4 Numerical methods for non linear analysis

During the previous chapters, the nonlinear theories and models for the analysis of continuum bodies and cable and frame structures have been exposed and discussed.

Following either the direct stiffness or finite element methods, one can always write the total or incremental equilibrium equations that represents the configuration of the structure in a certain moment t (Here, the parameter t plays the role of a parameter that characterizes the load and deformative path, not necessary related to the actual time). Using the two main formulations explained in this thesis, total or updated lagrangian formulation, the equilibrium equations in total or incremental forms can be obtained and solved to get the displacements, strains and stresses of the structure.

Due to the complexity of the systems treated, analytical techniques can not be applied unless the equations that govern the problem are simple, which only happens when load and boundary conditions are very simple. Advanced numerical techniques have to be developed to solve, or at least approximate, the equilibrium equations. In this chapter, we shall review the most commonly used nonlinear solution schemes. Comments will be provided for each method in terms of *stability* and *efficiency*.

The first method is the *Stiffness method*, which is the most intuitive method to solve the nonlinear equilibrium equations and particularly adapted to be used in the total lagrangian formulation. Then, some numerical procedures will be discussed considering the updated lagrangian formulation. The most simple, commonly used and oldest method is the so-called *Newton-Raphson method*. Others numerical schemes presented in this work are a variation of this method.

Finally, some numerical techniques to deal with high order nonlinear structures are presented. Due to the nonlinearity of certain structures, the Newton-Raphson scheme may not be enough to ensure the convergence of the method and more advanced control procedure are needed. The two procedures presented in this text are *displacement control procedure* and *work control procedure*, both adapted to deal with phenomena of *limit* *points* and *snap-back*. These two concepts will be explained in section 4.1.4.

4.1 Definition and solution of the non linear problem

Let us consider the general form of a nonlinear problem.

$$\boldsymbol{G} = \boldsymbol{f}(\boldsymbol{v}) - \boldsymbol{R}(\boldsymbol{v}) = \boldsymbol{0} \tag{4.1}$$

with $\boldsymbol{f}(\boldsymbol{v})$ the external load applied on the structure, $\boldsymbol{R}(\boldsymbol{v})$ the internal reaction of the structure and $\boldsymbol{G}$ the so-called *equilibrium function* which measures how far from the equilibrium the structures is, while $\boldsymbol{v}$ is the displacement vector of the structure.

The system of nonlinear equilibrium equations can be solved in many ways. From the easiest one to the most complex one, the different methods used in this thesis will be exposed and explained. The first one is the stiffness method, where the equilibrium equations are linearized to obtain the so-called *secant stiffness matrix*, which represents the stiffness of a virtual structure in the configuration of the real structure.

4.1.1 Stiffness method

Imagine a certain structure whose displacement-force path can be characterized as the one in figure 4.1. Considering the stiffness method, the principal goal is to describe each configuration of the structure by means of the secant stiffness matrix. Let us consider the general form of the nonlinear problem.

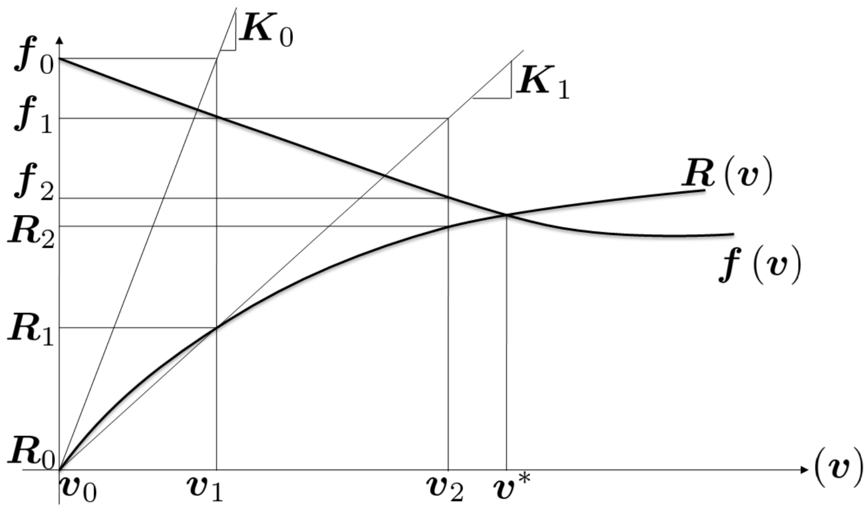


Figure 4.1: Secant iterative procedure to solve the equilibrium equations.

$$\boldsymbol{G} = \boldsymbol{f}(\boldsymbol{v}) - \boldsymbol{R}(\boldsymbol{v}) = \boldsymbol{0} \tag{4.2}$$

From figure 4.1, on each configuration of the structure the equilibrium equations can be rewritten as

$$\boldsymbol{K}(\boldsymbol{v})\,\boldsymbol{v} = \boldsymbol{f}(\boldsymbol{v}) \tag{4.3}$$

with $\boldsymbol{K}(\boldsymbol{v})$ the secant stiffness matrix.

In this chapter, the procedure to obtain the secant stiffness matrix will not be explained due to the fact that it depends completely on the tipology of the structure and it will be considered in chapter 6. For the moment, let us consider that the secant stiffness matrix as a function of the total displacement vector is known.

Under this hypothesis, a flow chart can be stablished (figure 4.2) , which is:

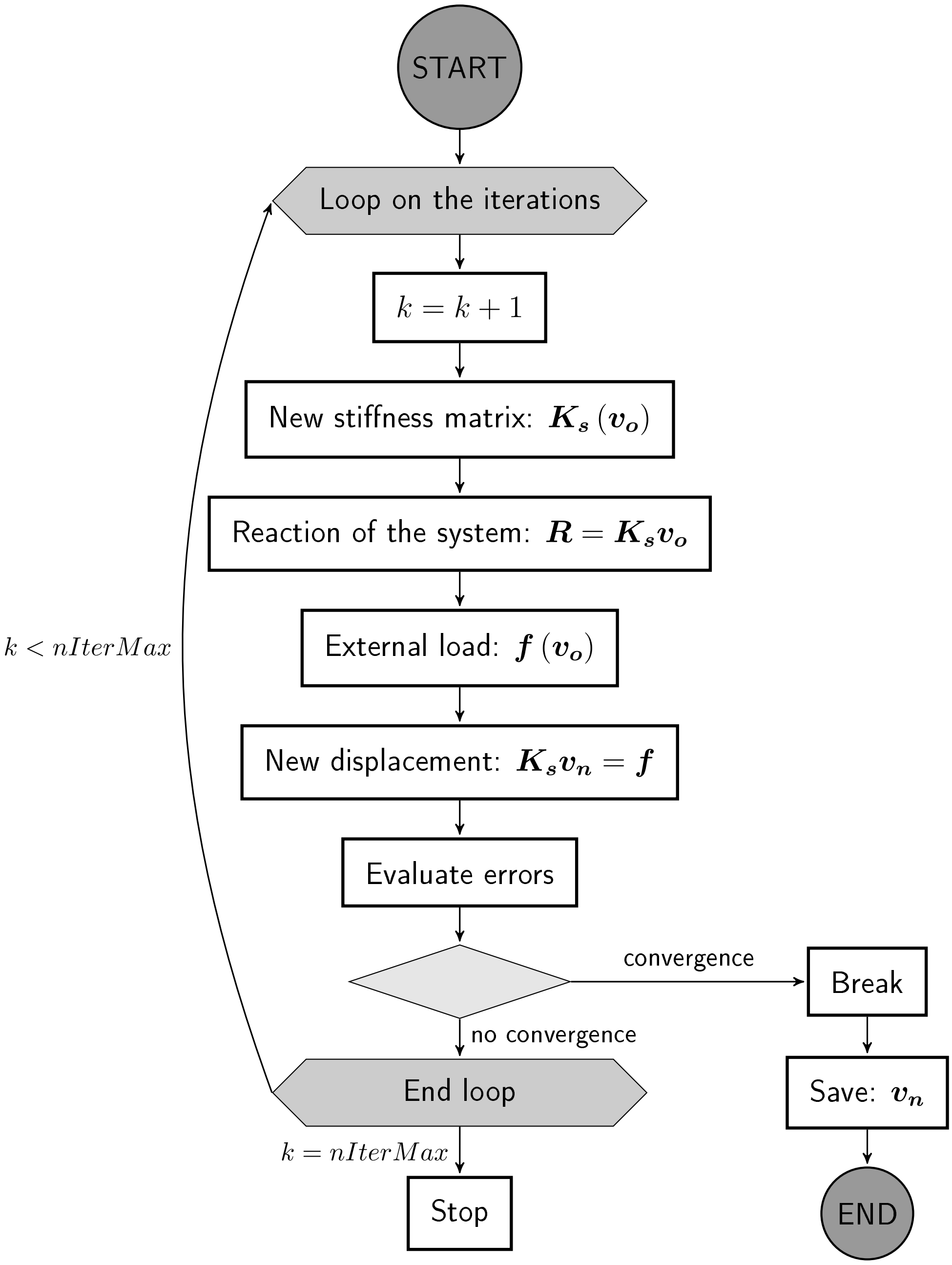


Figure 4.2: Flow chart of the stiffness method

As shown in the flow chart, the main goal of the stiffness method is the computation of the total secant stiffness matrix, which characterizes the response of an equivalent system to an external load. Once the stiffness matrix has been computed, the reaction of the system in the current configuration can be calculated as $\boldsymbol{R} = \boldsymbol{K_s v_o}$ (the current configuration is characterized by the vector $\boldsymbol{v_o}$). The external load, which in general will depend also on the displacements, can be computed considering $\boldsymbol{f}(\boldsymbol{v_o})$. Finally, the new displacement vector $\boldsymbol{v_n}$ can be computed and the convergence checking can be done. In this case, the convergence checking has been considered using the following three errors.

$$\begin{aligned} e_w &= \frac{\left|\boldsymbol{v}_{\boldsymbol{o}}^T\left(\boldsymbol{f}-\boldsymbol{R}\right)\right|}{\left|\boldsymbol{v}_{\boldsymbol{o}}^T\boldsymbol{f}\right|} &\leq & \; tol_w \\ e_d &= \frac{\left|\boldsymbol{v}_{\boldsymbol{n}}-\boldsymbol{v}_{\boldsymbol{o}}\right|}{\left|\boldsymbol{v}_{\boldsymbol{n}}\right|} &\leq & \; tol_d \\ e_f &= \frac{\left|\boldsymbol{f}-\boldsymbol{R}\right|}{\left|\boldsymbol{f}\right|} &\leq & \; tol_f \end{aligned} \tag{4.4}$$

which can be defined as *work error*, *displacement error* and *force error* respectively and they are demanded to be smaller than their corresponding tolerances.

