INSTITUT DE FRANCE
Académie des sciences

Comptes Rendus
Mécanique

Giuseppe Rastiello, Hugo Luiz Oliveira and Alain Millard

Path-following methods for unstable structural responses induced by strain softening: a critical review

Volume 350 (2022), p. 205-236

Published online: 16 June 2022

https://doi.org/10.5802/crmeca.112

Les Comptes Rendus. Mécanique *sont membres du Centre Mersenne pour l'édition scientifique ouverte*
www.centre-mersenne.org
e-ISSN : 1873-7234

Comptes Rendus
Mécanique
2022, 350, p. 205-236
https://doi.org/10.5802/crmeca.112

Short paper / *Note*

Path-following methods for unstable structural responses induced by strain softening: a critical review

Méthodes de pilotage indirect du chargement pour la description de réponses structurales instables induites par l'adoucissement en déformation : une analyse critique

Giuseppe Rastiello [*, a], **Hugo Luiz Oliveira** [a] **and Alain Millard** [a]

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

E-mails: giuseppe.rastiello@cea.fr (G. Rastiello), hugoitaime@alumni.usp.br (H. L. Oliveira), alain.millard.mail@gmail.com (A. Millard)

Abstract. Path-following methods for describing unstable structural responses induced by strain-softening are discussed. The main ingredients of the formalisms introduced by Riks and Crisfield for arc-length methods for geometrical non-linearities are presented. A link between two ways (monolithic and partitioned) of solving the resulting augmented equilibrium problem is discussed based on the Sherman–Morrison formula. The original monolithic approach assumes that the path-following constraint equation is differentiable with respect to the unknown displacement field and load factor. However, when dealing with material non-linearities, it is often preferred to consider constraint equations controlling the maximum of a field defined on the computational domain (e.g., a scalar strain measure, the rate of variation of an internal variable of the constitutive model). In that case, differentiability cannot be guaranteed due to the presence of the maximum operator. This makes only the partitioned formulation usable. Several path-following constraint equations from the literature are presented, and the corresponding implementations in the finite element method are discussed. The different formulations are compared based on a simple two-dimensional test case of damage localization in a beam submitted to tension. A test case involving multiple snap-backs is illustrated, finally, to show the robustness of the considered formulations.

Résumé. Des techniques de pilotage indirect du chargement pour décrire des réponses structurelles instables induites par l'adoucissement en déformation sont discutées. Après avoir rappelé les principaux ingrédients des formalismes introduits par Riks et Crisfield pour les méthodes de longueur d'arc pour le traitement de

* Corresponding author.

non-linéarités géométriques, un lien entre les deux façons (monolithique et partitionnée) de résoudre le problème d'équilibre augmenté qui en résulte est établi en se servant de la formule de Sherman–Morrison. L'utilisation de solveurs monolithiques repose sur l'hypothèse de différentiabilité des équations de pilotage par rapport aux inconnues du problème, le champ de déplacement et le facteur de chargement. Cependant, lorsqu'il s'agit de décrire des instabilités d'origines matérielles, des phénomènes locaux sont souvent responsables de la réponse instable obtenue au niveau global. Par conséquent, il est parfois préférable d'introduire des équations de pilotage permettant de contrôler le maximum d'un champ défini sur le domaine de calcul (e.g., la variation d'une mesure scalaire de déformation, le taux de variation d'une variable interne du modèle constitutif). Dans ce cas, la différentiabilité de l'équation de pilotage n'est plus garantie en raison de la présence de l'opérateur de maximum, ce qui rend utilisable uniquement la formulation partitionnée. Plusieurs exemples issus de la littérature scientifique pour les deux classes d'équations de pilotage sont présentés. Ensuite, on dérive les formulations d'éléments finis correspondantes et on décrit les avantages et inconvénients pour chacune d'elles. Une comparaison détaillée entre les différentes formulations est présentée sur la base d'un cas test bidimensionnel simple simulant la localisation de l'endommagement dans une poutre soumise à une sollicitation de traction. Un cas de test impliquant l'apparition de plusieurs snap-backs dans la courbe de réponse globale est enfin illustré pour montrer la robustesse des formulations considérées.

Keywords. Path-following methods, Monolithic and partitioned formulations, Dissipative constraint equations, Non-dissipative constraint equations, Material non-linearities.

Mots-clés. Méthodes de pilotage indirect du chargement, Formulations monolithique et partitionnée, Méthodes dissipatives, Méthodes non dissipatives, Non-linéarités matériaux.

Manuscript received 16th December 2021, revised 21st March 2022, accepted 28th April 2022.

1. Introduction

Unstable structural responses are often encountered in the numerical simulation of mechanical processes due to both material and geometrical non-linearities. The first kind of non-linearity depends on the mechanical behavior law that is assumed at the material level. In contrast, the second mainly depends on structural movements appearing in the medium due to large displacements and large strains. Both non-linearities may lead, under some circumstances, to unstable structural responses characterized by snap-backs and snap-throughs.

These kinds of behaviors cannot be adequately described through standard force- or displacement-controlled Newton-like solvers. The so-called path-following (in the engineering community) or continuation (in the mathematics community) methods allow overcoming such limitations by making the intensity of the external loading depend on the fulfillment of an additional problem equation: the path-following constraint equation [1–4]. The sole direction of the external load (displacement/traction) is prescribed, whereas the algorithm automatically determines its intensity to fulfill the resulting augmented equilibrium problem. This representation requires the introduction of an additional Degree-of-Freedom (DoF), the load factor, thereby expanding the dimensions of the solution space by one. Thanks to its generality, this formalism can also be extended to multi-parametric loading to model multiple indirectly controlled loading conditions [5]. In that case, m load factors and m constraint equations are introduced, thus increasing the size of the solution space by m.

There is extensive literature concerning path-following methods for unstable structural responses of both geometrical and material origins. Different numerical approaches have been developed, mainly in the Finite Element Method (FEM). Riks [2, 3] and Wempner [1] first proposed decomposing the external load vector into a prescribed part and an unknown part and introduced the so-called arc-length method. In this method, the constraint equation is a function of all the DoFs and can be linearized [6]. In this way, one can use a standard Newton solver where the stiffness matrix is augmented by additional blocks corresponding to the derivative of the equilibrium residual with respect to the load factor and to the derivatives of the constraint

equation with respect to the displacement vector and load factor. Consequently, the problem unknowns are computed by solving a "slightly" non-symmetric system of equations. However, the implementation of such an approach into existing FE simulation softwares requires freely modifying the stiffness and residual operators assembly process and lowers the performance of linear solvers since the banded nature of the stiffness matrix is destroyed. This motivated several authors [4, 7] to solve this problem in a partitioned/staggered fashion. This approach requires first solving the equilibrium equation twice to compute two independent contributions to the displacement correction in an iterative Newton-like solving process (the sole stiffness matrix has to be inverted, and the algorithm does not affect the assembly routines). Then, one solves the constraint equation to compute the load factor correction. This introduces an additional advantage with respect to monolithic solvers because the differentiability of the constraint equation with respect to the problem unknowns is no longer needed.

Arc-length methods [1–4, 7, 8] are well suited for geometrical non-linearities. However, they do not properly represent structural instabilities associated with localized dissipative processes occurring at the material level (e.g., strain localization) [9, 10]. In that case, just a few DoFs control the structural response [11]. Based on this consideration, De Borst [9] proposed a constraint equation for controlling a user-defined combination of a limited number of DoFs. Drawing from this idea, several constraint equations for controlling the strain increment at a given location of the finite element mesh [12–14] or the maximum strain variation on the computational domain [15] were proposed. Despite their relative simplicity, these formulations may fail due to the impossibility of distinguishing between dissipative and non-dissipative branches of the global equilibrium path. This motivated several authors to propose constraint equations ensuring that the dissipation is always positive [16–18]. This has been done through global constraints on the total dissipation [19–21], or using more local constraint equations, where attention is focused on the dissipative process occurring at the material level. Several methods of this kind were proposed in the literature. For instance, a constraint equation on the maximum value of the elastic predictor of the damage/plasticity function was proposed by Lorentz and Badel [15], whereas Rastiello *et al.* [22] developed a constraint equation allowing to control the maximum rate of variation of the internal variables of the constitutive model used to represent material non-linearity.

These methods were applied to non-linear FE simulations where material non-linearities were represented using Continuum Damage Mechanics models [15, 23, 24], phase-field formulations [16], and Thick Level-Set damage models [25]. Path-following methods were also employed in strong discontinuity simulations, where cracks were modeled using zero-thickness interface FEs [15, 26, 27], in the eXtended Finite Element Method (X-FEM) [28] and in the Embedded Finite Element Method (E-FEM) [22, 29–31]. Path-following algorithms for the Boundary Element Method (BEM) were also proposed in [32, 33], and more recently in [34].

Focusing on structural instabilities induced by strain-softening, the main aim of the present contribution is to propose a possible classification of available path-following constraint equations and test some of them to underline their advantages and inconveniences, in terms of not only obtained responses but also simplicity of implementation and resolution.

The paper is structured as follows. After presenting the Boundary Value Problem (BVP) to be solved briefly, attention is focused on monolithic and partitioned solving schemes. Then, several path-following constraint equations suitable for performing non-linear FE simulations in the presence of sharp snap-backs induced by material softening (a simple isotropic damage constitutive model is considered in this work) are presented. A classification of constraint equations according to their main features (global/local, differentiable/non-differentiable, dissipative/non-dissipative, model-dependent/model-independent) is proposed, several formulations are detailed, their implementations in a FE framework are illustrated, and their main

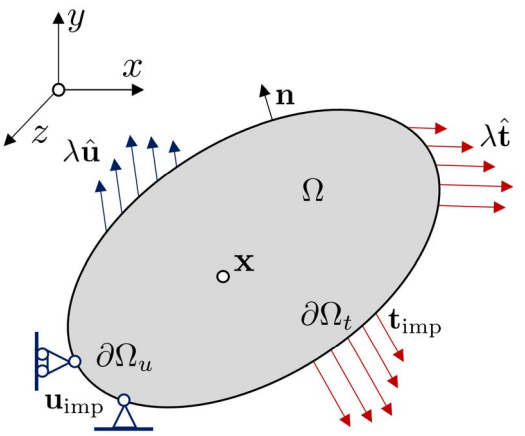

Figure 1. Continuum body Ω submitted to imposed and indirectly controlled tractions and displacements on its boundary.

advantages/inconveniences are underlined. Finally, some test cases are illustrated to show the different formulations main features. Some conclusions and perspectives close the paper.

2. Monolithic and partitioned path-following approaches

After presenting the variational BVP, this section introduces the FE formulation of path-following methods. Both monolithic and sequential approaches are shown. Then, attention is focused on a possible way to include indirectly controlled Dirichlet boundary conditions into the formulation.

2.1. *Boundary value problem*

Let us consider a continuum (Figure 1) occupying all points $\mathbf{x}$ belonging to a domain $\Omega \in \mathbb{R}^d$ (with $d = 1,2,3$) with boundary $\partial\Omega = \overline{\partial\Omega_u \cup \partial\Omega_t}$ (the over-bar denotes a closure). $\partial\Omega_t$ is the part of the boundary where external tractions are applied (Neumann conditions) and $\partial\Omega_u$ is the boundary where displacements are imposed (Dirichlet conditions). The split of $\partial\Omega$ is such that $\partial\Omega_u \cap \partial\Omega_t = \emptyset$. Body forces are neglected for the sake of simplicity.

Load decomposition. In path-following methods, the traction and displacement boundary conditions imposed to the system are written as [2, 4]:

$$\mathbf{u} = \mathbf{u}_{\text{imp}} + \lambda\hat{\mathbf{u}} \quad \forall\, \mathbf{x} \in \partial\Omega_u \tag{1}$$

$$\boldsymbol{\sigma}(\mathbf{u}) \cdot \mathbf{n} = \mathbf{t}_{\text{imp}} + \lambda\hat{\mathbf{t}} \quad \forall\, \mathbf{x} \in \partial\Omega_t \tag{2}$$

where $\mathbf{u} = \mathbf{u}(\mathbf{x}, t)$ is the unknown displacement field, $\boldsymbol{\sigma}(\mathbf{u})$ is the Cauchy stress tensor, λ is the unknown scalar load factor, and $\mathbf{n} = \mathbf{n}(\mathbf{x})$ is the outward normal vector to $\partial\Omega$. Finally, $(\mathbf{t}_{\text{imp}}, \mathbf{u}_{\text{imp}})$ are the imposed traction and displacement vectors respectively, and $(\hat{\mathbf{t}}, \hat{\mathbf{u}})$ are the directions of the indirectly controlled tractions and displacements.

Variational problem. Let us introduce the following admissibility spaces:

$$\mathscr{U}(\mathbf{u}_{\text{imp}} + \lambda\hat{\mathbf{u}}) = \{\mathbf{v} \,|\, \mathbf{v} \text{ regular enough on } \Omega,\, \mathbf{v} = \mathbf{u}_{\text{imp}} + \lambda\hat{\mathbf{u}} \text{ on } \partial\Omega_u\} \tag{3}$$

$$\mathscr{U}(0) = \{\mathbf{v} \,|\, \mathbf{v} \text{ regular enough on } \Omega,\, \mathbf{v} = 0 \text{ on } \partial\Omega_u\} \tag{4}$$

Under quasi-static and small strain conditions, the variational problem to be solved at time $t \in [0, T]$ reads:

Find $(\mathbf{u}, \lambda) \in \mathscr{U}(\mathbf{u}_{\text{imp}} + \lambda \hat{\mathbf{u}}) \times \mathbb{R}$ such that:

$$\int_\Omega \boldsymbol{\sigma}(\mathbf{u}) : \boldsymbol{\varepsilon}(\mathbf{w}) \, \mathrm{d}v - \lambda \int_{\partial\Omega_t} \hat{\mathbf{t}} \cdot \mathbf{w} \, \mathrm{d}a = \int_{\partial\Omega_t} \mathbf{t}_{\text{imp}} \cdot \mathbf{w} \, \mathrm{d}a \quad \forall \mathbf{w} \in \mathscr{U}(0) \tag{5}$$

$$p(\mathbf{u}, \lambda) = 0 \tag{6}$$

where ":" denotes the double contraction between tensors (second or higher order), $\boldsymbol{\varepsilon}(\mathbf{w}) = (\nabla \mathbf{w} + \nabla^\top \mathbf{w})/2$ is the small strain tensor applied to the virtual vector-valued field $\mathbf{w}$ (with ∇ denoting the gradient operator and $\square^\top$ the transpose of the second order tensor $\square$), and p is the so-called "path-following constraint equation". In the following, the latter is also called "constraint equation" or "constraint" for the sake of conciseness.

2.2. *Finite element formulation*

Let us now discretize Ω into a finite element mesh Ω^h containing n_{el} FEs (Ω_e), such that $\Omega^h = \cup_{e=1}^{n_{el}} \Omega_e$. The problem to be solved consists in finding the nodal displacement vector ($\mathbf{d}$) and the load factor (λ) such that:

$$\mathbf{r}(\mathbf{d}, \lambda) = \mathbf{f}^{\text{ext}}_{\text{imp}} + \lambda \hat{\mathbf{f}} - \mathbf{f}^{\text{int}}(\mathbf{d}) = \mathbf{0} \tag{7}$$

$$p(\mathbf{d}, \lambda) = 0 \tag{8}$$

where $\mathbf{r}(\mathbf{d}, \lambda)$ is the residual equilibrium vector and:

$$\mathbf{f}^{\text{ext}}_{\text{imp}} = \mathop{\mathbf{A}}_{e=1}^{n_{el}} \int_{\partial\Omega_e} \mathbf{N}^\top \mathbf{t}_{\text{imp}} \, \mathrm{d}a \tag{9}$$

$$\hat{\mathbf{f}} = \mathop{\mathbf{A}}_{e=1}^{n_{el}} \int_{\partial\Omega_e} \mathbf{N}^\top \hat{\mathbf{t}} \, \mathrm{d}a \tag{10}$$

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

Here, the symbol $\mathbf{A}$ denotes the assembly operator on Ω^h, $\mathbf{N}$ is the shape functions matrix, and $\mathbf{B}$ denotes the strain-displacement interpolation matrix. Moreover, $\boldsymbol{\sigma}$ is expressed in vector (matrix column) format.

2.3. *Solution*

The pseudo-time interval $[0, T]$ is discretized into an ordered sequence of steps $[t_0, t_1, \ldots, t_n, t_{n+1}, \ldots, T]$. Given the solution $\{\mathbf{d}_n, \lambda_n\}^\top$ at time t_n, one seeks iteratively for the increments $\{\Delta\mathbf{d}, \Delta\lambda\}^\top$ such that $\{\mathbf{d}_{n+1}, \lambda_{n+1}\}^\top = \{\mathbf{d}_n + \Delta\mathbf{d}, \lambda_n + \Delta\lambda\}^\top$ at time t_{n+1}. At iteration $k+1$, the accumulated solution increment is written as $\{\Delta\mathbf{d}^{k+1}, \Delta\lambda^{k+1}\}^\top = \{\Delta\mathbf{d}^k + \delta\mathbf{d}^{k+1}, \Delta\lambda^k + \delta\lambda^{k+1}\}^\top$, where iterate $\{\delta\mathbf{d}^{k+1}, \delta\lambda^{k+1}\}^\top$ is computed by solving a linearized augmented equilibrium problem. From now on, subscript n will always denote quantities computed at the last converged time step (t_n).

