Modular implementation framework of partitioned path-following strategies: Formulation, algorithms and application to the finite element software Cast3M

Hugo Luiz Oliveira, Giuseppe Rastiello, Alain Millard, Ibrahim Bitar, Benjamin Richard

Modular implementation framework of partitioned path-following strategies: Formulation, algorithms and application to the finite element software Cast3M

Hugo Luiz Oliveira, Giuseppe Rastiello, Alain Millard, Ibrahim Bitar, Benjamin Richard

Modular implementation framework of partitioned path-following strategies: formulation, algorithms and application to the finite element software Cast3M

Hugo Luiz Oliveira[a], Giuseppe Rastiello[a], Alain Millard[a], Ibrahim Bitar[b], Benjamin Richard[b]

[a] *Université Paris-Saclay, CEA, Service d'études mécaniques et thermiques, 91191, Gif-sur-Yvette, France*

[b] *Institut de Radioprotection et de Sûreté Nucléaire (IRSN), PSN-EXP/SES/LMAPS, 92262 Fontenay-aux-Roses Cedex, France*

Abstract

Damage, cracking, and strain localization mechanisms often lead to unstable structural responses characterized by snap-backs (i.e., force and displacement decrease simultaneously). Standard nonlinear Newton-based solution algorithms with displacement/force control cannot capture the equilibrium curve in its entirety. This can be overcome using path-following formulations. A general (i.e., valid for any finite element code) implementation framework can be designed and applied to the Cast3M software by collecting the essential concepts and the formalism of partitioned path-following arc-length algorithms. Thanks to these developments, Cast3M is now capable of processing path-following equations without any major modifications. Three path-following constraints were selected to demonstrate the applicability of this framework: a first one on the combination of the displacement increment at a given set of nodes, a second one on the maximum strain increment over the computational domain, and a third one on the maximum elastic predictor of the damage/plastic criterion function over the computational domain. Two- and three-dimensional strain localization simulations show that the proposed framework behaves in a stable and convergent manner, even when multiple severe snap-back instabilities are present. Users of Cast3M shall find the proposed study helpful in that it allows them to focus on developing new path-following equations for the software. Cast3M is developed by the French Alternative Energies and Atomic Energy Commission (CEA) and freely available for research purposes. The developments discussed in this paper have been made available to the user/developer community along with Cast3M 2021 (release date: June 2021).

Keywords: Path-following methods, Snap-back response, Nonlinear material models, Strain localization, Cast3M software

1. Introduction

Stability is a constant worry in most activities related to the design and construction of structures. This importance comes from the fact that, without stability, one can hardly guarantee that the structures will fulfill the requirements for

*Corresponding author
Email address: giuseppe.rastiello@cea.fr (Giuseppe Rastiello)

PDF-XChange Viewer Click to buy NOW! www.docu-track.com

which they were designed. A lack of stability can have disastrous consequences.

Scientific literature generally distinguishes between two sources of instabilities: geometrical instabilities, which are induced by large displacements (Riks, 1972, Ramm, 1981, Crisfield, 1981); and material-induced instabilities (De Borst, 1987, Verhoosel et al., 2009a, Rastiello et al., 2019). The mathematical principles leveraged in analyses of the latter type of instability originate in the work of Hadamard (1903) for elastic materials, and Hill (1962), Mandel (1966), Thomas (1961), Rice (1976), Needleman and Rice (1978) for elastoplastic materials. In this approach, the localization phenomena are associated with a loss of ellipticity of the governing equations, which may lead to the formation of shear bands depending on the ductility of the material. For quasi-brittle materials, it may also be associated with dissipative phenomena occurring at the material level (e.g., cracking and strain localization (Bažant, 2000)). Both sources of instability may sometimes coexist (Verhoosel et al., 2009a, Bellora and Vescovini, 2016).

This work focuses on material-induced instabilities that are currently encountered in computational fracture/damage mechanics. The developed framework is, however, general, and can be used to treat geometrical nonlinearities.

Fracture and strain localization phenomena are often responsible for unstable structural responses characterized by snap-backs (De Borst, 1987), i.e., new equilibrium configurations are found, leading to a decrease in both external loads and displacements. In that case, standard Newton-based procedures, which rely on force and displacement control, actually fall short and show discontinuities in the force-displacement equilibrium curves. From a structural mechanics viewpoint, these discontinuities represent a loss of information because they hide equilibrium states that may be relevant. From a numerical modeling viewpoint, these "jumps" correspond to abrupt changes in the constitutive model's internal variables, with detrimental effects on the convergence of the solution algorithm.

The so-called "continuation" (in the mathematics community) or "path-following" (in the engineering community) methods provide a solution to this issue of discontinuity. Riks (1979) first suggested that the control variable during the analysis should neither be a displacement nor a force and that these should be controlled indirectly instead. In order to handle non-linearity, an additional unknown and a supplementary equation, the path-following equation, are required. Despite the significant benefits of Riks' proposal, following it means that one ends up solving an augmented equilibrium problem where the stiffness-like matrix that needs to be inverted is non-symmetric. To solve this issue, Crisfield (1981, 1983) proposed a specific way of splitting the displacement correction, which allowed for the use of classical solvers without any modifications. In that case, the displacement correction can be computed during the iterative solution process in a partitioned/staggered manner.

The global constraint equations[1] proposed in (Riks, 1979, Crisfield, 1981, Ramm, 1981) may be well suited for geometrical instabilities, but they may fail in the presence of structural instabilities induced by material non-linearities, especially when localization phenomena occur (De Borst, 1987). In that case, only a limited number of DOFs is

[1] Called "global" since they account for all degrees of freedom (DOFs) of the problem.

responsible for the unstable response observed at the structural level, and specific constraint equations have to be written.

Since De Borst (1987)'s work, many algorithms have been proposed in the literature to overcome these limitations. Among these, are worth mentioning: constraint equations on the rate of variation of selected sets of DOFs (De Borst, 1987, May and Duan, 1997); on strain measures (Chen and Schreyer, 1990, Geers, 1999, Pohl et al., 2014); or quantities associated with the energy dissipation occurring in the system during the development of non-linearities (Gutiérrez, 2004, Verhoosel et al., 2009b, Lorentz and Badel, 2004, Singh et al., 2016, Stanić and Brank, 2017, Barbieri et al., 2017). Mixed path-following approaches in terms of stresses, strains, displacements, damage, and other variables were also proposed (Garcea et al., 1998, Formica et al., 2002, Bilotta et al., 2012, Magisano et al., 2017, 2019) to improve the convergence of standard path-following methods.

These techniques were successfully used in finite element simulations where the material response was modeled using local and nonlocal continuum damage mechanics laws (Lorentz and Badel, 2004, Rastiello et al., 2018, Thierry et al., 2020), phase-field models (Singh et al., 2016), and Thick Level-Set (TLS) damage formulations (Moreau et al., 2017). Path-following methods were also used in strong discontinuity simulations of failure in solids, where cracks were represented as zero-thickness interface finite elements (Alfano and Crisfield, 2003, Lorentz and Badel, 2004), according to the extended (Massin et al., 2011, Wang and Waisman, 2018) or the embedded finite element method (Oliver et al., 2008, Brank et al., 2016, Cazes et al., 2016, Rastiello et al., 2019).

This overview of the evolution of path-following methods reveals the essential aspects of the present work. Firstly, although remarkable advances have been made in the last decades, the theoretical framework established in the pioneering works by Riks (1972), Crisfield (1981), and De Borst (1987) remains unchanged. Secondly, the numerical procedures discussed above still haven't been made available to the Cast3M users community.[2]

These two aspects bring us to the main purpose of this study, namely proposing a general framework, applicable to the Cast3M software, capable of processing instabilities generated by strain localization. The proposed framework is designed in a flexible and modular way to make the study of new path-following conditions possible. In this case, the proposed framework can serve as the starting point for further investigations involving path-following methodologies.

This paper is structured as follows. First, the boundary value problem that needs solving is presented according to the formalism of path-following theories. Then, the procedure used to find the solution of the problem is abstracted in order to create the proposed framework. This framework's main feature is its modularity; that is, it is designed to allow Cast3M to accept different user-defined path-following conditions without modifying the framework. Once the framework is introduced, three path-following equations are detailed. These will showcase how to use the proposed framework. These criteria are of practical interest and encompass both dissipative and non-dissipative formulations.

[2]Cast3M (http://www-cast3m.cea.fr/index.php) is a well-established finite element software (Verpeaux et al., 1989) developed and distributed by the CEA, the French Alternative Energies and Atomic Energy Commission (https://www.cea.fr/english).

Finally, representative two and three-dimensional case studies involving one or more snap-back phases are discussed. The paper ends with a summary of the findings.

2. Path-following methods

This section starts with the introduction of the boundary value problem that needs solving. Then, some dissipative and non-dissipative path-following constraints are introduced. The term "dissipative" refers to constraint equations which only describe dissipative solutions (i.e., the dissipation in the system stays positive). Conversely, the term "non-dissipative" refers to constraint equations in which this characteristic is not fulfilled. In other words, one can also describe artificial/elastic unloading paths. Finally, some details concerning the discrete finite element problem are given.

2.1. Boundary value problem

Let $\Omega \subset \mathbb{R}^n$ (with $n = 1, 2, 3$ the problem's dimension) represent the structure of interest (fig. 1) for this study and $t \in [0, T]$ be the pseudo-time under consideration. The boundary $\partial\Omega \subset \mathbb{R}^n$ of the structure is composed of two complementary parts: $\partial\Omega_u \subset \partial\Omega$, where the external actions can be specified in terms of displacements (Dirichlet boundary conditions) and $\partial\Omega_t \subset \partial\Omega$, where the external actions are defined in terms of forces (Neumann boundary conditions). Moreover $\mathbf{n}$ denotes the outward normal vector field to $\partial\Omega$.

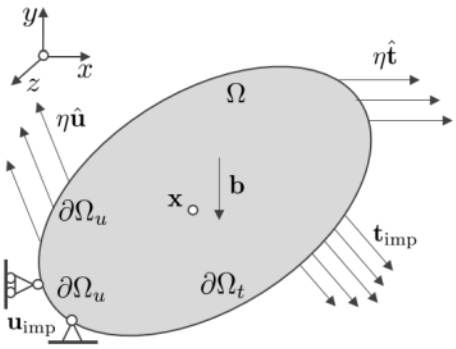

Figure 1: General domain of study.

Under the small perturbations assumption, solving the time-independent boundary value problem (BVP) consists in finding the vector-valued displacement field $\mathbf{u} = \mathbf{u}(\mathbf{x}, t)$ and the real scalar parameter η such that:

$$\nabla \cdot \boldsymbol{\sigma}(\mathbf{u}) + \mathbf{b} = 0 \qquad \mathbf{x} \in \Omega \tag{1}$$

$$\boldsymbol{\sigma}(\mathbf{u}) = \boldsymbol{\mathcal{A}}(\boldsymbol{\epsilon}(\mathbf{u})) \qquad \mathbf{x} \in \Omega \tag{2}$$

$$\boldsymbol{\epsilon}(\mathbf{u}) = \mathrm{Sym}(\nabla \mathbf{u}) \qquad \mathbf{x} \in \Omega \tag{3}$$

$$\mathbf{u} = \mathbf{u}_{\mathrm{imp}} + \eta \hat{\mathbf{u}} \qquad \mathbf{x} \in \partial\Omega_u \tag{4}$$

$$\boldsymbol{\sigma} \cdot \mathbf{n} = \mathbf{t}_{\mathrm{imp}} + \eta \hat{\mathbf{t}} \qquad \mathbf{x} \in \partial\Omega_t \tag{5}$$

$$p(\mathbf{u}, \eta) = 0 \qquad \mathbf{x} \in \Omega \tag{6}$$

where $\boldsymbol{\sigma}(\mathbf{u})$ stands for the Cauchy stress tensor, $\boldsymbol{\epsilon}(\mathbf{u})$ is the infinitesimal strain tensor, $\boldsymbol{\mathcal{A}}$ is the constitutive relationship between $\boldsymbol{\sigma}$ and $\boldsymbol{\epsilon}$, Sym denotes the symmetric part operator, $\mathbf{u}_{\mathrm{imp}}$ is the imposed displacement, $\hat{\mathbf{u}}$ is the direction of the indirectly-controlled displacement, $\mathbf{t}_{\mathrm{imp}}$ is the imposed traction vector, $\hat{\mathbf{t}}$ is a vector providing the direction of the indirectly-controlled tractions, "$\nabla\cdot$" denotes the divergence operator and "∇" the gradient. Finally, $p(\mathbf{u}, \eta)$ is the path-following constraint equation.

2.2. *Finite Element formulation*

The finite element formulation of the BVP can easily be derived once the problem has been rewritten in its variational form using the Principle of Virtual Works (Hughes, 1987, Zienkiewicz et al., 2005, De Borst et al., 2012). Once the variational formulation is established, a finite element discretization Ω^h of Ω is considered. This discretization is such that inside each finite element $\Omega_e \subset \Omega^h$, the unknown field $\mathbf{u}$ can be approximated from its nodal values ($\mathbf{d}$) through polynomial shape functions.