4.1.2 Incremental load procedure

Althought the numerical scheme presented before has been adopted in this work in the total lagrangian formulation, it has limitations. In particular, the stiffness method is suitable to be used in structures that presents only a *softening* or *stiffening* (figure 4.3) behaviours. Explained with other words, the structure must behave monotonously.

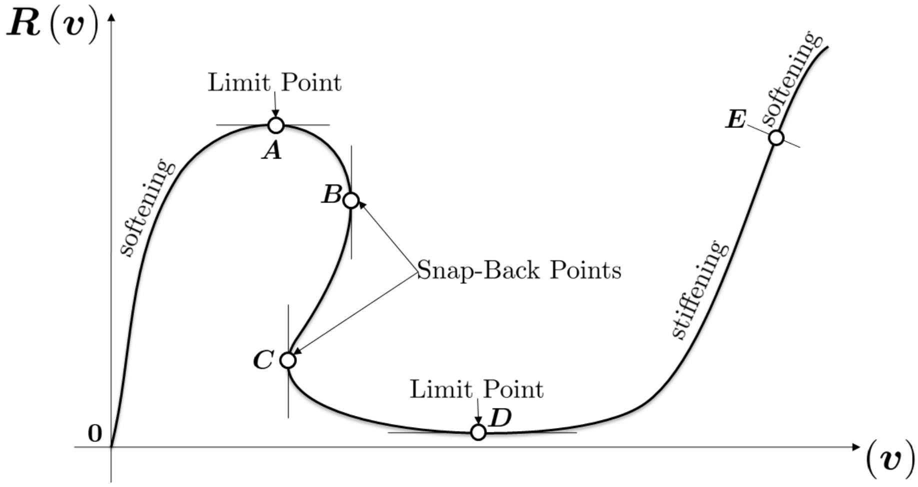


Figure 4.3: Possible behaviours of an structure during the deformative path

If the interest is the *displacement-force curve*, more advanced techniques have to be developed, able to overcome several pathologies that the structure could present during the deformative path (limit and snap-back points, stiffening and softening zones,...).

From now on, the numerical methods known as *Incremental-Iterative algorithms* will be presented. These algorithms consider the general expressions of the nonlinear problem and assume the load vector to be proportional to a certain *reference load vector*.

$$\boldsymbol{f}\left(\boldsymbol{v}\right) = \lambda \boldsymbol{f}_{\boldsymbol{r}} \rightarrow \boldsymbol{G} = \lambda \boldsymbol{f}_{\boldsymbol{r}} - \boldsymbol{R}\left(\boldsymbol{v}\right) = \boldsymbol{0} \tag{4.5}$$

These procedures are known as incremental-iterative because they increase the parameter λ from an initial to a final value and solve the incremental equilibrium equations for each load step. The classic Newton-Raphson scheme presented in the following section uses this technique and allows the study of the path-dependant behaviour of the structure in terms of displacements and load.

Althought the generality of the Newton-Raphson scheme, structures highly nonlinear require more advanced techniques, which allow the study of the behaviour of the structure around the critic points such as snap-back and limit points and are based on the *Displacement control method* or the *Work control method*, both explained in subsection 4.1.4.

Obviously, an ideal incremental scheme should allow the load increment to vary accordingly to the degree of nonlinearity as exemplified by the stiffness of the structure considered in order to optimize the efficiency of computation. This will be exposed in the last part of this chapter, where automatic algorithms to choose properly the load step will be reported.

4.1.3 Classic Newton-Raphson scheme

Let us consider the nonlinear equilibrium equations with the reference load vector

$$\lambda \boldsymbol{f_r} - \boldsymbol{R}(\boldsymbol{v}) = \boldsymbol{0} \tag{4.6}$$

and consider the situation in which the structure is in equilibrium at a certain configuration ${}^{\lambda}\boldsymbol{C}$ characterized by a value of the parameter λ. In this situation, the incremental equilibrium equation may be established in the new configuration ${}^{\lambda+\Delta\lambda}\boldsymbol{C}$. From now on, the notation will follow this rule: the left superscript index will denote the actual increment load and the left subscript index will indicate the number of iteration performed within the increment load. For example, ${}^{i}_{j}\boldsymbol{\Delta v}$ will be the increment of displacement computed in the iteration number j on the increment load number i.

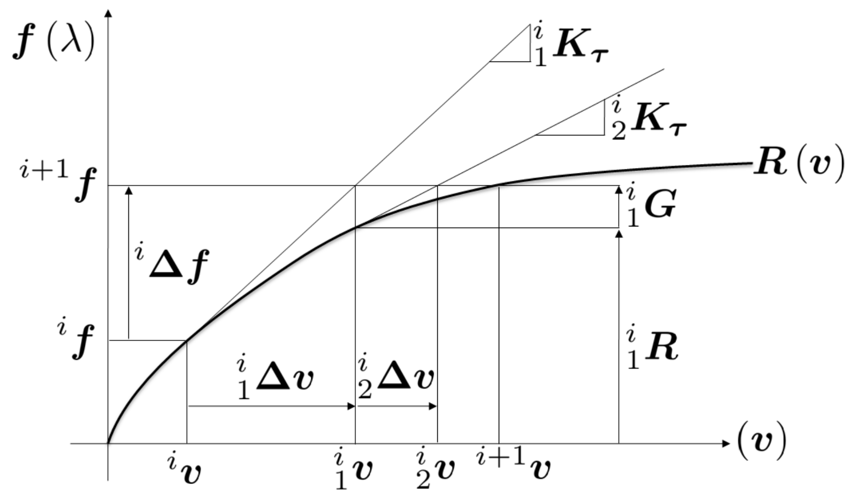


Figure 4.4: Tangent iterative procedure to solve the equilibrium equations.

Figure 4.4 represents the Newton-Raphson scheme in the configuration ${}^{i}\boldsymbol{C}$, where an increment of load has been imposed as ${}^{i+1}\boldsymbol{f} = {}^{i}\boldsymbol{f} + \Delta\lambda \boldsymbol{f_r}$ and the first two iterations in this new incremental load step are depicted and characterized by ${}^{i}_{1}\boldsymbol{K}$ (first tangent stiffness matrix), ${}^{i}_{1}\boldsymbol{\Delta v}$ (first increment of displacement), ${}^{i}_{1}\boldsymbol{R}$ (reaction of the system at the new incremental configuration) and ${}^{i}_{1}\boldsymbol{G}$ (Equilibrium functions in the incremental configuration).

The flow chart of the Incremental-Iterative Newtion-Raphson method can be written as:

