy.

Therefore, a good algorithm must be able to automatically adjust the size of the predicted increment, according to the degree of nonlinearity of the system. This automatic adjustment should provide large increments when the response is almost linear, in order to reduce the computational cost, and lead to small increments when the response is strongly nonlinear, so drift-off error is reduced or the iterative cycle can converge to a new equilibrium configuration.

The option of whether or not to use the automatic adjustment of the load ratio increment is usually available on computer programs, as in the case of the developed tool. However, it is recommended that the option to automatically adjust the increment be selected. Thus, the increments of all steps do not depend only on the value provided for the first step, which can be a bad value if the user does not know what to expect from the behavior of the structure.

In addition, when the solution passes a load limit point and reaches an unstable equilibrium configuration, the load increment necessary to continuing tracing the equilibrium path must be a negative value, for a positive increment of displacement. Hence, the algorithm must also be able to choose the correct sign for the increment, allowing continuation methods to go beyond load limit points.

The strategies for calculating the value of the predicted increment of load ratio and choosing the correct sign of this increment are presented in the next sections.

3.4.1. Increment Strategies

Two classes of strategy to compute the predicted increment are usually found in the literature of nonlinear analysis of structures. Each one has a different way of adjusting the size of the increment to account for the system nonlinearity. One is based on the number of performed iterations and the other is based on the stiffness of the system. Both of them are presented next.

3.4.1.1. Strategies Based on the Number of Iterations

The idea of these strategies starts by setting the increment of a control parameter, φ, as the same value of the increment of this parameter in the previous step, according to Eq. (3.29). The selection of the control parameter is what distinguishes the strategies. The most common parameters are the load ratio, the external work, and an arc-length.

$$\delta\varphi_i^1 = \delta\varphi_{i-1}^1 \tag{3.29}$$

The corresponding increment of load ratio, necessary to provide the prescribed increment of the control parameter, is then computed. Thus, when the control parameter is not the load ratio itself, a coefficient, *H*, is necessary to convert the increment of the control parameter to the increment of load ratio. The constraint equation becomes:

$$\delta\lambda_i^1 = \sqrt{|H|}\left|\delta\lambda_{i-1}^1\right| \tag{3.30}$$

The load ratio increment is adjusted by selecting the correct sign and multiplying it by an adjustment factor that measures the degree of nonlinearity of the system based on the number of performed iterations. The general form of the expression for obtaining the predicted increment of load ratio in the *i*-th step, for *i* > 1, is given in Eq. (3.31), where *J* is the adjustment factor of the increment size and the correct sign is selected according to the criterion presented in Section 3.4.2.

$$\delta\lambda_i^1 = \pm J\sqrt{|H|}\left|\delta\lambda_{i-1}^1\right| \tag{3.31}$$

The adjustment factor of these strategies, as introduced by Ramm (1981), is given in Eq. (3.32). In that expression, N_i and $N_{i\text{-}1}$ are, respectively, the desired number of iterations for the current step, which is a user input data, and the number of iterations required to achieve convergence in the previous step. The exponential variable η typically ranges from 0.5 to 1.0, but the lower value is usually employed, as suggested by Ramm (1982) and Krenk (2009).

$$J = \left(\frac{N_i}{N_{i-1}}\right)^{\eta} \tag{3.32}$$

In the next sections, the formulation of each increment strategy is developed, to arrive at the expressions for the converting factor *H* of the corresponding control parameter, and determine the predicted increment of the load ratio.

Some remarks should be made:

- Since these strategies depend on the number of iterations to measure the degree of nonlinearity of the systems, they are only applicable to incremental-iterative solution methods.
- The constraint equation, defined by the increment of the control parameter, is used in the iterative corrections of some incremental-iterative methods to keep this increment constant during the entire step. Therefore, in this case, the increment of the control parameter to reach the predicted solution is the same increment of the end of the step, and Eq. (3.29) could be expressed as a function of the total increment of the previous step:

$$\delta\varphi_i^1 = \Delta\varphi_{i-1} \tag{3.33}$$

However, in some iterative strategies, the predicted increment of the control parameter is not kept constant throughout the step. The Linear Arc-Length, Minimum Norm, and Orthogonal Residue control methods are some examples. In these cases, the value of the increment of the control parameter at the end of the iterative cycle is different from the predicted value. Therefore, the constraint equation developed in terms of the total increments of the previous step, according to Eq. (3.33), should be applied for the prediction of the load ratio increment in the iterative strategies that do not preserve the increment of the control parameter.

Furthermore, the constraint equation in terms of the total incremental values of the previous step is easier to implement, since these are information that are naturally stored to update the results. In contrast, the information about the predicted solution of the previous step would have to be stored just for computing the constraint equation by means of Eq. (3.29). Both expressions for the constraint equation, developed according to Eq. (3.29) and Eq. (3.33), are presented in the next sections.

- As previously commented, in the first step of the analysis the predicted increment of load ratio is a prescribed value by the analyst. The reason for requiring the increment of the load ratio, and not the increment of the control parameter of the selected strategy, is that a user has a much better sense of the load increment to be taken than of a more abstract parameter such as an external work or an arc-length.
- In the implemented algorithm, a check is made to verify if the predicted increment results in a total load ratio greater than the prescribed maximum value for the analysis. In this case, the load ratio increment is limited to the difference between the maximum value and the total load ratio from the previous step ($\delta\lambda_i^1 = \lambda_{\max} - \lambda_{i-1}$).

3.4.1.1.1. Load Increment

In this strategy, the predicted increment of load ratio is directly obtained from the previous step, so the constraint equation of Eq. (3.30) is used with unit value assigned to the coefficient *H*. Notice that this constraint equation corresponds to assuming the values given in Eq. (3.35) to the constraint coefficients of Eq. (3.28).

$$\delta\lambda_i^1 = \left|\delta\lambda_{i-1}^1\right| \tag{3.34}$$

$$\boldsymbol{A}_i^1 = \boldsymbol{0} \quad , \quad B_i^1 = 1 \quad , \quad C_i^1 = \left|\delta\lambda_{i-1}^1\right| \tag{3.35}$$

By applying the correct sign and the size adjustment factor:

$$\delta\lambda_i^1 = \pm J\left|\delta\lambda_{i-1}^1\right| \tag{3.36}$$

This increment strategy is appropriate to be used by the Load Control Method (Section 3.5.2.1), with the purpose to keep the predicted increment of load ratio constant after each iteration. Alternatively, the total increment of load ratio of the previous step could be employed:

$$\delta\lambda_i^1 = \pm J\left|\Delta\lambda_{i-1}\right| \tag{3.37}$$

3.4.1.1.2. External Work Increment

The control parameter to be incremented in this strategy is the external work, *W*, applied to the system. Equation (3.29) becomes:

$$\delta W_i^1 = \delta W_{i-1}^1 \tag{3.38}$$

The work increment to obtain the predicted solution can be expressed by the product of the increment of external forces by the increment of displacements:

$$\delta W_i^1 = \delta\lambda_i^1 \bar{\boldsymbol{P}} \cdot \delta\boldsymbol{U}_i^1 \tag{3.39}$$

Equation (3.27) is substituted into $\delta\boldsymbol{U}_i{}^1$, with residual increments being zero, which results in:

$$\delta W_i^1 = \left(\delta\lambda_i^1\right)^2 \bar{\boldsymbol{P}} \cdot \delta\bar{\boldsymbol{U}}_i^1 \tag{3.40}$$

The increment of external work of the previous step is similarly defined as:

$$\delta W_{i-1}^1 = \left(\delta\lambda_{i-1}^1\right)^2 \bar{\boldsymbol{P}} \cdot \delta\bar{\boldsymbol{U}}_{i-1}^1 \tag{3.41}$$

Therefore, the constraint equation to find the increment of load ratio corresponding to the increment of external work can be obtained as follows:

$$\left(\delta\lambda_i^1\right)^2 \bar{\boldsymbol{P}}\cdot\delta\bar{\boldsymbol{U}}_i^1 = \left(\delta\lambda_{i-1}^1\right)^2 \bar{\boldsymbol{P}}\cdot\delta\bar{\boldsymbol{U}}_{i-1}^1 \tag{3.42}$$

$$\delta\lambda_i^1 = \pm\left|\delta\lambda_{i-1}^1\right|\sqrt{\left|\frac{\bar{\boldsymbol{P}}\cdot\delta\bar{\boldsymbol{U}}_{i-1}^1}{\bar{\boldsymbol{P}}\cdot\delta\bar{\boldsymbol{U}}_i^1}\right|} \tag{3.43}$$

The constraint coefficients employed in Eq. (3.28) to arrive at the same constraint equation are:

$$\boldsymbol{A}_i^1 = \delta\lambda_i^1\bar{\boldsymbol{P}} \quad , \quad B_i^1 = 0 \quad , \quad C_i^1 = \delta W_{i-1}^1 \tag{3.44}$$

The adjusted increment of load ratio is obtained by imposing the adjustment factor *J* to Eq. (3.43):

$$\delta\lambda_i^1 = \pm J\left|\delta\lambda_{i-1}^1\right|\sqrt{\left|\frac{\bar{\boldsymbol{P}}\cdot\delta\bar{\boldsymbol{U}}_{i-1}^1}{\bar{\boldsymbol{P}}\cdot\delta\bar{\boldsymbol{U}}_i^1}\right|} \tag{3.45}$$

where the coefficient *H* can be identified as:

$$H = \frac{\bar{\boldsymbol{P}}\cdot\delta\bar{\boldsymbol{U}}_{i-1}^1}{\bar{\boldsymbol{P}}\cdot\delta\bar{\boldsymbol{U}}_i^1} \tag{3.46}$$

When the first step (*i* = 1) is taken as the reference for adjusting the increment of external work, this coefficient is known as the Current Stiffness Parameter (CSP), and this work-based strategy is interpreted as a stiffness-based strategy, described in Section 3.4.1.2.1.

The External Work Increment strategy should be used together with the Work Control Method of iterative corrections (Section 3.5.2.2) to maintain the same work increment during the entire step. In this case, the total increment of external work of the previous step could also be used to develop the constraint equation, according to Eq. (3.33). The result is:

$$\delta\lambda_i^1 = \pm J\sqrt{\left|\frac{\Delta\lambda_{i-1}\bar{\boldsymbol{P}}\cdot\Delta\boldsymbol{U}_{i-1}}{\bar{\boldsymbol{P}}\cdot\delta\bar{\boldsymbol{U}}_i^1}\right|} \tag{3.47}$$

Some weaknesses of the External Work Increment strategy have been examined by Yang & Shieh (1990). For structures with a small number of degrees-of-freedom, and which the major forcing directions present a snap-back behavior, the denominator of Eq. (3.46) tends to zero, forcing the increment to infinity. For this reason, this strategy has limited success when facing displacement limit points.

3.4.1.1.3. Arc-Length Increment

This strategy considers an arc-length, L, as the control parameter to be incremented:

$$\delta L_i^1 = \delta L_{i-1}^1 \tag{3.48}$$

The arc-length is calculated via a norm of the predicted increment:

$$\delta L_i^1 = \left(\delta \boldsymbol{U}_i^1 \cdot \delta \boldsymbol{U}_i^1 + \beta \left(\delta\lambda_i^1 \right)^2 \bar{\boldsymbol{P}} \cdot \bar{\boldsymbol{P}} \right)^{1/2} \tag{3.49}$$

where β is a non-negative real parameter that defines different versions of the arc-length measurement: cylindrical arc-length (β = 0), spherical arc-length (β = 1), and elliptical arc-length (0 < β < 1). Only the cylindrical and spherical versions were implemented.

Using Eq. (3.27), with residual increments being zero, the arc-length increment is expressed as function of the tangent increment of displacements:

$$\delta L_i^1 = \left(\left(\delta\lambda_i^1 \right)^2 \delta \bar{\boldsymbol{U}}_i^1 \cdot \delta \bar{\boldsymbol{U}}_i^1 + \beta \left(\delta\lambda_i^1 \right)^2 \bar{\boldsymbol{P}} \cdot \bar{\boldsymbol{P}} \right)^{1/2} \tag{3.50}$$

$$\delta L_i^1 = \delta\lambda_i^1 \left(\delta \bar{\boldsymbol{U}}_i^1 \cdot \delta \bar{\boldsymbol{U}}_i^1 + \beta \bar{\boldsymbol{P}} \cdot \bar{\boldsymbol{P}} \right)^{1/2} \tag{3.51}$$

The arc-length increment of the previous step is expressed in the same way:

$$\delta L_{i-1}^1 = \delta\lambda_{i-1}^1 \left(\delta \bar{\boldsymbol{U}}_{i-1}^1 \cdot \delta \bar{\boldsymbol{U}}_{i-1}^1 + \beta \bar{\boldsymbol{P}} \cdot \bar{\boldsymbol{P}} \right)^{1/2} \tag{3.52}$$

Therefore, the constraint equation is obtained as follows:

$$\delta\lambda_i^1 \left(\delta \bar{\boldsymbol{U}}_i^1 \cdot \delta \bar{\boldsymbol{U}}_i^1 + \beta \bar{\boldsymbol{P}} \cdot \bar{\boldsymbol{P}} \right)^{1/2} = \delta\lambda_{i-1}^1 \left(\delta \bar{\boldsymbol{U}}_{i-1}^1 \cdot \delta \bar{\boldsymbol{U}}_{i-1}^1 + \beta \bar{\boldsymbol{P}} \cdot \bar{\boldsymbol{P}} \right)^{1/2} \tag{3.53}$$

$$\delta\lambda_i^1 = \left| \delta\lambda_{i-1}^1 \right| \sqrt{ \frac{ \delta \bar{\boldsymbol{U}}_{i-1}^1 \cdot \delta \bar{\boldsymbol{U}}_{i-1}^1 + \beta \bar{\boldsymbol{P}} \cdot \bar{\boldsymbol{P}} }{ \delta \bar{\boldsymbol{U}}_i^1 \cdot \delta \bar{\boldsymbol{U}}_i^1 + \beta \bar{\boldsymbol{P}} \cdot \bar{\boldsymbol{P}} } } \tag{3.54}$$

where the coefficient H can be identified as:

$$H = \frac{ \delta \bar{\boldsymbol{U}}_{i-1}^1 \cdot \delta \bar{\boldsymbol{U}}_{i-1}^1 + \beta \bar{\boldsymbol{P}} \cdot \bar{\boldsymbol{P}} }{ \delta \bar{\boldsymbol{U}}_i^1 \cdot \delta \bar{\boldsymbol{U}}_i^1 + \beta \bar{\boldsymbol{P}} \cdot \bar{\boldsymbol{P}} } \tag{3.55}$$

This constraint equation is equivalent to assuming the following values for the constraint coefficients of Eq. (3.28):

$$\boldsymbol{A}_i^1 = \delta\lambda_i^1 \delta \bar{\boldsymbol{U}}_i^1 \quad , \quad B_i^1 = \beta \delta\lambda_i^1 \quad , \quad C_i^1 = \left(\delta L_{i-1}^1 \right)^2 \tag{3.56}$$

With the introduction of the adjustment factor and the appropriate sign, the results for the cylindrical and spherical versions of the arc-length increment are respectively:

$$\delta\lambda_i^1 = \pm J \left|\delta\lambda_{i-1}^1\right| \sqrt{\frac{\delta\bar{\boldsymbol{U}}_{i-1}^1 \cdot \delta\bar{\boldsymbol{U}}_{i-1}^1}{\delta\bar{\boldsymbol{U}}_i^1 \cdot \delta\bar{\boldsymbol{U}}_i^1}} \tag{3.57}$$

$$\delta\lambda_i^1 = \pm J \left|\delta\lambda_{i-1}^1\right| \sqrt{\frac{\delta\bar{\boldsymbol{U}}_{i-1}^1 \cdot \delta\bar{\boldsymbol{U}}_{i-1}^1 + \bar{\boldsymbol{P}} \cdot \bar{\boldsymbol{P}}}{\delta\bar{\boldsymbol{U}}_i^1 \cdot \delta\bar{\boldsymbol{U}}_i^1 + \bar{\boldsymbol{P}} \cdot \bar{\boldsymbol{P}}}} \tag{3.58}$$

Alternatively, the constraint equation can be developed in terms of the total increments of the previous step. The results for the cylindrical and spherical versions are respectively:

$$\delta\lambda_i^1 = \pm J \sqrt{\frac{\Delta\boldsymbol{U}_{i-1} \cdot \Delta\boldsymbol{U}_{i-1}}{\delta\bar{\boldsymbol{U}}_i^1 \cdot \delta\bar{\boldsymbol{U}}_i^1}} \tag{3.59}$$

$$\delta\lambda_i^1 = \pm J \sqrt{\frac{\Delta\boldsymbol{U}_{i-1} \cdot \Delta\boldsymbol{U}_{i-1} + \left(\Delta\lambda_{i-1}\right)^2 \bar{\boldsymbol{P}} \cdot \bar{\boldsymbol{P}}}{\delta\bar{\boldsymbol{U}}_i^1 \cdot \delta\bar{\boldsymbol{U}}_i^1 + \bar{\boldsymbol{P}} \cdot \bar{\boldsymbol{P}}}} \tag{3.60}$$

The increment strategies based on an arc-length should be used by the iteration strategies that restrict the corrected solutions to the corresponding constraint hyper-surface. That is, the Cylindrical Arc-Length Control Method and the Spherical Arc-Length Control Method (Section 3.5.2.3), both for iterative corrections, use the Cylindrical Arc-Length Increment and the Spherical Arc-Length Increment strategies, respectively. The reason is to avoid numerical issues due to the possible inconsistency between increment sizes (Santana, 2015).

Some iteration strategies, however, are not bound to any increment strategy, which is the case of the Linear Arc-Length Control Methods (Section 3.5.2.4), the Minimum Norm Control Method (Section 3.5.2.5), and the Orthogonal Residue Control Method (Section 3.5.2.6). For these iteration strategies, it was opted to use the Cylindrical Arc-Length Increment in the *predictor* phase, due to its better numerical efficiency. The expression of Eq. (3.59), based on the total arc-length increment of the previous step, is used for these iteration techniques. The reason, as previously mentioned, is that the arc-length increment changes after each iteration, and the total increment at the end of the previous step is preferably taken as the reference to predict the increment of the current step.

A shortcoming of the strategies that work with arc-length increments is that they are not consistent in physical units. The expressions involve the scalar product of displacement vectors that contain both translations and rotations, which are different both in units and in orders of magnitudes (Yang & McGuire, 1985). Numerical difficulties of certain sort are likely to occur due to such inconsistencies. This problem is not a concern when dealing with work increments, for example.

3.4.1.2. Strategies Based on the Change of Stiffness

The strategies of this class rely on the stiffness of the system, rather than the number of iterations, to measure its degree of nonlinearity and adjust the size of the predicted increment of load ratio. The adjustment factor of the increment size is a stiffness parameter that relates the stiffness of the system at the beginning of the analysis to the stiffness in the current step. The predicted increment of load ratio in the first step is used as the reference to compute the new adjusted increment. The correct sign must also be imposed.

Equation (3.61) brings the general expression for this type of strategy to obtain the predicted increment of load ratio, for $i > 1$. S is the adjustment factor given by a stiffness parameter, and η can vary from 0.5 to 1.0, but typically assumes the value of 0.5.

$$\delta\lambda_i^1 = \pm\left|S\right|^{\eta}\,\delta\lambda_1^1 \tag{3.61}$$

Different stiffness parameters can be employed as the adjustment factor. Two of them are presented in the following sections.

3.4.1.2.1. Current Stiffness Parameter

The Current Stiffness Parameter (CSP) was presented by Bergan *et al.* (1978). It was proved in Section 3.4.1.1.2 that this parameter corresponds to the conversion coefficient H of the External Work Increment strategy when the first step of the analysis is taken as the reference for adjusting the predicted increment of external work:

$$\text{CSP} = \frac{\bar{\boldsymbol{P}}\cdot\delta\bar{\boldsymbol{U}}_1^1}{\bar{\boldsymbol{P}}\cdot\delta\bar{\boldsymbol{U}}_i^1} \tag{3.62}$$

The CSP is a non-dimensional scalar quantity that was introduced to provide a better way to characterize the overall behavior of multiple degree-of-freedom nonlinear systems. This parameter indicates whether the system is being loaded or unloaded. It has the initial value of unity for any nonlinear system. It is less than unity when the system becomes “softer” than the beginning of the analysis and greater than unity for stiffening systems.

The behavior of this parameter is depicted in Fig. 3.4, for a typical instability problem with snap-through behavior. The horizontal axis of both diagrams is some norm of the displacements vector, intended to characterize the behavior of all degrees-of-freedom with a single scalar value. It can be seen that the unstable behavior of the problem is characterized by a value of the CSP less than zero. Therefore, it is possible to identify load limit points of instability when the stiffness parameter becomes negative.

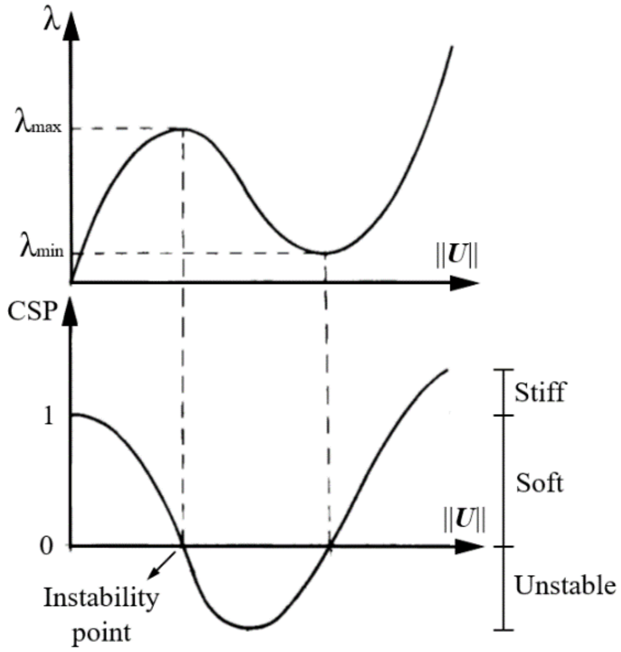


Figure 3.4 – Behavior of the Current Stiffness Parameter (adapted from Bergan *et al.*, 1978)

A drawback of using this parameter is when the structure presents snap-back behavior. As explained in Section 3.4.1.1.2, the quantity in the denominator can approach zero in the vicinity of displacement limit points, leading the CSP to infinity. A more stable stiffness parameter is the Generalized Displacement Parameter, presented next.

3.4.1.2.2. Generalized Stiffness Parameter

The stiffness parameter considered in this work is the Generalized Stiffness Parameter (GSP), introduced by Yang & Kuo (1994). This parameter is used in a method that is based on the increment of a generalized displacement, *D*, which can be expressed by:

$$\delta D_i^j = \delta\lambda_i^j \delta\bar{\boldsymbol{U}}_{i-1}^1 \cdot \delta\boldsymbol{U}_i^j \tag{3.63}$$

The predicted increment of generalized displacement for the current step is compared to the increment of the first step of the analysis. Using Eq. (3.27) to express the generalized displacement in terms of the tangent increment of displacements, and considering $\delta\bar{\boldsymbol{U}}_0^1 = \delta\bar{\boldsymbol{U}}_1^1$, the constraint equation is derived as follows:

$$\delta D_i^1 = \delta D_1^1 \tag{3.64}$$

$$\left(\delta\lambda_i^1\right)^2 \delta\bar{\boldsymbol{U}}_{i-1}^1 \cdot \delta\bar{\boldsymbol{U}}_i^1 = \left(\delta\lambda_1^1\right)^2 \delta\bar{\boldsymbol{U}}_1^1 \cdot \delta\bar{\boldsymbol{U}}_1^1 \tag{3.65}$$

$$\delta\lambda_i^1 = \pm\delta\lambda_1^1 \sqrt{\left|\frac{\delta\bar{\boldsymbol{U}}_1^1 \cdot \delta\bar{\boldsymbol{U}}_1^1}{\delta\bar{\boldsymbol{U}}_{i-1}^1 \cdot \delta\bar{\boldsymbol{U}}_i^1}\right|} \tag{3.66}$$

where the GSP can be identified as:

$$\mathrm{GSP} = \frac{\delta\bar{\boldsymbol{U}}_1^1 \cdot \delta\bar{\boldsymbol{U}}_1^1}{\delta\bar{\boldsymbol{U}}_{i-1}^1 \cdot \delta\bar{\boldsymbol{U}}_i^1} \tag{3.67}$$

Alternatively, the constraint equation could be obtained with the following constraint parameters, suggested by Yang & Shieh (1990) to improve the numerical stability of the solution method:

$$\boldsymbol{A}_i^1 = \delta\lambda_i^1 \delta\bar{\boldsymbol{U}}_{i-1}^1 \quad , \quad B_i^1 = 0 \quad , \quad C_i^1 = \delta D_1^1 \tag{3.68}$$

Just like the use of the CSP, when this strategy is adopted the stiffness of the structure is measured with respect to the first incremental step, so stiffening and softening behavior are readily identified. The GSP will always have an initial value of unity, and stiffening or softening of the structural system are also indicated by values greater or less than one, respectively.

However, a negative value is only assumed in the steps immediately after load limit points. As it can be verified in Eq. (3.67), the numerator is a constant positive value, and the denominator, given by the scalar product of the tangent increment of displacements of the current and previous steps, controls the sign of the expression. The sign of this product indicates whether the increments of the previous and current steps are in the same "direction", as illustrated in Fig. 3.5 for a single degree-of-freedom system. These directions are different when there is a load limit point between the steps.

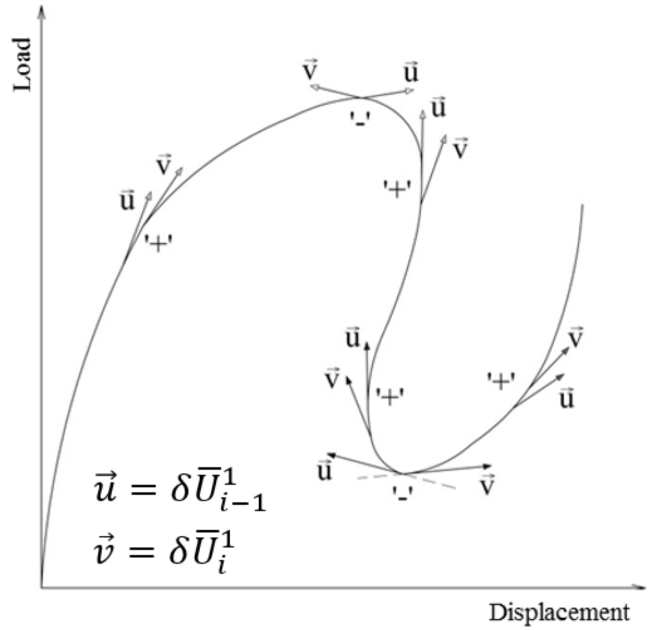

Load
Displacement
$\vec{u}$
$\vec{v}$
'+'
'-'
$\vec{u} = \delta \overline{U}_{i-1}^{1}$
$\vec{v} = \delta \overline{U}_{i}^{1}$

Figure 3.5 – Behavior of the GSP sign (Yang & Kuo, 1994)

This strategy is used in the *predictor* phase of the Generalized Displacement Control Method (Section 3.5.2.7), which performs corrective iterations based on a constant value of generalized displacement increment.

Moreover, in the developed tool, this strategy is also used to adjust the increment size of incremental single-step methods, since the stiffness parameter does not depend on the number of iteration to evaluate the degree of nonlinearity of the solution.

3.4.2. Determination of the Increment Sign

The appropriate sign of the predicted increment can be selected according to several criteria. The most common are based on the properties of the tangent stiffness matrix and the behavior of some stiffness parameter, such as the CSP and the GSP.

Crisfield (1991) suggested that the sign of the load increment should be positive whenever the tangent stiffness matrix of the beginning of the step is positive definite. In his studies, it was shown that the sign of the increment must be inverted when a negative pivot is detected in the factored stiffness matrix using the Crout decomposition algorithm. Equivalently, the sign of the increment must be the same of the previous increment unless the sign of the determinant of the tangent stiffness matrix changes. However, as reported by Meek & Tan (1984), this procedure is not recommended for systems that exhibit multiple negative eigenvalues.

Alternatively, the CSP, presented in Section 3.4.1.2.1, indicates whether the system is being loaded or unloaded. The sign of this parameter should be applied to the load ratio increment: if the parameter is positive, the system is being loaded and the load ratio should increase; if it is negative, the load is decreasing and the load ratio increment should be negative. However, it was mentioned that the CSP might present some numerical instabilities near displacement limit points.

In this work, the criterion to select the appropriate sign for the increment is based on the sign of the GSP, whose behavior was explained in Section 3.4.1.2.2. This parameter is negative only immediately after load limit points. Therefore, the predicted increment sign must be inverted every time the GSP assumes a negative value. The increment sign, represented by *s* in Eq. (3.69), is monitored throughout the analysis:

$$\begin{cases} i = 1 & \rightarrow \quad s = \oplus \\ \text{GSP} < 0 & \rightarrow \quad s = -s \end{cases} \tag{3.69}$$

3.5. Corrected Solution

The *corrector* phase tries to correct the predicted solution. In the case of higher order incremental single-step methods, another linear analysis is performed with an average stiffness matrix that better represents the stiffness of the step, in order to reduce the residual forces generated by the predicted solution. In the case of incremental-iterative solution methods, the residual forces are vanished through a cycle of iterative corrections of load ratio and nodal displacements.

3.5.1. Single-Step Correction

The incremental single-step solution methods all employ a strategy that is analogous to solving systems of differential equations by the Runge-Kutta methods (Butcher, 1996). This class of method performs a linear analysis with an average stiffness matrix that better represents the stiffness of the step, in order to reduce the residual forces generated by the predicted solution, but not necessarily make it numerically null.

The total increment of displacements of the *i*-th step is found by solving the purely incremental system of Eq. (3.11), but with an average stiffness matrix, $\tilde{\boldsymbol{K}}$, that represents the stiffness of the entire increment:

$$\tilde{\boldsymbol{K}}_i \Delta \boldsymbol{U}_i = \delta\lambda_i^1 \bar{\boldsymbol{P}} \tag{3.70}$$

The update process of external forces and nodal displacements follows Eq. (3.3) and Eq. (3.4), respectively, where the total increment of load ratio for the current step is the predicted value ($\Delta\lambda_i = \delta\lambda_i^1$). As mentioned earlier in this work, the adopted strategy to adjust the predicted increment value of load ratio in single-step methods is the GSP-based strategy of Section 3.4.1.2.2.

The fundamental task of higher order single-step methods is obtaining the average stiffness matrix for the current step. Taken as a weighted average, this matrix is written as in Eq. (3.71), where α^m is the weighting coefficient corresponding to one of the *n* sampling points within the increment in which the tangent stiffness matrices, $\boldsymbol{K}_i^m$, are evaluated. Each tangent matrix is evaluated with the total displacements, $\boldsymbol{U}_i^m$, and the corresponding internal forces, $\boldsymbol{F}(\boldsymbol{U}_i^m)$, at the sampling points.

$$\tilde{\boldsymbol{K}}_i = \sum_{m=1}^{n} \alpha^m \boldsymbol{K}_i^m \qquad (3.71)$$

By varying the weighting coefficients and the number and location of the sampling points, various types of incremental single-step methods can be devised from Eq. (3.71). The number of sampling points employed in each step defines the order of the method. In general, the accuracy of the nonlinear response will improve with an increase in this order.

Figure 3.6 shows a schematic of a second-order single-step method, which uses two sampling points, *m1* and *m2*, to take an incremental step in a single degree-of-freedom system. This figure also shows how a linear analysis performed with the average stiffness matrix tends to reduce the residual forces generated by the predicted solution, $\boldsymbol{R}_i$, to a new corrected value, $\tilde{\boldsymbol{R}}_i$. The step positions of the sampling points are represented by the coefficient μ, which are fractions of the predicted increment.

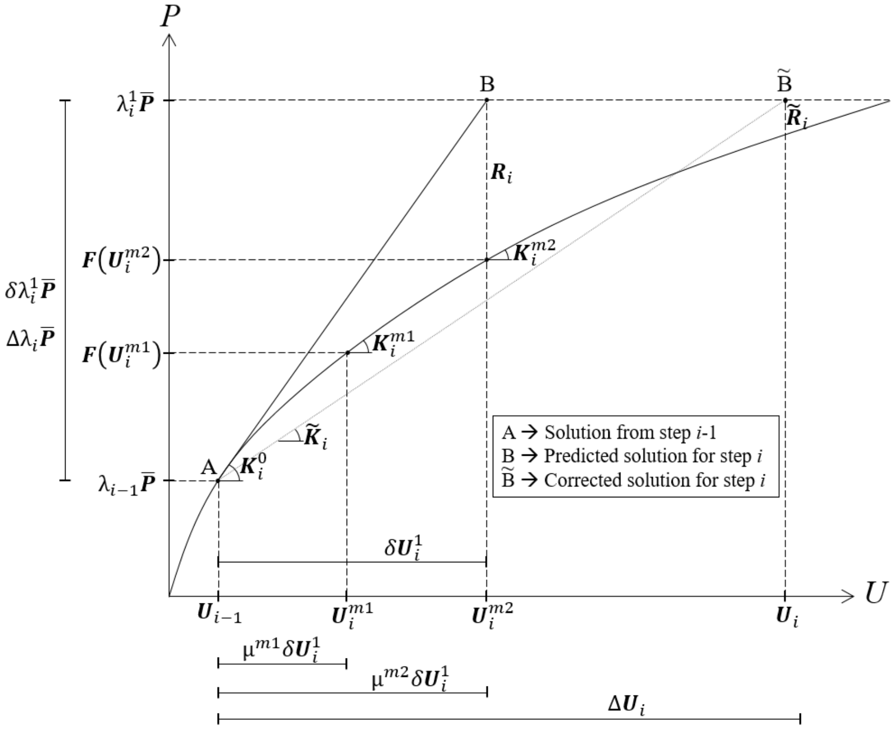
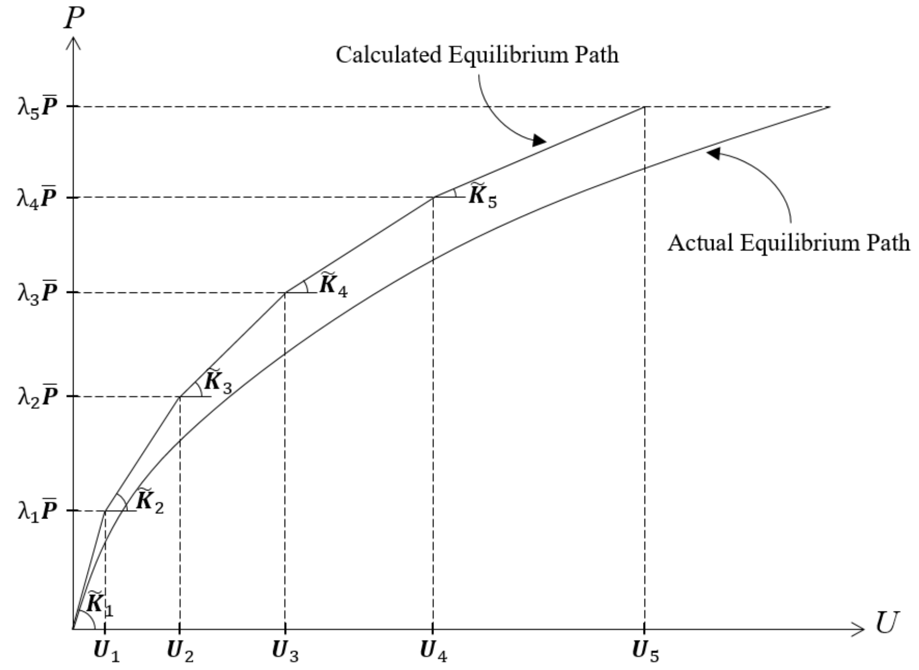

P
B
$\tilde{\text{B}}$
$\lambda_i^1 \bar{\boldsymbol{P}}$
$\tilde{\boldsymbol{R}}_i$
$\boldsymbol{R}_i$
$\boldsymbol{F}(\boldsymbol{U}_i^{m2})$
$\boldsymbol{K}_i^{m2}$
$\delta\lambda_i^1 \bar{\boldsymbol{P}}$
$\Delta\lambda_i \bar{\boldsymbol{P}}$
$\boldsymbol{F}(\boldsymbol{U}_i^{m1})$
$\boldsymbol{K}_i^{m1}$
$\tilde{\boldsymbol{K}}_i$
A
$\boldsymbol{K}_i^0$
$\lambda_{i-1}\bar{\boldsymbol{P}}$
A → Solution from step *i*-1
B → Predicted solution for step *i*
$\tilde{\text{B}}$ → Corrected solution for step *i*
$\delta\boldsymbol{U}_i^1$
U
$\boldsymbol{U}_{i-1}$
$\boldsymbol{U}_i^{m1}$
$\boldsymbol{U}_i^{m2}$
$\boldsymbol{U}_i$
$\mu^{m1}\delta\boldsymbol{U}_i^1$
$\mu^{m2}\delta\boldsymbol{U}_i^1$
$\Delta\boldsymbol{U}_i$

Figure 3.6 – Schematic of an increment with a second-order single-step method

Major advantages of incremental single-step methods are their implementation simplicity and efficiency. The number of linear analyses performed in each increment is the same as the order of the method. For the implemented methods, only one or two analysis are required per step. In this regard, they are particularly attractive for the analysis of structures exhibiting smooth nonlinearity.

A drawback of these methods is that the residual forces are not vanished after the single-step correction, because of the use of a single representative stiffness in each increment. Therefore, a drift-off error is generated in each incremental step and accumulated along the analysis. If the increments are not sufficiently small, this error can lead to very inaccurate responses, as illustrated in Fig. 3.7. Although the drift-off error can be reduced by using smaller increments, the additional number of steps required for analyzing highly nonlinear systems may become unreasonable. In this case, the use of an iterative scheme is more appropriate.



P
Calculated Equilibrium Path
Actual Equilibrium Path
$\lambda_5\bar{\boldsymbol{P}}$
$\lambda_4\bar{\boldsymbol{P}}$
$\lambda_3\bar{\boldsymbol{P}}$
$\lambda_2\bar{\boldsymbol{P}}$
$\lambda_1\bar{\boldsymbol{P}}$
$\tilde{\boldsymbol{K}}_1$
$\tilde{\boldsymbol{K}}_2$
$\tilde{\boldsymbol{K}}_3$
$\tilde{\boldsymbol{K}}_4$
$\tilde{\boldsymbol{K}}_5$
$\boldsymbol{U}_1$
$\boldsymbol{U}_2$
$\boldsymbol{U}_3$
$\boldsymbol{U}_4$
$\boldsymbol{U}_5$
U

Figure 3.7 – Drift-off error resulting from residue accumulation in single-step methods

Three incremental single-step methods were implemented in the developed tool: Euler, Heun, and Midpoint methods. Their properties are discussed next.

3.5.1.1. First-Order Method

The Euler method, or simple step method, is the most elementary. It consists of a first-order Runge-Kutta method that uses an average stiffness matrix calculated with the deformed geometry and corresponding internal forces that exist at the start of the increment. In other words, the Euler method uses only the predicted solution, with no corrections. Therefore, no other procedure, other than updating geometry and external forces, is necessary to advance to the next step after obtaining the predicted solution.

3.5.1.2. Second-Order Methods

The fundamental source of error in the Euler method is the assumption that the stiffness at the beginning of the incremental step can be used for the entire increment. The family of second-order Runge-Kutta methods provides an improvement to the Euler method approach. Such methods use two sampling points to evaluate the tangent stiffness matrices. These matrices are then used to calculate the representative stiffness of the step using Eq. (3.71). Decreasing step size causes the error to decrease at a faster rate than the Euler method (Chapra & Canale, 2013).

3.5.1.2.1. Heun Method

The Heun method takes the average stiffness matrix as the arithmetic mean of the tangent stiffness matrices evaluated at the start of the increment and at the end of the predicted increment. Therefore, the values of the position coefficients (μ^1, μ^2) and corresponding weights (α^1, α^2) for the first and second sampling points for this method are:

$$\begin{aligned} \mu^1 &= 0.0 \quad \mu^2 = 1.0 \\ \alpha^1 &= 0.5 \quad \alpha^2 = 0.5 \end{aligned} \tag{3.72}$$

3.5.1.2.2. Midpoint Method

The Midpoint method, also known as improved polygon method, relies fully on using the stiffness of the middle of the predicted increment as the representative stiffness for the entire step. This consideration is equivalent to set the following values for the position coefficients (μ^1, μ^2) and corresponding weights (α^1, α^2) for the first and second sampling points:

$$\begin{aligned} \mu^1 &= 0.0 \quad \mu^2 = 0.5 \\ \alpha^1 &= 0.0 \quad \alpha^2 = 1.0 \end{aligned} \tag{3.73}$$

3.5.2. Iterative Correction

The corrective phase of incremental-iterative methods aims to restore the structure equilibrium by vanishing the residual forces, generated by the predicted solution, through an iterative cycle of improved solutions. This cycle begins with the update of the total load ratio, λ_i^1, and the total nodal displacements, $\boldsymbol{U}_i^1$, by adding the predicted increments ($\delta\lambda_i^1$ and $\delta\boldsymbol{U}_i^1$) to the results from the previous step ($\lambda_{i\text{-}1}$ and $\boldsymbol{U}_{i\text{-}1}$). With geometry updated, the corresponding internal forces, $\boldsymbol{F}(\boldsymbol{U}_i^1)$, are calculated, and the residual forces, $\boldsymbol{R}_i^1$, can be obtained by the difference between external and internal nodal forces. At this point, convergence is checked.