Using a double Lagrange multipliers formalism (Pegon and Anthoine, 1997) for imposing Dirichlet boundary conditions, solving the finite element problem becomes a matter of finding the nodal displacement vector $\mathbf{d}$, the discrete Lagrange multipliers $\boldsymbol{\lambda}_1$ and $\boldsymbol{\lambda}_2$ (i.e., the nodal reaction forces), and the load multiplier η such that (Richard et al., 2019):

$$\begin{aligned}\mathbf{r}(\mathbf{d}, \boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2, \eta) = \mathbf{f}^{\mathrm{int}}(\mathbf{d}) + \mathbf{C}^\top \boldsymbol{\lambda}_1 + \mathbf{C}^\top \boldsymbol{\lambda}_2 \\ - \eta \hat{\mathbf{f}}^{\mathrm{ext}} - \mathbf{f}^{\mathrm{ext}}_{\mathrm{imp}} = 0\end{aligned} \tag{7}$$

$$\mathbf{l}_1(\mathbf{d}, \boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2, \eta) = \mathbf{C}\mathbf{d} + \boldsymbol{\lambda}_1 - \boldsymbol{\lambda}_2 - \eta \hat{\mathbf{d}} - \mathbf{d}_{\mathrm{imp}} = 0 \tag{8}$$

$$\mathbf{l}_2(\mathbf{d}, \boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2, \eta) = \mathbf{C}\mathbf{d} - \boldsymbol{\lambda}_1 + \boldsymbol{\lambda}_2 - \eta \hat{\mathbf{d}} - \mathbf{d}_{\mathrm{imp}} = 0 \tag{9}$$

$$p(\mathbf{d}, \eta) = 0 \tag{10}$$

where $\hat{\mathbf{d}}$ is the discrete counterpart of $\hat{\mathbf{u}}$ and:

$$\mathbf{f}^{\text{int}}(\mathbf{d}) = \mathop{\mathbf{A}}_{e=1}^{n_{el}} \int_{\Omega_e} \mathbf{B}^\top \boldsymbol{\sigma}(\mathbf{d})\, dv \tag{11}$$

$$\mathbf{f}^{\text{ext}}_{\text{imp}} = \mathop{\mathbf{A}}_{e=1}^{n_{el}} \int_{\Gamma_e} \mathbf{N}^\top \mathbf{s}_{\text{imp}}\, ds + \mathop{\mathbf{A}}_{e=1}^{n_{el}} \int_{\Omega_e} \mathbf{N}^\top \mathbf{b}\, dv \tag{12}$$

$$\hat{\mathbf{f}}^{\text{ext}} = \mathop{\mathbf{A}}_{e=1}^{n_{el}} \int_{\Gamma_e} \mathbf{N}^\top \hat{\mathbf{s}}\, dv \tag{13}$$

In eqs. (11) to (13) the symbol $\mathbf{A}$ denotes the standard assembly operator over the n_{el} elements of Ω^h, $\mathbf{N}$ is the displacement interpolation (shape function) matrix, $\mathbf{B}$ denotes the strain-displacement matrix, "$\top$" denotes the transpose operator, and Γ_e is the element boundary. Finally, $(\mathbf{s}_{\text{imp}}, \hat{\mathbf{s}})$ are the discretized counterparts of $(\mathbf{t}_{\text{imp}}, \hat{\mathbf{t}})$.

According to this formulation, $\boldsymbol{\lambda} = \boldsymbol{\lambda}_1 + \boldsymbol{\lambda}_2$ corresponds to the reaction forces at the nodes where Dirichlet boundary conditions are imposed and $\boldsymbol{\lambda}_1 - \boldsymbol{\lambda}_2 = \mathbf{0}$. This formalism is the one used in Cast3M, since it prevents us from having to solve a saddle-point problem, as opposed to the use of a single set of Lagrange multipliers to enforce the Dirichlet boundary conditions (Babuška, 1973).

2.3. Linearized problem

Equations (7) to (10) are discretized in pseudo-time according to an ordered sequence of steps $[t_0, t_1, \ldots, T]$, such that $t_{n+1} > t_n$ for all $n \in \mathbb{N}$, and the problem is solved in an incremental manner. Given the solution $\mathbf{x}_n = \{\mathbf{d}_n, \boldsymbol{\lambda}_{1,n}, \boldsymbol{\lambda}_{2,n}, \eta_n\}^\top$ at time t_n, one searches for $\Delta\mathbf{x} = \{\Delta\mathbf{d}, \Delta\boldsymbol{\lambda}_1, \Delta\boldsymbol{\lambda}_2, \Delta\eta\}^\top$, the solution variation in the interval $[t_n, t_{n+1}]$, such that $\mathbf{x}_{n+1} = \mathbf{x}_n + \Delta\mathbf{x}$ at time t_{n+1}.

Since nonlinear constitutive laws are commonly used in solid mechanics computations, the incremental problem is solved by means of an iterative procedure. Accordingly, the total solution increment at global iteration $k+1$ is written as $\Delta\mathbf{x}^{k+1} = \Delta\mathbf{x}^k + \delta\mathbf{x}^{k+1}$, where $\delta\mathbf{x}^{k+1} = \{\delta\mathbf{d}^{k+1}, \delta\boldsymbol{\lambda}_1^{k+1}, \delta\boldsymbol{\lambda}_2^{k+1}, \delta\eta^{k+1}\}^\top$ denotes the correction of the solution between iterations k and $k+1$.

The subscript $n+1$ has been omitted in the previous equations for the sake of keeping them short. The same nomenclature will be used in the rest of this article. As a result, any quantity without a subscript shall actually refer to the pseudo-time step t_{n+1}.

2.3.1. Global equilibrium equations

The incremental equilibrium equations that have to be solved at each iteration arise from the linearization of eqs. (7) to (9) based on the solution at iteration k.

At iteration $k+1$ of the nonlinear solution process, the iterated solution is thus computed from:

$$\tilde{\mathbf{K}}^k \delta\mathbf{a}^{k+1} = -\tilde{\mathbf{r}}^k + \delta\eta^{k+1}\tilde{\mathbf{f}} \tag{14}$$

where:

$$\tilde{\mathbf{K}}^k = \begin{bmatrix} \mathbf{K}^k & \mathbf{C}^\top & \mathbf{C}^\top \\ \mathbf{C} & \mathbf{I} & -\mathbf{I} \\ \mathbf{C} & -\mathbf{I} & \mathbf{I} \end{bmatrix} \tag{15}$$

$$\delta\mathbf{a}^{k+1} = \{\delta\mathbf{d}^{k+1},\, \delta\boldsymbol{\lambda}_1^{k+1},\, \delta\boldsymbol{\lambda}_2^{k+1}\}^\top \tag{16}$$

$$\tilde{\mathbf{r}}^k = \{\mathbf{r}^k,\, \mathbf{l}_1^k,\, \mathbf{l}_2^k\}^\top \tag{17}$$

$$\tilde{\mathbf{f}} = \{\hat{\mathbf{f}}^{\text{ext}},\, \hat{\mathbf{d}},\, \hat{\mathbf{d}}\}^\top \tag{18}$$

with $\mathbf{r}^k = \mathbf{r}(\mathbf{d}^k, \boldsymbol{\lambda}_1^k, \boldsymbol{\lambda}_2^k, \eta^k)$ and $\mathbf{l}_\bullet^k = \mathbf{l}_\bullet(\mathbf{d}^k, \boldsymbol{\lambda}_1^k, \boldsymbol{\lambda}_2^k, \eta^k)$ (with $\bullet = 1, 2$).

The formalism introduced by Crisfield (1983) allows us to write the unknown vector $\delta\mathbf{a}^{k+1}$ as the sum of two independent contributions:

$$\delta\mathbf{a}^{k+1} = \delta\mathbf{a}_{\text{I}}^{k+1} + \delta\eta^{k+1}\delta\mathbf{a}_{\text{II}}^{k+1} \tag{19}$$

where $\delta\mathbf{a}_{\text{I}}^{k+1}$ and $\delta\mathbf{a}_{\text{II}}^{k+1}$ can be calculated by simply knowing the solution at the k-th iteration:

$$\delta\mathbf{a}_{\text{I}}^{k+1} = -(\tilde{\mathbf{K}}^k)^{-1}\tilde{\mathbf{r}}^k \quad \delta\mathbf{a}_{\text{II}}^{k+1} = (\tilde{\mathbf{K}}^k)^{-1}\tilde{\mathbf{f}} \tag{20}$$

2.3.2. Path-following constraint equations

$\delta\eta^{k+1}$ can be computed in several ways depending on the nature of the path-following equation. In particular:

(i) When function p is linear with respect to $\mathbf{d}$, one can obtain $\delta\eta^{k+1}$ directly.

(ii) Conversely, when p is nonlinear, it has to be linearized in order to find $\delta\eta^{k+1}$. However, such linearization is not always possible, e.g., when the constraint equation is expressed by maximizing some quantity over the computational domain. In that case, two options are available:

- When function p is differentiable with respect to $\mathbf{d}$ and η. Differentiability can be guaranteed, for instance, when path-following constraints are expressed in terms of sets of global DOFs fixed *a priori* (De Borst, 1987) or in terms of the dissipation over the whole structure (Gutiérrez, 2004, May et al., 2016, Verhoosel et al., 2009a). In such a case, the linearization based on the solution at the k-th iteration gives:

$$p^k + (\mathbf{h}^k)^\top\delta\mathbf{d}^{k+1} + w^k\delta\eta^{k+1} = 0 \tag{21}$$

where:

$$p^k = p(\mathbf{d}^k, \eta^k) \tag{22}$$

$$\mathbf{h}^k = (\partial p/\partial \mathbf{d})^k \tag{23}$$

$$w^k = (\partial p/\partial \eta)^k \tag{24}$$

Finally, substituting eq. (19) into eq. (21) and solving for $\delta\eta^{k+1}$ gives:

$$\delta\eta^{k+1} = -\frac{p^k + (\mathbf{h}^k)^\top \delta\mathbf{d}_{\mathrm{I}}^{k+1}}{w^k + (\mathbf{h}^k)^\top \delta\mathbf{d}_{\mathrm{II}}^{k+1}} \tag{25}$$

– When general and robust constraint equations try to establish a load multiplier variation such that the field of interest $g = g(\mathbf{d}, \eta)$ (e.g., strain field variation, local energy dissipation variation, elastic predictor of the damage/plasticity criterion) over Ω^h reaches a maximum value for a user-defined step length ($\Delta\tau$).

This kind of constraint equations takes the following form:

$$p^{k+1} = \max_{\alpha \in \Omega^h} (g_\alpha^{k+1}) - \Delta\tau = 0 \tag{26}$$

where α denotes a portion (or a numerical entity) of the discretized domain (e.g., a node, an integration point). In the rest of this article, $(\bullet)_\alpha$ indicates that the quantity $(\bullet)$ is computed at the position of the entity α.

The main drawback of this approach is that the maximum operator function renders (26) non-differentiable. As a consequence, a direct estimation of $\delta\eta^{k+1}$ is not possible even though g_α^{k+1} is linear, or can be linearized, with respect to $\delta\eta^{k+1}$.

Nevertheless, the computation can be performed through an iterative procedure, e.g., with a nested interval algorithm (Lorentz and Badel, 2004).

2.4. Staggered solution

Thanks to the formalism introduced in Crisfield (1983), it is possible to solve the augmented global equilibrium problem (eqs. (7) to (10)) in a sequential manner (Algorithm 1).

Indeed, since the linearization of the equation (14) that expresses the structural equilibrium does not take into account the path-following constraint in (10), the strategy for finding a solution to the equilibrium problem preserves the essential characteristics of Newton's method, which is used in commercial FEM softwares.

Therefore, one can implement path-following solvers in any FE analysis software provided that one can modify the main FEM matrices directly.

PDF-XChange Viewer Click to buy NOW! www.docu-track.com

Algorithm 1: General finite element implementation of path-following solvers using the displacement breakdown method of Crisfield (1983).

conv $\leftarrow 0$, $k \leftarrow 0$

$\mathbf{a}^k \leftarrow \mathbf{a}_n$, $\eta^k \leftarrow \eta_n$

$\Delta\mathbf{a}^k = 0$, $\Delta\eta^k = 0$

$\tilde{\mathbf{r}}^k = \tilde{\mathbf{r}}^k(\mathbf{d}^k, \boldsymbol{\lambda}_1^k, \boldsymbol{\lambda}_2^k, \eta^k; \mathbf{f}_{\text{imp}}^{\text{ext}}, \mathbf{d}_{\text{imp}})$

while conv $= 0$ **do**

  Compute $\mathbf{K}^k$

  Build $\tilde{\mathbf{K}}^k$

  Solve $\tilde{\mathbf{K}}^k \delta\mathbf{a}_{\text{I}}^{k+1} = -\tilde{\mathbf{r}}^k$ for $\delta\mathbf{a}_{\text{I}}^{k+1}$

  Solve $\tilde{\mathbf{K}}^k \delta\mathbf{a}_{\text{II}}^{k+1} = \tilde{\mathbf{f}}$ for $\delta\mathbf{a}_{\text{II}}^{k+1}$

  Solve the constraint equation to compute $\delta\eta^{k+1}$

  Compute $\eta^{k+1} \leftarrow \eta^k + \delta\eta^{k+1}$

  Compute $\mathbf{a}^{k+1} \leftarrow \mathbf{a}^k + \delta\mathbf{a}_{\text{I}}^{k+1} + \delta\eta^{k+1}\delta\mathbf{a}_{\text{II}}^{k+1}$

  Compute $\tilde{\mathbf{r}}^{k+1}$

  if Convergence is met **then**

    conv $= 1$

    Solution at time t_{n+1}: $\eta \leftarrow \eta^{k+1}$

    Solution at time t_{n+1}: $\mathbf{a} \leftarrow \mathbf{a}^{k+1}$

  else

    $k \leftarrow k + 1$

  end if

PDF-XChange Viewer Click to buy NOW! www.docu-track.com

2.5. *Additional remarks*

If the constraint equation is linear or linearizable with respect to the unknown displacements and the load multiplier, solving the problem with the Crisfield formalism (i.e., partitioned approach) is entirely equivalent to using Riks' direct approach. Indeed, the solution of a non-symmetric system of equations in the Riks' approach can be circumvented with the well-known Shermann-Morrison formula (May et al., 2016, Verhoosel et al., 2009a). The main advantage of the partitioned approach is that differentiability of the constraint equation is not required, which increases the flexibility of the developed numerical framework.

3. Cast3M implementation

This section starts with a brief presentation of the main modifications implemented in the standard nonlinear solver of the code. The focus will then shift to the procedure for solving the path-following constraint equations. In particular, details will be given regarding the default constraint equation and the possibility of specifying user-defined constraint equations by "overloading" the standard procedure.