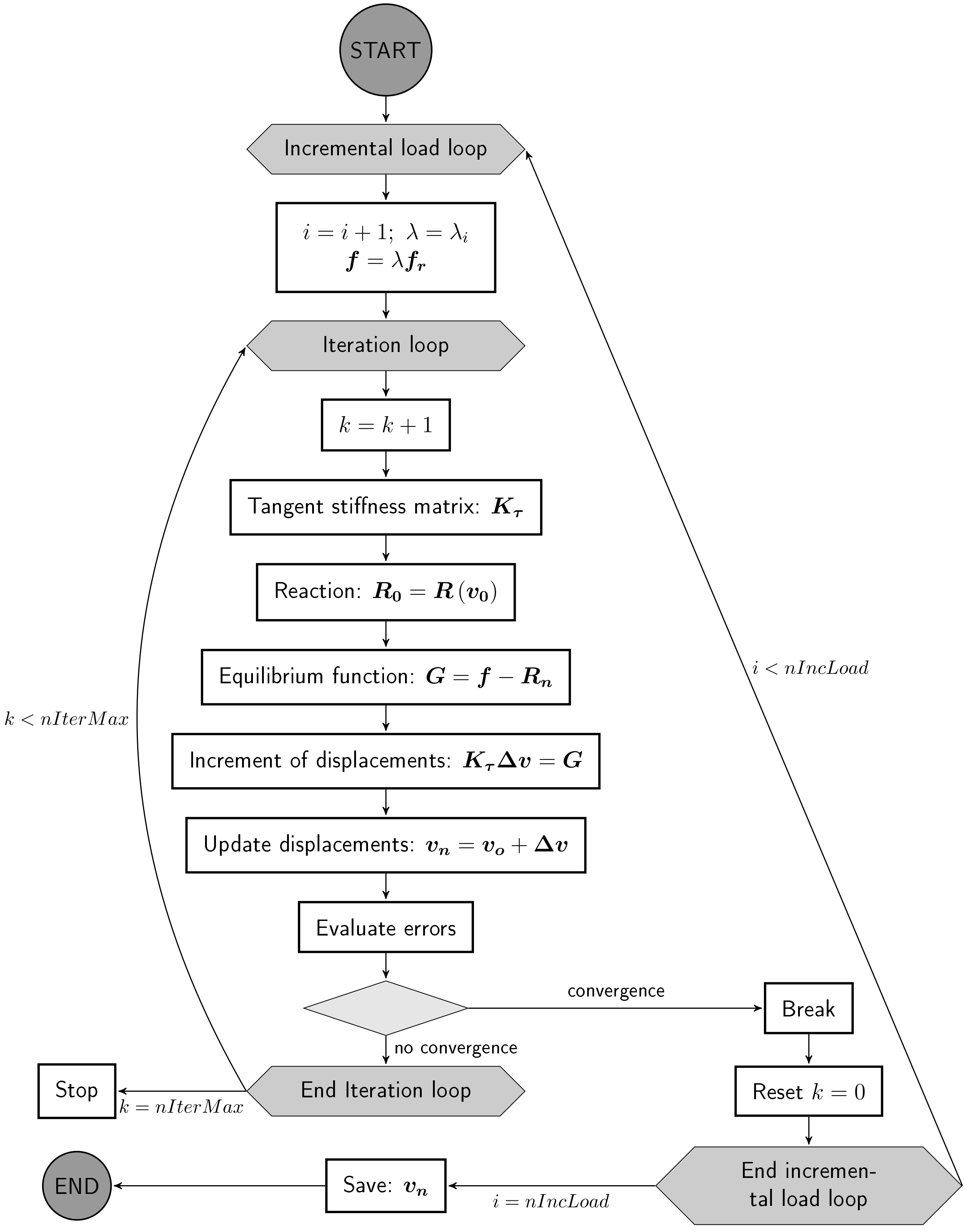


Figure 4.5: Flow chart of the incremental Newton-Raphson method

In the Newton-Raphson procedure, only the maximum number of iterations for each incremental load step has been considered as a control parameter. Indeed, in case the algorithm exceeds the maximum number of iterations without convergence, the program will stop and no outputs will be given. This may happen in situations where the structure does not present a monotonic behaviour (stiffering or softening) or where it presents some limit points or snap-back situations (figure 4.3). To deal with such an inconvenients, new techniques will be explained such as the *Arc Length* and *Work control* methods.

4.1.4 Improved Newton-Raphson schemes

Before describing the new methods, some considerations about new approaches that go beyond the analysis provided by the Newton-Raphson scheme must be said. This section will talk first about the *Prediction phase* and *Correction phase*. Both are vital concepts in the development of the new numerical techniques and will be deeply explained.

As stated previously, the nonlinear deformation process of a structure can be described by three typical configurations: the initial configuration ${}^0\boldsymbol{C}$, the last calculated configuration ${}^1\boldsymbol{C}$ and the current unknown configuration ${}^2\boldsymbol{C}$. To explain better the incremental iterative process in the following sections, a new notation will be used. From now on, the last calculated configuration will be denoted as ${}^i\boldsymbol{C}$ and the current unknown configuration as ${}^{i+1}\boldsymbol{C}$.

In the following, all the schemes presented will be divided in two parts. The first part, called prediction phase, will be devoted to compute the direction and amplitude of the load increment, while the second part (correction phase) will be devoted to perform some iterations to obtain the equilibrated system in the new configuration. Figure 4.6 shows the flow chart of the numerical method implemented.

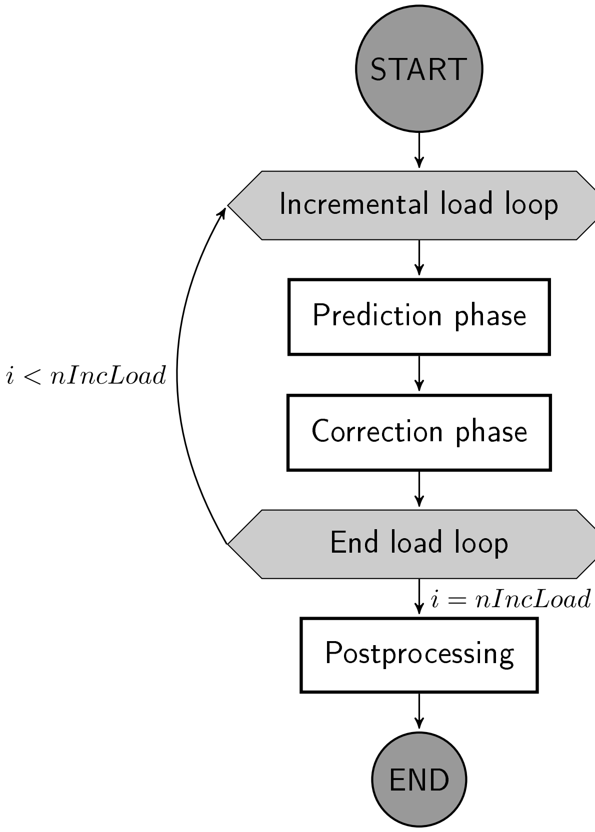


Figure 4.6: Flow chart of the improved Newton-Raphson schemes

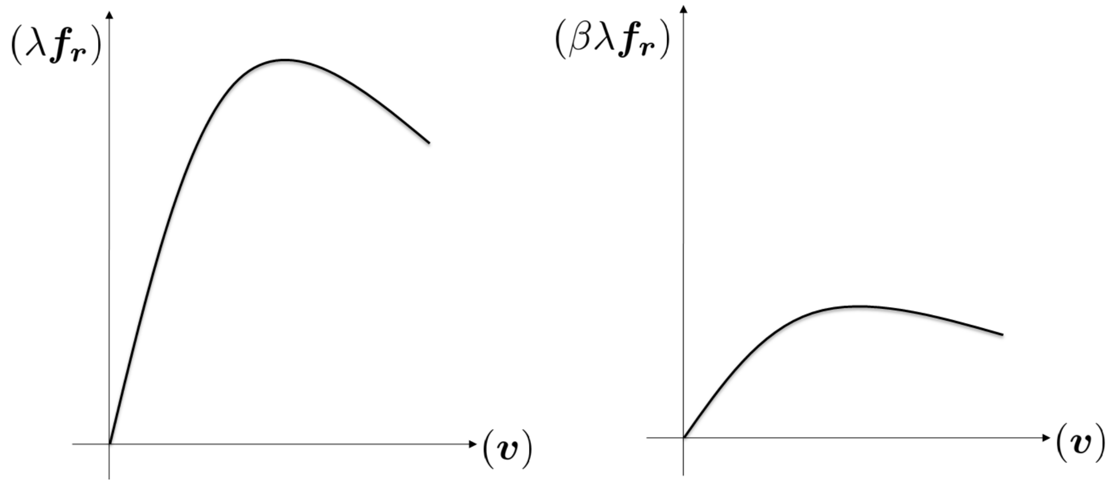
Before starting with the explanation of both the arc-length and work control methods, some definitions will be presented. First, the concept of normalization of the *force-displacement path* must be introduced. Then, the indicators that will be used to characterize the quality of the solution and convergence of the iterative method will be presented.

First, consider the force-displacement path depicted in figure 4.7. In a force-displacement path, the force and displacements are measured using different units. This may lead to different values of both variables. To avoid this scale problem, a normalization is required. From now on, the force-displacement path will be normalized and will be plotted in terms of the new variables

$$\begin{aligned} \boldsymbol{V} &= \boldsymbol{v} \\ \boldsymbol{F_r} &= \beta \boldsymbol{f_r} \end{aligned} \tag{4.7}$$

with

$$\beta = \frac{|\boldsymbol{v}|}{|\boldsymbol{f_r}|} \tag{4.8}$$



(a) Force-Displacement path without being normalized

(b) Normalized force-Displacement path

Figure 4.7: Normalization of the force-displacement curve

Second, in a general force-displacement path as the one in figure 4.3 on page 115, several zones can be characterized:

- Paths 0-A and D-E: It is an stable zone because an increment in the load has an inmediate increment of displacement and the stiffness matrix is definite positive.
- Paths A-B and C-D: It is an unstable zone because the stiffness matrix is not definite positive. This zone is also called *Pre-critic softening*. Usually the snap-back points are located in this zone.
- Path B-C: Unstable zone. The stiffness matrix is not positive definite. This zone is called *Post-critic softening*.

From the description below, a new parameter to describe the stiffness of the structure in a certain point of the force-displacement path is required. One of the most commonly used parameter is the *Bergam parameter*, also named as *current stiffness parameter* (CSP).

Let us consider the figure 4.8 where three configurations ${}^0\boldsymbol{C}$, ${}^1\boldsymbol{C}$ and ${}^2\boldsymbol{C}$ are depicted in the force-displacement path. Consider, from configuration ${}^0\boldsymbol{C}$ an increment of load equal to $\boldsymbol{\Delta f} = \beta^0\Delta\lambda \boldsymbol{f_r}$ that produces an increment of displacement equal to ${}^0\boldsymbol{\Delta v}$. The same can be establish in configuration ${}^1\boldsymbol{C}$ and ${}^2\boldsymbol{C}$ and in any other configuration defined in the deformative path. Each increment of displacement will produce, associated with the increment of load, a certain increment of work that can be defined as ${}^i\Delta w = \beta^i\Delta\lambda \boldsymbol{f_r}^{Ti}\boldsymbol{\Delta u}$. Consistently, the follow ratio can be defined

$$B = \frac{{}^0\Delta w/\left(\beta^0\Delta\lambda\right)}{{}^i\Delta w/\left(\beta^i\Delta\lambda\right)} = \frac{\boldsymbol{f_r}^{T0}\boldsymbol{\Delta v}}{\boldsymbol{f_r}^{Ti}\boldsymbol{\Delta v}} \tag{4.9}$$

This is the so-called *Bergam parameter* and indicates the ratio between the current stiffness and the stiffness in the initial state. It must be noticed that, for structures that become soft when the displacements increase, the Bergam parameter decreases (figure 4.3, 0-A path). On the other hand, the Bergam parameter could become negative and even $-\infty$ (figure 4.3 point B). Another definition for the Bergam parameter is the ratio between the increment of elastic energy at the initial configuration and the increment of elastic energy in the current configuration.