2.3.1. *Monolithic method*

The linearization of (7) and (8) based on the solution $(\mathbf{d}^k, \lambda^k)$ at iteration k gives [2, 3]:

$$\underbrace{\begin{bmatrix} \mathbf{K} & -\hat{\mathbf{f}} \\ \mathbf{h}^\top & w \end{bmatrix}}_{:=\tilde{\mathbf{K}}} \begin{Bmatrix} \delta\mathbf{d}^{k+1} \\ \delta\lambda^{k+1} \end{Bmatrix} = \begin{Bmatrix} \mathbf{r}^k \\ -p^k \end{Bmatrix} \tag{12}$$

where the stiffness matrix $\mathbf{K}$, the column matrix $\mathbf{h}$ and the scalar w read:

$$\mathbf{K} = -\frac{\partial \mathbf{r}}{\partial \mathbf{d}} = \frac{\partial \mathbf{f}^{\text{int}}}{\partial \mathbf{d}} \tag{13}$$

$$\mathbf{h} = \frac{\partial p}{\partial \mathbf{d}} \tag{14}$$

$$w = \frac{\partial p}{\partial \lambda} \tag{15}$$

Moreover:

$$\mathbf{r}^k = \mathbf{r}(\mathbf{d}^k, \lambda^k) = \mathbf{f}^{\text{ext}}_{\text{imp}} + \lambda^k \hat{\mathbf{f}} - \mathbf{f}^{\text{int}}(\mathbf{d}^k) \tag{16}$$

$$p^k = p(\mathbf{d}^k, \lambda^k) \tag{17}$$

Remark 1. According to (12), the symmetry of the stiffness-like operator to be inverted is lost. Solving can be efficiently done using direct (e.g., LU (Lower–Upper)) or iterative (e.g., GMRES (Generalized Minimum RESidual)) solvers. However, the banded nature of $\mathbf{K}$ is destroyed with detrimental effects on solvers convergence.

2.3.2. *Partitioned method*

In the early 80s, implementing the monolithic formulation into industrial FE analysis softwares was tricky due to the need to build the augmented system of equations (12). This difficulty motivated several authors (see, e.g., [4, 35]) to propose partitioned solving strategies such that only matrix $\mathbf{K}$ had to be inverted in (12) thus preserving the essential features of standard Newton solvers.

By rewriting the first equation of system (12) as [35]:

$$\mathbf{K}\delta\mathbf{d}^{k+1} = \mathbf{r}^k + \delta\lambda^{k+1}\hat{\mathbf{f}} \tag{18}$$

it is possible to write $\delta\mathbf{d}^{k+1}$ as the sum of two independent contributions:

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

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

$$\delta\mathbf{d}_{\text{I}}^{k+1} = \mathbf{K}^{-1}\mathbf{r}^k \tag{20}$$

$$\delta\mathbf{d}_{\text{II}}^{k+1} = \mathbf{K}^{-1}\hat{\mathbf{f}} \tag{21}$$

Once $\delta\mathbf{d}_{\text{I}}^{k+1}$ and $\delta\mathbf{d}_{\text{II}}^{k+1}$ are known, the load factor correction can be evaluated by substituting (19) into the second equation of (12):

$$\delta\lambda^{k+1} = -\frac{p^k + \mathbf{h}^\top\delta\mathbf{d}_{\text{I}}^{k+1}}{w + \mathbf{h}^\top\delta\mathbf{d}_{\text{II}}^{k+1}} \tag{22}$$

Finally, the solution correction vector can be written as:

$$\begin{Bmatrix}\delta\mathbf{d}^{k+1}\\ \delta\lambda^{k+1}\end{Bmatrix} = \begin{Bmatrix}\delta\mathbf{d}_{\text{I}}^{k+1}\\ 0\end{Bmatrix} - \frac{p^k + \mathbf{h}^\top\delta\mathbf{d}_{\text{I}}^{k+1}}{w + \mathbf{h}^\top\delta\mathbf{d}_{\text{II}}^{k+1}}\begin{Bmatrix}\delta\mathbf{d}_{\text{II}}^{k+1}\\ 1\end{Bmatrix} \tag{23}$$

Remark 2. According to such a partitioned approach, two symmetric system of equations are solved at each iteration to compute $(\delta\mathbf{d}_{\text{I}}^{k+1}, \delta\mathbf{d}_{\text{II}}^{k+1})$; then, the constraint equation is solved for $\delta\lambda^{k+1}$. Although this might be costly from a computational viewpoint (matrix $\mathbf{K}$ has to be inverted twice at every iteration),[1] its staggered nature renders the sequential approach easy to implement into most FE analysis softwares.

[1] Such a computational cost can be reduced by using direct/iterative solvers (e.g., MUMPS, SuperLU) supporting the solution of matrix systems for multiple right-hand sides at once.

Remark 3. The partitioned formulation is often preferable to the monolithic one because it can be used with differentiable and non-differentiable constraint equations. Sometimes, non-differentiable constraint equations are chosen to control the dissipative processes occurring in the domain more locally. In that case, linearizing the constraint equation becomes impossible. As a consequence, the monolithic formulation cannot be used.

Remark 4. Here, the external loading has been supposed to depend on a single load parameter λ since this is the classical choice. The generalization to multi-parametric loading is, however, possible [5]. More details are given in Appendix B.

2.4. *Dirichlet boundary conditions*

Indirectly controlled Dirichlet boundary conditions can be imposed in different ways (e.g., through penalization, row-column elimination, Lagrange multipliers, etc.). Here, a Lagrange multipliers formalism is described.

The variational problem is formulated by incorporating the unknown reaction forces ($\mathbf{t}$) associated in duality with the kinematic admissibility conditions on $\partial\Omega_u$. The variational problem to be solved now reads:

Find $(\mathbf{u}, \mathbf{t}, \lambda) \in \mathscr{U} \times \mathscr{U}'(\partial\Omega_u) \times \mathbb{R}$ such that:

$$\int_\Omega \boldsymbol{\sigma}(\mathbf{u}) : \boldsymbol{\epsilon}(\mathbf{w}) \, \mathrm{d}v - \int_{\partial\Omega_u} \mathbf{t}\cdot\mathbf{w} \, \mathrm{d}a - \lambda \int_{\partial\Omega_t} \hat{\mathbf{t}}\cdot\mathbf{w} \, \mathrm{d}a = \int_{\partial\Omega_u} \mathbf{t}_{\mathrm{imp}}\cdot\mathbf{w} \, \mathrm{d}a \quad \forall\, \mathbf{w} \in \mathscr{U} \tag{24}$$

$$\int_{\partial\Omega_u} \mathbf{u}\cdot\mathbf{s} \, \mathrm{d}a - \lambda \int_{\partial\Omega_u} \hat{\mathbf{u}}\cdot\mathbf{s} \, \mathrm{d}a = \int_{\partial\Omega_u} \mathbf{u}_{\mathrm{imp}}\cdot\mathbf{s} \, \mathrm{d}a \quad \forall\, \mathbf{s} \in \mathscr{U}'(\partial\Omega_u) \tag{25}$$

$$p(\mathbf{u}, \lambda) = 0 \quad \forall\, \mathbf{x} \in \Omega \tag{26}$$

where:

$$\mathscr{U} = \{\mathbf{v} \mid \mathbf{v} \text{ regular enough on } \Omega\} \tag{27}$$

and $\mathscr{U}'(\partial\Omega_u)$ is defined by duality with $\mathscr{U}$ [36].

The FE discretization of this problem is obtained by introducing an additional set of unknowns: the Lagrange multipliers vector ($\boldsymbol{\eta}$) [37]. The linearized problem to be solved reads:

$$\begin{bmatrix} \mathbf{K}_* & -\hat{\mathbf{f}}_* \\ \mathbf{h}_*^\top & w \end{bmatrix} \begin{Bmatrix} \delta\mathbf{d}_*^{k+1} \\ \delta\lambda^{k+1} \end{Bmatrix} = \begin{Bmatrix} \mathbf{r}_*^k \\ -p^k \end{Bmatrix} \tag{28}$$

where:

$$\mathbf{K}_* = \begin{bmatrix} \mathbf{K} & -\mathbf{C} \\ \mathbf{C}^\top & \mathbf{0} \end{bmatrix} \quad \mathbf{r}_*^k = \begin{Bmatrix} \mathbf{r}^k \\ \mathbf{l}^k \end{Bmatrix} \quad \hat{\mathbf{f}}_* = \begin{Bmatrix} \hat{\mathbf{f}} \\ \hat{\mathbf{d}} \end{Bmatrix} \quad \mathbf{h}_* = \begin{Bmatrix} \mathbf{h} \\ \mathbf{0} \end{Bmatrix} \quad \delta\mathbf{d}_*^{k+1} = \begin{Bmatrix} \delta\mathbf{d}^{k+1} \\ \delta\boldsymbol{\eta}^{k+1} \end{Bmatrix} \tag{29}$$

Here, $\mathbf{C}$ is the matrix of the Dirichlet boundary conditions such that:

$$\mathbf{C}^\top\mathbf{d} = \mathbf{d}_{\mathrm{imp}} + \lambda\,\hat{\mathbf{d}} \tag{30}$$

with $(\mathbf{d}_{\mathrm{imp}}, \hat{\mathbf{d}})$ denoting the nodal counterparts of $(\mathbf{u}_{\mathrm{imp}}, \hat{\mathbf{u}})$. Moreover:

$$\mathbf{r}^k = \mathbf{r}(\mathbf{d}^k, \boldsymbol{\eta}^k, \lambda^k) = \mathbf{f}_{\mathrm{imp}}^{\mathrm{ext}} + \lambda^k\hat{\mathbf{f}} + \mathbf{C}\boldsymbol{\eta}^k - \mathbf{f}^{\mathrm{int}}(\mathbf{d}^k) \tag{31}$$

$$\mathbf{l}^k = \mathbf{l}(\mathbf{d}^k, \lambda^k) = -\mathbf{C}^\top\mathbf{d}^k + \mathbf{d}_{\mathrm{imp}} + \lambda^k\hat{\mathbf{d}} \tag{32}$$

Remark 5. Since the constraint equation (26) is independent of the Lagrange multipliers [38], deriving the partitioned formulation is straightforward [39].

Remark 6. It is well known that solving (28) can become tricky from a computational viewpoint since one has to solve a so-called "saddle point problem". To circumvent this issue, some FE softwares (e.g., Cast3M [40] and CodeAster [41]) use a slightly different formalism based on the introduction of two sets of Lagrange multipliers [42]. The resulting discretized linearized system to be solved is still non-symmetric, but no zero values are present on the diagonal of the stiffness-like operator [38, 39].

3. Path-following constraint equations

After introducing a possible classification of constraint equations,[2] some formulations suitable for representing unstable responses caused by non-linear material behaviors (strain softening) are presented. Then, the main equations for implementing them into monolithic and sequential path-following solvers are illustrated. Their main advantages and inconveniences are discussed.

3.1. *Classification of constraint equations*

Differentiable vs. non-differentiable. Constraint equations are first classified according to two main classes depending on their differentiability.

Differentiable constraint equations are certainly the simplest to develop and solve. They are such that the dependency on the problem DoFs ($\mathbf{d}, \lambda$) is explicit [9, 19, 43], which makes immediate the use of monolithic and sequential solvers [6].

Despite their relative simplicity of formulation and implementation, these constraint equations may sometimes be unable to adequately describe unstable responses, particularly for structures undergoing highly localized failure processes. To solve this issue, it is often preferable to write constraint equations allowing to control the maximum of a field of interest on the computational domain [15, 22]. In that case, the differentiability of the constraint equation cannot be ensured due to the maximum operator presence, and using a monolithic solver is no longer possible (one cannot compute $\mathbf{h}$ and w). Conversely, the sequential approach can still be used. In that case, however, solving the constraint equation requires ad-hoc minimization procedures to compute $\delta\lambda^{k+1}$ after having derived a mesh-partition-wise differentiable constraint equation (in the most common case, one looks for integration point-wise differentiable equations). This adds an extra computational cost to the numerical procedure but naturally allows describing complex damaging/cracking processes adequately (e.g., in the presence of multiple cracks, crack branching).

It should be noted that element-wise linearization is not mandatory. Most of the algorithms introduced in the following sections use linear (by construction) or linearized constraint equations at integration points for the sake of simplicity and robustness. However, in the case of quadratic constraint equations, for instance, one could also imagine solving them numerically or analytically.

Global vs. local. An additional distinction can be made between "global" and "local" constraint equations. With the term global, we designate constraint equations involving the whole unknown displacement vector. In contrast, the term local indicates constraint equations where attention is restrained on a reduced set of DoFs. This can be done explicitly by the user [9, 44] or in an automatic/adaptive way [14, 15, 22].

Dissipative vs. non-dissipative. Inside each family of constraint equations, one can also distinguish between dissipative and non-dissipative formulations. Dissipative methods [15, 19, 22, 43] allow discriminating between dissipative equilibrium paths and elastic unloading-reloading branches, whereas non-dissipative cannot. In other words, given an equilibrium state, the next equilibrium solution found by the algorithm can be dissipative (i.e., such that additional dissipation occurs in the domain) or non-dissipative (i.e., such that elastic unloading of the medium is described). In that second case, unloading may be observed in the overall structural response, even though it has neither been prescribed nor an unstable bifurcation point has been passed.

[2] A classification of constraint equations was also proposed by Geers [13].

This unloading phenomenon is often referred to as artificial unloading. That means that unnecessary computational efforts are made to describe unwanted equilibrium states. More importantly, nothing ensures that these non-dissipative methods achieve a dissipative solution after artificial unloading.

Model-independent vs. model-dependent. Finally, one can distinguish between model-dependent and model-independent constraint equations. In particular, model-independent constraint equations are independent of the material behavior, e.g., when the constraint equation is a function of the problem DoFs or the strains (and thus, indirectly, of the DoFs). They are, therefore, quite general and can be employed with any constitutive model. Conversely, model-dependent constraints require ad-hoc developments depending on the constitutive assumptions introduced by the user (e.g., when the linearization of a plasticity/damage criterion is needed [15, 22]).

3.2. *Constitutive assumptions*

In the presentation of model-dependent constraint equations, it will be assumed that dissipation occurs in the material according to a simple isotropic Continuum Damage model [45] for the sake of simplicity. Moreover, no regularization methods will be used to induce sharp snap-back responses in the numerical examples. The presented path-following techniques can, however, be used even in that case without significant changes (see, e.g., [13, 15, 23, 24, 46]).

Denoting by (λ, μ) the Lame's constants, by "tr" the trace operator, and by $\mathbf{I}$ the second-order identity tensor, the Cauchy stress tensor reads:

$$\boldsymbol{\sigma}(\mathbf{u}) = (1 - D)[\lambda \mathrm{tr}(\boldsymbol{\varepsilon}(\mathbf{u}))\mathbf{I} + 2\mu\boldsymbol{\varepsilon}(\mathbf{u})] \tag{33}$$

where $D \in [0, 1]$ is a scalar isotropic damage variable. It rises from zero (intact state) to one (fully broken state) according to the exponential evolution law:

$$D = r(\kappa) = 1 - \exp[-\beta(\kappa - \kappa_0)] \tag{34}$$

where κ is a history variable giving the historical maximum of an equivalent strain measure ($\tilde{\epsilon}$) during the damaging phase:

$$\kappa = \max_t(\kappa_0, \tilde{\epsilon}) \tag{35}$$

In the previous equations, κ_0 is the damage activation threshold and β is a scalar parameter controlling the shape of the damage evolution function (i.e., the dissipated energy). Here, $\tilde{\epsilon}$ is the equivalent Mazars' strain [47], $\tilde{\epsilon} = \tilde{\epsilon}(\boldsymbol{\varepsilon}(\mathbf{u})) = \sqrt{\langle\boldsymbol{\varepsilon}(\mathbf{u})\rangle : \langle\boldsymbol{\varepsilon}(\mathbf{u})\rangle}$, with $\langle\square\rangle$ denoting the positive part operator. Alternatively, one could have used, for instance, the dissipative thermodynamic force associated with the damage variable [48].

Finally, the damage activation function (f) is:

$$f = f(\tilde{\epsilon}, \kappa) = \tilde{\epsilon} - \kappa \tag{36}$$

and the Karush–Kuhn–Tucker (KKT) conditions read:

$$f \leq 0 \tag{37}$$

$$\dot{\kappa} \geq 0 \tag{38}$$

$$f\dot{\kappa} = 0 \tag{39}$$

3.3. *Differentiable constraint equations*

The formulations proposed in [1–4,7] (among other works) for geometrical non-linearities belong to the category of differentiable constraint equations. In a unified form, they can be written as:[3]

$$p = \Delta\mathbf{d}^\top \Delta\mathbf{d} + \zeta^2 \Delta\lambda^2 \hat{\mathbf{f}}^\top \hat{\mathbf{f}} - \Delta\tau^2 = 0 \tag{40}$$

where $\Delta\tau$ is the step length parametrizing the equilibrium path and ζ is a real parameter.