All implementations discussed in this study are carried out using the Gibiane scripting language (Ebersolt et al., 1987), an object-oriented language developed, maintained, and distributed by the CEA. This language is used in Cast3M (Verpeaux et al., 1989). Notice, however, that the modular implementation framework proposed herein and the considered constraint equations (formulation, solving techniques) are entirely general and can be adapted to any finite element software.

3.1. Cast3M procedures

The main modifications implemented in the Cast3M code concern three existing procedures (PAS_DEFA, PAS_INIT, and UNPAS) of the main nonlinear incremental Cast3M solver (PASAPAS). These procedures are modified (see fig. 2 for more details) in order to create and handle all the data structures for the path-following methods; to compute the solution variations $\delta\mathbf{a}_{\mathrm{I}}^{k+1}$ and $\delta\mathbf{a}_{\mathrm{II}}^{k+1}$; and to build the total correction $\delta\mathbf{a}^{k+1}$ once the load multiplier variation $\delta\eta^{k+1}$ has been determined. Moreover, an additional procedure called PILOINDI is added to handle the constraint equation's solution (i.e., to compute $\delta\eta^{k+1}$ depending on the selected path-following equation) once $\delta\mathbf{a}_{\mathrm{I}}^{k+1}$ and $\delta\mathbf{a}_{\mathrm{II}}^{k+1}$ have been determined.

3.2. The PILOINDI procedure and default constraint equation

The path-following equation implemented as the default option in the PILOINDI procedure is based on the maximization of a scalar measure of the strain field variation (Control by Maximum Strain Increment, CMSI) through Ω^h or a part of it, $\Omega^h_{\mathrm{red}} \subset \Omega^h$.

The constraint equation reads similarly as in eq. (26), with function g_α^{k+1} defined as:

$$g_\alpha^{k+1} = \mathbf{q}_\alpha^\top \Delta\boldsymbol{\epsilon}_\alpha^{k+1} \tag{27}$$

with $\mathbf{q}_\alpha$ being the direction of the strain vector at the position of the integration point α at the previous time step:

$$\mathbf{q}_\alpha = \frac{\mathbf{B}_\alpha \mathbf{d}_n}{||\mathbf{B}_\alpha \mathbf{d}_n||} \tag{28}$$

Given the additive decomposition in eq. (19), $\Delta\boldsymbol{\epsilon}_\alpha^{k+1}$ depends linearly on $\delta\eta^{k+1}$ according to:

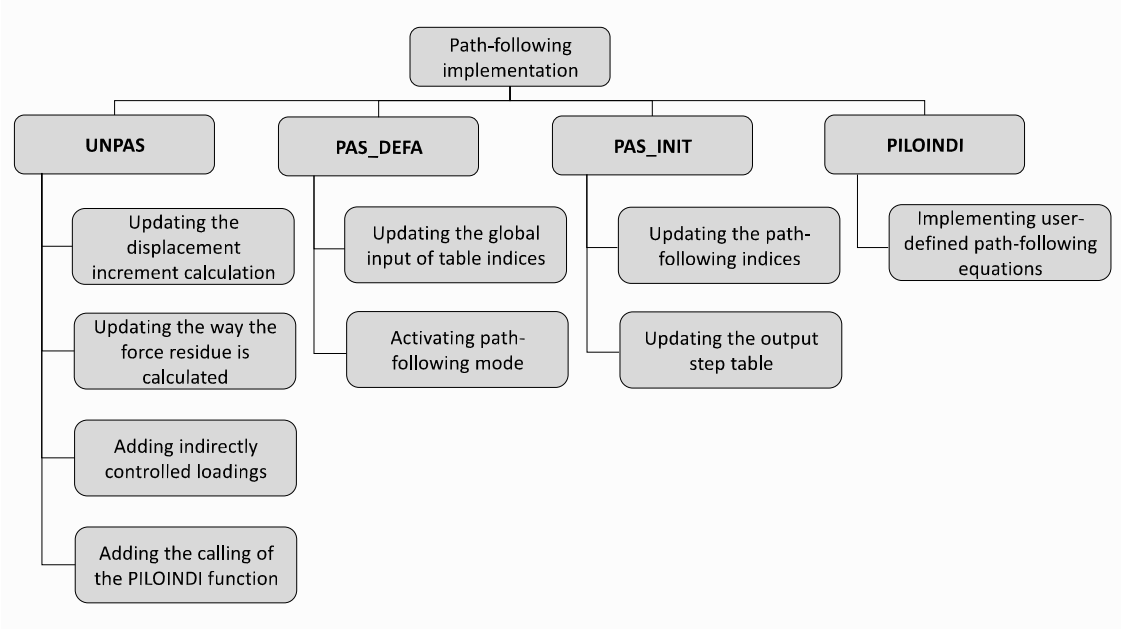

Figure 2: Schematic description of the main modifications implemented in Cast3M's nonlinear incremental solver.

$$\Delta\boldsymbol{\epsilon}_{\alpha}^{k+1} = \mathbf{q}_{\alpha}^{\top}\mathbf{B}_{\alpha}(\Delta\mathbf{d}^{k} + \delta\mathbf{d}_{\mathrm{I}}^{k+1} + \delta\eta^{k+1}\delta\mathbf{d}_{\mathrm{II}}^{k+1}) \tag{29}$$

As a consequence, the constraint equation that needs to be solved can be rewritten as:

$$\max_{\alpha\in\Omega^{h}} \left(a_{0,\alpha} + a_{1,\alpha}\delta\eta^{k+1}\right) - \Delta\tau = 0 \tag{30}$$

where the coefficients $a_{0,\alpha}$ and $a_{1,\alpha}$ read:

$$a_{0,\alpha} = \mathbf{q}_{\alpha}^{\top}\mathbf{B}_{\alpha}(\Delta\mathbf{d}^{k} + \delta\mathbf{d}_{\mathrm{I}}^{k+1}) \tag{31}$$

$$a_{1,\alpha} = \mathbf{q}_{\alpha}^{\top}\mathbf{B}_{\alpha}\delta\mathbf{d}_{\mathrm{II}}^{k+1} \tag{32}$$

Element-wise, function (30) is convex and linear. However, a direct determination of $\delta\eta^{k+1}$ is not possible since the maximum function is not differentiable. Instead, an iterative minimization algorithm is used.

Following Lorentz and Badel (2004), the method is chosen for determining the value of $\delta\eta^{k+1}$ is the nested interval method. Accordingly, the constraint equation is solved at each integration point in order to compute the load multiplier corrections $\delta\eta_{\alpha}^{k+1}$ such that:

$$a_{0,\alpha} + a_{1,\alpha}\delta\eta_{\alpha}^{k+1} - \Delta\tau = 0 \tag{33}$$

Then, a single $\delta\eta^{k+1}$ value is selected. More precisely, the sequence of tasks can be summarized as follows:

1. Initialize the interval $I = [-s, +s]$, where s is a very large value;
2. Loop over the integration points $\alpha \in \Omega^h$;
   (a) Calculate $\delta\eta_\alpha^{k+1}$ by solving eq. (33)
   (b) Update the interval I:
      i. if $a_{1,\alpha} > 0 \rightarrow I = I \cap [-s, \delta\eta_\alpha^{k+1}]$
      ii. if $a_{1,\alpha} < 0 \rightarrow I = I \cap [\delta\eta_\alpha^{k+1}, +s]$
3. Compute the solution:

$$\delta\eta^{k+1} \in \{\eta_{\min}, \eta_{\max}\} \tag{34}$$

Of the two values $\delta\eta_{\min}$ and $\delta\eta_{\max}$, the correct $\delta\eta$ value will be the one corresponding to the smallest force residue, or, alternatively, to the smallest norm $||\delta\mathbf{d}^{k+1}||$.

3.3. Building a data file

Using the path-following solver with the standalone implementation is easy and only requires minor modifications to existing Cast3M data files.[3]

In particular, the main steps for building the simulation file can be summarized as follows:

(i) The definition of the simulation always starts with the definition of the geometry of the problem, the definition of the material model, and the force/displacement boundary conditions. At this stage, only the known (imposed) contributions to the boundary conditions are prescribed.

(ii) Some initializations are then necessary for the path-following solver. In particular, one has to prescribe the domain's boundary over which the external load (displacement or force) is indirectly controlled. The direction of this loading is also given. In the example of listing 1 below, an indirectly-controlled horizontal displacement (component `UX`) is applied to the boundary labeled `BOUNDIND`. This information is stored in the object `DISPIND` and added to the object `BCTOT` together with information concerning other Dirichlet boundary conditions. Finally, the unity nodal field `DHAT` (representing $\hat{\mathbf{d}}$) for the solution process is defined.

(iii) Once `DHAT` has been defined, the variation of the path step length as a function of time is prescribed. The simplest solution is to consider $\Delta\tau$ a constant. However, the implementation proposed here allows it to vary as the simulation progresses, depending on the user's considerations (listing 2). For instance, one could imagine changing the path step length during the simulation in order to follow the evolution of the damage. In the simple example illustrated here, constant evolution is stored in the object called `EVOLTAU`. Notice that, in the code, the evolution of τ is given as an input, and that $\Delta\tau$ is computed at each time-step as $\tau - \tau_n$.

[3] The full syntax of the new procedure is available here: http://www-cast3m.cea.fr/index.php?page=noticesnotice=PILOINDI.

PDF-XChange Viewer Click to buy NOW! www.docu-track.com

Listing 1: Syntax for using the new PILOINDI procedure (definition of the indirectly-controlled displacement).

```
*****************************************************
* Pre-processing

    * Geometry
    * Model and materials
    * Boundary conditions (imposed contributions)
    ...
*****************************************************
* Initialization needed for using the
* path-following solver

    * Direction of the indirectly-controlled
    * displacement on the boundary labeled BOUNDIND
    * ('UX','UY','UZ') stand for the displacements
    * along the (X,Y,Z) directions.
    DISPIND = 'BLOQ' 'UX' BOUNDIND;

    * Add this indirectly-controlled boundary
    * condition to other Dirichlet boundary
    * conditions CLT
    BCTOT = BCTOT 'ET' DISPIND;

    * Create the unity field defining the direction
    * of the indirectly-controlled displacement
    * applied to LATD
    DHAT = 'DEPI' CL3 1.0 ;
*****************************************************
```

(iv) The data structures for running the nonlinear solving process are then created (listing 3). This information is stored in the object called `TAB1`. The user then provides, in addition to standard commands used with the incremental PASAPAS solver, three additional pieces of information to: activate the path-following solver, know if one indirectly controls displacements or forces, and give the evolution of τ. An optional fourth piece of information can also be provided regarding the definition of the part of the mesh on which the maximum of the constraint equation must be sought. If this last piece of information isn't provided, the maximization is done over the whole finite element mesh.

(v) Finally, one can run the PASAPAS solver and proceed with the post-processing of the obtained results, as usual with Cast3M.

Listing 2: Syntax for using the new PILOINDI procedure (evolution of the time-step length parameter).

```
*****************************************************
    * Maximum number of pseudo-time steps
    NS=100;

    * Definition of the path step length
    * Reference value for tau
    TAU = 7.0E-5;

    * Create a variation as a function of time for
    * the path step length parameter
    LIS1 = PROG 0. PAS TAU  (NS * TAU) ;
    EVOLTAU = EVOL MANU 't' LIS1 'tau' LIS1 ;
*****************************************************
```

Listing 3: Syntax for using the new PILOINDI procedure (activation of the path-following solver).

```
*****************************************************
* Standard definition of the data structure needed
* for using the nonlinear incremental solver
* PASAPAS
    TAB1 = TABLE;
    TAB1 . mova = ...;
    ...
*****************************************************
* Lines to include for the activation of the
* path-following solver

    * activate the path-following solver
    * VRAI = TRUE, FAUX = FALSE
    TAB1 . 'PILOTAGE_INDIRECT' = VRAI ;

    * indirectly-controlled force or displacement
    * use uch for displacement and fch for force
    TAB1 . 'DEPLACEMENTS_PILOTES' = DHAT ;

    * pseudo-time evolution of the path step
    * length parameter
    TAB1 . 'PARAMETRE_DE_PILOTAGE' = EVOLTAU;
*****************************************************
* Run the simulation using the nonlinear solver
* PASAPAS with path-following methods
    PASAPAS TAB1;
*****************************************************
* Post-processing
    ...
*****************************************************
```

PDF-XChange Viewer Click to buy NOW! www.docu-track.com

4. User-defined constraint equations

The CMSI formulation can be used without needing to perform any additional developments. However, user-defined constraint equations can be added in a standalone manner by overloading the PILOINDI procedure in the simulation's script (i.e., without touching the source code). The general syntax used to write new constraint equations is presented in Listing 4.

Two possible user-defined constraints are implemented to showcase this feature:

(i) The first constraint equation, the so-called Control by Nodal Displacement Increment (CNDI), is fully general (De Borst, 1987) and can be used with any material model. It is particularly interesting for simulating experimental loading conditions; for instance, when a combination of displacement measures is used to drive the variation of the external load (see, e.g., Rastiello et al. (2014, 2015)).

(ii) The second constraint equation is more model-dependent since it relies on Controlling the Maximum Elastic Predictor (CMEP) of the damage/yield function (Lorentz and Badel, 2004).

For details concerning their implementation, the interested reader is referred to example files made available on the Cast3M website (http://www-cast3m.cea.fr).[4]

4.1. Control by nodal displacement increment (CNDI)

The CNDI criterion specifies that the increment of the linear combination of the displacements computed at a given set of nodes equals $\Delta\tau$.