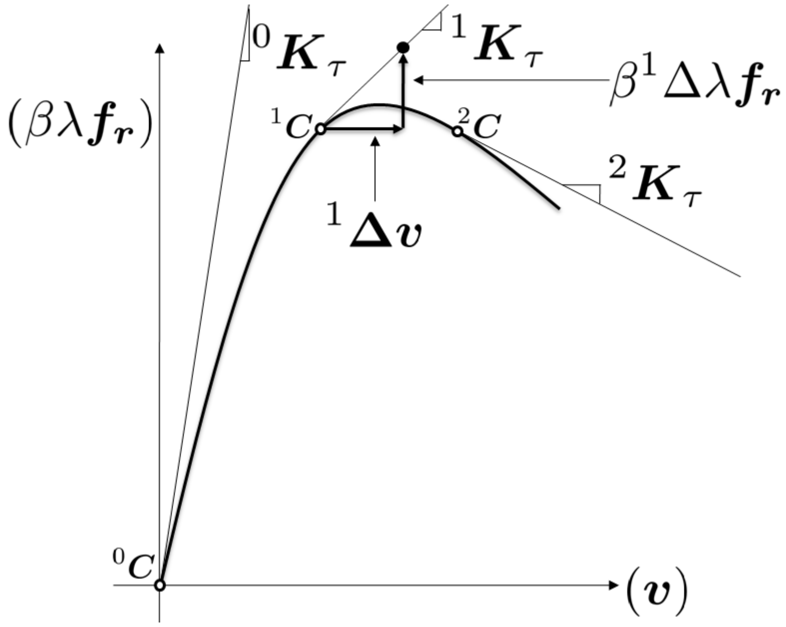


Figure 4.8: Definition of the Bergam parameter

Another important definition is the *Batoz and Dhatt notation*, which will be very useful in the explanation of the two procedures adopted. Imagine an increment of displacements produced in the current configuration $^i\boldsymbol{\Delta v}$. This vector has been obtained from the resolution of the linear system

$$^i\boldsymbol{K_T}\,^i\boldsymbol{\Delta v} = \Delta\lambda\boldsymbol{f_r} + {}^i\boldsymbol{G} \tag{4.10}$$

which can be divided in two contributions defined as

$$^i\boldsymbol{\Delta v} = \Delta\lambda^i\boldsymbol{\Delta v}_t + {}^i\boldsymbol{\Delta v}_r \tag{4.11}$$

$$^i\boldsymbol{\Delta v}_t = {}^i\boldsymbol{K_T}^{-1}\boldsymbol{f_r};\; {}^i\boldsymbol{\Delta v}_r = {}^i\boldsymbol{K_T}^{-1i}\boldsymbol{G} \tag{4.12}$$

The first displacement vector $^i\boldsymbol{\Delta v}_t$ represents the displacement that the structure would have if it was completely linear. On the other hand, the second term is derived from the equilibrium function $^i\boldsymbol{G}$.

The two methods will be discussed and their numerical implementation exposed in both the prediction and correction phase. Deep attention will be dedicated to the flowcharts of both methods.

4.1.4.1 Prediction phase

In the prediction phase, the initial iteration (prediction) is performed to obtain an approximation as the linearized solution of the equilibrium equations, of the incremental load step. Later, this solution will be "corrected" in the following phase by means of several methods.

In this phase, the direction and amplitude of the incremental load step will be defined. The need to define the direction of the increment of load is considered because, sometimes, the structure varies its stiffness during the deformative path and, if a force-displacement path analysis is performed, the increment of load has to be defined accordingly to the stiffness at the current configuration. For example, imagine that a certain structure, during its deformative process, is near to the point A in figure 4.3. At this point, the stiffness of the structure presents its first singularity and the structure is not able to counteract any increment[(1)] of load. Obviously, methods to take into account these singularities will be explained.

The prediction phase can be performed using two methods: *Arc-Length* and *Work control* methods. The explanation will start with the arc-length method.

Consider figure 4.9 where an iteration in the prediction phase is depicted. As it can be seen, the arc-length method controls the amplitude of the increment of displacements considering the equality

$$\Delta l^2 = {}^1\boldsymbol{\Delta v}^{T\,1}\boldsymbol{\Delta v} = \left({}^1\Delta\lambda\boldsymbol{\Delta v}_t + \boldsymbol{\Delta v}_r\right)^T\left({}^1\Delta\lambda\boldsymbol{\Delta v}_t + \boldsymbol{\Delta v}_r\right) \tag{4.13}$$

This equation can be solved in terms of the incremental load factor ${}^1\Delta\lambda$ as

$${}^1\Delta\lambda = \frac{\pm\Delta l}{\sqrt{\boldsymbol{\Delta v}_t^T\boldsymbol{\Delta v}_t + \beta^2\boldsymbol{f_r}^T\boldsymbol{f_r}}} \tag{4.14}$$

As it has been proved, the arc-length method imposes a geometric restriction in terms of the amplitude of the displacement vector. The parameter Δl is, indeed, non adimensional. This may lead to some problems in the definition of its value due to the fact that its optimum value might be different for each problem and the units that are used to describe the geometry, mechanical properties and forces. This is one of the disadvantages of the arc-length method.

To overcome this problem, another method will be considered, which is the work control method. In this case, a different restriction can be established in terms of the increment of work produced by the incremental load vector.

$$\overline{\Delta w} = {}^1\Delta\lambda\boldsymbol{f_r}^{T\,1}\Delta\lambda\boldsymbol{\Delta v}_t \tag{4.15}$$

[(1)]Here, the notation "increment" and "decrement" is considered as an intuitive way to understand what happens on the limit points because the plots are considered in a one-dimensional analysis. In general, the load will be increased or decreased by means of the load factor in a vectorial approach.

This equation can be seen as the restriction in the increment of external work (work produced by the increment of the load). Solving it in terms of $\Delta\lambda$ it can be obtained

$$^{1}\Delta\lambda = \pm\sqrt{\frac{\overline{\Delta w}}{\boldsymbol{f}_{\boldsymbol{r}}^{T}\boldsymbol{\Delta v}_{t}}} \tag{4.16}$$

As it can be seen in equations 4.14 and 4.16, the possible values for the variable $\Delta\lambda$ must be also characterized by its sign. To do so, the Bergam parameter can be used. Indeed, as it has been explained before, the Bergam parameter characterizes the stiffness of the structure in the current configuration referred to the initial configuration. This means that negative values of this parameter indicate the structure is not able to resist positive increments of load and a negative increment must be chosen to follow the deformative path of the structure. On the other hand, positive values of the Bergam parameter indicate the structure can yet resist positive increments of load and the variable $\Delta\lambda$ will be chosen as the positive one.

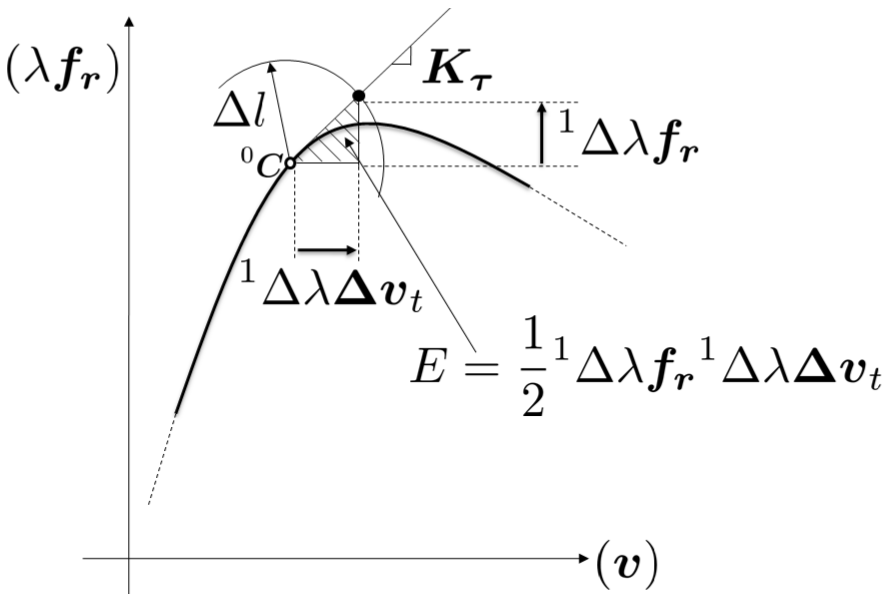


Figure 4.9: Iteration performed in the prediction phase

The flowchart of the prediction phase is therefore presented. It can be noticed the procedures to take into account the current stiffness of the structure by means of the Bergam parameter, the two methods (arc-length and work control) explained before and the choice between increasing or decreasing the load considering the sign of the Bergam parameter.

