HAL
open science

Implementation of path following techniques into the finite element code LAGAMINE

Panagiotis Kotronis, Frédéric Collin

To cite this version:

Panagiotis Kotronis, Frédéric Collin. Implementation of path following techniques into the finite element code LAGAMINE. [Research Report] Rapport Interne Géomac/3S-R. 2005. hal-00100402

HAL Id: hal-00100402

https://hal.science/hal-00100402v1

Submitted on 26 Sep 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Internal report GéomaC/3S, September 2005

IMPLEMENTATION OF PATH FOLLOWING TECHNIQUES INTO THE FINITE ELEMENT CODE LAGAMINE

P. KOTRONIS AND F. COLLIN

Abstract. This report presents a study on existing advanced incremental-iterative solution techniques for geometrically and physically non-linear analysis and their implementation into the finite element code LAGAMINE. After a theoretical presentation of different path following techniques, a general algorithm and details about the specific implementation into LAGAMINE follow. Challenging examples show the advantages but also the efficiency of each method.

1. Introduction

Geometrically or physically non-linear problems are often characterized by the presence of critical points with snapping behavior in the structural response. Conventional Newton-type iterative strategies hold the load parameter constant whilst iterating to convergence and thus often fail to reproduce structural or material instabilities. This report presents a study on existing automatic following path techniques and their implementation into the finite element code LAGAMINE (finite element code developed at the department 'GEOMAC' of the University of Liège under the direction of Prof. R. Charlier).

2. Newton-Raphson Procedure

2.1. **An Incremental-Iterative Strategy ?** The classical finite element discretization process yields the following set of simultaneous equations [5],[20],[44]:

$$[K]\{\delta\} = \{F_{ext}\} \ , \tag{2.1}$$

where for structural analysis $[K]$ is the stiffness matrix, $\{\delta\}$ is the vector of DOF (degrees of freedom) and $\{F_{ext}\}$ the vector of applied loads. If the stiffness matrix $[K]$ is itself a function of the DOF values (or their derivatives) then eq.(2.1) is a nonlinear equation. The Newton-Raphson method is an *iterative process* of solving the nonlinear equations and can be written as:

$$\left[K^{tg}\right]^{j-1}\{\Delta\delta\}^{j} = \{F_{ext}\} - \{F_{int}\}^{j-1} \ , \tag{2.2}$$

$$\{\delta\}^{j} = \{\delta\}^{j-1} + \{\Delta\delta\}^{j} \ , \tag{2.3}$$

Key words and phrases. arc-length method, snap-back, snap-through, bifurcation, non-uniqueness, geometrically and physically non-linear analysis, path following technique, automated solution control .

Panagiotis KOTRONIS is grateful for the FNRS (Fonds National pour la Recherche Scientifique - Belgium) research fellowship obtained for his two month visit at the department 'GEOMAC' of the University of Liège.

where $[K^{tg}]^{j-1}$ is the tangent stiffness matrix, j is the subscript representing the equilibrium iteration and $\{F_{int}\}^{j-1}$ is the vector of restoring loads corresponding to the element internal loads. Both $[K^{tg}]^{j-1}$ and $\{F_{int}\}^{j-1}$ are evaluated based on the values given by $\{\delta\}^{j-1}$. The right-hand side of eq.(2.2) is the residual or out-of-balance load vector; i.e., the amount the system is out of equilibrium. A single solution iteration for a one DOF model is depicted in Figure 1.

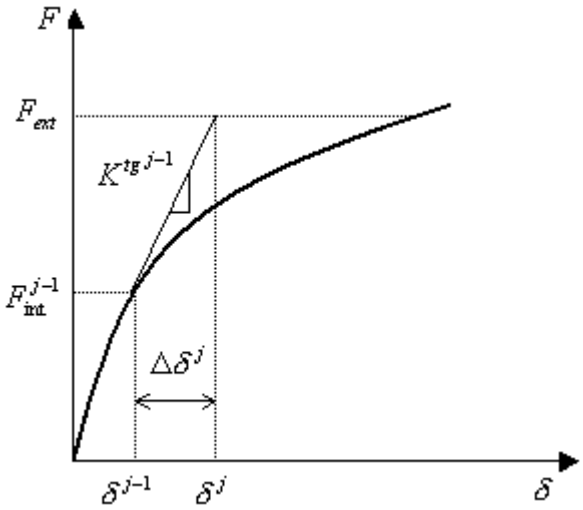

FIGURE 1. *Newton-Raphson procedure: One iteration for a problem with a single DOF.*

It is obvious that *more than one iteration is needed* in order to obtain a converged solution. The general algorithm proceeds as follows [2], Figure 2:

(1) Compute the updated tangent stiffness matrix $[K^{tg}]^{j-1}$ and the internal loads $\{F_{int}\}^{j-1}$ from configuration $\{\delta\}^{j-1}$;
(2) Calculate $\{\Delta\delta\}^j$ from eq.(2.2);
(3) Add $\{\Delta\delta\}^j$ to $\{\delta\}^{j-1}$ in order to obtain the next approximation $\{\delta\}^j$ from eq.(2.3);
(4) Repeat steps 1 to 4 until convergence is obtained.

The solution obtained at the end of the iteration process would correspond to load level $\{F_{ext}\}$. The final converged solution would be in equilibrium, such that the internal loads vector $\{F_{int}\}^{j-1}$ would equal to the applied load vector $\{F_{ext}\}$ (or at least within some tolerance). None of the intermediate solutions would be in equilibrium.

If the analysis included path-dependent nonlinearities (such as damage or plasticity), then the solution process requires that some intermediate steps be in equilibrium in order to correctly follow the load step. This is accomplished effectively by specifying a *step-by-step incremental analysis*; i.e., the final load vector $\{F_{ext}\}$ is reached by *applying the load in increments and performing the Newton-Raphson iterations at each step*, Figure 3:

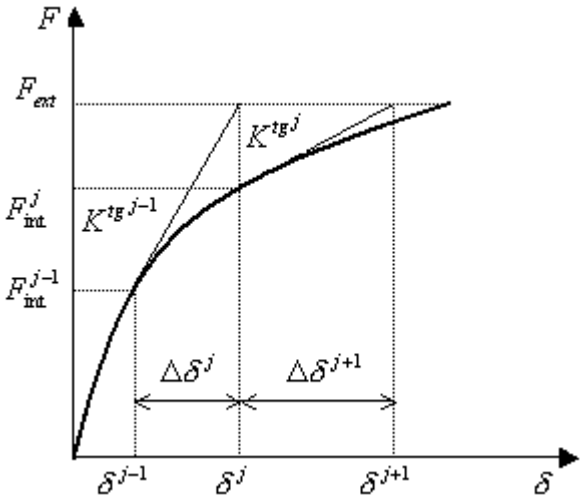

FIGURE 2. *Newton-Raphson procedure: Several iterations within a step for a problem with a single DOF.*

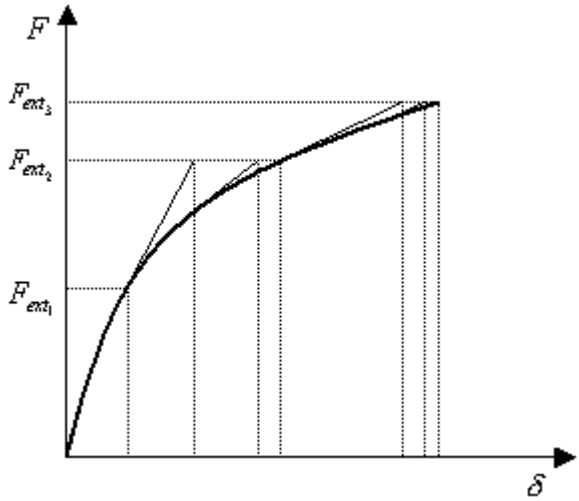

FIGURE 3. *Full Newton-Raphson procedure (single DOF problem).*

$$\left[K^{tg}\right]_i^{j-1}\{\Delta\delta\}_i^j = \{F_{ext}\}_i - \{F_{int}\}_i^{j-1} \ , \tag{2.4}$$

$$\{\delta\}_i^j = \{\delta\}_i^{j-1} + \{\Delta\delta\}_i^j \ , \tag{2.5}$$

with $[K^{tg}]_i^{j-1}$ the tangent stiffness matrix for time step i, iteration j, $\{F_{ext}\}_i$ the total applied force vector at time step i (*constant throughout the step*) and $\{F_{int}\}_i^{j-1}$ the internal loads vector for time step i, iteration j.

The Newton-Raphson procedure guarantees convergence if and only if the solution at any iteration $\{\delta\}_i^j$ is 'near' the exact solution. Therefore, even without path-dependent nonlinearity, the incremental approach (i.e., applying the loads in increments) is sometimes required in order to obtain a solution corresponding to the final load level. That is the reason why the Newton-Raphson procedure belongs to the family of *Incremental-Iterative Strategies*.

When the stiffness matrix is updated at every iteration the process is termed a *Full Newton-Raphson procedure*, Figure 3. Alternatively, the stiffness matrix could be updated less frequently (*Modified Newton-Raphson procedure*), for example during the first iteration of each step, Figure 4. Finally if no updating of the stiffness matrix is taken place one has the *Initial Newton-Raphson procedure*, Figure 5. The last two converged more slowly than the *Full Newton-Raphson procedure*, but they require fewer matrix manipulations and inversions.

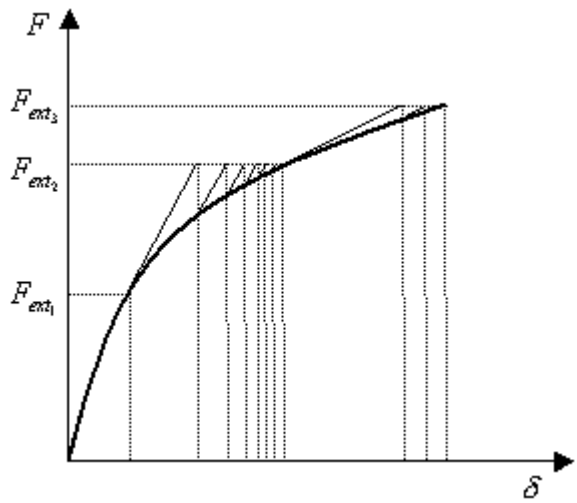

FIGURE 4. *Modified Newton-Raphson procedure (single DOF problem).*

Remarks:

- At the beginning of each step i of the Newton-Raphson procedure, the user can make an initial guess for $\{\delta\}^0$ (it is usually taken equal to the displacements at the end of the converged previous step. For the first step $\{\delta\}^0$=$\{0\}$). For numerical problems having only one possible solution, the Newton-Raphson procedure must provide this solution whatever the initial guesses are (if of course convergence is achieved). In case of problems admitting more that one solution (i.e., the formation of shear bands in a second gradient medium) the different solutions can be found by using random initialization at certain steps (see [12] and [13]).
- In case of proportional loading the external load vector takes the following form:

$$\{F_{ext}\}_i = \lambda tot_i \{F_I\} \tag{2.6}$$

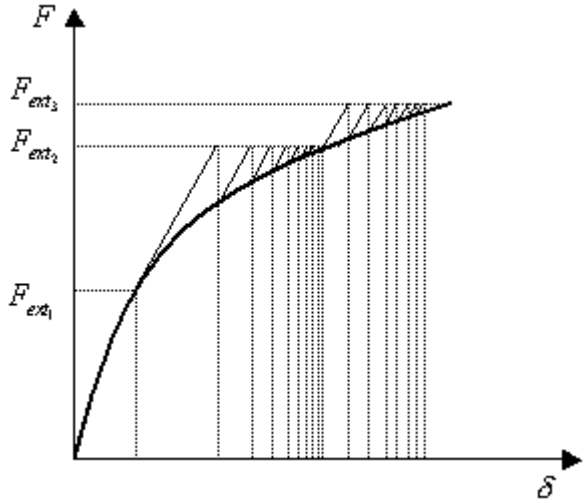

FIGURE 5. *Initial Newton-Raphson procedure (single DOF problem).*

in which $\{F_I\}$ is the reference external load vector, typically as specified in the input data for the problem and λtot_i the total factor *that stays constant within the step i* [20].