From De Borst (1987), the path-following constraint equation is written as:

$$p^{k+1} = \mathbf{p}^{\top}\Delta\mathbf{d}^{k+1} - \Delta\tau = 0 \tag{35}$$

where $\mathbf{p}$ is a selection vector containing the multiplication factors for the corresponding DOFs, and:

$$\Delta\mathbf{d}^{k+1} = \Delta\mathbf{d}^{k} + \delta\mathbf{d}_{\mathrm{I}}^{k+1} + \delta\eta^{k+1}\delta\mathbf{d}_{\mathrm{II}}^{k+1} \tag{36}$$

Since eq. (35) is linear in $\delta\mathbf{d}^{k+1}$, the value of $\delta\eta^{k+1}$ can be expressed as:

$$\delta\eta^{k+1} = \frac{\Delta\tau - a_0}{a_1} \tag{37}$$

[4] Files: pilotage_indirect_1.dgibi and pilotage_indirect_2.dgibi.

Listing 4: Syntax for adding user-defined constraint equations at the beginning of a Cast3M input file without changing the source code. Note (according to Cast3M keywords): 'CHPOINT' stands for point-wise field (nodal field in this case), 'FLOTTANT' stands for floating number.

```
* USER-DEFINED PROCEDURE ***************************
******************************************************
* Beginning of the user procedure
* Input:
*   Info on geometry, materials, models (PRECED)
*   Displacement at previous time step (DEPT)
*   Displ. var. at iteration k (DU)
*   Displ. increment I at iteration k+1 (DUI)
*   Displ. increment II at iteration k+1 (DUII)
*   Path step length (DTAU)
'DEBPROC' PILOINDI
    PRECED      *'TABLE'
    U           *'CHPOINT'
    DU          *'CHPOINT'
    DUI         *'CHPOINT'
    DUII        *'CHPOINT'
    DTAU        *'FLOTTANT';
******************************************************
* Solution of the user-defined path-following
* constraint equation
...
...
******************************************************
* End of the user procedure
* Output: load multiplier variation (D_ETA)
'FINPROC' D_ETA ;
******************************************************
```

where coefficients a_0 and a_1 are defined as:

$$a_0 = \mathbf{p}^\top (\delta \mathbf{d}_{\mathrm{I}}^{k+1} + \Delta \mathbf{d}^k) \tag{38}$$

$$a_1 = \mathbf{p}^\top \delta \mathbf{d}_{\mathrm{II}}^{k+1} \tag{39}$$

Using Equation (37), it is possible to obtain the value of $\delta\eta^{k+1}$ both in the prediction and correction phase. For the prediction phase ($k+1=1$), eq. (37) can be used with $\Delta\mathbf{d}^k = \Delta\mathbf{d}^0 = \mathbf{0}$. During computations, $\delta\mathbf{d}_{\mathrm{I}}^{k+1}$ becomes smaller with each passing iteration (it is calculated from the unbalanced forces at the k-th iteration), whereas $\Delta\mathbf{d}^k$ tends to stay stationary. Since the quantity $\delta\mathbf{d}^{k+1}$ is always different from zero, this implies that the value of $\delta\eta^{k+1}$ should get ever closer to zero. This means that $\mathbf{p}^\top \Delta\mathbf{d}^{k+1}$ gets, in turn, ever closer to $\Delta\tau$, thus satisfying eq. (35).

4.2. Controlling the maximum elastic predictor (CMEP)

The effectiveness of the CNDI technique depends on whether the chosen combination of DOFs increases monotonically throughout the simulation. Unfortunately, finding a set of DOFs with this characteristic is not an easy task for general structures. That's why the CMSI formulation is chosen as default.

Although the CMSI approach can be applied to various situations, some studies indicate that it doesn't distinguish between dissipative and non-dissipative (i.e., elastic unloading) solutions.

Path-following equations based on the maximization of the elastic predictor of the damage/plasticity function (i.e., the value of the damage/yield criterion function when considering an incrementally elastic response) (Lorentz and Badel, 2004) can overcome this limitation, though this is achieved at the cost of a more intrusive and model-dependent implementation.

4.2.1. Formulation

A path-following CMEP of the damage criterion function turns out to be well suited to capture localized dissipative phenomena resulting from strain localization.

Dissipative model. As an example, we may assume that the material behaves according to a rate-independent constitutive model with a single internal variable κ. Its evolution satisfies the Karush-Kuhn-Tucker and consistency conditions:

$$f \leq 0 \quad \dot{\kappa} \geq 0 \quad \dot{\kappa} f = 0 \quad \dot{\kappa} \dot{f} = 0 \tag{40}$$

where $f = f(Y, \kappa)$ is a damage (plastic) criterion function and $Y = Y(\boldsymbol{\epsilon}, \kappa)$ is the thermodynamic force associated with κ.

Constraint equation function. According to Lorentz and Badel (2004), the constraint equation is expressed as in eq. (26), with:

$$g_\alpha^{k+1} = f_\alpha^{\text{elas},k+1} \tag{41}$$

where the elastic predictor of the damage/plasticity criterion function is expressed as:

$$f_g^{\text{elas},k+1} = f(Y_\alpha(\boldsymbol{\epsilon}_\alpha^{k+1}, \kappa_n), \kappa_n) \tag{42}$$

with κ_n standing for the internal variable at pseudo-time step t_n. Moreover:

$$\epsilon_\alpha^{k+1} = \mathbf{B}_\alpha(\mathbf{d}_n + \Delta\mathbf{d}^k + \delta\mathbf{d}_{\text{I}}^{k+1} + \delta\eta^{k+1}\delta\mathbf{d}_{\text{II}}^{k+1}) \tag{43}$$

As shown by Lorentz and Badel (2004), a positive value for the elastic predictor is associated with a dissipative branch of the equilibrium path. For more details on the properties of the CMEP method, the interested reader can turn to said paper.

Resolution. Since f_α is commonly chosen to be a convex function of $\boldsymbol{\epsilon}_\alpha$, f_α^{elas} is also convex. Furthermore, given the convexity of the "max" function, the path-following constraint (41) preserves this same property. As a consequence, it may admit up to two distinct real or complex (inadmissible) roots. Their direct determination is, however, often impossible because the f_α^{elas} function is generally nonlinear, and the "max" operator represents an additional source of non-linearity.

As mentioned in previous sections, a possible solution strategy may be to use a "nested interval algorithm". For this purpose, the elastic predictor function is linearized based on a known value of $\delta\eta$. The most straightforward choice consists in using an explicit solution procedure to avoid further local sub-iterations.

In that case, the f_α^{elas} function is linearized based on $\delta\eta^{k+1} = 0$ (i.e., about η^k) as:

$$f_\alpha^{\text{elas},k+1} = \left(f_\alpha^{\text{elas}}\right)_{\delta\eta=0} + \left(\frac{\partial f_\alpha^{\text{elas}}}{\partial(\delta\eta)}\right)_{\delta\eta=0} \delta\eta^{k+1} \tag{44}$$

where:

$$\left(f_\alpha^{\text{elas}}\right)_{\delta\eta=0} = f(Y_\alpha(\mathbf{B}_\alpha(\mathbf{d}_n + \Delta\mathbf{d}^k + \delta\mathbf{d}_{\text{I}}^{k+1}), \kappa_n), \kappa_n) \tag{45}$$

The constraint equation thus reads once again like (30), where coefficients $a_{0,\alpha}$ and $a_{1,\alpha}$ can be easily identified after computing $\partial f_\alpha^{\text{elas}}/\partial(\delta\eta)$.

An example. We may consider a simple isotropic continuum damage model where the damage criterion function can be expressed as:

$$f = \tilde{\epsilon} - \kappa \tag{46}$$

where $\tilde{\epsilon}$ is an equivalent strain measure.

Here, we consider the so-called Mazars (1984) equivalent strain, $\tilde{\varepsilon} = \sqrt{\langle\boldsymbol{\epsilon}_p\rangle \cdot \langle\boldsymbol{\epsilon}_p\rangle}$, with $\langle\boldsymbol{\epsilon}_p\rangle$ the positive part of the principal strain tensor (represented in vector format), and κ is the historical maximum of the equivalent strain.

The idea is to ensure that the elastic predictor function is equal to $\Delta\tau > 0$ on at least one integration point $\alpha \in \Omega^h$ for each time step. The integration point α that serves as the basis for calculating the criterion is chosen automatically during iterations. To achieve this, the elastic predictor of the damage criterion function is defined for each integration point α:

$$f_\alpha^{\text{elas},k+1} = \tilde{\epsilon}_\alpha^{k+1} - \kappa_{\alpha,n} \tag{47}$$

where $\kappa_{\alpha,n}$ is the maximum equivalent strain experienced up to the beginning of the current time step.

The $a_{1,\alpha}$ and $a_{0,\alpha}$ factors can be computed by linearizing eq. (47) with respect to the unknown variable $\delta\eta$, from which:

$$\begin{aligned} a_{0,\alpha} &= \left(f_\alpha^{\text{elas}}\right)_{\delta\eta=0} \\ &= \Big[\mathbf{G}_{\alpha,\text{I}}^{k+1}\mathbf{R}_{\alpha,\text{I}}^{k+1}\mathbf{B}_\alpha(\mathbf{d}^k+\delta\mathbf{d}_\text{I}^{k+1}) \\ &\qquad \cdot\mathbf{G}_{\alpha,\text{I}}^{k+1}\mathbf{R}_{\alpha,\text{I}}^{k+1}\mathbf{B}_\alpha(\mathbf{d}^k+\delta\mathbf{d}_\text{I}^{k+1})\Big]^{1/2} - \kappa_{\alpha,n} \end{aligned} \tag{48}$$

and

$$\begin{aligned} a_{1,\alpha} &= \left(\frac{\partial f_\alpha^{\text{elas}}}{\partial(\delta\eta)}\right)_{\delta\eta=0} = \left(\frac{\partial\tilde{\varepsilon}_\alpha}{(\partial\delta\eta)}\right)_{\delta\eta=0} \\ &= \left(\left(\frac{\partial\tilde{\varepsilon}_\alpha}{\partial\left\langle\boldsymbol{\epsilon}_{\alpha,p}\right\rangle}\right)^\top \frac{\partial\left\langle\boldsymbol{\epsilon}_{\alpha,p}\right\rangle}{\partial\boldsymbol{\epsilon}_{\alpha,p}}\frac{\partial\boldsymbol{\epsilon}_{\alpha,p}}{\partial\boldsymbol{\epsilon}_\alpha}\frac{\partial\boldsymbol{\epsilon}_\alpha}{\partial(\delta\eta)}\right)_{\delta\eta=0} \\ &= \frac{\left\langle\boldsymbol{\epsilon}_{\alpha,p}^k\right\rangle^\top}{\tilde{\varepsilon}_\alpha^k}\mathbf{G}_\alpha^k\mathbf{R}_{\alpha,\text{II}}^{k+1}\mathbf{B}_\alpha\delta\mathbf{d}_\text{II}^{k+1} \end{aligned} \tag{49}$$

where $\mathbf{G}_{\alpha,\text{I}}^{k+1}$ is a diagonal matrix such that $\mathbf{G}_{\alpha,\text{I},ii}^{k+1} = 1$ if $(\mathbf{R}_{\alpha,\text{I}}^{k+1}\mathbf{B}_\alpha(\mathbf{d}^k+\delta\mathbf{d}_\text{I}^{k+1})_i$ is positive and $\mathbf{G}_{\alpha,\text{I},ii}^{k+1} = 0$ otherwise; $\mathbf{G}_\alpha^k$ is such that $G_{\alpha,ii}^k = 1$ if $\epsilon_{\alpha,p,i}^k$ is positive and $G_{\alpha,ii}^k = 0$ otherwise; and $\mathbf{R}_{\alpha,\text{I}}^{k+1}$ and $\mathbf{R}_{\alpha,\text{II}}^{k+1}$ are the matrices that rotate the strains $\mathbf{B}_\alpha(\mathbf{d}^k+\delta\mathbf{d}_\text{I}^{k+1})$ and $\mathbf{B}_\alpha\delta\mathbf{d}_\text{II}^{k+1}$ into their respective principal basis.[5]

[5] Alternatively, one could have replaced $\mathbf{R}_{\alpha,\text{I}}^{k+1}$ and $\mathbf{R}_{\alpha,\text{II}}^{k+1}$ by $\mathbf{R}_\alpha^k$, the rotation matrix of the strain vector at iteration k into its principal basis.

PDF-XChange Viewer Click to buy NOW! www.docu-track.com

5. Performance assessment

The performance of the implementation proposed herein is assessed through a simple three-point bending test for which a stable global response is obtained. This allows for a direct comparison between the standard iterative Newton solver and the path-following solver (with different constraint equations) in terms of CPU times and number of iterations needed to converge.

All performed simulations use the elastic stiffness operator to compute $\mathbf{K}^k$ in conjunction with convergence acceleration (default choice in Cast3M). Of course, the use of other stiffness tensors could affect the convergence properties of the proposed methods. Notice, however, that no additional constraints are created by the studied path-following methods on the iterative strategy and on the choice of this operator.

5.1. *Constitutive assumptions and damage regularization*

In this example, as well as in the examples illustrated in the next section, the regions that must remain intact throughout the analysis are modeled using an isotropic linear elastic constitutive model. The isotropic continuum damage mechanics model from Mazars (1984) is used for areas susceptible to damage.[6]

It is well known that softening material models lead to mesh-dependent results when used with a finite element method. In such cases, the structural dissipation becomes dependent on the mesh size and may eventually become smaller as the mesh size decreases. For these models, regularization methods (see, among others, Hillerborg et al. (1976), Peerlings et al. (1998), Giry et al. (2011), Rastiello et al. (2018)) are generally adopted in order to generate mesh-independent results. In this study, no regularization techniques are used, both for the sake of simplicity and in order to induce sharp snap-backs (see next section). However, the path-following strategies presented herein can be used with regularized models without significant modifications.