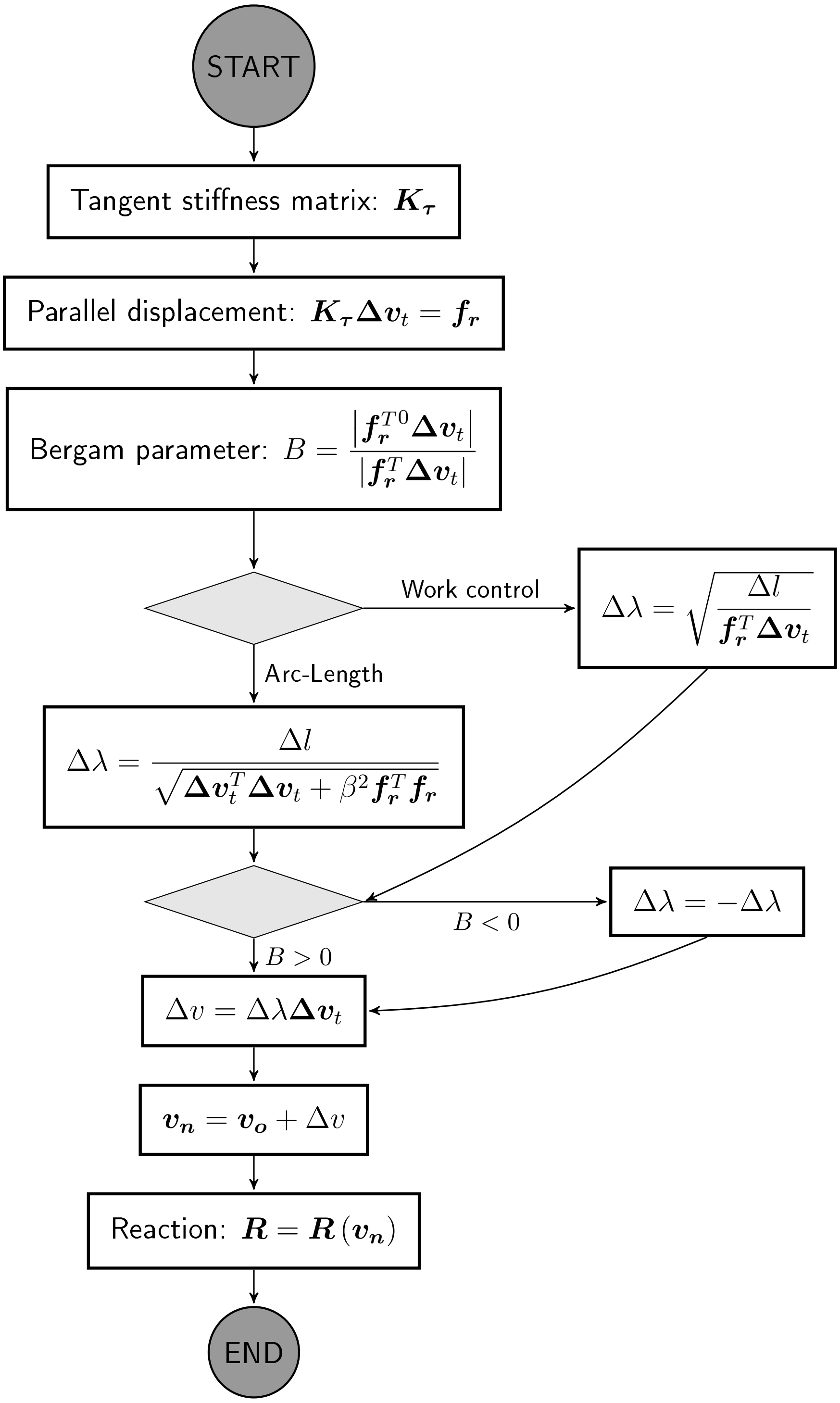


Figure 4.10: Flow chart of the prediction phase

4.1.4.2 Correction phase

In the correction phase, an iterative procedure is performed to satisfy the equilibrium conditions and to give a more accurate solution of the incremental load step. The situation depicted in figure 4.11is considered, where a prediction iteration has been performed and the actual configuration is not an equilibrated one because it does not lie on the force-displacement equilibrium path. To achieve the equilibrium, a correction scheme must be performed. Iteratively, a new incremental displacement vector $\boldsymbol{\delta v}$ will be computed and added to the already obtained $\boldsymbol{\Delta v}$. Using the notation of Batoz and Dhatt previously explained, it can be written

$$\boldsymbol{\delta v} = \boldsymbol{\delta v}_r + \delta\lambda \boldsymbol{\delta v}_t \tag{4.17}$$

$$\boldsymbol{\delta v}_r = \boldsymbol{K_\tau}^{-1}\boldsymbol{G};\ \boldsymbol{\delta v}_t = \boldsymbol{K_\tau}^{-1}\boldsymbol{f_r} \tag{4.18}$$

with the variables already defined as: $\boldsymbol{G}$ the equilibrium function, $\boldsymbol{f_r}$ the reference load vector, $\boldsymbol{K_\tau}$ the current tangent stiffness matrix, $\delta\lambda$ the correction increment of the load parameter. This means that, during the correction procedure, the displacements but also the external loads change to ensure the equilibrium in the current configuration.

The two methods proposed for the correction phase will be presented and discussed. The arc-length method is first introducced and after the Work control method will be discussed.

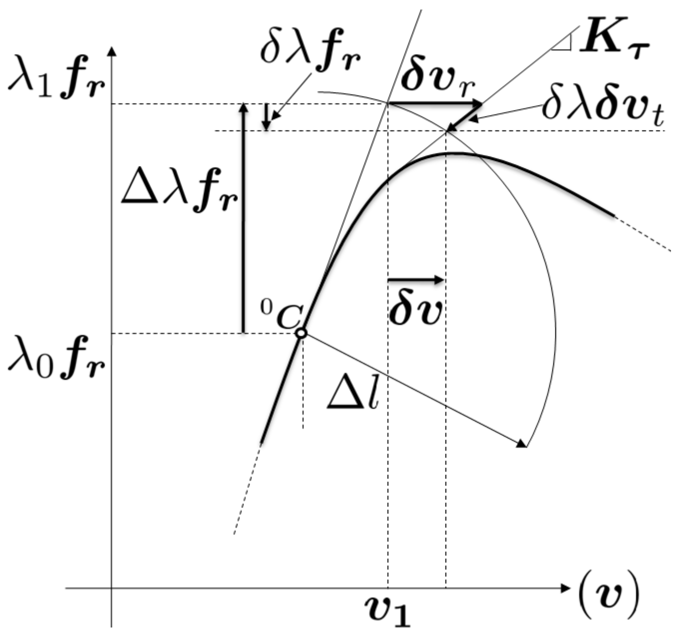


Figure 4.11: Correction phase during the nonlinear procedure

In the arc-length method, a general iteration in the correction procedure is characterized by the increment of load factor ${}^1\delta\lambda$ and the incremental displacement vector ${}^1\boldsymbol{\delta v}$. The arc-length method considers the following restriction on the displacements

$$\Delta l^2 = {}^2\boldsymbol{\Delta v}^{T2}\boldsymbol{\Delta v} + {}^2\Delta\lambda \boldsymbol{f_r}^{T2}\Delta\lambda \boldsymbol{f_r} \tag{4.19}$$

where the variables in the new configuration ${}^2\boldsymbol{C}$ are defined as

$$\begin{aligned} {}^2\boldsymbol{\Delta v} &= {}^1\boldsymbol{\Delta v} + {}^1\boldsymbol{\delta v} \\ {}^2\Delta\lambda &= {}^1\Delta\lambda + {}^1\delta\lambda \end{aligned} \tag{4.20}$$

Using equations 4.12 and 4.20, it mat be written

$$\left({}^1\boldsymbol{\Delta v}^T + {}^1\boldsymbol{\delta v}^T\right)\left({}^1\boldsymbol{\Delta v} + {}^1\boldsymbol{\delta v}\right) + \left(\beta^1\Delta\lambda + \beta^1\delta\lambda\right)^2 \boldsymbol{f_r}^T\boldsymbol{f_r} = \Delta l^2 \tag{4.21}$$

which can be transformed using equation 4.17 into

$$\left({}^1\boldsymbol{\Delta v}^T + {}^1\boldsymbol{\delta v}_r^T + \beta\delta\lambda\boldsymbol{f_r}^T\right)\left({}^1\boldsymbol{\Delta v} + {}^1\boldsymbol{\delta v}_r + \beta\delta\lambda\boldsymbol{f_r}\right) + \left(\beta^1\Delta\lambda + \beta^1\delta\lambda\right)^2 \boldsymbol{f_r}^T\boldsymbol{f_r} = \Delta l^2 \tag{4.22}$$

This equation can be transformed in a parabolic equation with the following coefficients

$$\begin{gathered} a^1\delta\lambda^2 + b^1\delta\lambda + c = 0 \\ a = \boldsymbol{\delta v}_t^T\boldsymbol{\delta v}_t + \beta^2\boldsymbol{f_r}^T\boldsymbol{f_r} \\ b = 2\boldsymbol{\delta v}_r^T\boldsymbol{\delta v}_r + 2^1\boldsymbol{\Delta v}^T\boldsymbol{\delta v}_t + 2^1\Delta\lambda\beta^2\boldsymbol{f_r}^T\boldsymbol{f_r} \\ c = -\Delta l^2 + {}^1\boldsymbol{\Delta v}^{T1}\boldsymbol{\Delta v} + \boldsymbol{\delta v}_r^T\boldsymbol{\delta v}_r + 2^1\boldsymbol{\Delta v}^T\boldsymbol{\delta v}_r + {}^1\Delta\lambda^2\beta^2\boldsymbol{f_r}^T\boldsymbol{f_r} \end{gathered} \tag{4.23}$$

Solving it, two increments of load factor may be obtained.

$${}^1\delta\lambda_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \tag{4.24}$$

For each increment, an incremental displacement vector can be defined as

$$\begin{gathered} {}^1\boldsymbol{\delta v}_1 = \boldsymbol{\delta v}_r + \delta\lambda_1\boldsymbol{\delta v}_t \\ {}^1\boldsymbol{\delta v}_2 = \boldsymbol{\delta v}_r + \delta\lambda_2\boldsymbol{\delta v}_t \end{gathered} \tag{4.25}$$

In this thesis, two methods have been used: *angle method* and *restoring method*.