2.2. **Why there is a need for more advanced Incremental-Iterative Strategies ?** Geometrically or physically non-linear problems are often characterized by the presence of critical points in the structural response, Figure 6. Critical points can either be classified as bifurcation points or limit points, the major difference being the multiplicity of admissible solutions in a bifurcation point. A load limit point is reached when the applied load reaches a local maximum after which snap-through behavior occurs. A control limit point is reached whenever the solution path presents a local snap-back behavior (simultaneous decrease of the force and the displacement). A control limit point is often called a displacement limit point in the literature, since displacements are generally used in the control algorithm [25]. These instabilities are generally caused by an elastic energy release (elastic unloading) due to stress redistributions higher than the dissipation of the damage process [19],[30]. In addition to these physical instabilities, spurious snap-backs are also reported in the literature [1],[11],[30]. They originate from the spatial discretization and strengthen still more the convergences difficulties.

Conventional Newton-Raphson incremental-iterative strategy *holds the load parameter* $\{F_{ext}\}_i$ *(or the load factor* λtot_i*) constant during the step i* whilst iterating to convergence. An alternative option is to take a single, constant within the step i displacement compound as the controlling parameter and the corresponding level as unknown (*direct displacement control* [3],[6]). Both methods implicitly assume that the evolution of at least one degree of freedom is *monotonous*. Passing critical points is thus extremely difficult owing to the near singular nature of the tangent stiffness matrix in their neighborhood and the possible change of the sign of the load factor. The first approach fails in the presence of load limit points and the second in the presence of displacement limit points.

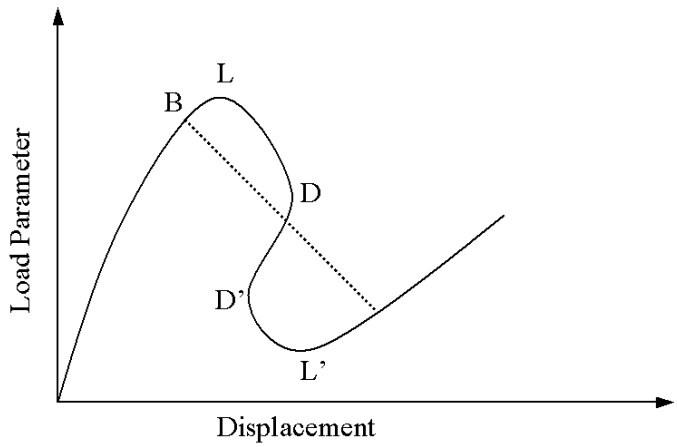

FIGURE 6. *Critical points: B=Bifurcation, D,D'=Displacement limit points (snap-back), L,L'=Load limit points (snap-through).*

Remark:

- As it is indicated in [20],[21] the true response of a structure would involve both statics and dynamics and so the 'unstable' paths of Figure 6 may be omitted. For example, under displacement control the behavior of the structure in Figure 6 would jump from point D to L' (instead of following the 'unstable'path D-D'-L). When this observation is coupled with the difficulties that can accompany complex 'static path-following procedures' it is easily understood why a number of papers consider a static-dynamic option [37],[40]. However, it is sometimes necessary to be able to trace the complete load-displacement curve under static loadings (for example in order to study the creation of shear bands, to reproduce buckling, to gain insight into the mechanics or cause of a structural failure, to obtain plastic mechanism solutions for different materials...).

3. Following path solution techniques

In the advanced incremental-iterative strategies studied hereafter it is assumed that all load magnitudes are controlled by a single scalar parameter (i.e., the total load factor) *that changes automatically at each iteration*:

$$\{F_{ext}\}_i^j = \lambda tot_i^j \, \{F_I\} \tag{3.1}$$

in which $\{F_I\}$ is, as before, the reference external load vector. Writing the total load factor in an incremental form (see Figure 7) one obtains:

$$\lambda tot_i^j = \lambda tot_{i-1}^{conv} + \lambda_i^1 + \sum_{j=2}^{j} \Delta\lambda_i^j \, , \tag{3.2}$$

λtot_{i-1}^{conv} is the value of the total load factor at the end of the (converged) previous step, λ_i^1 is the incremental load factor *measured from the beginning of the load step*

i and calculated during the first iteration and $\Delta\lambda_i^j$ the corrections to apply in order to restore equilibrium. During the first iteration, a special procedure is followed in order to calculate λ_i^1, while $\Delta\lambda_i^1$ is taken equal to zero (see sections 3.1 and 3.3).

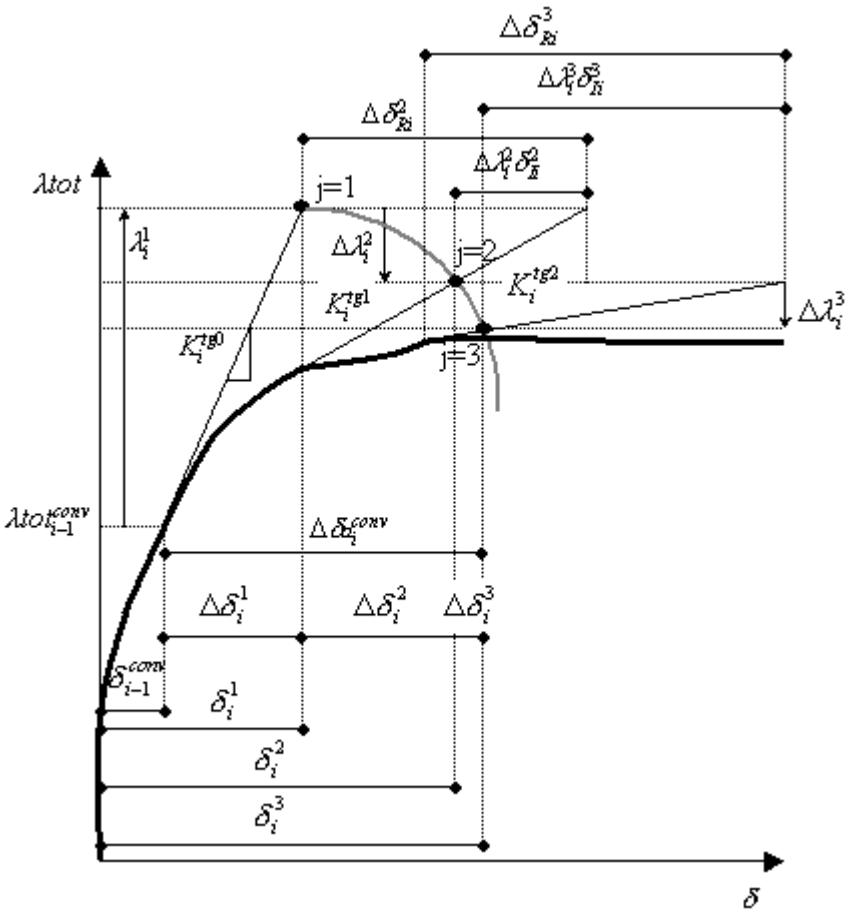

FIGURE 7. *Following path techniques: General incremental formulation (single DOF problem converged after $j = 3$ iterations).*

In other words, the idea behind those advanced numerical techniques is the following:*A first approximation* of the load factor is calculated during the first iteration of each step (λ_i^1, *Predictor Step*). Further iterations are however necessary in order to correct this approximation. A constraint equation is thus solved within each iteration in order to calculate $\Delta\lambda_i^j$ and to *automatically change the external load*, (Figure 7).

Mathematically those increments can be viewed as the trace of a single equilibrium curve in a space spanned by the nodal displacements variables and the total load factor (see gray line, Figure 7). Therefore, all the options of the Newton-Raphson method are still used. As the displacements vectors *and* the load factor are treated as unknowns, an 'automatic load step strategy' has to be introduced and an additional constraint equation is required. *It is the form of this equation*

which distinguishes the various iterative strategies studied hereafter. The choice of the form of the constraint equation is often driven by geometrical considerations (updated normal plane, spherical path... see section 3.4 for more details).

The following presentation is based on the excellent work presented in [16]. Some elements of the presentation have also been inspired from [2]. Iterative cycles within a step i start at $j = 1$, which corresponds to *an increment* of the external load. The equilibrium iterations commence at $j = 2$. Two distinct strategies are required for the successful completion of a single load step in an incremental-iterative scheme:

(1) Selection of a *suitable external load increment* λ_i^1 for the first iterative cycle ($j = 1$) using a *load incrementation strategy*;
(2) Selection of an *appropriate iterative strategy* for application in subsequent iterative cycles ($j \geq 2$) with the aim of restoring equilibrium as rapidly as possible.

A general presentation of appropriate *iterative* and *load incrementation strategies* combined with the advantages of a *full Newton-Raphson procedure* follows:

3.1. **Iterative strategy - The first iterative cycle, $j = 1$ (*Predictor step*).** A new load step i starts with the computation of the tangent stiffness matrix $[K^{tg}]_i^0$ based on the known displacements and stresses at the conclusion of the previous step. The 'tangent' displacements $\{\delta_I\}_i^1$ for this load step are then computed as the solution of

$$\left[K^{tg}\right]_i^0 \{\delta_I\}_i^1 = \{F_I\} \ , \tag{3.3}$$

The magnitude of the displacements is arbitrary only their direction is important. Next the value of the initial load increment λ_i^1 is determined according to a particular *load incrementation strategy* described in section 3.3. The incremental displacements are then evaluated by scaling the tangent displacements

$$\{\Delta\delta\}_i^1 = \lambda_i^1 \{\delta_I\}_i^1 \ , \tag{3.4}$$

and the total displacements and total load level are updated from those at the conclusion of the previous load step as follows ($\Delta\lambda_i^1$ is considered equal to zero):

$$\{\delta\}_i^1 = \{\delta\}_{i-1}^{conv} + \{\Delta\delta\}_i^1 \ , \tag{3.5}$$

$$\lambda tot_i^1 = \lambda tot_{i-1}^{conv} + \lambda_i^1 \ , \tag{3.6}$$

At this stage the solution invariably does not satisfy total equilibrium and so additional iterative cycles are required to restore equilibrium.

3.2. **Iterative strategy - Equilibrium iterative cycles, $j \geq 2$.** The incremental change in the displacements can be written as the solution of

$$\left[K^{tg}\right]_i^{j-1} \{\Delta\delta\}_i^j = \lambda tot_i^j \{F_I\} - \{F_{int}\}_i^{j-1} \ , \tag{3.7}$$

or using equation (3.2)

$$\left[K^{tg}\right]_i^{j-1}\{\Delta\delta\}_i^j = \Delta\lambda_i^j\{F_I\} - \{\psi\}_i^{j-1} \ , \tag{3.8}$$

$$\{\psi\}_i^{j-1} = \{F_{int}\}_i^{j-1} - (\lambda tot_{i-1}^{conv} + \lambda_i^1 + \sum_{j=2}^{j}\Delta\lambda_i^{j-1})\{F_I\} \ , \tag{3.9}$$

In the above expression $\{\psi\}_i^{j-1}$ represents the internal out-of-balance ('residual') force acting on the structure. $[K^{tg}]_i^{j-1}$ is calculated at $\{\delta\}_i^{j-1}$, the accumulated incremental displacement vector of the step i from iteration 1 to $j-1$. The internal force vector $\{F_{int}\}_i^{j-1}$ is computed as usual with the Finite Element Method by integrating the generalized stress resultants through the volume of each element and then summing the elemental contributions [5].

The right-hand side of the equation (3.8) is linear in $\Delta\lambda_i^j$ and so the final solution can be written *as the linear combination of two vectors*:

$$\{\Delta\delta\}_i^j = \Delta\lambda_i^j\{\delta_I\}_i^j + \{\Delta\delta_R\}_i^j \ , \tag{3.10}$$

$\{\delta_I\}_i^j$ are the displacements computed as follows:

$$\left[K^{tg}\right]_i^{j-1}\{\delta_I\}_i^j = \{F_I\} \ , \tag{3.11}$$

$\{\Delta\delta_R\}_i^j$ are the displacements obtained as the solution of

$$\left[K^{tg}\right]_i^{j-1}\{\Delta\delta_R\}_i^j = -\{\psi\}_i^{j-1} \ , \tag{3.12}$$

using a conventional *Full Newton-Raphson procedure*. The total load level is calculated according to eq.(3.2) and the total displacements as follows:

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

The variation of the load parameter $\Delta\lambda_i^j$ is obtained by solving an appropriate constraint equation as described in section 3.4. Iterative cycles are continued until a convergence criterion based on the forces and/or the displacements is satisfied. If convergence in not achieved within a number of iterations specified by the user, or if divergence of the solution is detected, the solution for this load step should recommence with the application of a reduced initial load increment.

