Available online at www.sciencedirect.com

ScienceDirect

Comput. Methods Appl. Mech. Engrg. 197 (2008) 1865–1889

Computer methods in applied mechanics and engineering

www.elsevier.com/locate/cma

An implicit/explicit integration scheme to increase computability of non-linear material and contact/friction problems

J. Oliver [a,*], A.E. Huespe [c], J.C. Cante [b]

[a] E.T.S. d'Enginyers de Camins, Canals i Ports, Technical University of Catalonia (UPC), Campus Nord UPC, Mòdul C-1, c/Jordi Girona 1-3, 08034 Barcelona, Spain

[b] E.T.S. d'Enginyería Industrial i, Aeronáutica de Terrassa, Technical University of Catalonia (UPC), Campus Nord UPC, Mòdul C-1, c/Jordi Girona 1-3, 08034 Barcelona, Spain

[c] CIMEC/Intec, Conicet, UNL, Guemes 3450, Santa Fe 3000, Argentina

Received 10 September 2007; received in revised form 24 November 2007; accepted 29 November 2007
Available online 8 December 2007

Abstract

An implicit/explicit integration scheme for non-linear constitutive models is presented. It aims at providing additional computability to those solid mechanics problems were robustness is an important issue, i.e. material failure models equipped with strain softening, soft materials, contact-friction models, etc., although it can also provide important advantages, in terms of computational cost, with respect to purely implicit integration schemes. The proposed scheme is presented based on general families of constitutive models (continuum damage and elasto-plasticity) and its properties, in terms of robustness and accuracy, are analytically derived and computationally assessed by means of numerical simulations. An adaptive time stepping algorithm, based on a priori control of the committed error and the application of the proposed scheme to contact/friction interfaces are also presented.

Keywords: Constitutive models integration; Contact/friction; Implicit explicit schemes; Continuum damage; Elasto-plasticity; Robustness; Computability

1. Introduction

Let us consider a typical, displacement driven, material non-linear solid mechanics problem, appropriately discretized in time, $t \in [0, T]$, and space, $\mathbf{x} \in \Omega$, which, after application of the selected time marching algorithm and spatial discretization scheme, at time step $n+1$, reads:

Find:

$$\mathbf{a}_{n+1}; \quad \alpha_{n+1}; \quad \sigma_{n+1},$$

Such that:

$$\mathbf{F}_{\text{int}}(\mathbf{a}_{n+1}, \boldsymbol{\sigma}_{n+1}, t_{n+1}) - \mathbf{F}_{\text{ext}}(t_{n+1}) = \mathbf{G}(\mathbf{a}_{n+1}, \boldsymbol{\sigma}_{n+1}(\mathbf{a}_{n+1}), t_{n+1}) = \mathbf{0} \quad \text{(balance of forces)}, \tag{1}$$

$$g(\alpha_{n+1}, \boldsymbol{\sigma}_{n+1}, t_{n+1}) = 0 \quad \text{(state evolution equation)}, \tag{2}$$

$$\dot{\boldsymbol{\sigma}}_{n+1} \equiv \frac{\boldsymbol{\sigma}_{n+1} - \boldsymbol{\sigma}_n}{\Delta t_{n+1}} = \boldsymbol{\Sigma}(\boldsymbol{\varepsilon}(\mathbf{a}_{n+1}), \alpha_{n+1}, \boldsymbol{\sigma}_{(\cdot)}) \quad \text{(constitutive equation)}, \tag{3}$$

where $\mathbf{a}_{n+1}$ are the nodal displacements, at the end of time step $n+1$, and α_{n+1} and $\boldsymbol{\sigma}_{n+1}$ are, respectively, the strain-like variable and the stresses at the sampling points. Additionally, $\boldsymbol{\varepsilon}_{n+1}$ are the strains, related to the stresses through the (non-linear) constitutive function, $\boldsymbol{\Sigma}$, in rate form in Eq. (3), and $\mathbf{F}_{\text{ext}}$ and $\mathbf{F}_{\text{int}}$ stand, respectively, for the external and internal forces whose balance is established in Eq. (1). Therein t_{n+1} ($t_{n+1} \geqslant 0$, $\Delta t_{n+1} \equiv t_{n+1} - t_n \geqslant 0$) stands for that increasing parameter being either the actual time (as in dynamic problems) or playing the role of time (the pseudo-time identified as the loading factor or the arc length parameter) in quasi-static problems.