Depending on the chosen ζ value, different formulations can be obtained. In particular, $\zeta = 1$ corresponds to the so-called "spherical arc-length" method, whereas $\zeta = 0$ gives the "cylindrical arc-length" method. For values of ζ between 0 and 1, equation (40) defines a hyper-ellipse in the multidimensional space $(\mathbf{d}, \lambda)$.

3.3.1. *Control on nodal displ. increment, CNDI (non-dissipative, local, model-independent)*

Equation (40) is suitable for geometrical non-linearities. However, it may fail when dealing with localized failure phenomena [9, 13] since only a few DoFs control the structural response. Napoleão et al. [11], for instance, showed that only a few eigenvectors extracted from the tangent stiffness matrix are active when the material response is non-linear.

Drawing from the cylindrical arc-length method [4] and the updated normal path equation [7], De Borst [9] was the first to propose a constraint equation to control a weighted combination of the DoFs. This formulation is often called "indirect displacement control". A similar approach was developed by Chen and Schreyer [44, 49]. In the cited works, instead of the DoFs, the authors proposed to control a combination of scalar measures of the strain variation of a set of FEs.

Constraint equation. Denoting by $\mathbf{S}$ a square DoFs selection matrix, the constraint equation to be solved at iteration $k+1$ can be written as [19]:

$$p = (\Delta\mathbf{d}_n)^\top \mathbf{S} \Delta\mathbf{d}^{k+1} - \Delta\tau^2 = 0 \tag{41}$$

where $\Delta\mathbf{d}_n$ is the accumulated displacement variation at the last converged time step. Alternatively, one may write [22, 39]:

$$p = \mathbf{s}^\top \Delta\mathbf{d}^{k+1} - \Delta\tau = 0 = \mathbf{s}^\top (\Delta\mathbf{d}^k + \delta\mathbf{d}^{k+1}) - \Delta\tau = 0 \tag{42}$$

where $\mathbf{s}$ is a column matrix containing the coefficients of the combination of the DoFs.

Monolithic method. Starting from (42), terms $(p^k, w, \mathbf{h})$ appearing in (12) can be expressed as:

$$p^k = \mathbf{s}^\top \Delta\mathbf{d}^k - \Delta\tau \tag{43}$$

$$\mathbf{h} = \mathbf{s} \tag{44}$$

$$w = 0 \tag{45}$$

Partitioned method. Substituting (19) into (42) one obtains:

$$p = \mathbf{s}^\top (\Delta\mathbf{d}^k + \delta\mathbf{d}_{\mathrm{I}}^{k+1} + \delta\lambda^{k+1} \delta\mathbf{d}_{\mathrm{II}}^{k+1}) - \Delta\tau = 0 \tag{46}$$

Consequently, $\delta\lambda^{k+1}$ can be computed as:

$$\delta\lambda^{k+1} = \frac{\Delta\tau - a_0}{a_1} \tag{47}$$

where coefficients a_0 and a_1 depend on already computed quantities as:

$$a_0 = \mathbf{s}^\top (\Delta\mathbf{d}^k + \delta\mathbf{d}_{\mathrm{I}}^{k+1}) \tag{48}$$

$$a_1 = \mathbf{s}^\top \delta\mathbf{d}_{\mathrm{II}}^{k+1} \tag{49}$$

[3] Equation (40) is often called the Riks–Wempner arc-length equation.

Alternatively, one could have directly inserted equations (43) to (45) into (22), which leads exactly to (47). The proof is straightforward.

Remark 7. Constraint (42) ensures excellent numerical robustness, in particular when dealing with highly localized failure processes. Furthermore, it is useful for numerically reproducing experimental loading conditions[4] (servo-controlled loading) [50, 51]. Nevertheless, choosing **s** is not always straightforward because it demands knowing/predicting the failure process before performing the calculations. If vector **s** is not conveniently chosen, the method may fail [19, 22] and some parts of the equilibrium path may be skipped. Moreover, although numerical strategies have been proposed to adapt **s** throughout the simulation, the existence of a solution is not guaranteed [15].

3.3.2. *Control of the integral of the internal variable increment, CI2VI (dissipative, global, model-dependent)*

Fayezioghani *et al.* [21] recently proposed a global constraint equation on the integral of the variation of the history variable of the material model over Ω, i.e., $p = \int_\Omega \dot{\bar{\kappa}}\,\mathrm{d}v - \dot{\tau} = 0$, with $\bar{\kappa} = \max_{\tau<t}(\tilde{e}(\tau)) \neq \kappa$. The same authors also introduced an alternative formulation to diminish the role of material points having already undergone high damage levels, thus controlling the response of material points experiencing the first phases of damage evolution, $p = \int_\Omega (1-D)\dot{\bar{\kappa}}\,\mathrm{d}v - \dot{\tau} = 0$. Notice that $\bar{\kappa} = \kappa > \kappa_0$ only during the damaging phase, whereas it is lower before. According to [21], such a constraint equation allows describing both the dissipative and non-dissipative branches of the equilibrium solution.

Constraint equation. Here, these formulations are restated to control the integral of $\dot{\kappa}$ instead of $\dot{\bar{\kappa}}$ and make them dissipative. The constraint equation is written in a generalized format as:

$$p = \int_\Omega (1-D)^\phi \dot{\kappa}\,\mathrm{d}v - \dot{\tau} = 0 \tag{50}$$

where $\phi \in [0,1]$ is a real parameter such that both formulations can be recovered depending on the chosen value. Moreover, during the damaging phase (i.e., $\dot{\kappa} > 0$) one has:

$$\dot{\kappa} = \frac{\mathrm{d}\kappa}{\mathrm{d}\langle\boldsymbol{\varepsilon}\rangle} : \frac{\mathrm{d}\langle\boldsymbol{\varepsilon}\rangle}{\mathrm{d}\boldsymbol{\varepsilon}} : \dot{\boldsymbol{\varepsilon}} = \frac{\mathrm{d}\kappa}{\mathrm{d}\langle\boldsymbol{\varepsilon}_p\rangle} : \frac{\mathrm{d}\langle\boldsymbol{\varepsilon}_p\rangle}{\mathrm{d}\boldsymbol{\varepsilon}_p} : \frac{\mathrm{d}\boldsymbol{\varepsilon}_p}{\mathrm{d}\boldsymbol{\varepsilon}} : \dot{\boldsymbol{\varepsilon}} \tag{51}$$

where $\boldsymbol{\varepsilon}_p$ is the strain tensor represented in its principal basis. Accordingly, $\langle\boldsymbol{\varepsilon}_p\rangle$ is the diagonal tensor of the positive principal strains.

In a FE context, using an explicit prediction for the damage variable at integration points, the time-discretized constraint equation reads:

$$p = \sum_{\alpha\in\mathscr{G}} [(1-D_{\alpha,n})^\phi \Delta\kappa_\alpha^{k+1} W_\alpha] - \Delta\tau = 0 \tag{52}$$

where W_α is the integration weight associated with integration point α (the jacobian of the FE mapping is already included in W_α for the sake of conciseness), $\mathscr{G}$ is the set of integration points of Ω^h, and:

$$\Delta\kappa_\alpha^{k+1} = \kappa_\alpha^{k+1} - \kappa_{\alpha,n} \tag{53}$$

[4] For instance, when the loading is indirectly controlled through the Crack Mouth Opening Displacement (CMOD).

Monolithic method. Scalars (p^k, w) and vector $\mathbf{h}$ figuring in (12) read:

$$p^k = \sum_{\alpha\in\mathcal{G}} [(1-D_{\alpha,n})^{\phi}\Delta\kappa_\alpha^k W_\alpha] - \Delta\tau \tag{54}$$

$$\mathbf{h} = \mathop{\mathbf{A}}_{e=1}^{n_{el}} \mathbf{h}_e \quad \mathbf{h}_e = \sum_{\alpha\in\mathcal{G}_e} \begin{cases} [(1-D_{\alpha,n})^{\phi} W_\alpha \mathbf{b}_\alpha^\top] & \kappa_\alpha^k > \kappa_{\alpha,n} \\ \mathbf{0} & \text{otherwise} \end{cases} \tag{55}$$

$$w = 0 \tag{56}$$

where $\mathcal{G}_e$ is the set of integration points of Ω_e, and the derivative of the history variable with respect to the total strain tensor ($\mathbf{b}_\alpha$) depends on the chosen damage evolution model. For the model considered herein, one has:

$$\mathbf{b}_\alpha = \frac{\langle \boldsymbol{\epsilon}_{p,\alpha}^k \rangle^\top}{\tilde{\epsilon}_\alpha^k} \mathbf{G}_\alpha^k \mathbf{R}_\alpha^k \mathbf{B}_\alpha \tag{57}$$

where $\langle \boldsymbol{\epsilon}_{p,\alpha}^k \rangle = \mathbf{G}_\alpha^k \mathbf{R}_\alpha^k \mathbf{B}_\alpha \mathbf{d}^k$ is the vector representation of tensor $\langle \boldsymbol{\epsilon}_\alpha^k \rangle$, $\mathbf{G}_\alpha^k$ is the square diagonal matrix such that $G_{\alpha,ii}^k = 1$ if $\epsilon_{p,\alpha,i}^k > 0$ and is null otherwise, and $\mathbf{R}_\alpha^k$ is the matrix that rotates $\boldsymbol{\epsilon}_\alpha^k = \mathbf{B}_\alpha \mathbf{d}^k$ into $\boldsymbol{\epsilon}_{p,\alpha}^k$.

Partitioned method. $\delta\lambda^{k+1}$ can be immediately obtained from (47) with:

$$a_0 = \sum_{\alpha\in\mathcal{G}} [(1-D_{\alpha,n})^{\phi}\Delta\kappa_\alpha^k W_\alpha] + \mathbf{h}^\top \delta\mathbf{d}_{\mathrm{I}}^{k+1} \tag{58}$$

$$a_1 = \mathbf{h}^\top \delta\mathbf{d}_{\mathrm{II}}^{k+1} \tag{59}$$

Remark 8. Implementing such constraint equations is straightforward, both into monolithic and partitioned solvers. However, the integration process makes the choice of $\Delta\tau$ less straightforward (and less easy to justify from a physical viewpoint) than in other methods since the size of the integration domain has to be taken into account. Implementing the original formulation by [21] is similar, the main difference being that $\dot{\tilde{\kappa}}$ is used instead of $\dot{\kappa}$ in (50). It is worth noticing, however, that given the choice of controlling the integral of $\dot{\kappa}$ instead of the integral of $\dot{\tilde{\kappa}}$, using (50) requires a preliminary computation (at the first pseudo-time step) of the load factor such that dissipation starts in the domain. This will not be the case with the other dissipative constraints illustrated in the remainder of this work.

3.4. *Non-differentiable constraint equations*

Attention is now focused on a second family of constraint equations where the load factor is computed by maximizing a function of interest (g) on the domain. In discretized form, these constraint equations can be put in the general form:

$$p = \max_{\alpha\in\mathcal{G}}(g_\alpha) - \Delta\tau = \max_{\alpha\in\mathcal{G}}(g_\alpha(\delta\lambda^{k+1}; \underbrace{\mathbf{d}_n, \lambda_n, \Delta\mathbf{d}^k, \Delta\lambda^k, \delta\mathbf{d}_{\mathrm{I}}^{k+1}, \delta\mathbf{d}_{\mathrm{II}}^{k+1}}_{\text{known quantities}})) - \Delta\tau = 0 \tag{60}$$

where the displacement decomposition (19) was already taken into account.[5] All constraint equations are written in this manuscript considering integration point-wise fields (i.e., fields defined at integration points), such as strains, damage, etc. The maximum is thus sought over integration points, but nodal fields could also be considered.

[5] As already mentioned, this kind of constraint equation is not differentiable due to the presence of the maximum operator. As a result, the sole partitioned formulation can be used.

Nested interval algorithm. During the solving process, the integration point α such that (60) is fulfilled, is sought automatically, without requiring any arbitrary user choice. This can be done, for instance, using a "nested interval algorithm" [15]. The sequence of tasks to be performed can be summarized as follows:

- *Initialization.* One first initializes the admissibility interval of $\delta\lambda^{k+1}$:

$$\mathscr{L} = (-\infty, \infty) \tag{61}$$

- *Iteration.* Then, one starts looping over integration points $\alpha \in \mathscr{G}$ and solves the constraint equation for each α to compute the corresponding load factor correction $\delta\lambda_\alpha^{k+1}$. Since function $g_\alpha(\delta\lambda^{k+1}) = g_\alpha(\delta\lambda^{k+1};\ldots)$ is generally non-linear, a linearized constraint equation $g_\alpha^{\text{lin}}(\delta\lambda_\alpha^{k+1}) = g_\alpha^{\text{lin}}(\delta\lambda_\alpha^{k+1};\ldots)$ is used. One solves:

$$g_\alpha^{\text{lin}}(\delta\lambda_\alpha^{k+1}) - \Delta\tau = a_{\alpha,0} + a_{\alpha,1}\delta\lambda_\alpha^{k+1} - \Delta\tau = 0 \tag{62}$$

  where, thanks to (19), coefficients $a_{\alpha,0}$ and $a_{\alpha,1}$ depend on known quantities:

$$a_{\alpha,\square} = a_{\alpha,\square}(\mathbf{d}_n, \lambda_n, \Delta\mathbf{d}^k, \Delta\lambda^k, \delta\mathbf{d}_{\text{I}}^{k+1}, \delta\mathbf{d}_{\text{II}}^{k+1}) \qquad \square = 0, 1 \tag{63}$$

  Iterate $\delta\lambda_\alpha^{k+1}$ is then computed as:

$$\delta\lambda_\alpha^{k+1} = \frac{\Delta\tau - a_{\alpha,0}}{a_{\alpha,1}} \tag{64}$$

  Finally, one refines set $\mathscr{L}$ based on $\delta\lambda_\alpha^{k+1}$ and the sign of $a_{\alpha,1}$:

$$\mathscr{L} \leftarrow \mathscr{L} \cap \begin{cases} (-\infty, \delta\lambda_\alpha^{k+1}] & \text{if } a_{\alpha,1} > 0 \\ [\delta\lambda_\alpha^{k+1}, +\infty) & \text{otherwise} \end{cases} \tag{65}$$

  If set $\mathscr{L}$ is empty, no solutions exist. In that case, one should reduce $\Delta\tau$ and restart iterating from the previous equilibrium solution.
- *Selection.* Finally, one selects the load factor correction $\delta\lambda^{k+1} \in \mathscr{L} = [\delta\lambda_{\min}, \delta\lambda_{\max}]$ that minimizes, for instance, the norm of the displacement correction:

$$\delta\lambda^{k+1} = \text{argmin}_{\delta\lambda^\star \in \mathscr{L}} \|\delta\mathbf{d}_{\text{I}}^{k+1} + \delta\lambda^\star \delta\mathbf{d}_{\text{II}}^{k+1}\| \tag{66}$$

  Some authors [15, 52] suggest other criteria. For instance, one can minimize the global residual to improve the stability of the convergence of the Newton–Raphson algorithm:

$$\delta\lambda^{k+1} = \text{argmin}_{\delta\lambda^\star \in \mathscr{L}} \|\mathbf{r}^{k+1}(\delta\lambda^\star)\| \tag{67}$$

  A more common choice in arc-length methods is to select the solution which minimizes the angle between the present and past incremental displacements:

$$\delta\lambda^{k+1} = \text{argmin}_{\delta\lambda^\star \in \mathscr{L}} \left[\frac{-(\Delta\mathbf{d}^k)^\top \Delta\mathbf{d}^{k+1}(\delta\lambda^\star)}{\|(\Delta\mathbf{d}^k)^\top \Delta\mathbf{d}^{k+1}(\delta\lambda^\star)\|} \right] \tag{68}$$

  Unfortunately, there is no universally valid choice for such a criterion. Ad-hoc choices need to be made depending on the adopted path-following constraint equations.

3.4.1. *Control by maximal strain increment, CMSI (non-dissipative, local, model-independent)*

The first attempt to control a scalar measure of the strain variation in the "most critical" mesh region was made by Chen and Schreyer [44]. In that work, the critical FE was chosen as the one having experienced the largest distortion. Selecting such an element is, however, not straightforward.