3.3. **Load incrementation strategy.** When commencing a load step i an initial load increment λ_i^1 must be chosen. The choice of the *increment size* is important and should reflect the current degree of non-linearity. If the initial load increment is too large then convergence will be slow or may not occur at all or we take the risk to miss certain 'unstable paths' of the nonlinear behavior of the structure (the solution would for example 'jump' from point L to point L', Figure 6). If the initial load increment is too small then more converged states are computed than are strictly necessary [20].

The automatically chosen load increment must also be of the *correct sign*, necessitating measures capable of detecting when maximum and minimum points on

the load-deflection curve have been passed. If the uncorrect sign is chosen solution 'doubles back' or oscillates and subsequently fails to converge [20].

`Choosing the increment size:` Two classes of methods are often used to proper adapt the load factor from one increment to another. The first class is based on *the direct adaption of the load increment* [17],[20],[35] and the second one on the evolution of *the current stiffness parameter* defined in [8],[9],[10].

The direct adaption of load increment uses the ratio of the actual number of iterations required for convergence in the previous load step J_{d-1}, a user-defined desired number of iterations for convergence J_d and an exponent γ

$$X_i^1 = \pm X_{i-1}^1 (\frac{J_d}{J_{d-1}})^\gamma \ , \tag{3.14}$$

X can be either λ_i^1 or the arc-length l (see eq.(4.2)). The value of the exponent γ lies typically between 0.25 and 1 [7],[17],[18],[20],[35]. The desired number of iterations is usually set equal to 3 [20], 5 or 6 [25].

Following the second method, the initial load increment is calculated according to an unscaled version of the current stiffness parameter as explained in [16]:

$$S_{X,i} = \frac{\left[\{\delta_I\}_1^1\right]^T \{F_I\}}{\left[\{\delta_I\}_i^1\right]^T \{F_I\}} \ , \tag{3.15}$$

In the previous expression T states for the transpose of a vector. $S_{X,i}$ has the initial value of unity for any non-linear system. Values of $S_{X,i}$ less than and greater than unity indicate 'softening' and 'stiffening' systems respectively. The expression for automatic load incrementation takes then the following form:

$$X_i^1 = \pm X_{i-1}^1 \left| \frac{\Delta S_X}{\Delta S_{X,i}.} \right| \ , \tag{3.16}$$

in which ΔS_X is a prescribed scalar constant which may either be entered as input to the program, or calculated using the values of the current stiffness parameters calculated for the first two load steps, i.e. $\Delta S_X = S_{X,2} - S_{X,1}$ and $\Delta S_{X,i}$ is the change in the current stiffness parameter from the previous load step to the current load step i.e $\Delta S_{X,i} = S_{X,i} - S_{X,i-1}$.

In regions of the load-displacement response which are nearly linear $\Delta S_{X,i}$ may be small, in which case equation (3.16) will lead to large initial load increments X_i^1. For this reason it is desirable to specify a maximum absolute value for the initial load increment calculated by eq.(3.16), [16].

Chan [14] uses the following simpler application of the current stiffness parameter for automatic load incrementation

$$X_i^1 = \pm X_1^1 \mid S_{X,i}|^\eta \ , \tag{3.17}$$

in which the exponent η usually equals 1. In regions of displacement limit points S_X may become larger (greater than one) and so it is desirable to specify a maximum value of $|S_X|$ to limit the load step size.

`Choosing the sign of the increment:` The proper sign of the new load factor increment λ_i^1 should follow that of the previous increment unless the determinant of the stiffness matrix *at the beginning of the new increment* changes sign. One has to store the sign of the determinant of the stiffness matrix at the end of each step and to look the existence of a *negative pivot* in the triangularization of the global stiffness matrix at the beginning of the new increment. This is an indicator that the determinant of the tangent stiffness matrix changes sign [17],[20],[35] and that a critical point has been overcome. In relation to the adopted solution procedure, this will occur when one of the terms in $[D]$, the diagonal matrix of the $[L]\,[D]\,[L]^T$ factorisation of $[K^{tg}]_i^1$ is negative (which implies one negative eigenvalue for the stiffness matrix).

Remark:

- The determinant criterion does not work properly when multiple negative eigenvalues exist (i.e., in the presence of bifurcation points). In that case, the code will oscillate about this bifurcation point. It is recommended then, until more advanced path-following techniques are available to stop the code and with the aid of 'restarts' and manual intervention to try to understand what is happening [20].

Another criterion was proposed by Bergan [8] who suggested that a change in the sign should occur upon reversal of the sign of the incremental work ΔW:

$$\Delta W = \lambda_i^1 \{\delta_I\}_i^1 \{F_I\} \tag{3.18}$$

According to [25], if all increments would be infinitely small and if no bifurcation points are present on the loading path, the incremental work criterion coincides with the appearance of a negative determinant of $[K^{tg}]_i^1$. In [16] however, it is mentioned that the external work sign is deficient in the vicinity of displacement limit points, while the determinant criterion is not. On the basis of the two previous criteria, the following proposal to determine the proper sign of the load estimator is proposed in [7].

$$\lambda_i^1 = +\left|\lambda_i^1\right| \quad if \quad \{\Delta\delta_a\}_{i-1}^{conv} \{\delta_I\}_i^1 \geq 0 \tag{3.19}$$

$$\lambda_i^1 = -\left|\lambda_i^1\right| \quad if \quad \{\Delta\delta_a\}_{i-1}^{conv} \{\delta_I\}_i^1 < 0 \tag{3.20}$$

$\{\Delta\delta_a\}_{i-1}^{conv}$ is the accumulated converged incremental displacement vector of the previous step (Figure 7). The effectiveness of the criterion was confirmed in [38]. However, criteria based on the sign of the incremental work seem to be insensitive (do not respond) to bifurcation according to [20]...

3.4. **Constraint equations for calculating $\Delta\lambda_i^j$.** The iterative change $\Delta\lambda_i^j$ is regarded as *an additional unknown variable*. In order to calculate it, various constraint equations are proposed by different workers. Some of them are summarized below:

- If $\Delta\lambda_i^j$ is held equal to 0 at every iteration one obtains the classical *Full Newton-Raphson method* (with load control). As mentioned in section

2.2, this iterative strategy does not permit to pass load limit points (displacement limit points) if a force (displacement) compound is used as the controlling parameter.

- Chan [14] proposed the following equation defined as the '*minimum residual displacement*' method:

$$\Delta\lambda_i^j = \frac{-\left[\{\delta_I\}_i^j\right]^T \{\Delta\delta_R\}_i^j}{\left[\{\delta_I\}_i^j\right]^T \{\delta_I\}_i^j}, \tag{3.21}$$

This method guarantees a minimum value for the unbalanced displacement norm in each iteration.

- The '*updated normal plane method*', where the iterative path follows a 'plane' normal to the tangent vector $\{t\}$ of Figure 8, [35]:

$$\Delta\lambda_i^j = \frac{-\left[\{\Delta\delta_a\}_i^{j-1}\right]^T \{\Delta\delta_R\}_i^j}{\left[\{\Delta\delta_a\}_i^{j-1}\right]^T \{\delta_I\}_i^j}, \tag{3.22}$$

with $\{\Delta\delta_a\}_i^{j-1}$ the accumulated incremental displacement vector of the step i from iteration 1 to $j-1$.

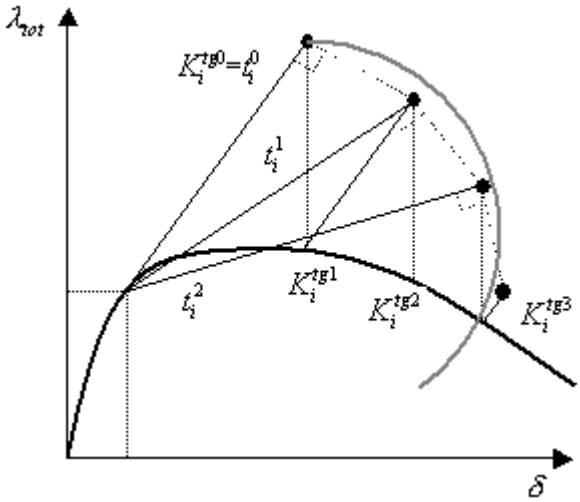

FIGURE 8. *The updated normal plane method (single DOF problem).*

- The '*consistently linearized method*', where although the update direction tends initially away from the spherical path, the application of a residual draws the path back to the arc *once convergence is achieved* ([39], Figure 9). To do that, the difference between the length of the tangent vector $\{t\}_i^{j-1}$ and the desired length s_i (s stays constant within the step i and equals the radius of the arc) is projected into the current tangent vector to provide

the residual for the orthogonality expression (for more details see [24]) :

$$\Delta\lambda_i^j = \frac{-\left\|\{t\}_i^{j-1}\right\| \left(\left\|\{t\}_i^{j-1}\right\| - s_i\right) - \left[\{\Delta\delta_a\}_i^{j-1}\right]^T \{\Delta\delta_R\}_i^j}{\left[\{\Delta\delta_a\}_i^{j-1}\right]^T \{\delta_I\}_i^j}, \tag{3.23}$$

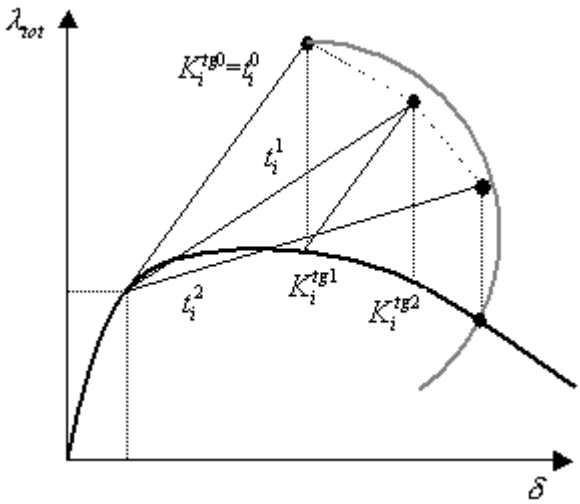

FIGURE 9. *The consistently linearized method (single DOF problem).*

- The *'explicit iteration on spheres'* requires the formulation of a residual based on the error that would be obtained using an orthogonal iteration path. This error can be corrected *in advance* to provide the desired arc. Contrary to the *'consistently linearized method'* where the path is finally drown back to the arc only at the end of the step (once convergence is achieved), now the path is drown back to the arc *at the end of each iteration.* The following algorithm is proposed in [24], (Figure 10):

  (1) Calculate the update for orthogonal iteration

$$\Delta\lambda_i^j = \frac{-\left[\{\Delta\delta_a\}_i^{j-1}\right]^T \{\Delta\delta_R\}_i^j}{\left[\{\Delta\delta_a\}_i^{j-1}\right]^T \{\delta_I\}_i^j}, \tag{3.24}$$

  (2) Calculate the associate displacement vector

$$\{\Delta\delta\}_i^j = \Delta\lambda_i^j \{\delta_I\}_i^j + \{\Delta\delta_R\}_i^j, \tag{3.25}$$

  (3) Find the length of the tangent in the potential configuration

$$t_i^j = \sqrt{\left\|\{t\}_i^{j-1}\right\|^2 + \left[\{\Delta\delta\}_i^j\right]^T \{\Delta\delta\}_i^j}, \tag{3.26}$$

  (4) Calculate the required residual R_i^j for explicit iteration on a sphere

$$R_i^j = -\frac{s_i^2}{\left\|\{t\}_i^{j-1}\right\|}\left(\left\|\{t\}_i^{j-1}\right\| - s_i\right), \tag{3.27}$$

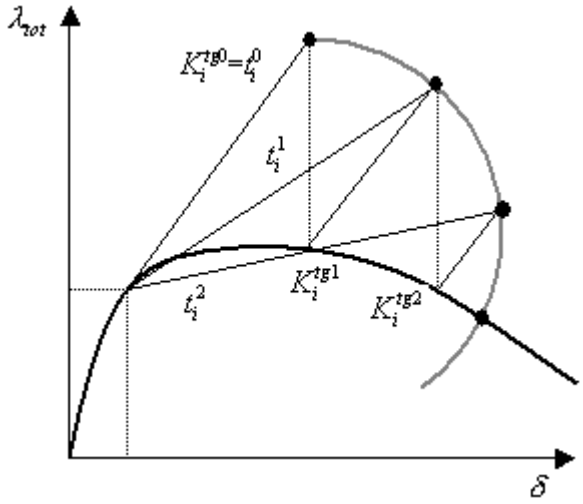

FIGURE 10. *Explicit iteration on spheres (single DOF problem).*

(5) Return to the general formula for iteration path direction

$$\Delta\lambda_i^j = \frac{R_i^j - \left[\{\Delta\delta_a\}_i^{j-1}\right]^T \{\Delta\delta_R\}_i^j}{\left[\{\Delta\delta_a\}_i^{j-1}\right]^T \{\delta_I\}_i^j} \,, \tag{3.28}$$

(6) Calculate the desired displacement vector

$$\{\Delta\delta\}_i^j = \Delta\lambda_i^j \{\delta_I\}_i^j + \{\Delta\delta_R\}_i^j \,, \tag{3.29}$$

The authors claim that the larger domain of attraction offered by this algorithm may be important in the vicinity of limit points with sharp gradients.