There are a few different criteria to check the convergence of the iterative process (Bergan & Clough, 1972). The most common are based on the residual forces, residual displacements, or the work produced by these residual results. The adopted convergence criterion for the developed algorithm is a force-based check, which is done according to Eq. (3.74). It consists of analyzing the ratio between the Euclidean norms of the residual force vector and the reference load vector, which must be lower than a given tolerance, ε. This tolerance is a value stipulated by the analysis and is usually in the order of 10^{-5} to 10^{-3}, depending on the desired accuracy. If convergence is satisfied, the predicted solution was sufficient to achieve a tolerable equilibrium state and the algorithm can proceed to the next step, otherwise the first corrective iteration starts.

$$\frac{\left\|\boldsymbol{R}_i^j\right\|}{\left\|\bar{\boldsymbol{P}}\right\|} \leq \varepsilon \tag{3.74}$$

The first procedure of each corrective iteration is evaluating the tangent stiffness matrix considering the nodal displacements and internal forces of the last obtained configuration. If the modified version of the Newton-Raphson iteration scheme is adopted, this step is skipped, and the tangent matrix evaluated at the beginning of the *predictor* phase is used. The tangent and residual increments of displacements are calculated with the reference load vector and the last obtained residual force vector, respectively, as in the linear systems of Eq. (3.26). Then, the iterative increment of load ratio is calculated according to the constraint equation of the corrective method (presented in the following sections). Finally, the iterative increment of displacements is obtained with Eq. (3.27).

The iterative increments of load ratio and displacements are restricted to the hyper-surface defined by the constraint equation that characterizes the selected solution method. If the performed iterations involve not only displacements corrections, but also corrections of load ratio, then it is called a continuation method, because it can continue to trace the equilibrium path beyond limit points. In this case, the constraint surface that controls the corrected solutions crosses the equilibrium path at one or more points.

After obtaining a corrected solution, the next procedures are the same used to check the convergence of the predicted solution: update the total values of load ratio and nodal displacements, calculate external, internal, and residual forces, and finally check the convergence for the current iteration. Figure 3.8 brings a schematic illustration of the described procedures.

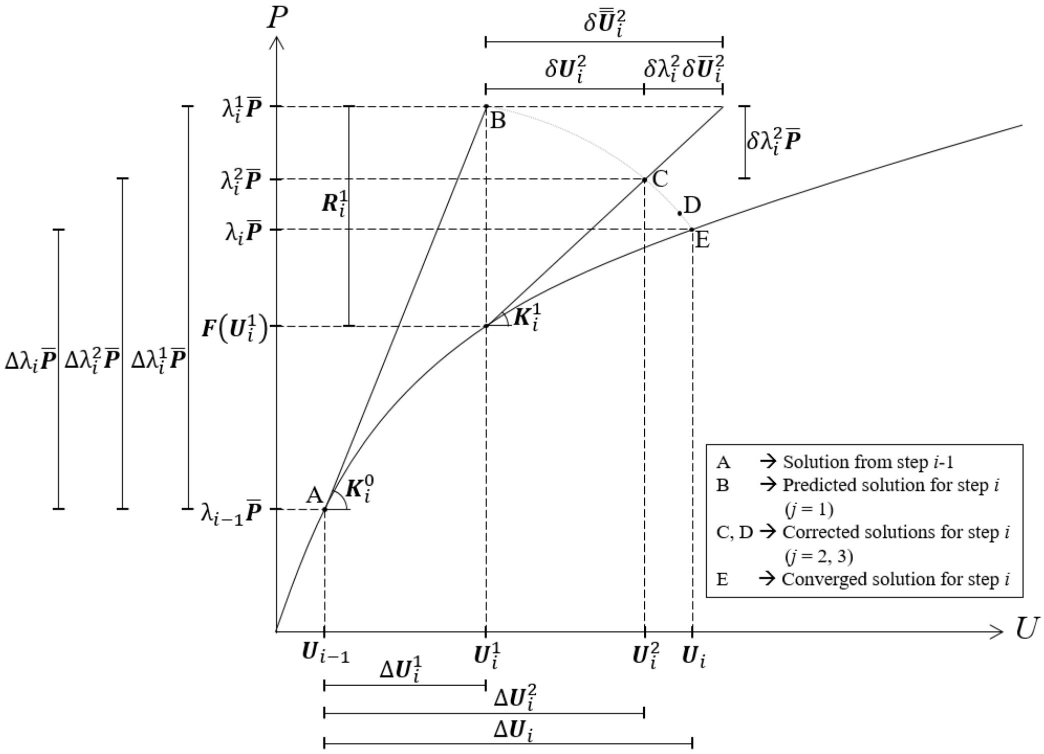

P
$\delta\bar{\bar{U}}_i^2$
δU_i^2
$\delta\lambda_i^2\delta\bar{U}_i^2$
$\lambda_i^1\bar{P}$
$\lambda_i^2\bar{P}$
$\lambda_i\bar{P}$
$F(U_i^1)$
$\lambda_{i-1}\bar{P}$
$\delta\lambda_i^2\bar{P}$
R_i^1
K_i^1
K_i^0
$\Delta\lambda_i\bar{P}$
$\Delta\lambda_i^2\bar{P}$
$\Delta\lambda_i^1\bar{P}$
A
B
C
D
E
A → Solution from step i-1
B → Predicted solution for step i (j = 1)
C, D → Corrected solutions for step i (j = 2, 3)
E → Converged solution for step i
U
U_{i-1}
U_i^1
U_i^2
U_i
ΔU_i^1
ΔU_i^2
ΔU_i

Figure 3.8 – Schematic of iterative corrections

As mentioned in the introduction of this work, one single solution method may not be capable of solving any general nonlinear problem. Therefore, most of the well-known corrective techniques were implemented. Some of these iterative techniques 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 user input data, the solution method refers to the technique for the corrective iterations and it includes the predictive technique that is more appropriate. In the next sections, the characteristics of each corrective technique are described.

It should be mentioned that, in the developed algorithm, it was opted to limit the correction of the load ratio value, in all techniques, to 0.5, avoiding exorbitant results.

3.5.2.1. Load Control Method

The Load Control Method (LCM) is often referred to as the conventional Newton-Raphson method. In this method, a fixed amount of load is employed in each increment and kept constant after each iteration. The entire increment of load ratio is applied in the predicted solution with the Load Increment Method (Section 3.4.1.1.1). Then, corrective iterations are performed in an attempt to satisfy equilibrium requirements with displacement corrections only. Therefore, the iterative increments of load ratio are null during the corrective cycle. For *j* > 1:

$$\delta\lambda_i^j = 0 \tag{3.75}$$

This procedure is illustrated next:

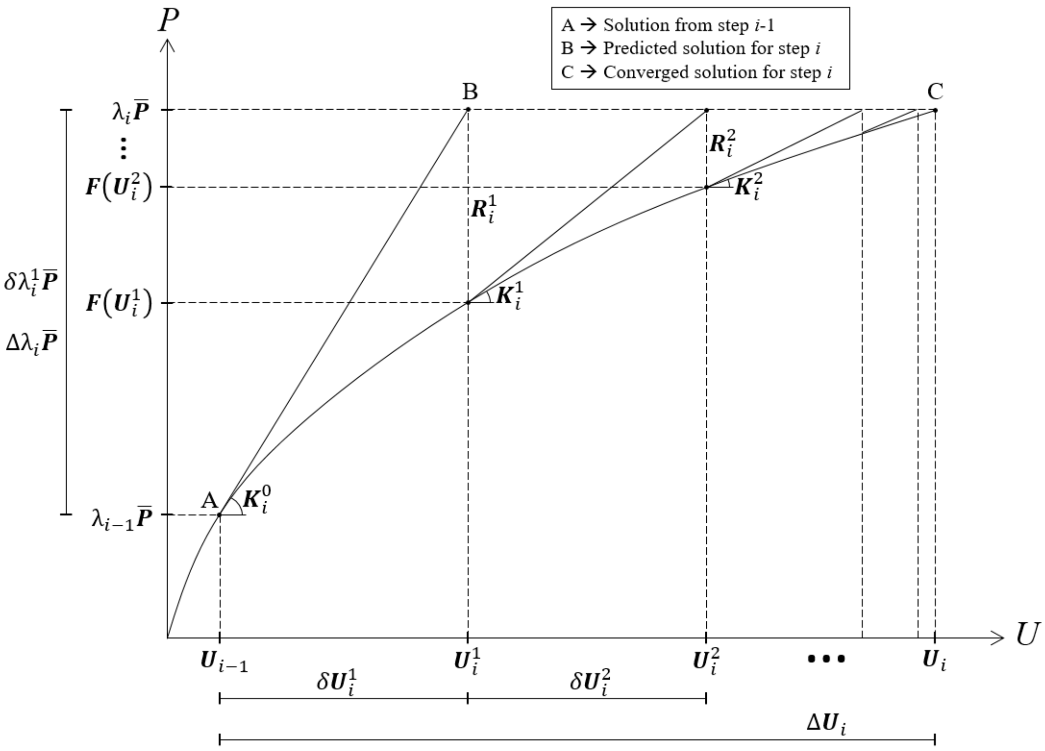

P
A → Solution from step i-1
B → Predicted solution for step i
C → Converged solution for step i
B
C
$\lambda_i \bar{\boldsymbol{P}}$
⋮
$\boldsymbol{F}(\boldsymbol{U}_i^2)$
$\boldsymbol{R}_i^2$
$\boldsymbol{K}_i^2$
$\boldsymbol{R}_i^1$
$\delta\lambda_i^1 \bar{\boldsymbol{P}}$
$\boldsymbol{F}(\boldsymbol{U}_i^1)$
$\boldsymbol{K}_i^1$
$\Delta\lambda_i \bar{\boldsymbol{P}}$
A
$\boldsymbol{K}_i^0$
$\lambda_{i-1} \bar{\boldsymbol{P}}$
U
$\boldsymbol{U}_{i-1}$
$\boldsymbol{U}_i^1$
$\boldsymbol{U}_i^2$
•••
$\boldsymbol{U}_i$
$\delta\boldsymbol{U}_i^1$
$\delta\boldsymbol{U}_i^2$
$\Delta\boldsymbol{U}_i$

Figure 3.9 – Schematic of the Load Control Method

A significant shortcoming of this method becomes apparent when attempting to solve problems with load limit points. Once a fixed load is defined in the *predictor* phase, there is no way to modify the load vector should a limit point occur within the increment. Although performing the steps with a reduced load ratio increment can enable one to approach the limit point slowly, the resulting near singular nature of the stiffness matrix makes it difficult to trace the post-limit state response of a structure.

A typical result when tracing the equilibrium path of systems with snap-through behavior using the LCM is illustrated in Fig. 3.10. For systems with displacement limit points only, no problems should be expected. In practice, this method represents the behavior of structures subjected to load testing.

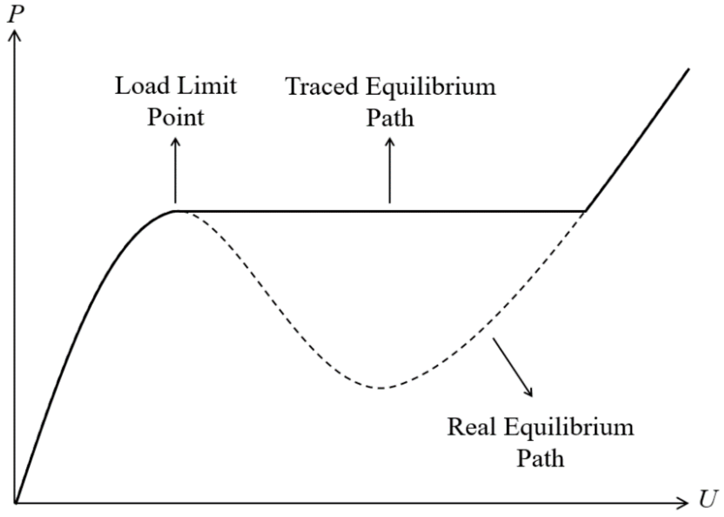


Figure 3.10 – Typical result for snap-through behavior using the Load Control Method

Among the nine incremental-iterative solution methods implemented, only this strategy is not a continuation method. All the others are supposed to capture the real equilibrium path in post-limit analyses, although some limitations should be expected for each one.

An advantage of using the LCM is that, as the corrections of load ratio is always zero, it is not necessary to compute the tangent increment of displacements, so only one linear system of Eq. (3.26) need to be solved, making it computationally more efficient. Therefore, this method should be used when the analyst is sure about the absence of load limit points before the total load is reached.

3.5.2.2. Work Control Method

The Work Control Method (WCM), proposed by Powell & Simons (1981), uses the predictive technique described in Section 3.4.1.1.2 to apply an increment of external work, δW, to the system, which is kept constant in the corrective cycle. Therefore, the constraint condition requires a zero increment of external work for each iteration. For $j > 1$, this is represented by:

$$\delta W_i^j = 0 \tag{3.76}$$

$$\delta\lambda_i^j \bar{\boldsymbol{P}} \cdot \delta \boldsymbol{U}_i^j = 0 \tag{3.77}$$

Substituting Eq. (3.27) into Eq. (3.77), and considering that the increment of load ratio is not null, the constraint equation of Eq. (3.79) is obtained.

$$\delta\lambda_i^j \bar{\boldsymbol{P}} \cdot \left(\delta\lambda_i^j \delta\bar{\boldsymbol{U}}_i^j + \delta\bar{\bar{\boldsymbol{U}}}_i^j \right) = 0 \tag{3.78}$$

$$\delta\lambda_i^j = -\frac{\bar{\boldsymbol{P}} \cdot \delta\bar{\bar{\boldsymbol{U}}}_i^j}{\bar{\boldsymbol{P}} \cdot \delta\bar{\boldsymbol{U}}_i^j} \tag{3.79}$$

This process is illustrated in Fig. 3.11, where it is observed how the external work is conserved by adjusting both displacements and load ratio in each iteration.

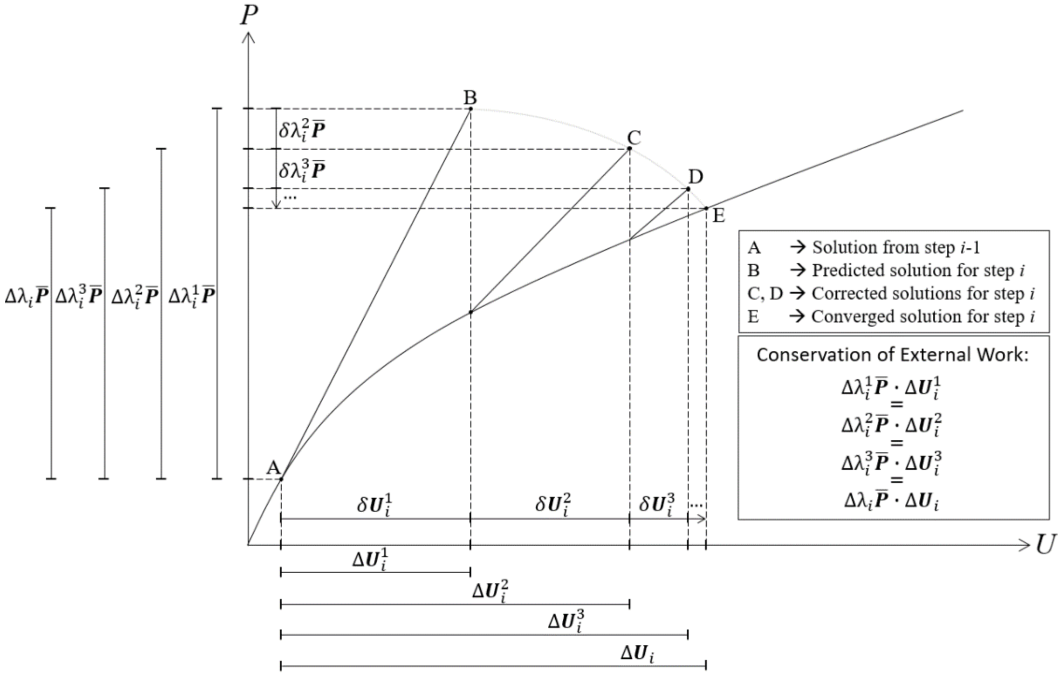

P
B
C
D
E
A
$\delta\lambda_i^2\bar{P}$
$\delta\lambda_i^3\bar{P}$
...
$\Delta\lambda_i\bar{P}$
$\Delta\lambda_i^3\bar{P}$
$\Delta\lambda_i^2\bar{P}$
$\Delta\lambda_i^1\bar{P}$
δU_i^1
δU_i^2
δU_i^3
U
ΔU_i^1
ΔU_i^2
ΔU_i^3
ΔU_i
A → Solution from step i-1
B → Predicted solution for step i
C, D → Corrected solutions for step i
E → Converged solution for step i
Conservation of External Work:
$\Delta\lambda_i^1\bar{P} \cdot \Delta U_i^1$
=
$\Delta\lambda_i^2\bar{P} \cdot \Delta U_i^2$
=
$\Delta\lambda_i^3\bar{P} \cdot \Delta U_i^3$
=
$\Delta\lambda_i\bar{P} \cdot \Delta U_i$

Figure 3.11 – Schematic of the Work Control Method

As the WCM corrects the value of the load ratio in each iteration, it is more suitable for capturing snap-through behavior than the LCM. However, as mentioned in Section 3.4.1.1.2, for structures with a small number of degrees-of-freedom and which the major forcing directions present a snap-back behavior, the scalar product in the denominator of Eq. (3.79) tends to zero near a displacement limit point (Yang & Shieh, 1990). Thus, this method has limited success for capturing snap-back behavior.

3.5.2.3. Constant Arc-Length Control Methods

In these methods, an increment of arc-length, Δ*L*, defined by the norm of the displacements and load increments, is kept constant throughout the iteration process. This means that the arc-length defined by the increments of the current iteration must have the same value of the arc-length defined by the increments of the previous iteration. Therefore, for *j* > 1 we have:

$$\Delta L_i^{j} = \Delta L_i^{j-1} \tag{3.80}$$

The expressions of the increment of arc-length for the previous and current iterations are respectively:

$$\Delta L_i^{j-1} = \left(\Delta \boldsymbol{U}_i^{j-1} \cdot \Delta \boldsymbol{U}_i^{j-1} + \beta \left(\Delta \lambda_i^{j-1} \right)^2 \bar{\boldsymbol{P}} \cdot \bar{\boldsymbol{P}} \right)^{1/2} \tag{3.81}$$

$$\Delta L_i^{j} = \left(\left(\Delta \boldsymbol{U}_i^{j-1} + \delta \boldsymbol{U}_i^{j} \right) \cdot \left(\Delta \boldsymbol{U}_i^{j-1} + \delta \boldsymbol{U}_i^{j} \right) + \beta \left(\Delta \lambda_i^{j-1} + \delta \lambda_i^{j} \right)^2 \bar{\boldsymbol{P}} \cdot \bar{\boldsymbol{P}} \right)^{1/2} \tag{3.82}$$

where β is the previously presented parameter that defines different versions of the arc-length measurement to restrict the iterations: cylindrical arc-length (β = 0), spherical arc-length (β = 1), and elliptical arc-length (0 < β < 1). The interpretation of these versions are depicted in Fig. 3.12 for a system of two degrees-of-freedom. In that figure, A is the starting point of the current incremental step and B is the equilibrium solution reached after the iterative cycle. In this work, only the cylindrical and the spherical versions are implemented, according to Crisfield (1981, 1991).

As commented in Section 3.4.1.1.3, the cylindrical and spherical versions of the Constant Arc-Length Control Method (abbreviated by ALCM_CYL and ALCM_SPH) use their corresponding predictive strategies (Cylindrical Arc-Length Increment and Spherical Arc-Length Increment) to calculate the increment that defines the constraint hyper-surface for iterating. The reason is to avoid numerical problems due to the possible inconsistency between increment sizes when using other techniques (Santana, 2015).

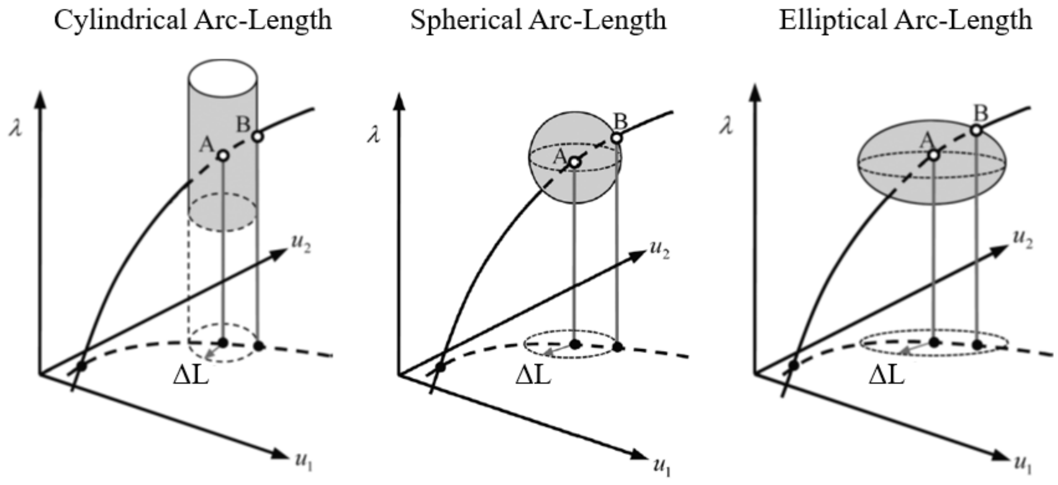


Figure 3.12 – Variations of the arc-length measurements for controlling iterations (Leon *et al.*, 2011)

Substituting Eq. (3.81) and Eq. (3.82) into Eq. (3.80), using Eq. (3.27) to express the increment of displacements of the current iteration in terms of the tangent and residual components, and making the necessary algebraic manipulations, we arrive at the following quadratic equation in terms of the unknown iterative increment of load ratio:

$$a\left(\delta\lambda_i^j\right)^2 + 2b\left(\delta\lambda_i^j\right) + c = 0 \qquad (3.83)$$

where:

$$\begin{cases} a = \delta\bar{\boldsymbol{U}}_i^j \cdot \delta\bar{\boldsymbol{U}}_i^j + \beta\bar{\boldsymbol{P}} \cdot \bar{\boldsymbol{P}} \\ b = \delta\bar{\boldsymbol{U}}_i^j \cdot \left(\delta\bar{\bar{\boldsymbol{U}}}_i^j + \Delta\boldsymbol{U}_i^{j-1}\right) + \Delta\lambda_i^{j-1}\beta\bar{\boldsymbol{P}} \cdot \bar{\boldsymbol{P}} \\ c = \delta\bar{\bar{\boldsymbol{U}}}_i^j \cdot \left(\delta\bar{\bar{\boldsymbol{U}}}_i^j + 2\Delta\boldsymbol{U}_i^{j-1}\right) \end{cases} \qquad (3.84)$$

The quadratic equation has two solutions for the increment of load ratio, $(\delta\lambda_i^j)_1$ and $(\delta\lambda_i^j)_2$, given by Eq. (3.85). Each solution corresponds to a point on the constraint hyper-surface that satisfies the condition of keeping the arc-length constant.

$$\left(\delta\lambda_i^j\right)_{1,2} = -\frac{b}{a} \pm \sqrt{\left(\frac{b}{a}\right)^2 - \frac{c}{a}} \qquad (3.85)$$

Selecting the correct solution of the quadratic equation is important to avoid regression on the equilibrium path. If the correct solution is selected in all iterations, the algorithm advances in the equilibrium path. Otherwise, it goes back, as indicated in Fig. 3.13.

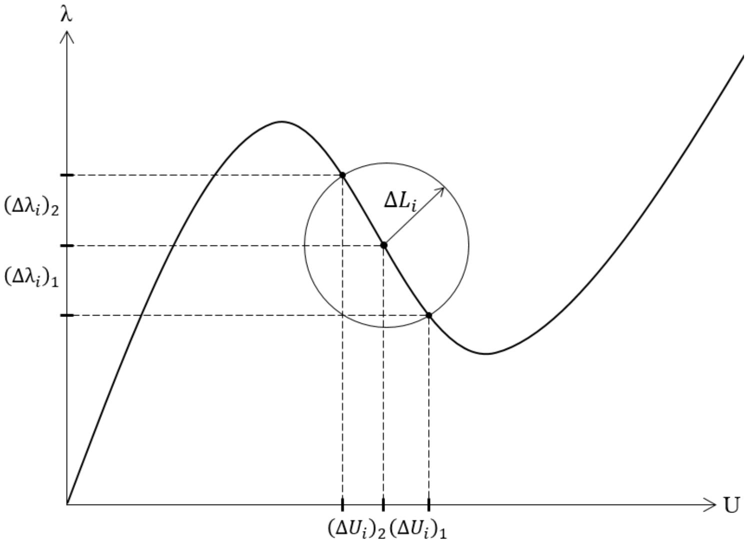


Figure 3.13 – Solutions of the quadratic equation of Constant Arc-Length Control Methods

The correct value of the load ratio increment is the one that provides a solution for the increment of displacements that is closer to the previous solution. Mathematically, it means that the scalar product between the vectors of displacement increments of the previous and current iterations assumes a greater value when the correct solution of the quadratic equation is selected. That is, considering the correct solution to be $(\delta\lambda_i^j)_1$:

$$\Delta \boldsymbol{U}_i^{j-1} \cdot \left(\Delta \boldsymbol{U}_i^j\right)_{\delta\lambda_1} > \Delta \boldsymbol{U}_i^{j-1} \cdot \left(\Delta \boldsymbol{U}_i^j\right)_{\delta\lambda_2} \tag{3.86}$$

$$\begin{gathered} \Delta \boldsymbol{U}_i^{j-1} \cdot \left(\Delta \boldsymbol{U}_i^{j-1} + \left(\delta\lambda_i^j\right)_1 \delta\bar{\boldsymbol{U}}_i^j + \delta\bar{\bar{\boldsymbol{U}}}_i^j\right) \\ > \\ \Delta \boldsymbol{U}_i^{j-1} \cdot \left(\Delta \boldsymbol{U}_i^{j-1} + \left(\delta\lambda_i^j\right)_2 \delta\bar{\boldsymbol{U}}_i^j + \delta\bar{\bar{\boldsymbol{U}}}_i^j\right) \end{gathered} \tag{3.87}$$

$$\left(\left(\delta\lambda_i^j\right)_1 - \left(\delta\lambda_i^j\right)_2\right) \Delta \boldsymbol{U}_i^{j-1} \cdot \delta\bar{\boldsymbol{U}}_i^j > 0 \tag{3.88}$$

If the scalar product of the displacement vectors in Eq. (3.88) is positive, the correct solution must be the greater value provided by Eq. (3.85). Otherwise, the correct solution is the smaller value of that expression. Therefore, it can be concluded that the correct solution of the quadratic equation depends on the sign of the scalar product between the two vectors of Eq. (3.88), and the constraint equation for obtaining the iterative increment of load ratio is:

$$\delta\lambda_i^j = -\frac{b}{a} + \mathrm{sign}\left(\Delta\boldsymbol{U}_i^{j-1} \cdot \delta\bar{\boldsymbol{U}}_i^j\right)\sqrt{\left(\frac{b}{a}\right)^2 - \frac{c}{a}} \tag{3.89}$$

This expression, for both the cylindrical and spherical versions, may result in a complex value if the incremental steps are not sufficiently small (Krenk, 1995), or near multiple equilibrium paths (Meek & Tan, 1984). When it happens, the program developed in this work interrupts the analysis and returns a warning.

3.5.2.4. Linear Arc-Length Control Methods

These types of arc-length control method, also known as orthogonal arc-length methods, consist of restricting the iterative increments to a hyper-plane that is orthogonal to the increments of a previous solution. The selection of this previous solution to set the orthogonality condition is what distinguishes the types of linearized arc-length methods.

In a version presented by Riks (1972) and Riks (1979), known as Fixed Normal Plane (ALCM_FNP), the vectors of the iterative solution ($\delta\boldsymbol{U}_i^j$ and $\delta\lambda_i^j\bar{\boldsymbol{P}}$) are orthogonal to the vectors of the predicted solution ($\delta\boldsymbol{U}_i^1$ and $\delta\lambda_i^1\bar{\boldsymbol{P}}$). The application of this orthogonality condition is given in Eq. (3.90). Then, Eq. (3.27) is used to decompose the iterative increment of displacements into a tangent and a residual component, in Eq. (3.91). Finally, the constraint equation for calculating the iterative increment of load ratio is obtained in Eq. (3.92), for $j > 1$.

$$\delta\boldsymbol{U}_i^1 \cdot \delta\boldsymbol{U}_i^j + \left(\delta\lambda_i^1\delta\lambda_i^j\right)\bar{\boldsymbol{P}} \cdot \bar{\boldsymbol{P}} = 0 \tag{3.90}$$

$$\delta\boldsymbol{U}_i^1 \cdot \left(\delta\lambda_i^j\delta\bar{\boldsymbol{U}}_i^j + \delta\bar{\bar{\boldsymbol{U}}}_i^j\right) + \left(\delta\lambda_i^1\delta\lambda_i^j\right)\bar{\boldsymbol{P}} \cdot \bar{\boldsymbol{P}} = 0 \tag{3.91}$$

$$\delta\lambda_i^j = -\frac{\delta\boldsymbol{U}_i^1 \cdot \delta\bar{\bar{\boldsymbol{U}}}_i^j}{\delta\boldsymbol{U}_i^1 \cdot \delta\bar{\boldsymbol{U}}_i^j + \delta\lambda_i^1\bar{\boldsymbol{P}} \cdot \bar{\boldsymbol{P}}} \tag{3.92}$$

Alternatively, a version presented by Ramm (1981) and Ramm (1982), known as Updated Normal Plane (ALCM_UNP), uses the vectors of the incremental solution of the previous iteration ($\Delta\boldsymbol{U}_i^{j-1}$ and $\Delta\lambda_i^{j-1}\bar{\boldsymbol{P}}$) to set the orthogonality condition. The constraint equation for this version is similarly obtained as follows:

$$\Delta \boldsymbol{U}_i^{j-1} \cdot \delta \boldsymbol{U}_i^j + \left(\Delta\lambda_i^{j-1} \delta\lambda_i^j \right) \bar{\boldsymbol{P}} \cdot \bar{\boldsymbol{P}} = 0 \tag{3.93}$$

$$\Delta \boldsymbol{U}_i^{j-1} \cdot \left(\delta\lambda_i^j \delta\bar{\boldsymbol{U}}_i^j + \delta\bar{\bar{\boldsymbol{U}}}_i^j \right) + \left(\Delta\lambda_i^{j-1} \delta\lambda_i^j \right) \bar{\boldsymbol{P}} \cdot \bar{\boldsymbol{P}} = 0 \tag{3.94}$$

$$\delta\lambda_i^j = - \frac{\Delta \boldsymbol{U}_i^{j-1} \cdot \delta\bar{\bar{\boldsymbol{U}}}_i^j}{\Delta \boldsymbol{U}_i^{j-1} \cdot \delta\bar{\boldsymbol{U}}_i^j + \Delta\lambda_i^{j-1} \bar{\boldsymbol{P}} \cdot \bar{\boldsymbol{P}}} \tag{3.95}$$

These methods are depicted in Fig. 3.14. Points B_0, B_1, and B_2 correspond, respectively, to the equilibrium solution obtained with the Constant Arc-Length Control Method (cylindrical or spherical version), Updated Normal Plane, and Fixed Normal Plane versions of the Linear Arc-Length Control Method.

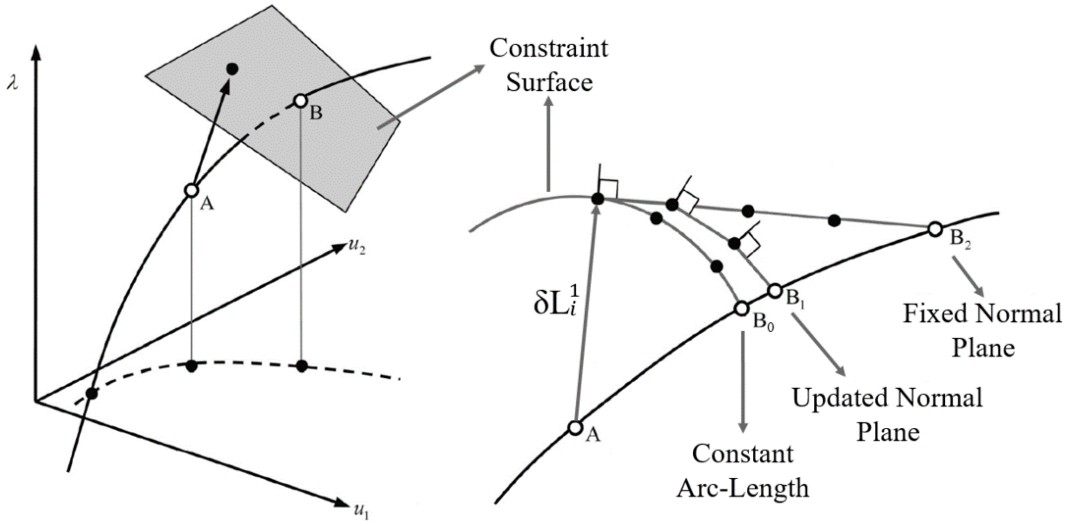


Figure 3.14 – Schematic of the Linear Arc-Length Control Methods (adapted from Leon *et al.*, 2011)

Differently from the Constant Arc-Length Control Methods, in which the constraint equation is nonlinear, the result of the orthogonality conditions is a simpler linear equation, so less effort is spent evaluating the expression.

In the developed tool, the adopted strategy for the predicted increment, when these iterative techniques are selected, is the cylindrical version of the Arc-Length Increment (Section 3.4.1.1.3). Notice, however, that the iterative cycle do not maintain the predicted increment of arc-length constant throughout the entire step.

3.5.2.5. Minimum Norm Control Method

The Minimum Norm Control Method (MNCM), presented by Chan (1988), tries to find the iterative increment of load ratio that results in the minimum increment of displacements. This is equivalent to set the norm of the vector of iterative increment of displacements to a minimum value. Mathematically, this condition means that the derivative of the norm of the displacement increments with respect to the increment of load ratio is zero. For *j* > 1:

$$\frac{\partial\left(\delta\boldsymbol{U}_i^j\cdot\delta\boldsymbol{U}_i^j\right)}{\partial\delta\lambda_i^j}=0 \tag{3.96}$$

Using Eq. (3.27) to express the increment of displacements in terms of the tangent and residual components, expanding the expression, and performing the derivatives, we arrive at the constraint equation to satisfy the proposed condition:

$$\frac{\partial\left(\left(\delta\lambda_i^j\delta\bar{\boldsymbol{U}}_i^j+\delta\bar{\bar{\boldsymbol{U}}}_i^j\right)\cdot\left(\delta\lambda_i^j\delta\bar{\boldsymbol{U}}_i^j+\delta\bar{\bar{\boldsymbol{U}}}_i^j\right)\right)}{\partial\delta\lambda_i^j}=0 \tag{3.97}$$

$$\frac{\partial\left(\left(\delta\lambda_i^j\right)^2\delta\bar{\boldsymbol{U}}_i^j\cdot\delta\bar{\boldsymbol{U}}_i^j+2\delta\lambda_i^j\delta\bar{\boldsymbol{U}}_i^j\cdot\delta\bar{\bar{\boldsymbol{U}}}_i^j+\delta\bar{\bar{\boldsymbol{U}}}_i^j\cdot\delta\bar{\bar{\boldsymbol{U}}}_i^j\right)}{\partial\delta\lambda_i^j}=0 \tag{3.98}$$

$$\delta\lambda_i^j=-\frac{\delta\bar{\bar{\boldsymbol{U}}}_i^j\cdot\delta\bar{\boldsymbol{U}}_i^j}{\delta\bar{\boldsymbol{U}}_i^j\cdot\delta\bar{\boldsymbol{U}}_i^j} \tag{3.99}$$

This iterative strategy is not bound to any strategy to calculate the predicted increment, so the cylindrical version of the Arc-Length Increment (Section 3.4.1.1.3) was selected for this purpose, due to its efficiency.

3.5.2.6. Orthogonal Residue Control Method

This strategy was proposed by Krenk (1995) and Krenk & Hededal (1995). To summary the idea, a vector of average residual forces, $\boldsymbol{R}_i^a$, is introduced. This average residue is defined as the difference between the external forces in the current iteration and the internal forces in the previous iteration. For *j* > 1, this is expressed as:

$$\boldsymbol{R}_i^a=\left(\lambda_{i-1}+\Delta\lambda_i^{j-1}+\delta\lambda_i^j\right)\bar{\boldsymbol{P}}-\boldsymbol{F}\left(\boldsymbol{U}_{i-1}+\Delta\boldsymbol{U}_i^{j-1}\right)=\boldsymbol{R}_i^{j-1}+\delta\lambda_i^j\bar{\boldsymbol{P}} \tag{3.100}$$

The constraint condition of this iterative technique is to take the average vector of residual forces to be orthogonal to the incremental vector of displacements of the previous iteration. This condition is applied by setting the scalar product between these two vectors to zero:

$$\boldsymbol{R}_i^a \cdot \Delta \boldsymbol{U}_i^{j-1} = 0 \tag{3.101}$$

which leads to the following expression for the constraint equation, for *j* > 1:

$$\delta\lambda_i^j = -\frac{\Delta \boldsymbol{U}_i^{j-1} \cdot \boldsymbol{R}_i^{j-1}}{\Delta \boldsymbol{U}_i^{j-1} \cdot \bar{\boldsymbol{P}}} \tag{3.102}$$

As with the MNCM, this iterative strategy is not bound to any strategy to calculate the predicted increment and it is not intended to keep the predicted increment of any control parameter. The cylindrical version of the Arc-Length Increment was also adopted due to its efficiency.

3.5.2.7. Generalized Displacement Control Method

The Generalized Displacement Control Method (GDCM) consists of maintaining the predicted increment of generalized displacement, applied with the technique described in Section 3.4.1.2.2, constant throughout the entire step. The expression for the increment of generalized displacement was given in Eq. (3.63), and it is reproduced here in terms of the tangent and residual increments of displacements, for *j* > 1:

$$\delta D_i^j = \left(\delta\lambda_i^j\right)^2 \delta\bar{\boldsymbol{U}}_{i-1}^j \cdot \delta\bar{\boldsymbol{U}}_i^1 + \delta\lambda_i^j \delta\bar{\boldsymbol{U}}_{i-1}^1 \cdot \delta\bar{\bar{\boldsymbol{U}}}_i^j \tag{3.103}$$

To force the increment of generalized displacement to be null in each iteration, this expression must be zero. Therefore, the following constraint equation is obtained:

$$\delta\lambda_i^j = -\frac{\delta\bar{\boldsymbol{U}}_{i-1}^1 \cdot \delta\bar{\bar{\boldsymbol{U}}}_i^j}{\delta\bar{\boldsymbol{U}}_{i-1}^1 \cdot \delta\bar{\boldsymbol{U}}_i^j} \tag{3.104}$$

The GDCM has little physical meaning. However, Cardoso & Fonseca (2007) show that it can be seen as a linear arc-length method.

3.6. Summary of Solution Methods

In the previous sections, the procedures for solving the linearized system of equilibrium equations, by means of incremental single-step or incremental-iterative numerical methods, were described. A flow-chart of the general solution algorithm is presented in the diagram of Fig. 3.15.

It is important to mention that this algorithm is general not only for the geometrically nonlinear formulations, but also for any nonlinear structural problem, that includes other sources of nonlinearity. The differences lie in the processes of evaluating the tangent stiffness matrix and calculating the internal nodal forces, which are indicated by the bold boxes in the diagram. In the developed tool, these steps are distinguished between UL and CR formulations. The computational implementation of these processes is described in the next chapter.