5.2. *Three-point bending test*

5.2.1. *Problem setting*

Here, a notched beam submitted to a three-point bending test is considered. The geometry and boundary conditions are defined in fig. 3. The elements between the notch and the upper horizontal boundary are supposed damageable, whereas a linear elastic material behavior is assumed for the other elements. The material parameters used in these computations (table 1) have been selected to generate a stable response (i.e., without snap-back) in the post-load peak phase.

[6] This constitutive law was selected because it is a popular choice for modeling damage of quasi-brittle materials. Other constitutive equations may, of course, be considered.

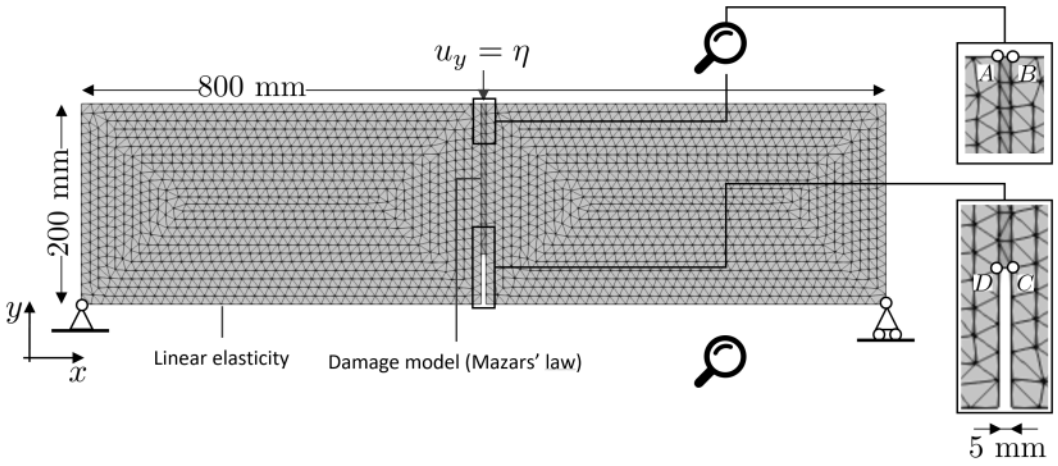

Figure 3: Three-point bending test – geometry and boundary conditions.

A reference simulation is performed by applying a vertical displacement (Direct Dirichlet Condition, DDC) to the bearing surface (nodes A and B in fig. 3), and increasing its value. The response obtained in this manner is then compared with the responses obtained using the path-following strategies.

5.2.2. *Constraint equations*

The path-following solver is set up so as to replicate the global responses of the DDC simulation. In particular, the CNDI and CMSI formulations are used as follows:

CNDI-top. The CNDI algorithm is used to replicate the directly-imposed displacement boundary condition. The constraint equation controls the vertical displacement of nodes A and B (fig. 3):

$$p^{k+1} = -\frac{\Delta d_{y,A}^{k+1} + \Delta d_{y,B}^{k+1}}{2} - \Delta\tau = 0 \tag{50}$$

where $d_{y,\bullet}$ ($\bullet = A, B$) is the vertical component of vector $\mathbf{d}$ at node $\bullet$ and $\Delta\tau > 0$.

CNDI-cmod. The CNDI algorithm is then used to control the Crack Mouth Opening displacement at the notch, i.e.:

$$p^{k+1} = \Delta d_{x,C}^{k+1} - \Delta d_{x,D}^{k+1} - \Delta\tau = 0 \tag{51}$$

where $d_{x,\bullet}$ ($\bullet = C, D$) is the horizontal component of $\mathbf{d}$ at node $\bullet$. In order to ensure a proper comparison, the simulation is performed with exactly the same number of loading steps as the reference simulation. Moreover, the $\Delta\tau$ parameter is not supposed constant but is defined by the variations of $\Delta d_{x,C}^{k+1} - \Delta d_{x,D}^{k+1}$ recorded during the DDC simulation (fig. 4).

PDF-XChange Viewer Click to buy NOW! www.docu-track.com

Table 1: Material constants of Mazars' model. Numerical values selected for simulating a three-point bending test without snap-backs.

Parameter	Description	Value
ν	Poisson's ratio	0.15
E	Young modulus	30 GPa
ρ	Density	2300 kg/m^3
ε_0	Damage initiation threshold	1.0×10^{-5}
A_t	Mazars' model input	0.2
A_c	Mazars' model input	1.2
B_t	Mazars' model input	0.8×10^3
B_c	Mazars' model input	1.5×10^3
β	Mazars' model input	1.06

CMSI. Finally, the CMSI algorithm is applied. Once again, the $\Delta\tau$ parameter varies with $\max_{\Omega^h}(\mathbf{q}_\alpha^\top \Delta\boldsymbol{\epsilon}_\alpha^{k+1})$, which was recorded during the standard Newton solver simulation. The number of loading steps used is, again, exactly the same as in the DDC simulation.

As shown in fig. 4, these choices yield the same global force-displacement response as the one obtained with the DDC simulation.

5.2.3. *Performance assessment*

The comparison between Newton and path-following solvers is performed in terms of the total time spent on the iterative solving process (t_{it}) and the time spent on solving the constraint equation (t_{pf}). Time t_{it} includes any time spent, over the iterations, on assembling the matrices, imposing boundary conditions, solving the linear systems and updating the solution. Accordingly, the total time spent running the simulation is $t_{\text{tot}} = t_{\text{it}} + t_{\text{pf}}$.

For a proper comparison, each simulation is repeated 5 times in order to average the values of t_{it} and t_{pf}. As shown in fig. 5:

- t_{pf} is always very small compared to t_{it}. As expected, t_{pf} is larger for the CMSI method than for the CNDI formulations because of the nested interval algorithm. This CPU time can be reduced by limiting the resolution of the constraint equation exclusively to the part of the mesh likely to experience damage.
- t_{tot} is very similar for all direct/indirect ways of imposing Dirichlet boundary conditions. The time spent solving the equilibrium problem is quite stable despite the fact that with path-following solvers, each global iteration requires two computations of the linear system. This can easily be explained in light of the results

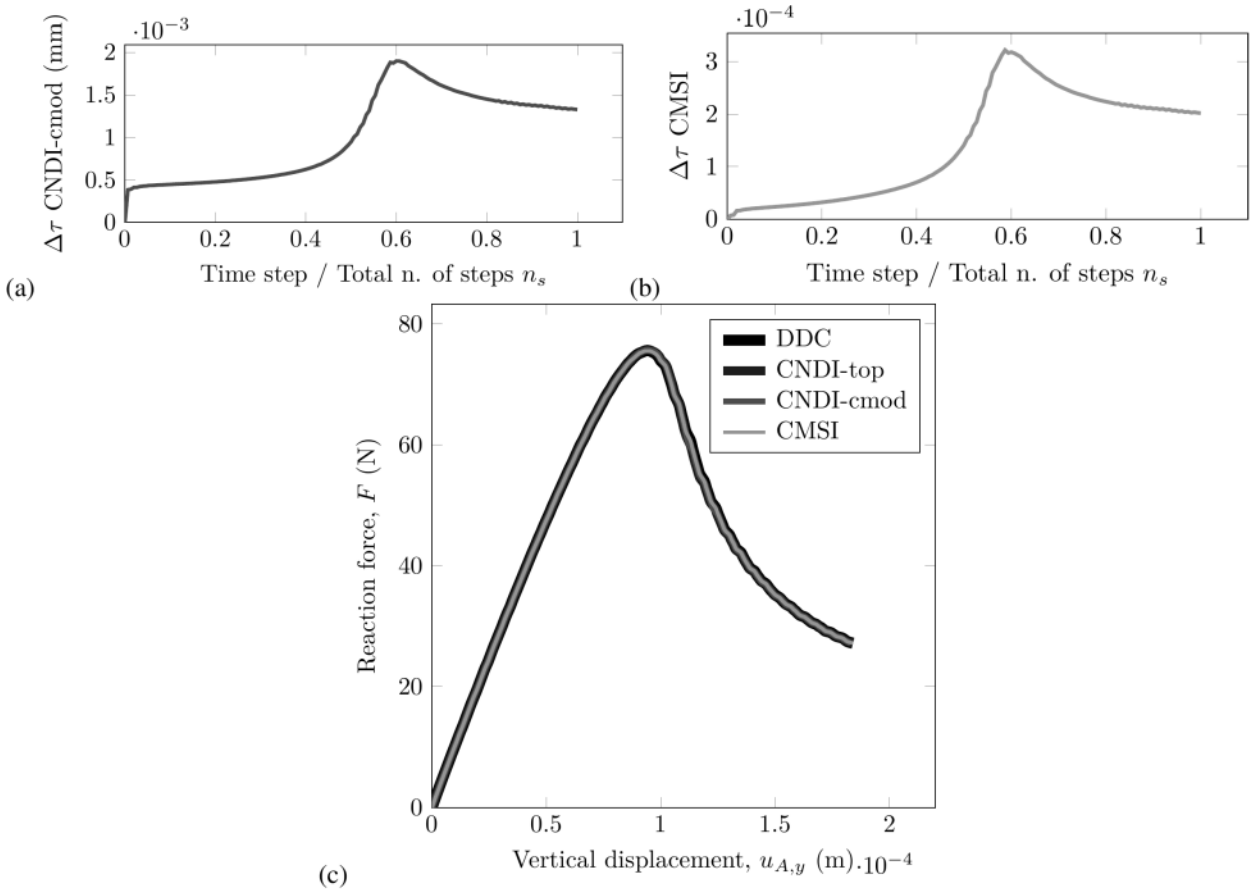

Figure 4: Three-point bending test – (a) Prescribed variation of the path step length in the CNDI-cmod computation as a function of time, (b) Prescribed variation of the path step length in the CMSI computation as a function of time, and (c) Global responses.

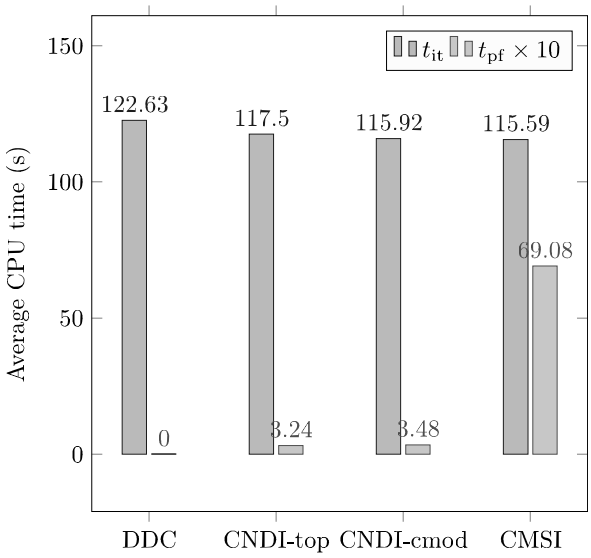

Figure 5: Three-point bending test – Average CPU times (a scale factor of 10 is applied to t_{pf} for clarity).

depicted in fig. 6, which represents the evolution of the number of iterations per pseudo-time step. These curves clearly show that the number of iterations required by the path-following solvers is generally half less than what is required by the Newton solver to achieve convergence. In other words, convergence is accelerated. Of course, more studies will be needed to better explain these results, but the trend is clear for all the constraint equations considered.

PDF-XChange Viewer
Click to buy NOW!
www.docu-track.com

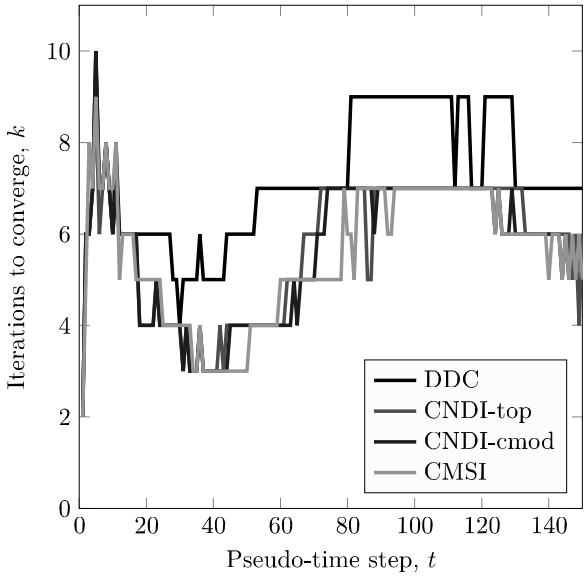

Figure 6: Three-point bending test – Number of iterations required for convergence.

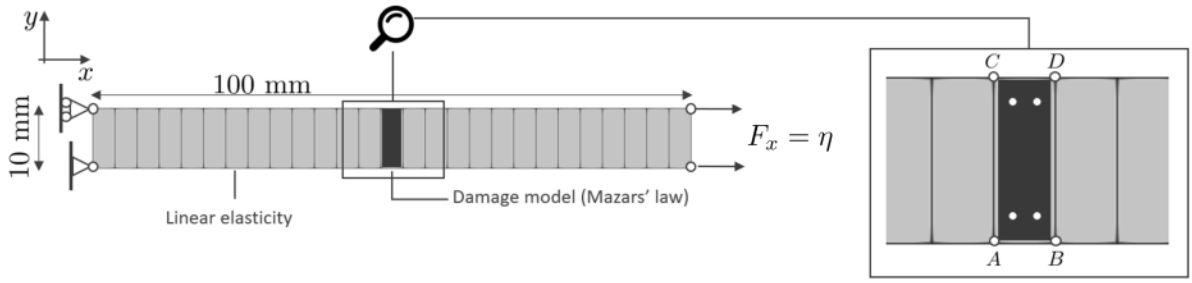

Figure 7: Beam under tension – geometry and boundary conditions.

6. Numerical examples with sharp snap-back responses

Three test cases are considered in order to show the convergence and stability of the proposed numerical framework. The problems considered are 2D and 3D, but the proposed framework also works in a one-dimensional (1D) context. However, this case is not considered here as it is pretty trivial compared with 2D and 3D simulations.