- The concept of using an *arc-length* constraint equation was introduced in [36] and [41]. The method has been modified by Crisfield [17] in the light of numerical experience and improved suitability for the finite element method. One finally has the following quadratic equation in $\Delta\lambda_i^j$ [16]:

$$A(\Delta\lambda_i^j)^2 + B\Delta\lambda_i^j + C = 0 \,, \tag{3.30}$$

in which

$$A = \left[\{\delta_I\}_i^j\right]^T \{\delta_I\}_i^j \,, \tag{3.31}$$

$$B = 2\left[\{\Delta\delta_a\}_i^{j-1} + \{\Delta\delta_R\}_i^j\right]^T \{\delta_I\}_i^j \,, \tag{3.32}$$

$$C = \left[\{\Delta\delta_a\}_i^{j-1} + \{\Delta\delta_R\}_i^j\right]^T \left[\{\Delta a\delta\}_i^{j-1} + \{\Delta\delta_R\}_i^j\right] - l_i^2 \,, \tag{3.33}$$

with l_i the radius of the arc considered constant throughout the step. Suppose that two roots of equation (3.30) are denoted $(\Delta\lambda_i^j)_1$ and $(\Delta\lambda_i^j)_2$. The correct choice of root is one which avoids 'doubling back' on the load-displacement response. The angle between the incremental displacement

vector *before* the present iteration and the incremental displacement vector *after* the current iteration should be positive. For the two possible roots, the corresponding angles are defined in [16],[17] as:

$$\theta_{1,2} = \left[\{\Delta\delta_a\}_i^{j-1} + (\Delta\lambda_i^j)_{1,2} \{\delta_I\}_i^j + \{\Delta\delta_R\}_i^j \right]^T \{\Delta\delta_a\}_i^{j-1} \ , \tag{3.34}$$

The correct choice of root $\Delta\lambda_i^j$ is the one which gives the *minimum positive* angle θ, [21].

The quadratic equation (3.30) possess imaginary roots if $B^2 - 4AC$ is less than zero. This situation can arise if the initial load increment is too large and the structure exhibits multiple instability directions at a point. For this case it is preferred that the finite element code writes a warning message and recommences the solution for this load step with the application of a reduced initial load increment.

Remarks:

- For graphical purposes, the stiffness matrix in Figure 8 to Figure 10 stays constant within the iterations. However, the equations (3.21) to (3.34) are presented in their general form.
- At the beginning of the section 3.1 it is mentioned that the calculation of the tangent stiffness matrix $[K^{tg}]_i^0$ is based on the known displacements and stresses at the conclusion of the previous step. However, in order to take a better guess for starting the incremental-iterative scheme within a step some finite element codes usually extrapolate the DOF solution using the previous history (i.e., in LAGAMINE the velocities of the step i are considered equal to the converged velocities of the step $i-1$, see also [2]).
- A general formulation based on orthogonality principles able to derive most of the previous constraint equations, to illustrate their relationships and to provide a geometrical explication is proposed in [24].
- A careful analysis of the incremental/iterative behavior using equation 3.30 showed severe numerical difficulties in the presence of very sharp snap-backs (particularly those occurring in fracture mechanics or damage mechanics both for failure initiation and crack propagation [21]). The problems were associated with an 'incorrect choice of root' from the two solutions to the quadratic equation in the load level change. The idea behind the choice of minimum angle is to avoid the solution 'doubling back'. However, for a very sharp snap-back , one wants the solution to double back. Hellweg and Criesfild proposed to select the root with the minimum residual [21],[28]. This new criterion can easily be added to an existing arc-length method by including one additional loop over the number of the roots. Naturally, this approach is more expensive than the conventional one. In addition, for most relatively smooth problems, this technique is not needed. Criesfild [21] propose to turn on this option when the code exhibits convergence difficulties or the analyst anticipates the existence of snap-backs. A more recent paper however [30], showed that even this new criterion may prove insufficient.
- The *'explicit iteration on spheres'* provides exactly the same results as the concept of the *arc-length* constraint equation [17] without the solution of a quadratic equation and the selection of one root [24]. However, this

approach requires a solution for the intersection of the tangent with the updated normal plane. In the case of sharp turns on the loading path this intersection may still yield poor results [25].

- An additional scalar parameter β is often found in the literature in the expressions of the different constraint equations. It is a scaling factor which deals with the dimensional incompatibility between the degrees of freedom and the load factor λtot_i^j (this inconsistency appears only if different types of degree of freedoms are present). Setting β equal to zero (which is the case for the equations presented in this work) results in constraint equations that are explicitly written in the hyperspace of the DOF only. Although it is known that $\beta = 0$ is often the recommended choice, general conclusions cannot be drown and the best choice of β is in fact structure dependent [25]. With β equal to zero the quadratic equation proposed by Crisfield [17] corresponds to *the cylindrical arc-length method*, while a non-zero β corresponds to *the spherical arc-length method*.
- Geometrically non-linear problem are mainly solved by techniques which involve *the control of all DOF*. Physically non-linear problems often require the control of a *limited number* of degrees of freedom, due to localized deformations [20],[22],[25],[30],[31],[42], [43]. Localization problems are characterized by the fact that local DOF dominate the overall mechanical behavior of the considered discretized medium. Using all DOF's may results in an unstable Newton-Raphson scheme [25].

4. Algorithm used for the implementation of path following techniques into LAGAMINE

The principal subroutines that have been changed in order to implement the previous following path techniques into the finite element code LAGAMINE are: SOLSYS.F, ARCLEN.F, LAMIN2.F, LICHAB.F, INIDDL.F, NORME1.F, AUTOMA.F, AUTOMU.F, AUTORD.F, ASSEL.F, CLCOQ4.F... *The direct adaption of load increment* is used to choose the increment size (jstep=-3. However, considering jstep=-1 one uses the FACMU variable as it is usually the case in LAGAMINE). The sign of the load increment is determined according to the sign of the determinant of the tangent stiffness matrix. The algorithm takes the following form:

Step $i = 1$

- The 'classical strategy' already implemented into LAGAMINE is used. The user defines an initial loading *(force or displacement)* and the code proceeds to the resolution with the *Full Newton-Raphson procedure*. In that way one can take advantage of all the functionalities already presented in the code (define a maximum number of iterations needed for convergence, a maximum and a minimum value step, a multiplicator for the load step - if one does not want to use *the direct adaption of the load increment* -, a termination load factor etc. The relative norms of *both residual forces and displacements* are adopted as criteria for convergence.
- During this first step the code calculates the reference external load factor $\{F_I\}$ that stays constant throughout the whole loading (*proportional loading*, eq.(2.6)).

- Reading the desired number of iterations for convergence J_d and the exponent γ. Those parameters are used for the *direct adaption of load increment method*, eq.(3.14).

Step $i = 2$ - Iteration $j = 1$

- The starting load factor λ_2^1 is taken equal to load factor (force or displacement) of the converged first step. This starting load factor has to be between 20% and 40% of the anticipated maximum load [16].
- Calculate the tangent stiffness matrix $[K^{tg}]_2^0$.
- Calculate the 'tangent' displacements $\{\delta_I\}_2^1$ from eq.(3.3).
- Calculate the incremental displacements $\{\Delta\delta\}_2^1$ by scaling the 'tangent' displacements, eq.(3.4).
- Update the total displacements $\{\delta\}_2^1 = \{\Delta\delta\}_2^1$
- Update the total load factor $\lambda tot_2^1 = \lambda_2^1$
- Calculate the initial radius (positive value) using the following equation [16]:

$$(l_1)^2 = (\Delta\lambda_2^1)^2 \left[\{\delta_I\}_2^1\right]^T \{\delta_I\}_2^1 \ , \tag{4.1}$$

Step $i > 2$ - Iteration $j = 1$

- Read the ratio of the actual number of iterations required for convergence in the previous load step J_{d-1}. Update the radius l as follows:

$$l_i = l_{i-1}(\frac{J_d}{J_{d-1}})^\gamma \ , \tag{4.2}$$

  The updated radius stays constant throughout the whole step i.
- Calculate the tangent stiffness matrix $[K^{tg}]_i^0$ (at $\{\delta\}_{i-1}^{conv}$, the accumulated converged displacement vector of the previous step, unless the code extrapolates the DOF solution using the previous history - see section 3.4). Proceed to the factorisation of the stiffness matrix $[K^{tg}]_i^0 = [L]\,[D]\,[L]^T$, check the existence of negative pivots and the sign of the determinant of the stiffness matrix at convergence of the previous step. Define in that way if $m = +1$ or $m = -1$.
- Calculate λ_i^1 as follows:

$$\lambda_i^1 = m\frac{l_i}{\sqrt{\left[\{\delta_I\}_i^1\right]^T \{\delta_I\}_i^1}} \ , \tag{4.3}$$

- Calculate the 'tangent' displacements $\{\delta_I\}_i^1$ from eq.(3.3).
- Calculate the incremental displacements $\{\Delta\delta\}_i^1$ by scaling the 'tangent' displacements, eq.(3.4).
- Update the total displacements $\{\delta\}_i^1 = \{\Delta\delta\}_i^1$
- Update the total load factor $\lambda tot_i^1 = \lambda_i^1$

Step $i \neq 1$ - Iteration $j \neq 1$

- Calculate the tangent stiffness matrix $[K^{tg}]_i^{j-1}$ (at $\{\delta\}_i^{j-1}$, the accumulated displacement vector at step i iteration $j-1$).

- Calculate the internal ('residual') force vector $\{F_{int}\}_i^{j-1}$ acting on the structure as usual with the Finite Element Method by integrating the generalized stress resultants through the volume of each element and then summing the elemental contributions.
- Calculate the internal out-of-balance ('residual') force acting on the structure according to eq.(3.9).
- Calculate $\{\Delta\delta_R\}_i^j$ using eq.(3.12) and the classical *Full Newton-Raphson method*.
- Calculate the 'tangent' displacements $\{\delta_I\}_i^j$ according to eq.(3.11).
- Calculate $\Delta\lambda_i^j$ using one of the constraint equations presented in section 3.4 (equations (3.21) to (3.34)).
- Calculate the incremental displacements $\{\Delta\delta\}_i^j$ from eq.(3.10).
- Update the total displacements $\{\delta\}_i^j$ with eq.(3.13).
- Update the total load factors according to eq.(3.2).
- Test of convergence - Stock the sign of the determinant of the stiffness matrix. If the number of iterations needed for convergence exceeds the maximum number defined by the user or if divergence of the solution is detected, the solution for this load step recommence with the application of a reduced initial load increment.

Remark:

- One of the disadvantages of the following path techniques is that the termination load factor is not respected (the incrementation of the load factor is *automatic and not user controlled*). In order to be sure that the code stops at the prescribed termination load factor the following procedure is used in LAGAMINE: The code detects when the total load factor is near the desired termination load factor and changes automatically to the 'classical strategy'. It imposes then an incremental load factor such as the final step is done with the desired termination factor. Unfortunately, this option did not perform correctly in some of the examples presented hereafter.

5. Examples using different following path techniques

The first three examples concern geometrical non-linear problems and the last one a 1D concrete bar.

5.1. **Simple truss structure.** The first example is a simple truss structure with 8 bars and 9 nodes as shown in Figure 11, ($EA = 3.10^6$). This structure has been examined in [23],[26],[34]. A similar structure was studied in [4].

The structure has been modeled using 8 TRUSS elements. The load is applied to node 1 and the load-displacement curve in this node typically presents a snap-back behavior as shown in Figure 12. The load parameters are: an initial load F of 40kN, a final load F of 300kN, a maximum step of 100kN, a minimum step of 1kN and a maximum number of iterations of 20. The load-horizontal displacement curve for the loaded node and for different constrain equations are shown in Figures 12 to 16. The parameters for *the direct adaption of the load increment* are $J_d = 5$ and $\gamma = 1/2$. The tolerance for the convergence norm, for which both the relative norm of forces and displacements have been adopted, is taken equal to 10^{-3}.