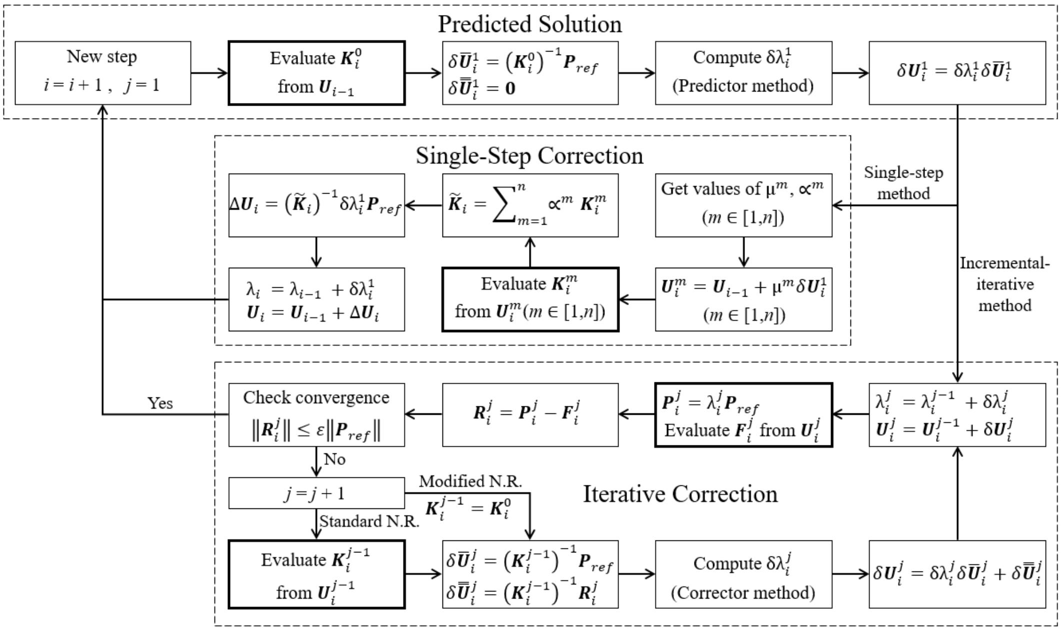

Predicted Solution
New step
$i = i + 1$, $j = 1$
Evaluate $\boldsymbol{K}_i^0$ from $\boldsymbol{U}_{i-1}$
$\delta\bar{\boldsymbol{U}}_i^1 = (\boldsymbol{K}_i^0)^{-1}\boldsymbol{P}_{ref}$
$\delta\bar{\bar{\boldsymbol{U}}}_i^1 = \boldsymbol{0}$
Compute $\delta\lambda_i^1$
(Predictor method)
$\delta\boldsymbol{U}_i^1 = \delta\lambda_i^1\delta\bar{\boldsymbol{U}}_i^1$
Single-Step Correction
Single-step method
$\Delta\boldsymbol{U}_i = (\tilde{\boldsymbol{K}}_i)^{-1}\delta\lambda_i^1\boldsymbol{P}_{ref}$
$\tilde{\boldsymbol{K}}_i = \sum_{m=1}^{n} \propto^m \boldsymbol{K}_i^m$
Get values of μ^m, $\propto^m$
($m \in [1,n]$)
$\lambda_i = \lambda_{i-1} + \delta\lambda_i^1$
$\boldsymbol{U}_i = \boldsymbol{U}_{i-1} + \Delta\boldsymbol{U}_i$
Evaluate $\boldsymbol{K}_i^m$ from $\boldsymbol{U}_i^m$($m \in [1,n]$)
$\boldsymbol{U}_i^m = \boldsymbol{U}_{i-1} + \mu^m\delta\boldsymbol{U}_i^1$
($m \in [1,n]$)
Incremental-iterative method
Yes
Check convergence
$\|\boldsymbol{R}_i^j\| \leq \varepsilon\|\boldsymbol{P}_{ref}\|$
$\boldsymbol{R}_i^j = \boldsymbol{P}_i^j - \boldsymbol{F}_i^j$
$\boldsymbol{P}_i^j = \lambda_i^j\boldsymbol{P}_{ref}$
Evaluate $\boldsymbol{F}_i^j$ from $\boldsymbol{U}_i^j$
$\lambda_i^j = \lambda_i^{j-1} + \delta\lambda_i^j$
$\boldsymbol{U}_i^j = \boldsymbol{U}_i^{j-1} + \delta\boldsymbol{U}_i^j$
No
$j = j + 1$
Modified N.R.
$\boldsymbol{K}_i^{j-1} = \boldsymbol{K}_i^0$
Standard N.R.
Iterative Correction
Evaluate $\boldsymbol{K}_i^{j-1}$ from $\boldsymbol{U}_i^{j-1}$
$\delta\bar{\boldsymbol{U}}_i^j = (\boldsymbol{K}_i^{j-1})^{-1}\boldsymbol{P}_{ref}$
$\delta\bar{\bar{\boldsymbol{U}}}_i^j = (\boldsymbol{K}_i^{j-1})^{-1}\boldsymbol{R}_i^j$
Compute $\delta\lambda_i^j$
(Corrector method)
$\delta\boldsymbol{U}_i^j = \delta\lambda_i^j\delta\bar{\boldsymbol{U}}_i^j + \delta\bar{\bar{\boldsymbol{U}}}_i^j$

Figure 3.15 – General solution algorithm of the incremental system of equilibrium equations

Some of the steps of the general algorithm depend on the selected methods for calculating the predicted and the corrected solutions. The formulas to obtain the predicted increment of load ratio for the techniques implemented in this work are summarized in Table 3.2. Both the expression based on the previous predicted solution and the expression based on the total increments of the previous step are shown. In the sequence, Table 3.3 brings the equations for the corrective techniques to calculate the iterative increment of load ratio. The adopted predictive techniques are also indicated.

Table 3.2 – Summary of equations for calculating the predicted increment of load ratio

Predictive Technique	**Load Ratio Increment** (i > 1)
Load Increment (LI)	$\delta\lambda_i^1 = s\cdot J\left\lvert\delta\lambda_{i-1}^1\right\rvert$
	$\delta\lambda_i^1 = s\cdot J\left\lvert\Delta\lambda_{i-1}\right\rvert$
External Work Increment (EWI)	$\delta\lambda_i^1 = s\cdot J\left\lvert\delta\lambda_{i-1}^1\right\rvert\sqrt{\left\lvert\frac{\bar{\boldsymbol{P}}\cdot\delta\bar{\boldsymbol{U}}_{i-1}^1}{\bar{\boldsymbol{P}}\cdot\delta\bar{\boldsymbol{U}}_i^1}\right\rvert}$
	$\delta\lambda_i^1 = s\cdot J\sqrt{\left\lvert\frac{\Delta\lambda_{i-1}\bar{\boldsymbol{P}}\cdot\Delta\boldsymbol{U}_{i-1}}{\bar{\boldsymbol{P}}\cdot\delta\bar{\boldsymbol{U}}_i^1}\right\rvert}$
Cylindrical Arc-Length Increment (CALI)	$\delta\lambda_i^1 = s\cdot J\left\lvert\delta\lambda_{i-1}^1\right\rvert\sqrt{\frac{\delta\bar{\boldsymbol{U}}_{i-1}^1\cdot\delta\bar{\boldsymbol{U}}_{i-1}^1}{\delta\bar{\boldsymbol{U}}_i^1\cdot\delta\bar{\boldsymbol{U}}_i^1}}$
	$\delta\lambda_i^1 = s\cdot J\sqrt{\frac{\Delta\boldsymbol{U}_{i-1}\cdot\Delta\boldsymbol{U}_{i-1}}{\delta\bar{\boldsymbol{U}}_i^1\cdot\delta\bar{\boldsymbol{U}}_i^1}}$
Spherical Arc-Length Increment (SALI)	$\delta\lambda_i^1 = s\cdot J\left\lvert\delta\lambda_{i-1}^1\right\rvert\sqrt{\frac{\delta\bar{\boldsymbol{U}}_{i-1}^1\cdot\delta\bar{\boldsymbol{U}}_{i-1}^1+\bar{\boldsymbol{P}}\cdot\bar{\boldsymbol{P}}}{\delta\bar{\boldsymbol{U}}_i^1\cdot\delta\bar{\boldsymbol{U}}_i^1+\bar{\boldsymbol{P}}\cdot\bar{\boldsymbol{P}}}}$
	$\delta\lambda_i^1 = s\cdot J\sqrt{\frac{\Delta\boldsymbol{U}_{i-1}\cdot\Delta\boldsymbol{U}_{i-1}+\left(\Delta\lambda_{i-1}\right)^2\bar{\boldsymbol{P}}\cdot\bar{\boldsymbol{P}}}{\delta\bar{\boldsymbol{U}}_i^1\cdot\delta\bar{\boldsymbol{U}}_i^1+\bar{\boldsymbol{P}}\cdot\bar{\boldsymbol{P}}}}$
Generalized Displacement Increment (GDI)	$\delta\lambda_i^1 = s\cdot\delta\lambda_1^1\sqrt{\left\lvert\mathrm{GSP}\right\rvert}$
Size adjustment factor: $J=\left(\frac{N_i}{N_{i-1}}\right)^{1/2}$	
Generalized Displacement Parameter: $\mathrm{GSP}=\frac{\delta\bar{\boldsymbol{U}}_1^1\cdot\delta\bar{\boldsymbol{U}}_1^1}{\delta\bar{\boldsymbol{U}}_{i-1}^1\cdot\delta\bar{\boldsymbol{U}}_i^1}$	
Increment sign: $\begin{cases} i=1 & \rightarrow \quad s=\oplus \\ \mathrm{GSP}<0 & \rightarrow \quad s=-s \end{cases}$	

Table 3.3 – Summary of equations for calculating the iterative increment of load ratio

Solution Method	**Predictive Technique**	**Load Ratio Correction** ($j > 1$)
Load Control	LI	$\delta\lambda_i^j = 0$
Work Control	EWI	$\delta\lambda_i^j = -\dfrac{\bar{\boldsymbol{P}} \cdot \delta\bar{\bar{\boldsymbol{U}}}_i^j}{\bar{\boldsymbol{P}} \cdot \delta\bar{\boldsymbol{U}}_i^j}$
Constant Arc-Length*	CALI/ SALI	$\delta\lambda_i^j = -\dfrac{b}{a} + s\sqrt{\left(\dfrac{b}{a}\right)^2 - \dfrac{c}{a}}$
Linear Arc-Length (Fixed Normal Plane)	CALI	$\delta\lambda_i^j = -\dfrac{\delta\boldsymbol{U}_i^1 \cdot \delta\bar{\bar{\boldsymbol{U}}}_i^j}{\delta\boldsymbol{U}_i^1 \cdot \delta\bar{\boldsymbol{U}}_i^j + \delta\lambda_i^1 \bar{\boldsymbol{P}} \cdot \bar{\boldsymbol{P}}}$
Linear Arc-Length (Updated Normal Plane)	CALI	$\delta\lambda_i^j = -\dfrac{\Delta\boldsymbol{U}_i^{j-1} \cdot \delta\bar{\bar{\boldsymbol{U}}}_i^j}{\Delta\boldsymbol{U}_i^{j-1} \cdot \delta\bar{\boldsymbol{U}}_i^j + \Delta\lambda_i^{j-1} \bar{\boldsymbol{P}} \cdot \bar{\boldsymbol{P}}}$
Minimum Norm	CALI	$\delta\lambda_i^j = -\dfrac{\delta\bar{\bar{\boldsymbol{U}}}_i^j \cdot \delta\bar{\boldsymbol{U}}_i^j}{\delta\bar{\boldsymbol{U}}_i^j \cdot \delta\bar{\boldsymbol{U}}_i^j}$
Orthogonal Residue	CALI	$\delta\lambda_i^j = -\dfrac{\Delta\boldsymbol{U}_i^{j-1} \cdot \boldsymbol{R}_i^{j-1}}{\Delta\boldsymbol{U}_i^{j-1} \cdot \bar{\boldsymbol{P}}}$
Generalized Displacement	GDI	$\delta\lambda_i^j = -\dfrac{\delta\bar{\boldsymbol{U}}_{i-1}^1 \cdot \delta\bar{\bar{\boldsymbol{U}}}_i^j}{\delta\bar{\boldsymbol{U}}_{i-1}^1 \cdot \delta\bar{\boldsymbol{U}}_i^j}$

* Parameters for Constant Arc-Length Methods:

$a = \delta\bar{\boldsymbol{U}}_i^j \cdot \delta\bar{\boldsymbol{U}}_i^j + \beta \bar{\boldsymbol{P}} \cdot \bar{\boldsymbol{P}}$

$b = \delta\bar{\boldsymbol{U}}_i^j \cdot \left(\delta\bar{\bar{\boldsymbol{U}}}_i^j + \Delta\boldsymbol{U}_i^{j-1}\right) + \Delta\lambda_i^{j-1} \beta \bar{\boldsymbol{P}} \cdot \bar{\boldsymbol{P}}$

$c = \delta\bar{\bar{\boldsymbol{U}}}_i^j \cdot \left(\delta\bar{\bar{\boldsymbol{U}}}_i^j + 2\Delta\boldsymbol{U}_i^{j-1}\right)$

$s = \text{sign}\left(\Delta\boldsymbol{U}_i^{j-1} \cdot \delta\bar{\boldsymbol{U}}_i^j\right)$

$$\begin{cases} \text{Cylindrical Arc-Length} \rightarrow \beta = 0 \\ \text{Spherical Arc-Length} \quad \rightarrow \beta = 1 \end{cases}$$

4. Developed Tool

4.1. Introduction

Based on the theory presented in previous chapters on the nonlinear formulation of equilibrium equations and the solution methods for dealing with the incremental system, this chapter describes the development and use of a graphical-interactive tool for performing geometrically nonlinear analysis of two-dimensional frame structural models. This application was incorporated as a new analysis module of the Ftool (Two-dimensional Frame Analysis Tool) program (Martha, 1999), which is a largely used software in the civil engineering community and has demonstrated to be a valuable program for teaching structural analysis over the last decades.

When creating a computer program to execute a task that would be manually impractical, a series of questions must be considered besides the theoretical understanding of the problem. These questions include concerns about memory storage and efficiency, a data structure to manage the information in the pre, post, and processing steps, a robust source code to avoid unexpected errors, and a good architecture and modularization of the code to make it easy to understand, maintain, and expand. In graphical-interactive programs, it is still necessary to design and create a user-friendly graphical interface, predict and control user actions, and deal with topics on computer graphics and computational geometry to provide the best experience for those who want to simulate and visualize the physical phenomenon studied.

All of that had to be contemplated during the development of the tool that motivated this work. However, not all of these issues are discussed in this document, in which the attention is turned to the implementation of the numerical methods to perform the geometrically nonlinear analysis of two-dimensional frame models, as well as the developed graphical user interface, considering its functionalities and characteristics.

The second section of this chapter is focused on the computer implementation of the solution algorithms. The goal is to present how the concepts covered in Chapter 3 were implemented to the developed tool by means of a step-by-step description of an auxiliary code that has didactics purposes only. The third section is intended to show how the developed tool works, from a user point of view, and its main features. The new analysis menus of the graphical user interface and the developed plotting environment are described.

4.2. Computer Implementation

The task of processing and computing the results of a mechanical analysis is done by a solver, where the algorithms of the solution methods are implemented. The analysis solver of the Ftool program is the FRAMOOP system, a simplified version of the FEMOOP (Finite Element Method Object Oriented Program) system (Martha & Parente 2002), modified to perform only linear-elastic analysis of framed structure models. This system is an external library, leaving the Ftool responsible for the pre and post-processing phases, while the FRAMOOP performs the structural analysis. This strategy helps future implementations and independent code maintenance of both programs.

The FRAMOOP system is written in the C programming language and adopts a programming philosophy similar to the Object Oriented Programming (OOP) paradigm (Cox & Novobilski, 1991). This programming paradigm is advantageous for structural analysis codes, providing modularization and reusability by means of classes, heritance, polymorphism, and encapsulation mechanisms (Rangel & Martha, 2019). Since the C language is not object-oriented, the simulation of the OOP paradigm in the FRAMOOP system is formally done by using pointers to functions as macros to generically call abstract methods of super classes.

Presenting the FRAMOOP code would not be didactic, as C is a low level programming language. For this purpose, an auxiliary program, called NLframe2D, was developed as a parallel implementation using the MATLAB script language. This program is a non-graphical application with the same general code structure of the nonlinear analysis module of the FRAMOOP system. The advantage is that MATLAB provides a programming language with a much simpler syntax, when compared to the C language, ideal to be presented for those who are interested in the computer implementation of an engineering problem.

This auxiliary program is used in the next sections only to present the implementation of the solution process of the nonlinear structural problem considered in this work. It is important to mention that this code is a very simplified version of what was developed in FRAMOOP. The intention here is to show the general structure of the nonlinear analysis algorithm. Therefore, many details have been omitted to emphasize what is pertinent to solving the problem. These details include, for example, checks of matrix singularity and real values, among other particularities of the implementation that are not relevant. The code also does not bother to show the strategy to perform the analysis in an interactive-adaptive fashion. In addition, efficiency and performance are not a concern, so code clarity can be prioritized. It should be noted the importance of a good documentation by means of comments that help the understanding of the code lines.

In addition, this auxiliary program is open-source, intended to receive the collaboration and suggestions from researchers interested in the subject. It is available for download in (the code may have been modified and updated):

https://www.mathworks.com/matlabcentral/fileexchange/73129-nlframe2d.

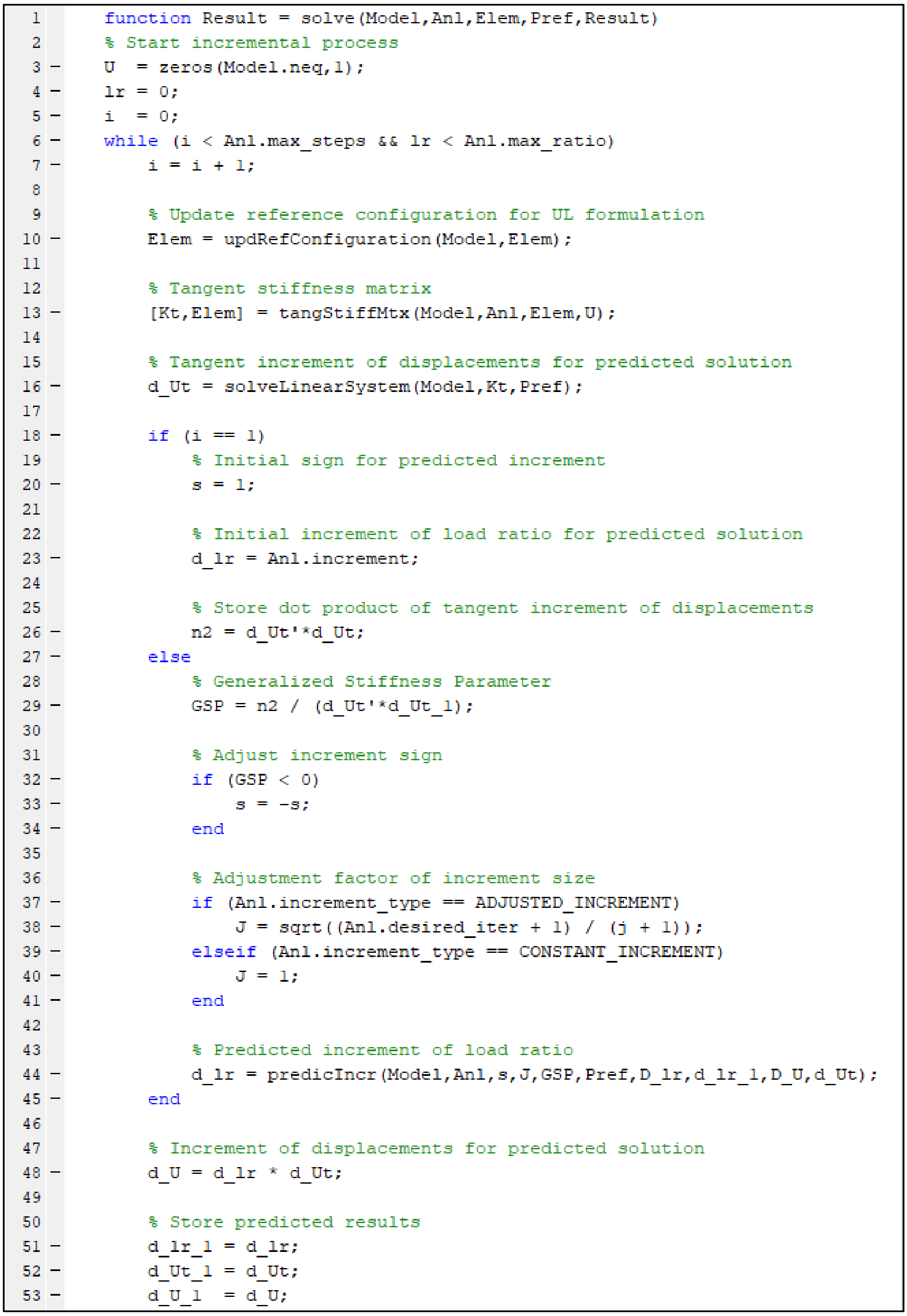
4.2.1. Main Algorithm

The solver of the NLframe2D program is all implemented in a single script file, named *solve.m*, that contains several functions to perform the incremental analysis. The main function of the program is responsible for implementing the code that execute the steps of the general algorithm for nonlinear problems, depicted in Fig. 3.15. Auxiliary functions for the particular nonlinear formulation, such as the evaluation of the tangent stiffness matrix and the vector of internal forces, are presented in the following sections.

The main function has the same name of the file. It receives, as input arguments, some data structures with model information and analysis options, which were input by user and have already been pre-processed. These pre analysis steps are the common tasks to any computer implementation for structural analysis using the FEM: reading model information, counting and numbering degrees-of-freedom, assembling vectors and matrix of global degrees-of-freedom numbers, etc. The input arguments of the main function are described in Table 4.1. The implementation of the predicted solution is given in the code snippet of Fig. 4.1.

Table 4.1 – Input arguments of the main solver function of the NLframe2D program

Variable	Description
`Model`	Data structure with fields for storing model information, such as total number of equations (`.neq`), number of equations of free degrees-of-freedom (`.neqf`), etc.
`Anl`	Data structure with fields for storing analysis options and parameters. The analysis options include type of formulation (`.formulation`), type of geometric stiffness matrix (`.geom_matrix`), solution algorithm (`.algorithm`), type of increment adjustment (`.increment_type`), and type of Newton-Raphson iteration scheme (`.iteration_type`). The analysis parameters include initial load ratio increment (`.increment`), maximum load ratio (`.max_ratio`), maximum number of steps (`.max_steps`), maximum number of iterations (`.max_iter`), desired number of iterations (`.desired_iter`), and tolerance for convergence (`.tol`).
`Elem`	Vector of data structures with fields for storing element information, such as end nodes (`.node1`, `.node2`), initial length (`.L`), initial angle with horizontal axis (`.angle`), gather vector with degrees-of-freedom numbers (`.gle`), etc.
`Pref`	Vector of reference nodal loads, taken as the total applied loads.
`Result`	Data structure to be filled with the results for each incremental step, such as the vector of all load ratios (`.lr`), and a matrix with all vectors of nodal displacements (`.U`). This structure is also an output of the function.



```
function Result = solve(Model,Anl,Elem,Pref,Result)
% Start incremental process
U  = zeros(Model.neq,1);
lr = 0;
i  = 0;
while (i < Anl.max_steps && lr < Anl.max_ratio)
    i = i + 1;
    
    % Update reference configuration for UL formulation
    Elem = updRefConfiguration(Model,Elem);
    
    % Tangent stiffness matrix
    [Kt,Elem] = tangStiffMtx(Model,Anl,Elem,U);
    
    % Tangent increment of displacements for predicted solution
    d_Ut = solveLinearSystem(Model,Kt,Pref);
    
    if (i == 1)
        % Initial sign for predicted increment
        s = 1;
        
        % Initial increment of load ratio for predicted solution
        d_lr = Anl.increment;
        
        % Store dot product of tangent increment of displacements
        n2 = d_Ut'*d_Ut;
    else
        % Generalized Stiffness Parameter
        GSP = n2 / (d_Ut'*d_Ut_1);
        
        % Adjust increment sign
        if (GSP < 0)
            s = -s;
        end
        
        % Adjustment factor of increment size
        if (Anl.increment_type == ADJUSTED_INCREMENT)
            J = sqrt((Anl.desired_iter + 1) / (j + 1));
        elseif (Anl.increment_type == CONSTANT_INCREMENT)
            J = 1;
        end
        
        % Predicted increment of load ratio
        d_lr = predicIncr(Model,Anl,s,J,GSP,Pref,D_lr,d_lr_1,D_U,d_Ut);
    end
    
    % Increment of displacements for predicted solution
    d_U = d_lr * d_Ut;
    
    % Store predicted results
    d_lr_1 = d_lr;
    d_Ut_1 = d_Ut;
    d_U_1  = d_U;
```

Figure 4.1 – Implementation of the predicted solution

As seen in Fig. 4.1, the main function starts by initializing the vector of nodal displacements, `U`, the value of load ratio, `lr`, and the counter of number of steps, `i`. The incremental process then begins with a loop that runs until the maximum number of steps or the maximum load ratio value is reached. The step counter is immediately incremented in the beginning of the incremental loop.

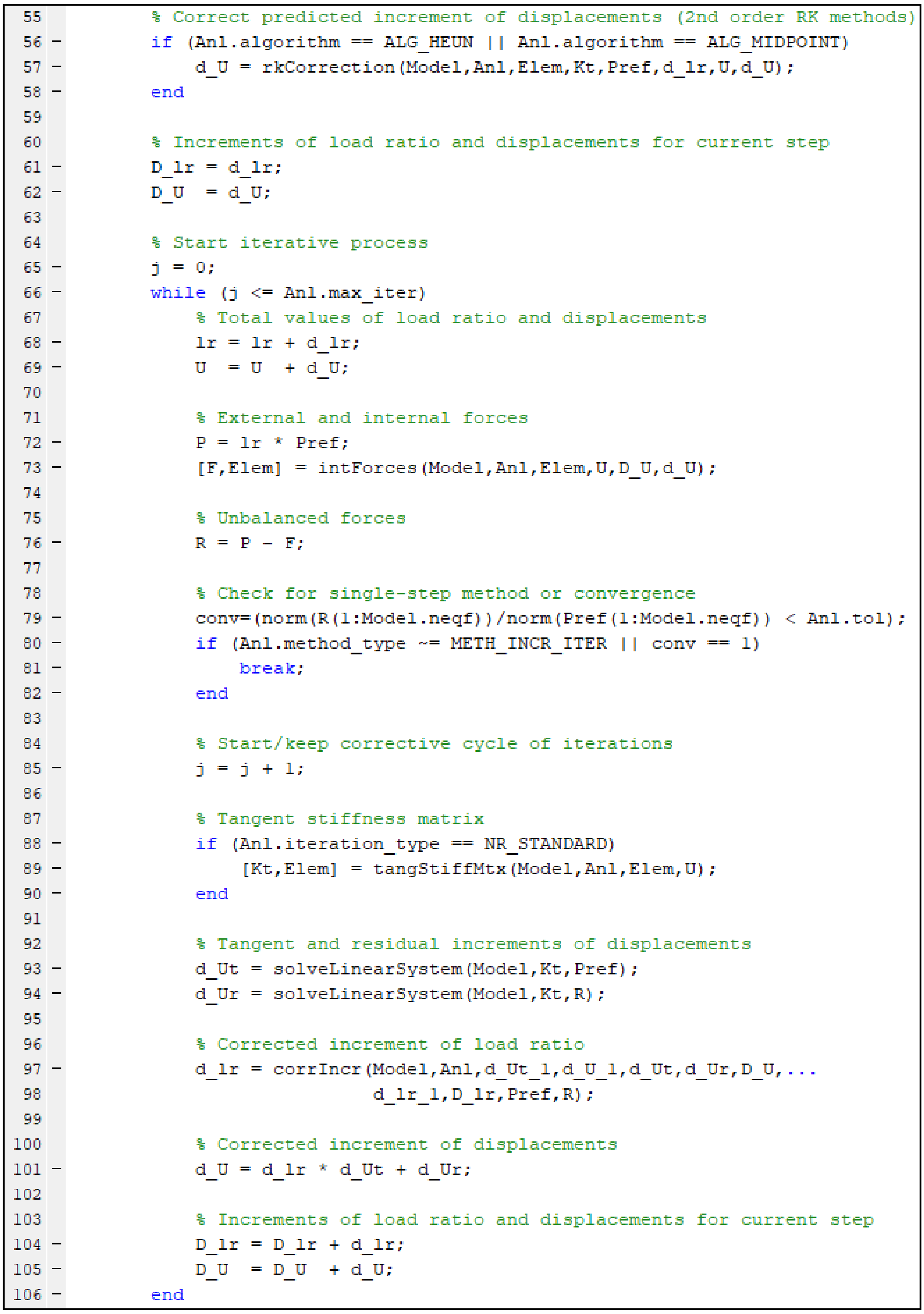
The *predictor* phase starts with the update of the reference configuration for the UL formulation (line 10). As explained in Chapter 2, this formulation is based on a kinematic description that takes the last obtained equilibrium configuration (step beginning) as the reference configuration to measure the increment of variables to reach the next equilibrium configuration. For this purpose, an auxiliary function is called to store the values of length, angle with global *X*-axis, and internal forces of each element, at the beginning of the step. These reference values are then used in the iterative cycle to compute increments of element internal forces using the UL formulation. This auxiliary function will not be shown because it can be easily implemented by storing the current values of the mentioned properties in appropriate fields of each element data structure.

The tangent stiffness matrix, `Kt`, for the predicted solution is evaluated (line 13) considering the current equilibrium configuration, i.e., element internal forces and nodal displacements obtained at the end of the previous step. Therefore, this function receives the vector of element data structures and the vector of nodal displacements, both updated with the results from the previous step, as input arguments. This function has a particular implementation for the UL and the CR formulations. Each implementation is described in Section 4.2.2. A check is recommended at this point to interrupt the analysis when the evaluated tangent matrix is singular. The tangent stiffness matrix is then used to calculate the tangent increment of displacements, `d_Ut`, by solving a linear system with the reference load vector (line 16). An auxiliary function for solving linear systems was implemented for this task, but it is not shown here.

The next step is to calculate the predicted increment of load ratio. This task is executed in different ways depending on whether the incremental step is the first or not. For the first step, the initial sign of the predicted increment, `s`, is set to positive (line 20). Then, the predicted increment of load ratio, `d_lr`, is taken as the value prescribed by the analyst (line 23). It is also necessary to store the squared value of the norm of the tangent increment of displacements in the first step, `n2` (line 26). This value is used in the expression to calculate the GSP in subsequent steps.

For the remaining incremental steps (i > 1), the GSP is calculated according to Eq. (3.67) in line 29. The increment sign is adjusted in line 32, according to the criterion described in Section 3.4.2, where it was explained that the sign of the predicted increment must be inverted every time the GSP value is negative. After that, the adjustment factor of the increment size, `J`, is obtained according to Eq. (3.32). The variable `j` in the expression of line 38 stores the number of iterations performed in the previous step, which must be increased by one to avoid division by zero, since the predicted solution corresponds to zero iterations. If the user selects the option to perform the analysis with constant increments, the value of the adjustment factor is set to unity. Finally, the predicted increment of load ratio is calculated in an auxiliary function (line 44), where the expressions of Table 3.2 are implemented. Again, there is no need to show this function here, as it has a simple implementation. To avoid exorbitant results, it is recommended to limit the value of the load ratio increment returned by this function to a reasonable value.

After obtaining the predicted increment of load ratio, the predicted increment of displacements, `d_U`, is calculated according to Eq. (3.27) in line 48 (using only the tangent increment of displacements). To finish the *predictor* phase, the predicted increments of load ratio, `d_lr_1`, tangent displacements, `d_Ut_1`, and full displacements, `d_U_1`, are stored to be used in the next step and in the iterative cycle. The *corrector* phase then starts, and its implementation is given in Fig. 4.2.



```
        % Correct predicted increment of displacements (2nd order RK methods)
        if (Anl.algorithm == ALG_HEUN || Anl.algorithm == ALG_MIDPOINT)
            d_U = rkCorrection(Model,Anl,Elem,Kt,Pref,d_lr,U,d_U);
        end

        % Increments of load ratio and displacements for current step
        D_lr = d_lr;
        D_U  = d_U;

        % Start iterative process
        j = 0;
        while (j <= Anl.max_iter)
            % Total values of load ratio and displacements
            lr = lr + d_lr;
            U  = U  + d_U;

            % External and internal forces
            P = lr * Pref;
            [F,Elem] = intForces(Model,Anl,Elem,U,D_U,d_U);

            % Unbalanced forces
            R = P - F;

            % Check for single-step method or convergence
            conv=(norm(R(1:Model.neqf))/norm(Pref(1:Model.neqf)) < Anl.tol);
            if (Anl.method_type ~= METH_INCR_ITER || conv == 1)
                break;
            end

            % Start/keep corrective cycle of iterations
            j = j + 1;

            % Tangent stiffness matrix
            if (Anl.iteration_type == NR_STANDARD)
                [Kt,Elem] = tangStiffMtx(Model,Anl,Elem,U);
            end

            % Tangent and residual increments of displacements
            d_Ut = solveLinearSystem(Model,Kt,Pref);
            d_Ur = solveLinearSystem(Model,Kt,R);

            % Corrected increment of load ratio
            d_lr = corrIncr(Model,Anl,d_Ut_1,d_U_1,d_Ut,d_Ur,D_U,...
                            d_lr_1,D_lr,Pref,R);

            % Corrected increment of displacements
            d_U = d_lr * d_Ut + d_Ur;

            % Increments of load ratio and displacements for current step
            D_lr = D_lr + d_lr;
            D_U  = D_U  + d_U;
        end
```

Figure 4.2 – Implementation of the corrected solution

The *corrector* phase starts with a check of the solution method. If a second-order single-step method is selected (Heun method or Midpoint method), an auxiliary function is called to return the corrected solution for the increment of displacements `d_U`. This function is described afterwards. Then, the increments of load ratio and displacements for the current step (`D_lr` and `D_U`) are set as the values obtained so far (lines 61 and 62). The counter for the number of iterations, `j`, is initialized as zero (line 65), since it is assumed that the first corrective iteration starts only after the convergence for the predicted solution has been checked. The iterative cycle then begins in a loop that runs until the maximum number of iterations is reached. If this condition occurs, it is because the algorithm failed to converge to an equilibrium configuration in the current incremental step.

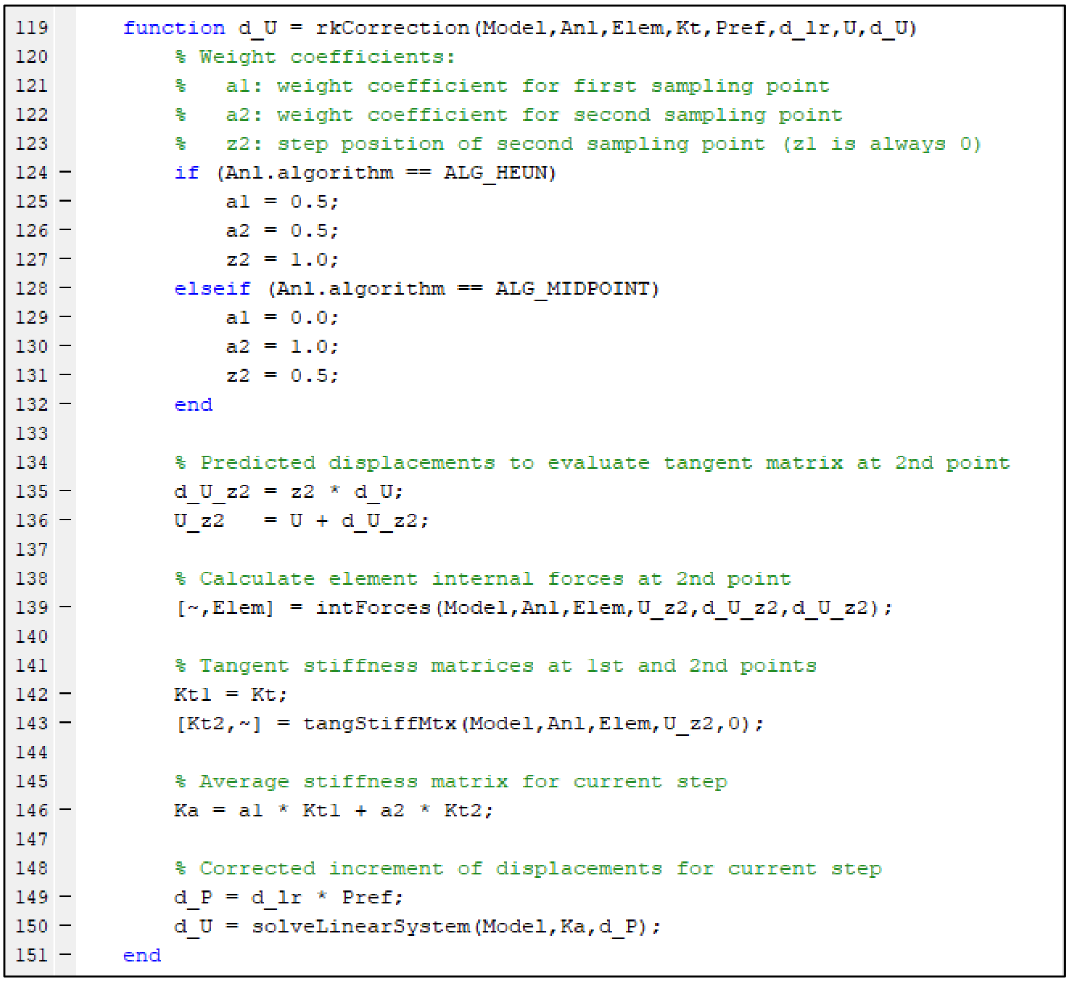
In the beginning of the iterative cycle, the total values of load ratio and nodal displacements are updated with the corrected increments obtained from the last iteration, or with the predicted increments if it has just entered the loop (lines 68 and 69). The vector of external nodal forces, `P`, are easily obtained from the total load ratio (line 72), and the vector of internal nodal forces, `F`, need to be evaluated for the current level of nodal displacements in an auxiliary function. Just like the function to evaluate the tangent stiffness matrix, this auxiliary function also has a particular implementation for each nonlinear formulation. Section 4.2.3 describes the implementation for the UL and CR formulations. The vector of residual forces, `R`, is then obtained by the difference between the vectors of external and internal forces (line 76).

The convergence for the current iteration is checked in accordance to the criterion based on the norm of the residual forces, given in Eq. (3.74). If this norm is sufficiently small, when compared to the norm of the reference load vector, the algorithm converged to a tolerable equilibrium configuration. Notice, in line 79, that only the components related to free degrees-of-freedom of the vectors of residual forces and reference loads are considered for taking the norms. If convergence is guaranteed or if a single-step solution method is selected, the algorithm exits the loop to proceed to the next incremental step. Otherwise, if convergence has not yet been reached for incremental-iterative methods, the algorithm proceeds to the next iterative correction and immediately increments the counter of iteration number (line 85).

The correction of load and displacement increments starts by checking the type of iteration scheme. If the standard version of Newton-Raphson iterations is selected, a new tangent stiffness matrix is evaluated with the updated configuration (line 89), i.e., with the element internal forces and nodal displacements obtained at the end of the previous iteration. Again, a check for matrix singularity is recommended after evaluating the new tangent stiffness matrix. Otherwise, if the modified version of Newton-Raphson iteration scheme is selected, the tangent stiffness matrix evaluated at the beginning of the incremental step for the predicted solution (line 13) continue being used.

The tangent and residual increments of the displacement corrections are calculated with linear systems using, respectively, the reference loads and the residual forces (lines 93 and 94). Since the same coefficient matrix is used in both systems, an efficient program should not solve two systems separately, but optimize the process to use the same decomposed coefficient matrix. The iterative correction of load ratio is return by a function (line 97) that implements the expressions for each iterative technique, presented in Table 3.3. This function is omitted here and, just as the predicted increment of load ratio, the returned result should be limited to a reasonable value. Furthermore, a check for a real number is recommended, since the expressions of the Constant Arc-Length strategies (cylindrical and spherical), given by Eq. (3.89), can assume complex values. The iterative correction of nodal displacements is finally obtained in accordance with Eq. (3.27), in line 101. Once the corrections of load and displacements are obtained, the incremental values of these quantities, for the current step, are updated (lines 104 and 105). Finally, the same process for updating total results and checking convergence, previously described, is repeated (lines 68 to 80).