* Corresponding author.
E-mail address: xavier.oliver@upc.edu (J. Oliver).

0045-7825/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.cma.2007.11.027

In the context of the theory of dissipative material models equipped with internal variables [1,2], in Eq. (2) function $g(\alpha_{n+1}, \boldsymbol{\sigma}_{n+1}, t_{n+1})$ implicitly defines the current value of these internal variables, α_{n+1}. For *rate-dependent* models, this function can be identified from the time-discretized version of the evolution equations of the internal variables i.e.

$$\dot{\alpha}_{n+1} \equiv \frac{\alpha_{n+1} - \alpha_n}{\Delta t_{n+1}} = h(\alpha_{n+1}, \boldsymbol{\sigma}_{n+1}, t_{n+1}) \quad \text{(evolution equation)}, \tag{4}$$

$$g(\alpha_{n+1}, \boldsymbol{\sigma}_{n+1}, t_{n+1}) \equiv (\alpha_{n+1} - \alpha_n) - (t_{n+1} - t_n) \times h(\alpha_{n+1}, \boldsymbol{\sigma}_{n+1}, t_{n+1}) \quad \text{(state equation)}, \tag{5}$$

whereas in *rate-independent* models it comes out from the combination of the Kuhn–Tucker algorithmic loading/unloading conditions and the evolution equations of the internal variables, typically [2]:

$$\Delta\lambda_{n+1} \geqslant 0; \quad f(\boldsymbol{\sigma}_{n+1}, \alpha_{n+1}) \geqslant 0; \quad \Delta\lambda_{n+1} f(\boldsymbol{\sigma}_{n+1}, \alpha_{n+1}) = 0 \quad \text{(loading/unloading conditions)}, \tag{6}$$

$$\dot{\alpha}_{n+1} \equiv \frac{\alpha_{n+1} - \alpha_n}{\Delta t_{n+1}} = \frac{\Delta\lambda_{n+1}}{\Delta t_{n+1}} \quad \text{(evolution equation)}, \tag{7}$$

$$\begin{cases} \text{unloading} \rightarrow \Delta\lambda_{n+1} = 0 \Rightarrow g(\alpha_{n+1}, \boldsymbol{\sigma}_{n+1}, t_{n+1}) \equiv \alpha_{n+1} - \alpha_n \\ \text{loading} \rightarrow \Delta\lambda_{n+1} \neq 0 \Rightarrow g(\alpha_{n+1}, \boldsymbol{\sigma}_{n+1}, t_{n+1}) \equiv f(\boldsymbol{\sigma}_{n+1}, \alpha_{n+1}) \end{cases} \quad \text{(state equation)}, \tag{8}$$

where $\Delta\lambda_{n+1}$ and $f(\boldsymbol{\sigma}_{n+1}, \alpha_{n+1})$, in Eq. (7), are, respectively, the algorithmic Lagrange multiplier and the restriction defining the closure of the elastic domain in the stress space ($\mathsf{E}_{\boldsymbol{\sigma}_{n+1}} := \{\boldsymbol{\sigma}_{n+1}; f(\boldsymbol{\sigma}_{n+1}, \alpha_{n+1}) \leqslant 0\}$).

Regarding Eq. (3) the specific format of function Σ is determined by the selected algorithm for integration of the material model. Typically, a true dependence of $\Sigma(\mathbf{a}_{n+1}, \alpha_{n+1}, \boldsymbol{\sigma}_{n+1})$ on the values of the stresses at the end of the time step ($\boldsymbol{\sigma}_{(\cdot)} \equiv \boldsymbol{\sigma}_{n+1}$) corresponds to a classical *implicit* (backward-Euler) *integration*, whereas dependence on values at previous time steps ($\boldsymbol{\sigma}_{(\cdot)} = \varphi(\boldsymbol{\sigma}_n, \boldsymbol{\sigma}_{n-1}, \ldots)$ characterizes an *explicit integration* of the material model.