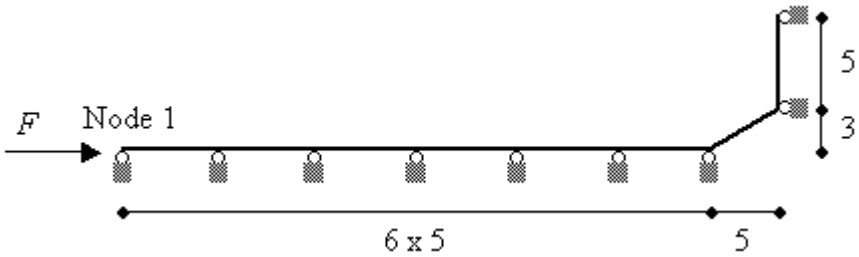

FIGURE 11. *Simple truss structure.*

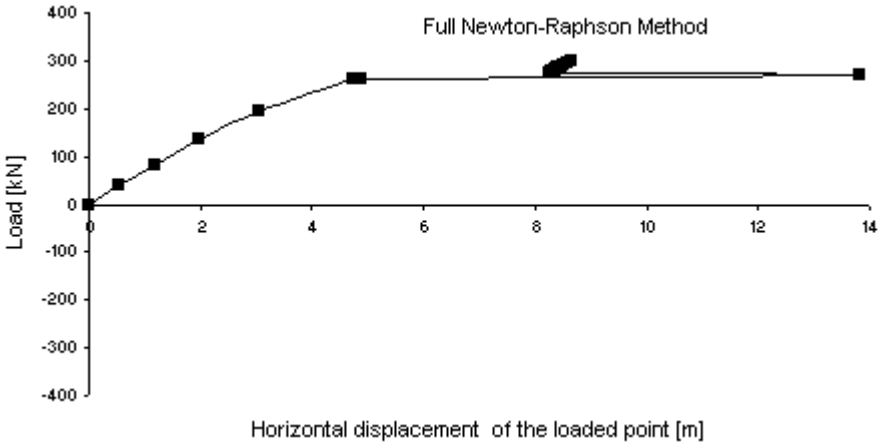

FIGURE 12. *Simple truss structure: Load-displacement curve of the loaded point (Full Newton-Raphson).*

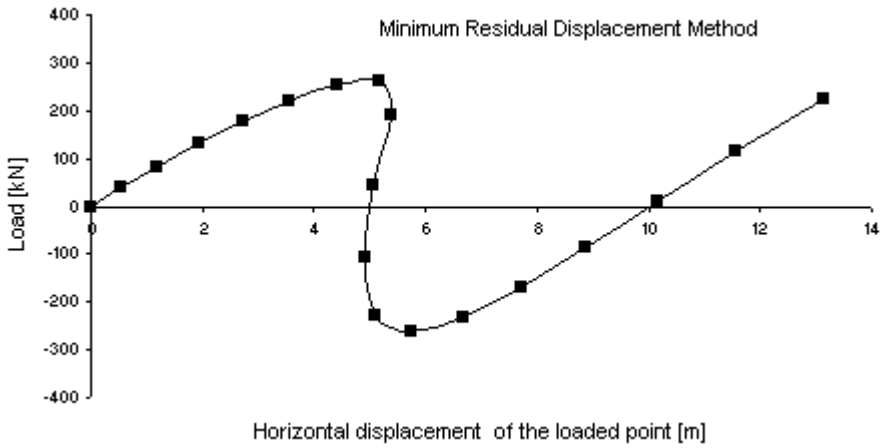

FIGURE 13. *Simple truss structure: Load-displacement curve of the loaded point (Minimum residual displacement).*

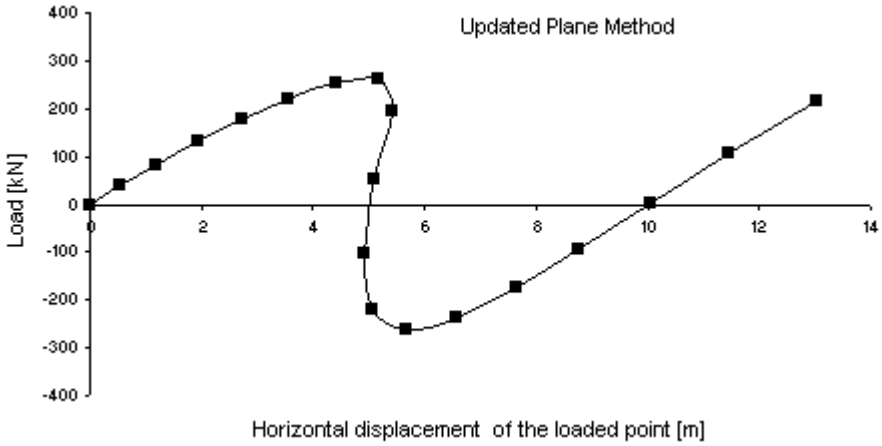

FIGURE 14. *Simple truss structure: Load-displacement curve of the loaded point (Updated normal plane).*

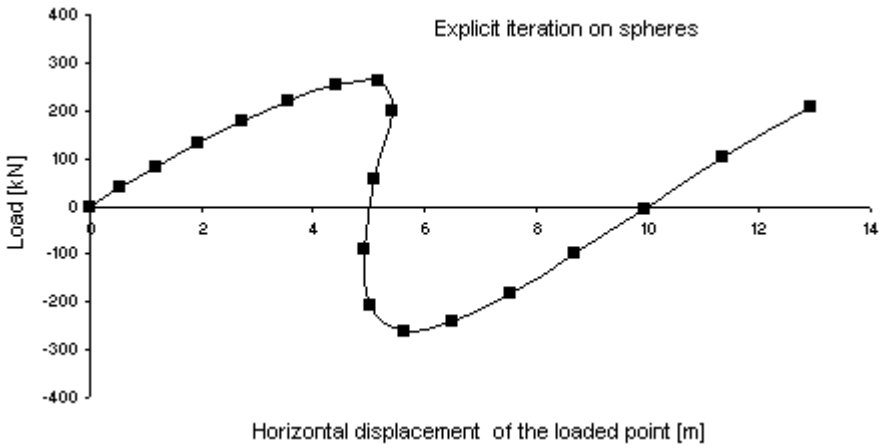

FIGURE 15. *Simple truss structure: Load-displacement curve of the loaded point (Explicit iteration on spheres).*

Table 1 resumes the total number of increments and the cumulative sum of all iterations used by each following path method in order to reach the imposed terminal load (the increments and iterations of the 'classic strategy' are not counted, all the others are counted including the lost iterations prior to a load step reduction).

5.2. **The Lee frame.** The Lee frame is an example with large displacements and instabilities, accompanied with a snap-back response. The analytical solution for this problem is given in [29] and the problem has been used several times as a benchmark problem for the analysis of path following techniques [15],[26],[38],[39]. The configuration of the frame is given in Figure 17 ($A = 6cm^2, I = 2cm^4, E = 720kN/cm^2$).

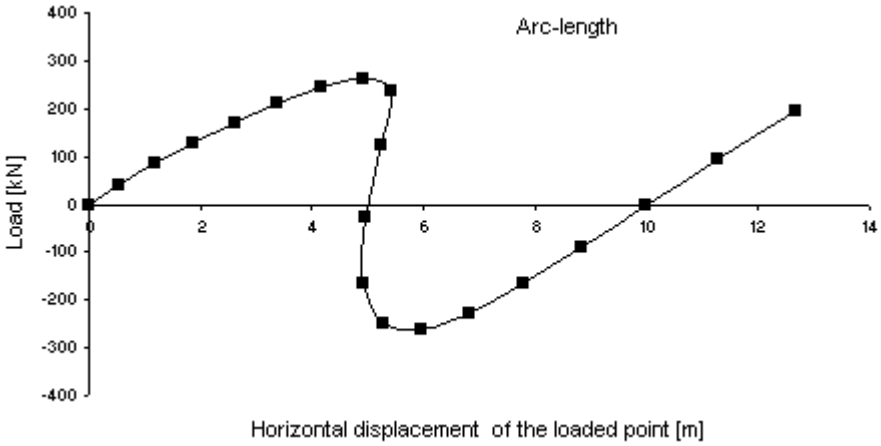

FIGURE 16. *Simple truss structure: Load-displacement curve of the loaded point (Arc-length method).*

Type of constraint equation	No. incr.	No.iter.
Newton-Raphson	failed	failed
Minimum residual displacement	17	58
Updated normal plane	17	58
Explicit iteration on spheres	17	58
Arc-length	19	68

TABLE 1. *Simple truss structure* - Performance and incrementation.

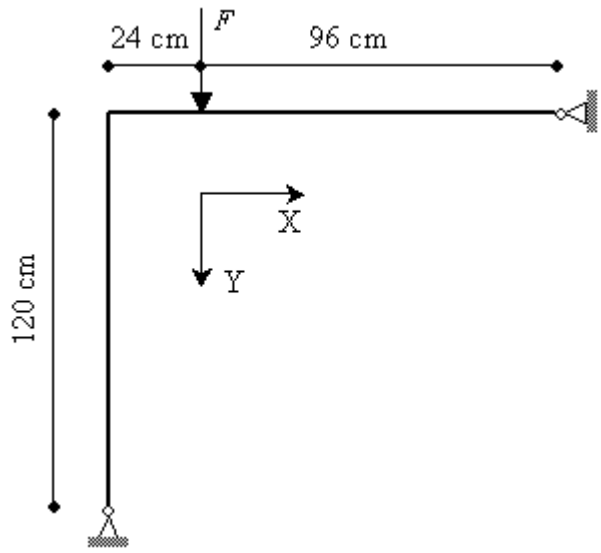

FIGURE 17. *The Lee frame.*

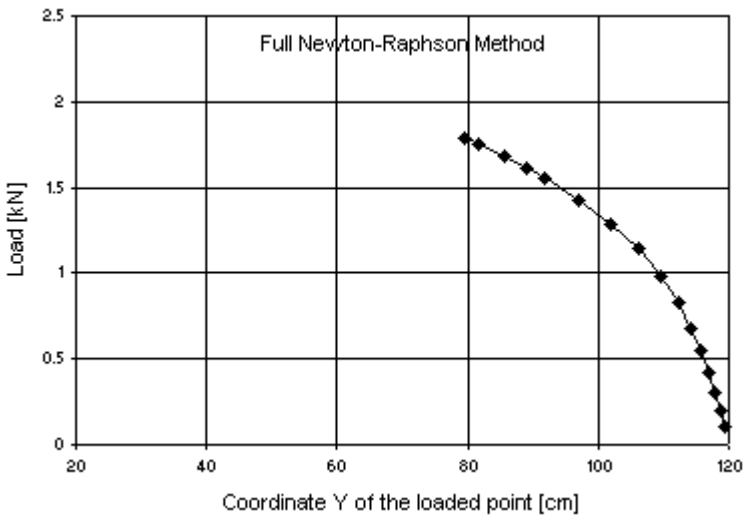

FIGURE 18. *The Lee frame: Load-coordinate curve of the loaded point (Full Newton-Raphson).*

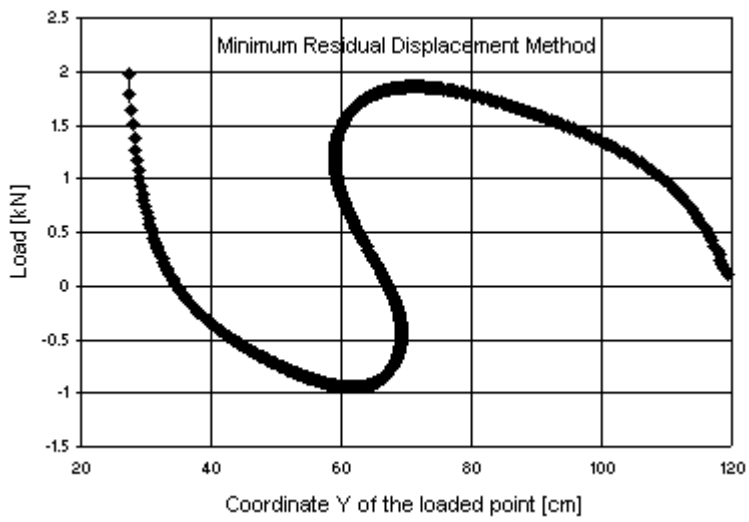

FIGURE 19. *The Lee frame: Load-coordinate curve of the loaded point (Minimum residual displacement).*