The angle method consists on choosing the options whose angle with respect the incremental displacement vector defined in the prediction phase ${}^1\boldsymbol{\Delta v}$ is smaller. Consider the two cosines defined as

$$\cos\theta_1 = \frac{{}^1\boldsymbol{\Delta v}^{T1}\boldsymbol{\delta v}_1}{\left|{}^1\boldsymbol{\Delta v}\right|\left|{}^1\boldsymbol{\delta v}_1\right|};\ \cos\theta_2 = \frac{{}^1\boldsymbol{\Delta v}^{T1}\boldsymbol{\delta v}_2}{\left|{}^1\boldsymbol{\Delta v}\right|\left|{}^1\boldsymbol{\delta v}_2\right|} \tag{4.26}$$

This method is easily depicted in figure 4.12, where the two options with their corresponding cosines are considered.

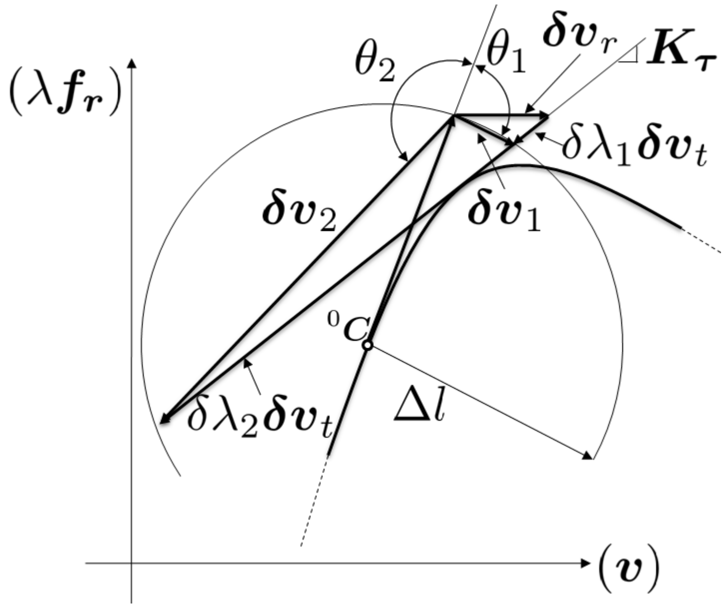


Figure 4.12: Angle method in the arc-length correction procedure

This approach is considered to be bery robust and efficient, but sometimes, when the structure presents strong snap-back conditions, the method is not able to converge and unefficiently small arc parameter Δl must be used.

As a consequence, a second method is presented to deal with situations where the angle method is not able to catch the force-displacement path of the structure. Instead of choosing the incremental displacement vector as done in the angle method, the condition is imposed in the reaction of the two solutions of the system. For example, consider the two solutions of the correction phase $\boldsymbol{\delta v}_1$ and $\boldsymbol{\delta v}_2$ which have been obtained solving equation 4.23.

$$\begin{aligned} {}^2\boldsymbol{v}_1 &= {}^1\boldsymbol{v} + {}^1\boldsymbol{\Delta v} + {}^1\boldsymbol{\delta v}_1 \\ {}^2\boldsymbol{v}_2 &= {}^1\boldsymbol{v} + {}^1\boldsymbol{\Delta v} + {}^1\boldsymbol{\delta v}_2 \end{aligned} \tag{4.27}$$

For each new configuration, the reactions of the system can be computed and compared on each case.

$$\begin{aligned} {}^2\boldsymbol{v}_1 \rightarrow {}^2\boldsymbol{R}_1 \rightarrow {}^2\boldsymbol{G}_1 \\ {}^2\boldsymbol{v}_2 \rightarrow {}^2\boldsymbol{R}_2 \rightarrow {}^2\boldsymbol{G}_2 \end{aligned} \tag{4.28}$$

The restoring method considers the solution as the one which equilibrium function has the lowest modulus. In other words, the solution chosen between the two possibles will be the one whose reaction is closer to the external load. See figure 4.13 for more details.

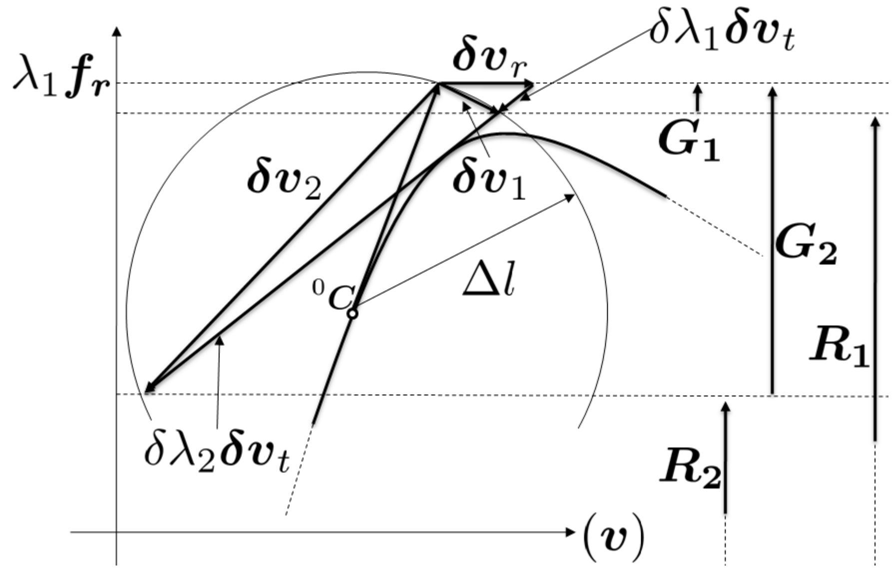


Figure 4.13: Sketch of the choice made by the restoring method in the arc-length procedure

The arc-length method has been described and two possible alternatives have been discussed. Now, the attention is focused on the second method proposed in this thesis, the Work control method.

The restriction that the work control method imposes to the displacements is that the work produced by the increment of force with the new incremental vector in the correction phase must be null. This leads to

$$\delta w = {}^1\delta\lambda \boldsymbol{f_r}^{T\,1}\boldsymbol{\delta v} = 0 \tag{4.29}$$

Using equation 4.17, the equation can be rewritten as

$${}^1\delta\lambda \boldsymbol{f_r}^T \left(\boldsymbol{\delta v}_r + {}^1\delta\lambda \boldsymbol{\delta v}_t\right) = 0 \tag{4.30}$$

This equation has two possible solutions. The first one is the trivial solution, which leads to a null displacement increment (this solution is not interesting for the procedure). On the other hand, the second solution gives an incremental displacement vector that is perpendicular to the reference load vector. Solving equation 4.30 with respect to ${}^1\delta\lambda$, it can be obtained

$${}^1\delta\lambda = -\frac{\boldsymbol{f_r}^T \boldsymbol{\delta v}_r}{\boldsymbol{f_r}^T \boldsymbol{\delta v}_t} \tag{4.31}$$

The correction phase is performed until the reaction of the system in the current configuration equals the external load (or when the equilibrium functions goes under the fixed tolerance) and the increment of displacement is small enough. At each iteration of this process, the two methods exposed may be used to achieve the equilibrium.

In order to clarify the numerical procedure, two flow charts will be presented. The first one corresponds to the flow diagram of the correction procedure with the two methods discussed before (arc-length and work control methods in figure 4.14). The second diagram focuses on the arc-length methods (figure 4.15) because the number of operations required with this method is larger than the work control method.