When the algorithm exists the iterative loop, a few command lines are necessary to check if the reason for breaking was the convergence success or failure. In case of success, the algorithm stores the results of the new obtained equilibrium configuration. Otherwise, in case of convergence failure, the program exists the main function and displays a warning. There is no need to show these simple code lines. The basic results to be stored are the load ratio value and the vector of nodal displacements, in order to plot the curve of equilibrium path. However, several other interesting results can be stored, to provide a wider range of information to users.



```
119       function d_U = rkCorrection(Model,Anl,Elem,Kt,Pref,d_lr,U,d_U)
120           % Weight coefficients:
121           %   a1: weight coefficient for first sampling point
122           %   a2: weight coefficient for second sampling point
123           %   z2: step position of second sampling point (z1 is always 0)
124 -         if (Anl.algorithm == ALG_HEUN)
125 -             a1 = 0.5;
126 -             a2 = 0.5;
127 -             z2 = 1.0;
128 -         elseif (Anl.algorithm == ALG_MIDPOINT)
129 -             a1 = 0.0;
130 -             a2 = 1.0;
131 -             z2 = 0.5;
132 -         end
133
134           % Predicted displacements to evaluate tangent matrix at 2nd point
135 -         d_U_z2 = z2 * d_U;
136 -         U_z2   = U + d_U_z2;
137
138           % Calculate element internal forces at 2nd point
139 -         [~,Elem] = intForces(Model,Anl,Elem,U_z2,d_U_z2,d_U_z2);
140
141           % Tangent stiffness matrices at 1st and 2nd points
142 -         Kt1 = Kt;
143 -         [Kt2,~] = tangStiffMtx(Model,Anl,Elem,U_z2,0);
144
145           % Average stiffness matrix for current step
146 -         Ka = a1 * Kt1 + a2 * Kt2;
147
148           % Corrected increment of displacements for current step
149 -         d_P = d_lr * Pref;
150 -         d_U = solveLinearSystem(Model,Ka,d_P);
151 -     end
```

Figure 4.3 – Implementation of the single-step correction

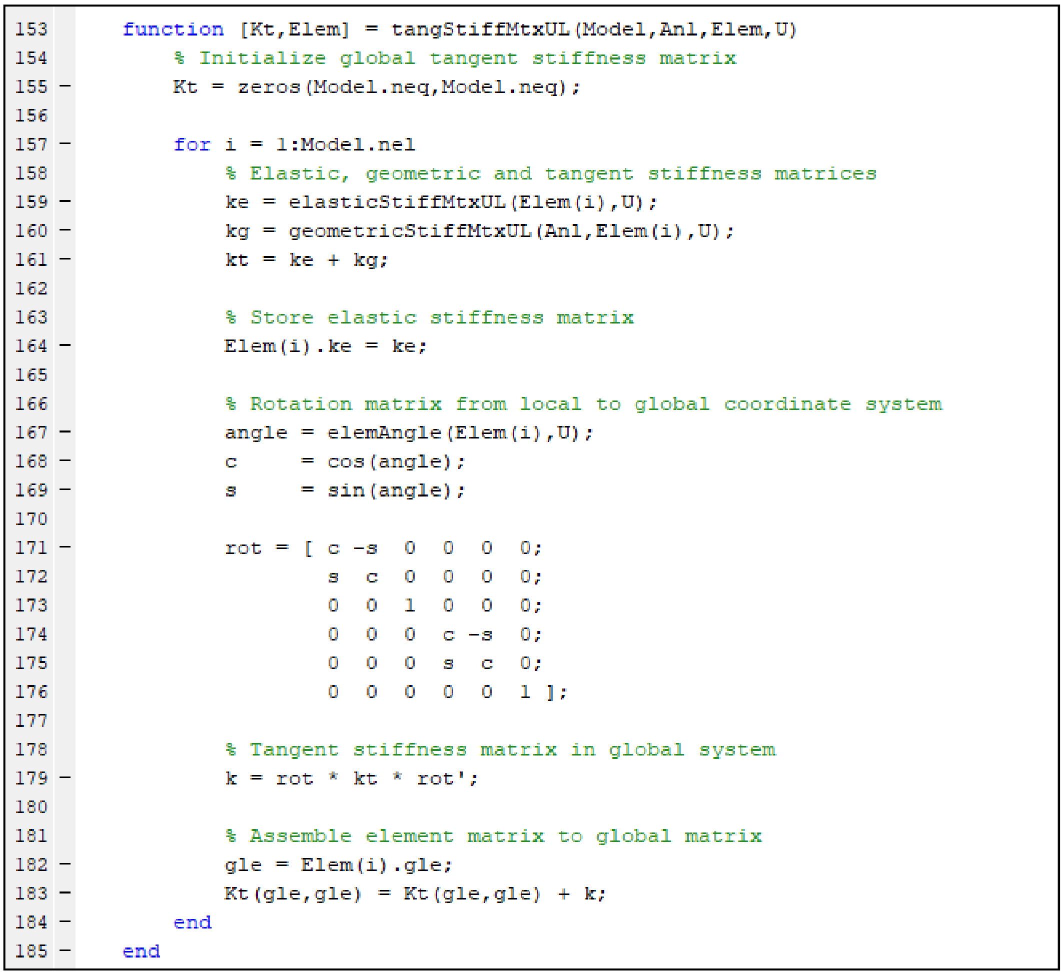
Figure 4.3 presents the implementation of the function to correct the predicted increment of displacements by means of second-order single-step methods (also known as second-order Runge-Kutta methods). This function was called in line 57 of the main function, after the predicted increments of load ratio and nodal displacements were obtained. The function receives as input arguments, respectively, the data structures with model, analysis, and elements information, the tangent stiffness matrix evaluated at the beginning of the step, the reference load vector, the predicted increment of load ratio, the total displacements, and the predicted increment of displacements. It returns the vector of corrected increment of displacements for the current step.

The function gets the weighting coefficients, `a1` and `a2`, and the second sampling point position, `z2`, for the selected method (lines 124 to 132), according to Eq. (3.72) and Eq. (3.73). Notice that the position of the first sampling point is always zero. The reason is that the tangent stiffness matrix of the beginning of the step is always considered for calculating the average matrix in the implemented methods. The nodal displacements of the second sampling point, `U_z2`, is then calculated (lines 135 and 136). The internal forces are evaluated for this level of displacements (line 139). Only the vector of elements with updated internal forces is returned by the auxiliary function because the global vector of nodal internal forces is not needed here. The tangent stiffness matrix of the first sampling point is set as the same matrix of the beginning of the step (line 142), and the tangent matrix of the second sampling point is evaluated with the corresponding displacements and element internal forces (line 143). The average matrix is calculated according to Eq. (3.71), in line 146. The corrected increment of displacements is then obtained with a linear system using the average matrix and the vector of predicted increment of loads.

4.2.2. Evaluation of the Tangent Stiffness Matrix

The function to evaluate the tangent stiffness matrix, for a given level of displacements and element internal forces, is called at two moments of the main algorithm: in the predicted solution and in the corrected solution (of the single-step and incremental-iterative methods). The called function is only responsible for checking the selected formulation and call the corresponding function that implements the code for evaluating the tangent matrix. These implementations are shown in Fig. 4.4 for the UL formulation and in Fig. 4.5 for the CR formulation. Their input arguments are the data structures with model, analysis, and element information, and the vector of total displacements. The outputs are the new tangent stiffness matrix and, for the UL implementation, the vector of element data structures with the updated elastic stiffness matrix of each element.

The `tangStiffMtxUL` function, in Fig. 4.4, starts by initializing the global matrix of the structural model with zeros (line 155). Its dimension is the total number of degrees-of-freedom (number of equations) of the structure. Next, a loop through all elements begins. The tangent stiffness matrix of each element will be computed in the local coordinate system, rotated to the global system, and inserted into the global matrix.

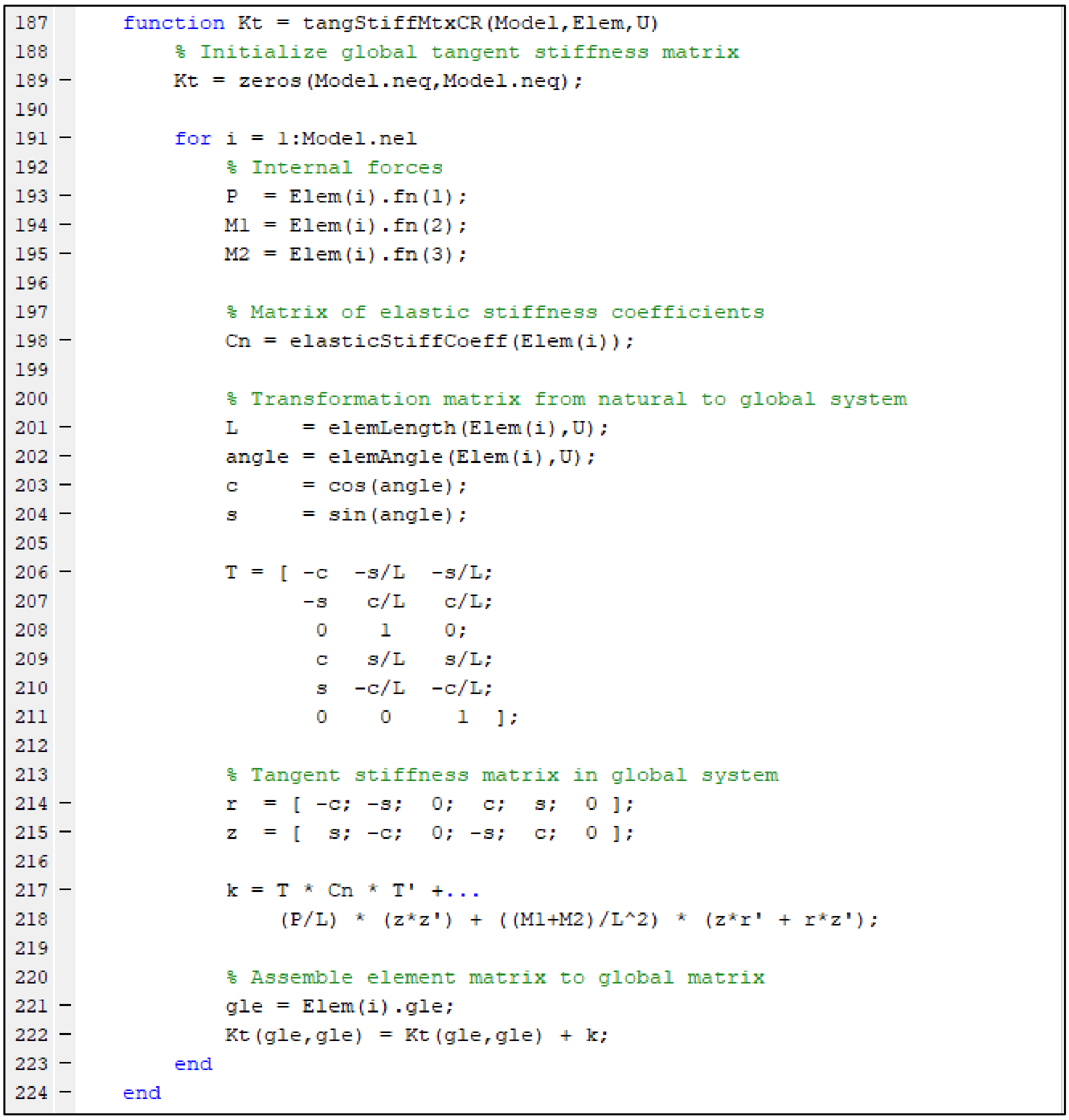


```
153      function [Kt,Elem] = tangStiffMtxUL(Model,Anl,Elem,U)
154          % Initialize global tangent stiffness matrix
155 –        Kt = zeros(Model.neq,Model.neq);
156
157 –        for i = 1:Model.nel
158              % Elastic, geometric and tangent stiffness matrices
159 –            ke = elasticStiffMtxUL(Elem(i),U);
160 –            kg = geometricStiffMtxUL(Anl,Elem(i),U);
161 –            kt = ke + kg;
162
163              % Store elastic stiffness matrix
164 –            Elem(i).ke = ke;
165
166              % Rotation matrix from local to global coordinate system
167 –            angle = elemAngle(Elem(i),U);
168 –            c     = cos(angle);
169 –            s     = sin(angle);
170
171 –            rot = [ c -s  0  0  0  0;
172                      s  c  0  0  0  0;
173                      0  0  1  0  0  0;
174                      0  0  0  c -s  0;
175                      0  0  0  s  c  0;
176                      0  0  0  0  0  1 ];
177
178              % Tangent stiffness matrix in global system
179 –            k = rot * kt * rot';
180
181              % Assemble element matrix to global matrix
182 –            gle = Elem(i).gle;
183 –            Kt(gle,gle) = Kt(gle,gle) + k;
184 –        end
185 –    end
```

Figure 4.4 – Implementation of the tangent stiffness matrix evaluation for the Updated Lagrangian formulation

For each element, the elastic and geometric stiffness matrices are evaluated in auxiliary functions. The function to evaluate the elastic matrix (line 159) must implement the expression given in Eq. (2.70). This function receives the vector of total displacements as input argument because it needs to calculate the current element length for the stiffness coefficients. The function to evaluate the geometric stiffness matrix (line 160) implements one of the three types of geometric matrix, given in Section 2.3.5.2: Small Rotation 2nd Order (Eq. (2.73)), Large Rotation 2nd Order (Eq. (2.76)), or Large Rotation 4th Order (Rodrigues, 2019). This function also needs the vector of total displacements to calculate the current element length, the analysis options to check the matrix type, and the current element internal forces, which are stored in the element data structure. The element tangent matrix, in local coordinate system, is then obtained by the sum of the elastic and geometric components (line 161). The local elastic stiffness matrix is stored in the element data structure (line 164) to be used when computing element internal forces.

The current element angle with the horizontal axis is calculated in an auxiliary function (line 167). This function has a simple implementation and uses only the initial nodal coordinates and the total displacements of nodes. The rotation matrix of the element degrees-of-freedom from local to global coordinate system is assembled in line 171 using the cosine and sine of the element angle, according to Eq. (2.123). Then, the tangent stiffness matrix in local coordinate system is rotated to the global system by pre and post multiplying it by the rotation matrix (line 179). Finally, the element tangent matrix is inserted into the global matrix using the gather vector (line 183), `gle`, that provides the global number of element degrees-of-freedom.



```
function Kt = tangStiffMtxCR(Model,Elem,U)
    % Initialize global tangent stiffness matrix
    Kt = zeros(Model.neq,Model.neq);

    for i = 1:Model.nel
        % Internal forces
        P  = Elem(i).fn(1);
        M1 = Elem(i).fn(2);
        M2 = Elem(i).fn(3);

        % Matrix of elastic stiffness coefficients
        Cn = elasticStiffCoeff(Elem(i));

        % Transformation matrix from natural to global system
        L     = elemLength(Elem(i),U);
        angle = elemAngle(Elem(i),U);
        c     = cos(angle);
        s     = sin(angle);

        T = [ -c   -s/L  -s/L;
              -s    c/L   c/L;
               0     1     0;
               c    s/L   s/L;
               s   -c/L  -c/L;
               0     0      1  ];

        % Tangent stiffness matrix in global system
        r  = [ -c; -s;  0;  c;  s;  0 ];
        z  = [  s; -c;  0; -s;  c;  0 ];

        k = T * Cn * T' +...
            (P/L) * (z*z') + ((M1+M2)/L^2) * (z*r' + r*z');

        % Assemble element matrix to global matrix
        gle = Elem(i).gle;
        Kt(gle,gle) = Kt(gle,gle) + k;
    end
end
```

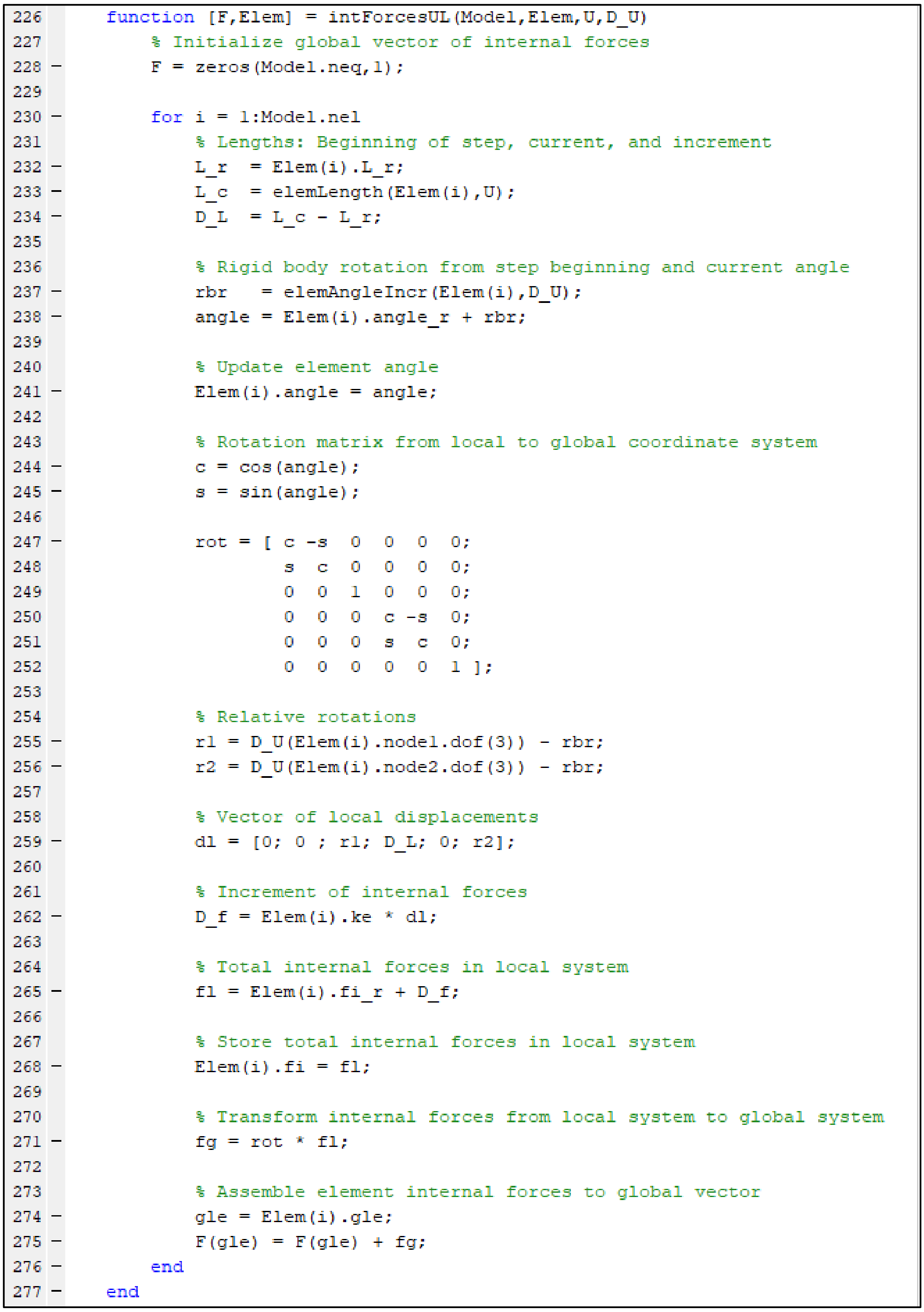
Figure 4.5 – Implementation of the tangent stiffness matrix evaluation for the Corotational formulation

The `tangStiffMtxCR` function to evaluate the tangent stiffness matrix of the CR formulation has a very simple implementation, as shown in Fig. 4.5. Again, it starts by initializing the global matrix of the structure (line 189), which will receive the contribution of each element tangent matrix computed in the global coordinate system. For each element, the internal forces (axial force and bending moments at end nodes) are extracted from the element data structure (line 193 to 195). The matrix of elastic stiffness coefficients that relates displacements and forces in natural system is obtained from an auxiliary function (line 198). This function simply returns the constant 3x3 matrix of Eq. (2.93). The transformation matrix from natural to global system, given in Eq. (2.114), is assembled in line 206. This matrix uses the current element length, returned by an auxiliary function (line 201), and the cosine and sine of the element angle with the horizontal axis, also returned by an auxiliary function (line 202). The element tangent matrix is computed in line 214, according to the expression of Eq. (2.146). This matrix is already in the global system, and it is inserted into the global matrix using the gather vector (line 222). This function could simply implement the final expression of each element tangent matrix, given in Eq. (2.147), but it was opted to show it in a more didactic form.

An important aspect that was omitted in the evaluation of tangent stiffness matrices for both UL and CR formulation was the treatment of rotation liberation at element ends (hinges). In the FRAMOOP system, this is done by a process of static condensation of the rotation degrees-of-freedom of the local tangent matrix. Furthermore, the stiffness matrices for Timoshenko beam theory were also implemented in FRAMOOP (although it was not presented in this document).

4.2.3. Evaluation of the Vector of Internal Forces

The function to evaluate the vector of internal forces, for a given level of displacements, also has two implementations, one for each type of nonlinear formulation. These implementations are shown in Fig. 4.6 for the UL formulation and in Fig. 4.8 for the CR formulation. Their input arguments are the data structures with model and element information, the vector of total displacements, and the vector of displacement increments (accumulated step increments for the UL formulation and iterative increments for CR formulation). The outputs are the global vector of internal forces and the vector of elements with updated internal forces in local or natural systems.



```
function [F,Elem] = intForcesUL(Model,Elem,U,D_U)
    % Initialize global vector of internal forces
    F = zeros(Model.neq,1);

    for i = 1:Model.nel
        % Lengths: Beginning of step, current, and increment
        L_r  = Elem(i).L_r;
        L_c  = elemLength(Elem(i),U);
        D_L  = L_c - L_r;

        % Rigid body rotation from step beginning and current angle
        rbr   = elemAngleIncr(Elem(i),D_U);
        angle = Elem(i).angle_r + rbr;

        % Update element angle
        Elem(i).angle = angle;

        % Rotation matrix from local to global coordinate system
        c = cos(angle);
        s = sin(angle);

        rot = [ c -s  0  0  0  0;
                s  c  0  0  0  0;
                0  0  1  0  0  0;
                0  0  0  c -s  0;
                0  0  0  s  c  0;
                0  0  0  0  0  1 ];

        % Relative rotations
        r1 = D_U(Elem(i).node1.dof(3)) - rbr;
        r2 = D_U(Elem(i).node2.dof(3)) - rbr;

        % Vector of local displacements
        dl = [0; 0 ; r1; D_L; 0; r2];

        % Increment of internal forces
        D_f = Elem(i).ke * dl;

        % Total internal forces in local system
        fl = Elem(i).fi_r + D_f;

        % Store total internal forces in local system
        Elem(i).fi = fl;

        % Transform internal forces from local system to global system
        fg = rot * fl;

        % Assemble element internal forces to global vector
        gle = Elem(i).gle;
        F(gle) = F(gle) + fg;
    end
end
```

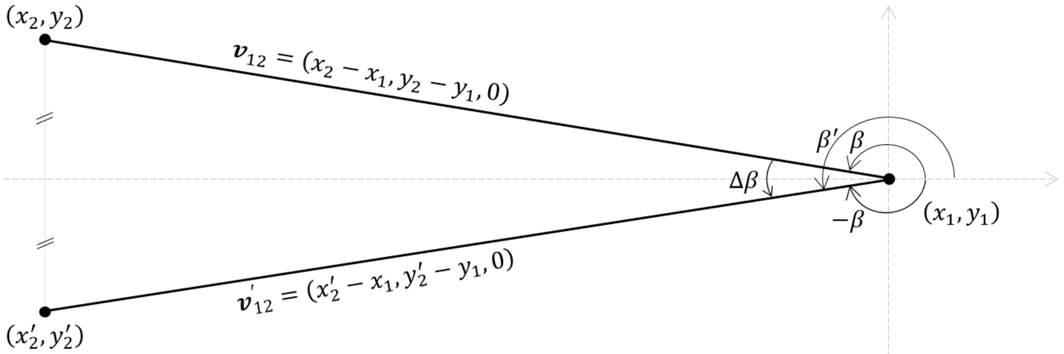
Figure 4.6 – Implementation of the internal forces evaluation for the Updated Lagrangian formulation

The implementation for the UL formulation starts by initializing the global vector of internal forces (line 228). This is a column vector with dimension equivalent to the total number of degrees-of-freedom (number of equations) of the structure. Then, a loop through all elements is executed to compute the vector of internal forces of each element in local system, rotate it to global system, and insert it into the global vector.

In the beginning of the loop, the element elongation, `D_L`, is calculated (line 234). It is obtained by the difference between the current element length, `L_c`, and the length at the reference configuration (step beginning), `L_r`. The current length is returned by an auxiliary function that receives the total nodal displacements (line 233). The length at the beginning of the step was stored in the element data structure when an auxiliary function was called in line 10 of the main function.

After that, the increment of element rotation (rigid-body rotation) from the step beginning, `rbr`, is calculated (line 236). The auxiliary function for this task receives the vector of displacement increments, from which the element rotation is calculated. The current element angle with the horizontal axis, `angle`, is then obtained (line 238) by adding the increment of rotation to the element angle of the step beginning. The current angle is updated in the element data structure (line 241).

The reason for the incremental approach in obtaining the current element angle, instead of simply calculate it using inverse trigonometric functions and current nodal coordinates, is that it must account for accumulated rotations over 180 or even 360 degrees, regardless of its quadrant. An example of a situation that may occur is provided in Fig. 4.7. An element is initially described by vector $\boldsymbol{v}_{12}$ and angle with the horizontal axis β. It passes from the second to the third quadrant, where it is describe by vector $\boldsymbol{v}'_{12}$ and angle β' in the new (current) configuration. The current angle is an increased value of the old angle. However, if the current angle is calculated using the inverse tangent function (the four-quadrant inverse tangent function `atan2`) and the nodal coordinates, the result is a negative angle. Therefore, the adopted solution is to increment the angle in small values, Δβ. Special attention should be given to the function that calculates this increment. Since the angle increment may be very small, the inverse cosine function `acos` should not be used because it is admittedly inaccurate for small values. The implemented expression to calculate the angle between two vectors is also shown.


(x_2, y_2)
$\boldsymbol{v}_{12} = (x_2 - x_1, y_2 - y_1, 0)$
β'
β
$\Delta\beta$
$-\beta$
(x_1, y_1)
$\boldsymbol{v}_{12}' = (x_2' - x_1, y_2' - y_1, 0)$
(x_2', y_2')

Wrong implementation: $\beta' = \tan^{-1}\left(\dfrac{y_2' - y_1}{x_2' - x_1}\right) = -\,\beta$

Adopted solution: $\beta' = \beta\ + \Delta\beta$; $\Delta\beta = \pm\tan^{-1}\left(\dfrac{|\boldsymbol{v}_{12} \times \boldsymbol{v}_{12}'|}{\boldsymbol{v}_{12} \cdot \boldsymbol{v}_{12}'}\right)$

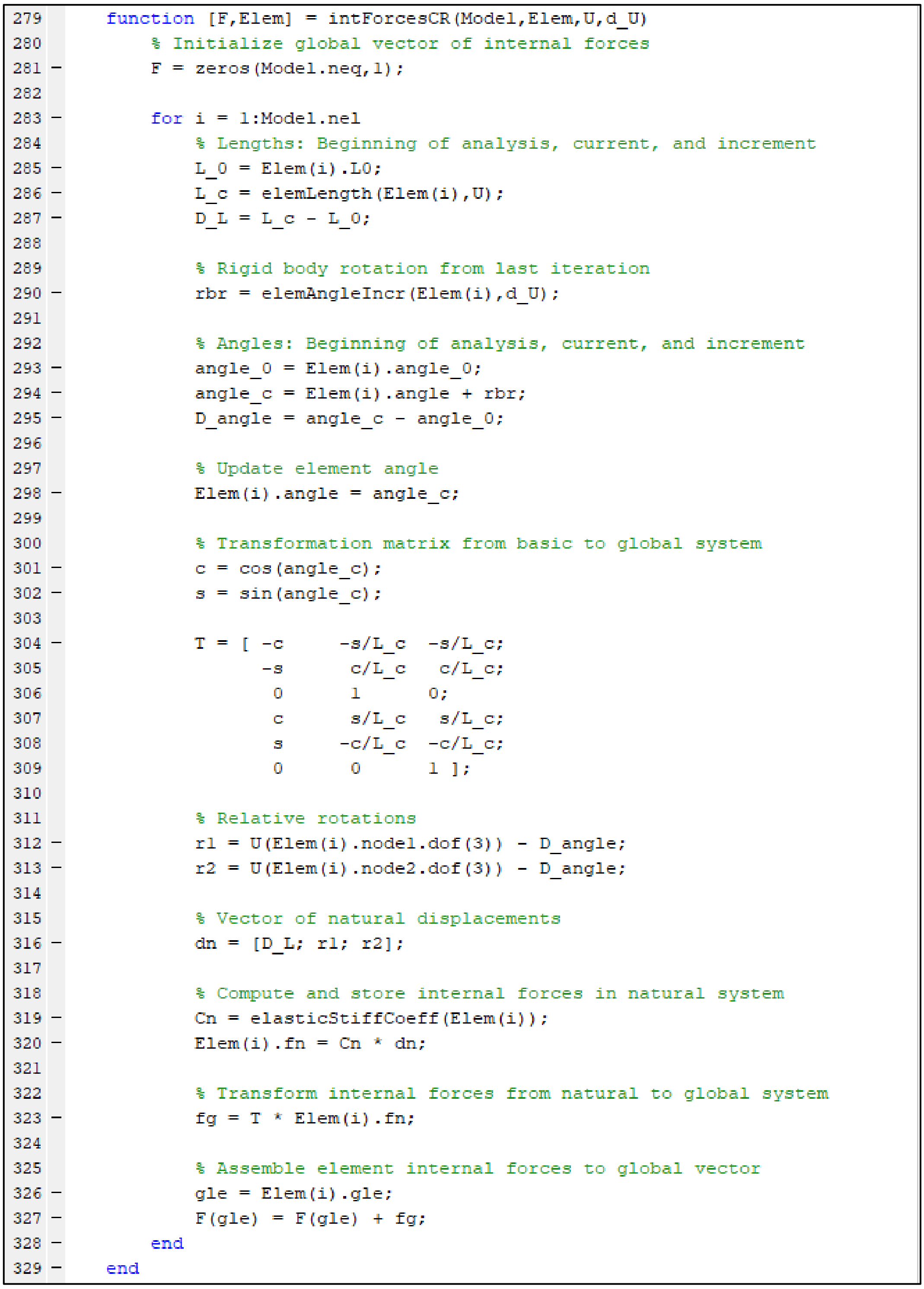
Sign determined by the direction of $\boldsymbol{v}_{12} \times \boldsymbol{v}_{12}'$

Figure 4.7 – Issue in calculating element rotation angle

Thereafter, the rotation matrix of the element degrees-of-freedom from local to global system is assembled (line 247) using the cosine and sine of the current element angle, according to Eq. (2.123). The incremental relative rotations of first and second nodes, `r1` and `r2`, are given by the difference between the increment of nodal rotations and the rigid-body rotation (lines 255 and 256). A 6x1 vector of relative displacements, `dl`, without rigid-body motions, is assembled with the element elongation and the relative rotations (line 259).

The vector of internal forces increment, `D_f`, is calculated with the product of the elastic stiffness matrix, stored in line 164, and the vector of relative displacements (line 262). This increment is added to the vector of total internal forces at the step beginning to obtain the total internal forces for the current configuration, `fl`, in the local coordinate system (line 265). The total internal forces in the local coordinate system is stored in the element data structure (line 268) to be used when evaluating the tangent stiffness matrix.

The vector of internal forces in the local system is rotated to the global system by multiplying it by the rotation matrix (line 271). Finally, the element internal forces are inserted into the global vector using the element gather vector (line 275).



```
function [F,Elem] = intForcesCR(Model,Elem,U,d_U)
    % Initialize global vector of internal forces
    F = zeros(Model.neq,1);

    for i = 1:Model.nel
        % Lengths: Beginning of analysis, current, and increment
        L_0 = Elem(i).L0;
        L_c = elemLength(Elem(i),U);
        D_L = L_c - L_0;

        % Rigid body rotation from last iteration
        rbr = elemAngleIncr(Elem(i),d_U);

        % Angles: Beginning of analysis, current, and increment
        angle_0 = Elem(i).angle_0;
        angle_c = Elem(i).angle + rbr;
        D_angle = angle_c - angle_0;

        % Update element angle
        Elem(i).angle = angle_c;

        % Transformation matrix from basic to global system
        c = cos(angle_c);
        s = sin(angle_c);

        T = [ -c     -s/L_c  -s/L_c;
              -s      c/L_c   c/L_c;
               0      1      0;
               c      s/L_c   s/L_c;
               s     -c/L_c  -c/L_c;
               0      0      1 ];

        % Relative rotations
        r1 = U(Elem(i).node1.dof(3)) - D_angle;
        r2 = U(Elem(i).node2.dof(3)) - D_angle;

        % Vector of natural displacements
        dn = [D_L; r1; r2];

        % Compute and store internal forces in natural system
        Cn = elasticStiffCoeff(Elem(i));
        Elem(i).fn = Cn * dn;

        % Transform internal forces from natural to global system
        fg = T * Elem(i).fn;

        % Assemble element internal forces to global vector
        gle = Elem(i).gle;
        F(gle) = F(gle) + fg;
    end
end
```

Figure 4.8 – Implementation of the internal forces evaluation for the Corotational formulation

In the implementation of the CR formulation, the first difference is using the initial length of each element to compute its total elongation (line 287). The initial length, `L_0`, was stored in the element data structure in the pre-processing stage, and the current length, `L_c`, is calculated with the total displacements.

The increment of element rotation (rigid-body rotation) from the previous iteration, `rbr`, is calculated with the iterative increment of displacements (line 290). The iterative increment of element rotation is then added to the previous element angle with horizontal axis to obtain the current element angle (line 294). The purpose of this incremental calculation of the current element angle is the same as previously explained for the UL implementation. The difference now is that it is incremented with iterative values. The total increment of the element rotation angle (rigid-body rotation) from the beginning of the analysis, `D_angle`, is given by the difference between the current angle and the initial angle (line 295). The current angle is stored in the element data structure (line 298).

The transformation matrix from natural to global system is assembled according to Eq. (2.114) using the cosine and sine of the current element angle (line 304). The relative rotations of first and second nodes, `r1` and `r2`, are obtained by the difference between total nodal rotations and total element rotation (lines 312 and 313). The 3x1 vector of natural displacements, `dn`, is assembled with the total element elongation and relative nodal rotations (line 316).

The vector of force components in the natural system (axial force and bending moments at end nodes) is given by the product of the matrix of elastic stiffness coefficients, `Cn`, provided in Eq. (2.93), and the vector of natural displacements (line 320). The resulting natural internal forces are readily stored in the element data structure to be used when evaluating the tangent stiffness matrix. The internal forces in the natural system are then rotated and expanded to the global system using the transformation matrix (line 323). To finish each loop iteration, the internal forces, in the global system, are assembled to the global vector (line 327).

It is notable the similarity between the implementation of the function to evaluate internal forces for both formulations. Actually, the UL formulation uses a Corotational approach to remove the rigid body motions from the vector of local displacements. The non-deformable motions need to be removed from the displacements to evaluate internal forces in any nonlinear formulation, as they can lead to spurious force values.

4.3. Graphical User Interface and Its Functionalities

Besides the implementation of the geometrically nonlinear analysis algorithm in the FRAMOOP system, which is an external library responsible only for the analysis process, the Ftool (Two-dimensional Frame Analysis Tool) program (Martha, 1999) also needed to be modified in order to receive the new functionalities of nonlinear analyses. This section is dedicated to presenting the new version of the Ftool program, showing the modifications from a user point of view, i.e., the new features in the graphical interface, their purposes, and how to use them.

The Ftool program (https://www.ftool.com.br) was conceived in 1991, from a research project of the Tecgraf Institute of Technical-Scientific Software Development of PUC-Rio (Tecgraf/PUC-Rio). It consists of a graphical-interactive structural analysis program, based on the direct manipulation of structural models via mouse and keyboard. Its main goal is to provide a software that motivates the engineering student to learn the theory of structural analysis methods by showing how frame models behave in practice. There is no concern to teach the mathematical and computational processes for the analysis of these structures. Therefore, Ftool needs to be a simple application that brings together, in a single platform, all the necessary resources for efficient model creation and manipulation, with attributes application (pre-processing), fast and transparent numerical analysis (solver), and fast and effective visualization of results (post-processing). This integration of all phases of the structural analysis is the fundamental aspect in the learning process, providing students with a complete control over the model being analyzed and allowing them to quickly experiment different structural conceptions and, thus, better understand the behavior of structures.

Given the purposes of the program, during its creation, it was necessary a data structure that was common to all phases of the simulation. The data structure should allow the detection of inconsistencies in the model definition, an efficient way to register adjacency relations between model entities, and provide efficient geometric operators, including the automatic detection of member intersection. The chosen data structure is centered on a complete topological representation of a planar subdivision, with efficient search for adjacency information based on the theory of solid modeling (Mäntylä, 1988), called Half-Edge Data Structure (HED) (Carvalho *et al.*, 1990).

Moreover, the program must offer a user-friendly graphical interface, with intuitive resources. The graphical user interface of the Ftool program is built using the IUP (Portable User Interface) system (Levy *et al.*, 1996), developed at Tecgraf/PUC-Rio. This system is a multi-platform toolkit that offers a simple API (Application Program Interface) with many functions for creating and manipulating dialogs and interface elements in different programming languages, including C. It is intended to allow the developed software to be executed on various platforms by simply recompiling it on the desired platform and linking it with the appropriate graphic libraries, without the need to modify the source code, which gives the program a high portability. This portability is only possible because IUP uses native interface elements of the operating systems. The layout of the interface elements within the Ftool dialog is stored in a text file written in LED (Dialog Specification Language). The LED file is converted to a C file that is compiled with the rest of the Ftool code.

These described characteristics of the Ftool development are responsible for its success in teaching structural engineering over the last decades. It has been used on solid mechanics, structural analysis, and structural design courses at many universities all over the world. Because of that, it provides the ideal environment for the intuitive nonlinear analysis module developed in this work. Thereby, the program gains important features that bring more possibilities to be explored in academic and professional activities. Despite the advanced concepts involved in the nonlinear analysis, Ftool will retain the simplicity features already enshrined in previous versions, as its basic purpose remains educational.

To support this new version, the main window of the program had to be modified. Figure 4.9 shows the main dialog of Ftool, highlighting the buttons to display the new menus created for this work: Analysis Menu and Plotting Menu. When selected, these menus appear along the right side of the screen, and their features are described next.

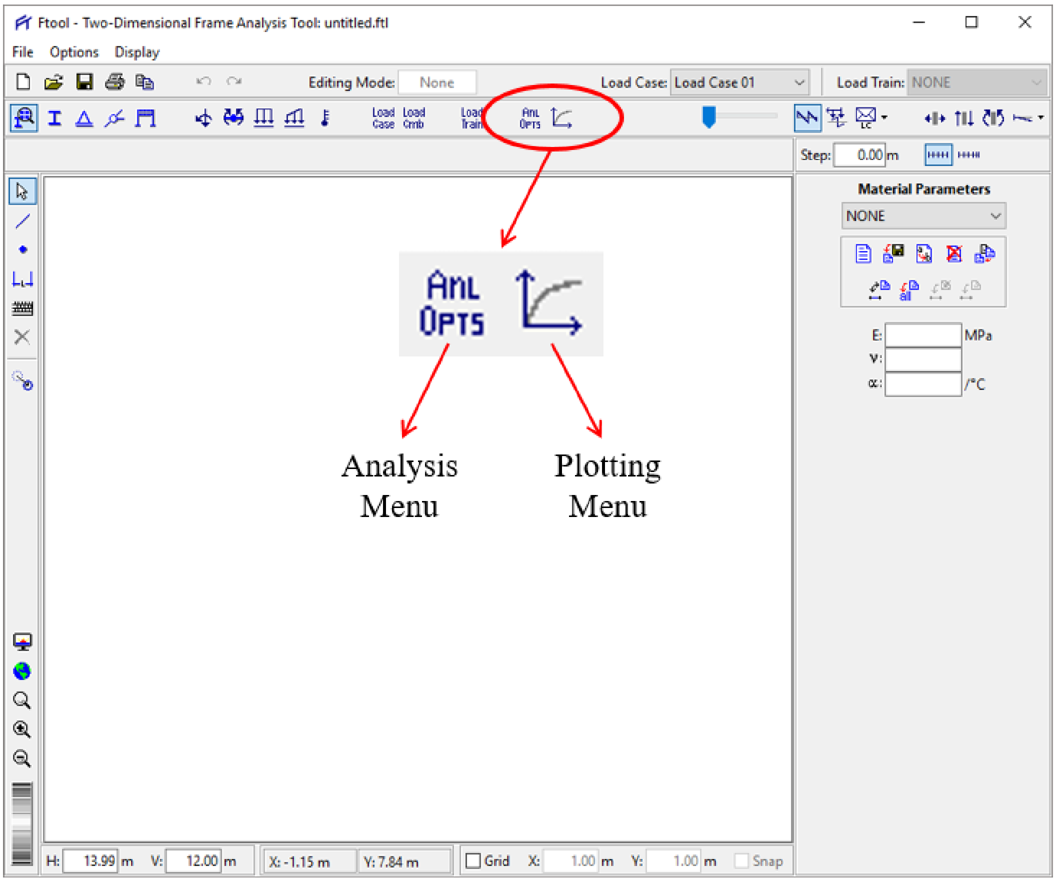
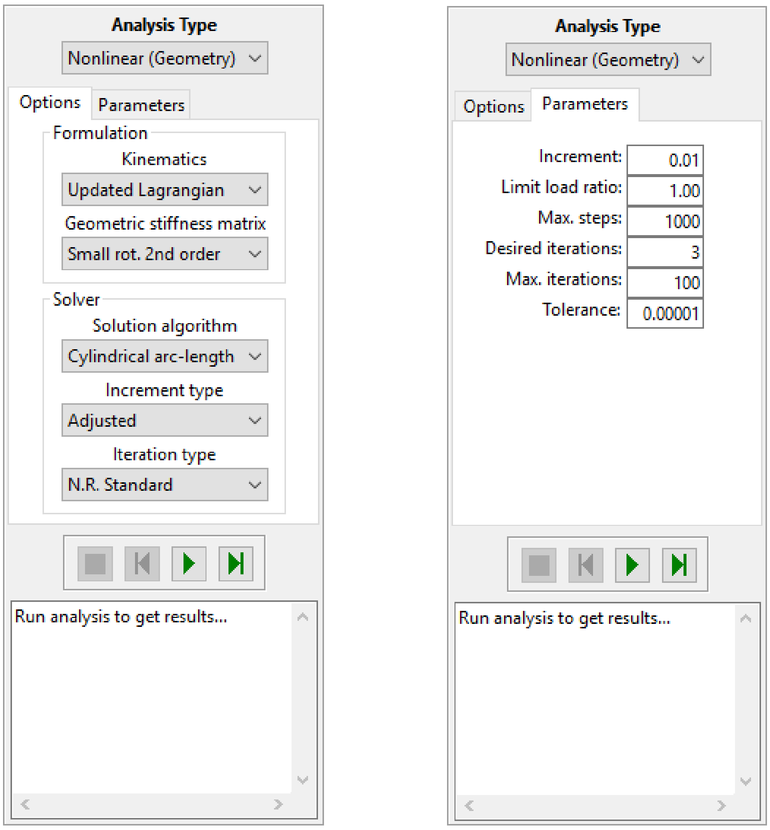


Figure 4.9 – Developed menus for performing nonlinear analyses in the Ftool program

4.3.1. Analysis Options and Parameters

In the Analysis Menu, users can switch between the linear-elastic analysis and the geometrically nonlinear analysis with a drop-down list, entitled *Analysis Type*, that offers both options. The former is the default option when a new model is created. When this option is selected, the Analysis Menu turns empty. In this case, the program works as in previous versions, with the analysis being automatically performed when a diagram result is requested. On the other hand, when the geometrically nonlinear analysis is selected, a series of options and parameters show up to be set by users. In this case, the analysis must be launched manually before checking the results. The reasons for requesting a manual analysis for the nonlinear behavior were given throughout this work: the convergence is not always achieved, the results may vary slightly for different methods and parameters used, and it is desired to study the history of equilibrium solutions of the structural model with an incremental approach.