Much has been written in the literature about implicit vs. explicit integration schemes and the advantages and disadvantages of each of them. They can be summarized as follows:

- *Explicit integration schemes* are in many cases conditionally stable. This translates into a limitation of the time step length and, therefore, a large number of time steps are needed to solve the problem. On the other hand, $\boldsymbol{\sigma}_{n+1}$ in Eqs. (1)–(3) becomes, in many cases, linearly or quasi-linearly dependent on the problem unknowns, $\mathbf{a}_{n+1}$. In many cases this translates into a linear or a quasi-linear structure of function $\mathbf{G}$ in Eq. (1), and the global algorithm for its resolution becomes, generally, very robust.
- *Implicit integration schemes* are generally unconditionally stable. Therefore, there is no intrinsic limitation on the length of the time step, other than the control of the integration error, which uses to be small, and the number of required time steps, is small when compared with explicit algorithms. On the other hand, $\boldsymbol{\sigma}_{n+1}$, in Eqs. (1)–(3), uses to be highly non-linear in terms of the main unknowns $\mathbf{a}_{n+1}$. This non-linearity is inherited by Eq. (1) and the resulting solving algorithm (typically a Newton–Raphson iterative procedure) often can be made robust only by using very skillful procedures (namely, continuation methods) and dramatic shortenings of the time step values. In certain cases, for instance when strain softening appears in the constitutive model, the algorithm becomes so ill conditioned that no convergence, and then no result, can be achieved for problems of practical interest.

In summary: *explicit integration schemes* yield robust *but expensive* (in terms of the computational cost) solving algorithms, whereas *implicit integration schemes* lead to *accurate results*, even for large time steps, but at the cost of a *loss of robustness* of the resulting numerical algorithm which, for cases of practical interest, can also dramatically affect the corresponding computational cost.

This work proposes a combination of implicit and explicit integration schemes that exploits the advantages of both, while overcoming some of their drawbacks. In essence, it is a combination of a standard implicit integration scheme of the stresses, $\boldsymbol{\sigma}_{n+1}$, in the constitutive model in Eq. (3) with an explicit extrapolation of the involved internal variables, α_{n+1}, in Eqs. (2)–(3). The proposed implicit/explicit integration scheme, from now on shortened as IMPL-EX, is presented based on two representative families of rate-independent material constitutive models: continuum damage models and elasto-plastic models. However, this does not imply intrinsic restrictions in terms of its application to other families of inelastic constitutive models.

At the cost of few, and simple, additional operations, to be performed at the constitutive driver level, the IMPL-EX algorithm, renders relevant benefits when it is conveniently exploited in computational mechanics. They can be summarized as follows:

- The algorithmic tangent constitutive tensor becomes symmetric and semi-positive definite even in those cases as the analytical one is not. This leads to dramatic improvements of the robustness in problems where implicit integrations result in singularity or the negative character of the algorithmic tangent operators.
- In many cases, the algorithmic tangent constitutive tensor becomes constant. Therefore, in absence of sources of non-linearity other than the constitutive model, the complete non-linear problem reduces to a sequence of linear (at every time step) problems. The classical Newton–Raphson procedure takes a unique iteration to converge and the problem becomes step-linear. The effects on the computational costs are also dramatic.
- The good stability properties of the implicit integration algorithm are inherited by the proposed IMPL-EX integration algorithm.
- The order of accuracy of the IMPL-EX integration algorithm, with respect to the size of the time step, is, at least, linear; the same as many classical backward-Euler

implicit algorithms. Nevertheless, the absolute error is larger for the same time step length.

- The method can be exploited to render robust, and step-linear, complex non-linear problems. For instance: in bilateral contact/friction problems, a strategy based on using an anticipating contact interface mesh, whose elements are equipped with an appropriate constitutive model, to penalize interpenetration and define the friction effects, can be linearized by using an IMPL-EX integration of that constitutive model, resulting in a sequence of linear problems.

Previous works [3] of the authors have displayed the benefits of the IMPL-EX integration scheme when applied to modeling material failure. Here, the theoretical aspects of the method are described in detail and generalized to a broader family of material models, and its accuracy properties are analyzed. Additional applications, as the ones in contact/friction problems, are presented as well.