The frame has been modeled using 22 thin - no shear deformations - shell elements (KIRSH element). Each element has two nodes and two Gauss points. The load parameters are: an initial load F of 0.1kN, a final load F of 2kN, a maximum step of 10, a minimum step of 0.01 and a maximum number of iterations of 20. The load-coordinate curves for the loaded node and for different constrain equations are shown in Figures 18 to 22. The parameters for *the direct adaption of the load increment* are $J_d = 5$ and $\gamma = 1/2$. The tolerance for the convergence norm, for

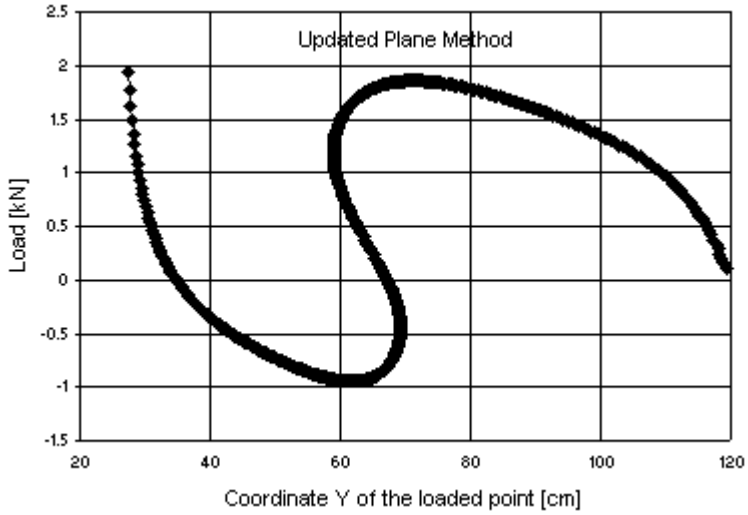

FIGURE 20. *The Lee frame: Load-coordinate curve of the loaded point (Updated normal plane).*

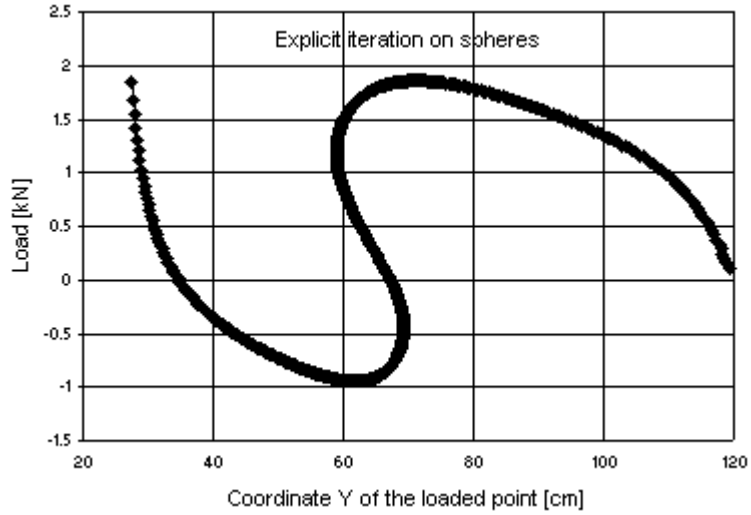

FIGURE 21. *The Lee frame: Load-coordinate curve of the loaded point (Explicit iteration on spheres).*

which both the relative norm of forces and displacements have been adopted, is taken equal to 10^{-3}.

Table 2 resumes the total number of increments and the cumulative sum of all iterations used by each following path method in order to reach the imposed terminal load (the increments and iterations of the 'classic strategy' are not counted, all the others are counted including the lost iterations prior to a load step reduction).

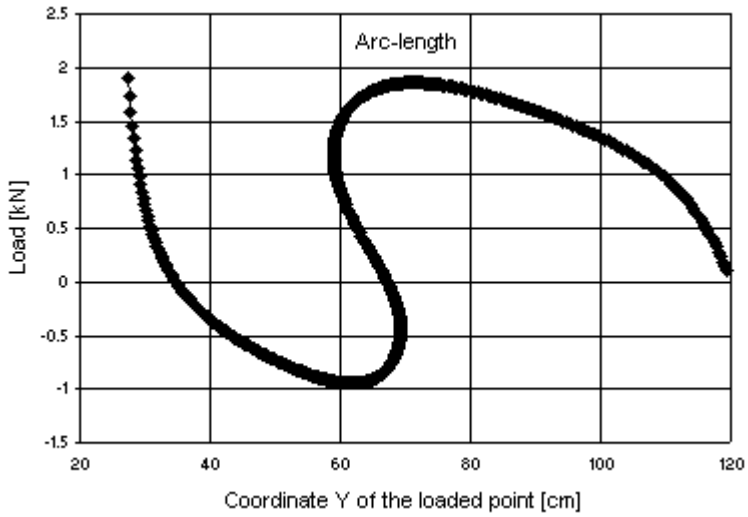

FIGURE 22. *The Lee frame: Load-coordinate curve of the loaded point (Arc-length method).*

Type of constraint equation	No. incr.	No.iter.
Newton-Raphson	failed	failed
Minimum residual displacement	506	2530
Updated normal plane	496	2489
Explicit iteration on spheres	505	2525
Arc-length	512	2586

TABLE 2. *Lee frame* - Performance and incrementation.

5.3. **Circular arch.** The circular arch presented in Figure 23 has been also studied in [16] and [27]. It exhibits substantial geometric non-linear behavior, four limit points and two displacement limit points. The arch is subject to a near central point and the principal parameters are: Young's modulus $E = 200$, cross-sectional area $A = 1.10^4$ and second moment of area $I = 1.10^8$.

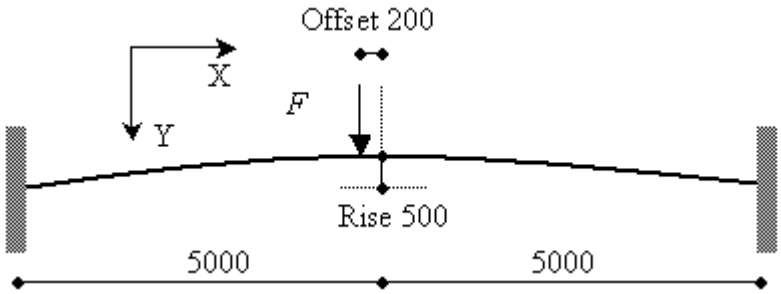

FIGURE 23. *Swallow circular arch subject to near central point load.*

The circular arch has been modeled using 25 thin - no shear deformations - shell elements (KIRSH element). Each element has two nodes and two Gauss points. The load parameters are: an initial load F of 400, a final load F of 1300, a maximum step of 200, a minimum step of 0.01 and a maximum number of iterations of 15. The load parameter-vertical displacement curves for the loaded node and for different constrain equations are shown in Figures 24 to 28. The parameters for *the direct adaption of the load increment* are $J_d = 5$ and $\gamma = 1/2$. The tolerance for the convergence norm, for which both the relative norm of forces and displacements have been adopted, is taken equal to 10^{-3}.

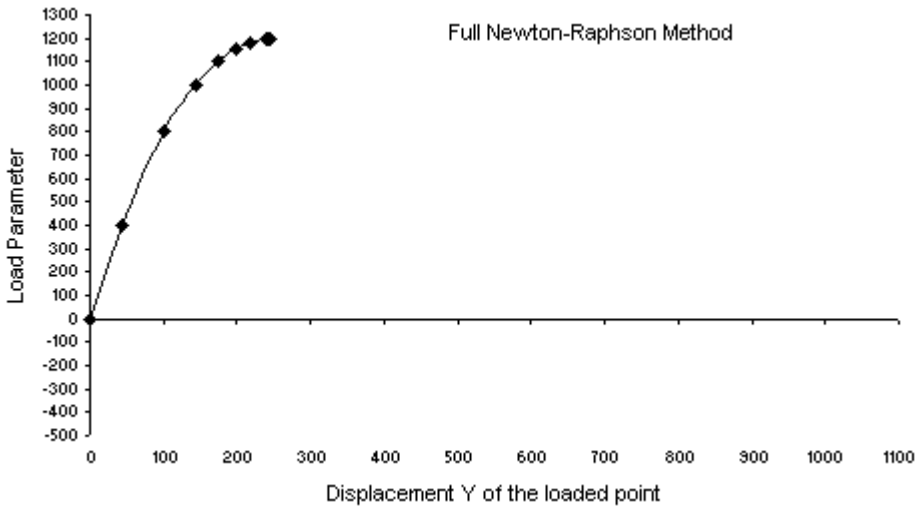

FIGURE 24. *Circular arch: Load parameter-vertical displacement curve of the loaded point (Full Newton-Raphson).*

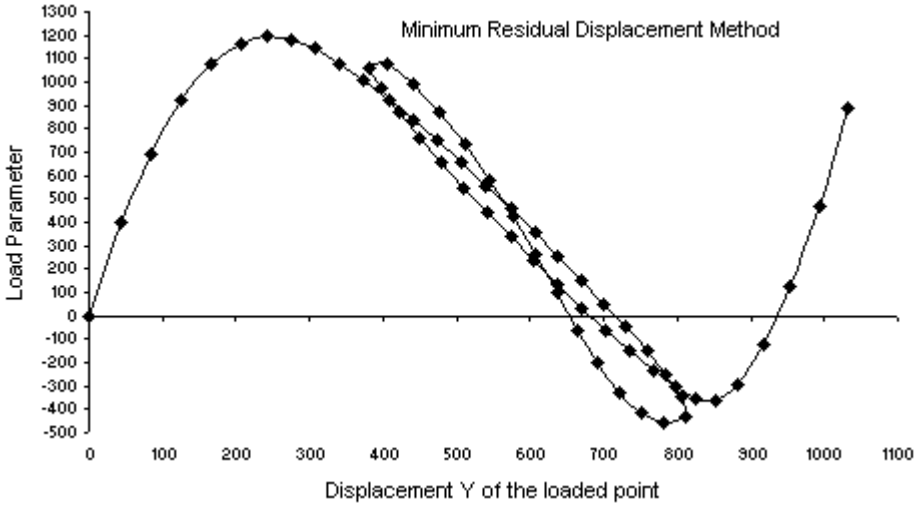

FIGURE 25. *Circular arch: Load parameter-vertical displacement curve of the loaded point (Minimum residual displacement).*

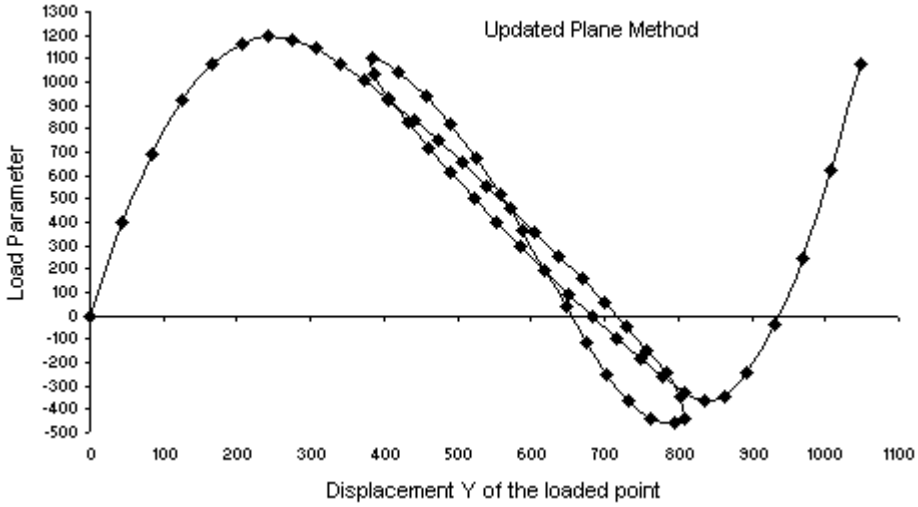

FIGURE 26. *Circular arch: Load parameter-vertical displacement curve of the loaded point (Updated normal plane).*

Table 3 resumes the total number of increments and the cumulative sum of all iterations used by each following path method in order to reach the imposed terminal load (the increments and iterations of the 'classic strategy' are not counted, all the others are counted including the lost iterations prior to a load step reduction).