The Analysis Menu is shown in Fig. 4.10. It can be seen that the nonlinear analysis options and parameters are organized in two tabs: *Options* tab (Fig. 4.10a) and *Parameters* tab (Fig. 4.10b). The former is designed to provide a wide range of options related to the nonlinear formulation and to the strategies to solve the problem. The latter is where analysts can provide the numerical parameters that are used by these strategies.



Analysis Type
Nonlinear (Geometry)

Formulation
Kinematics: Updated Lagrangian
Geometric stiffness matrix: Small rot. 2nd order

Solver
Solution algorithm: Cylindrical arc-length
Increment type: Adjusted
Iteration type: N.R. Standard

Run analysis to get results...

(a)

Analysis Type
Nonlinear (Geometry)

Increment: 0.01
Limit load ratio: 1.00
Max. steps: 1000
Desired iterations: 3
Max. iterations: 100
Tolerance: 0.00001

Run analysis to get results...

(b)

Figure 4.10 – Nonlinear analysis menu showing (a) analysis options and (b) analysis parameters

In the *Options* tab, a *Formulation* panel provides options for the kinematic description used to formulate the nonlinear structural problem, which were described in Chapter 2: Updated Lagrangian formulation and Corotational formulation. In the same panel, the type of geometric stiffness matrix can also be selected. If the UL formulation is set as the kinematic description, three options of geometric stiffness matrix are available: Small Rotation 2nd Order (Eq. (2.73)), Large Rotation 2nd Order (Eq. (2.76)), and Large Rotation 4th Order (Rodrigues, 2019). On the other hand, if the CR formulation is selected as kinematics, the geometric matrix option is locked in Small Rotation 2nd Order, which corresponds to the only stiffness matrix developed for this formulation (Eq. (2.147)).

A second panel in the *Options* tab, entitled *Solver*, contains the options related to the strategies to solve the nonlinear problem, which were discussed in Chapter 3. In the *Solution algorithm* pop-up menu, users can select one of the single-step or incremental-iterative methods implemented to perform the analysis. The type of increment can be set right below the solution method. This option can be set to adjusted or constant. The adjusted option automatically modifies the size of the predicted increment of load ratio according to the degree of nonlinearity of the solution. The constant option does not use the adjustment factor of the methods based on the number of iterations to modify the predicted increment, keeping the increment of the control parameter with the prescribed value for the entire analysis. The last option is the specification of the iteration scheme to be used in incremental-iterative methods. The available options are the standard and modified versions of the Newton-Raphson (*N.R.*) iteration scheme, which define whether the tangent stiffness matrix is updated in all iterations or just at the beginning of the incremental step. This option is not available to be set in incremental single-step methods.

In the *Parameters* tab, the first input field is the value of the predicted increment of load ratio for the first analysis step. If the increment type option is set to adjusted, this predicted increment is automatically modified in subsequent steps, to optimize the analysis. Otherwise, if the increment type option is set to constant, it remains the same in all steps. The next two parameters (limit load ratio and maximum number of steps) determine when the analysis will stop. When any of these limiting values, of load ratio or steps, is reached, the analysis breaks. It is not guaranteed that the solution will be obtained up to these values, as problems related convergence, stiffness matrix singularity, and complex numbers can occur earlier. Following is the desired number of iterations, which defines the N_i value in Eq. (3.32). 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. The last two parameters (maximum number of iterations and tolerance) are information for the convergence criterion. The former defines the number of iterations to be performed within an incremental step to assume that the algorithm failed to converge to an equilibrium solution. The latter is the value of the numerical tolerance, variable ε used in Eq. (3.74), to assume that an equilibrium configuration has been established. Notice that the last three parameters are information related to incremental-iterative methods only, so they stay inactive when a single-step method is selected.

Below the tabs of options and parameters are the analysis control buttons. These buttons allow users to have full control of the nonlinear analysis and perform it in an interactive-adaptive fashion. There are four control buttons: *Reset*, *Backward*, *Play*, and *Forward*.

The *Play* button runs the analysis up to the maximum number of steps, the limit value of load ratio, or the detection of a problem. The analysis only starts if it is verified that the model is stable and the attributes have been set correctly. When the nonlinear analysis starts, and reaches convergence at least in the first step, the program enters in post-processing mode. That is, the modeling options stay blocked and users can request any diagram result. The default diagram that is exhibited after the analysis stops is the deformed configuration. If a problem is detected before the end of the analysis, the result is shown for the last obtained equilibrium configuration. The *Forward* button performs a single step at a time. The *Backward* button is only enabled when at least one analysis step has been successfully performed. This button returns the analysis in one step, by cleaning the saved information of the removed step. To return a selected number of steps, one can click on the desired step number in the analysis feedback at the bottom of the menu and then press the *Backward* button. The *Reset* button is also enabled only when the analysis is running. When clicked, all analysis information is cleaned and the program returns to pre-processing mode. A keyboard shortcut has been set for each of these control buttons.

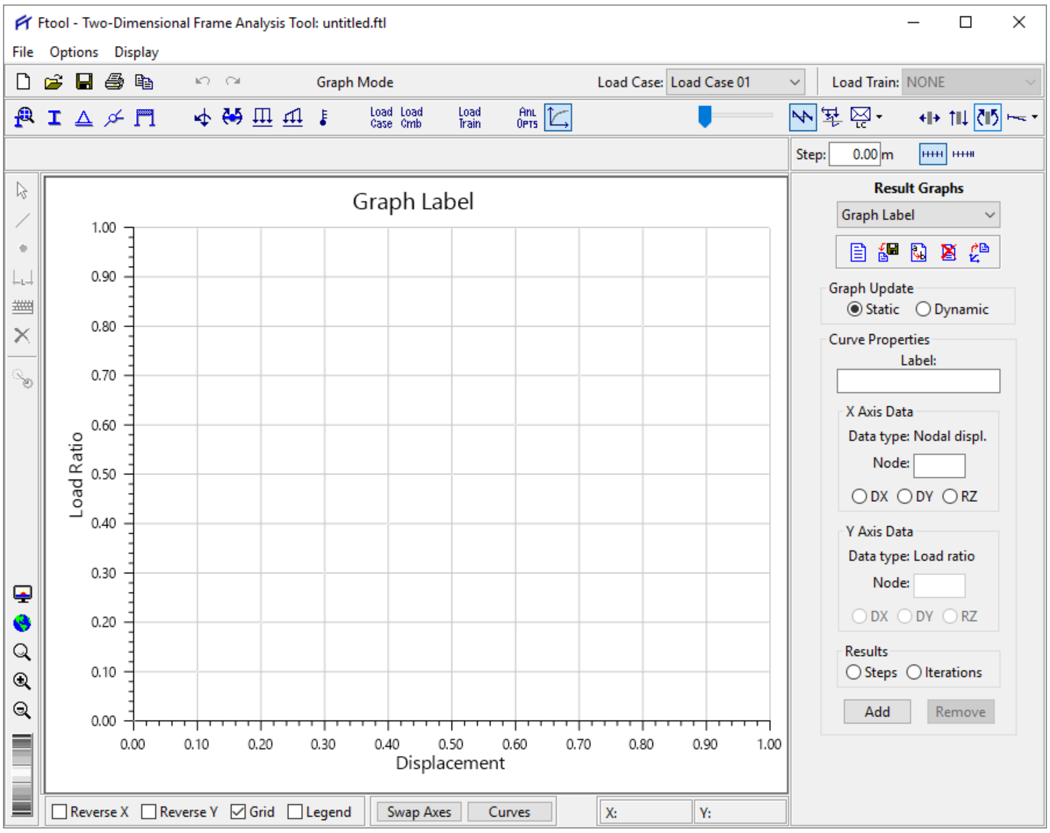
Changes in any of the analysis options and parameters are allowed in between steps, so the analysis will resume with the new input data. This possibility to perform the analysis in an interactive-adaptive way is an important feature of the developed tool. When a non-converging point is found, one can change the parameters or use other solution method, in the same analysis, to go beyond that point without having to restart the entire process. These options for driving the analysis can help even experienced users to work with numerical algorithms, and increase their sensitivity on the use of the numerical methods to obtain the nonlinear response, allowing studies on the influence of the input parameters to the converged solution. The implementation of these control options was done by saving to a linked list all the necessary data to start the analysis at any given step, based on the history of the results.

A step-by-step feedback of the analysis progress is given in a text field at the bottom of the Analysis Menu. It dynamically provides the step number, load ratio, and number of performed iteration after each increment.

4.3.2. Plotting Environment

One of the most important result in a nonlinear analysis is the history of the behavior of some variables, especially the relation between the load ratio and the displacement of a particular degree-of-freedom in the equilibrium configurations, also known as equilibrium path. The best way to show these results is in the form of graphs. For this reason, a sophisticated graph-plotting environment was developed, where users can create interactive graphs and select between several options of data to be plotted in each axis.

The new plotting area of the Ftool program is developed and managed by an additional control of the IUP system called IupPlot, which provides a library of functions to deal with graph plotting. The developed menu for creating and controlling graphs is shown in Fig. 4.11, with an empty graph.



Load Case: Load Case 01

Load Train: NONE

Step: 0.00 m

Graph Label

Graph Update: ● Static ○ Dynamic

Label:

X Axis Data — Data type: Nodal displ. Node: ○ DX ○ DY ○ RZ

Y Axis Data — Data type: Load ratio Node: ○ DX ○ DY ○ RZ

Results: ○ Steps ○ Iterations

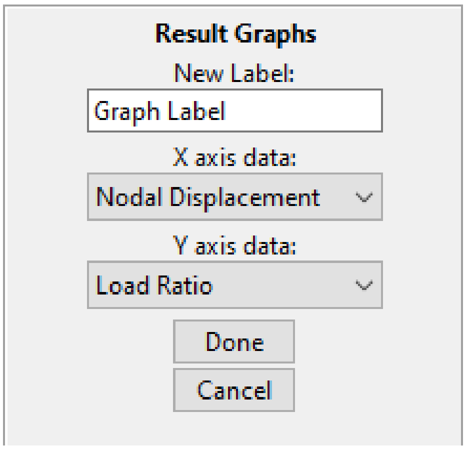
□ Reverse X □ Reverse Y ☑ Grid □ Legend

Figure 4.11 – Plotting Menu showing an empty graph

When the Plotting Menu button is clicked, the program enters in plotting mode. This mode is a new state of the program in which the canvas, where the model is drawn, is replaced with the plotting area; The options of the edit toolbar, on the left side of the screen, for the creation and manipulation of models, are disabled; The coordinate controls at the bottom of the screen is replaced with graph control options; The Plotting menu is displayed on the right side of the screen.

The menu initially displays a drop-down list with the created graphs, and five managing buttons. The graph selected in the drop-down list appears in the plotting area. If no graph has been created or the option *None* is selected, the potting area will be blank.

The first managing button is the *New* button. When clicked, a few options for creating a new graph are displayed: graph label, *X*-axis data, and *Y*-axis data (Fig. 4.12). The label is the string that appears on the graph title. The *X* and *Y*-axis data are the information that can be plotted on each axis. The available options are nodal displacement, load ratio, step number, displacement increment, and load ratio increment. The default information is nodal displacement for the *X*-axis and load ratio for the *Y*-axis, i.e., the conventional graph of equilibrium path. With these options, many aspects of the solution can be investigated. 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 ratio increment for each analysis step, and so on. When the new graph is created, it appears empty in the plotting area.



Result Graphs

New Label:

Graph Label

X axis data:

Nodal Displacement

Y axis data:

Load Ratio

Figure 4.12 – Options for creating a new graph

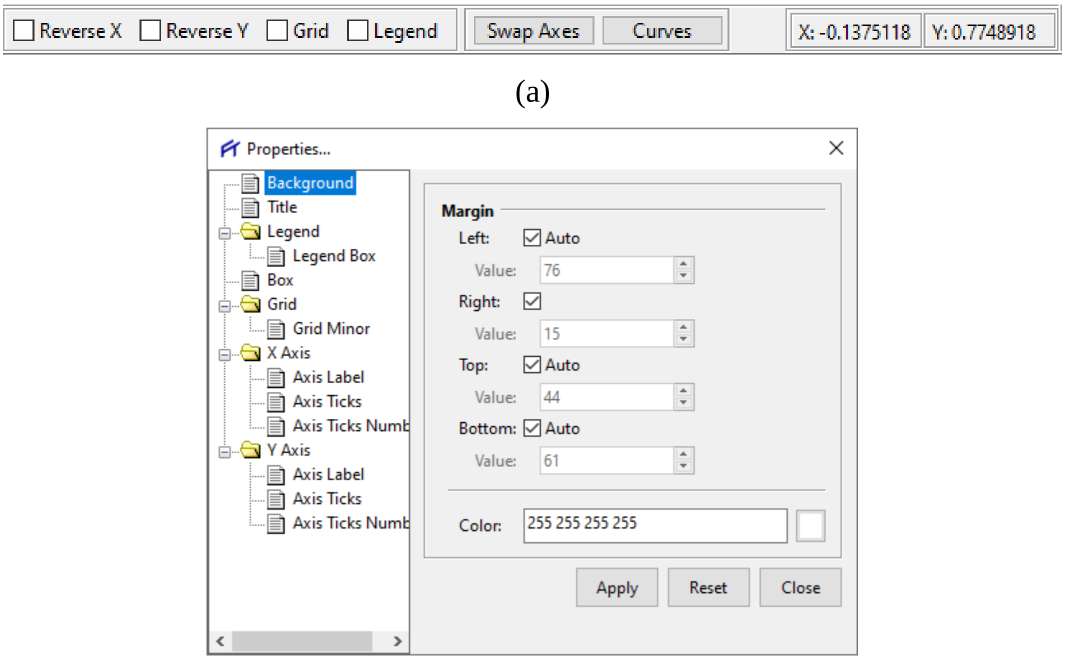
The other managing buttons are the *Import* button, to load a created graph from a data file that has been saved, the *Rename* and *Remove* buttons, to rename and delete the current selected graph, and the *Export* button, to save the information of the current graph to a data file.

When a created or imported graph is selected from the drop-down list, some options appear in the Plotting Menu, below the managing buttons. The first is the update type of the current graph. Each graph can be set as static or dynamic. Static graphs never change its data while dynamic graphs are automatically updated with the ongoing analysis. This means that when the analysis step changes, the graph updates its data to be synchronized with the analysis. In addition, the properties of the curves of a dynamic graph can be changed at any time. Imported graphs from other analysis cannot be set as dynamic.

Next is the *Curve Properties* panel, where users can add new curves to the current graph by setting their properties. For a curve to be added, a label must be provided. Then, the degree-of-freedom for each axis data must be specified by providing the node number and selecting one of its three degrees-of-freedom. To assist users, when the cursor is placed in the text field to provide the node number, the model appears on the screen and the desired node can be picked with a mouse click. Depending on the data type, it is not necessary to specify a degree-of-freedom for the axis. For example, an axis that plots the value of the load ratio does not need the information of a degree-of-freedom, so these options are blocked for that axis.

Before adding the new curve, the last option to be set is the type of result for that curve. Two options can be selected: steps and iterations. The former indicates that the curve points correspond only to the converged equilibrium solutions. The latter plots the solutions obtained in all iterations, so that users can examine the behavior of the iterative cycle within each analysis step of incremental-iterative methods. A more in-depth study of the solution algorithms can be done from this feature. After setting the curve properties, it can be added to the graph with the *Add* button. When an existing curve is clicked, it is highlighted and its properties appear in the Plotting Menu, also enabling the *Remove* button next to the *Add* button. As commented earlier, if the current graph, to which the selected curve belongs, is dynamic, the curve properties can be changed. Otherwise, only its label can be modified.

The graph control toolbar at the bottom of the screen (Fig. 4.13a) brings options to reverse the axes directions, display or hide the grid and legend, swap axes data and properties, and open a dialog with all the curves of the current graph, where users can select which ones are displayed. In addition, a properties menu (Fig. 4.13b) can be accessed by right clicking on the graph. In this menu, many visual properties of the graph can be set, such as colors, limits, curve thickness, etc.



X: -0.1375118 Y: 0.7748918

(a)

Margin

Left: ☑ Auto
Value: 76
Right: ☑
Value: 15
Top: ☑ Auto
Value: 44
Bottom: ☑ Auto
Value: 61

Color: 255 255 255 255

(b)

Figure 4.13 – Graph control toolbar (a) and properties menu (b)

4.3.3. Diagram Results

In addition to graph results, usual diagram results of internal forces (axial force, shear force, and bending moment) and deformed shape are also provided in the geometrically nonlinear analysis. All these diagrams are plotted over the initial (undeformed) configuration of the model to ease visualization and interpretation. Unlike a linear-elastic analysis, in which the deformed configuration is interpolated from nodal displacements using shape functions, the deformed configuration of a geometrically nonlinear analysis is drawn with straight lines connecting the deformed nodal coordinates. The reason is that the shape functions used in linear analyses cannot represent well the nonlinear deformations, leading to erroneous deflection modes.

5. Numerical Results

5.1. Introduction

In this chapter, four benchmark problems are analyzed, each one with a distinct nonlinear behavior. The first two problems consist, respectively, of a cantilever beam under a vertical point load and a concentrated moment at the tip. These models do not present any critical point along the equilibrium path, but some interesting aspects of the geometrically nonlinear analysis can be demonstrated with them. The third problem is the Williams Toggle Frame, which includes a snap-through behavior in its solution. Finally, the last problem is the Lee Frame, whose solution has the strongest nonlinearities among the analyzed examples, with load and displacement limit points in its equilibrium path.

The analysis of these models has two main goals. One is to validate the results obtained by the new version of the Ftool program, based on the analytical or numerical solutions for the geometrically nonlinear behavior of the benchmark problems. The second is to carry out a study on the performance of the incremental-iterative methods and the geometrically nonlinear formulations implemented in the program to solve the system of equilibrium equations. The solution methods were described in Chapter 3, which include 9 incremental-iterative methods. The nonlinear formulations were presented in Chapter 2, which are the Updated Lagrangian formulations (Small Rotation 2nd Order, Large Rotation 2nd Order, and Large Rotation 4th Order) and the Corotational formulation.

To accomplish the performance evaluation, each solution method is used together with each nonlinear formulation, considering different analysis options (type of increment and type of iteration scheme) to trace the equilibrium path. Four series of analyses are then performed in each example, and their results are grouped into tables. These tables consider the following combinations of analysis options to solve the problem with all methods and formulations: constant increments with standard iteration scheme; constant increments with modified iteration scheme; adjusted increments with standard iteration scheme; adjusted increments with modified iteration scheme. The numerical parameters, such as initial increment, tolerance for convergence, etc., are the same for all the analyses.

Remember that the difference between constant and adjusted increments is whether or not to use the adjustment factor to automatically modify the size of the predicted increment of load ratio according to the degree of nonlinearity of the solution. This is valid for the methods in which the increment strategy is based on the number of iterations (all but the GDCM), described in Section 3.4.1.1. The difference between the standard and modified iteration schemes is whether the tangent stiffness matrix is updated in all iterations or just at the beginning of the incremental step, as explained in Section 3.3.2.

The adopted criteria to measure the performance and efficiency of the solution methods and nonlinear formulations are based on the number of steps and corrective iterations. In each cell of the result tables, the first line indicates, respectively, the total number of steps and the total number of corrective iterations to complete the analysis. The average number of iterations per step is indicated on the second line, between parentheses. Keep in mind that the corrective iterations do not include the predicted step. When the solver fails to capture full solution, it is indicated by an “X”. The mean values of steps and iterations are also provided, but approximated to the nearest integer, and considering only the successfully performed analysis of each line or column of the table.

In addition, some other aspects of geometrically nonlinear analyses are worked in the examples. The importance and effects of discretizing each beam element is shown for the first two models. It is concluded that, for the implemented formulations, a 10-elements discretization is sufficiently fine to match the analytical solution of a beam subjected to shear force and bending. The numerical equivalency of the nonlinear formulations is also checked for plane beam elements with Euler-Bernoulli behavior, i.e., the solution obtained with the different formulations are numerically the same, what changes is the performance and efficiency when solving the problem. A few more results that can be extracted from the Ftool program are also shown, including graphs other than the classical curve of the equilibrium path.

5.2. Cantilever Beam with Vertical Tip Load

The first example is a simple model of a cantilever beam with a vertical load at the free end, illustrated in Fig. 5.1. The beam is 1 meter long, the material has a modulus of elasticity of $E = 10^7$ kN/m^2, the cross-section has an area of $A = 10^{-2}$ m^2 and moment of inertia of $I = 10^{-5}$ m^4, and the magnitude of the applied load is $P =$ 1000 kN. The displacements and rotation of the degrees-of-freedom of the free node (u, v, θ) are also indicated. Euler-Bernoulli bending behavior is assumed.

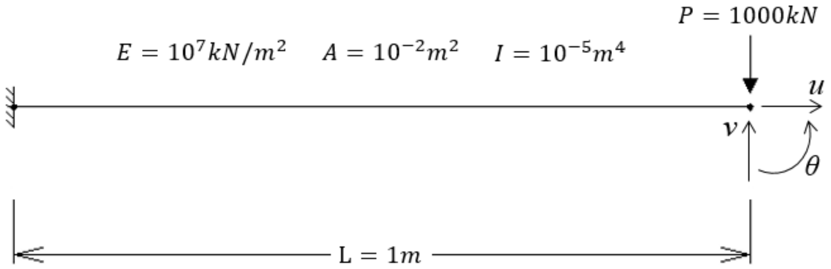


Figure 5.1 – Cantilever beam with vertical tip load

The analytical solution for this problem was studied by Bisshopp & Drucker (1945) and Mattiasson (1981) using elliptical integrals. The dimensionless results of the transcendental elliptical functions are frequently used to validate numerical models and they are given in Table 5.1 (Timoshenko & Gere, 1982).

Table 5.1 – Analytical solution for the cantilever beam with vertical tip load

PL^2/EI	u/L	v/L
0.00	0.000	0.000
0.25	0.004	0.083
0.50	0.016	0.162
0.75	0.034	0.235
1.00	0.056	0.302
2.00	0.160	0.494
3.00	0.255	0.603
4.00	0.329	0.670
5.00	0.388	0.714
6.00	0.434	0.744
7.00	0.472	0.767
8.00	0.504	0.785
9.00	0.531	0.799
10.00	0.555	0.811

Initially, a study on the model discretization will be conducted. The results will be shown for a 1-element discretization of the beam, which is done in a linear-elastic analysis without loss of precision, and for a discretization of 10 equal length elements to show that nonlinear results depend on model refinement to converge to the analytical solution. Furthermore, the numerical equivalency of the geometrically nonlinear formulations implemented is verified.

Figure 5.2 shows the deformed configuration exhibited by the Ftool program for different load levels, from the beginning of the analysis to the total load, using the two refinements. The corresponding load ratio value of each configuration is indicated. The deformed configuration of the single element discretization is drawn with a straight line from the deformed position of first and second nodes, while the discretized model uses piecewise linear segments between the deformed nodal coordinates. The beam discretized by multiple elements better represents the real behavior, as it will be proved by the numerical results. Clearly, an expected consequence of the poor discretization of this model is that the vertical displacement of the free node becomes greater than the analytical solution.

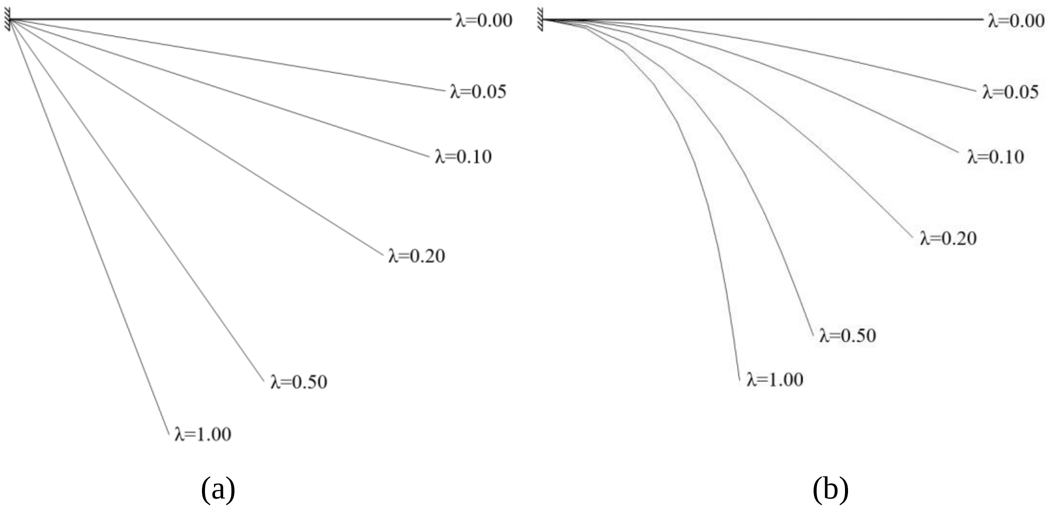


Figure 5.2 – Example 1: Deformed configurations of the cantilever beam with (a) 1-element and (b) 10-elements discretization

The geometrically nonlinear behavior of this model is simple, with a smooth equilibrium path that does not present any critical point. As the load increases, the structure becomes stiffer, which is caused by tension stiffening of the beam in its deformed configuration.

The equilibrium paths for the horizontal and vertical displacements of the 1-element discretized model are shown in Fig. 5.3 and Fig. 5.4, respectively. The equilibrium paths for the horizontal and vertical displacements of the 10-elements discretization are shown in Fig. 5.5 and Fig. 5.6, respectively. In each graph, the results obtained with different nonlinear formulations are compared to the analytical solution. For each nonlinear formulation, the equilibrium configurations were obtained with the Load Control Method, since there is no load limit point in the solution. A constant increment of 0.01, with the standard Newton-Raphson iteration scheme and a tolerance of 10^{-5}, were employed to perform the analysis.

As expected, the results of the poorly discretized model tend to move away from the analytical solution as the behavior of the structure leaves the region in which it can be considered linear. The 10-elements discretization, on the other hand, provides results very close to the analytical solution. It is also noticed that the results for different nonlinear formulations are numerically the same, and their curves are overlapping on the graphs.

The relative error of the 1-element discretization, between the average result of the different formulations and the analytical solution, at the last point, is 14.95% for the horizontal displacement and 16.15% for the vertical displacement. For the 10-elements discretization, this error is 0.36% for the horizontal displacement and 0.98% for the vertical displacement. As commented, the consequence of a bad discretization of this model is a larger value of vertical displacements.

The internal force diagrams corresponding to the application of the total load are shown in Fig. 5.7 for both refinements. The diagrams are plotted over the initial configuration of the model to help visualization of the results. In the discretized model, it can be noted that the free end has a greater normal force and a smaller shear force than the fixed end because the applied force (constant vertical direction) acts almost in the axial direction of the beam in the final configuration. The bending moment at the fixed end is notably smaller than the result expected for a linear analysis because the lever arm of the applied load is smaller.

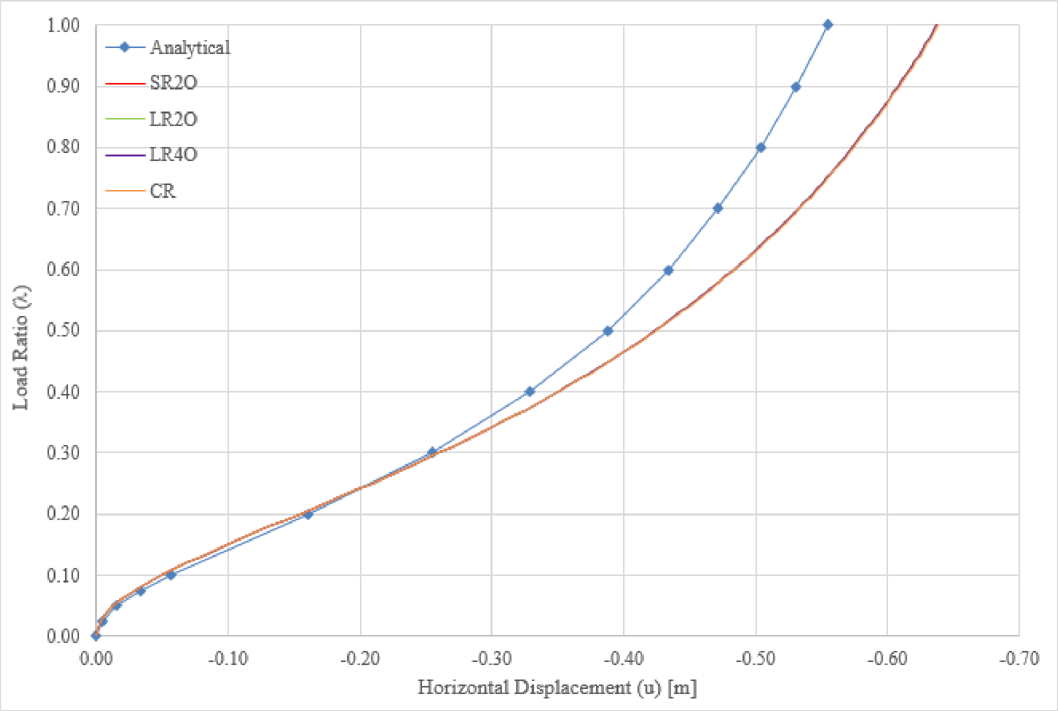


Figure 5.3 – Example 1: Horizontal displacement of the free end with a 1-element discretization

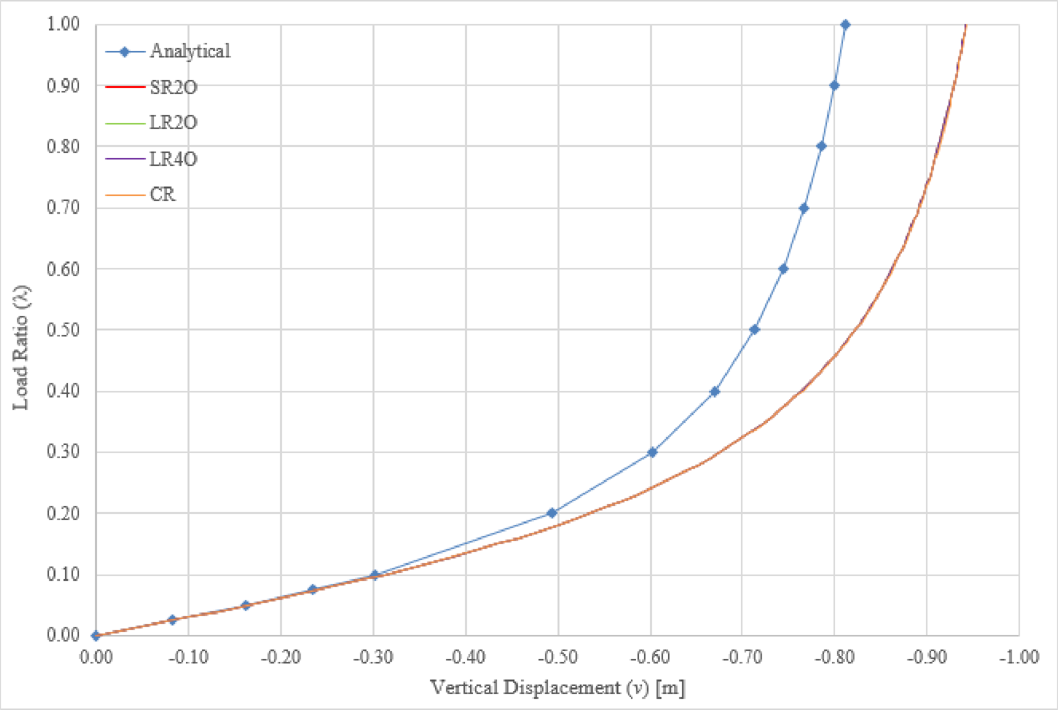


Figure 5.4 – Example 1: Vertical displacement of the free end with a 1-element discretization

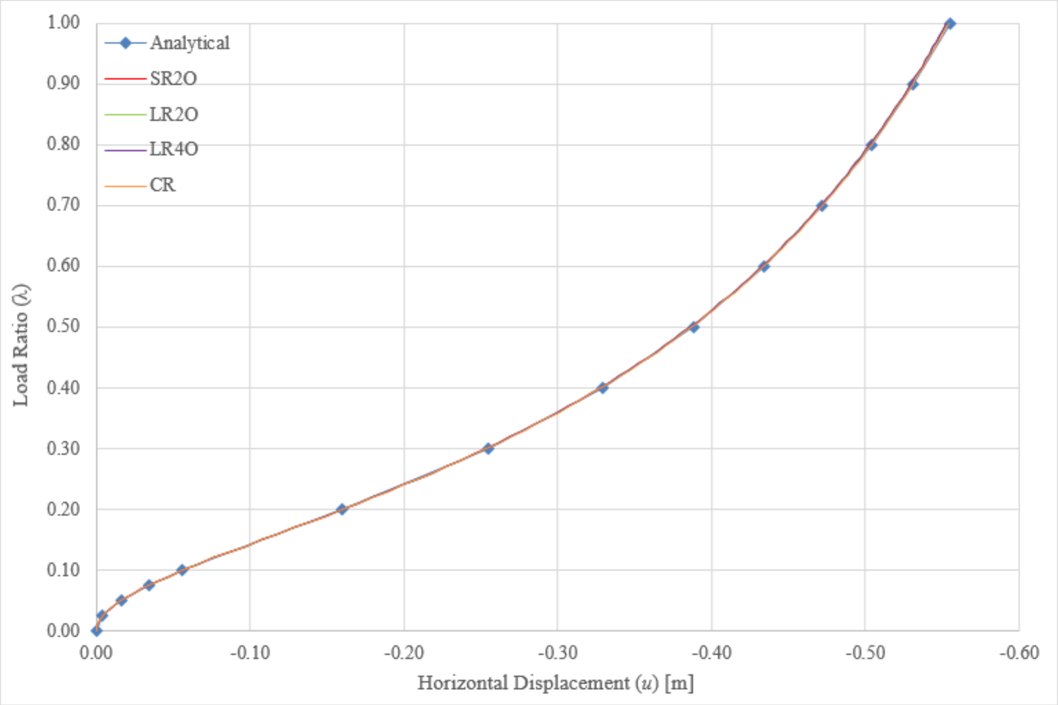


Figure 5.5 – Example 1: Horizontal displacement of the free end with a 10-elements discretization

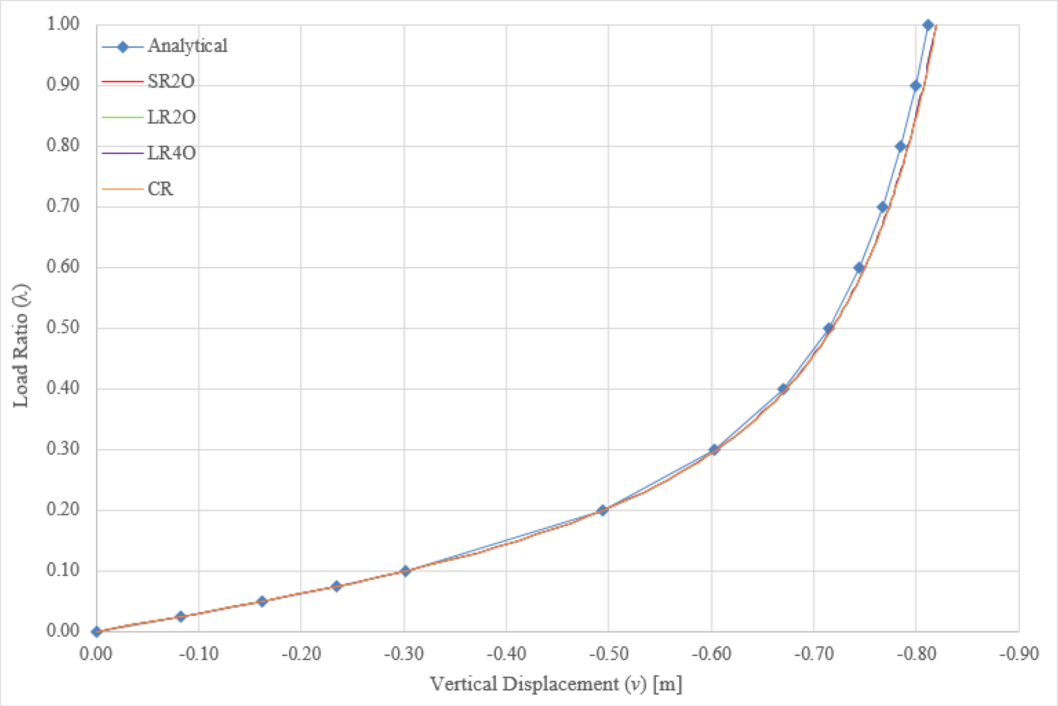


Figure 5.6 – Example 1: Vertical displacement of the free end with a 10-elements discretization

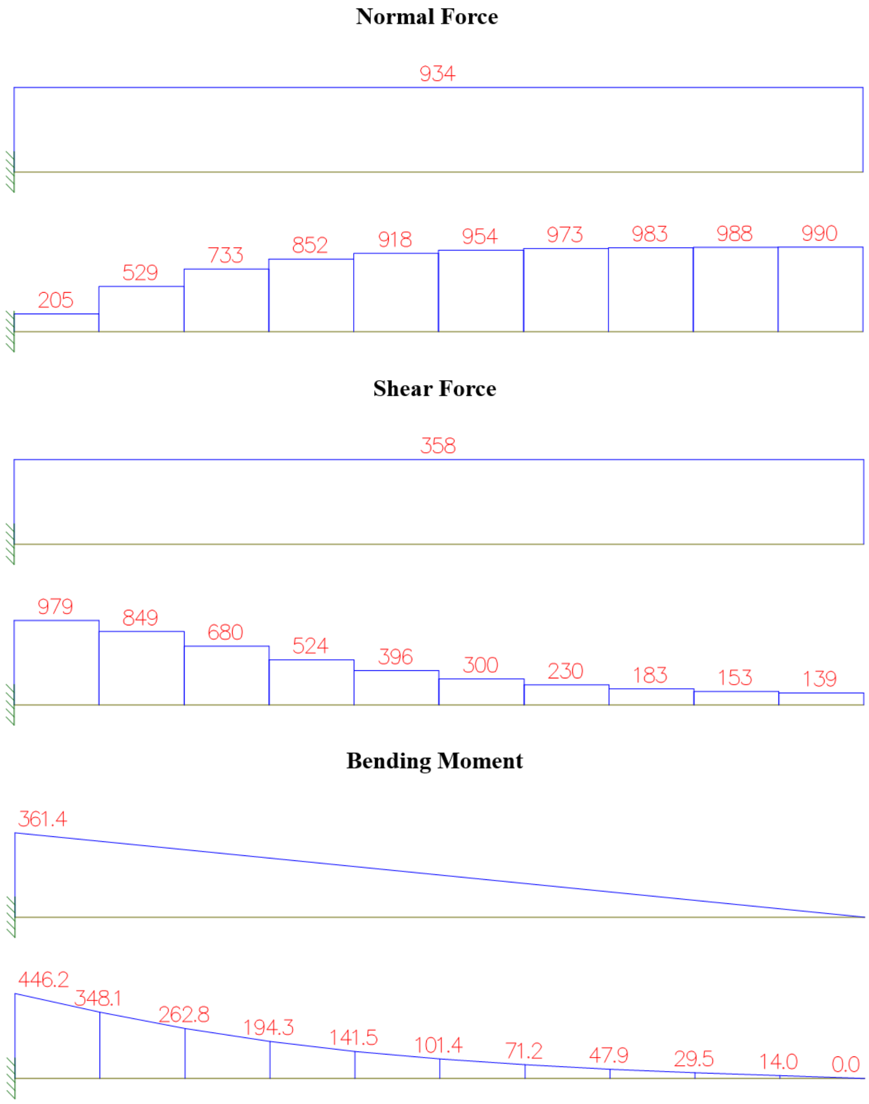


Figure 5.7 – Internal force diagrams of the cantilever beam with 1-element and 10-elements discretization

The study on the performance of the incremental-iterative methods and the nonlinear formulations, when using different analysis options to trace the equilibrium path, is presented in Table 5.2 to Table 5.5. The numerical parameters used in all analyses are the same: an initial increment of 0.01, a maximum number of iterations per step of 500, a tolerance for convergence of 10^{-5}, and a desired number of iteration of 3 (when the increment is adjusted). An equally spaced 10-elements discretization of the beam, which proved to be sufficient for obtaining a satisfactory result, is considered in all analyses. The total number of steps and corrective iterations to obtain full response with a particular analysis setting are provided in each table, as described in the introduction.