6.1. 2D beam under tensile loading (indirectly-controlled force)

The first test case simulates a tensile load on a 2D beam. This test was chosen because it is often used to study mesh sensitivity when using strain-softening constitutive models (see, e.g., Giry et al. (2011), Rastiello et al. (2018)).

6.1.1. Problem setting

The structure consists of two elastic regions connected by an intermediate damageable zone where strain localization occurs. The geometry and boundary conditions are illustrated in fig. 7. The beam's left side is constrained, while loading is applied under indirect force control by pulling on the beam's right side. Plane stress conditions are assumed.

The finite element discretization Ω^h is obtained using linear quadrilateral finite elements (QUA4). Several mesh refinement levels are considered to study the algorithm's capability to capture pronounced snap-back responses as the damaging band gets thinner. The number of finite elements along the longitudinal direction is denoted n. Its values range from 3 to 29 in order to obtain structural responses that are increasingly unstable. Alternatively, one could have chosen a fixed size for the damaging element and increased the beam's length progressively.

The material parameters used in the computations are given in Table 2. The same parameters will be used for all the examples described in this section.

6.1.2. Discussion

An equilibrium path is obtained for each criterion proposed previously. In particular, the CNDI method is applied to control the relative horizontal displacement of nodes B and A, i.e.:

Table 2: Material constants of Mazars' model. Numerical values selected for the numerical examples.

Parameter	Description	Value
ν	Poisson's ratio	0
E	Young modulus	1 GPa
ρ	Density	2300 kg/m^3
ε_0	Damage initiation threshold	1.0×10^{-4}
A_t	Mazars' model input	1.0
A_c	Mazars' model input	1.2
B_t	Mazars' model input	1.0×10^4
B_c	Mazars' model input	1.5×10^3
β	Mazars' model input	1.06

$$p^{k+1} = \Delta d_{x,B}^{k+1} - \Delta d_{x,A}^{k+1} - \Delta\tau = 0 \tag{52}$$

where $d_{x,\bullet}$ ($\bullet = A, B$) is the horizontal component of the displacement vector $\mathbf{d}$ at node $\bullet$ and $\Delta\tau > 0$. Moreover, the CMSI and CMEP methods were applied by restricting the constraint equation to the set of finite elements where damage is expected to vary.

The simplicity of the geometry and the boundary conditions ensures that the three path-following criteria considered provide precisely the same force-displacement responses. However, the CMEP formulation presents a significant difference with the other criteria: it is possible to reach the peak of the local stress-strain law during the first load increment. This feature is a clear advantage if the focus of the analysis is the post-peak trajectory, because it reduces the calculation time.

Numerical results show that by successively increasing n from 3 to 29, the structural response becomes increasingly unstable (fig. 8a). For $n = 3$ the area where the damage occurs is large enough to dissipate energy in a controlled manner, so no snap-back is observed. However, when n increases, this region shrinks and loses its ability to dissipate energy in a controlled way. This means that once the damage starts, new equilibrium solutions may only be found at the cost of a simultaneous reduction in load and displacements. The severity of the snap-back reflects the intensity of this effect.

Regarding the fulfillment of the path-following constraint equation, fig. 8b shows that the required value of $\Delta\tau$ is attained at every time step until the end of the analysis, regardless of the chosen path-following equation. This indicates that the proposed algorithm stays stable and convergent, even in the presence of severe instabilities. Moreover, the proposed implementation can capture the complete equilibrium path.

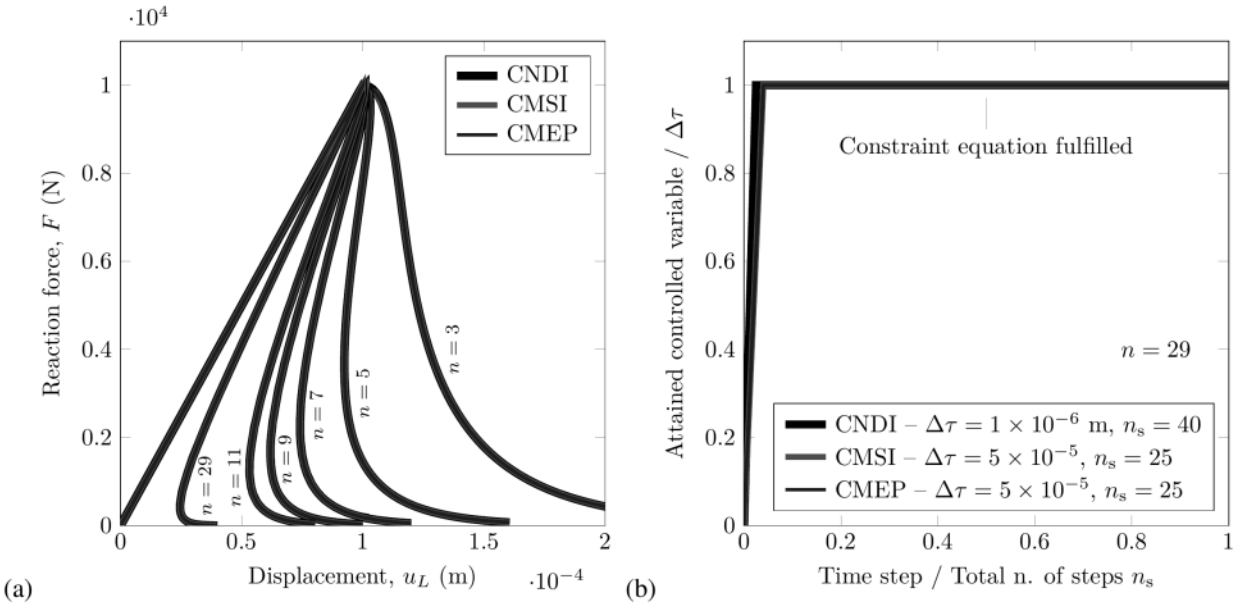

Figure 8: Beam under tension – Effect of damage localization on the global response. The occurrence of increasingly severe snap-backs reflects the intensity of the strain localization effects. The value of the constraint equation is checked throughout the analysis for each criterion used.

6.2. 2D notched beam under tensile loading (indirectly-controlled displacement)

The second 2D example is a traction test on a notched beam. This test case is quite close to the previous one. However, the presence of the two notches complexify the strain localization process since one may expect that damage will start at the notches and progressively propagate towards the center of the beam.

6.2.1. Problem setting

The geometry of the beam is given in fig. 9. The displacement is constrained at the left edge, whereas the load is applied at the right edge by imposing an indirectly-controlled displacement. Plane stress conditions are assumed.

A regular mesh Ω^h is used to discretize the computational domain for simplicity and to identify the localization zone clearly; more complex unstructured meshes can be used without significant modifications.

6.2.2. Selection of the controlled DOFs with the CNDI

In order to use the CNDI criterion, it is necessary to choose a set of DOFs to be controlled. This choice is generally made based on engineering or physical considerations regarding the expected damaging process. Alternatively, it can be made for the purpose of replicating experimental control techniques, e.g., when crack-opening displacements at given positions are controlled through Linear Variable Displacement Transducers (LVDTs).

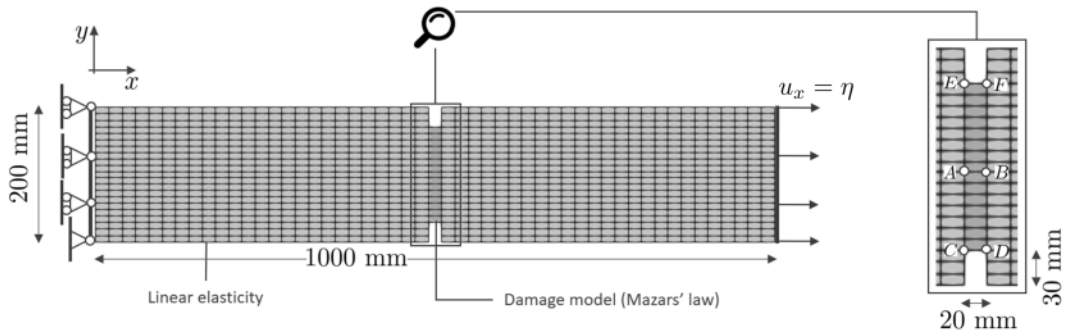

Figure 9: Tensile test on a notched beam – Geometry, boundary conditions and designation of the nodes considered for the application of the CNDI formulation.

Now, given the considered geometry and boundary conditions, the most relevant choice is to control the linear combination of the horizontal displacements of the nodes located on the vertical boundaries of each of the damage zones (Figure 9). In order to study the effects of such a choice, we considered:

CNDI-1. Two nodes (denoted *A* and *B*) located at mid-height of the beam. The variation of their relative displacement is controlled, i.e.:

$$p^{k+1} = \Delta d_{x,B}^{k+1} - \Delta d_{x,A}^{k+1} - \Delta\tau = 0 \tag{53}$$

CNDI-2. Two nodes located at the bottom notch. The Crack Mouth Opening Displacement (CMOD) variation at the bottom notch is controlled, i.e.:

$$p^{k+1} = \Delta d_{x,D}^{k+1} - \Delta d_{x,C}^{k+1} - \Delta\tau = 0 \tag{54}$$

CNDI-3. The mean value of the CMOD measures at the top and bottom notches, as is classically done in experiments, i.e.:

$$p^{k+1} = \frac{\Delta d_{x,D}^{k+1} - \Delta d_{x,C}^{k+1} + \Delta d_{x,F}^{k+1} - \Delta d_{x,E}^{k+1}}{2} - \Delta\tau = 0 \tag{55}$$

The results depicted in fig. 10 show that the choice of points on one notch exclusively (case CNDI-2) can lead to a loss of part of the equilibrium path. This occurs because damage originates at the notches, and because small numerical differences in CMOD are observed between the two notches during the simulation. Indeed, the CMOD at the top notch is found to be slightly larger than at the bottom one, which leads to the observed loss of part of the equilibrium path. However, in the subsequent phases of the simulation, the algorithm continues to follow the curve correctly. Moreover, this effect cannot be mitigated by controlling the displacement at both notches simultaneously using the average CMOD (case CNDI-3).

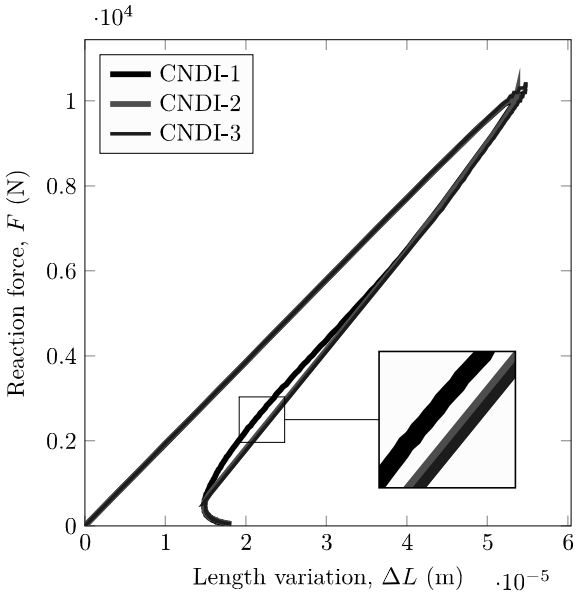

Figure 10: Tensile test on a notched bar – Comparison between the equilibrium paths obtained via CNDI for different base points.

On the other hand, no such limitation is found for a choice of points located in the middle of the ligament zone (case CNDI-1). This is because damage to the corresponding elements occurs late (at the end of the test). Over the course of the entire damage propagation process, these nodes experience a smaller relative displacement than the one experienced at notches. For this reason, the constraint eq. (53) is used for the comparison with the CMSI and CMEP formulations.

6.2.3. Discussion

As we already observed in the previous example, fig. 11a shows a good agreement between the force-displacement equilibrium curves computed using the three different path-following constraint equations. Moreover, for all of them, the values chosen for the variable $\Delta\tau$ are strictly respected throughout the simulations (fig. 11b).

Such a simultaneous reduction of force and displacement enables us to simulate a gradual damaging process (fig. 12). As expected, damage originates at both notches and progressively propagates towards the beam's center. As it progresses, the structure's effective resisting area becomes smaller and so does the amount of energy that can be dissipated in a controlled manner. After reaching the peak load, new equilibrium points can only be found if there is a simultaneous decrease in displacements and the external load.

Replicating such a gradual process would not have been possible without an indirect loading control method. One could have observed a sharp damage localization increase (from zero to one) on the beam's central part.

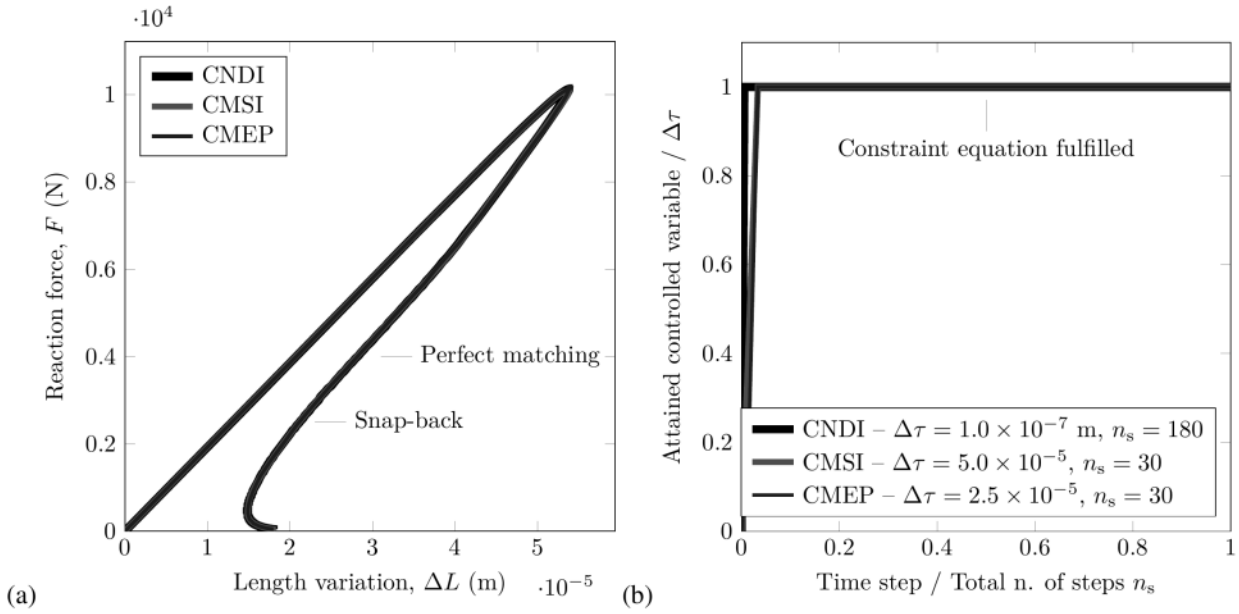

Figure 11: Tensile test on a notched bar – (a) Global force-displacement responses; (b) Fulfillment of the constraint equation.