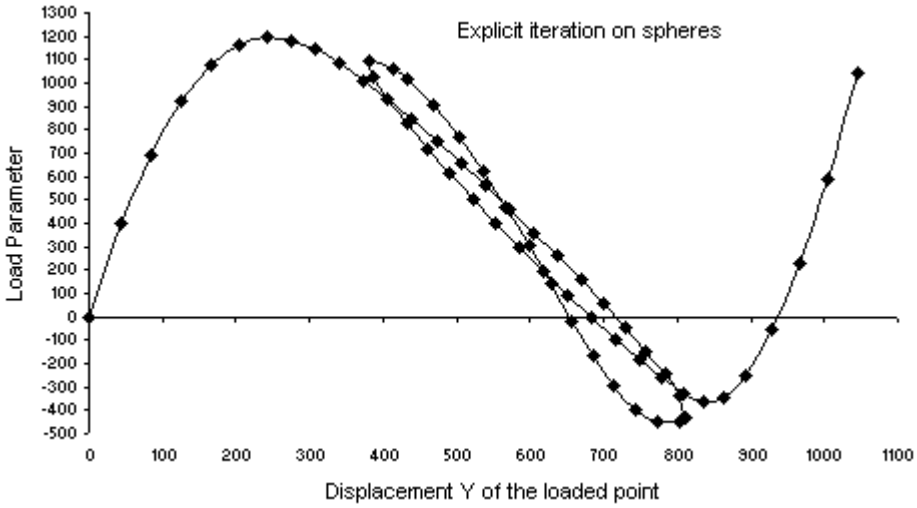

FIGURE 27. *Circular arch: Load parameter-vertical displacement curve of the loaded point (Explicit iteration on spheres).*

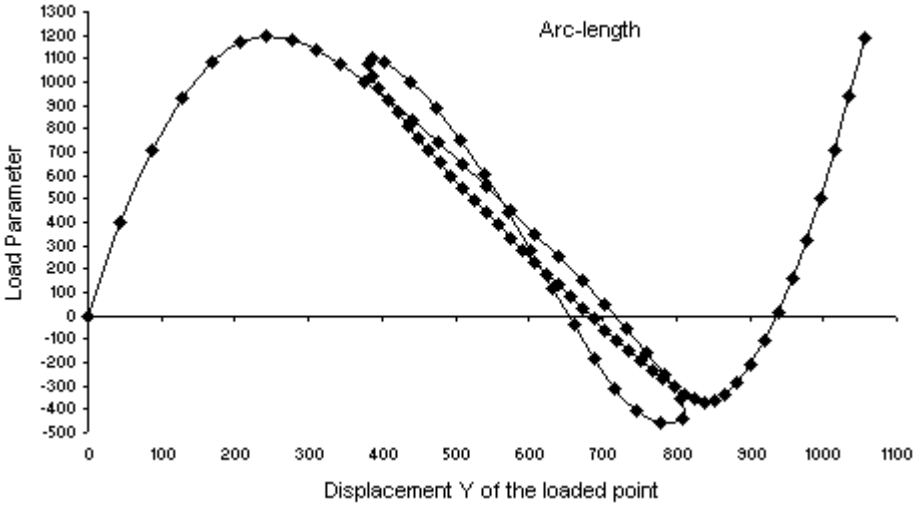

FIGURE 28. *Circular arch: Load parameter-vertical displacement curve of the loaded point (Arc-length method).*

Type of constraint equation	No. incr.	No.iter.
Newton-Raphson	failed	failed
Minimum residual displacement	59	220
Updated normal plane	61	230
Explicit iteration on spheres	62	234
Arc-length	82	304

TABLE 3. *Circular arch* - Performance and incrementation.

5.4. **Concrete bar.** A one-dimensional concrete bar submitted to a traction force F is studied hereafter. The bar has a section of $0.1x1m^2$ and a length of $1m$. It is modeled using 14 PLXLS elements of LAGAMINE (plane deformation). Figure 29 shows the boundary conditions adopted in order to avoid any 2D effects (the vertical displacements are prohibited along the boundaries, the right end of the bar is fixed horizontally). The external traction force is applied at the left end by means of a LICHA element.

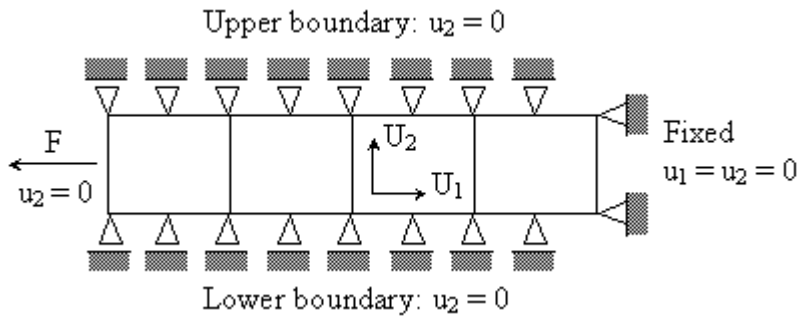

FIGURE 29. *Concrete bar - Boundary conditions.*

The constitutive relation used in order to reproduce the non-linear behavior of concrete is the classical Mazar's law [32],[33]. This constitutive relation is based on continuous damage mechanics and accounts for the asymmetric behavior of concrete under tension and compression. The parameters chosen for the damage law correspond to a typical concrete specimen ($E = 30GPa$, $\sigma_{tract} = 3MPa$, damage threshold $\epsilon_{d0} = 1.e^{-4}$). The load parameters are: an initial load F of 1.10^5, a final load F of 3.10^5, a maximum step of 2, a minimum step of 0.01 and a maximum number of iterations of 25. The stress-deformation curves for different constrain equations are shown in Figures 30 to 34. The parameters for *the direct adaption of the load increment* are $J_d = 3$ and $\gamma = 1/3$. The tolerance for the convergence norm is taken 10^{-4} for the displacements and 10^{-5} for the forces. The secant stiffness matrix is used for the resolution of the system.

Table 4 resumes the total number of increments for each method (the increments of the 'classic strategy' are not counted). The calculations stopped automatically by the program with the message ' stopped because too many steps have been performed with the minimum multiplicator'.

Type of constraint equation	No. incr.
Newton-Raphson	failed
Minimum residual displacement	26
Updated normal plane	26
Explicit iteration on spheres	26
Arc-length	27

TABLE 4. *Concrete bar* - Performance and incrementation.

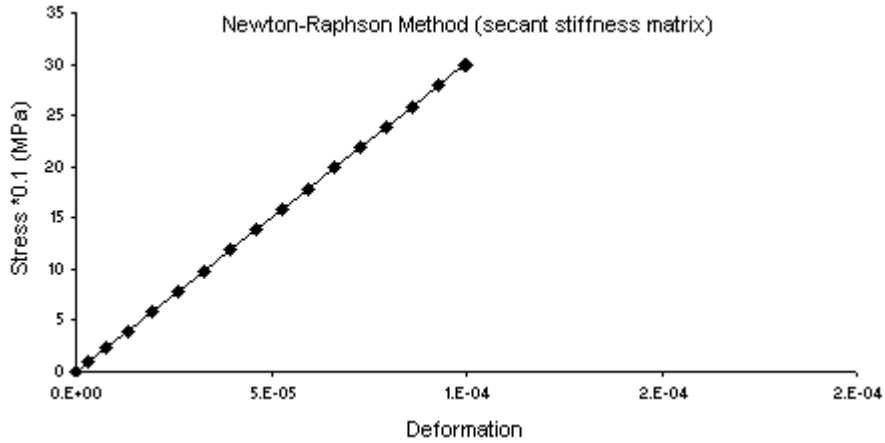

FIGURE 30. *Concrete bar: Stress-deformation curve (Newton-Raphson using the secant stiffness matrix).*

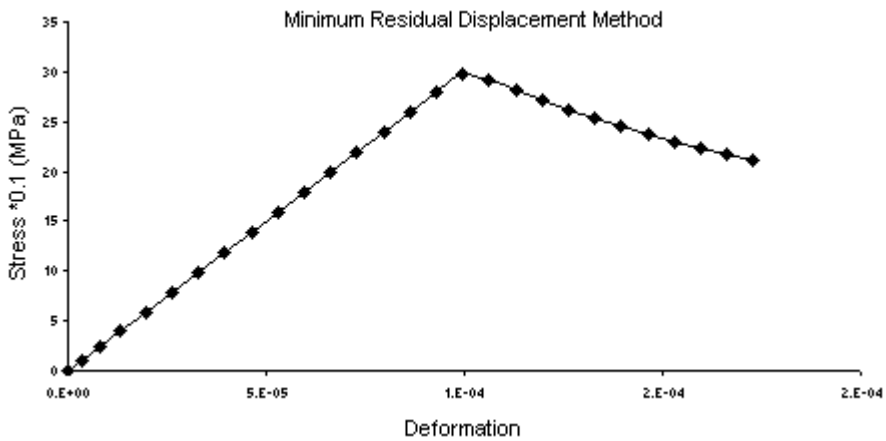

FIGURE 31. *Concrete bar: Stress-deformation curve (Minimum residual displacement).*

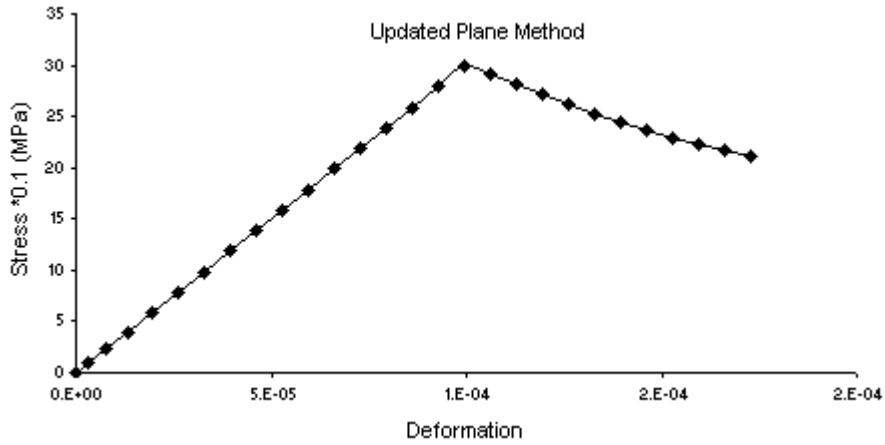

FIGURE 32. *Concrete bar: Stress-deformation curve (Updated normal plane).*

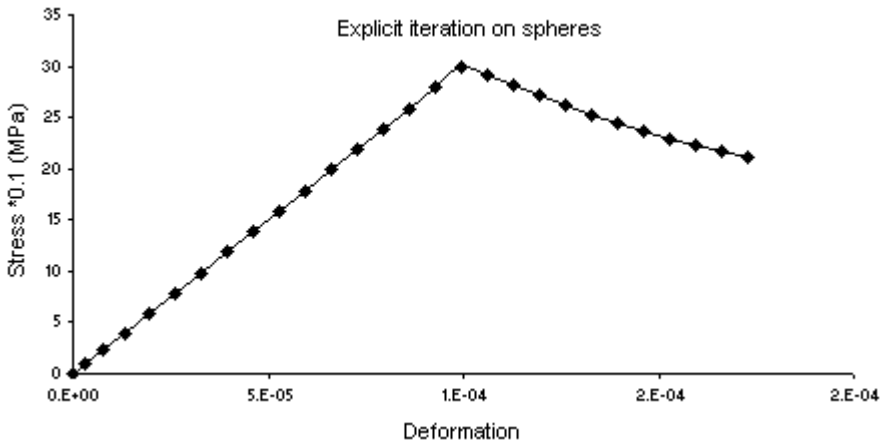

FIGURE 33. *Concrete bar: Stress-deformation curve (Explicit iteration on spheres).*

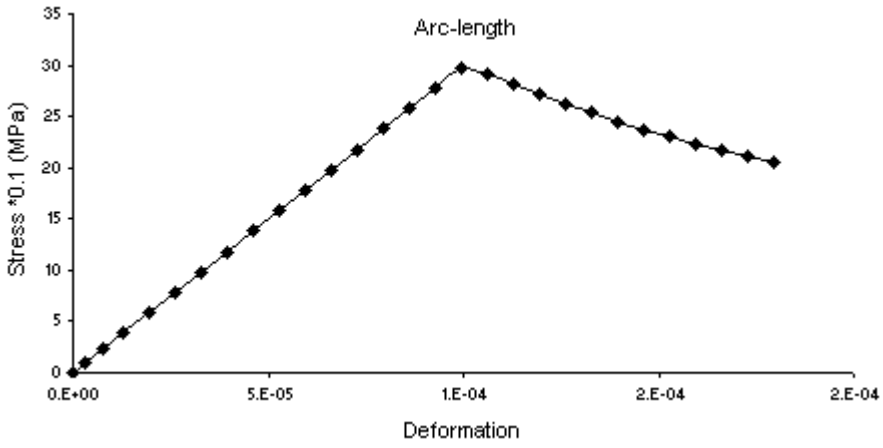

FIGURE 34. *Concrete bar: Stress-deformation curve (Arc-length method).*

6. Conclusions and Perspectives

Different automatic following path techniques have been presented and implemented into the finite element code LAGAMINE. For the challenging non-linear geometry cases treated in this report all methods were able to reproduce the behavior of the structures and performed in an almost identical way. However - as it is often mentioned in the literature - *the explicit iteration on spheres* and *the arc-length method* are able to simulate even very sharp snap-backs. Work is progress in order to demonstrate that argument with examples that the other methods fail to reproduce. For material non-linearities, a first example showed the efficiency of the different constrain equations. However, as it is already mentioned in section 3.4, physically non-linear problems often require the control of a *limited number* of degrees of freedom, due to localized deformations [25],[30]. The implementation of local following path techniques into LAGAMINE seems thus a necessary step in order to reproduce the non-linear behavior of higher order continua (second gradient models)...