Table 5.2 – Example 1: Number of steps and iterations for constant increments with standard iteration scheme

	SR2O	LR2O	LR4O	CR	Mean
LCM	100 | 190 (1.90)	100 | 300 (3.00)	100 | 300 (3.00)	100 | 122 (1.22)	100 | 228 (2.28)
WCM	32 | 95 (2.97)	32 | 108 (3.38)	32 | 108 (3.38)	32 | 61 (1.91)	32 | 93 (2.91)
ALCM_FNP	52 | 133 (2.56)	54 | 162 (3.00)	54 | 162 (3.00)	52 | 94 (1.81)	53 | 114 (2.15)
ALCM_UNP	52 | 133 (2.56)	54 | 162 (3.00)	54 | 162 (3.00)	52 | 94 (1.81)	53 | 114 (2.15)
ALCM_CYL	32 | 95 (2.97)	32 | 107 (3.34)	32 | 107 (3.34)	32 | 42 (1.31)	32 | 88 (2.75)
ALCM_SPH	100 | 190 (1.90)	100 | 300 (3.00)	100 | 300 (3.00)	100 | 122 (1.22)	100 | 228 (2.28)
MNCM	32 | 95 (2.97)	32 | 107 (3.34)	32 | 107 (3.34)	32 | 42 (1.31)	32 | 88 (2.75)
ORCM	X	X	X	X	X
GDCM	33 | 98 (2.97)	33 | 110 (3.33)	33 | 110 (3.33)	33 | 42 (1.27)	33 | 90 (2.73)
Mean	54 | 129 (2.39)	55 | 170 (3.09)	55 | 170 (3.09)	54 | 77 (1.43)	54 | 136 (2.52)

Table 5.3 – Example 1: Number of steps and iterations for constant increments with modified iteration scheme

	SR2O	LR2O	LR4O	CR	Mean
LCM	X	X	X	X	X
WCM	32 | 256 (8.00)	32 | 272 (8.50)	32 | 272 (8.50)	32 | 270 (8.44)	32 | 268 (8.38)
ALCM_FNP	X	X	X	X	X
ALCM_UNP	X	X	X	X	X
ALCM_CYL	32 | 239 (7.47)	32 | 245 (7.66)	32 | 245 (7.66)	32 | 245 (7.66)	32 | 244 (7.63)
ALCM_SPH	X	X	X	X	X
MNCM	32 | 239 (7.47)	32 | 245 (7.66)	32 | 245 (7.66)	32 | 244 (7.63)	32 | 243 (7.59)
ORCM	27 | 342 (12.67)	27 | 353 (13.07)	27 | 353 (13.07)	27 | 365 (13.52)	27 | 353 (13.07)
GDCM	33 | 236 (7.15)	33 | 243 (7.36)	33 | 243 (7.36)	33 | 241 (7.30)	33 | 241 (7.30)
Mean	31 | 242 (7.81)	31 | 252 (8.13)	31 | 252 (8.13)	31 | 273 (8.81)	31 | 276 (8.90)

Table 5.4 – Example 1: Number of steps and iterations for adjusted increments with standard iteration scheme

	SR2O	LR2O	LR4O	CR	Mean
LCM	51 | 140 (2.75)	100 | 300 (3.00)	100 | 300 (3.00)	30 | 71 (2.37)	70 | 203 (2.90)
WCM	32 | 95 (2.97)	38 | 122 (3.21)	38 | 122 (3.21)	13 | 32 (2.38)	30 | 93 (3.10)
ALCM_FNP	47 | 137 (2.91)	54 | 162 (3.00)	54 | 162 (3.00)	27 | 71 (2.63)	46 | 133 (2.89)
ALCM_UNP	47 | 137 (2.91)	54 | 162 (3.00)	54 | 162 (3.00)	27 | 71 (2.63)	46 | 133 (2.89)
ALCM_CYL	32 | 95 (2.97)	38 | 122 (3.21)	38 | 122 (3.21)	13 | 30 (2.31)	30 | 92 (3.07)
ALCM_SPH	51 | 140 (2.75)	100 | 300 (3.00)	100 | 300 (3.00)	30 | 71 (2.37)	70 | 203 (2.90)
MNCM	32 | 95 (2.97)	38 | 122 (3.21)	38 | 122 (3.21)	13 | 30 (2.31)	30 | 92 (3.07)
ORCM	X	X	X	X	X
GDCM	33 | 98 (2.97)	33 | 110 (3.33)	33 | 110 (3.33)	33 | 42 (1.86)	33 | 90 (2.73)
Mean	41 | 117 (2.85)	57 | 175 (3.07)	57 | 175 (3.07)	23 | 52 (2.26)	44 | 130 (2.95)

Table 5.5 – Example 1: Number of steps and iterations for adjusted increments with modified iteration scheme

	SR2O	LR2O	LR4O	CR	Mean
LCM	X	X	X	X	X
WCM	74 | 232 (3.14)	72 | 226 (3.14)	72 | 226 (3.14)	73 | 228 (3.12)	73 | 228 (3.12)
ALCM_FNP	X	X	X	X	X
ALCM_UNP	X	X	X	X	X
ALCM_CYL	69 | 215 (3.12)	69 | 216 (3.13)	69 | 216 (3.13)	69 | 215 (3.12)	69 | 216 (3.13)
ALCM_SPH	X	X	X	X	X
MNCM	69 | 215 (3.12)	69 | 216 (3.13)	69 | 216 (3.13)	69 | 215 (3.12)	69 | 216 (3.13)
ORCM	78 | 248 (3.18)	77 | 245 (3.18)	77 | 245 (3.18)	77 | 245 (3.18)	77 | 246 (3.19)
GDCM	33 | 236 (7.15)	33 | 243 (7.36)	33 | 243 (7.36)	33 | 241 (7.30)	33 | 241 (7.30)
Mean	65 | 229 (3.52)	64 | 229 (3.58)	64 | 229 (3.58)	64 | 229 (3.58)	64 | 229 (3.58)

In all successfully performed analyses, the numerical results obtained with different settings were equivalent, i.e., the equilibrium paths for distinct methods, formulations, increment and iteration types are all overlapped. However, not all of these analysis settings were able to capture full solution, and performance (number of steps and iterations) also changes between them.

In the first tests, using constant increments and standard iterations, the ORCM did not work well. In this case, the iterative corrections caused the load ratio to explode to a very high value during the fourth step of the analysis. The other methods had similar performances, when considering the average number of iterations per step. This average has a greater variation for the results from different formulations. The CR formulation required the lowest number of iterations to complete the analysis, being is the most efficient for this case.

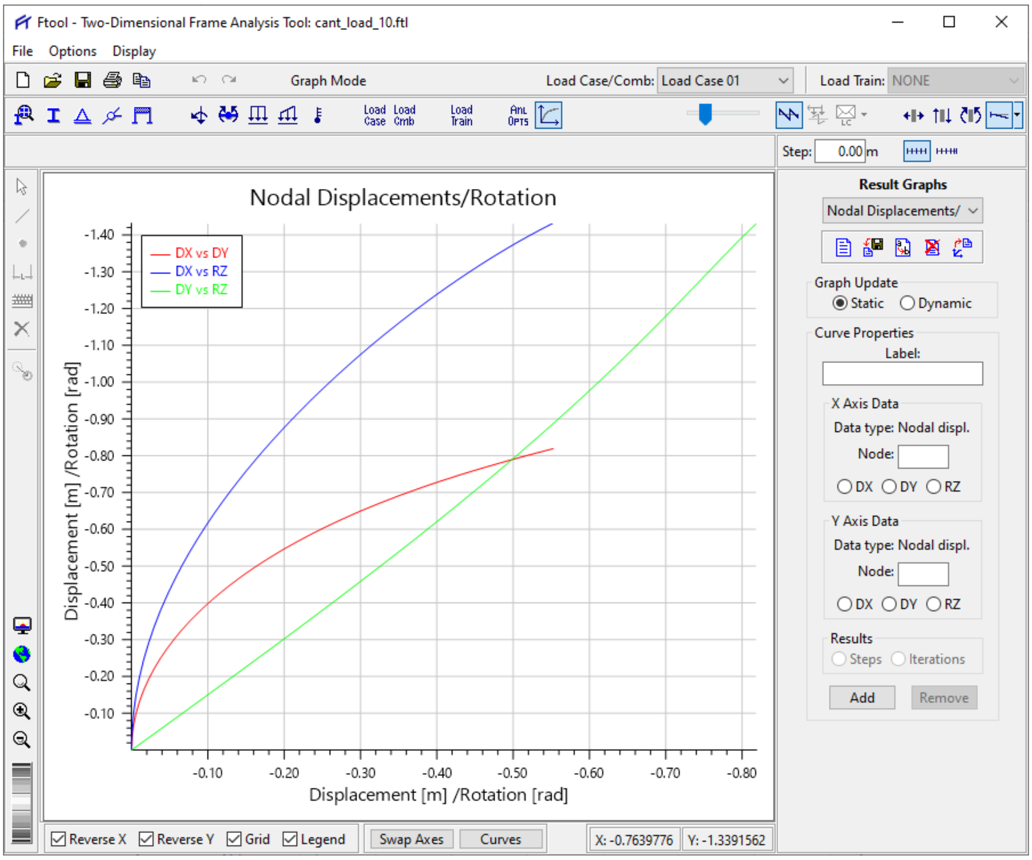
Using constant increments but with the modified iteration scheme, the LCM, ALCM_FNP, ALCM_UNP, and ALCM_SPH were not able to solve the problem. These methods could not reach convergence in the first step. On the other hand, the ORCM, which failed with the standard scheme, was able to solve the problem, but with a higher number of iterations per step than the other methods. The CR formulation, which provided the most efficient results for the standard iteration scheme, cannot be considered the most advantageous for this case. Furthermore, it is observed that, for all the successfully performed analyses, the number of steps is the same of the previous table, but more iterations are executed. This is an expected result of not updating the tangent stiffness matrix in all iterations.

In the tests with adjusted increments and standard iterations (Table 5.4), it is noted that the number of steps decreased compared to Table 5.2, for the analyzes that had an average number of iterations of less than 3 (desired number). The reason is that to adjust each step to perform the desired number of iterations, the size of the steps had to be larger. The analyzes with an average number of iterations greater than 3 (when constant steps were used) had their step size decreased, and consequently more steps were performed. Again, the ORCM did not work well, and the CR formulation was the most efficient. However, in some cases the number of steps performed by the CR formulation was very small, leading to a less smooth curve for the equilibrium path.

The last tests, combining adjusted increments with modified iterations, show that the LCM, ALCM_FNP, ALCM_UNP, and ALCM_SPH keep failing to solve the problem. The general number of steps is higher than in the previous table because smaller increments are necessary to hold the number of modified iterations to the desired value. It is interesting to note that the performance of the methods and formulations became more homogeneous. The only discrepancy is the GDCM, which uses a different parameter to adjust the increment size that is not based on the number of iterations (it uses the GSP parameter).

Some concluding remarks about this model is that the ORCM does not work well when the standard iteration scheme is used. Several other methods are not able to solve the problem when the modified iteration scheme is employed. As previously mentioned in this work, for increasing stiffness systems, the modified iteration scheme is not good (McGuire *et al.*, 2000), which explains why more methods have failed with the modified iterations. The CR formulation is more efficient than the other formulations, when using the standard iterations. This is not true for the modified iteration scheme. It should be noted that, in all analyses, the performance of the ALCM_FNP and ALCM_UNP were the same, as well as the performance of the LR2O and LR4O formulations.

Some other interesting results that can be obtained from the new version of the Ftool program are shown next. These results were obtained for a 10-elements discretization, analyzed with the LCM, a constant increment of 0.01, and the standard iteration scheme. Fig. 5.8 shows the graph-plotting environment with a chart of the relation between the horizontal and vertical displacements (red), horizontal displacement and rotation (blue), and vertical displacement and rotation (green) of the free end of the beam. Figures 5.9 and 5.10 provide the values of the displacements and rotation of the beam tip as well as the increment of these quantities for each analysis step. Notably, the curves of Fig. 5.9 represent the tangent of the curves of Fig. 5.10. Finally, in Fig. 5.11, a portion of the equilibrium path, with three converged equilibrium configurations, is shown with the results for each iteration of the LCM. Clearly, a load increment is given at each step and kept constant at each iteration.



Ftool - Two-Dimensional Frame Analysis Tool: cant_load_10.ftl

Load Case/Comb: Load Case 01

Load Train: NONE

Step: 0.00 m

Result Graphs: Nodal Displacements/

Graph Update: ● Static ○ Dynamic

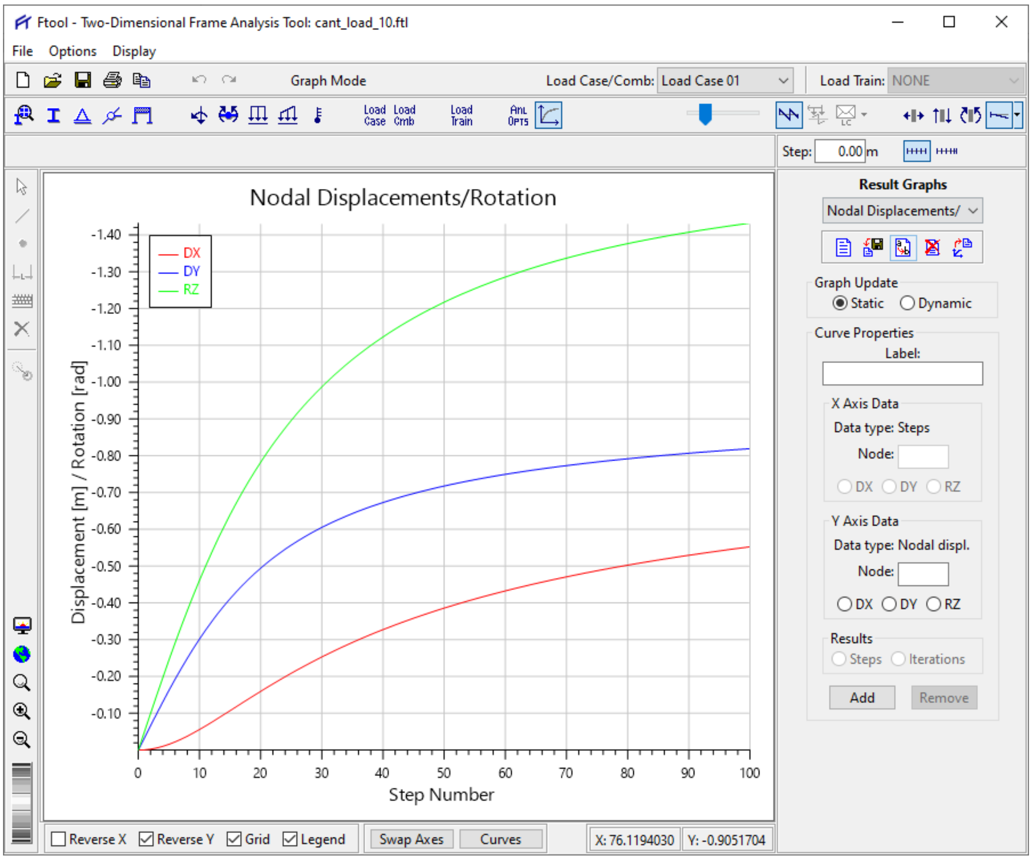
X Axis Data — Data type: Nodal displ. Node: ○ DX ○ DY ○ RZ

Y Axis Data — Data type: Nodal displ. Node: ○ DX ○ DY ○ RZ

☑ Reverse X ☑ Reverse Y ☑ Grid ☑ Legend

X: -0.7639776 Y: -1.3391562

Figure 5.8 – Relation between nodal displacements and rotation



Ftool - Two-Dimensional Frame Analysis Tool: cant_load_10.ftl

Load Case/Comb: Load Case 01

Load Train: NONE

Step: 0.00 m

Result Graphs: Nodal Displacements/

Graph Update: ● Static ○ Dynamic

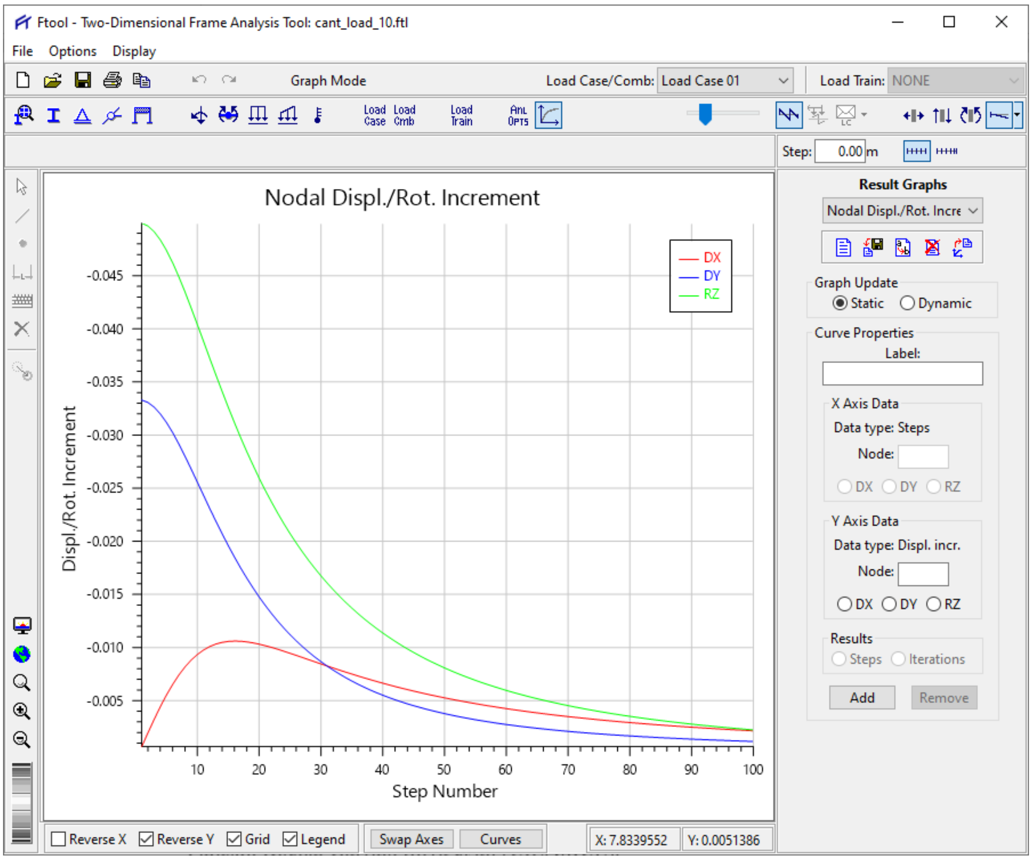
X Axis Data — Data type: Steps Node: ○ DX ○ DY ○ RZ

Y Axis Data — Data type: Nodal displ. Node: ○ DX ○ DY ○ RZ

□ Reverse X ☑ Reverse Y ☑ Grid ☑ Legend

X: 76.1194030 Y: -0.9051704

Figure 5.9 –Nodal displacements and rotation for each step



Load Case/Comb: Load Case 01

Load Train: NONE

Step: 0.00 m

Result Graphs: Nodal Displ./Rot. Incre

Graph Update: ● Static ○ Dynamic

Label:

X Axis Data
Data type: Steps
Node:
○ DX ○ DY ○ RZ

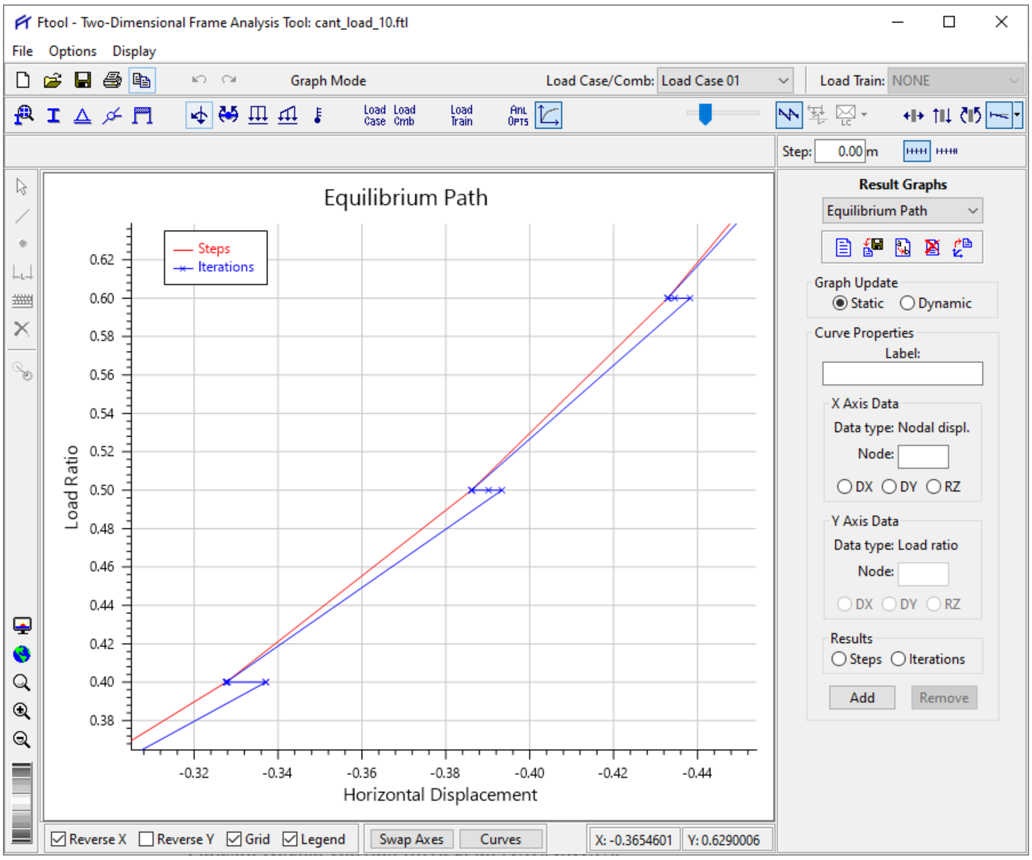
Y Axis Data
Data type: Displ. incr.
Node:
○ DX ○ DY ○ RZ

Results
○ Steps ○ Iterations

□ Reverse X ☑ Reverse Y ☑ Grid ☑ Legend

X: 7.8339552 Y: 0.0051386

Figure 5.10 –Increments of nodal displacements and rotation for each step



Load Case/Comb: Load Case 01

Load Train: NONE

Step: 0.00 m

Result Graphs: Equilibrium Path

Graph Update: ● Static ○ Dynamic

Label:

X Axis Data
Data type: Nodal displ.
Node:
○ DX ○ DY ○ RZ

Y Axis Data
Data type: Load ratio
Node:
○ DX ○ DY ○ RZ

Results
○ Steps ○ Iterations

☑ Reverse X □ Reverse Y ☑ Grid ☑ Legend

X: -0.3654601 Y: 0.6290006

Figure 5.11 –Equilibrium path showing incremental and iterative results

5.3. Cantilever Beam with End Moment

The second example deals with the same model of the previous section. However, instead of a vertical load, a concentrated bending moment, *M*, is applied at the tip of the cantilever beam. The beam is illustrated in Fig. 5.12, with its physical and geometric properties provided and the displacements and rotation of the free end indicated.

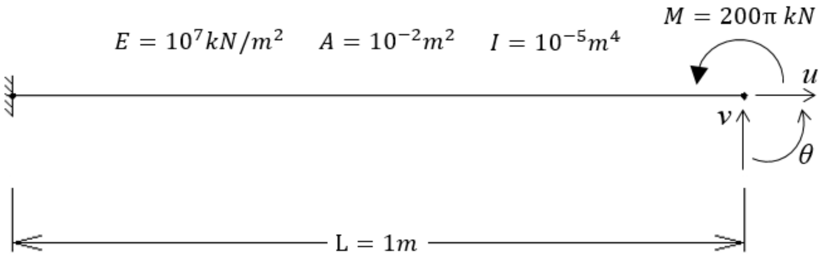


Figure 5.12 – Cantilever beam with end moment

This problem has been analyzed by a number of researchers in order to test new nonlinear formulations and solution methods under extreme bending. The exact solution for the deformed shape of this model is a perfect circle, since the bending moment, and hence the curvature, is constant along the beam. The analytical solution for this problem is given in Eq. (5.1) to Eq. (5.3) (Almeida *et al.*, 2011). When the bending moment reaches the value of $M = 2\pi(EI/L)$, the beam is rolled up into a circle.

$$\theta = \frac{ML}{EI} \tag{5.1}$$

$$u = L\left(1 - \frac{\sin\theta}{\theta}\right) \tag{5.2}$$

$$v = L\left(\frac{1-\cos\theta}{\theta}\right) \tag{5.3}$$

The objective of this example is to illustrate the ability of the nonlinear formulations to handle very large rotations, as long as the structural members are subdivided into small elements. Figure 5.13 illustrates the cantilever beam discretized with 1 and 10 elements, subjected to twice the moment necessary to roll it according to the analytical solution. Clearly, a good discretization is fundamental to simulate the nonlinear behavior.

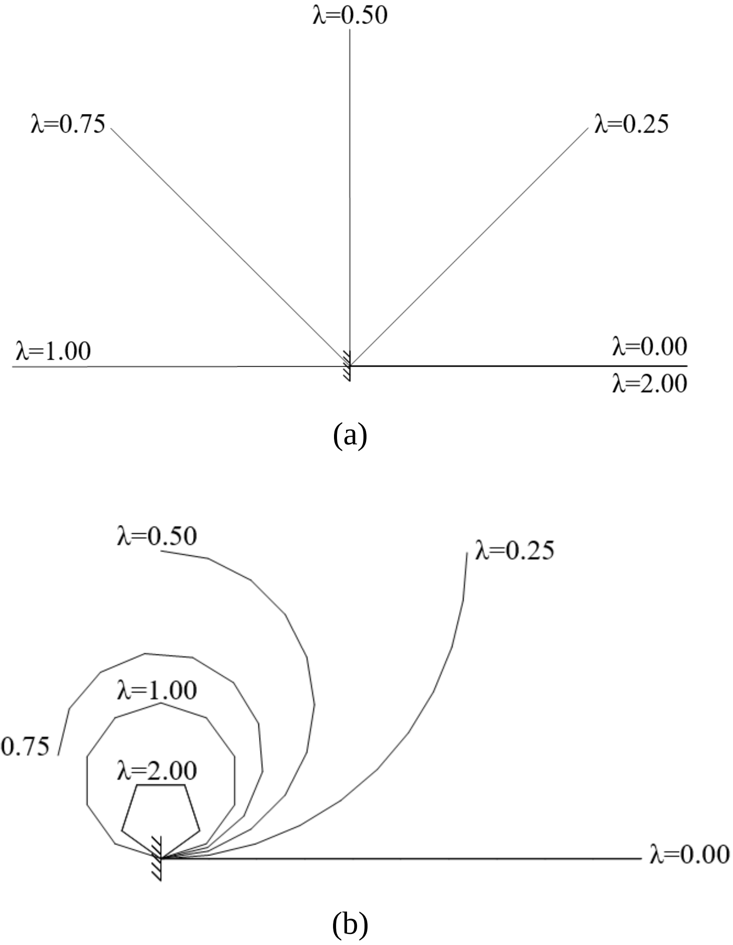


Figure 5.13 – Example 2: Deformed configurations of the cantilever beam with (a) 1-element and (b) 10-elements discretization

The geometrically nonlinear behavior of this model has no load limit point. Therefore, using the LCM, the equilibrium paths for the horizontal and vertical displacements at the tip of the beam, discretized with 10 elements, are shown in Fig. 5.14 and Fig. 5.15, respectively. In each graph, the results obtained with different formulations are compared with good agreement to the analytical solution. Again, it is checked that the nonlinear formulations are numerically equivalent for this type of problem, and a discretization of the beam into 10 elements is sufficiently fine. However, efficiency is not the same for each formulation.

In the sequence, Table 5.6 to Table 5.9 bring a study on the performance of the incremental-iterative methods and the formulations to solve this problem when different types of increment and iteration schemes are considered. An initial increment of 0.01, a maximum number of iterations per step of 500, a tolerance for convergence of 10^{-5}, and a desired number of iteration of 3 (when the increment is adjusted) are employed for all analyzes.

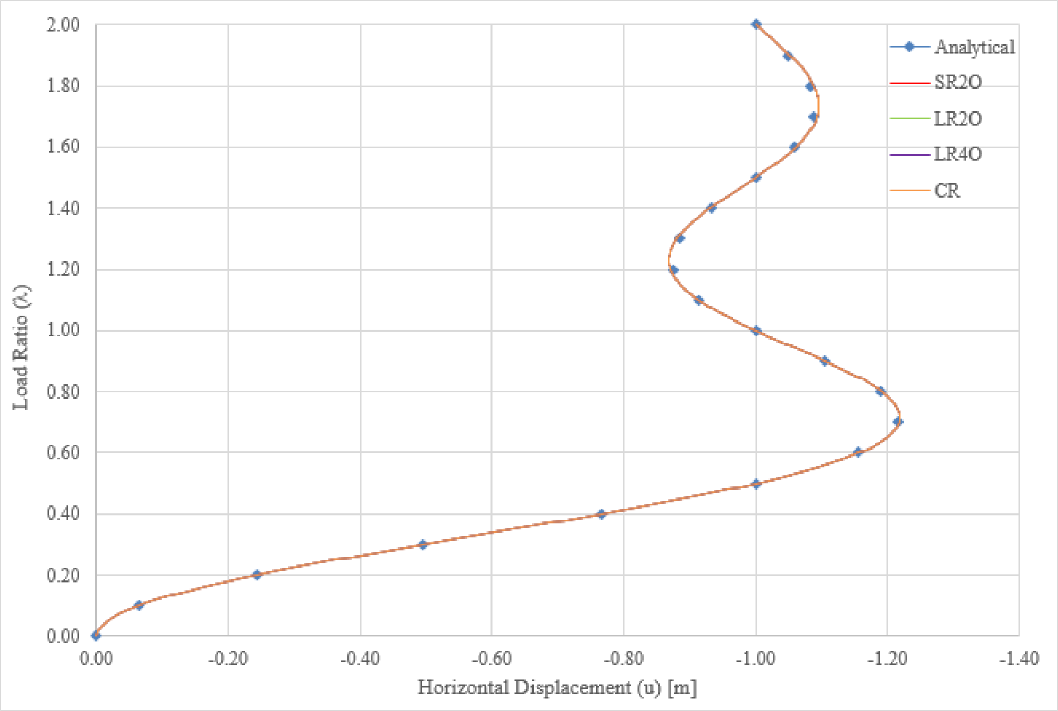


Figure 5.14 – Example 2: Horizontal displacement of the free end with a 10-elements discretization

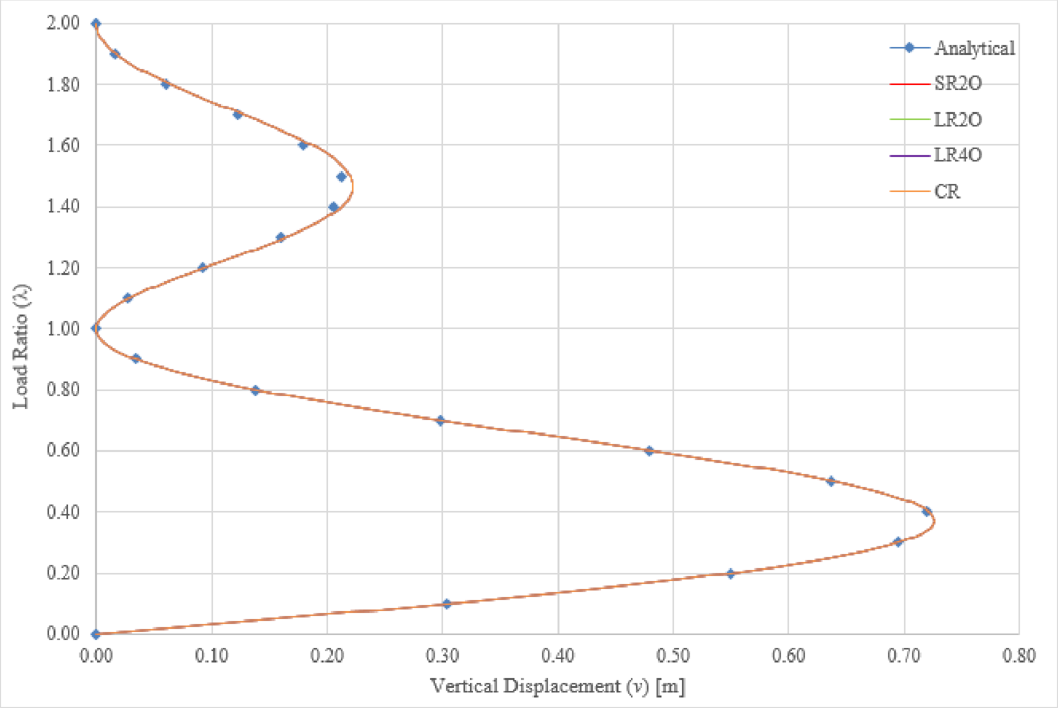


Figure 5.15 – Example 2: Vertical displacement of the free end with a 10-elements discretization

Table 5.6 – Example 2: Number of steps and iterations for constant increments with standard iteration scheme

	SR2O	LR2O	LR4O	CR	Mean
LCM	100 | 300 (3.00)	100 | 490 (4.90)	100 | 490 (4.90)	100 | 200 (2.00)	100 | 370 (3.70)
WCM	100 | 300 (3.00)	88 | 412 (4.68)	88 | 412 (4.68)	100 | 200 (2.00)	94 | 331 (3.52)
ALCM_FNP	99 | 296 (2.99)	X	X	99 | 197 (2.99)	99 | 247 (2.49)
ALCM_UNP	99 | 296 (2.99)	X	X	99 | 197 (2.99)	99 | 247 (2.49)
ALCM_CYL	98 | 292 (2.98)	98 | 473 (4.83)	98 | 473 (4.83)	98 | 195 (1.99)	98 | 358 (3.65)
ALCM_SPH	100 | 300 (3.00)	100 | 490 (4.90)	100 | 490 (4.90)	100 | 200 (2.00)	100 | 370 (3.70)
MNCM	98 | 294 (3.00)	99 | 479 (4.84)	99 | 479 (4.84)	98 | 196 (2.00)	99 | 362 (3.66)
ORCM	X	X	X	X	X
GDCM	98 | 292 (2.98)	98 | 473 (4.83)	98 | 473 (4.83)	98 | 195 (1.99)	98 | 358 (3.65)
Mean	99 | 296 (2.99)	97 | 470 (4.85)	97 | 470 (4.85)	99 | 198 (2.00)	98 | 342 (3.49)

Table 5.7 – Example 2: Number of steps and iterations for constant increments with modified iteration scheme

	SR2O	LR2O	LR4O	CR	Mean
LCM	X	X	X	X	X
WCM	100 | 1101 (11.01)	X	X	100 | 1101 (11.01)	100 | 1101 (11.01)
ALCM_FNP	X	X	X	X	X
ALCM_UNP	X	X	X	X	X
ALCM_CYL	98 | 682 (6.96)	98 | 824 (8.41)	98 | 824 (8.41)	98 | 682 (6.96)	98 | 753 (7.68)
ALCM_SPH	X	X	X	X	X
MNCM	98 | 682 (6.96)	98 | 821 (8.38)	98 | 821 (8.38)	98 | 682 (6.96)	98 | 752 (7.67)
ORCM	X	X	X	X	X
GDCM	98 | 683 (6.97)	98 | 821 (8.38)	98 | 821 (8.38)	98 | 683 (6.97)	98 | 752 (7.67)
Mean	99 | 787 (7.95)	98 | 822 (8.39)	98 | 822 (8.39)	99 | 787 (7.95)	98 | 802 (8.18)

Table 5.8 – Example 2: Number of steps and iterations for adjusted increments with standard iteration scheme

	SR2O	LR2O	LR4O	CR	Mean
LCM	100 | 300 (3.00)	1429 | 4329 (3.03)	1429 | 4329 (3.03)	29 | 76 (2.62)	747 | 2259 (3.02)
WCM	100 | 300 (3.00)	385 | 1218 (3.16)	385 | 1218 (3.16)	27 | 69 (2.56)	224 | 701 (3.13)
ALCM_FNP	99 | 296 (2.99)	X	X	29 | 76 (2.62)	64 | 186 (2.91)
ALCM_UNP	99 | 296 (2.99)	X	X	29 | 76 (2.62)	64 | 186 (2.91)
ALCM_CYL	98 | 292 (2.98)	420 | 1284 (3.06)	420 | 1284 (3.06)	26 | 66 (2.54)	241 | 732 (3.04)
ALCM_SPH	100 | 300 (3.00)	1428 | 4326 (3.03)	1428 | 4326 (3.03)	29 | 76 (2.62)	746 | 2257 (3.03)
MNCM	98 | 294 (3.00)	420 | 1283 (3.05)	420 | 1283 (3.05)	27 | 69 (2.56)	241 | 732 (3.04)
ORCM	X	X	X	X	X
GDCM	98 | 292 (2.98)	98 | 473 (4.83)	98 | 473 (4.83)	98 | 195 (1.99)	98 | 358 (3.65)
Mean	99 | 296 (2.99)	696 | 2152 (3.09)	696 | 2152 (3.09)	37 | 88 (2.38)	337 | 1032 (3.06)

Table 5.9 – Example 2: Number of steps and iterations for adjusted increments with modified iteration scheme

	SR2O	LR2O	LR4O	CR	Mean
LCM	X	X	X	X	X
WCM	298 | 908 (3.05)	461 | 1440 (3.12)	461 | 1440 (3.12)	298 | 908 (3.05)	380 | 1174 (3.09)
ALCM_FNP	X	X	X	X	X
ALCM_UNP	X	X	X	X	X
ALCM_CYL	255 | 775 (3.04)	388 | 1184 (3.05)	388 | 1184 (3.05)	255 | 775 (3.04)	322 | 980 (3.04)
ALCM_SPH	X	X	X	X	X
MNCM	255 | 775 (3.04)	388 | 1184 (3.05)	388 | 1184 (3.05)	255 | 775 (3.04)	322 | 980 (3.04)
ORCM	359 | 1101 (3.07)	488 | 1493 (3.06)	488 | 1493 (3.06)	361 | 1108 (3.07)	424 | 1112 (2.62)
GDCM	98 | 683 (6.97)	98 | 821 (8.38)	98 | 821 (8.38)	98 | 683 (6.97)	98 | 752 (7.67)
Mean	253 | 848 (3.35)	365 | 1224 (3.35)	365 | 1224 (3.35)	253 | 850 (3.36)	309 | 1037 (3.36)

Using constant increments and standard iterations, the ORCM failed to obtain the solution because of the same problem presented in the previous example: the iterative corrections cause the load ratio to explode to a very high value during the fourth step. The ALCM_FNP and ALCM_UNP, using the formulations LR2O and LR4O, were able to follow the solution up to a load ratio of 0.91, but were stuck at this point. The LR2O and LR4O formulations also needed more iterations per step to complete the analysis when other methods were used. The CR formulation is the most efficient for this case because it required the lowest number of iterations in each step.

By changing the iteration scheme to the modified version, only the ALCM_CYL, MNCM, and GDCM were able to capture full solution using all formulations. The WCM solved the problem with the SR2O and CR formulations, but with a higher number of iterations than other methods. As expected, the number of iterations of all analyses was increased compared to the tests with the standard iteration scheme.

Considering an adjusted increment, it can be observed that the analyzes with the LR2O and LR4O formulations are less efficient, since the step sizes to honor the desired number of iterations need to be very small. The CR formulation turns to be the best choice in this case, as it requires much fewer steps and iterations for each solution method. The same problems experienced by the ALCM_FNP, ALCM_UNP, and ORCM in the tests of Table 5.6, also occur with the adjusted increment.

When the adjusted increment and the modified iteration scheme are employed, the WCM can be used with the LR2O and LR4O formulations, which is not possible when the modified iterations are performed with constant increments. Furthermore, the ORCM can be used without problems. However, these analysis settings provide a larger number of steps and total iterations when compared to the previous tests.

In all analyzes of this model, the same performance was registered by the ALCM_FNP and ALCM_UNP methods and by the LR2O and LR4O formulations. These formulations, however, presented more convergence problems and required more iterations to reach the final solution. The ALCM_CYL, MNCM, and GDCM were the only methods that did not have any trouble to solve the problem with any nonlinear formulation.

5.4. Williams Toggle Frame

The Williams Toggle Frame is a low-rise arch composed of two beams with clamped ends and a vertical concentrated load applied at the apex. Figure 5.16 illustrates the out-of-scale model for easy viewing of dimensions. The arch has a span of 65.715 cm and a height of 0.98 cm. The material of the beams has a modulus of elasticity of E = 199714 MPa, the cross-section is circular with a diameter of 0.721 cm, and the magnitude of the applied load is P = 0.25 kN. The positive direction of the vertical displacement of the apex, v, which is the only effective degree-of-freedom, is also indicated. Euler-Bernoulli bending behavior is assumed.