The remaining of the paper is structured as follows: in Section 2, two target families of, widely representative, non-linear material models (continuum damage and elasto-plasticity) are described, their classical implicit integration schemes are presented and the corresponding properties in terms of robustness are examined. Then, in Section 3, the proposed implicit/explicit integration scheme is presented, its accuracy properties are derived and a corresponding error control method is presented. Section 4 is devoted to the numerical assessment of the proposed integration method via selected representative examples. In Section 5 the method is extended to contact/friction interfaces and the provided additional robustness is assessed. Finally, in Section 6, some final remarks are provided.

2. Representative constitutive models

For the sake of covering a wide range of constitutive models, in the remaining of this paper two families of rate-independent models will be considered: (1) isotropic continuum damage models and (2) elasto-plastic models.

	Isotropic continuum damage model	Elasto-plastic model with isotropic hardening	
Free energy	$\varphi(\boldsymbol{\varepsilon},\alpha)=(1-d)\varphi_0;\quad \varphi_0=\frac{1}{2}\boldsymbol{\varepsilon}:\mathbf{C}^e:\boldsymbol{\varepsilon}$ $d(\alpha)=1-(q(\alpha)/\alpha)\geqslant 0$	$\varphi(\boldsymbol{\varepsilon}^e,\alpha)=\frac{1}{2}\boldsymbol{\varepsilon}^e:\mathbf{C}^e:\boldsymbol{\varepsilon}^e+\varphi^p(\alpha)$ $\boldsymbol{\varepsilon}=\boldsymbol{\varepsilon}^e+\boldsymbol{\varepsilon}^p$	(9)
Internal variables	$\dot{\alpha}=\dot{\lambda}\quad \alpha\vert_{t=0}=\alpha_0=\sigma_u/\sqrt{E}$	$\dot{\alpha}=\dot{\lambda};\quad \alpha\vert_{t=0}=0$ $\dot{\boldsymbol{\varepsilon}}^p=\dot{\lambda}\mathbf{m};\quad \mathbf{m}=\partial_{\boldsymbol{\sigma}}\Phi(\boldsymbol{\sigma})$	(10)
Constitutive equation	$\boldsymbol{\sigma}=(1-d)\mathbf{C}^e:\boldsymbol{\varepsilon}=\frac{q}{\alpha}\underbrace{\mathbf{C}^e:\boldsymbol{\varepsilon}}_{\bar{\boldsymbol{\sigma}}}=\frac{q}{\alpha}\bar{\boldsymbol{\sigma}}$	$\boldsymbol{\sigma}=\mathbf{C}^e:\boldsymbol{\varepsilon}^e=\mathbf{C}^e:(\boldsymbol{\varepsilon}-\boldsymbol{\varepsilon}^p)$	(11)
Damage/yield function	$g(\boldsymbol{\varepsilon},\alpha)\equiv\tau(\boldsymbol{\varepsilon})-\alpha$ $\tau(\boldsymbol{\varepsilon})\equiv\sqrt{\bar{\boldsymbol{\sigma}}:\mathbf{C}^{e^{-1}}:\bar{\boldsymbol{\sigma}}}=\sqrt{\boldsymbol{\varepsilon}:\mathbf{C}^e:\boldsymbol{\varepsilon}}$	$g(\boldsymbol{\sigma},q)=\Phi(\boldsymbol{\sigma})-q$	(12)
Loading–unloading condition	$\dot{\lambda}\geqslant 0;\quad g\leqslant 0;\quad \dot{\lambda}g=0$	$\dot{\lambda}\geqslant 0;\quad g\leqslant 0;\quad \dot{\lambda}g=0$	(13)
Stress-like internal variable evolution	$\dot{q}=H\dot{\alpha};\quad q\geqslant 0$ $q\vert_{t=0}=\alpha_0=\sigma_u/\sqrt{E}$	$\dot{q}=H\dot{\alpha};\quad q\geqslant 0$ $q\vert_{t=0}=\sigma_u$	(14)
Constitutive tangent tensor	$\dot{\boldsymbol{\sigma}}=\mathbf{C}^{\tan}:\dot{\boldsymbol{\varepsilon}};$ $\mathbf{C}^{\tan}=\begin{cases}\mathbf{C}^{(\text{unl.})}\equiv(1-d)\mathbf{C}^e=\frac{q}{\alpha}\mathbf{C}^e\\ \mathbf{C}^{(\text{load.})}\equiv\frac{q}{\alpha}\mathbf{C}^e-\frac{q-H\alpha}{q^2\alpha}\boldsymbol{\sigma}\otimes\boldsymbol{\sigma}\end{cases}$	$\dot{\boldsymbol{\sigma}}=\mathbf{C}^{\tan}:\dot{\boldsymbol{\varepsilon}}$ $\mathbf{C}^{\tan}=\begin{cases}\mathbf{C}^{(\text{unl.})}\equiv\mathbf{C}^e\\ \mathbf{C}^{(\text{load.})}\equiv\mathbf{C}^e-\frac{\mathbf{C}^e:\mathbf{m}\otimes\mathbf{C}^e:\mathbf{m}}{\mathbf{m}:\mathbf{C}^e:\mathbf{m}+H}\end{cases}$	(15)