A robust way to make the algorithm automatically determine the critical FE to be controlled is to write a constraint equation on the maximum value of a scalar measure of the strain variation on

the computational domain [15, 22]. With $\mathbf{q}_\alpha = \boldsymbol{\epsilon}_n / \|\boldsymbol{\epsilon}_n\| = \mathbf{B}_\alpha \mathbf{d}_n / \|\mathbf{B}_\alpha \mathbf{d}_n\|$ denoting a unity column matrix defined according to [44], the constraint equation reads:

$$p = \max_{\alpha \in \mathcal{G}}(g_\alpha) - \Delta\tau = \max_{\alpha \in \mathcal{G}}(\mathbf{q}_\alpha^\top \Delta\boldsymbol{\epsilon}_\alpha^{k+1}) - \Delta\tau = 0 \tag{69}$$

where:

$$\Delta\boldsymbol{\epsilon}_\alpha^{k+1} = \Delta\boldsymbol{\epsilon}_\alpha^{k+1}(\delta\lambda^{k+1}) = \mathbf{B}_\alpha(\Delta\mathbf{d}^k + \delta\mathbf{d}_{\mathrm{I}}^{k+1} + \delta\lambda^{k+1}\delta\mathbf{d}_{\mathrm{II}}^{k+1}) \tag{70}$$

The linear equation to be solved at each integration point thus reads as in (62) with:

$$a_{\alpha,0} = \mathbf{q}_\alpha^\top \mathbf{B}_\alpha(\Delta\mathbf{d}^k + \delta\mathbf{d}_{\mathrm{I}}^{k+1}) \tag{71}$$

$$a_{\alpha,1} = \mathbf{q}_\alpha^\top \mathbf{B}_\alpha \delta\mathbf{d}_{\mathrm{II}}^{k+1} \tag{72}$$

Remark 9. The CMSI constraint is the simplest formulation to control the simulation through a local quantity. Its crucial feature is generality, the constraint equation being independent of the constitutive model used to describe the material response. In other words, once implemented in the FE software, no additional developments are needed for using it with different material laws. As a counterpart of this, however, one cannot avoid artificial elastic unloading paths [15, 22]. Moreover, the number of artificial unloading paths cannot be predicted before running the simulation, meaning that much computational time is possibly spent to describe uninteresting elastic solutions.

3.4.2. *Control by maximal elastic predictor, CMEP (dissipative, local, model-dependent)*

A path-following constraint equation on the maximal elastic predictor (f_α^{elas}) of the damage criterion function (i.e., the value of f_α when considering an incrementally elastic material response) was proposed by Lorentz and Badel [15]:

$$p = \max_{\alpha \in \mathcal{G}}(g_\alpha) - \Delta\tau = \max_{\alpha \in \mathcal{G}}(f_\alpha^{\mathrm{elas},k+1}) - \Delta\tau = 0, \quad \Delta\tau > 0 \tag{73}$$

Since a positive value of the elastic predictor function is always associated with a dissipative branch of the equilibrium path,[6] formulation (73) allows obtaining dissipative equilibrium solutions only. For the isotropic damage model presented in Section 3.2, f_α^{elas} depends on $\delta\lambda^{k+1}$ as:

$$f_\alpha^{\mathrm{elas}} = f_\alpha^{\mathrm{elas}}(\delta\lambda^{k+1}) = \tilde{\epsilon}_\alpha(\delta\lambda^{k+1}) - \kappa_{\alpha,n} \tag{74}$$

with:

$$\tilde{\epsilon}_\alpha(\delta\lambda^{k+1}) = \tilde{\epsilon}_\alpha(\boldsymbol{\epsilon}_\alpha^{k+1}) \tag{75}$$

and:

$$\boldsymbol{\epsilon}_\alpha^{k+1} = \boldsymbol{\epsilon}_\alpha^{k+1}(\delta\lambda^{k+1}) = \boldsymbol{\epsilon}_{\alpha,n} + \Delta\boldsymbol{\epsilon}_\alpha^k + \delta\lambda^{k+1}\delta\boldsymbol{\epsilon}_{\alpha,\mathrm{I}}^{k+1} + \delta\boldsymbol{\epsilon}_{\alpha,\mathrm{II}}^{k+1} \tag{76}$$

Linearization. Function f_α^{elas} is convex because f_α is convex. Moreover, since the "max" function is convex, the constraint (73) is also convex. As a result, it may admit up to two distinct real roots (complex roots are inadmissible). However, their direct determination is often impossible, because f_α^{elas} is a non-linear function of $\delta\lambda^{k+1}$ and the "max" operator adds further non-linearity.

Function f_α^{elas} can be linearized based on the solution at iteration k (i.e., $\delta\lambda = 0$) as:

$$g_\alpha^{\mathrm{lin}}(\delta\lambda_\alpha^{k+1}) = f_\alpha^{\mathrm{elas}}\Big|_{\delta\lambda=0} + \left.\frac{\mathrm{d}f_\alpha^{\mathrm{elas}}}{\mathrm{d}\delta\lambda}\right|_{\delta\lambda=0} \delta\lambda_\alpha^{k+1} \tag{77}$$

[6] For a detailed demonstration the interested reader can refer to [15].

The linear equation to be solved at each integration point thus reads as in (64) with:

$$a_{\alpha,0} = f_\alpha^{\text{elas}}\Big|_{\delta\lambda=0} \tag{78}$$

$$a_{\alpha,1} = \frac{\mathrm{d}f_\alpha^{\text{elas}}}{\mathrm{d}\delta\lambda}\Big|_{\delta\lambda=0} \tag{79}$$

Remark 10. As proved in [15], controlling the maximum elastic predictor of the damage criterion function allows for avoiding artificial elastic unloading since only dissipative solutions are selected. However, this constraint equation is strongly model-dependent through (78) and (79).

3.4.3. *Control by maximal internal variable increment, CIVI (dissipative, local, model-dependent)*

A constraint equation to control the maximum variation of the internal/history variables of the constitutive model was proposed by Rastiello *et al.* [22]. These authors focused on a discrete strong-discontinuity model developed in the E-FEM. The internal/history variables of the cohesive model describing the response of the embedded discontinuity were controlled through the path-following equation. Here, this method is employed for constraining the variation of the internal/history variables of the isotropic damage model of Section 3.2. Similar developments can be done for other constitutive models (e.g., isotropic/anisotropic damage laws, plasticity models, etc.).

Controlling the history variable κ (CIVI-κ). A constraint equation for controlling the variation of the history variable (κ) of the damage model can be written as:

$$p = \max_{\alpha\in\mathcal{G}}(g_\alpha) - \Delta\tau = \max_{\alpha\in\mathcal{G}}(\Delta\kappa_\alpha^{k+1}) - \Delta\tau = 0, \quad \Delta\tau > 0 \tag{80}$$

To impose this constraint at the integration point level and keep the algorithm dissipative, Rastiello *et al.* [22] exploited the fact that for the integration point α such that:

$$\Delta\kappa_\alpha^{k+1} = \Delta\kappa_\alpha^{k} + \delta\kappa_\alpha^{k+1} = \Delta\tau \tag{81}$$

fulfilling the KKT conditions implies that:

$$\tilde{f}_\alpha = \tilde{f}_\alpha(\delta\lambda_\alpha^{k+1}, \delta\kappa_\alpha^{k+1}) = f_\alpha^{k+1}(\tilde{\epsilon}_\alpha(\delta\lambda^{k+1}), \kappa_{\alpha,n} + \Delta\kappa_\alpha^{k} + \delta\kappa_\alpha^{k+1}) = 0 \tag{82}$$

For applying the "nested intervals algorithm", function $\tilde{f}_\alpha$ is linearized in the vicinity of the solution at $(\delta\lambda, \delta\kappa) = (0,0)$. Enforcing condition (81), the problem to be solved at each integration point consists in finding $\delta\lambda_\alpha^{k+1}$ such that:

$$\tilde{f}_\alpha^{\text{lin}}(\delta\lambda_\alpha^{k+1}) = \tilde{f}_\alpha\big|_{(\delta\lambda=0,\delta\kappa=0)} + \frac{\partial\tilde{f}_\alpha}{\partial\delta\lambda}\Big|_{(\delta\lambda=0,\delta\kappa=0)}\delta\lambda_\alpha^{k+1} + \frac{\partial\tilde{f}_\alpha}{\partial\delta\kappa}\Big|_{(\delta\lambda=0,\delta\kappa=0)}(\Delta\tau - \Delta\kappa_\alpha^{k}) = 0 \tag{83}$$

The integration point-wise linearized constraint equation finally reads as in (64) with:

$$a_{\alpha,0} = \Delta\kappa_\alpha^{k} - \frac{\tilde{f}_\alpha|_{(\delta\lambda=0,\delta\kappa=0)}}{\frac{\partial\tilde{f}_\alpha}{\partial\delta\kappa}\Big|_{(\delta\lambda=0,\delta\kappa=0)}} \tag{84}$$

$$a_{\alpha,1} = -\frac{\frac{\partial\tilde{f}_\alpha}{\partial\delta\lambda}\Big|_{(\delta\lambda=0,\delta\kappa=0)}}{\frac{\partial\tilde{f}_\alpha}{\partial\delta\kappa}\Big|_{(\delta\lambda=0,\delta\kappa=0)}} \tag{85}$$

Controlling the damage variable D (CIVI-D). Similarly to (80), a constraint equation on the maximum damage variation in the domain can also be written as:

$$p = \max_{\alpha\in\mathcal{G}}(g_\alpha) - \Delta\tau = \max_{\alpha\in\mathcal{G}}(\Delta D_\alpha^{k+1}) - \Delta\tau = 0, \quad \Delta\tau \in (0,1) \tag{86}$$

The discretized KKT conditions and the incremental damage constraint now read:

$$\tilde{f}_\alpha(\delta\lambda_\alpha^{k+1}, \delta D_\alpha^{k+1}) = f_\alpha^{k+1}(\tilde{\epsilon}_\alpha(\delta\lambda^{k+1}), r^{-1}(D_n + \Delta D_\alpha^{k+1})) = 0 \tag{87}$$

where r^{-1} is the inverse of the damage evolution function (34) and:

$$\Delta D_\alpha^{k+1} = \Delta D_\alpha^k + \delta D_\alpha^{k+1} = \Delta\tau \tag{88}$$

Linearizing the damage criterion function around the solution at $(\delta\lambda, \delta D) = (0,0)$ and enforcing condition (88) one has:

$$\tilde{f}_\alpha^{\text{lin}}(\delta\lambda_\alpha^{k+1}) = \tilde{f}_\alpha\big|_{(\delta\lambda=0,\delta D=0)} + \left.\frac{\partial\tilde{f}_\alpha}{\partial\delta\lambda}\right|_{(\delta\lambda=0,\delta D=0)} \delta\lambda_\alpha^{k+1} + \left.\frac{\partial\tilde{f}_\alpha}{\partial\delta\kappa}\frac{\mathrm{d}\delta\kappa}{\mathrm{d}\delta D}\right|_{(\delta\lambda=0,\delta D=0)} (\Delta\tau - \Delta D_\alpha^k) = 0 \tag{89}$$

Finally, coefficients appearing in (64) read:

$$a_{\alpha,0} = \Delta D_\alpha^k - \frac{\tilde{f}_\alpha\big|_{(\delta\lambda=0,\delta D=0)}}{\left.\frac{\partial\tilde{f}_\alpha}{\partial\delta\kappa}\frac{\mathrm{d}\delta\kappa}{\mathrm{d}\delta D}\right|_{(\delta\lambda=0,\delta D=0)}} \tag{90}$$

$$a_{\alpha,1} = -\frac{\left.\frac{\partial\tilde{f}_\alpha}{\partial\delta\lambda}\right|_{(\delta\lambda=0,\delta D=0)}}{\left.\frac{\partial\tilde{f}_\alpha}{\partial\delta\kappa}\frac{\mathrm{d}\delta\kappa}{\mathrm{d}\delta D}\right|_{(\delta\lambda=0,\delta D=0)}} \tag{91}$$

Remark 11. This kind of path-following constraint is dissipative by nature. A positive variation of κ implies an increase in the damage variable. This ensures that dissipation occurs in the system. As a counterpart of this, however, such constraint equations are strongly model-dependent since the final expressions of coefficients (84), (85), (90), and (91) depend on the chosen damage activation function and evolution model.

4. Numerical studies

A comparison between the formulations presented before is established considering simple test cases characterized by damage localization. The simulations are performed using a partitioned solver to test differentiable and non-differentiable constraint equations. Moreover, the stiffness operator **K** is not updated throughout iterations, but it is kept equal to the elastic stiffness matrix. This choice only affects the convergence rate of the iterative solver but does not influence the studied path-following methods. Notice that, in a general case, convergence (stopping) criteria on the force residual and constraint equation can be used (see, e.g., [21]). In the examples illustrated in the following sections, however, convergence is mainly controlled by the equilibrium problem.

4.1. *Beam under tension*

A 2D beam under uni-axial indirectly-controlled loading is modeled. The bar is weakened in the middle to induce damage. In that case, it is well known that a snap-back in the overall force-displacement response occurs if the damaging element is small enough compared to the size of the structure.

4.1.1. *Geometry and boundary conditions*

The geometry and the boundary conditions are depicted in Figure 2. A mesh containing 29 linear quadrilateral FEs is considered. Indirect tensile loading is imposed on the right side of the beam, whereas the left side is constrained. Strain localization is induced by assuming that only the central (14th) FE is damageable, whereas other FEs are considered to behave according to an isotropic linear elastic law. Material properties are provided in the caption of Figure 2. Plane stress conditions are considered.

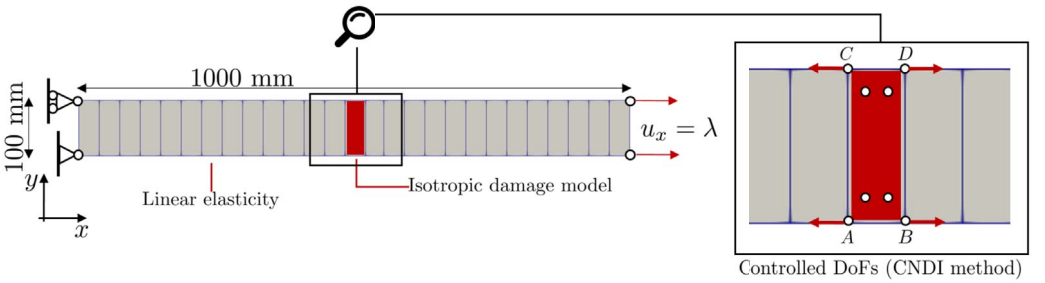

Figure 2. Beam under tension—Geometry, FE mesh, and boundary conditions. Material parameters used in computations: Young's modulus, $E = 1 \times 10^9$ Pa; Poisson's ratio, $\nu = 0$ (this is a purely arbitrary choice; totally similar results can be obtained by considering $\nu \neq 0$); Damage threshold, $\kappa_0 = 10^{-4}$; Dissipated energy parameter, $\beta = 10^4$. Thickness = 1000 mm.

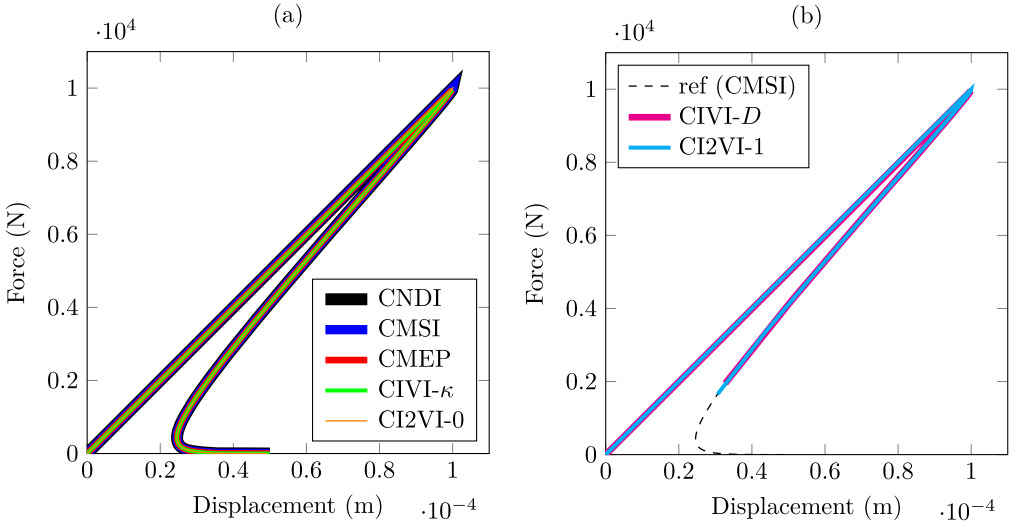

Figure 3. Beam under tension—Comparison between the global responses obtained using different path-following constraint equations. $\Delta\tau$ is taken equal to 3.45×10^{-7} m for the CNDI formulation, to 10^{-5} for the CMSI, CMEP and CIVI-κ formulations, to 0.3449×10^{-7} m^2 for the CI2VI-0 and CI2VI-1 formulations, and to 0.9516×10^{-1} for the CIVI-D formulation.

4.1.2. *Results*

A comparison between the overall structural responses obtained with all the constraint equations introduced previously is given in Figure 3. For the sake of comparison, $\Delta\tau$ values are chosen such that the horizontal strain variation ($\Delta\epsilon_{xx}$) of the damaged element at the onset of damage equals $\kappa_0/10 = 10^{-5}$ for each constraint equation. Moreover, the number of loading steps is adapted for each formulation to obtain the same final beam elongation ($u_L = u_x(x = L) = 0.5 \times 10^{-4}$ m).