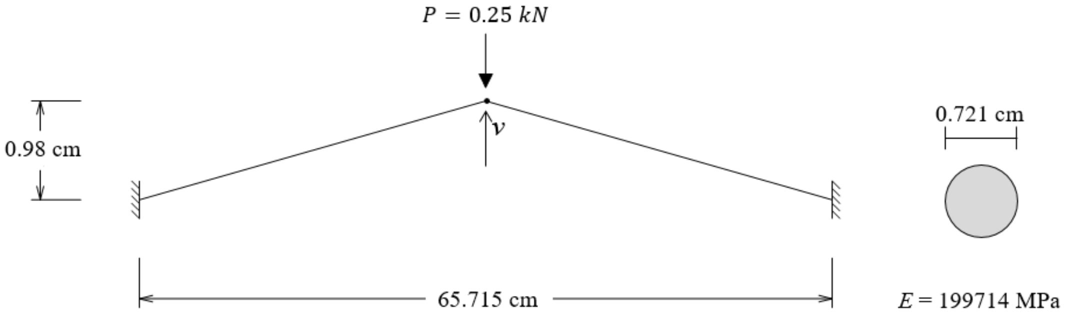


Figure 5.16 – Williams Toggle Frame

This model was first investigated by Williams (1964), who treated the frame both analytically and experimentally taking into account finite changes in geometry. Despite being a simple model, it has a highly nonlinear response. Depending on its dimensions, it exhibits load limit points with snap-through behavior. Therefore, this example has the intention to demonstrate the ability of the implemented methods to capture this type of nonlinear behavior.

Figure 5.17 shows the equilibrium path of the Williams Toggle Frame, considering a discretization of each beam into 10 equal length elements. The results of the MASTAN program (McGuire *et al.*, 2000) are compared to the results of the Ftool program. The continuation method available in the MASTAN program is the WCM, which is able to capture the solution beyond load limit points and matches with good accuracy the results provided by the Ftool program using the same method. It is also shown the equilibrium path obtained with the Ftool program using the LCM. As expected, this method cannot capture full solution. When the load ratio exceeds the first load limit point, the method will either diverge or snap through the unstable behavior to find another equilibrium configuration corresponding to the new load level, as illustrated.

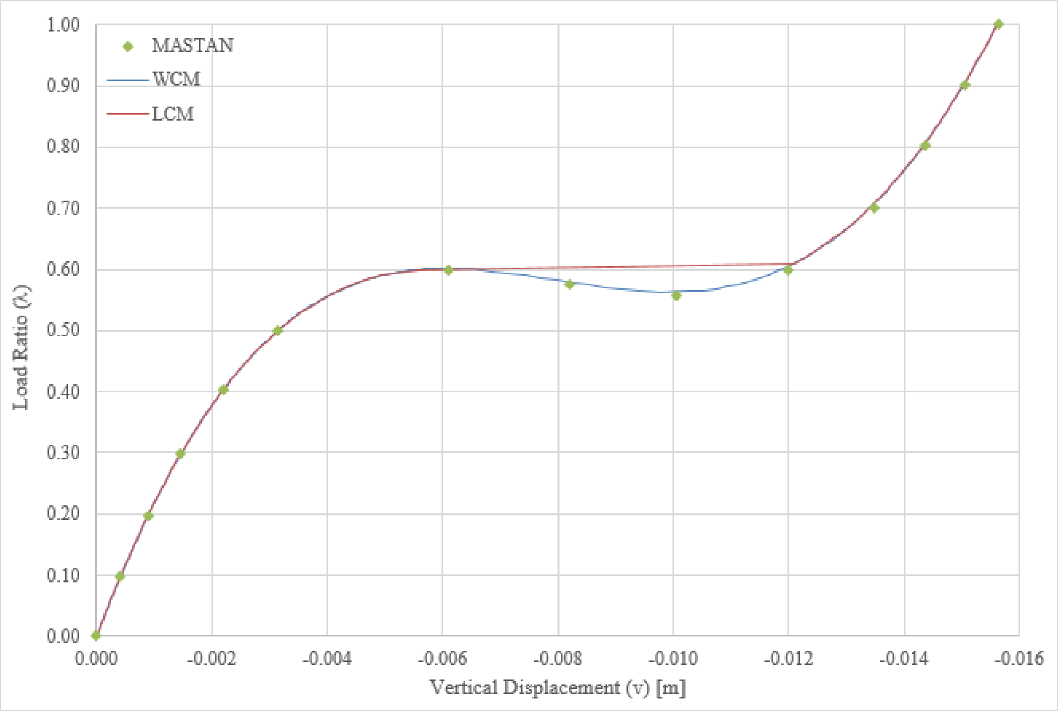


Figure 5.17 – Equilibrium path of the Williams Toggle Frame

As in the previous examples, the results obtained for the equilibrium configurations, using different methods, formulations, and analysis options, are numerically the same in all the analyses that successfully trace the equilibrium path. The difference lies only on the number of steps and iterations.

The performance evaluation of the solution methods and geometrically nonlinear formulations implemented in the Ftool program is found in Table 5.10 to Table 5.13. The numerical parameters for all analyses are the same as those used in the previous examples: initial increment of 0.01, maximum number of iterations per step of 500, tolerance for convergence of 10^{-5}, and desired number of iteration of 3 (when the increment is adjusted). In the following tables, an asterisk indicates when the solution could be obtained up to the total applied load, but snapping through the unstable region of the equilibrium path.

Table 5.10 – Example 3: Number of steps and iterations for constant increments with standard iteration scheme

	SR2O	LR2O	LR4O	CR	Mean
LCM	100 | 157 (1.57)*	100 | 127 (1.27)*	100 | 127 (1.27)*	100 | 133 (1.33)*	100 | 136 (1.36)
WCM	X	214 | 215 (1.00)	188 | 189 (1.01)	162 | 162 (1.00)	188 | 189 (1.00)
ALCM_FNP	291 | 290 (1.00)	293 | 293 (1.00)	293 | 293 (1.00)	237 | 237 (1.00)	279 | 278 (1.00)
ALCM_UNP	291 | 290 (1.00)	293 | 293 (1.00)	293 | 293 (1.00)	237 | 237 (1.00)	279 | 278 (1.00)
ALCM_CYL	389 | 389 (1.00)	389 | 389 (1.00)	389 | 389 (1.00)	389 | 389 (1.00)	389 | 389 (1.00)
ALCM_SPH	154 | 154 (1.00)	154 | 154 (1.00)	154 | 154 (1.00)	154 | 154 (1.00)	154 | 154 (1.00)
MNCM	389 | 389 (1.00)	389 | 389 (1.00)	389 | 389 (1.00)	389 | 389 (1.00)	389 | 389 (1.00)
ORCM	397 | 397 (1.00)	394 | 394 (1.00)	394 | 394 (1.00)	388 | 388 (1.00)	393 | 393 (1.00)
GDCM	388 | 388 (1.00)	388 | 388 (1.00)	388 | 388 (1.00)	388 | 388 (1.00)	388 | 388 (1.00)
Mean	300 | 307 (1.02)	290 | 294 (1.01)	287 | 291 (1.01)	272 | 275 (1.01)	287 | 291 (1.01)

Table 5.11 – Example 3: Number of steps and iterations for constant increments with modified iteration scheme

	SR2O	LR2O	LR4O	CR	Mean
LCM	X	X	X	X	X
WCM	X	225 | 262 (1.16)	208 | 262 (1.26)	149 | 202 (1.36)	194 | 242 (1.25)
ALCM_FNP	290 | 380 (1.31)	293 | 391 (1.33)	293 | 391 (1.33)	237 | 279 (1.18)	278 | 360 (1.29)
ALCM_UNP	290 | 380 (1.31)	293 | 391 (1.33)	293 | 391 (1.33)	237 | 279 (1.18)	278 | 360 (1.29)
ALCM_CYL	389 | 389 (1.00)	389 | 389 (1.00)	389 | 389 (1.00)	389 | 389 (1.00)	389 | 389 (1.00)
ALCM_SPH	154 | 298 (1.94)	154 | 299 (1.94)	154 | 299 (1.94)	154 | 294 (1.91)	154 | 298 (1.94)
MNCM	389 | 389 (1.00)	389 | 389 (1.00)	389 | 389 (1.00)	389 | 389 (1.00)	389 | 389 (1.00)
ORCM	400 | 400 (1.00)	400 | 400 (1.00)	400 | 400 (1.00)	388 | 388 (1.00)	397 | 397 (1.00)
GDCM	388 | 388 (1.00)	388 | 388 (1.00)	388 | 388 (1.00)	388 | 388 (1.00)	388 | 388 (1.00)
Mean	329 | 375 (1.14)	316 | 364 (1.15)	314 | 364 (1.16)	291 | 326 (1.12)	312 | 356 (1.14)

Table 5.12 – Example 3: Number of steps and iterations for adjusted increments with standard iteration scheme

<table>
<tr><th></th><th>SR2O</th><th>LR2O</th><th>LR4O</th><th>CR</th><th>Mean</th></tr>
<tr><td><b>LCM</b></td><td>21 | 150
(7.14)*</td><td>17 | 46
(2.71)*</td><td>17 | 46
(2.71)*</td><td>16 | 31
(1.94)*</td><td>18 | 68
(3.78)</td></tr>
<tr><td><b>WCM</b></td><td>X</td><td>20 | 38
(1.90)</td><td>20 | 38
(1.90)</td><td>15 | 23
(1.53)</td><td>18 | 33
(1.83)</td></tr>
<tr><td><b>ALCM_FNP</b></td><td>22 | 47
(2.14)</td><td>21 | 44
(2.10)</td><td>21 | 44
(2.10)</td><td>19 | 36
(1.89)</td><td>21 | 43
(2.05)</td></tr>
<tr><td><b>ALCM_UNP</b></td><td>22 | 47
(2.14)</td><td>21 | 44
(2.10)</td><td>21 | 44
(2.10)</td><td>19 | 35
(1.84)</td><td>21 | 43
(2.05)</td></tr>
<tr><td><b>ALCM_CYL</b></td><td>21 | 40
(1.90)</td><td>21 | 40
(1.90)</td><td>21 | 40
(1.90)</td><td>17 | 24
(1.41)</td><td>20 | 36
(1.80)</td></tr>
<tr><td><b>ALCM_SPH</b></td><td>20 | 44
(2.20)</td><td>19 | 40
(2.11)</td><td>19 | 40
(2.11)</td><td>18 | 35
(1.94)</td><td>19 | 40
(2.11)</td></tr>
<tr><td><b>MNCM</b></td><td>21 | 40
(1.90)</td><td>21 | 40
(1.90)</td><td>21 | 40
(1.90)</td><td>17 | 24
(1.41)</td><td>20 | 36
(1.80)</td></tr>
<tr><td><b>ORCM</b></td><td>34 | 59
(1.74)</td><td>22 | 42
(1.91)</td><td>22 | 42
(1.91)</td><td>X</td><td>26 | 48
(1.85)</td></tr>
<tr><td><b>GDCM</b></td><td>388 | 388
(1.00)</td><td>388 | 388
(1.00)</td><td>388 | 388
(1.00)</td><td>388 | 388
(1.00)</td><td>388 | 388
(1.00)</td></tr>
<tr><td><b>Mean</b></td><td>69 | 102
(1.48)</td><td>61 | 80
(1.31)</td><td>61 | 80
(1.31)</td><td>64 | 75
(1.17)</td><td>63 | 84
(1.33)</td></tr>
</table>

Table 5.13 – Example 3: Number of steps and iterations for adjusted increments with modified iteration scheme

<table>
<tr><th></th><th>SR2O</th><th>LR2O</th><th>LR4O</th><th>CR</th><th>Mean</th></tr>
<tr><td><b>LCM</b></td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr>
<tr><td><b>WCM</b></td><td>34 | 101
(2.97)</td><td>48 | 130
(2.71)</td><td>48 | 130
(2.71)</td><td>47 | 156
(3.32)</td><td>44 | 130
(2.95)</td></tr>
<tr><td><b>ALCM_FNP</b></td><td>75 | 215
(2.87)</td><td>77 | 222
(2.88)</td><td>77 | 222
(2.88)</td><td>73 | 210
(2.88)</td><td>76 | 217
(2.86)</td></tr>
<tr><td><b>ALCM_UNP</b></td><td>74 | 214
(2.89)</td><td>77 | 222
(2.88)</td><td>77 | 222
(2.88)</td><td>73 | 210
(2.88)</td><td>75 | 217
(2.89)</td></tr>
<tr><td><b>ALCM_CYL</b></td><td>62 | 172
(2.77)</td><td>63 | 174
(2.76)</td><td>63 | 174
(2.76)</td><td>62 | 171
(2.76)</td><td>63 | 173
(2.75)</td></tr>
<tr><td><b>ALCM_SPH</b></td><td>71 | 206
(2.90)</td><td>71 | 207
(2.92)</td><td>71 | 207
(2.92)</td><td>70 | 203
(2.90)</td><td>71 | 206
(2.90)</td></tr>
<tr><td><b>MNCM</b></td><td>62 | 172
(2.77)</td><td>63 | 174
(2.76)</td><td>63 | 174
(2.76)</td><td>62 | 171
(2.76)</td><td>63 | 173
(2.75)</td></tr>
<tr><td><b>ORCM</b></td><td>64 | 183
(2.86)</td><td>X</td><td>X</td><td>62 | 172
(2.77)</td><td>63 | 178
(2.83)</td></tr>
<tr><td><b>GDCM</b></td><td>388 | 388
(1.00)</td><td>388 | 388
(1.00)</td><td>388 | 388
(1.00)</td><td>388 | 388
(1.00)</td><td>388 | 388
(1.00)</td></tr>
<tr><td><b>Mean</b></td><td>104 | 206
(1.98)</td><td>112 | 217
(1.94)</td><td>112 | 217
(1.94)</td><td>105 | 210
(2.00)</td><td>108 | 212
(1.96)</td></tr>
</table>

Considering constant increments with standard iterations, the LCM is able to reach the final equilibrium configuration, but snapping through the unstable configurations, as in Fig. 5.17. A large number of iterations is performed to find a new solution after the load limit point is reached (43 with SR2O, 12 with LR2O and LR4O, and 21 with CR). The WCM was not able to solve the problem with the SR2O formulation. It goes well until a point at about halfway the end of the analysis, where it stops incrementing and gets stuck. In general, all other methods and formulations had similar performances for this model and analysis settings. The initial increment of the analysis is small for this model, resulting in an average number of iterations per step of 1.00. A larger initial increment, or the use of adjusted increments, would be more efficient.

When the modified iteration scheme is used (Table 5.11), few differences are experienced, since the increment is small. One of them is that the LCM is not able to find a new solution after reaching the load limit point, diverging from the equilibrium path. Some of the other methods had their average number of iterations slightly increased, but the overall performance was quite homogeneous.

In analyzes with adjusted increments and standard iterations, the LCM captured the solution jumping over the unstable region. A common issue for most of these analyses is that a non-smooth curve was obtained due to large increments in each step, especially when the WCM method or the CR formulation was selected. A reason for this is that the desired number of iterations may be too large for this problem. Decreasing it would be the best option for getting a smooth curve with a reasonable number of steps. This issue did not occur with the GDCM, which is not based on the number of iteration, but on the GSP, to adjust the increment size.

Finally, running analyzes with adjusted increments and modified iterations resulted in smoother curves than those obtained in the tests of Table 5.12, but with less steps and total iterations needed by the analyzes of Table 5.10 and Table 5.11. The ORCM, however, failed to converge near the first load limit point when using the LR2O and LR4O formulations. The LCM also diverged when the load limit point was reached.

5.5. Lee Frame

The Lee Frame is a two-beam model, rigidly connected to each other in a 90-degree angle and simply supported at the other end, as illustrated in Fig. 5.18. Each beam is 120 cm long and is discretized into 10 equal length elements for this example. Euler-Bernoulli bending behavior is assumed. The material has a modulus of elasticity of E = 70632 MPa, the cross-section has an area of A = 6.0 cm^2 and a moment of inertia of I = 2.0 cm^4. A vertical point load is applied 24 cm from the joint, with a magnitude of P = 20 kN. The positive directions of the displacements and rotation of the loaded node (u, v, θ) are also indicated in the figure.

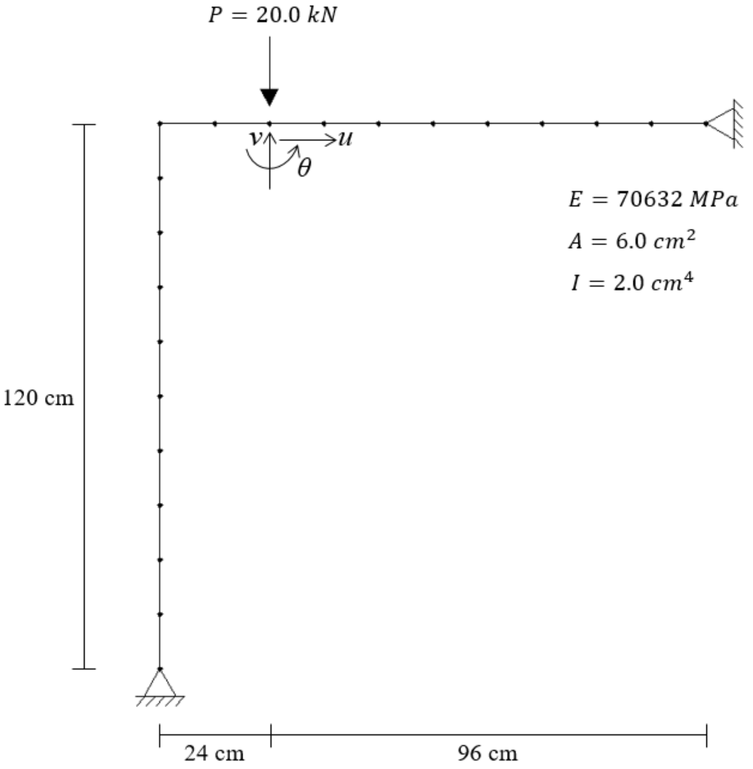


Figure 5.18 – Lee frame

This model is a well-known example for evaluating nonlinear solvers, since its behavior is highly nonlinear and the equilibrium path includes both snap-through and snap-back. It was first studied and solved by Lee *et al.* (1968). The finite element solution for the Lee Frame with the physical and geometric properties presented in Fig. 5.18 is given in Table 5.14 (Cichon, 1984).

Table 5.14 –Solution of the Lee Frame

P [kN]	***u*** [cm]	***v*** [cm]
0.0000	0.0000	0.0000
4.9033	0.3102	3.7514
12.7378	4.5534	18.2120
17.0194	14.5370	35.9240
18.3815	25.4490	47.0730
13.2291	57.3590	60.3470
-0.7074	79.6120	52.8500
-9.6871	90.3140	58.2570
2.6526	88.9740	87.6070
14.2157	86.1990	91.8670

The expected solution is compared to the results obtained with the Ftool program. Figure 5.19 shows the equilibrium path for the horizontal displacement of the loaded node, while Fig. 5.20 shows the equilibrium path for the vertical displacement. These results were obtained for the 10-elements discretization of beams and match with good accuracy the solution given by Cichon (1984). Two load limit points are identified, a maximum at the load ratio of $\lambda = 0.92$, and a minimum at the load ratio of $\lambda = -0.47$. A displacement limit point, with snap-back behavior, is also identified in the equilibrium path of the vertical displacement.

The deformed configurations at different points of the equilibrium path are shown in Fig. 5.21, with the corresponding load ratio values provided next to each deformed shape. The bold values indicate the equilibrium configurations corresponding to a load limit point.

The internal force diagrams (axial force, shear force, and bending moment), provided by the Ftool program, are given in Fig. 5.22. These internal forces correspond to the total applied load and the diagrams are plotted over the initial configuration of the model to help visualization of the results.

The comparative study of the efficiency and performance of the solution methods, using different formulations and analysis options, is presented in Table 5.15 to Table 5.18. An initial increment of 0.01, a maximum number of iterations per step of 500, a tolerance for convergence of 10^{-5}, and a desired number of iteration of 3 (when the increment is adjusted) were employed in all analyses.

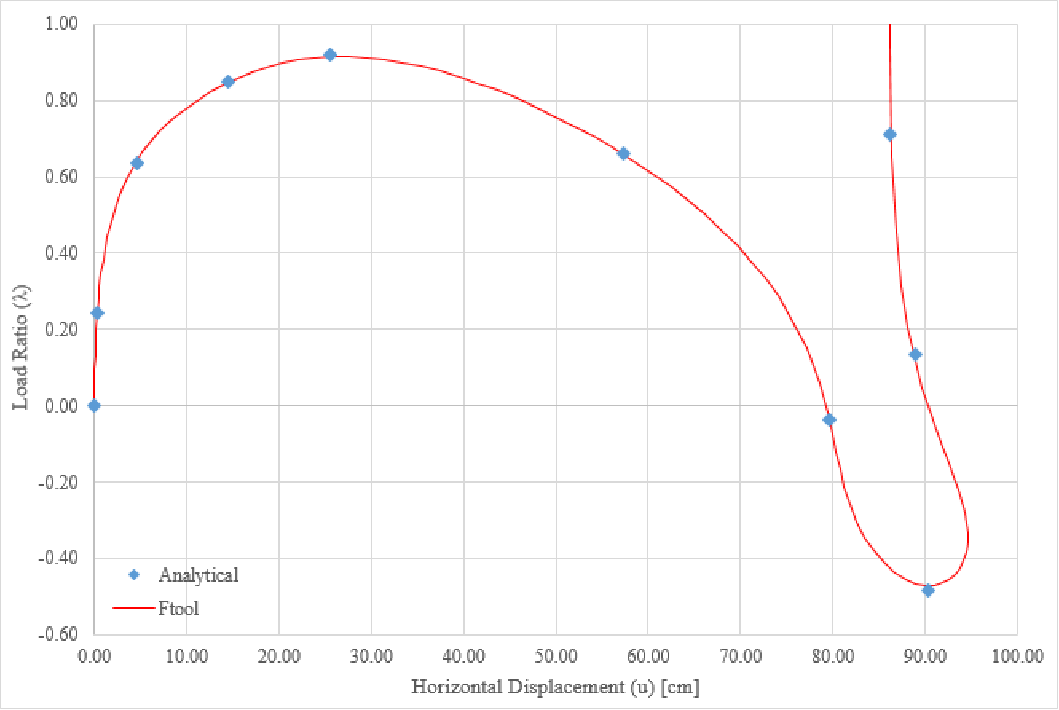


Figure 5.19 – Equilibrium path for the horizontal displacement

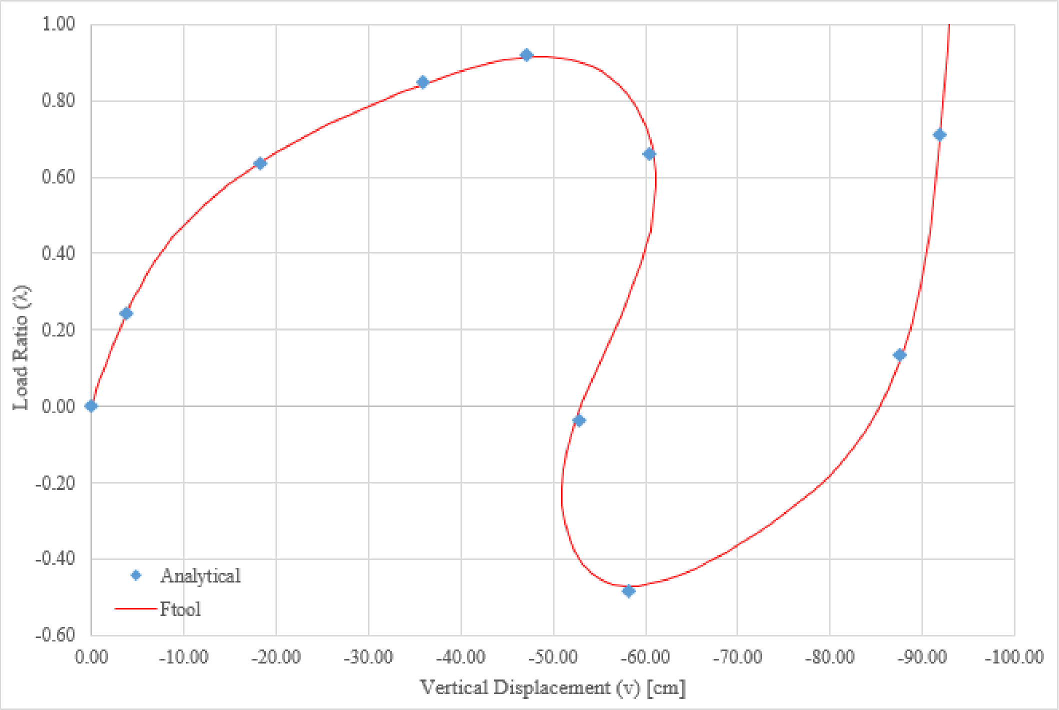


Figure 5.20 – Equilibrium path for the vertical displacement

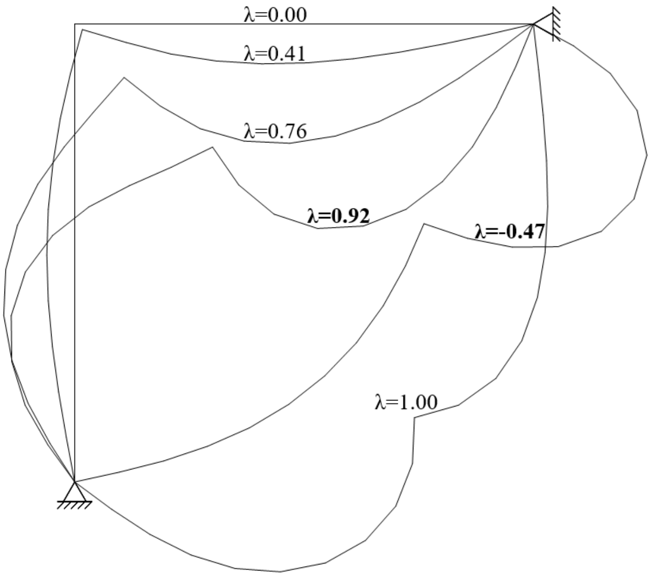
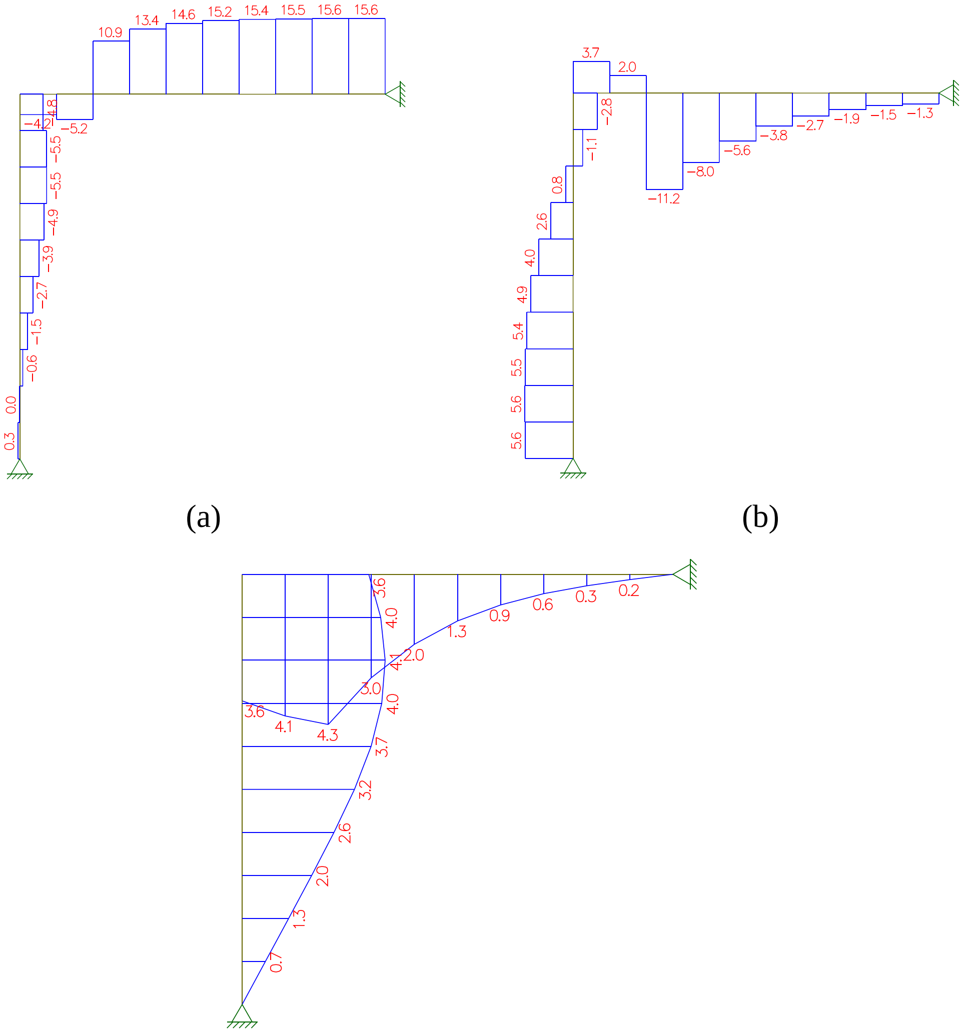


Figure 5.21 – Deformed configurations of the Lee Frame



(a)

(b)

(c)

Figure 5.22 – Internal force diagrams of the Lee Frame: (a) axial force, (b) shear force, and (c) bending moment

Table 5.15 – Example 4: Number of steps and iterations for constant increments with standard iteration scheme

	SR2O	LR2O	LR4O	CR	Mean
LCM	X	X	X	X	X
WCM	X	X	X	X	X
ALCM_FNP	X	X	X	X	X
ALCM_UNP	X	X	X	X	X
ALCM_CYL	1015 | 1597 (1.57)	1015 | 1790 (1.76)	1015 | 1790 (1.76)	1015 | 1015 (1.00)	1015 | 1548 (1.53)
ALCM_SPH	X	X	X	X	X
MNCM	1083 | 1666 (1.54)	1085 | 1872 (1.73)	1085 | 1873 (1.73)	1017 | 1017 (1.00)	1068 | 1607 (1.50)
ORCM	X	X	X	X	X
GDCM	1014 | 1598 (1.58)	1014 | 1787 (1.76)	1014 | 1787 (1.76)	1015 | 1015 (1.00)	1014 | 1547 (1.53)
Mean	1037 | 1620 (1.56)	1038 | 1816 (1.75)	1038 | 1817 (1.75)	1016 | 1016 (1.00)	1032 | 1567 (1.52)

Table 5.16 – Example 4: Number of steps and iterations for constant increments with modified iteration scheme

	SR2O	LR2O	LR4O	CR	Mean
LCM	X	X	X	X	X
WCM	X	X	X	X	X
ALCM_FNP	X	X	X	X	X
ALCM_UNP	X	X	X	X	X
ALCM_CYL	1015 | 4543 (4.48)	1015 | 4607 (4.54)	1015 | 4608 (4.54)	1015 | 4502 (4.44)	1015 | 4565 (4.50)
ALCM_SPH	X	X	X	X	X
MNCM	1001 | 4528 (4.52)	1000 | 4596 (4.60)	1000 | 4596 (4.60)	1013 | 4494 (4.44)	1004 | 4554 (4.54)
ORCM	X	X	X	X	X
GDCM	1014 | 4525 (4.46)	1014 | 4580 (4.52)	1014 | 4580 (4.52)	1015 | 4491 (4.42)	1014 | 4544 (4.48)
Mean	1010 | 4532 (4.49)	1010 | 4594 (4.55)	1010 | 4595 (4.55)	1014 | 4496 (4.43)	1011 | 4554 (4.50)

Table 5.17 – Example 4: Number of steps and iterations for adjusted increments with standard iteration scheme

<table>
<tr><th></th><th>SR2O</th><th>LR2O</th><th>LR4O</th><th>CR</th><th>Mean</th></tr>
<tr><td>LCM</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr>
<tr><td>WCM</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr>
<tr><td>ALCM_FNP</td><td>348 | 1026 (2.95)</td><td>402 | 1187 (2.95)</td><td>400 | 1180 (2.95)</td><td>210 | 624 (2.97)</td><td>340 | 1004 (2.95)</td></tr>
<tr><td>ALCM_UNP</td><td>246 | 761 (3.09)</td><td>X</td><td>239 | 779 (3.26)</td><td>196 | 593 (3.03)</td><td>227 | 711 (3.13)</td></tr>
<tr><td>ALCM_CYL</td><td>181 | 533 (2.94)</td><td>207 | 611 (2.95)</td><td>207 | 611 (2.95)</td><td>109 | 311 (2.85)</td><td>176 | 517 (2.94)</td></tr>
<tr><td>ALCM_SPH</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr>
<tr><td>MNCM</td><td>191 | 561 (2.94)</td><td>222 | 655 (2.95)</td><td>222 | 655 (2.95)</td><td>106 | 303 (2.86)</td><td>185 | 544 (2.94)</td></tr>
<tr><td>ORCM</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr>
<tr><td>GDCM</td><td>1014 | 1598 (1.58)</td><td>1014 | 1787 (1.76)</td><td>1014 | 1787 (1.76)</td><td>1015 | 1015 (1.00)</td><td>1014 | 1547 (1.53)</td></tr>
<tr><td>Mean</td><td>396 | 896 (2.26)</td><td>461 | 1060 (2.30)</td><td>416 | 1002 (2.41)</td><td>327 | 569 (1.74)</td><td>397 | 872 (2.20)</td></tr>
</table>

Table 5.18 – Example 4: Number of steps and iterations for adjusted increments with modified iteration scheme

<table>
<tr><th></th><th>SR2O</th><th>LR2O</th><th>LR4O</th><th>CR</th><th>Mean</th></tr>
<tr><td>LCM</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr>
<tr><td>WCM</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr>
<tr><td>ALCM_FNP</td><td>2989 | 8854 (2.96)</td><td>3195 | 9457 (2.96)</td><td>3194 | 9454 (2.96)</td><td>2572 | 7725 (3.00)</td><td>2988 | 8873 (2.97)</td></tr>
<tr><td>ALCM_UNP</td><td>2964 | 8780 (2.96)</td><td>3172 | 9390 (2.96)</td><td>3179 | 9410 (2.96)</td><td>2545 | 7646 (3.00)</td><td>2965 | 8807 (2.97)</td></tr>
<tr><td>ALCM_CYL</td><td>1567 | 4705 (3.00)</td><td>1625 | 4881 (3.00)</td><td>1625 | 4881 (3.00)</td><td>1437 | 4315 (3.00)</td><td>1564 | 4696 (3.00)</td></tr>
<tr><td>ALCM_SPH</td><td>2941 | 8830 (3.00)</td><td>2962 | 8896 (3.00)</td><td>2962 | 8896 (3.00)</td><td>2646 | 7948 (3.00)</td><td>2878 | 8643 (3.00)</td></tr>
<tr><td>MNCM</td><td>1605 | 4820 (3.00)</td><td>1583 | 4756 (3.00)</td><td>1583 | 4755 (3.00)</td><td>1433 | 4304 (3.00)</td><td>1551 | 4659 (3.00)</td></tr>
<tr><td>ORCM</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr>
<tr><td>GDCM</td><td>1014 | 4525 (4.46)</td><td>1014 | 4580 (4.52)</td><td>1014 | 4580 (4.52)</td><td>1015 | 4491 (4.42)</td><td>1014 | 4544 (4.48)</td></tr>
<tr><td>Mean</td><td>2180 | 6752 (3.10)</td><td>2259 | 6993 (3.10)</td><td>2260 | 6996 (3.10)</td><td>1941 | 6072 (3.13)</td><td>2160 | 6703 (3.10)</td></tr>
</table>

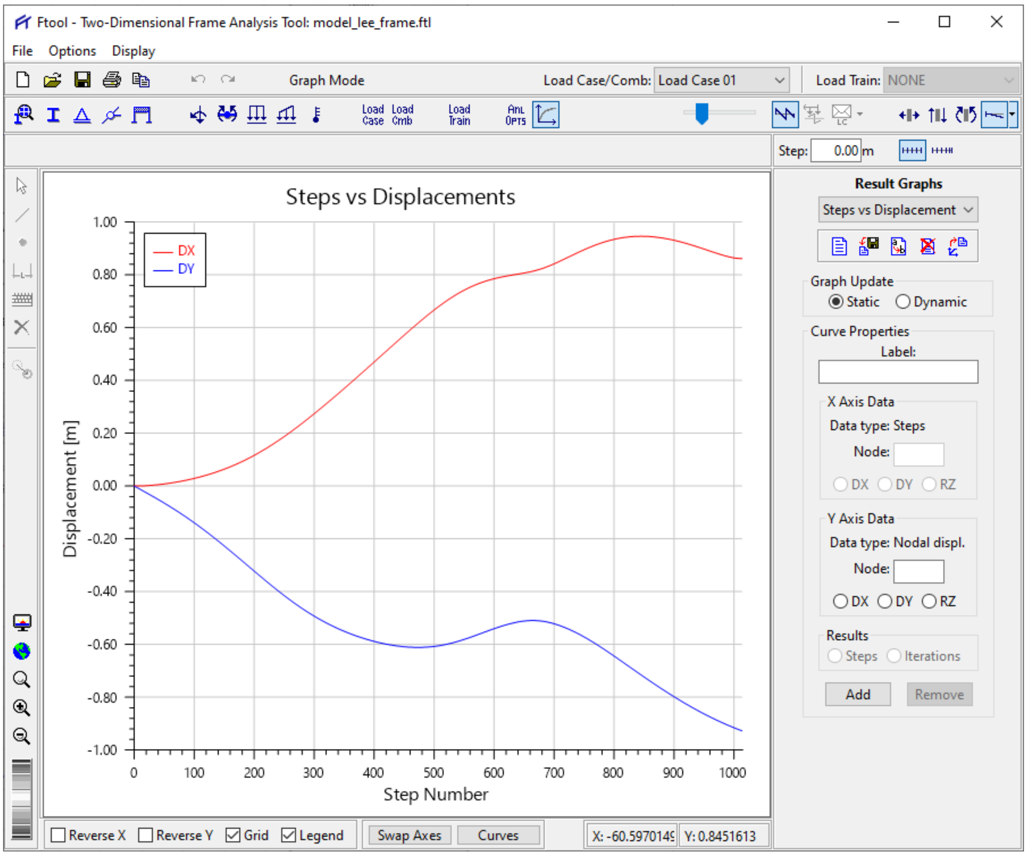
Only three methods worked well for the analysis of the Lee Frame by means of constant increments and the standard iteration scheme: ALCM_CYL, MNCM, and GDCM. A large number of steps was executed by these methods to complete the analysis, which indicates that the prescribed increment may be too small. These methods are more efficiently performed when the CR formulation is selected, since less iterations are required for a similar number of steps. The other methods faced numerical instability problems. Some expected problems that occurred were the struggles of the LCM to handle load limit points and the WCM to deal with displacement limit points, as explained earlier in this work.

The main effect of considering the modified iteration scheme to the analyzes with constant increments is the increase in the number of iterations per step. In this case, it is also verified that the CR formulation is no longer as advantageous as in the case of standard iterations.

Considering adjusted increments with standard iterations, both the ALCM_FNP and the ALCM_UNP become able to solve the problem. The latter, however, fails when using the LR2O formulation. The other methods that are based on the number of iterations to adjust the increment size had their number of steps decreased, making analysis more efficient. The CR formulation, again, happen to be the best choice between the nonlinear formulations, considering its low number of iterations in each step.

Finally, Table 5.18 shows the results of the analyzes with adjusted increments and modified iterations. These analysis options allowed more methods to solve the problem, despite the enormous number of steps required. The ORCM, together with the LCM and WCM, were the only methods that did not work well for this problem, considering any nonlinear formulation or analysis option.

For the next results, the ALCM_CYL was used with constant increments of 0.01 and standard iterations, totaling 1015 steps. The behavior of nodal displacements and applied load ratio throughout the analysis is investigated separately. Figures 5.23 and 5.24 show the value and the increment of horizontal (red) and vertical (blue) displacements of the loaded node in each analysis step. Figures 5.25 and 5.26 give the total value and the increment of the load ratio in each step. The increment of load ratio is not constant, but its values are those that provide a constant increment of a cylindrical arc-length. By visual inspection, it is possible to notice some relations between these curves and the equilibrium path of the model.