The main ingredients of both models can be described as follows [2,4,5]:
where, in Eq. (9), $\varphi(\boldsymbol{\varepsilon},\alpha)$ is the free energy depending on the strain tensor $\boldsymbol{\varepsilon}$ (or the elastic counterpart, $\boldsymbol{\varepsilon}^e$, in the plastic model) and the internal (strain like) variable α and $d(\alpha)$ $(0 \leqslant d \leqslant 1)$ is the damage variable. The φ_0 term, in the damage model, is the elastic strain energy for the elastic (undamaged) material, $\mathbf{C}^e = \bar{\lambda}(\mathbf{1}\otimes\mathbf{1}) + 2\bar{\mu}\mathbf{I}$ is the elastic constitutive tensor, where $\bar{\lambda}$ and $\bar{\mu}$ are the Lame's parameters and $\mathbf{1}$ and $\mathbf{I}$ are the identity tensors of second and fourth order, respectively.

In Eqs. (10) and (14), σ_u is the tensile strength (in the damage model) and the yield stress (in the plastic model), and E is the Young modulus, and in Eq. (11), $\bar{\boldsymbol{\sigma}} = \mathbf{C}^{(e)} : \boldsymbol{\varepsilon}$ is the effective stress.

In Eq. (12) $g(\bullet) = 0$ defines the boundary of the elastic domains. The initial elastic domain in the damage model is then defined, in the strain space, as $E^0_\varepsilon := \{\boldsymbol{\varepsilon}; \tau(\boldsymbol{\varepsilon}) \equiv \sqrt{\boldsymbol{\varepsilon} : \mathbf{C}^e : \boldsymbol{\varepsilon}} < \frac{\sigma_u}{\sqrt{E}}\}$ in terms of the strain norm $\tau(\boldsymbol{\varepsilon})$. For the plasticity model it is defined in the stress space, as $E^0_{\boldsymbol{\sigma}} := \{\boldsymbol{\sigma}; \Phi(\boldsymbol{\sigma}) < \sigma_u\}$, in terms of the equivalent uniaxial stress, $\Phi(\boldsymbol{\sigma})$.

The stresses, $\boldsymbol{\sigma}$, and the stress-like variable, q, are determined via Eqs. (11) and (14). This last equation defines the hardening/softening law in terms of the continuum hardening/softening modulus, $H(\alpha)$, which is assumed a known function of α. Finally, Eq. (15) is the rate constitutive law in terms of the tangent constitutive operator, $\mathbf{C}^{\tan}$, which is specified for loading, $\mathbf{C}^{(\text{load.})}$, or unloading, $\mathbf{C}^{(\text{unl.})}$, processes.