6.3. 3D perforated plate under tensile loading (indirectly-controlled displacement)

The final simulation is a tensile test on an eccentrically-perforated plate. The center of the circular hole is located along the vertical symmetry axis of the plate but is distinct from the center of the plate. The two-dimensional version of this example can be found in the literature (see, e.g., Lorentz and Badel (2004), Rastiello et al. (2019)) and is typically simulated to test path-following algorithms in the case of structural responses characterized by multiple snap-back phases.

6.3.1. Problem setting

The 3D geometry of the plate is depicted in fig. 13. The analysis is carried out by applying indirectly-controlled displacements on the right boundary of the plate, whereas the left one is constrained (fig. 13a). The discretization of the computational domain is obtained through linear tetrahedral finite elements (fig. 13b).

6.3.2. Discussion

All three criteria illustrated in this study are tested. Following Rastiello et al. (2019), the CNDI formulation is used for controlling the average CMOD at the hole. Since two cracks are expected to propagate from the hole toward the upper and lower boundaries, it seems quite natural to select the controlled DOFs. Here, the constraint equation is written as follows:

$$p^{k+1} = \frac{\Delta d_{x,D}^{k+1} - \Delta d_{x,C}^{k+1} + \Delta d_{x,B}^{k+1} - \Delta d_{x,A}^{k+1}}{2} - \Delta\tau = 0 \tag{56}$$

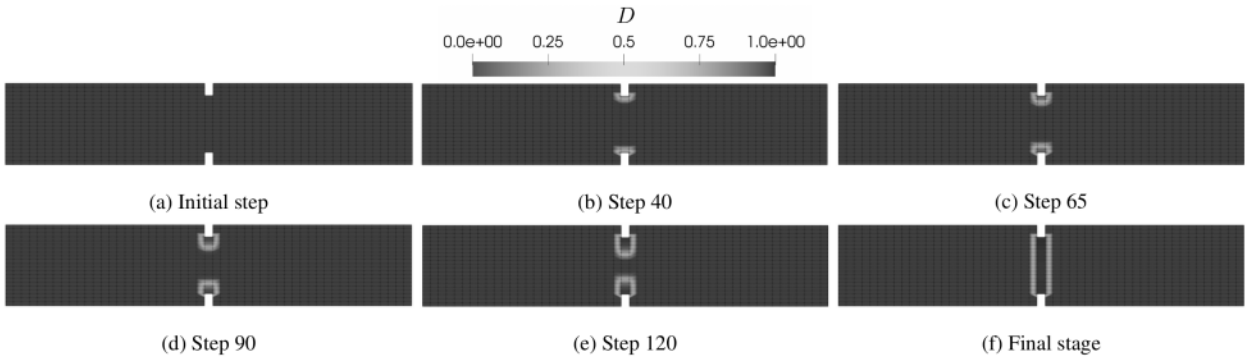

Figure 12: Tensile test on a notched bar – Damage progression during the simulation.

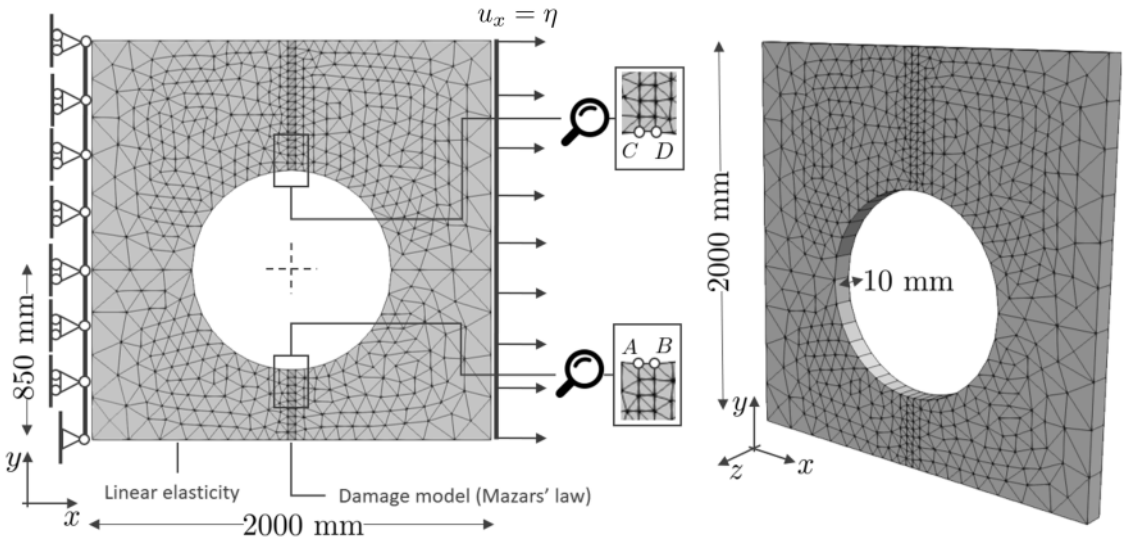

Figure 13: 3D perforated plate under tensile loading – Geometry and boundary conditions; Perspective view.

Numerical results (fig. 14a) show that the CNDI algorithm can predict the load peak and the existence of two snap-backs. However, it is unable to follow the equilibrium path smoothly. The reason for this is the choice of the reference points which guide the path-following equation. Conversely, the results obtained using the CMSI and CMEP formulations are in good agreement. In addition to predicting the peak load and the existence of the two snap-backs, they describe the entire equilibrium path all the way to the rupture. Remarkably, the proposed formulations provide solutions even in the presence of severe instabilities.

Figure 14b shows that the value of $\Delta\tau$ is attained with each criterion and for every analysis. However, convergence cannot be ensured when using the CNDI algorithm, in particular during the second snap-back phase. Notice that, as was already observed by Rastiello et al. (2019), the jump in the global response observed with the CNDI formulation (where the other methods instead predict the first snap-back) cannot be attributed to the non-fulfillment of the constraint equation. However, this is directly related to the controlled DOFs chosen. While the simulation ends prema-

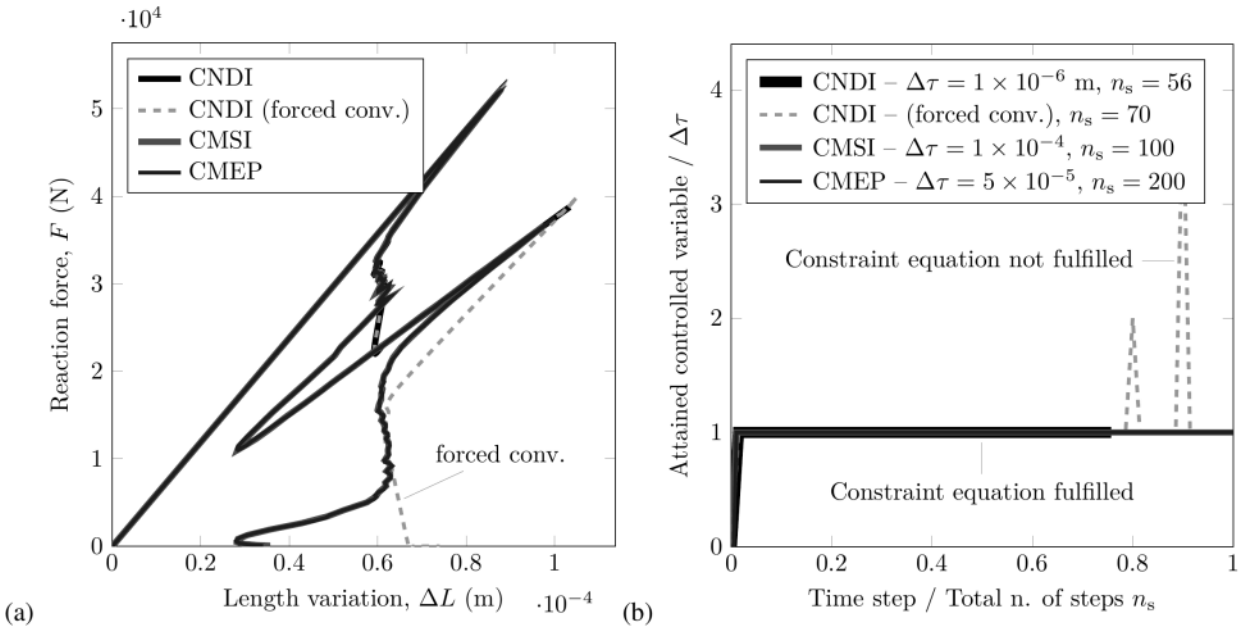

Figure 14: 3D holed plate under tensile loading – (a) Global force-displacement responses; (b) Fulfillment of the constraint equation.

turely with CNDI, this can be prevented by activating the "forced convergence" (a mixed implicit-explicit formulation in which the novel solution is predicted based on previous ones) feature in the Cast3M software (dashed green line). In that case, however, the constraint equation is not fulfilled a second time.

Observing the failure mechanism of this structure (fig. 15) is also quite interesting. In the linear phase, strain accumulates at the upper and lower edges of the circular hole until the maximum load is reached. Then, the local energy dissipation capacity is depleted and neighboring finite elements begin to be solicited. This process causes the strain localization to move. This first occurs in the plate's lower ligament zone and corresponds to the occurrence of the first snap-back. The structure does not suddenly break; it can withstand increases in tension, but not for too long. A new strain concentration zone appears in the superior ligament, which generates the second snap-back, culminating in the rupture of the piece.

7. Conclusions

The present study proposes path-following methods for the finite element software Cast3M (Verpeaux et al., 1989), developed by the CEA (French Alternative Energies and Atomic Energy Commission, France) and freely distributed for research and educational purposes. This framework preserves the essence of the successful numerical methods in this field of study:

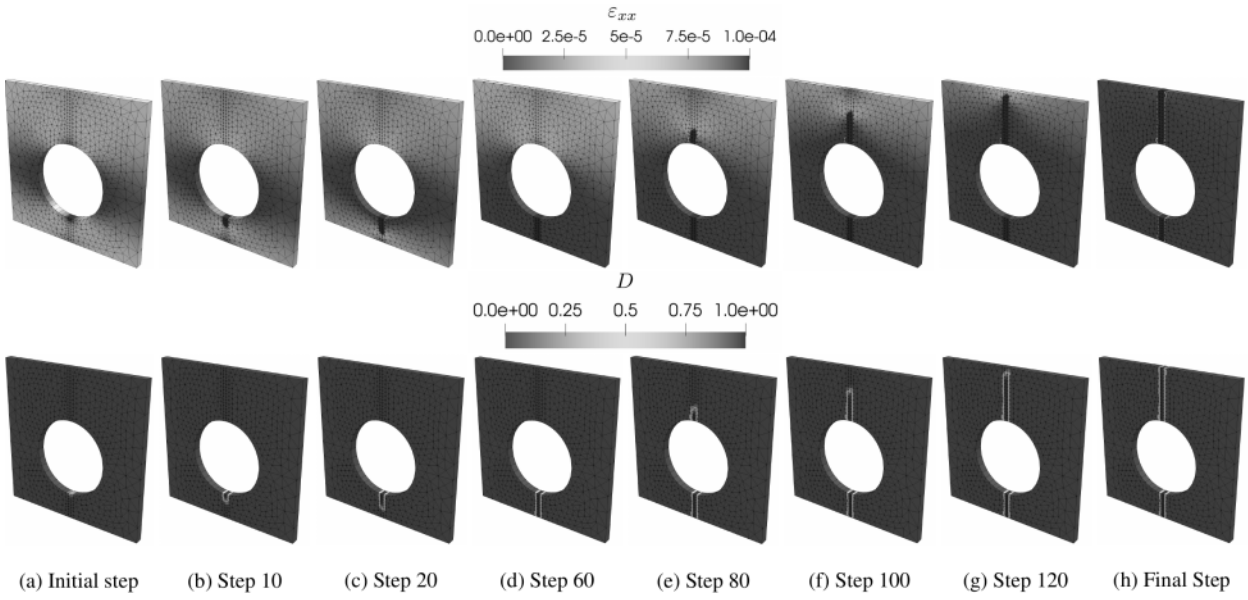

Figure 15: 3D holed plate under tensile loading – Horizontal strain (ϵ_{xx}) and damage field evolution.

(i) Breaking down the load into two components, one known and one unknown, the latter depending on an additional unknown: the load multiplier (Riks, 1979).

(ii) In a Newton-like solution process, breaking down the unknown displacement correction solution as the sum of two contributions that can be determined from two linearized systems of equilibrium equations that are independent of the load multiplier variation (Crisfield, 1983).

(iii) Determining the load multiplier variation from the solution of an additional equation of the problem: the path-following constraint equation.

The framework has a modular and independent design in order to accommodate multiple path-following constraint equations in a standalone way (i.e., without requiring modifications to the source code).