Load Case/Comb: Load Case 01

Load Train: NONE

Step: 0.00 m

Result Graphs: Steps vs Displacement

Graph Update: ● Static ○ Dynamic

Curve Properties
Label:

X Axis Data
Data type: Steps
Node:
○ DX ○ DY ○ RZ

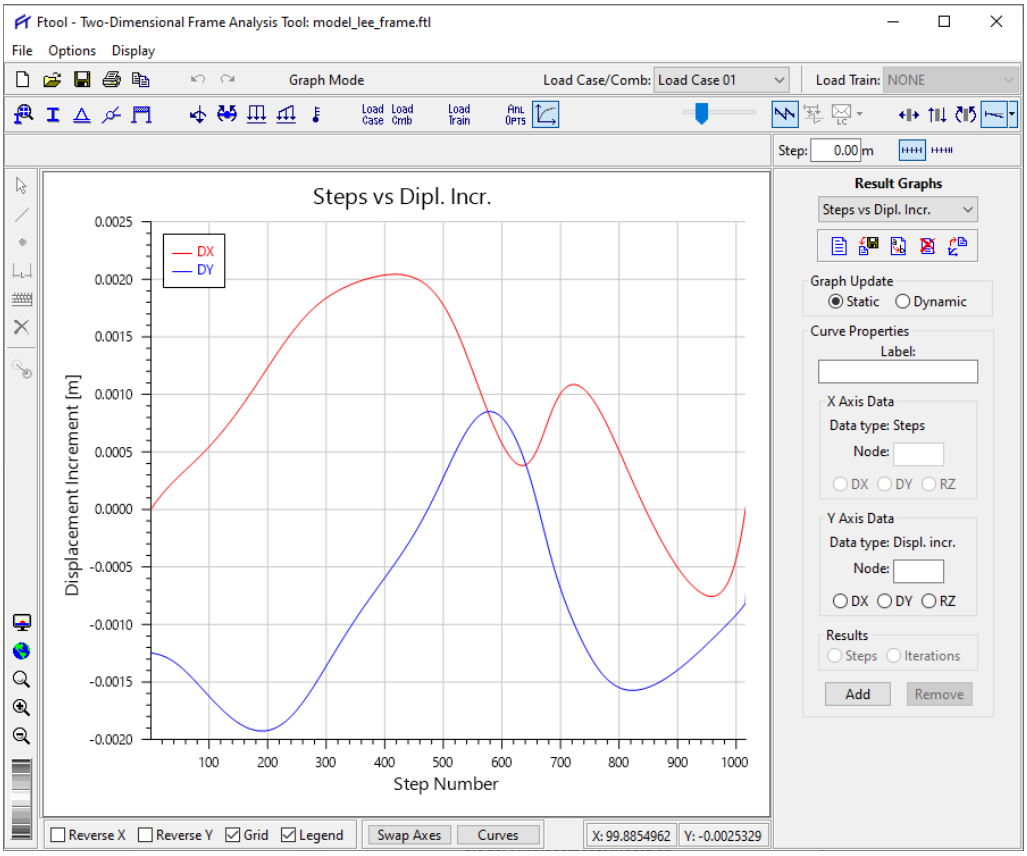
Y Axis Data
Data type: Nodal displ.
Node:
○ DX ○ DY ○ RZ

Results
○ Steps ○ Iterations

□ Reverse X □ Reverse Y ☑ Grid ☑ Legend

X: -60.597014ç Y: 0.8451613

Figure 5.23 – Nodal displacements for each step



Load Case/Comb: Load Case 01

Load Train: NONE

Step: 0.00 m

Result Graphs: Steps vs Dipl. Incr.

Graph Update: ● Static ○ Dynamic

Curve Properties
Label:

X Axis Data
Data type: Steps
Node:
○ DX ○ DY ○ RZ

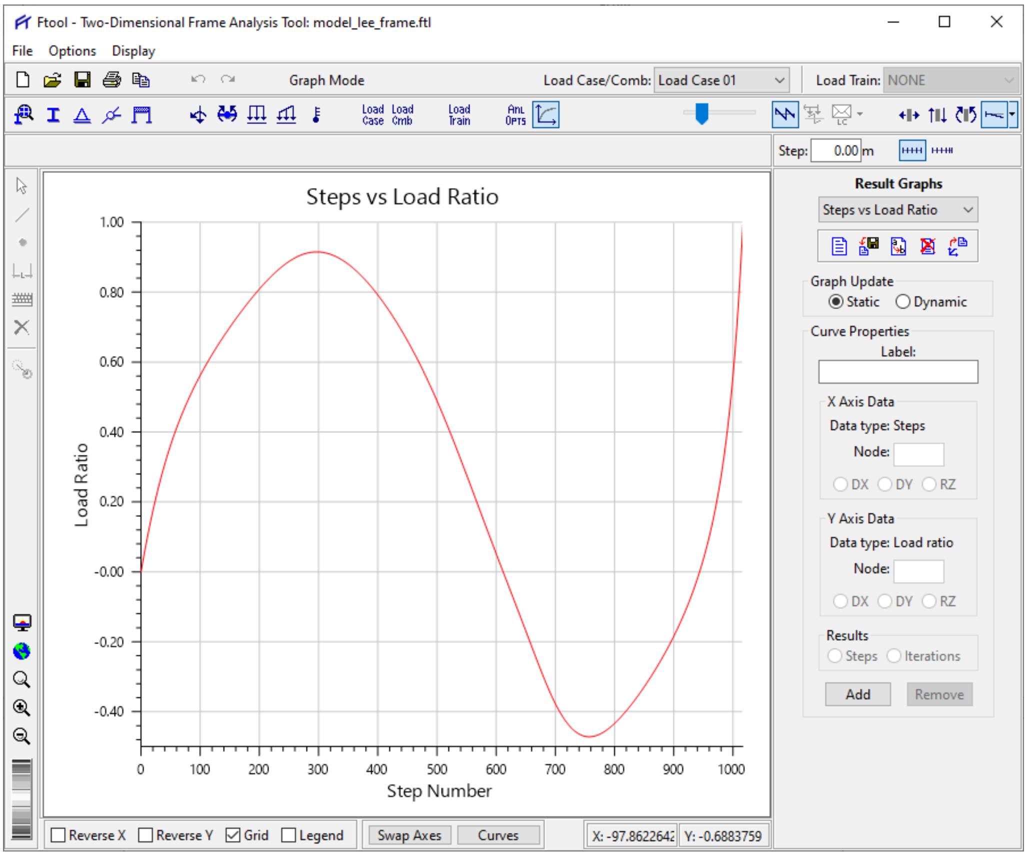
Y Axis Data
Data type: Displ. incr.
Node:
○ DX ○ DY ○ RZ

Results
○ Steps ○ Iterations

□ Reverse X □ Reverse Y ☑ Grid ☑ Legend

X: 99.8854962 Y: -0.0025329

Figure 5.24 – Increment of nodal displacements for each step



Load Case/Comb: Load Case 01

Load Train: NONE

Step: 0.00 m

Result Graphs: Steps vs Load Ratio

Graph Update: ● Static ○ Dynamic

Label:

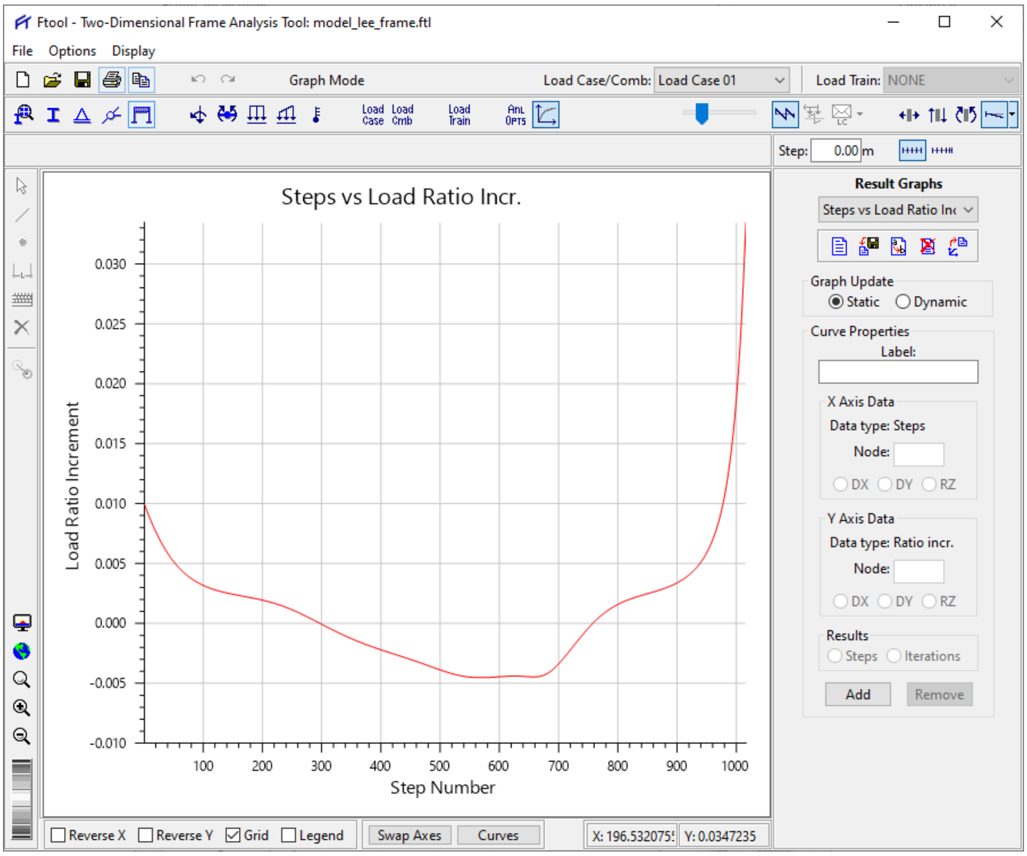
X Axis Data — Data type: Steps — Node:

Y Axis Data — Data type: Load ratio — Node:

□ Reverse X □ Reverse Y ☑ Grid □ Legend

X: -97.8622642 Y: -0.6883759

Figure 5.25 – Load ratio for each step



Load Case/Comb: Load Case 01

Load Train: NONE

Step: 0.00 m

Result Graphs: Steps vs Load Ratio Inc

Graph Update: ● Static ○ Dynamic

Label:

X Axis Data — Data type: Steps — Node:

Y Axis Data — Data type: Ratio incr. — Node:

□ Reverse X □ Reverse Y ☑ Grid □ Legend

X: 196.532075 Y: 0.0347235

Figure 5.26 – Increment of load ratio for each step

6. Final Considerations

6.1. Conclusions

The main observations about the overall behavior of the implemented solution methods and nonlinear formulations, for the analyzed models, are the following:

- The results provided by the different incremental-iterative methods and geometrically nonlinear formulations are numerically equivalent. That is, when the analysis is successfully performed, the converged equilibrium configurations are in the same curve (considering a numerical tolerance), regardless the method or formulation used to solve the problem. What distinguishes the analyzes with one or another method and formulation is the ability to capture the complete solution and the efficiency to accomplish it.
- The ALCM_CYL, MNCM, and GDCM are the most robust methods. These were the only methods able to get the complete solution in all analyzes without problems.
- The CR formulation is the most efficient, when considering the required number of steps and iterations per step to trace the equilibrium path. In most tests, considering the standard iteration scheme, this formulation needed less iterations to achieve convergence in each step than other formulations. This conclusion is not valid when the modified iteration scheme is used.
- The performance of the ALCM_FNP and ALCM_UNP are identical, or very similar, as well as the performance of the LR2O and LR4O formulations.
- The LR2O and LR4O formulations usually require more steps and iterations, and are more prone to face convergence problems.
- More methods fail to solve the problem with the modified iteration scheme than with the standard scheme.

The above observations on the efficiency and robustness of the solution methods and formulations are based on the results of a few models and cannot be taken as a rule for every geometrically nonlinear analysis of two-dimensional frame models. Even the same models, if analyzed with other program, may not lead to the same exact conclusions. This is because each computational implementation of the solution algorithms has some particular considerations that cause small variations in the analysis performance. However, similar conclusions are expected for problems of the same nature. The important point is to realize that what was stated in the introduction of this work is valid (Leon *et al*., 2011 and Bergan *et al.*, 1978):

"*One single solution method may not be capable of solving any general nonlinear problem. Dealing with numerical procedures, exposes us to problems of convergence and numerical stability that naturally occur in these analyses. Therefore, a computer program for nonlinear analysis should possess several alternative algorithms for the solution of the nonlinear system of equations.*"

Many options for performing the geometrically nonlinear analysis were implemented in the new version of the Ftool program, as described in detail throughout this document. These options include multiple solution methods and nonlinear formulations, different types of increment strategies and iteration schemes. In addition, the possibility to run the analysis in an interactive-adaptive fashion, allowing users to control the flow of the analysis, intervene when problems appear, and redefine parameters, is of great contribution to make Ftool an appropriate platform for nonlinear analyzes. All the new developments for solving the geometrically nonlinear problem were implemented in Ftool without compromising its original operation and respecting its educational philosophy, which is one of the most remarkable features of the program.

The focus on the educational aspect, with intuitive resources to deal with all the parameters involved in a nonlinear analysis, is essential to simplify the analysis process and make it easier to comprehend. A consequence is that the overall numerical experimentation process becomes more accessible to those with little experience and knowledge on the subject, arousing the interest of students in learning about structural nonlinearity. Therefore, it is expected that this new tool will contribute considerably to the teaching of structural analysis with second-order effects.

6.2. Future Works

Some suggestions for future implementations are:

- Parallel analysis: As defined by Gattass & Abel (1983), a parallel analysis are those analysis that do not directly give the response of the structure but aid in the comprehension of its behavior or in the selection of algorithms and analysis parameters. In the context of geometric nonlinearity, a handy parallel analysis is the ability to compute buckling (or critical) loads. The buckling loads provide an idea about the stability of the structure and can be used to aid the selection of the step increments. Buckling modes, which are the deformed configuration corresponding to the buckling loads, are also an interesting result to provide when performing the parallel analysis. These results can be obtained from an eigenvalue problem, which should be implemented in the analysis solver.
- Deformed configuration: To better represent the deflected shape of the elements, instead of straight lines between nodal coordinates, it is proposed to interpolate nodal rotations using the Hermitian shape functions over the corotated configuration. This would be useful when poorly discretized elements are used.
- Automatic discretization: An important feature to facilitate modelling, when considering a nonlinear analysis, is allowing an automatic discretization of beam elements into a desired number of sub-elements.
- Inclusion of new sources of nonlinearities: With the developed general algorithm for the solution of the system of nonlinear equilibrium equations, the inclusion of a new source of nonlinearity is straightforward. Material nonlinearity and the consideration of plastic hinges are some important nonlinear effects in the analysis of reticulated structures that should be included in the program.

7. Bibliography

ALMEIDA, C. A.; ALBINO, J. C. R.; MENEZES, I. F. M.; PAULINO, G. H. Geometric nonlinear analyses of functionally graded beams using a tailored Lagrangian formulation. **Mechanics Research Communications**, vol. 38, n. 8, pp. 553-559, 2011.

ALVES, R. V. **Formulação para Análise Não-Linear Geométrica em Referencial Lagrangiano Total**. 1º Seminário de Doutorado, COPPE/UFRJ, Rio de Janeiro, Brazil, 1993a.

ALVES, R. V. **Formulação para Análise Não-Linear Geométrica em Referencial Lagrangiano Atualizado**. 3º Seminário de Doutorado, COPPE/UFRJ, Rio de Janeiro, Brazil, 1993b.

ARGYRIS, J. H.; KELSEY, S.; KAMEL, H. Matrix methods of structural analysis: a precis of recent developments. **Matrix Methods of Structural Analysis**, B. F. deVeubeke (ed.), Pergamon Press, Oxford, 1964.

BAIÃO, M. V. **Análise Não Linear Geométrica de Estruturas Aporticadas Planas**. Undergraduate thesis, Escola de Engenharia, Universidade Federal Fluminense, Niterói, 2017.

BATHE, K. J.; BOLOURCHI, S. Large displacement analysis of three-dimensional beam structures. **International Journal for Numerical Methods in Engineering**, vol. 14, n. 7, pp. 961-986, 1979.

BATHE, K. J.; DVORKIN, E. N. On the automatic solution of nonlinear finite element equations. **Computers & Structures**, vol. 17, n. 5–6, pp. 871–879, 1983.

BATHE, K. J. **Finite Element Procedures**. Prentice-Hall, Englewood Cliffs, NJ, 1996.

BATOZ, J. L.; DHATT, G. Incremental displacement algorithms for nonlinear problems. **International Journal for Numerical Methods in Engineering**, vol. 14, n. 8, pp. 1262-1267, 1979.

BATTINI, J. M. **Co-rotational Beam Elements in Instability Problems**. PhD Thesis, Royal Institute of Technology, Stockholm, Sweden, 2002.

BAZANT, Z. P.; CEDOLIN, L. **Stability of Structures**. World Scientific Publishing Co. Pte. Ltd., Singapura, 2010.

BELYTSCHKO, T.; HSIEH, J. Non-linear transient finite element analysis with convected co-ordinates. **International Journal for Numerical Methods in Engineering**, vol. 7, n. 3, pp. 255-271, 1973.

BELYTSCHKO, T.; GLAUM, L. W. Applications of higher order corotational stretch theories to nonlinear finite element analysis. **Computers & Structures**, vol. 10, n. 1-2, pp. 175-182, 1979

BERGAN, P. G.; CLOUGH, R. W. Convergence criteria for iterative processes. **AIAA Journal**, vol. 10, n. 8, pp. 1107-1108, 1972.

BERGAN, P. G.; HORRIGMOE, G.; BRAKELAND, B.; SOREIDE, T. H. Solution techniques for non-linear finite element problems. **International Journal for Numerical Methods in Engineering**, vol. 12, n. 11, pp. 1677-1696, 1978.

BERGAN, P. G. Solution algorithms for nonlinear structural problems. **Computers & Structures**, vol. 12, n. 4, pp. 497–509, 1980.

BERGAN, P. G.; SIMONS, J. Hyperplane displacement control methods in nonlinear analysis. In: W. K. Liu, T. Belytschko and K. C. Park (eds.), **Innovative Methods for Nonlinear Problems**, Pineridge Press, Swansea, UK, pp. 345–364, 1984.

BERTSEKAS, D. P. **Constrained Optimization and Lagrange Multiplier Methods**. Academic Press, New York, 1982.

BISSHOPP, K. E.; DRUCKER, D. C. Large deflections of cantilever beams. **Quarterly of Applied Mathematics**, vol. 3, pp. 272-275, 1945.

BUDIANSKY, B. Notes on nonlinear shell theory. **Journal of Applied Mechanics**, vol. 35, n. 2, pp. 393-401, 1968.

BURGOS, R. B.; MARTHA, L. F. Exact shape functions and tangent stiffness matrix for the buckling of beam-columns considering shear deformation. In: **XXXIV Iberian Latin American Congress on Computational Methods in Engineering** (XXXIV CILAMCE), Pirenópolis, Brazil, 2013.

BUTCHER, J. C. A history of Runge-Kutta methods. **Applied Numerical Mathematics**, vol. 20, n. 3, pp. 247-260, 1996.

CARDOSO, E. L.; FONSECA, J. S. O. The GDC method as an orthogonal arc-length method. **Communications in Numerical Methods in Engineering**, vol. 23, n. 4, pp. 263-271, 2007.

CARVALHO, P. C. P.; GATTASS, M.; MARTHA, L. F. A software tool which allows interactive creation of planar subdivisions, and applications to educational programs. CATS’90 – **Proceedings of International Conference on Computer Aided Training in Science and Technology**, Edited by E. Oñate *et al.*, CIMNE–Pineridge Press, pp. 201-207, 1990.

CHAJES, A.; CHURCHILL, J. E. Nonlinear frame analysis by finite element methods. **Journal of Structural Engineering**, vol. 113, n. 6, pp. 1221-1235, 1987.

CHAN, S. L. Geometric and material nonlinear analysis of beam-columns and frames using the minimum residual displacement method. **International Journal for Numerical Methods in Engineering**, vol. 26, n. 2, pp. 2657-2669, 1988.

CHAPRA, S. C.; CANALE, R. P. **Numerical Methods for Engineers**. McGraw-Hill Education, New York, 2013.

CHEN, W. F.; LUI, E. M. **Stability Design of Steel Frames**. CRC Press, 1991.

CHEN, D. C. **Geometric Nonlinear Analysis of Three-Dimensional Structures**. MSc thesis, Cornell University, Ithaca, NY, 1994.

CICHON, C. Large displacements in-plane analysis of elastic-plastic frames. **Computers & Structures**, vol. 19, n. 5-6, pp. 737-745, 1984.

CLARKE, M. J.; HANCOCK, G. J. A study of incremental-iterative strategies for non-linear analyses. **International Journal for Numerical Methods in Engineering**, vol. 29, n. 7, pp. 1365–1391, 1990.

COX, B.; NOVOBILSKI, A. **Object-Oriented Programming: An Evolutionary Approach**. Addison-Wesley, Upper Saddle River, NJ, 1991.

CRISFIELD, M. A. A fast incremental/iterative solution procedure that handles snap-through. **Computers & Structures**, vol. 13, n.1-3, pp. 55–62, 1981.

CRISFIELD, M. A. **Non-Linear Finite Element Analysis of Solids and Structures**. Vol. 1, John Wiley & Sons, New York, 1991.

CRISFIELD, M. A. **Non-Linear Finite Element Analysis of Solids and Structures**. Vol. 2, John Wiley & Sons, New York, 1997.

CRISFIELD, M. A.; MOITA, G. F. A unified co-rotational framework for solids, shells and beams. **International Journal of Solids and Structures**, vol. 33, n. 20-22, pp. 2969-2992, 1996.

DEL SAVIO, A. **Modelagem Computacional de Estruturas de Aço com Ligações Semi-Rígidas**. MSc thesis, Departamento de Engenharia Civil, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2004.

EBNER, A. M.; UCCIFERRO, J. J. A theoretical and numerical comparison of elastic nonlinear finite element methods. **Computers & Structures**, vol. 2, n. 5-6, pp. 1043-1061, 1972.

EPSTEIN, M.; MURRAY, D. W. Large deformation in plane analysis of elastic beams. **Computers & Structures**, vol. 6, n. 1, pp. 1-9, 1976.

FELIPPA, C. A. **Nonlinear Finite Element Methods**. Class notes, Department of Aerospace Engineering Sciences, University of Colorado, Boulder, 2004.

FELIPPA, C. A. **Introduction to Finite Element Methods**. Class notes, Department of Aerospace Engineering Sciences, University of Colorado, Boulder, 2005.

FONSECA, F. T. **Sistema Computacional para Análise Dinâmica Geometricamente Não-Linear Através do Método dos Elementos Finitos**. MSc thesis, Programa de Pós-Graduação em Engenharia de Estruturas, Universidade Federal de Minas Gerais, Belo Horizonte, 2008.

FRIEDMAN, Z.; KOSMATKA, J. B. An improved two-node Timoshenko beam finite element. **Computers & Structures**, vol. 47, n. 3, pp. 473-481, 1993.

FUINA, J. S. **Métodos de Controle de Deformações para Análise Não-Linear de Estruturas**. MSc thesis, Programa de Pós-Graduação em Engenharia de Estruturas, Universidade Federal de Minas Gerais, Belo Horizonte, 2004.

GALVÃO, A. S. **Formulações Não-Lineares de Elementos Finitos para Análise de Sistemas Estruturais Metálicos Reticulados Planos**. MSc thesis, Programa de Pós Graduação em Engenharia Civil, Universidade Federal de Ouro Preto, Ouro Preto, 2000.

GALVÃO, A. S. **Instabilidade Estática e Dinâmica de Pórticos Planos com Ligações Semi-Rígidas**. PhD thesis, Departamento de Engenharia Civil, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2004.

GATTASS, M.; ABEL, J. F. Interactive-adaptive, large displacement analysis with real-time computer graphics. **Computers & Structures**, vol. 16, n. 1-4. pp. 141-152, 1981

GERMANIO, L. **Implementação Orientada a Objetos da Solução de Problemas Estruturais Dinâmicos via Método dos Elementos Finitos**. MSc thesis, Programa de Pós-Graduação em Engenharia de Estruturas, Universidade Federal de Minas Gerais, Belo Horizonte, 2005.

GOLDBERG, J. E.; RICHARDS, R. H. Analysis of nonlinear structures. **Journal of the Structural Division**, vol. 89, n. 4, pp. 333-352, 1963.

GOMES, B. C. **Ferramenta Gráfico-Interativa para Análise Não Linear Física de Pórticos Planos de Concreto Armado Considerando o Diagrama Momento-Curvatura**. MSc thesis, Departamento de Engenharia Civil e Ambiental, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2019.

GOTO, Y.; CHEN, W. Second-order elastic analysis for frame design. **Journal of Structural Engineering**, vol. 113, n. 7, pp. 1500-1519, 1987.

HEYLIGER, P. R.; REDDY, J. N. A higher order beam finite element for bending and vibration problems. **Journal of Sound and Vibration**. vol. 126, n. 2, pp. 309-326, 1988.

HIBBITT, H. D.; MARCAL, P. V.; RICE, J. R. A finite element formulation for problems of large strain and large displacement. **International Journal of Solids Structures**, vol. 6, n. 8, pp. 1069-1086, 1970.

HSIAO, K. M.; LIN, J. Y.; LIN, W. Y. A consistent co-rotational finite element formulation for geometrically nonlinear dynamic analysis of 3-D beams. **Computer Methods in Applied Mechanics and Engineering**, vol. 169, n. 1-2, pp. 1-18, 1999.

IURA, M. Effects of coordinate system on the accuracy of corotational formulation for Bernoulli-Euler's beam. **International Journal of Solids and Structures**, vol. 31, n. 20, pp. 2793- 2806, 1994.

IZZUDDIN, B. A.; ELNASHAI, A. S. Eulerian formulation for large-displacement analysis of space frames. **Journal of Engineering Mechanics**, vol. 119, n. 3, pp. 549-569, 1993.

JEAN, C. F. **Método de Controle de Comprimento de Arco por Variação de Energia para Análise Não Linear de Estruturas**. MSc thesis, Programa de Pós-Graduação em Engenharia de Estruturas, Universidade Federal de Minas Gerais, Belo Horizonte, 2017.

JENNINGS, A. Frame analysis including change of geometry. **Journal of the Structural Division**, vol. 94, n. 3, pp. 627-644, 1968.

KIEN, N. D. A Timoshenko beam element for large displacement analysis of planar beams and frames. **International Journal of Structural Stability and Dynamics**, vol. 12, n. 6, 2012.

KRENK, S. An orthogonal residual procedure for non-linear finite element equations. **International Journal for Numerical Methods in Engineering**, vol. 38, n. 5, pp. 823-839, 1995.

KRENK, S; HEDEDAL, O. A dual orthogonality procedure for nonlinear finite element equations. **Computer Methods in Applied Mechanics and Engineering**, vol. 123, n. 1-4, pp. 95-107, 1995.

KRENK, S.; VISSING-JORGENSEN, C.; THESBJERG, L. Efficient collapse analysis for framed structures. **Computers & Structures**, vol. 72, n. 4-5, 481-496, 1999.

KRENK, S. **Non-linear Modeling and Analysis of Solids and Structures**. Cambridge University Press, New York, 2009.

LEE, S.; MANUEL, F. S.; ROSSOW, E. C. Large deflections and stability of elastic frames. **Journal of the Engineering Mechanics Division**, vol. 94, n. 2, pp. 521-548, 1968.

LEON, S. E.; PAULINO, G. H.; PEREIRA, A.; MENEZES, I. F. M.; LAGES, E. N. A unified library of nonlinear solution schemes. **Applied Mechanics Reviews**, vol. 64, n. 4, 2011.

LEON, S. E.; LAGES, E.; ARAUJO, C. N.; PAULINO, G. H. On the effect of constraint parameters on the generalized displacement control method. **Mechanics Research Communications**, vol. 56, pp. 123-129, 2014.

LEVINSON, M. A new rectangular beam theory. **Journal of Sound and Vibration**. vol. 74, n. 1, pp. 81-87, 1981.

LEVY, C. H.; FIGUEIREDO, L. H.; GATTAS, M.; LUCENA, C.; COWAN, D. IUP/LED: A portable user interface development tool. **Software: Practice and Experience**, vol. 26, n. 7, pp. 737-762, 1996.

LIMA, S. S. **Análise de Estruturas com Computadores**. Ciência Moderna, Rio de Janeiro, 2017.

MACHADO, F. C. S. **Análise Inelástica de Segunda-Ordem de Sistemas Estruturais Metálicos**. MSc thesis, Programa de Pós Graduação em Engenharia Civil, Universidade Federal de Ouro Preto, Ouro Preto, 2005.

MALLET, R. H.; MARCAL, P. V. Finite element analysis of nonlinear structures. **Journal of the Structural Division**, vol. 94, n. 9, pp. 2081-2106, 1968.

MALVERN, L. E. **Introduction to the Mechanics of a Continuous Medium**. Prentice-Hall, Englewood Cliffs, NJ, 1969.

MÄNTYLÄ, M. **An Introduction to Solid Modeling**, Computer Science Press, Maryland, 1988.

MARCAL, P. V. A stiffness method for elastic-plastic problems. **International Journal of Mechanical Sciences**, vol. 7, n. 4, pp. 229-238, 1965.

MARTHA, L. F. Ftool: A structural analysis educational interactive tool. **Proceedings of the Workshop in Multimedia Computer Techniques in Engineering Education**, Institute for Structural Analysis, Technical University of Graz, Austria, pp. 51-65, 1999.

MARTHA, L. F.; PARENTE, E. An object-oriented framework for finite element programming. **Proceedings of the Fifth World Congress on Computational Mechanics** (IACM), Vienna, Austria, 2002.

MARTHA, L. F. **Análise de Estruturas: Conceitos e Métodos Básicos**. Elsevier, Rio de Janeiro, 2017.

MARTHA, L. F. **Análise Matricial de Estruturas com Orientação a Objetos**. Elsevier, Rio de Janeiro, 2018.

MARTIN, H. C. On the derivation of stiffness matrices for the analysis of large deflection and stability problems. **Conference of Matrix Methods in Structural Mechanics**, Wright-Patterson Air Force Base, Ohio, 1966.

MATTIASSON, E. Numerical results from large deflection beam and frame problems analyzed by means of elliptic integrals. **International Journal for Numerical Methods in Engineering**, vol. 17, n. 1, pp. 145-153, 1981.

MATTIASSON, K.; SAMUELSSON, A. Total and updated Lagrangian forms of the co-rotational finite element formulation in geometrically and materially nonlinear analysis. **Numerical Methods for Non-linear Problems**, Swansea, UK, pp. 134-151, 1984.

MATTIASSON, K.; BENGTSSON, A.; SAMUELSSON, A. On the accuracy and efficiency of numerical algorithms for geometrically nonlinear structural analysis. **Finite Element Methods for Nonlinear Problems**, Europe-US Symposium, Trondheim, Norway, 1985.

MAXIMIANO, D. P. **Uma Técnica Eficiente para Estabilizar a Estratégia do Resíduo Ortogonal na Análise Não Linear de Estruturas**. MSc thesis, Programa de Pós Graduação em Engenharia Civil, Universidade Federal de Ouro Preto, Ouro Preto, 2012.

MCGUIRE, W.; GALLAGHER, R. H.; ZIEMIAN, R. D. **Matrix Structural Analysis**. John Wiley & Sons, New York, 2000.

MEEK, J. L.; TAN, H. S. Geometrically nonlinear analysis of space frames by an incremental iterative technique. **Computer Methods in Applied Mechanics and Engineering**, vol. 47, n. 3, pp. 261-282, 1984.

MELO, J. F. A. S. A. C. **Análise Física e Geometricamente Não Linear de Pórticos Planos pela Descrição Corrotacional**. MSc thesis, Programa de Pós-Graduação em Engenharia de Estruturas, Universidade Federal de Minas Gerais, Belo Horizonte, 2017.

MONDKAR, D. P.; POWELL, G. H. Evaluation of solution schemes for nonlinear structures. **Computers & Structures**, vol. 9, n. 3, pp. 223–236, 1978.

MUÑOZ, L. F.; ROEHL, D. A continuation method with combined restrictions for nonlinear structure analysis. **Finite Elements in Analysis and Design**, vol. 130, pp. 53-64, 2017.

MURRAY, D. W.; WILSON, E. L. Finite-element large deflection analysis of plates. **Journal of the Engineering Mechanics Division**, vol. 95, n. 1, pp. 143-166, 1969.

NANAKORN, P.; VU, L. N. A 2D field-consistent beam element for large displacement analysis using the total Lagrangian formulation. **Finite Elements in Analysis and Design**, vol. 42, pp. 1240-1247, 2006.

NEUENHOFER, A.; FILIPPOU, F. C. Geometrically nonlinear flexibility-based frame finite element. **Journal of Structural Engineering**, vol. 124, n. 6, pp. 704-711, 1998.

OLIVEIRA, G. C.; SILVA, W. T. M. Análise não linear de arcos utilizando o elemento de viga unificado Bernoulli-Timoshenko e a formulação corotacional. **Revista Eletrônica de Engenharia Civil**, vol. 13, n. 2, pp. 1-16, 2017.

ORAN, C. Tangent stiffness in plane frames. **Journal of the Structural Division**, vol. 99, n. 6, pp. 973-985, 1973.

PACOSTE, C.; ERIKSSON, A. Element behavior in post-critical plane frames analysis. **Computer Methods in Applied Mechanics and Engineering**, vol. 125, n. 1-4, pp. 319-343, 1995.

PACOSTE, C.; ERIKSSON, A. Beam elements in instability problems. **Computer Methods in Applied Mechanics and Engineering**, vol. 144, n. 1-2, pp. 163-197, 1997.

PARENTE, E. **FAST 1.0.0 – User Manual**. Departamento de Engenharia Estrutural e Construção Civil, Universidade Federal do Ceara, Fortaleza, 2018.

PETROLITO, J. Stiffness analysis of beams using a higher-order theory. **Computers and Structures**. vol. 55, n. 1, pp. 33-39, 1995.

PILKEY, W. D.; KANG, W.; SCHRAMM, U. New structural matrices for a beam element with shear deformation. **Finite Elements in Analysis and Design**. vol. 19, n. 1-2, pp. 25-44, 1995.

PITANGUEIRA, R. L. S.; FONSECA, F. T. Insane: uma plataforma para computação científica. **X Encontro de Modelagem Computacional**, Nova Friburgo, 2007.

POWELL, G. H. Theory of nonlinear elastic structures. **Journal of the Structural Division**, vol. 95, n. 12, pp. 2687-2702, 1969.

POWELL, G. H.; SIMONS, J. Improved iteration strategy for nonlinear structures. **International Journal for Numerical Methods in Engineering**, vol. 17, n. 10, pp. 1455–1467, 1981.

PRADO, I. M. **CS-ASA Preprocessor: Sistema Gráfico Interativo de Pré-Processamento para Análise Avançada de Estruturas**. MSc thesis, Programa de Pós Graduação em Engenharia Civil, Universidade Federal de Ouro Preto, Ouro Preto, 2012.

RAMM, E. Strategies for tracing the non-linear response near limit-points. In: W. Wunderlich, E. Stein, K. J. Bathe (eds.), **Nonlinear Finite Element Analysis in Structural Mechanics**. Springer, Berlin, pp. 68-89, 1981.

RAMM, E. The Riks/Wempner Approach – An extension of displacement control method in non-linear analysis. In: E. Hinton *et al.* (eds.), **Non-linear Computational Mechanics**, Pineridge Press, pp. 63-86, 1982.

RANGEL, R. L.; MARTHA, L. F. LESM – An object-oriented MATLAB program for structural analysis of linear element models. **Computer Applications in Engineering Education**, vol. 27, n. 3, pp. 553-571, 2019.

REZAIEE-PAJAND, M.; TATAR, M.; MOGHADDASIE, B. Some geometrical bases for incremental-iterative methods. **International Journal of Engineering, Transactions B: Applications**, vol. 22, n. 3, pp. 245-256, 2009.

RIKS, E. The application of Newton's methods to the problems elastic stability. **Journal of Applied Mechanics**, vol. 39, n. 4, pp. 1060-1065, 1972.

RIKS, E. An incremental approach to the solution of snapping and buckling problems. **International Journal of Solids and Structures**, vol. 15, n. 7, pp. 529-551, 1979.

ROCHA, G. **Estratégias de Incremento de Carga e de Iteração para Análise Não Linear de Estruturas**. MSc thesis, Programa de Pós Graduação em Engenharia Civil, Universidade Federal de Ouro Preto, Ouro Preto, 2000.

RODRIGUES, M. A. C. **Soluções Integradas para as Formulações do Problema de Não Linearidade Geométrica**. PhD thesis, Departamento de Engenharia Civil e Ambiental, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2019.

ROEHL, D. **Uma Metodologia para Análise Não-Linear Geométrica de Pórticos Espaciais com Liberações de Extremidade de Barra e Cargas Distribuídas**. MSc thesis, Departamento de Engenharia Civil, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 1987.

SANTANA, M. V. B. **Desenvolvimento de Sistema Computacional via MATLAB/GUI (Graphical User Interface) para Análise Geometricamente Não Linear de Estruturas**. MSc thesis, Programa de Pós Graduação em Engenharia Civil, Universidade Federal de Ouro Preto, Ouro Preto, 2015.

SCHRAMM, U.; KITIS, L.; KANG, W.; PILKEY, W. D. On the shear deformation coefficient in beam theory. **Finite Elements in Analysis and Design**. vol. 16, n. 2, pp. 141-162, 1994.

SILVA, A. R. D. **Sistema Computacional para Análise Avançada Estática e Dinâmica de Estruturas Metálicas**. PhD thesis, Programa de Pós Graduação em Engenharia Civil, Universidade Federal de Ouro Preto, Ouro Preto, 2009.

SILVA, J. L. **Formulações Corrotacionais 2D para Análise Geometricamente Não Linear de Estruturas Reticuladas**. MSc thesis, Programa de Pós Graduação em Engenharia Civil, Universidade Federal de Ouro Preto, Ouro Preto, 2016.

SILVA, M. F. D. S.; MENEZES, I. F. M.; MARTHA, L. F. Um método simplificado para análise não-linear geométrica no Ftool. In: **XXXVII Iberian Latin-American Congress on Computational Methods in Engineering** (XXXVII CILAMCE), Brasília, Brazil, 2016.

SILVA, M. F. D. S. **Ferramenta Gráfico-Interativa para o Dimensionamento de Pórticos Planos de Concreto Armado Considerando Não Linearidade Geométrica**. MSc thesis, Departamento de Engenharia Civil e Ambiental, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2017.

SILVEIRA, R. A. M. **Análise de Elementos Estruturais Esbeltos com Restrições Unilaterais de Contato**. PhD thesis, Departamento de Engenharia Civil, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 1995.

SILVEIRA, R. A. M.; ROCHA, G.; GONÇALVES, P. B. Estratégias de incremento de carga e iteração para análise não-linear de estruturas. In: **XX Iberian Latin-American Congress on Computational Methods in Engineering** (XX CILAMCE), São Paulo, Brazil, pp. 213.1-213.20, 1999.

SOUZA, R. M. **Force-Based Finite Elements for Large Displacement Inelastic Analysis of Frames**. PhD thesis, Department of Civil and Environmental Engineering, University of California, Berkeley, 2000.

TAUCHERT, T. R. **Energy Principles in Structural Mechanics**. Indo American Books, New York, 2007.

TEH, L. H.; CLARKE, M. J. Co-rotational and Lagrangian formulations for elastic three-dimensional beam finite elements. **Journal of Constructional Steel Research**, vol. 48, n. 2-3, pp. 123–144, 1998.

TEH, L. H.; CLARKE, M. J. Tracing secondary equilibrium paths of elastic framed structures. **Journal of Engineering Mechanics**, vol. 125, n. 12, pp. 1358-1364, 1999.

THOMPSON, J. M. T.; HUNT, G. W. **A General Theory of Elastic Stability**. John Wiley & Sons, New York, 1973.

TIMOSHENKO, S. P.; GERE, J. M. **Mechanics of Materials**. Wadsworth International Group, 1982.

TIMOSHENKO, S. P.; GERE, J. M. **Theory of Elastic Stability**. Dover Publications, 2009.

TORKAMANI, M. A. M.; SONMEZ, M.; CAO, J. Second-order elastic plane-frame analysis using finite-element method. **Journal of Structural Engineering**, vol. 12, n. 9, pp. 1225-1235, 1997.

TURNER, M. J.; Dill, E. H.; MARTIN, H. C.; MELOSH, R. J. Large deflection of structures subjected to heating and external loads. **Journal of the Aerospace Sciences**, vol. 27, n. 2, pp. 97-107, 1960.

WEMPNER, G. A. Discrete approximations related to nonlinear theories of solids. **International Journal of Solids and Structures**, vol. 7, n. 11, pp. 1581–1599, 1971.

WEN, R. K.; RAHIMZADEH J. Nonlinear elastic frame analysis by finite element. **Journal of Structural Engineering**, vol. 109, n. 8, p. 1952-1971, 1983.

WHITE, R. N.; GERGELY, P.; SEXSMITH, R. G. **Structural Engineering – Combined Edition, v. 1: Introduction to Design Concepts and Analysis; v. 2: Indeterminate Structures**. John Wiley & Sons, New York, 1976.

WILLIAMS, F. W. An approach to the non-linear behaviour of the members of a rigid jointed plane framework with finite deflections. **The Quarterly Journal of Mechanics and Applied Mathematics**, vol. 17, n. 4, pp. 451–469, 1964.

WONG, M. B.; TIN-LOI, F. Geometrically nonlinear analysis of elastic framed structures. **Computers & Structures**, vol. 34, n. 4, pp. 633-640, 1990.

YANG, Y. B.; MCGUIRE, W. A work control method for geometrically nonlinear analysis. **Proceedings of the 1985 International Conference on Numerical Methods in Engineering: Theory and Applications**, J. Middleton and G. N. Pande (eds.), pp. 913–921, 1985.

YANG, Y. B.; SHIEH, M. S. Solution method for nonlinear problems with multiple critical points. **AIAA Journal**, vol. 28, n. 12, pp. 2110–2116, 1990.

YANG, Y. B.; KUO, S. R. **Theory and Analysis of Nonlinear Framed Structures**. Prentice-Hall, Simon & Schuster (Asia) Pte ltd, Singapore, 1994.

ZIENKIEWICZ, O. C. **The Finite Element Method in Engineering Science**. McGraw-Hill, 1971.