Global responses. Most of the constraint equations allow the reproduction of the whole force-displacement response (Figure 3a). The sole exceptions are the CIVI-D and CI2VI-1 (i.e., the CI2VI constraint with $\phi = 1$) formulations (Figure 3b). In particular, the CIVI-D formulation does not

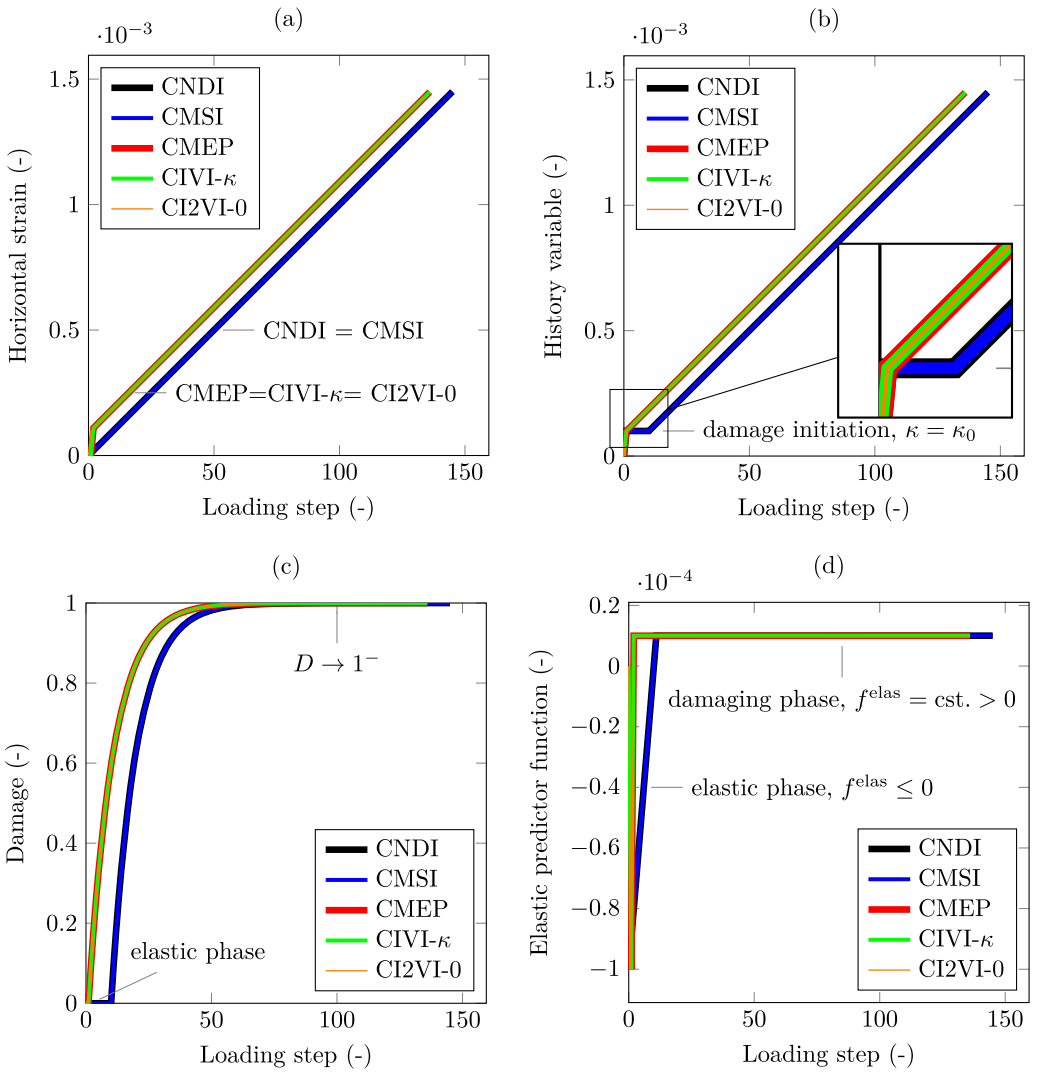

Figure 4. Beam under tension—CNDI, CMSI, CMEP, CIVI-κ, and CI2VI-0 formulations—Pseudo-time evolution of the horizontal strain (a), history variable (b), damage variable (c), and elastic predictor function (d).

enable converging after the 10th loading step ($u_L \approx 0.3 \times 10^{-4}$ m). At this step, D becomes larger than $1 - \Delta\tau$. As a consequence, imposing further damage increases is no more possible.

Constraint equations. The constraint equation is fulfilled for all the methods. As expected (Figure 4), the CNDI, CMSI, CMEP, CIVI-κ, and CI2VI-0 (i.e., the CI2VI constraint with $\phi = 0$) formulations provide the same results. The horizontal strain in the damageable FE increases linearly (Figure 4a), whereas the damage variable evolves non-linearly (Figure 4c). Given the considered geometrical and loading conditions, the CMSI, CMEP, and CIVI-κ methods allow controlling the horizontal strain variation of the damageable FE (the same strain value is computed at each of the four FE integration points). Similarly, the CNDI formulation prescribes the average value of the relative horizontal displacements between nodes (A, B) and (C, D), which is equal to the horizontal strain variation multiplied by the element size in the horizontal direction (L/n_{el}).

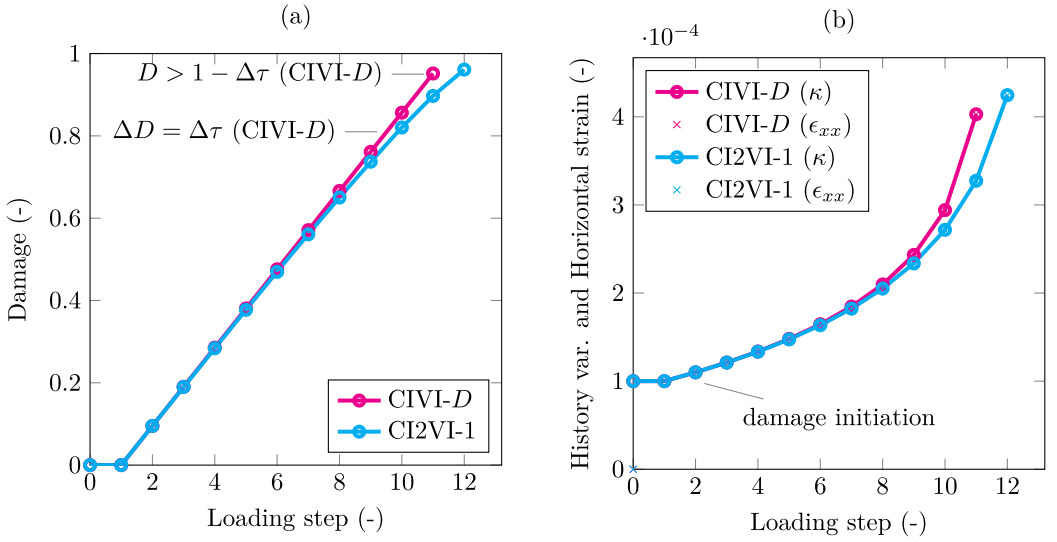

Figure 5. Beam under tension—CIVI-D and CI2VI-1 formulations—Pseudo-time evolution of the damage variable (a), history variable, and horizontal strain (b).

Finally, the CI2VI-0 formulation controls the horizontal strain variation multiplied by the area of the FE.

The main difference between these formulations is that the CNDI and CMSI are non-dissipative, whereas the CIVI-κ, CI2VI-0, and CMEP constraints are dissipative. For that reason, these latter constraints do not allow describing the elastic material response (i.e., one jumps from the very first loading step to a state close to the load peak without computing intermediate states). This is also illustrated by the initial phase when κ stays constant and equals κ_0 (Figure 4b) for the CNDI and the CMSI formulations. During this phase, f^{elas} takes negative values (Figure 4d); then, it stays equal to $\Delta\tau$ as for the other constraint equations during the damaging phase.

Slightly different considerations hold for the CIVI-D and CI2VI-1 formulations. With the CIVI-D method, D increases linearly (Figure 5a) as prescribed by the constraint equation. In parallel, κ and ϵ_{xx} rise exponentially (Figure 5b) with a clear tendency toward a vertical asymptotic growth at the end of the simulation. As already mentioned, the simulation stops when D reaches a value such that imposing additional damage increases without violating the admissibility condition $D \leq 1$ (i.e., the damage evolution model) becomes impossible.

Finally, concerning the CI2VI-1 formulation (Figure 6), the loss of convergence observed at the 12th time step (Figure 5) results from a competition between terms $1 - D_n$ and $\Delta\kappa$ figuring in the constraint equation. To ensure that dissipation occurs during the whole loading process, $1 - D_n$ has to decrease, whereas $\Delta\kappa$ increases. Since $\Delta\kappa$ rises exponentially, D_n becomes closer and closer to one, with the risk of exceeding such an upper bound. This issue can be alleviated by decreasing $\Delta\tau$, thus advancing more in the simulation. However, completing the calculation in a reasonable number of steps becomes very difficult.

4.2. *Multi-holed plate under tension*

The second test case simulates indirectly controlled tensile loading on a 2D plate with four holes aligned horizontally. Plane stress conditions are assumed. Given the considered boundary conditions (Figure 7), damage propagates from one side to the other of the plate crossing the

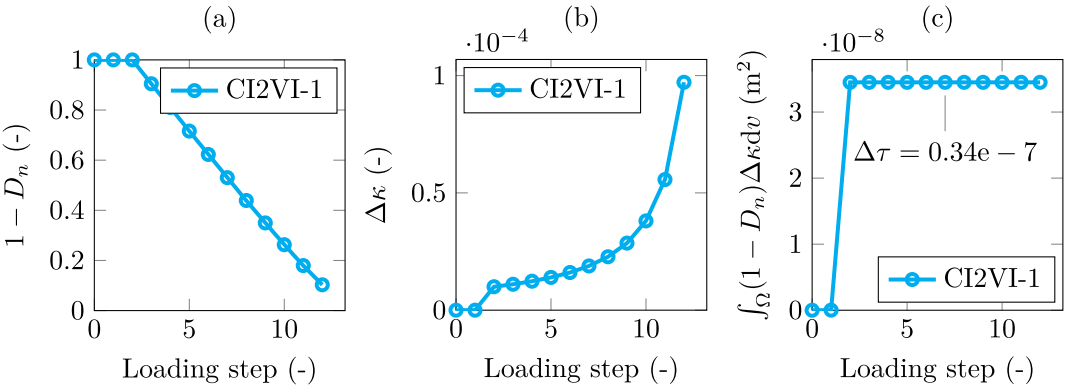

Figure 6. Beam under tension—CI2VI-1 formulation—Pseudo-time evolution of the terms appearing in the constraint equation: $1 - D_n$ (a), $\Delta\kappa$ (b), prescribed quantity (c).

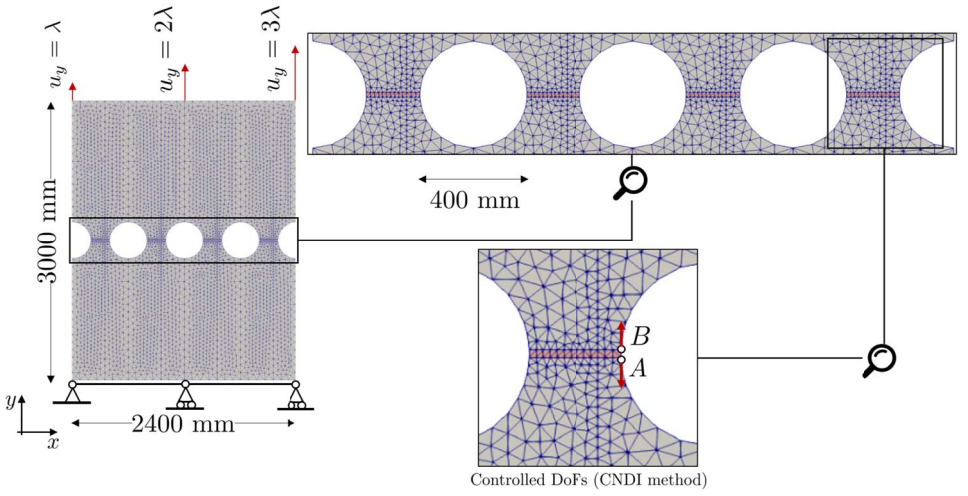

Figure 7. Holed plate under tension—Geometry, FE mesh, and boundary conditions. Material parameters used in computations: Young's modulus, $E = 1 \times 10^9$ Pa; Poisson's ratio, $\nu = 0$; Damage threshold, $\kappa_0 = 10^{-4}$; Dissipated energy parameter, $\beta = 10^4$. Thickness = 1000 mm.

holes. An unstable structural response characterized by multiple snap-backs is obtained. Here, for the sake of simplicity, the simulation is stopped before damage propagation between the last two holes (at this point, the residual structural resistance is already very low). As a consequence, only three snap-backs are described. A similar test case was considered, for instance, in [14, 21, 43, 53, 54].

4.2.1. *Geometry and boundary conditions*

Geometry and boundary conditions of the considered problem are given in Figure 7. The FE mesh used for these simulations contains about 15000 linear triangular FEs. Damage is allowed

to occur in the central horizontal line of FEs, where the isotropic damage model of Section 3.2 is used. A linear elastic material behavior is assumed for the other FEs. To compare results, $\Delta\tau$ values are chosen such that the maximum damage variation on the domain is equal to 0.1 (approximately) at the first damaging step for each constraint equation. Alternatively, one could have used an adaptive refinement strategy for modifying $\Delta\tau$ during the computation [13,20,21,54, 55]. As an example, one could have computed the path step-length as $\Delta\tau = \Delta\tau^{\text{ref}}/(i^{\text{restart}} \times n^{\text{opt}})$, where n^{opt} is the target number of iterations and i^{restart} is the number of restarting inside the time step. Alternatively, one could have used the relationship $\Delta\tau = \Delta\tau^{\text{ref}}/((n^{\text{opt}})^{i^{\text{restart}}})$.

4.2.2. *Results*

The main differences between dissipative and non-dissipative constraints are first shown. The CMSI and CIVI-*D* formulations are used for this comparison, but similar considerations hold for other constraint equations. Then, a comparison between the overall structural responses obtained with all the constraint equations is established to show the robustness of the resulting solvers.

Non-dissipative constraints. Figures 8a and 8b represent the global force-displacement response obtained using the CMSI constraint equation. The force corresponds to the total reaction force at the locations where displacements are imposed. The displacement is the vertical displacement registered at the upper right corner of the mesh.

As expected, three snap-back phases characterize the structural response. Each of them corresponds to the damage propagation between two adjacent holes, from the right to the left of the plate (Figure 9). Each snap-back is followed by a pseudo-elastic reloading phase (the force increases without damage evolution). During these phases, the strain in the damaged FEs increases without energy dissipation until damage occurs on the other side of the hole. As shown in Figure 9, the deformation process can be defined as "accordion-like". Once one ligament is entirely damaged, the displacement increases until damage start on the other side of the hole. The displacement then reduces to accommodate a controlled constant strain variation in the subsequent ligament. The external displacement progressively decreases during damage propagation until the ligament is fully damaged. Then, this process is repeated till the following ligament starts damaging.

The non-dissipative nature of the constraint equation is even more evident when observing the pseudo-time evolution of some representative quantities throughout the simulation (Figure 10). Figures 10a and 10b provide the evolution of the reaction force and displacement respectively, whereas Figure 10c gives the evolution of the maximum strain variation ($\max_{\alpha\in\mathcal{G}}(\mathbf{q}_\alpha^\top \Delta\boldsymbol{\epsilon}_\alpha)$) on the computational domain. Finally, Figure 10d provides the evolution of the maximum damage increase on the domain.

As expected (Figure 10c), the constraint equation is strictly fulfilled throughout the entire simulation, the maximum strain variation being constant and equal to the prescribed quantity $\Delta\tau$. Conversely, the maximum damage variation evolves during the simulation, taking positive values when dissipation is in progress and null values when elastic reloading occurs in the system. As illustrated in Figure 10d, these phases correspond to the first part of the reloading branches just after the different snap-backs. The oscillations observed in the damage variation evolution of Figure 10d correspond to the progressive damage propagation through adjacent FEs. They reflect the fact that during the same loading step, one FE can become fully damaged, and the adjacent one can start damaging. This effect can be strongly reduced by decreasing the loading parameter $\Delta\tau$.