Three path-following equations were selected for implementation and testing: the control of the nodal displacement increment (CNDI), the control of the maximum strain increment (CMSI), and the control of the maximum elastic predictor (CMEP) over the computational domain. The CMSI formulation is proposed as the default case because of its general nature that makes it applicable to most situations regardless of the constitutive model. The other formulations were implemented through a simple overload of the new PILOINDI procedure.

These three path-following constraint criteria were applied to simple 2D and 3D test cases involving strain localization. The algorithm was found to be stable and convergent, even in the presence of severe snap-backs. The three methods were able to predict the peak load and the existence of post-peak instabilities.

The proposed framework will prove particularly helpful in studies involving new path-following equations. As demonstrated with the CNDI, CMSI, and CMEP criteria, the software users can focus on developing new path equations tailored to their needs.

Finally, this study was limited to structural instabilities induced by a nonlinear material response. However, the implementation proposed herein also works for geometrical nonlinearities, with the inclusion of large strains.

The developments presented in this paper are made available to the users/developers community along with Cast3M 2021 (release date: June 2021).

Acknowledgments

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

G. Alfano and M. A. Crisfield. Solution strategies for the delamination analysis based on a combination of local-control arc-length and line searches. *International Journal for Numerical Methods in Engineering*, 58(7):999–1048, 2003. ISSN 1097-0207. doi: 10.1002/nme.806.

I. Babuška. The finite element method with lagrangian multipliers. *Numerische Mathematik*, 20(3):179–192, 1973.

E. Barbieri, F. Ongaro, and N. M. Pugno. A j-integral-based arc-length solver for brittle and ductile crack propagation in finite deformation-finite strain hyperelastic solids with an application to graphene kirigami. *Computer Methods in Applied Mechanics and Engineering*, 315:713 – 743, 2017. ISSN 0045-7825. doi: https://doi.org/10.1016/j.cma.2016.10.043.

Z. P. Bažant. Structural stability. *International Journal of Solids and Structures*, 37(1-2):55–67, 2000.

D. Bellora and R. Vescovini. Hybrid geometric-dissipative arc-length methods for the quasi-static analysis of delamination problems. *Computers & Structures*, 175:123–133, 2016.

A. Bilotta, L. Leonetti, and G. Garcea. An algorithm for incremental elastoplastic analysis using equality constrained sequential quadratic programming. *Computers & structures*, 102:97–107, 2012.

B. Brank, A. Stanić, and A. Ibrahimbegovic. A path-following method based on plastic dissipation control. In *Computational Methods for Solids and Fluids*, pages 29–47. Springer, 2016.

F. Cazes, G. Meschke, and M.-M. Zhou. Strong discontinuity approaches: An algorithm for robust performance and comparative assessment of accuracy. *International Journal of Solids and Structures*, 96:355 – 379, 2016. ISSN 0020-7683. doi: https://doi.org/10.1016/j.ijsolstr.2016.05.016.

Z. Chen and H. L. Schreyer. A numerical solution scheme for softening problems involving total strain control. *Computers & Structures*, 37(6): 1043–1050, 1990.

M. Crisfield. An arc-length method including line searches and accelerations. *International journal for numerical methods in engineering*, 19(9): 1269–1289, 1983.

M. A. Crisfield. A fast incremental/iterative solution procedure that handles snap-through. *Computers & Structures*, 13(1-3):55–62, 1981.

R. De Borst. Computation of post-bifurcation and post-failure behavior of strain-softening solids. *Computers & Structures*, 25(2):211–224, 1987.

R. De Borst, M. A. Crisfield, J. J. C. Remmers, and C. V. Verhoosel. *Nonlinear finite element analysis of solids and structures*. John Wiley & Sons, 2012.

L. Ebersolt, A. Combescure, A. Millard, and P. Verpeaux. Non-linear algorithms solved with the help of the gibiane macro-language. In *Transactions of the 9th international conference on structural mechanics in reactor technology. Vol. B*, 1987.

G. Formica, V. Sansalone, and R. Casciaro. A mixed solution strategy for the nonlinear analysis of brick masonry walls. *Computer Methods in Applied Mechanics and Engineering*, 191(51):5847–5876, 2002. ISSN 0045-7825. doi: https://doi.org/10.1016/S0045-7825(02)00501-7. URL https://www.sciencedirect.com/science/article/pii/S0045782502005017.

G. Garcea, G. A. Trunfio, and R. Casciaro. Mixed formulation and locking in path-following nonlinear analysis. *Computer Methods in Applied Mechanics and Engineering*, 165(1):247–272, 1998. ISSN 0045-7825. doi: https://doi.org/10.1016/S0045-7825(98)00068-1. URL https://www.sciencedirect.com/science/article/pii/S0045782598000681.

M. G. D. Geers. Enhanced solution control for physically and geometrically non-linear problems. Part I – the subplane control approach. *International Journal for Numerical Methods in Engineering*, 46(2):177–204, 1999.

C. Giry, F. Dufour, and J. Mazars. Stress-based nonlocal damage model. *International Journal of Solids and Structures*, 48(25):3431 – 3443, 2011. ISSN 0020-7683. doi: https://doi.org/10.1016/j.ijsolstr.2011.08.012.

M. A. Gutiérrez. Energy release control for numerical simulations of failure in quasi-brittle solids. *International Journal for Numerical Methods in Biomedical Engineering*, 20(1):19–29, 2004.

M. A. Gutiérrez. Energy release control for numerical simulations of failure in quasi-brittle solids. *Communications in Numerical Methods in Engineering*, 20(1):19–29, 2004. ISSN 1099-0887. doi: 10.1002/cnm.649.

J. Hadamard. *Leçons sur la propagation des ondes et les équations de l'hydrodynamique*. A. Hermann, 1903.

R. Hill. Acceleration waves in solids. *Journal of the Mechanics and Physics of Solids*, 10(1):1–16, 1962.

A. Hillerborg, M. Modeer, and P. Petersson. Analysis of crack formation and crack growth in concrete by means of fracture mechanics and finite elements. *Cement and Concrete Research*, 6(6):773–781, 1976.

T. J. Hughes. *The finite element method: linear static and dynamic finite element analysis*. Dover Publications, 1987.

E. Lorentz and P. Badel. A new path-following constraint for strain-softening finite element simulations. *International journal for numerical methods in engineering*, 60(2):499–526, 2004.

D. Magisano, L. Leonetti, and G. Garcea. Advantages of the mixed format in geometrically nonlinear analysis of beams and shells using solid finite elements. *International Journal for Numerical Methods in Engineering*, 109(9):1237–1262, 2017.

D. Magisano, F. Liguori, L. Leonetti, D. de Gregorio, G. Zuccaro, and G. Garcea. A quasi-static nonlinear analysis for assessing the fire resistance of reinforced concrete 3d frames exploiting time-dependent yield surfaces. *Computers & Structures*, 212:327–342, 2019.

J. Mandel. Conditions de stabilité et postulat de drucker. In *Rheology and Soil Mechanics/Rhéologie et Mécanique des Sols*, pages 58–68. Springer, 1966.

P. Massin, G. Ferté, A. Caron, and N. Moës. Pilotage du chargement en formulation x-fem: application aux lois cohésives. In *10e colloque national en calcul des structures*, pages Clé–USB, 2011.

I. M. May and Y. Duan. A local arc-length procedure for strain softening. *Computers & structures*, 64(1):297–303, 1997.

S. May, J. Vignollet, and R. de Borst. A new arc-length control method based on the rates of the internal and the dissipated energy. *Engineering Computations*, 2016.

J. Mazars. Application de la mécanique de l'endommagement au comportement non linéaire et à la rupture du béton de structure. *Thèse de docteur es sciences, Université Pierre et Marie -PARIS 6*, 1984.

K. Moreau, N. Moës, N. Chevaugeon, and A. Salzman. Concurrent development of local and non-local damage with the thick level set approach: Implementation aspects and application to quasi-brittle failure. *Computer Methods in Applied Mechanics and Engineering*, 327:306 – 326, 2017. ISSN 0045-7825. doi: https://doi.org/10.1016/j.cma.2017.08.045. Advances in Computational Mechanics and Scientific Computation—the Cutting Edge.

A. Needleman and J. Rice. Limits to ductility set by plastic flow localization. In *Mechanics of sheet metal forming*, pages 237–267. Springer, 1978.

J. Oliver, A. Huespe, and J. Cante. An implicit/explicit integration scheme to increase computability of non-linear material and contact/friction problems. *Computer Methods in Applied Mechanics and Engineering*, 197(21):1865 – 1889, 2008. ISSN 0045-7825. doi: https://doi.org/10.1016/j.cma.2007.11.027.

R. H. J. Peerlings, R. De Borst, W. A. M. Brekelmans, and M. G. D. Geers. Gradient-enhanced damage modelling of concrete fracture. *Mechanics*

601 *of Cohesive-frictional Materials*, 3(4):323–342, 1998.
602 P. Pegon and A. Anthoine. Numerical strategies for solving continuum damage problems with softening: application to the homogenization of
603 masonry. *Computers & structures*, 64(1-4):623–642, 1997.
604 T. Pohl, E. Ramm, and M. Bischoff. Adaptive path following schemes for problems with softening. *Finite Elements in Analysis and Design*, 86:12
605 – 22, 2014. ISSN 0168-874X. doi: https://doi.org/10.1016/j.finel.2014.02.005.
606 E. Ramm. Strategies for tracing the nonlinear response near limit points. In *Nonlinear finite element analysis in structural mechanics*, pages 63–89.
607 Springer, 1981.
608 G. Rastiello, C. Boulay, S. D. Pont, J. Tailhan, and P. Rossi. Real-time water permeability evolution of a localized crack in concrete under loading.
609 *Cement and Concrete Research*, 56:20 – 28, 2014. ISSN 0008-8846. doi: https://doi.org/10.1016/j.cemconres.2013.09.010.
610 G. Rastiello, J.-L. Tailhan, P. Rossi, and S. Dal Pont. Macroscopic probabilistic cracking approach for the numerical modelling of fluid leakage in
611 concrete. *Annals of Solid and Structural Mechanics*, 7(1):pp.1–16, Jan. 2015. doi: 10.1007/s12356-015-0038-6.
612 G. Rastiello, C. Giry, F. Gatuingt, and R. Desmorat. From diffuse damage to strain localization from an eikonal non-local (enl) continuum damage
613 model with evolving internal length. *Computer Methods in Applied Mechanics and Engineering*, 331:650 – 674, 2018. ISSN 0045-7825. doi:
614 https://doi.org/10.1016/j.cma.2017.12.006.
615 G. Rastiello, F. Riccardi, and B. Richard. Discontinuity-scale path-following methods for the embedded discontinuity finite element modeling of
616 failure in solids. *Computer Methods in Applied Mechanics and Engineering*, 349:431–457, 2019.
617 J. R. Rice. Localization of plastic deformation. Technical report, Brown Univ., Providence, RI (USA). Div. of Engineering, 1976.
618 B. Richard, G. Rastiello, C. Giry, F. Riccardi, R. Paredes, E. Zafati, S. Kakarla, and C. Lejouad. Castlab: an object-oriented finite element toolbox
619 within the matlab environment for educational and research purposes in computational solid mechanics. *Advances in Engineering Software*,
620 128:136 – 151, 2019. ISSN 0965-9978. doi: https://doi.org/10.1016/j.advengsoft.2018.08.016.
621 E. Riks. *The Application of Newton's Method to the Problem of Elastic Stability*. American Society of Mechanical Engineers, 1972.
622 E. Riks. An incremental approach to the solution of snapping and buckling problems. *International journal of solids and structures*, 15(7):529–551,
623 1979.
624 N. Singh, C. Verhoosel, R. de Borst, and E. van Brummelen. A fracture-controlled path-following technique for phase-field modeling of brittle
625 fracture. *Finite Elements in Analysis and Design*, 113:14 – 29, 2016. ISSN 0168-874X. doi: https://doi.org/10.1016/j.finel.2015.12.005.
626 A. Stanić and B. Brank. A path-following method for elasto-plastic solids and structures based on control of plastic dissipation and plastic work.
627 *Finite Elements in Analysis and Design*, 123:1 – 8, 2017. ISSN 0168-874X. doi: https://doi.org/10.1016/j.finel.2016.09.005.
628 F. Thierry, G. Rastiello, C. Giry, and F. Gatuingt. One-dimensional eikonal non-local (enl) damage models: Influence of the integration rule for
629 computing interaction distances and indirect loading control on damage localization. *Mechanics Research Communications*, 110:103620, 2020.
630 ISSN 0093-6413. doi: https://doi.org/10.1016/j.mechrescom.2020.103620.
631 T. Y. Thomas. *Plastic Flow and Fracture in Solids by Tracy Y Thomas*. Elsevier, 1961.
632 C. V. Verhoosel, J. J. Remmers, and M. A. Gutiérrez. A dissipation-based arc-length method for robust simulation of brittle and ductile failure.
633 *International Journal for Numerical Methods in Engineering*, 77(9):1290–1321, 2009a.
634 C. V. Verhoosel, J. J. C. Remmers, and M. A. Gutiérrez. A dissipation-based arc-length method for robust simulation of brittle and ductile failure.
635 *International Journal for Numerical Methods in Engineering*, 77(9):1290–1321, 2009b. ISSN 1097-0207. doi: 10.1002/nme.2447.
636 P. Verpeaux, A. Millard, T. Charras, and A. Combescure. A modern approach of large computer codes for structural analysis. *IASMiRT*, 1989.
637 Y. Wang and H. Waisman. An arc-length method for controlled cohesive crack propagation using high-order xfem and irwin's crack closure
638 integral. *Engineering Fracture Mechanics*, 199:235 – 256, 2018. ISSN 0013-7944. doi: https://doi.org/10.1016/j.engfracmech.2018.05.018.
639 O. C. Zienkiewicz, R. L. Taylor, and J. Z. Zhu. *The finite element method: its basis and fundamentals*. Elsevier, 2005.