List of Figures

1 *Newton-Raphson procedure: One iteration for a problem with a single DOF.* 2

2 *Newton-Raphson procedure: Several iterations within a step for a problem with a single DOF.* 3

3 *Full Newton-Raphson procedure (single DOF problem).* 3

4 *Modified Newton-Raphson procedure (single DOF problem).* 4

5 *Initial Newton-Raphson procedure (single DOF problem).* 5

6 *Critical points: B=Bifurcation, D,D'=Displacement limit points (snap-back), L,L'=Load limit points (snap-through).* 6

7 *Following path techniques: General incremental formulation (single DOF problem converged after $j = 3$ iterations).* 7

8 *The updated normal plane method (single DOF problem).* 12

9 *The consistently linearized method (single DOF problem).* 13

10 *Explicit iteration on spheres (single DOF problem).* 14

11 *Simple truss structure.* 19

12 *Simple truss structure: Load-displacement curve of the loaded point (Full Newton-Raphson).* 19

13 *Simple truss structure: Load-displacement curve of the loaded point (Minimum residual displacement).* 19

14 *Simple truss structure: Load-displacement curve of the loaded point (Updated normal plane).* 20

15 *Simple truss structure: Load-displacement curve of the loaded point (Explicit iteration on spheres).* 20

16 *Simple truss structure: Load-displacement curve of the loaded point (Arc-length method).* 21

17 *The Lee frame.* 21

18 *The Lee frame: Load-coordinate curve of the loaded point (Full Newton-Raphson).* 22
19 *The Lee frame: Load-coordinate curve of the loaded point (Minimum residual displacement).* 22
20 *The Lee frame: Load-coordinate curve of the loaded point (Updated normal plane).* 23
21 *The Lee frame: Load-coordinate curve of the loaded point (Explicit iteration on spheres).* 23
22 *The Lee frame: Load-coordinate curve of the loaded point (Arc-length method).* 24
23 *Swallow circular arch subject to near central point load.* 25
24 *Circular arch: Load parameter-vertical displacement curve of the loaded point (Full Newton-Raphson).* 25
25 *Circular arch: Load parameter-vertical displacement curve of the loaded point (Minimum residual displacement).* 26
26 *Circular arch: Load parameter-vertical displacement curve of the loaded point (Updated normal plane).* 26
27 *Circular arch: Load parameter-vertical displacement curve of the loaded point (Explicit iteration on spheres).* 27
28 *Circular arch: Load parameter-vertical displacement curve of the loaded point (Arc-length method).* 27
29 *Concrete bar - Boundary conditions.* 28
30 *Concrete bar: Stress-deformation curve (Newton-Raphson using the secant stiffness matrix).* 29
31 *Concrete bar: Stress-deformation curve (Minimum residual displacement).* 29
32 *Concrete bar: Stress-deformation curve (Updated normal plane).* 30
33 *Concrete bar: Stress-deformation curve (Explicit iteration on spheres).* 30
34 *Concrete bar: Stress-deformation curve (Arc-length method).* 31

List of Tables

1 *Simple truss structure* - Performance and incrementation. 21
2 *Lee frame* - Performance and incrementation. 24
3 *Circular arch* - Performance and incrementation. 27
4 *Concrete bar* - Performance and incrementation. 28

References

1. G. Alfano and M.A. Crisfield, *Finite element interface models for the delamination analysis of laminated composites:mechanical and computational issues*, International Journal for Numerical Methods in Engineering **50** (2001), no. 7, 1701–1736.
2. Inc. ANSYS, *Ansys 6.1 documentation*, Ansys Inc., 2004.
3. J.H. Argyris, *Continua and discontinua*, Proceedings of the 1st Conference on Matrix Methods in Structural Mechanics (Wright-Patterson A.F.B. OH, ed.), 1965, pp. 11–189.

4. K. Bathe and Dvorkin E.N., *On the automatic solution of non-linear finite element equations*, Computer and Structures **17** (1983), no. 5/6, 871–879.
5. K.J. Bathe, *Finite element procedures in engineering analysis*, Prentice Hall, Inc, New Jersey, 1996.
6. J.L. Batoz and G. Dhatt, *Incremental displacement algorithms for nonlinear problems*, International Journal for Numerical Methods in Engineering **14** (1979), 1262–1267.
7. P.X. Bellini and A. Chulya, *An improved automatic incremental algorithm for the efficient solution of non linear finite element equations*, Computer and Structures **26** (1987), no. 1/2, 99–110.
8. P.G. Bergan, *Solution techniques for non-linear finite element problems*, International Journal for Numerical Methods in Engineering **12** (1978), 1677–1696.
9. _____, *Solution algorithms for nonlinear structural problems*, Computer and Structures **12** (1980), 497–509.
10. _____, *Recent advances in non-linear computational mechanics*, ch. 2 - Automated incremental-iterative solution methods in structural mechanics, pp. 41–62, Hinton E. et al.(eds.) Pineridge Press: Swansea, U.K., 1982.
11. C. Bosco and A. Carpinteri, *Discontinuous constitutive response of brittle matrix fibrous composites*, Journal of the Mechanics and Physics of Solids **43** (1995), no. 2, 261–274.
12. R. Chambon, S. Crochepeyre, and R. Charlier, *An algorithm and a method to search bifurcation points in non-linear problems*, International Journal for Numerical Methods in Enginineering **51** (2001), 315–332.
13. R. Chambon and J.C. Moullet, *Uniqueness studies in boundary value problems involving some second gradient models*, Comput. Meth. Appl. Mech. Engng. **193** (2004), 2771–2796.
14. S.L. Chan, *Geometric and material nonlinear analysis of beam-columns and frames using the minimum residual displacement method*, International Journal of Numerical Methods in Engineering **26** (1988), 2657–2669.
15. Z. Chen and H.L. Schreyer, *Secant structural solution strategies under element constraint for incremental damage*, Computer Methods in Applied Mechanics and Engineering **90** (1991), 869–884.
16. M.J. Clarke and G.J. Hancock, *A study of incremental-iterative strategies for non-linear analyses*, International Journal for Numerical Methods in Engineering **29** (1990), 1365–1991.
17. M.A. Crisfield, *A fast incremental-iterative solution procedure that handles snap-through*, Computers and Structures **13** (1981), no. 1/3, 55–62.
18. _____, *An arc-length method including line searches and accelerations*, International Journal for Numerical Methods in Engineering **19** (1983), 1269–1289.
19. _____, *Snap-through and snap-back response in concrete structures and the dangers of under-integration*, International Journal for Numerical Methods in Engineering **22** (1986), 751–767.
20. _____, *Nonlinear finite element analysis of solids and structures, vol i: Essentials*, John Wiley, Chichester, 1991.
21. _____, *Nonlinear finite element analysis of solids and structures, vol ii: Advanced topics*, John Wiley, Chichester, 1991.
22. R. De Borst, *Computation of post-bifurcation and post-failure of strain-softening solids*, Computer and Structures **25** (1987), 211–224.
23. P.H. Feenstra and Schellekens J.C.J., *Self-adaptive solution algorithm for a constrained newton-raphson method*, Tech. Report Technical Report 25.2-91-2-13, Delft University of Technology, Delft, The Netherlands, 1991.
24. B.W.R. Forde and Stiemer S.F., *Improved arc length orthogonality methods for nonlinear finite element analysis*, Computer and Structures **27** (1987), no. 5, 625–630.
25. M.G.D Geers, *Enhanced solution control for physically and geometrically non-linear problems. part i - the subplane approach*, International Journal for Numerical Methods in Engineering **46** (1999), 177–204.
26. _____, *Enhanced solution control for physically and geometrically non-linear problems. part ii - comparative performance analysis*, International Journal for Numerical Methods in Engineering **46** (1999), 205–230.
27. H.B. Harrison, *Elastic-post-buckling response of plane frames*, Instability and Plastic collapse of steel structures, Granada (Morris L.J., ed.), 1983, pp. 56–65.

28. H.B. Hellweg and M.A. Crisfield, *A new arc-length method for handling sharp snap-backs*, Computers and Structures **66** (1998), no. 5, 705–709.
29. S. Lee, F.S. Manuel, and E.C. Rossow, *Large deflections and stability of elastic frames*, Journal of Enginnering Mechanics Division, Proceedings of the American Society of Civil Enginners **94** (1968), no. 2, 521–547.
30. E. Lorentz and P. Badel, *A new path-following constraint for strain-softening finite elements simulations*, International Journal for Numerical Methods in Engineering **60** (2004), 499–526.
31. I.M. May and Y. Duan, *A local arc-length procedure for strain softening*, Computer and Structures **64** (1997), no. 1/4, 297–303.
32. J. Mazars, *Application de la mécanique de l'endommagement au comportement non linéaire et à la rupture du béton de structure*, Ph.D. thesis, Université Paris 6, France, 1984, (Thèse de doctorat dès Sciences).
33. ———, *A description of micro and macroscale damage of concrete structures*, Engineering Fracture Mechanics **25** (1986), no. 5/6, 729–737.
34. G. Powell and J. Simons, *Improved iteration strategy for nonlinear structures*, International Journal for Numerical Methods in Engineering **17** (1981), 1455–1467.
35. E. Ramm, *Strategies for tracing the non-linear response near limit points*, Non-linear finite element analysis in structural mechanics (W. Wunderlich, ed.), Springer-Verlag, Berlin, 1981, pp. 63–89.
36. E. Riks, *An incremental approach to the solution of snapping and buckling problems*, International Journal of Solids and Structures **15** (1979), 529–551.
37. E. Riks, C. Rankin, and F.A. Brogan, *On the solution of mode jumping phenomena in thin wall structures*, Computer Methods in Applied Mechanics and Engineering **136** (1996), 59–92.
38. J.C.J. Schellekens, P.H. Feenstra, and R. de Borst, *A self-adaptive load estimator based on strain energy*, Computational Plasticity, Fundamentals and Applications (D.R.J. Owen, E. Onate, and E. Hinton, eds.), CIMNE, Barcelona, Pineridge Press, Swansea, U.K., April, 1992, pp. 187–198.
39. K.H. Schweizerhof and P. Wriggers, *Consistent linearization for path following methods in nonlinear finite element analysis*, Computers Methods in Applied Mechanics and Engineering **59** (1986), 261–279.
40. G. Skeie, O.C. Astrup, and P.G. Bergan, *Applications of adapted, nonlinear solution strategies*, Advances in Finite Element Technology, N.-E. Wiberg, CIMNE, Barcelona, 1995, pp. 212–236.
41. G.A. Wempner, *Discrete approximations related to nonlinear theories of solids*, International Journal of Solids and Structures **7** (1971), 1581–1599.
42. Z. Yang and J. Chen, *Fully automatic modelling of cohesive discrete crack propagation in concrete beams using local arc-length methods*, International Journal of Solids and Structures **41** (2004), 801–826.
43. Z.J. Yang and D. Proverbs, *A comparative study of numerical solutions to non-linear discrete crack modelling of concrete beams involving sharp snap-backs*, Engineering Fracture Mechanics **71** (2004), 81–105.
44. O.C. Zienkiewicz and R.L. Taylor, *The finite element method, fifth ed.*, Butterworth-Heinemann, 2000.

Panagiotis Kotronis, Assistant Professor, L3S, CNRS-UJF-INPG, Grenoble, France
E-mail address: `Panagiotis.Kotronis@inpg.fr`

Frédéric Collin, Chargé de Recherche, Geomac, FNRS-ULG, Liège, Belgium
E-mail address: `f.collin@ulg.ac.be`