Dissipative constraints. To illustrate the response of dissipative methods, the CIVI-*D* formulation is chosen. As expected, the non-dissipative equilibrium branches between two successive

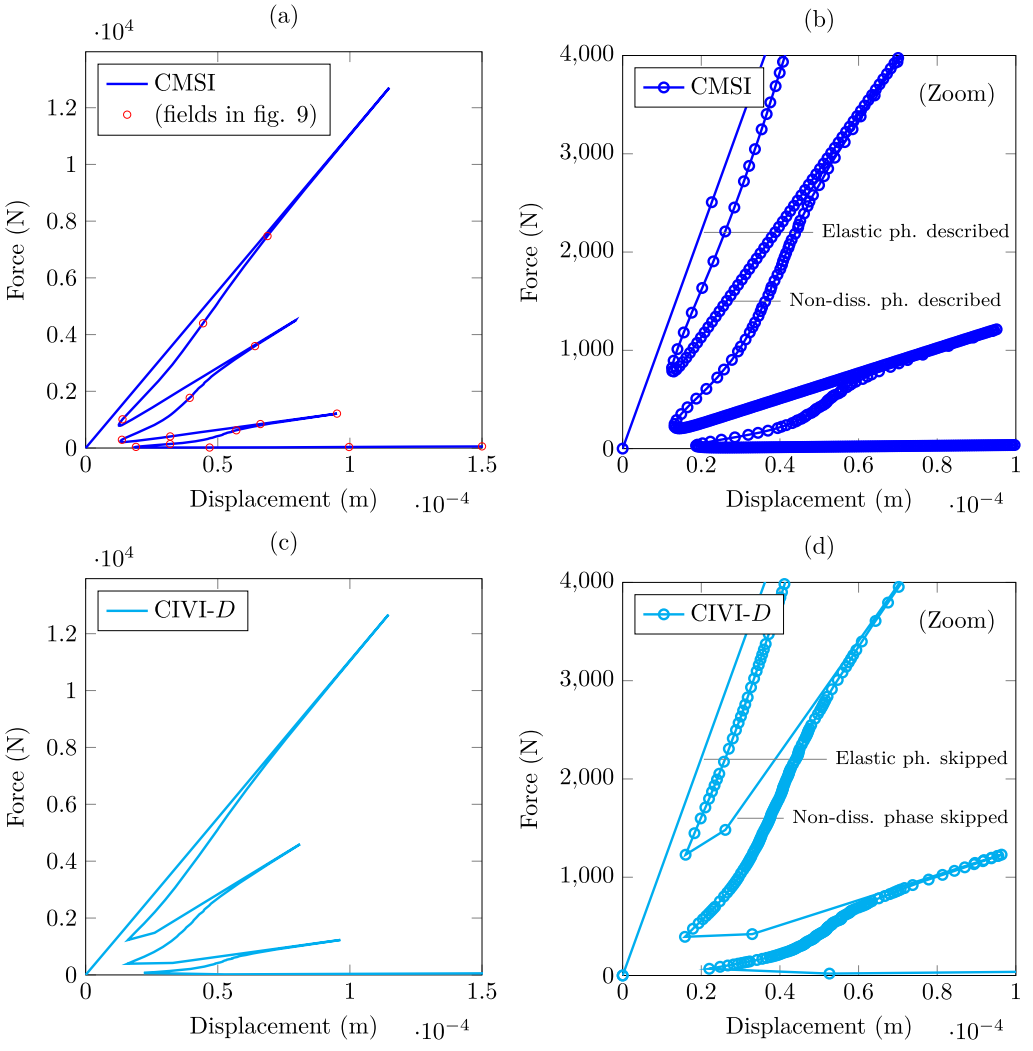

Figure 8. Holed plate under tension—CMSI (a,b) and CIVI-D (c,d) formulations—Global responses and zoom on specific parts of the force-displacement curves (Force-displacement states indicated with red circles in Figure 8a correspond to the damage fields of Figure 9).

snap-backs are skipped (Figures 8c and 8d). In other words, one jumps from a dissipative state to another dissipative state. This is also clear from the results depicted in Figure 11. The maximum damage variation in the system stays constant and equal to $\Delta\tau$ throughout the simulation, whereas the maximum strain variation (computed, once again, as $\max_{\alpha\in\mathcal{G}}(\mathbf{q}_\alpha^\top \Delta\boldsymbol{\epsilon}_\alpha)$) evolves to ensure that dissipation occurs in the system.

Comparison of constraint equations. The structural responses obtained using all the constraint equations described in Section 3 are in good agreement (Figure 12). The only exception is the one achieved using the CNDI formulation due to the well-known difficulty of choosing *a priori* the DoFs to control during the simulation. Here, the CNDI constraint was written considering the variation of the relative vertical displacement $\Delta d_{y,\mathrm{AB}}$ between nodes A and B (see Figure 7).

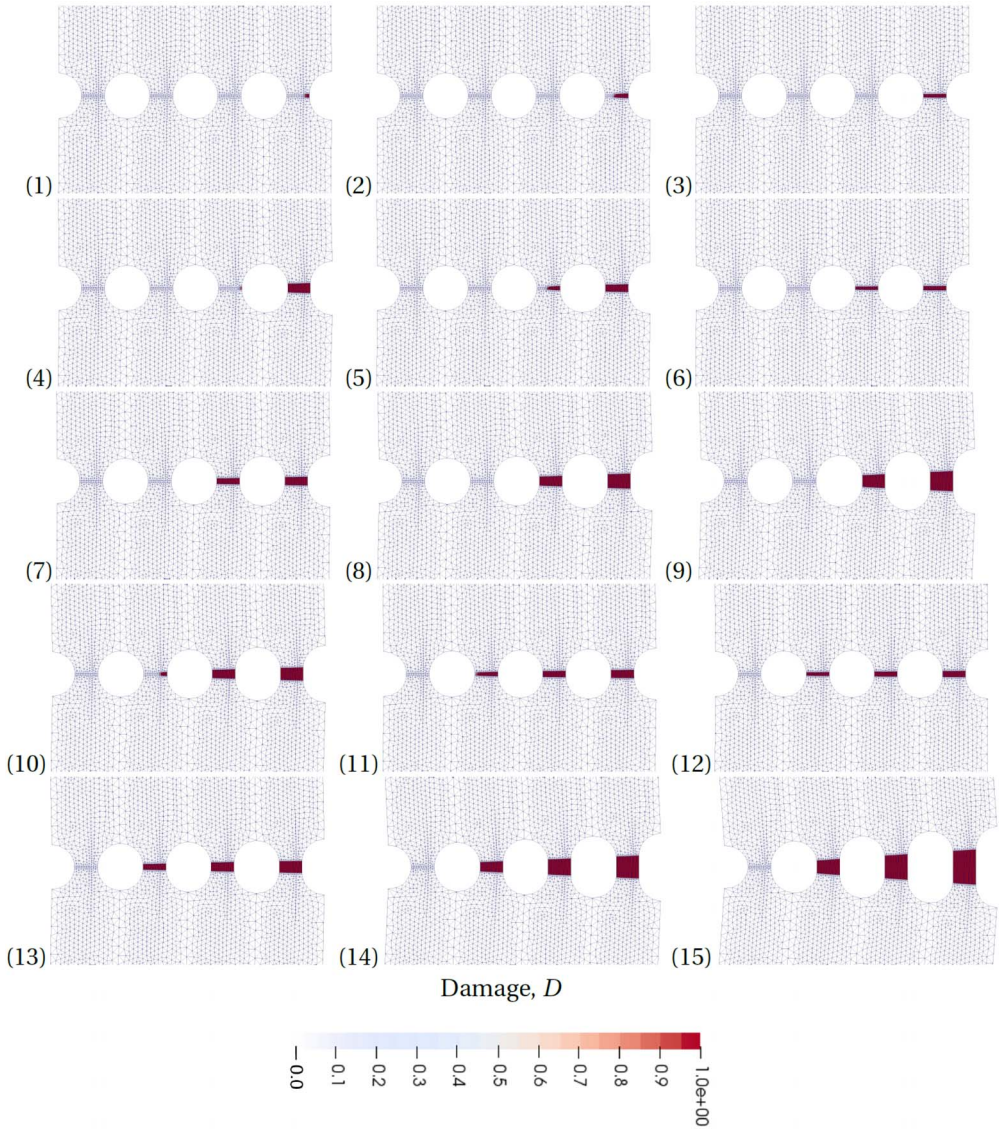

Figure 9. Holed plate under tension—CMSI formulation—"Accordion-like" deformation process. Damage field for 15 steps of the simulation (red circles in Figure 8a).

Given this arbitrary choice, the snap-back phases are not properly described. Indeed, imposing a positive variation of $\Delta d_{y,\mathrm{AB}}$ implies that the strain at this location can never decrease to accommodate a gradual damage growth (as in Figure 9). As a result, damage develops abruptly between holes.

All the other constraint equations allow describing a gradual damaging process. The main difference between them concerns the description of pseudo-elastic reloading branches. They can be conveniently reproduced using non-dissipative formulations only. This is not the case with dissipative methods; in that case, one moves directly from the end of one snap-back phase to a state close to the subsequent local load peak corresponding to the onset of damage on the other side of the hole. This can be seen as a drawback of dissipative methods; however, this also

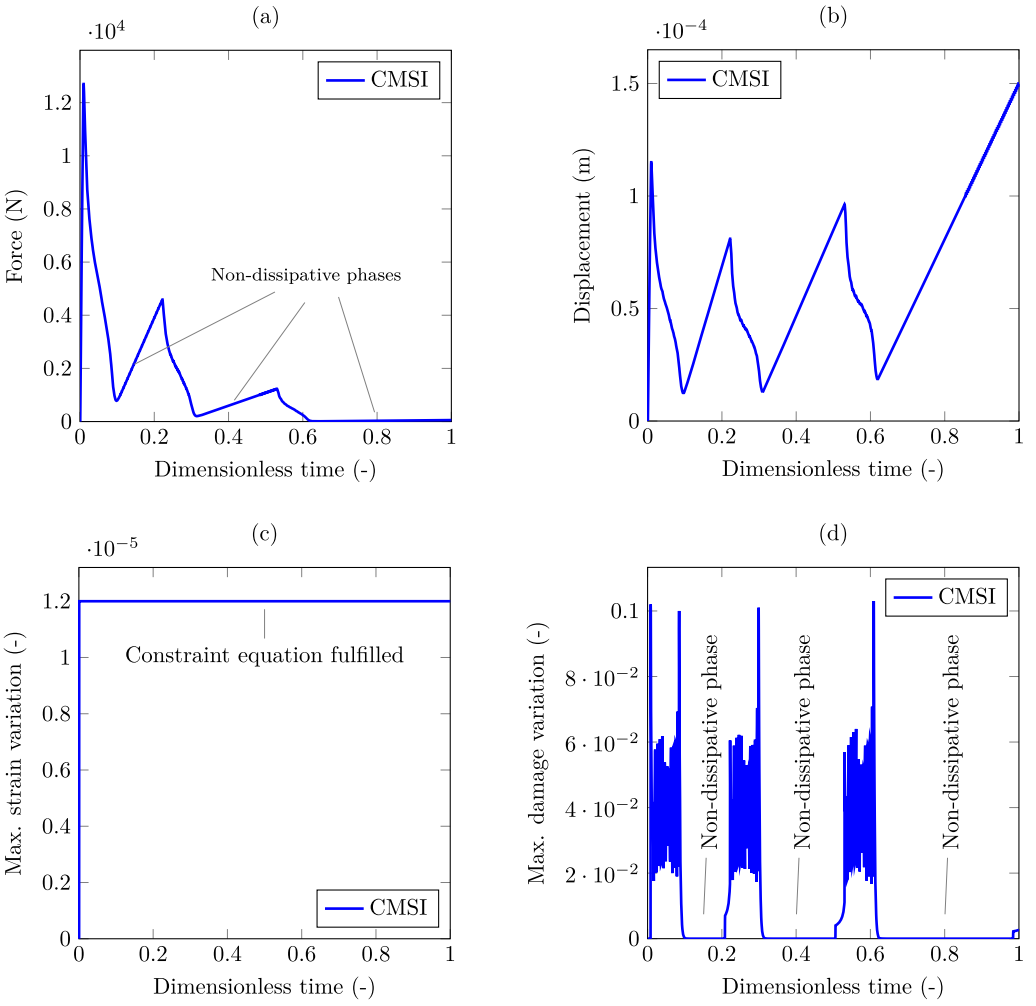

Figure 10. Holed plate under tension—CMSI formulation—Pseudo-time evolution of the reaction force (a), displacement (b), maximum strain variation ($\max_{\alpha\in\mathcal{G}}(\mathbf{q}_\alpha^\top \Delta\boldsymbol{\epsilon}_\alpha)$) (c), and maximum damage variation (d) on the structure.

means that the computational effort associated with the description of elastic reloading phases is drastically reduced.

5. Conclusions and perspectives

Path-following strategies for describing unstable structural responses induced by material strain-softening were discussed. These methods are based on the assumption that the external loading (either imposed displacements or tractions) is split into two independent contributions. The first one is known (in direction and intensity); the second one is unknown in intensity, the sole direction being prescribed by the user. To compute this additional unknown, the load factor, the equilibrium equation is supplemented by the path-following constraint equation.

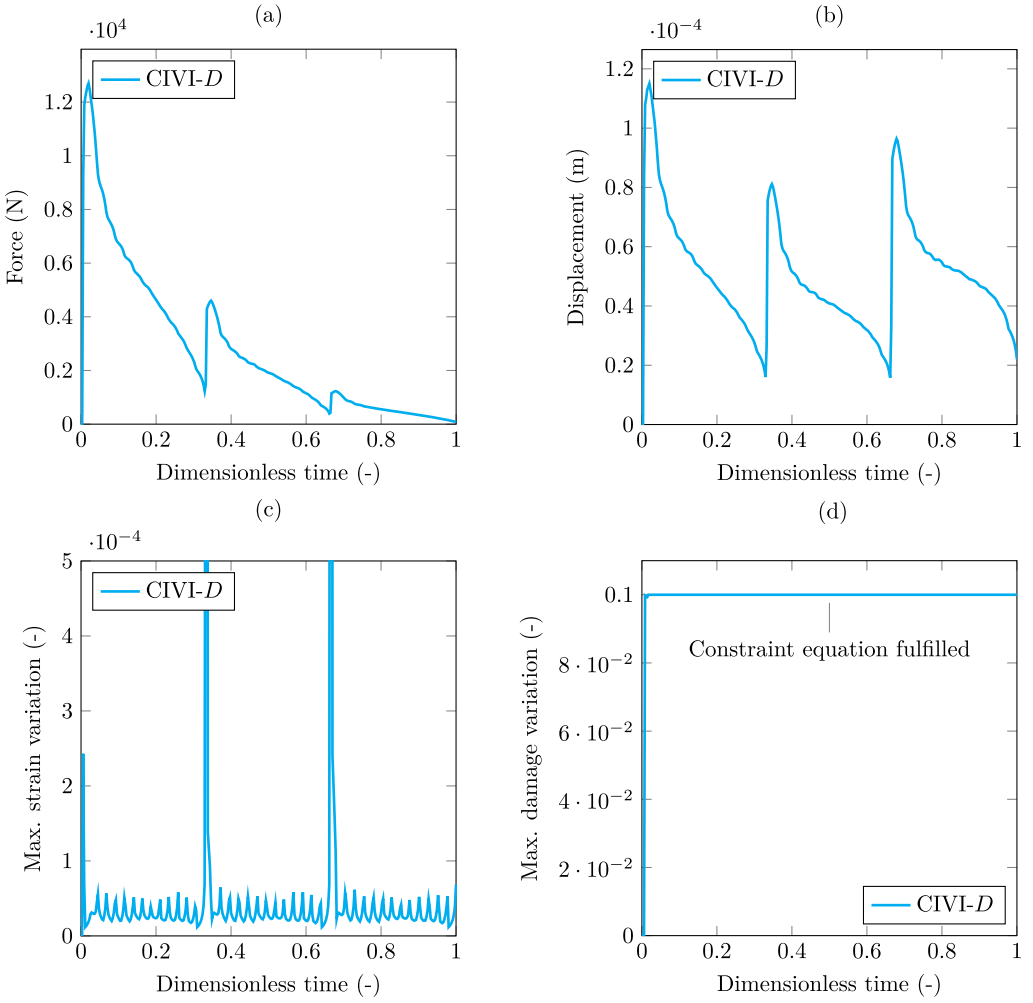

Figure 11. Holed plate under tension—CIVI-D formulation—Pseudo-time evolution of the reaction force (a), displacement (b), maximum strain variation ($\max_{\alpha\in\mathcal{G}}(\mathbf{q}_\alpha^\top \Delta\boldsymbol{\epsilon}_\alpha)$) (c), and maximum damage variation (d) on the structure.

The resulting augmented equilibrium problem can be solved using monolithic or partitioned solution methods. The monolithic formulation is the most natural choice but leads to a non-symmetric saddle-point problem, destroying the banded nature of the stiffness matrix. Moreover, it relies on the assumption that the constraint equation is differentiable with respect to the unknown displacement vector and load factor. Path-following constraint equations of this kind can be written for material non-linearities (several of them are presented in the paper). However, when modeling localized phenomena such as cracking, non-differentiable constraint equations prescribing the maximum of a field of interest on the computational domain are often preferred. In that case, only partitioned solving strategies can be used.

These considerations provided the first criterion to propose a classification of constraint equations. Other criteria were their global/local nature (depending on whether the whole problem unknowns or just a selection of them is considered in the constraint equation), their dissipative/non-dissipative nature, and their dependence on the chosen constitutive model.