2.1. Implicit integration

Let us consider the problem with the time domain, $[0,T]$, discretized in an appropriated number of time intervals. The classical implicit (backward-Euler) integration of the selected constitutive models in the time interval $[t_n, t_{n+1}]$ is done in the following way:

(a) Damage models: In the domain of non-linear constitutive models, isotropic damage models constitute a fairly unusual case, since they can be integrated in a closed form. In fact, inspection of the evolution Eqs. (10)–(13) allows computing exactly the internal variable $\alpha(t)$ as the historical maximum [6] of the strain norm $\tau(\varepsilon)$ at time t:

$$\alpha(t) = \max_{s\in[0,t]}(\alpha_0, \tau(\boldsymbol{\varepsilon}(s))). \tag{16}$$

From this, the constitutive model can be integrated through the steps as in Table 1.

(b) Elasto-plastic models (radial return): The most common procedure to implicitly integrate elasto-plastic models is the so-called *radial return algorithm*, defined through the steps as in Table 2 [2]:

Table 1
Implicit backward-Euler integration algorithm for the damage model

DATA: $\boldsymbol{\varepsilon}_{n+1}, \alpha_n, q_n$

1. Compute effective stresses and trial values

$$\bar{\boldsymbol{\sigma}}_{n+1} = \mathbf{C}^e : \boldsymbol{\varepsilon}_{n+1}; \tag{17}$$
$$\tau_{n+1} = \sqrt{\bar{\boldsymbol{\sigma}}_{n+1} : \mathbf{C}^{e^{-1}} : \bar{\boldsymbol{\sigma}}_{n+1}} = \|\bar{\boldsymbol{\sigma}}_{n+1}\|_{\mathbf{C}^{e^{-1}}};\ \alpha^{\text{trial}}_{n+1} = \alpha_n$$

2. Compute damage multiplier

$$\begin{aligned}&\text{if } g(\boldsymbol{\varepsilon}_{n+1}, \alpha^{\text{trial}}_{n+1}) \leqslant 0 \rightarrow \Delta\lambda_{n+1} = 0 \rightarrow \text{elastic/unloading}\\ &\text{else if } g(\boldsymbol{\varepsilon}_{n+1}, \alpha^{\text{trial}}_{n+1}) \equiv \tau_{n+1} - \alpha^{\text{trial}}_{n+1} > 0 \text{ then}\\ &\Delta\lambda_{n+1} = \tau_{n+1} - \alpha^{\text{trial}}_{n+1} > 0 \rightarrow \text{damage loading}\end{aligned} \tag{18}$$

3. Update internal and damage variables

$$\begin{aligned}&\alpha_{n+1} = \alpha_n + \Delta\lambda_{n+1} = \max(\alpha_n, \tau_{n+1});\ q_{n+1} = q_n + H_n\Delta\alpha_{n+1};\\ &d_{n+1} = 1 - \frac{q_{n+1}}{\alpha_{n+1}} \geqslant 0\end{aligned} \tag{19}$$

4. Compute stresses

$$\boldsymbol{\sigma}_{n+1} = (1-d_{n+1})\mathbf{C}^e : \boldsymbol{\varepsilon}_{n+1} = \frac{q_{n+1}}{\alpha_{n+1}}\mathbf{C}^e : \boldsymbol{\varepsilon}_{n+1} \tag{20}$$

5. Compute algorithmic tangent operators

$$\begin{aligned}&\mathbf{C}^{\text{alg}(\text{unl.})}_{n+1} = \frac{\partial\boldsymbol{\sigma}_{n+1}}{\partial\boldsymbol{\varepsilon}_{n+1}} = (1-d_{n+1})\mathbf{C}^e\\ &\mathbf{C}^{\text{alg}(\text{load.})}_{n+1} = \frac{\partial\boldsymbol{\sigma}_{n+1}}{\partial\boldsymbol{\varepsilon}_{n+1}} = (1-d_{n+1})\mathbf{C}^e - \frac{q_{n+1} - H_n\alpha_{n+1}}{(\alpha_{n+1})^3}\bar{\boldsymbol{\sigma}}_{n+1}\otimes\bar{\boldsymbol{\sigma}}_{n+1}\end{aligned} \tag{21}$$

Table 2
Implicit backward-Euler integration algorithm for the elasto-plastic model