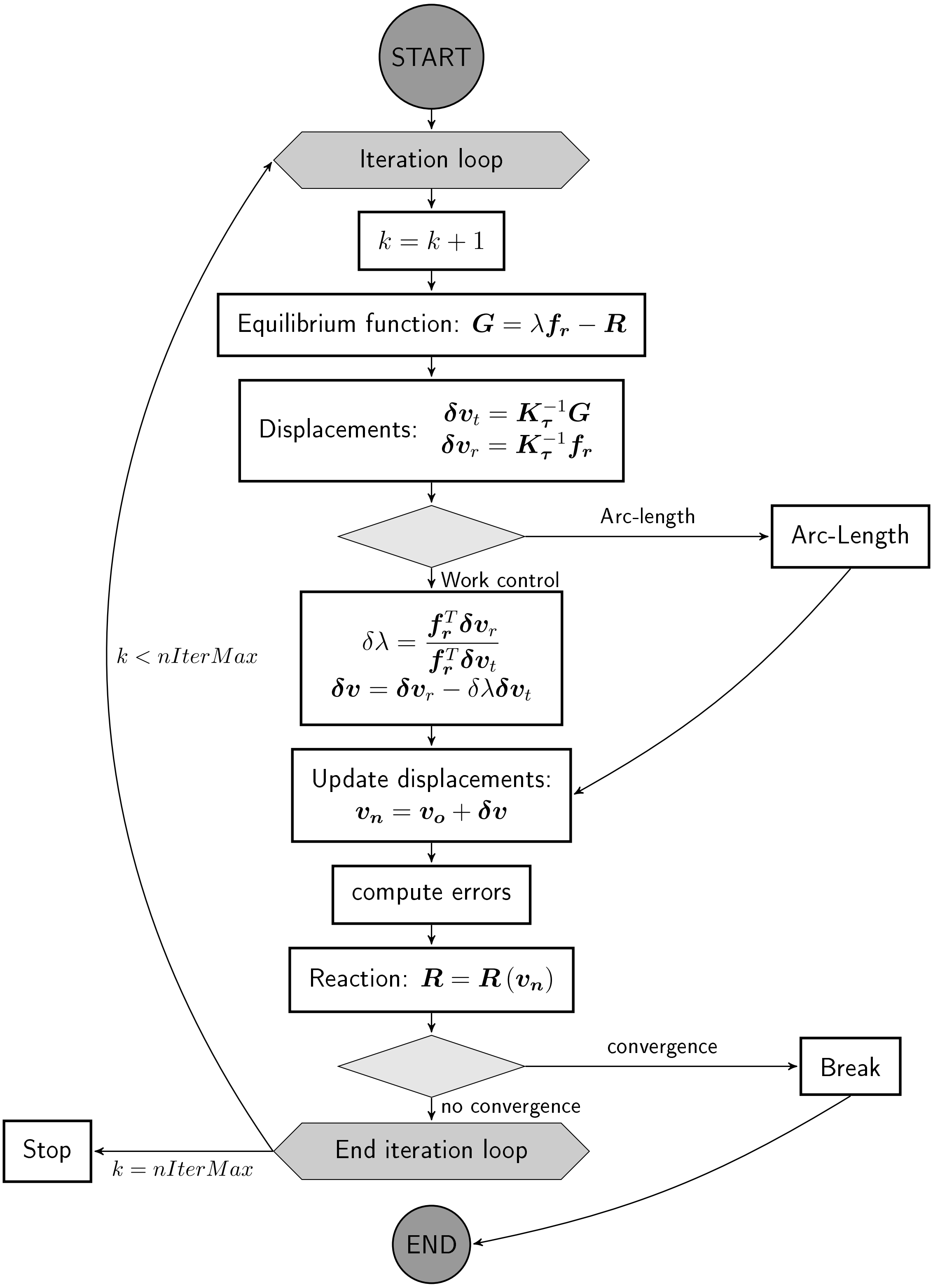


Figure 4.14: Flow chart of the correction phase

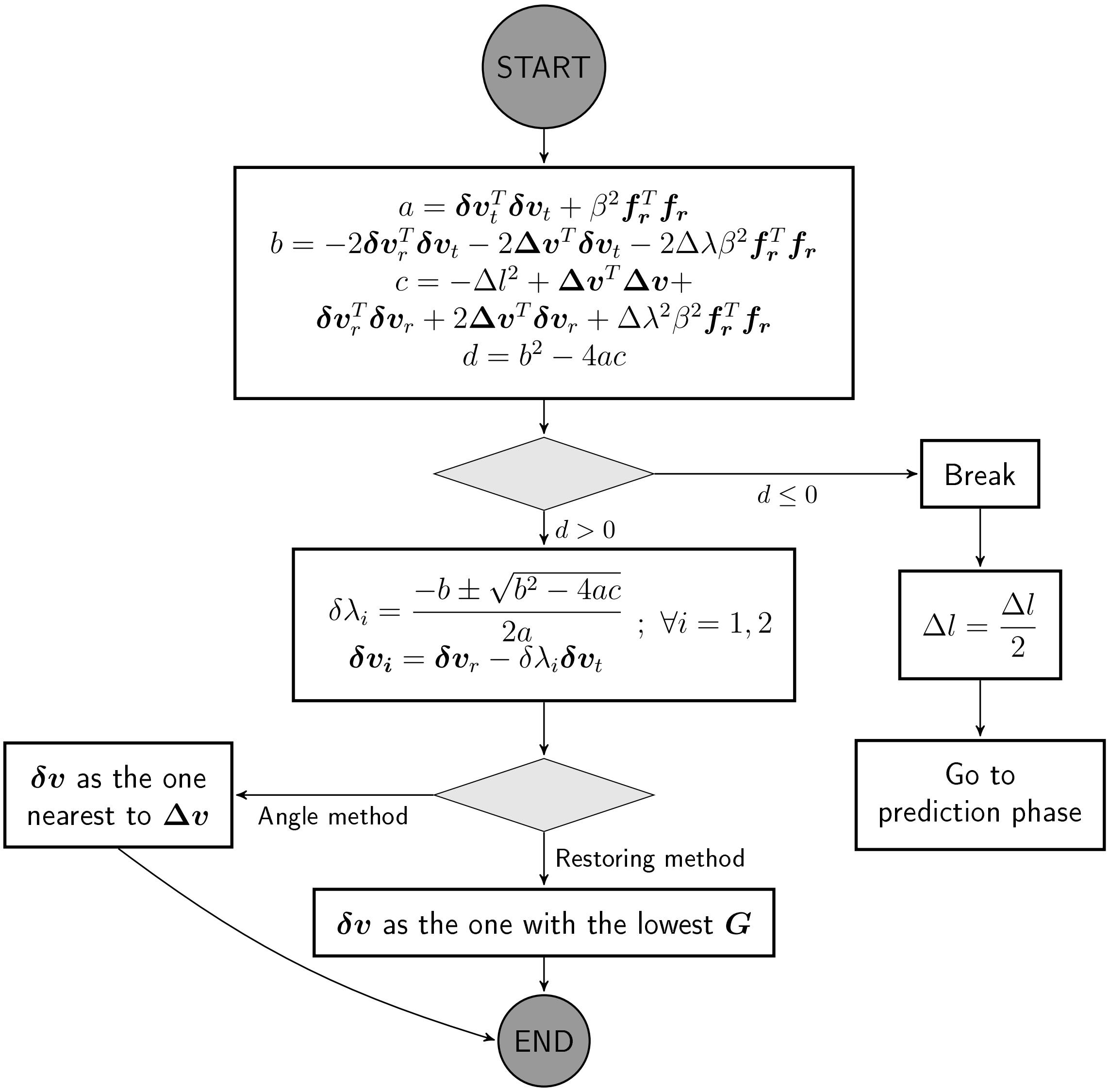


Figure 4.15: Flow chart of the Arc-Length method in the correction phase

4.2 Computational aspects

The different diagrams developed in the prediction and corrections phases might be used together to develop a numerical code to deal with nonlinear structural problems. The diagrams 4.10 and 4.14 are the two most important because define the procedure to follow in the prediction and correction phase, respectively.

Some numerical and computational aspects will be discussed and some procedures to update the performance and efficiency of the code will be provided. The *Switching* procedure will be discussed first. Later, a technique to modify the code to be able to define the load increment automatically will be provided and dicussed in terms of convergence problems.

4.2.1 Switching

Considering figure 4.3, where a general force-displacement path is depicted, the path 0-A has been considered as a stable zone because the stiffness was positive defined (The Bergam parameter has positive values). This part of the path can be faced using a classic Newton-Raphson scheme but, when the zone close to the A point is reached, the scheme may diverge. To solve this problem, a new procedure is presented to deal with. This new procedure, called *Switching procedure*, allows the program to jump from the classic Newton-Raphson to the more advanced Arc-length or work control methods during the iteration process.

To do so, the procedure is based on the Bergam parameter. The user may define a critic Bergam parameter value from which the Switching procedure may be activated. This check in the program must be included in both the prediction and correction phases considering the following flowchart.

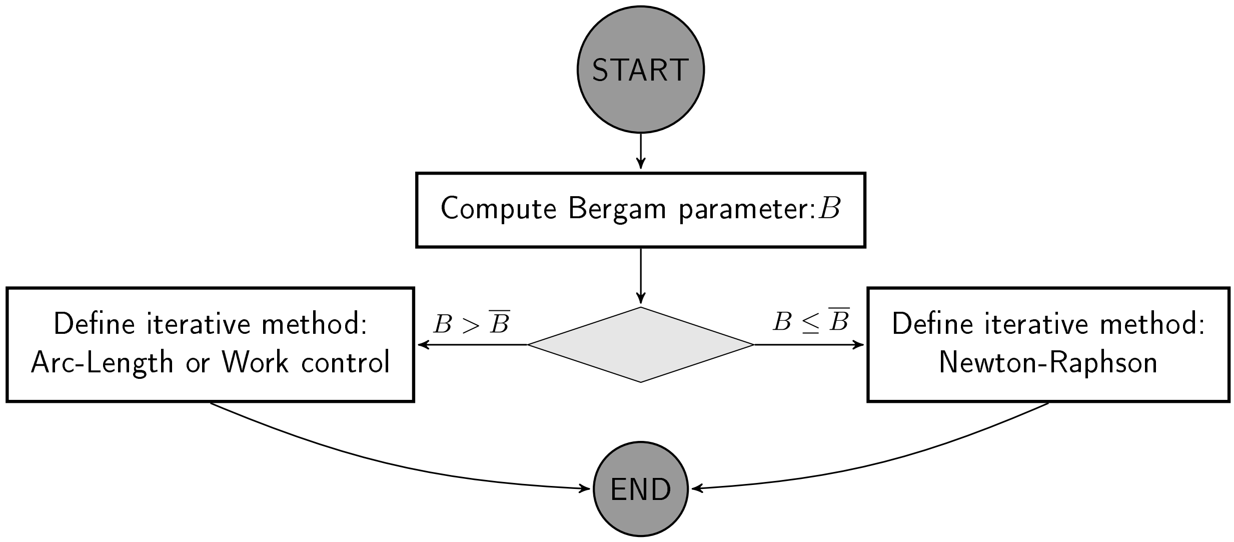


Figure 4.16: Flow chart of Switching procedure

4.2.2 Automatic load increment

Sometimes, the program performs many iterations in the correction phase due to the high nonlinearities of the structure. From the computational point of view, the convergence difficulty may be defined as the number of iterations performed on each correction phase. For example, if the iterations performed are few, the program does not find any difficulty to compute the solution. On the other hand, a large number of iterations may indicate that the increment of load parameter computed in the prediction phase is too big (in case a classic Newton-Raphson scheme is being used) or that the arc length parameter used in the correction phase is not adequate (in case an arc-length procedure is being used).

Let us define I_o as the optimum number of iterations introduced in the code by the user. In addition, I_i is defined as the number of iterations performed in the correction phase number i. Obviously, if the number of iterations performed in the current correction phase have been bigger the the optimum one, the program could adjust the increment of load factor or the arc-length parameter to take into account the local nonlinearity of the structure responds.