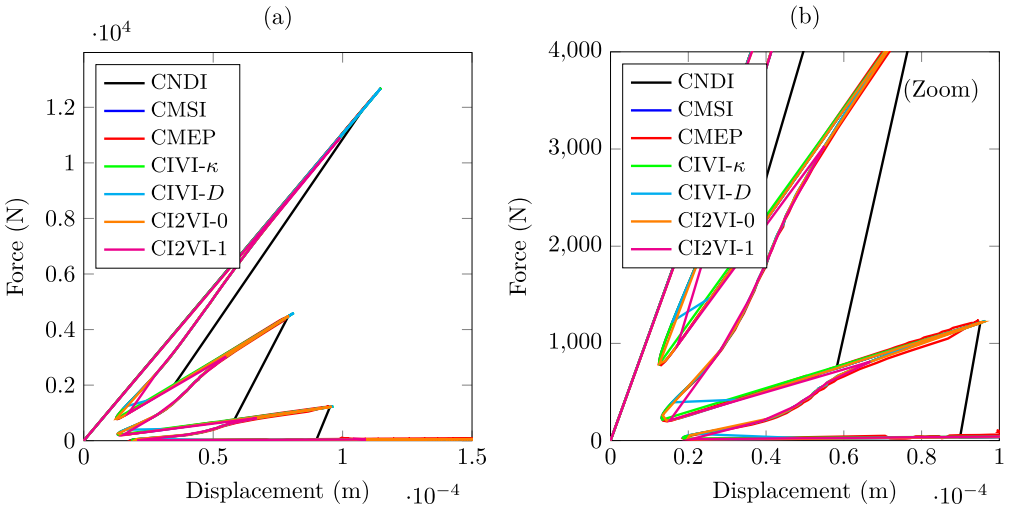

Figure 12. Holed plate under tension—CNDI, CMSI, CMEP, CIVI-κ, CIVI-D, CI2VI-0, and CI2VI-1 formulations—Global responses (a) and zoom close to some parts of the force-displacement curves (b).

Based on this classification, several constraint equations were detailed and tested. Attention was focused on: a constraint equation on the combination of DoFs (CNDI); a constraint equation on the integral on the computational domain of the variation (between two successive pseudo-time steps) of the history variable of the constitutive model (two versions: CI2VI-0 and CI2VI-1); a constraint on the maximum variation of a scalar measure of the strain field (CMSI); a constraint equation on the maximum value of the elastic predictor of the damage criterion function (CMEP); a constraint equation on the maximum variation of the history/internal variables of the constitutive model (two versions: CIVI-κ and CIVI-D). For each of them, the numerical implementation has been discussed according to both partitioned and monolithic (when possible) solution methods.

5.1. *Conclusions*

The purpose of the present paper was to illustrate different path-following methods and detail their implementation in the FEM context. Although proposing a generally valid criterion for choosing one formulation over the others was out of our scope, some conclusions can be drawn based on the numerical studies illustrated previously.

Choice of the constraint equation. An unambiguous and generally valid criterion to choose a constraint equation rather than another cannot be formulated; the choice is strictly related to the simulated problems and to the method used to represent material degradation (e.g., damage mechanics models, strong-discontinuity formulations, etc.).

In our numerical tests, except for the CNDI method, all the constraints allowed to represent the essential phases of the considered structural behaviors.

The general (already known) conclusion that can be drawn is that utilizing a constraint equation controlling the combination of a restricted number of DoFs chosen *a priori* [9] is not always a compelling choice. In most cases, the DoFs controlling the dissipative structural response change during the simulation and cannot be predicted before running the computation. Moreover, finding a solution is not guaranteed [13, 15], except in a few straightforward cases (even adapting the

controlled DoFs throughout the simulation),[7] since abrupt variations of the internal variables of the constitutive model may induce convergence issues and eventually non-convergence.

More "advanced" constraint equations (e.g., those based on maximizing fields of interest on the domain and/or containing information on the constitutive model) all made it possible to obtain satisfactory solutions. For the CMSI method, this is probably due to the chosen examples. The advantage of this constraint equation is its generality since, once implemented in the FE software, it can be used with any constitutive model. However, in general situations, the solving algorithm does not ensure finding dissipative solutions (elastic unloading-reloading paths can be found [22]). Dissipative constraint equations (CMEP, CIVI-D, CIVI-κ, CI2VI-0, and CI2VI-1) allow the selection of dissipative equilibrium states only. Still, they are generally more complex to implement and require ad-hoc developments depending on the constitutive assumptions.

In conclusion, choosing a path-following constraint equation should be made case-by-case, also based on the implementation and computational[8] costs that the user considers acceptable.

Concerning dissipative formulations, those based on maximizing the damage/history variable variation (CIVI-D and CIVI-κ) or the elastic predictor function (CMEP) are certainly more computationally expensive than the others (CI2VI-0 and CI2VI-1) since an additional iterative process is needed to solve the constraint equation. However, they allow a more straightforward interpretation of the physical significance of the path step-length $\Delta\tau$. Conversely, in the CI2VI-0 and CIV2VI-1 methods, the load factor calculation is simpler (no maximization problem to solve), but the choice of $\Delta\tau$ is less obvious (see next paragraph).

A final consideration can be made concerning the boundedness of the quantity (e.g., the damage variable) used to write the constraint equation. A substantial limitation of the CIVI-D and CI2VI-1 methods is that when the damage tends to unity, fulfilling the constraint equation with a given $\Delta\tau$ becomes impossible without violating the damage admissibility condition ($D \leq 1$) somewhere in the domain. The same consideration holds for any other constraint equation written in terms of bounded quantities. Formulations of this kind should thus be avoided. Instead, one should prefer controlling unbounded quantities, as in the CIVI-κ and CI2VI-0 formulations, where damage growth is indirectly driven through the history variable figuring in the damage evolution model.

Choice of the parameter $\Delta\tau$. Choosing the loading parameter $\Delta\tau$ is a challenging task in path-following methods. In the absence of adaptive refinement algorithms (as in our case), a choice can be made based on physical considerations. However, it is sometimes tough to provide a physical interpretation of this parameter. In particular, this is relatively easy for the CNDI, CMSI, CMEP, CIVI-D, and CIVI-κ formulations but is less evident for the CI2VI-0 and CI2VI-1 methods. For instance, the maximum increase of a strain measure (for the CMSI, CMEP, and CIVI-κ methods, given the chosen constitutive model) can be selected depending on the parameters of the damage model to ensure a "smooth" damage evolution. For the CNDI method, interpreting $\Delta\tau$ is even more straightforward; it can be chosen, for instance, to exactly reproduce experimental loading conditions (e.g., to impose the same rate of variation of the Crack Mouth Opening

[7] An example is the notched beam three-point bending test. In this case, controlling the simulation through the CMOD is an effective choice (see, e.g., [39]). Moreover, it may lead to a "smoother" damage evolution compared to a direct displacement control method.

[8] Concerning CPU time, Oliveria *et al.* [39] observed for a three-point bending test (without snap-back, to have a reference solution obtained through a standard incremental solver with direct displacement control) that the number of iterations to converge is lower when using path-following methods than when directly imposing displacements. In other words, the additional computational effort associated with the path-following algorithm can be partly justified by an accelerated convergence of the solving algorithm.

Displacement as in experiments). Concerning the CI2VI-0 and CI2VI-1 formulations, the difficulty of choosing $\Delta\tau$ is because the size of the computational domain directly intervenes in the constraint equation (via integration over Ω). $\Delta\tau$ becomes, in this case, a purely numerical parameter, and adaptive algorithms (see, e.g., [55]) could enormously help improve the algorithmic convergence.

5.2. *Perspectives*

An exhaustive analysis of the strengths and weaknesses of all the formulations presented has not been done in this article. Such a work will undoubtedly require more computations after having defined ad-hoc test cases for "stressing" each constraint equation. Performance assessment will also be performed after having introduced well-suited performance indicators. This is an essential point. As an example, Geers [56] studied performances through the number of iterations to achieve convergence, whereas Fayezioghani *et al.* [21] utilized several performance indicators (robustness, speed, accuracy, and smoothness) combined into an objective performance measure. Restarting methods (automatic adaption of $\Delta\tau$) will also be tested. Finally, hybrid dissipative/non-dissipative algorithms will be studied.

Conflicts of interest

Authors have no conflicts of interest to declare.

Acknowledgements

G. Rastiello was supported by the SEISM Institute (http://www.institut-seism.fr). 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.

Appendix A. Link between monolithic and partitioned methods

A link between the monolithic and partitioned formulations can be established by using the Sherman–Morrison formula [57].

Sherman–Morrison formula. Consider a square matrix $\mathbf{L} \in \mathbb{R}^{q\times q}$ such that it can be expressed as the sum of a invertible square matrix $\mathbf{M} \in \mathbb{R}^{q\times q}$ and the scalar product $\mathbf{v}\mathbf{w}^\top \in \mathbb{R}^{q\times q}$ of two vectors $\mathbf{v} \in \mathbb{R}^q$ and $\mathbf{w} \in \mathbb{R}^q$. $\mathbf{L}$ is invertible if $1 + \mathbf{w}^\top\mathbf{M}^{-1}\mathbf{v} \neq 0$ and its inverse $\mathbf{L}^{-1}$ reads:

$$\mathbf{L}^{-1} = (\mathbf{M} + \mathbf{v}\mathbf{w}^\top)^{-1} = \mathbf{M}^{-1} - \frac{\mathbf{M}^{-1}\mathbf{v}\mathbf{w}^\top\mathbf{M}^{-1}}{1 + \mathbf{w}^\top\mathbf{M}^{-1}\mathbf{v}} \tag{92}$$

Monolithic vs. partitioned formulations. Two different strategies can be followed to establish a link between the monolithic and partitioned formulations:

(i) *Derivation 1.* Let us start by multiplying the first equation of system (12) by w, and the second equation by $\hat{\mathbf{f}}$. After adding them, one obtains:

$$(w\mathbf{K} + \hat{\mathbf{f}}\mathbf{h}^\top)\delta\mathbf{d}^{k+1} = w\mathbf{r}^k - \hat{\mathbf{f}}p^k \tag{93}$$

Applying the Shermann–Morrison formula to matrix $(w\mathbf{K} + \hat{\mathbf{f}}\mathbf{h}^\top)$ gives:

$$(w\mathbf{K} + \hat{\mathbf{f}}\mathbf{h}^\top)^{-1} = \frac{\mathbf{K}^{-1}}{w} - \frac{\mathbf{K}^{-1}\hat{\mathbf{f}}\mathbf{h}^\top\mathbf{K}^{-1}}{w(w + \mathbf{h}^\top\mathbf{K}^{-1}\hat{\mathbf{f}})} \tag{94}$$

After a few mathematical manipulations, the displacement correction can be written as:

$$\begin{aligned}\delta\mathbf{d}^{k+1} &= \mathbf{K}^{-1}\mathbf{r}^k - \frac{(\mathbf{K}^{-1}\hat{\mathbf{f}})\,p^k}{w} - \frac{w(\mathbf{K}^{-1}\hat{\mathbf{f}})\,\mathbf{h}^\top(\mathbf{K}^{-1}\mathbf{r}^k)}{w(w+\mathbf{h}^\top(\mathbf{K}^{-1}\hat{\mathbf{f}}))} + \frac{(\mathbf{K}^{-1}\hat{\mathbf{f}})\,\mathbf{h}^\top(\mathbf{K}^{-1}\hat{\mathbf{f}})\,p^k}{w(w+\mathbf{h}^\top(\mathbf{K}^{-1}\hat{\mathbf{f}}))}\\ &= \delta\mathbf{d}_{\mathrm{I}}^{k+1} - \frac{\delta\mathbf{d}_{\mathrm{II}}^{k+1}p^k}{w} - \frac{\delta\mathbf{d}_{\mathrm{II}}^{k+1}\mathbf{h}^\top\delta\mathbf{d}_{\mathrm{I}}^{k+1}}{w+\mathbf{h}^\top\delta\mathbf{d}_{\mathrm{II}}^{k+1}} + \frac{\delta\mathbf{d}_{\mathrm{II}}^{k+1}\mathbf{h}^\top\delta\mathbf{d}_{\mathrm{II}}^{k+1}p^k}{w(w+\mathbf{h}^\top\delta\mathbf{d}_{\mathrm{II}}^{k+1})}\\ &= \delta\mathbf{d}_{\mathrm{I}}^{k+1} - \frac{1}{\mathbf{h}^\top\delta\mathbf{d}_{\mathrm{II}}^{k+1}+w}(\mathbf{h}^\top\delta\mathbf{d}_{\mathrm{I}}^{k+1}+p^k)\delta\mathbf{d}_{\mathrm{II}}^{k+1}\end{aligned} \tag{95}$$

In (95) it is also easy to recognize the expression of $\delta\lambda^{k+1}$ given in (22). The solution correction vector can thus be written in the form (23).

(ii) *Derivation 2.* Alternatively one could have chosen to follow the derivation illustrated by May *et al.* [43]. In that case, one starts by rewriting the augmented matrix operator as:

$$\tilde{\mathbf{K}} = \underbrace{\tilde{\mathbf{K}}_0 - \mathbf{v}_1\mathbf{w}_1^\top}_{:=\tilde{\mathbf{K}}_1} \underbrace{-\,\mathbf{v}_2\mathbf{w}_2^\top}_{:=\tilde{\mathbf{K}}_2} \tag{96}$$

where:

$$\tilde{\mathbf{K}}_0 = \begin{bmatrix}\mathbf{K} & \mathbf{0}\\ \mathbf{0}^\top & 1\end{bmatrix} \quad \mathbf{v}_1 = \begin{Bmatrix}\hat{\mathbf{f}}\\ 0\end{Bmatrix} \quad \mathbf{w}_1 = \begin{Bmatrix}\mathbf{0}\\ 1\end{Bmatrix} \quad \mathbf{v}_2 = \begin{Bmatrix}\mathbf{0}\\ -1\end{Bmatrix} \quad \mathbf{w}_2 = \begin{Bmatrix}\mathbf{h}\\ w-1\end{Bmatrix} \tag{97}$$

The inverse of matrix $\tilde{\mathbf{K}}$ can be computed by applying the Sherman–Morrison formula twice; the first time to express $\tilde{\mathbf{K}}^{-1}$ as a function of $\tilde{\mathbf{K}}_1^{-1}$:

$$\tilde{\mathbf{K}}^{-1} = (\tilde{\mathbf{K}}_1 - \tilde{\mathbf{K}}_2)^{-1} = \tilde{\mathbf{K}}_1^{-1} + \frac{\tilde{\mathbf{K}}_1^{-1}\mathbf{v}_2\mathbf{w}_2^\top\tilde{\mathbf{K}}_1^{-1}}{1-\mathbf{w}_2\tilde{\mathbf{K}}_1^{-1}\mathbf{v}_2} \tag{98}$$

and the second to express $\tilde{\mathbf{K}}_1^{-1}$ as a function of $\tilde{\mathbf{K}}_0^{-1}$:

$$\tilde{\mathbf{K}}_1^{-1} = (\tilde{\mathbf{K}}_0 - \mathbf{v}_1\mathbf{w}_1^\top)^{-1} = \tilde{\mathbf{K}}_0^{-1} + \frac{\tilde{\mathbf{K}}_0^{-1}\mathbf{v}_1\mathbf{w}_1^\top\tilde{\mathbf{K}}_0^{-1}}{1-\mathbf{w}_1\tilde{\mathbf{K}}_0^{-1}\mathbf{v}_1} \tag{99}$$

After some mathematical manipulations, one obtains:

$$\tilde{\mathbf{K}}^{-1} = \begin{bmatrix}\mathbf{K} & -\hat{\mathbf{f}}\\ \mathbf{h}^\top & w\end{bmatrix}^{-1} = \begin{bmatrix}\mathbf{K}^{-1} & \mathbf{0}\\ \mathbf{0}^\top & 1\end{bmatrix} + \frac{1}{w+\mathbf{h}^\top\mathbf{K}^{-1}\hat{\mathbf{f}}}\begin{bmatrix}-\mathbf{K}^{-1}\hat{\mathbf{f}}\mathbf{h}^\top\mathbf{K}^{-1} & \mathbf{K}^{-1}\hat{\mathbf{f}}\\ -\mathbf{h}^\top\mathbf{K}^{-1} & -(\mathbf{h}^\top\mathbf{K}^{-1}\hat{\mathbf{f}}+w-1)\end{bmatrix} \tag{100}$$

$$= \begin{bmatrix}\mathbf{K}^{-1} & \mathbf{0}\\ \mathbf{0}^\top & 1\end{bmatrix} + \frac{1}{w+\mathbf{h}^\top\delta\mathbf{d}_{\mathrm{II}}^{k+1}}\begin{bmatrix}-\delta\mathbf{d}_{\mathrm{II}}^{k+1}\mathbf{h}^\top\mathbf{K}^{-1} & \delta\mathbf{d}_{\mathrm{II}}^{k+1}\\ -\mathbf{h}^\top\mathbf{K}^{-1} & -(\mathbf{h}^\top\delta\mathbf{d}_{\mathrm{II}}^{k+1}+w-1)\end{bmatrix} \tag{101}$$

which after multiplication by $\{\mathbf{r}^k, -p^k\}^\top$ gives (23) or, equivalently:

$$\begin{Bmatrix}\delta\mathbf{d}^{k+1}\\ \delta\lambda^{k+1}\end{Bmatrix} = \begin{Bmatrix}\delta\mathbf{d}_{\mathrm{I}}^{k+1}\\ -p^k\end{Bmatrix} + \frac{1}{w+\mathbf{h}^\top\delta\mathbf{d}_{\mathrm{II}}^{k+1}}\begin{Bmatrix}-(\mathbf{h}^\top\delta\mathbf{d}_{\mathrm{I}}^{k+1}+p^k)\delta\mathbf{d}_{\mathrm{II}}^{k+1}\\ -\mathbf{h}^\top\delta\mathbf{d}_{\mathrm{I}}^{k+1} - p^k(1-\mathbf{h}^\top\delta\mathbf{d}_{\mathrm{II}}^{k+1}-w)\end{Bmatrix} \tag{102}$$

This way of expressing the solution corresponds to the one reported by [20]. Notice, however, that (102) is less attractive than (23) from a computational viewpoint since it implies more operations (one additional dot product and two multiplications).