In this thesis, the method proposed to update both the increment of load factor and arc-length parameter is the following

$$^{i+1}\Delta\lambda = {}^{i}\Delta\lambda\sqrt{\frac{I_d}{I_i}};\ {}^{i+1}\Delta l = {}^{i}\Delta l\sqrt{\frac{I_d}{I_i}} \tag{4.32}$$

The following flowchart (figure 4.17) explains where the previous modification must be implemented.

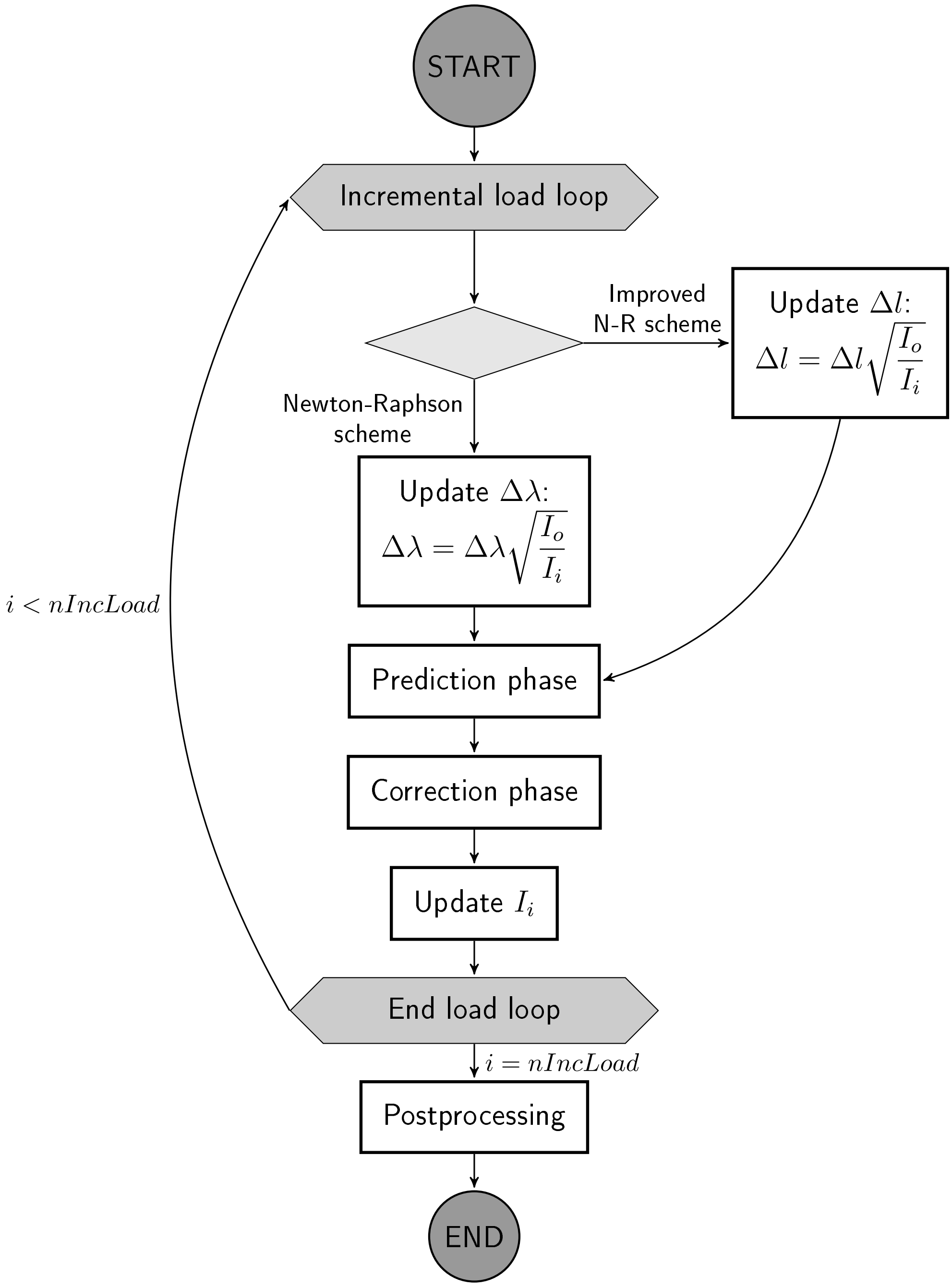


Figure 4.17: Flow chart of the automatic load increment procedure

4.2.3 Preconditioning and resolution of the linear system

Consider the linear system

$$\boldsymbol{K}\boldsymbol{u} = \boldsymbol{f} \tag{4.33}$$

If the entries of $\boldsymbol{K}$ vary greatly in size, it is likely that during the elimination process large entries are summed to small entries, with a consequent onset of rounding errors. To study this, the condition number of the matrix $\boldsymbol{K}$ must be evaluated.

By definition, the condition number $C(\boldsymbol{K})$ of a matrix $\boldsymbol{K}$ can be defined as

$$C(\boldsymbol{K}) = |\boldsymbol{K}| \left|\boldsymbol{K}^{-1}\right| \tag{4.34}$$

It can be proved (see [Quarteroni, 2000]) that for symmetric definite positive matrices, the expression of the condition number reduces to the ratio between the biggest and the smallest eigenvalue of the matrix, which are expressed as λ_M and λ_m respectively.

$$C(\boldsymbol{K}) = \frac{\lambda_M}{\lambda_m} \tag{4.35}$$

It is well known that the condition number of a matrix affects the stability[(2)] of the linear system. This sentences can be expressed mathematically considering the following formula.

$$\frac{\boldsymbol{\delta u}}{\boldsymbol{u}} \leq \frac{C(\boldsymbol{K})}{1 - C(\boldsymbol{K})} \frac{\boldsymbol{\delta f}}{\boldsymbol{f}} \tag{4.36}$$

In this thesis, a method to precondition the amtrix of the linear system on each iteration of the process has been implemented. Consider a diagonal matrix defined as follows

$$\boldsymbol{D} = \begin{bmatrix} \beta^{r_1} & & \\ & \ddots & \\ & & \beta^{r_n} \end{bmatrix} \tag{4.37}$$

and the preconditioned system

$$\underbrace{\boldsymbol{DKD}}_{\boldsymbol{A}}\underbrace{\boldsymbol{D}^{-1}\boldsymbol{u}}_{\boldsymbol{x}} = \underbrace{\boldsymbol{Df}}_{\boldsymbol{b}} \tag{4.38}$$

This new system can be solved and the vector $\boldsymbol{x}$ obtained. Then, the real vector $\boldsymbol{u}$ can be obtained from $\boldsymbol{x}$ as

$$\boldsymbol{u} = \boldsymbol{Dx} \tag{4.39}$$

An important matter is how to choose the matrix $\boldsymbol{D}$. In this work, the method used to obtained the matrix is imposing a restriction in the diagonal terms of the matrix $\boldsymbol{A}$, which is

$$\forall i = 1, \ldots, n \quad 1 \leq |\boldsymbol{A}_{ii}| \leq \beta \tag{4.40}$$

Considering that the diagonal entries of the matrix $\boldsymbol{A}$ can be obtained in terms of the diagonal elements of the matrix $\boldsymbol{K}$ as

$$\boldsymbol{A}_{ii} = \beta^{2r_i} \boldsymbol{K}_{ii} \tag{4.41}$$

(2) Here, the term stability is considered as how much the variation of the results $\boldsymbol{\delta u}$ is affectted by a variation in the input of the system $\boldsymbol{\delta f}$

the equation 4.40 leads to the following two conditions on the coefficients r_i.

$$r_m = -\frac{\ln(|\boldsymbol{K}_{ii}|)}{2\ln(\beta)} \leq r_i \leq \frac{\ln(\beta) - \ln(|\boldsymbol{K}_{ii}|)}{2\ln(\beta)} = r_M \tag{4.42}$$

Considering that these two inequalities are always satisfied nomatter the value of $\boldsymbol{K}_{ii}$, a solution that satisfies 4.42 can be found as the medium between r_m and r_M.

$$r_i = \frac{\ln(\beta) - 2\ln(|\boldsymbol{K}_{ii}|)}{4\ln(\beta)} \tag{4.43}$$

The method previously exposed allows to precondition the linear system. During the numerical analysis, this technique has been proved to be very effective.

In addittion, an iterative algorithm has been also implemented to deal with ill-conditioned systems when the method previously explained is not able to completely condition the system. The method implemented is called iterative refinement.

When the matrix of a linear system is ill-conditioned and the technique to condition it fails, the iterative refinement technique can be applied to solve the system. Consider the system 4.33 with an ill-conditioned matrix $\boldsymbol{A}$. The iterative refinement is a technic for improving the accuracy of a solution yielded by a direct method. Suppose that the linear system has been solved, and denote $\boldsymbol{x}^0$ the first solution. Having fixed a tolerance *tol* the iterative refinement performs as follows: for $i = 1, \ldots,$ untill convergence:

1. Compute residual vector: $\boldsymbol{r}^{(i)} = \boldsymbol{f} - \boldsymbol{K}\boldsymbol{x}^{(i)}$
2. Solve the linear system $\boldsymbol{K}\boldsymbol{z} = \boldsymbol{r}^{(i)}$
3. Update solution: $\boldsymbol{x}^{(i+1)} = \boldsymbol{x}^{(i)} + \boldsymbol{z}$
4. If $\dfrac{|\boldsymbol{z}|}{|\boldsymbol{x}^{(i+1)}|} \leq tol$ and $\dfrac{|\boldsymbol{r}^{(i)}|}{\boldsymbol{f}} \leq tol$ the procedure can stop, otherwise start at step 1.

This procedure allows to solve ill-conditioned systems with relatively well stability. For more details see [Quarteroni, 2000].