Appendix B. Extension to multi-parametric loading

The extension of path-following methods to multi-parametric loading can be easily done by writing the boundary conditions as:

$$\mathbf{u} = \mathbf{u}_{\text{imp}} + \sum_{i=1}^{m} \lambda_i \hat{\mathbf{u}}_i \quad \forall \mathbf{x} \in \partial\Omega_u \tag{103}$$

$$\boldsymbol{\sigma}(\mathbf{u}) \cdot \mathbf{n} = \mathbf{t}_{\text{imp}} + \sum_{i=1}^{m} \lambda_i \hat{\mathbf{t}}_i \quad \forall \mathbf{x} \in \partial\Omega_t \tag{104}$$

Since m load factors $(\lambda_1, \lambda_2, \ldots, \lambda_m)$ are introduced, m path-following equations need to be prescribed:

$$p_i(\mathbf{u}, \lambda_1, \lambda_2, \ldots, \lambda_m) = 0 \quad i = 1, \ldots, m \tag{105}$$

The resulting linearized equilibrium problem reads [5]:

$$\begin{bmatrix} \mathbf{K} & -\hat{\mathbf{f}}_1 & \ldots & -\hat{\mathbf{f}}_m \\ \mathbf{h}_1^\top & w_{11} & \ldots & w_{1m} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{h}_m^\top & w_{m1} & \ldots & w_{mm} \end{bmatrix} \begin{Bmatrix} \delta\mathbf{d}^{k+1} \\ \delta\lambda_1^{k+1} \\ \vdots \\ \delta\lambda_m^{k+1} \end{Bmatrix} = \begin{Bmatrix} \mathbf{r}^k \\ -p_1^k \\ \vdots \\ -p_m^k \end{Bmatrix} \tag{106}$$

where $\mathbf{K}$ is the standard stiffness matrix and:

$$\mathbf{h}_i = \frac{\partial p_i}{\partial \mathbf{d}} \tag{107}$$

$$w_{ij} = \frac{\partial p_i}{\partial \lambda_j} \quad i = 1, \ldots, m; j = 1, \ldots, m \tag{108}$$

$$\mathbf{r}^k = \mathbf{r}(\mathbf{d}^k, \lambda_1^k, \ldots, \lambda_m^k) = \mathbf{f}_{\text{imp}}^{\text{ext}} + \sum_{i=1}^{m} \lambda_i^k \hat{\mathbf{f}}_i - \mathbf{f}^{\text{int}}(\mathbf{d}^k) \tag{109}$$

$$p_i^k = p_i(\mathbf{d}^k, \lambda_1^k, \ldots, \lambda_m^k) \quad i = 1, \ldots, m \tag{110}$$

System (106) can be solved in a monolithic fashion. Alternatively, the partitioned formulation can be easily derived. In that case, the displacement correction is written as the sum of $m+1$ contributions as:

$$\delta\mathbf{d}^{k+1} = \delta\mathbf{d}_{\text{I}}^{k+1} + \sum_{i=1}^{m} \delta\lambda_i^{k+1} \delta\mathbf{d}_{\text{II},i}^{k+1} \tag{111}$$

where $\delta\mathbf{d}_{\text{I}}^{k+1}$ is computed as in (21) and vectors $\delta\mathbf{d}_{\text{II},i}^{k+1}$ can be calculated as:

$$\delta\mathbf{d}_{\text{II},i}^{k+1} = \mathbf{K}^{-1}\hat{\mathbf{f}}_i \quad i = 1, \ldots, m \tag{112}$$

Then, the constraint equations can be solved:

$$\begin{aligned} &p_1(\delta\lambda_1^{k+1}, \delta\lambda_2^{k+1}, \ldots, \delta\lambda_m^{k+1}; \delta\mathbf{d}_{\text{I}}^{k+1}, \delta\mathbf{d}_{\text{II},1}^{k+1}, \delta\mathbf{d}_{\text{II},2}^{k+1}, \ldots, \delta\mathbf{d}_{\text{II},m}^{k+1}) = 0 \\ &\ldots \\ &p_m(\delta\lambda_1^{k+1}, \delta\lambda_2^{k+1}, \ldots, \delta\lambda_m^{k+1}; \delta\mathbf{d}_{\text{I}}^{k+1}, \delta\mathbf{d}_{\text{II},1}^{k+1}, \delta\mathbf{d}_{\text{II},2}^{k+1}, \ldots, \delta\mathbf{d}_{\text{II},m}^{k+1}) = 0 \end{aligned} \tag{113}$$

References

[1] G. A. Wempner, "Discrete approximations related to nonlinear theories of solids", *Int. J. Solids Struct.* **7** (1971), no. 11, p. 1581-1599.

[2] E. Riks, "The application of Newton's method to the problem of elastic stability", *J. Appl. Mech.* **39** (1972), no. 4, p. 1060-1065.

[3] E. Riks, "An incremental approach to the solution of snapping and buckling problems", *Int. J. Solids Struct.* **15** (1979), no. 7, p. 529-551.

[4] M. Crisfield, "An arc-length method including line searches and accelerations", *Int. J. Numer. Methods Eng.* **19** (1983), no. 9, p. 1269-1289.

[5] S. Skatulla, C. Sansour, "On a path-following method for non-linear solid mechanics with applications to structural and cardiac mechanics subject to arbitrary loading scenarios", *Int. J. Solids Struct.* **96** (2016), p. 181-191.
[6] K. Schweizerhof, P. Wriggers, "Consistent linearization for path following methods in nonlinear fe analysis", *Comput. Methods Appl. Mech. Eng.* **59** (1986), no. 3, p. 261-279.
[7] E. Ramm, "Strategies for tracing the nonlinear response near limit points", in *Nonlinear Finite Element Analysis in Structural Mechanics* (W. Wunderlich, E. Stein, K.-J. Bathe, eds.), Springer, Berlin, Heidelberg, 1981, p. 63-89.
[8] M. Ritto-Corrêa, D. Camotim, "On the arc-length and other quadratic control methods: Established, less known and new implementation procedures", *Comput. Struct.* **86** (2008), no. 11, p. 1353-1368.
[9] R. De Borst, "Computation of post-bifurcation and post-failure behavior of strain-softening solids", *Comput. Struct.* **25** (1987), no. 2, p. 211-224.
[10] R. De Borst, M. A. Crisfield, J. J. C. Remmers, C. V. Verhoosel, *Nonlinear Finite Element Analysis of Solids and Structures*, John Wiley & Sons, 2012.
[11] J. Napoleão, F. A. Elwi, D. Murray, "An eigenvector-based strategy for the analysis of inelastic structures", *Comput. Struct.* **42** (1992), no. 5, p. 833-848.
[12] Z. Chen, H. L. Schreyer, "A numerical solution scheme for softening problems involving total strain control", *Comput. Struct.* **37** (1990), no. 6, p. 1043-1050.
[13] M. G. D. Geers, "Enhanced solution control for physically and geometrically non-linear problems. Part I—the subplane control approach", *Int. J. Numer. Methods Eng.* **46** (1999), no. 2, p. 177-204.
[14] T. Pohl, E. Ramm, M. Bischoff, "Adaptive path following schemes for problems with softening", *Finite Elem. Anal. Des.* **86** (2014), p. 12-22.
[15] E. Lorentz, P. Badel, "A new path-following constraint for strain-softening finite element simulations", *Int. J. Numer. Methods Eng.* **60** (2004), no. 2, p. 499-526.
[16] N. Singh, C. Verhoosel, R. de Borst, E. van Brummelen, "A fracture-controlled path-following technique for phase-field modeling of brittle fracture", *Finite Elem. Anal. Des.* **113** (2016), p. 14-29.
[17] A. Stanić, B. Brank, "A path-following method for elasto-plastic solids and structures based on control of plastic dissipation and plastic work", *Finite Elem. Anal. Des.* **123** (2017), p. 1-8.
[18] E. Barbieri, F. Ongaro, 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", *Comput. Methods Appl. Mech. Eng.* **315** (2017), p. 713-743.
[19] M. A. Gutiérrez, "Energy release control for numerical simulations of failure in quasi-brittle solids", *Commun. Numer. Methods Eng.* **20** (2004), no. 1, p. 19-29.
[20] C. V. Verhoosel, J. J. C. Remmers, M. A. Gutiérrez, "A dissipation-based arc-length method for robust simulation of brittle and ductile failure", *Int. J. Numer. Methods Eng.* **77** (2009), no. 9, p. 1290-1321.
[21] A. Fayezioghani, B. Vandoren, L. Sluys, "A posteriori performance-based comparison of three new path-following constraints for damage analysis of quasi-brittle materials", *Comput. Methods Appl. Mech. Eng.* **346** (2019), p. 746-768.
[22] G. Rastiello, F. Riccardi, B. Richard, "Discontinuity-scale path-following methods for the embedded discontinuity finite element modeling of failure in solids", *Comput. Methods Appl. Mech. Eng.* **349** (2019), p. 431-457.
[23] G. Rastiello, C. Giry, F. Gatuingt, R. Desmorat, "From diffuse damage to strain localization from an Eikonal Non-Local (ENL) continuum damage model with evolving internal length", *Comput. Methods Appl. Mech. Eng.* **331** (2018), p. 650-674.
[24] F. Thierry, G. Rastiello, C. Giry, F. Gatuingt, "One-dimensional Eikonal Non-Local (ENL) damage models: Influence of the integration rule for computing interaction distances and indirect loading control on damage localization", *Mech. Res. Commun.* **110** (2020), article no. 103620.
[25] K. Moreau, N. Moës, N. Chevaugeon, A. Salzman, "Concurrent development of local and non-local damage with the Thick Level Set approach: Implementation aspects and application to quasi-brittle failure", *Comput. Methods Appl. Mech. Eng.* **327** (2017), p. 306-326.
[26] G. Alfano, M. A. Crisfield, "Solution strategies for the delamination analysis based on a combination of local-control arc-length and line searches", *Int. J. Numer. Methods Eng.* **58** (2003), no. 7, p. 999-1048.
[27] D. Bellora, R. Vescovini, "Hybrid geometric-dissipative arc-length methods for the quasi-static analysis of delamination problems", *Comput. Struct.* **175** (2016), p. 123-133.
[28] P. Massin, G. Ferté, A. Caron, N. Moës, "Pilotage du chargement en formulation X-FEM : application aux lois cohésives", in *10e colloque national en calcul des structures*, 2011, p. Clé-USB.
[29] J. Oliver, A. Huespe, J. Cante, "An implicit/explicit integration scheme to increase computability of non-linear material and contact/friction problems", *Comput. Methods Appl. Mech. Eng.* **197** (2008), no. 21, p. 1865-1889.
[30] B. Brank, A. Stanić, A. Ibrahimbegovic, "A path-following method based on plastic dissipation control", in *Computational Methods for Solids and Fluids*, Springer, Cham, 2016, p. 29-47.
[31] F. Cazes, G. Meschke, M.-M. Zhou, "Strong discontinuity approaches: An algorithm for robust performance and comparative assessment of accuracy", *Int. J. Solids Struct.* **96** (2016), p. 355-379.

[32] A. Brencich, A. Carpinteri, "Interaction of a main crack with ordered distributions of microcracks: a numerical technique by displacement discontinuity boundary elements", *Int. J. Fract.* **76** (1996), no. 4, p. 373-389.
[33] A. Carpinteri, I. Monetto, "Snap-back analysis of fracture evolution in multi-cracked solids using boundary element method", *Int. J. Fract.* **98** (1999), no. 3, p. 225-241.
[34] H. L. Oliveira, G. Rastiello, A. Millard, "Partitioned path-following strategy for nonlinear structural analyses using the boundary element method", *Comput. Methods Appl. Mech. Eng.* **394** (2022), article no. 114875.
[35] J.-L. Batoz, G. Dhatt, "Incremental displacement algorithms for nonlinear problems", *Int. J. Numer. Methods Eng.* **14** (1979), no. 8, p. 1262-1267.
[36] M. Bonnet, A. Frangi, C. Rey, *The Finite Element Method in Solid Mechanics*, McGraw Hill Education, New York, 2014.
[37] I. Babuška, "The finite element method with Lagrangian multipliers", *Numer. Math.* **20** (1973), no. 3, p. 179-192.
[38] B. Richard, G. Rastiello, C. Giry, F. Riccardi, R. Paredes, E. Zafati, S. Kakarla, C. Lejouad, "CastLab: an object-oriented finite element toolbox within the Matlab environment for educational and research purposes in computational solid mechanics", *Adv. Eng. Softw.* **128** (2019), p. 136-151.
[39] H. L. Oliveira, G. Rastiello, A. Millard, I. Bitar, B. Richard, "Modular implementation framework of partitioned path-following strategies: Formulation, algorithms and application to the finite element software Cast3M", *Adv. Eng. Softw.* **161** (2021), article no. 103055.
[40] P. Verpeaux, A. Millard, T. Charras, A. Combescure, "A modern approach of large computer codes for structural analysis", in *IASMiRT, Proc. SMiRT 10 Conf.*, Anaheim, CA, USA, 1989.
[41] J. Pellet, "Dualisation of the boundary conditions", 2011, Code_Aster Open Source-General FEA Software.
[42] P. Pegon, A. Anthoine, "Numerical strategies for solving continuum damage problems with softening: application to the homogenization of masonry", *Comput. Struct.* **64** (1997), no. 1–4, p. 623-642.
[43] S. May, J. Vignollet, R. de Borst, "A new arc-length control method based on the rates of the internal and the dissipated energy", *Eng. Comput.* **33** (2016), no. 1, p. 100-115.
[44] Z. Chen, H. Schreyer, "A numerical solution scheme for softening problems involving total strain control", *Comput. Struct.* **37** (1990), no. 6, p. 1043-1050.
[45] J. Lemaitre, R. Desmorat, *Engineering Damage Mechanics: Ductile, Creep, Fatigue and Brittle Failures*, Springer-Verlag, Berlin, Heidelberg, 2005.
[46] N. Moës, N. Chevaugeon, "Lipschitz regularization for softening material models: the Lip-field approach", *C. R. Mécanique* **349** (2021), no. 2, p. 415-434.
[47] J. Mazars, "Application de la mécanique de l'endommagement au comportement non linéaire et à la rupture du béton de structure", PhD Thesis, Université Pierre et Marie Curie-PARIS 6, 1984, Thèse de docteur es sciences.
[48] J.-J. Marigo, "Formulation d'une loi d'endommagement d'un materiau élastique", *C. R. Acad. Sci. Sér. II* **292** (1981), no. 19, p. 1309-1312.
[49] Z. Chen, H. Schreyer, "Secant structural solution strategies under element constraint for incremental damage", *Comput. Methods Appl. Mech. Eng.* **90** (1991), no. 1, p. 869-884.
[50] J. G. Rots, R. De Borst, "Analysis of concrete fracture in "direct" tension", *Int. J. Solids Struct.* **25** (1989), no. 12, p. 1381-1394.
[51] G. Rastiello, J.-L. Tailhan, P. Rossi, S. Dal Pont, "Macroscopic probabilistic cracking approach for the numerical modelling of fluid leakage in concrete", *Ann. Solid Struct. Mech.* **7** (2015), no. 1, p. 1-16.
[52] H. B. Hellweg, M. A. Crisfield, "A new arc-length method for handling sharp snap-backs", *Comput. Struct.* **66** (1998), no. 5, p. 704-709.
[53] L. F. Paullo Muñoz, D. Roehl, "A Continuation method with combined restrictions for nonlinear structure analysis", *Finite Elem. Anal. Des.* **130** (2017), p. 53-64.
[54] A. Fayezioghani, B. Vandoren, L. Sluys, "Performance-based step-length adaptation laws for path-following methods", *Comput. Struct.* **223** (2019), article no. 106100.
[55] M. J. Clarke, G. J. Hancock, "A study of incremental-iterative strategies for non-linear analyses", *Int. J. Numer. Methods Eng.* **29** (1990), no. 7, p. 1365-1391.
[56] M. G. D. Geers, "Enhanced solution control for physically and geometrically non-linear problems. Part II—comparative performance analysis", *Int. J. Numer. Methods Eng.* **46** (1999), no. 2, p. 205-230.
[57] J. Sherman, W. J. Morrison, "Adjustment of an inverse matrix corresponding to a change in one element of a given matrix", *Ann. Math. Statist.* **21** (1950), no. 1, p. 124-127.