DATA: $\boldsymbol{\varepsilon}_{n+1}, \boldsymbol{\sigma}_n, \alpha_n, q_n$

1. Compute trial stresses

$$\boldsymbol{\sigma}_{n+1}^{\text{trial}} = \boldsymbol{\sigma}_n + \mathbf{C}^e : (\boldsymbol{\varepsilon}_{n+1} - \boldsymbol{\varepsilon}_n); \quad q_{n+1}^{\text{trial}} = q_n \tag{22}$$

2. Compute plastic multiplier

$$\begin{aligned}
&\text{if } g(\boldsymbol{\sigma}_{n+1}^{\text{trial}}, q_{n+1}^{\text{trial}}) \leqslant 0 \rightarrow \Delta\lambda_{n+1} = 0 \rightarrow \text{elastic/unloading} \\
&\text{else if } g(\boldsymbol{\sigma}_{n+1}^{\text{trial}}, q_{n+1}^{\text{trial}}) > 0 \text{ then} \\
&\left|\begin{array}{l}
\alpha_{n+1}(\Delta\lambda_{n+1}) = \alpha_n + \Delta\lambda_{n+1} \\
q_{n+1}(\Delta\lambda_{n+1}) = q_n + H_n(\alpha_{n+1} - \alpha_n) = q_n + H_n\Delta\lambda_{n+1} \\
\boldsymbol{\sigma}_{n+1}(\Delta\lambda_{n+1}) = \boldsymbol{\sigma}_{n+1}^{\text{trial}}(\boldsymbol{\varepsilon}_{n+1}) - \Delta\lambda_{n+1}\mathbf{C}^e : \mathbf{m}(\boldsymbol{\sigma}_{n+1}(\Delta\lambda_{n+1}), q_{n+1}(\Delta\lambda_{n+1})) \\
\text{Solve} : g(\boldsymbol{\sigma}_{n+1}(\Delta\lambda_{n+1}), q_{n+1}(\Delta\lambda_{n+1})) = 0 \rightarrow \Delta\lambda_{n+1} > 0 \rightarrow \left\{\begin{array}{l}\text{plastic}\\ \text{loading}\end{array}\right.
\end{array}\right.
\end{aligned} \tag{23}$$

3. Update internal variables

$$\alpha_{n+1} = \alpha_n + \Delta\lambda_{n+1}; \quad q_{n+1} = q_n + H_n\Delta\alpha_{n+1} \tag{24}$$

4. Compute stresses

$$\begin{aligned}
&\text{Solve for } \boldsymbol{\sigma}_{n+1} : \\
&\boldsymbol{\sigma}_{n+1} = \boldsymbol{\sigma}_n + \mathbf{C}^e : (\boldsymbol{\varepsilon}_{n+1} - \boldsymbol{\varepsilon}_n) - \Delta\lambda_{n+1}\mathbf{C}^e : \mathbf{m}(\boldsymbol{\sigma}_{n+1}, q_{n+1}) \rightarrow \boldsymbol{\sigma}_{n+1}
\end{aligned} \tag{25}$$

5. Compute algorithmic tangent operators

$$\begin{aligned}
&\mathbf{C}_{n+1}^{\text{alg}(\text{unl.})} = \frac{\partial\boldsymbol{\sigma}_{n+1}}{\partial\boldsymbol{\varepsilon}_{n+1}} = \mathbf{C}^e \\
&\left\{\begin{array}{l}
\mathbf{C}_{n+1}^{\text{alg}(\text{load.})} = \frac{\partial\boldsymbol{\sigma}_{n+1}}{\partial\boldsymbol{\varepsilon}_{n+1}} = \boldsymbol{\Xi}_{n+1} - \frac{\boldsymbol{\Xi}_{n+1}:\mathbf{m}_{n+1}\otimes\boldsymbol{\Xi}_{n+1}:\mathbf{m}_{n+1}}{\mathbf{m}_{n+1}:\boldsymbol{\Xi}_{n+1}:\mathbf{m}_{n+1}+H_n} \\
\boldsymbol{\Xi}_{n+1} = (\mathbf{C}^{e^{-1}} + \Delta\lambda_{n+1}\mathbf{A}_{n+1})^{-1}; \quad \mathbf{A}_{n+1} = \partial_{\boldsymbol{\sigma}_{n+1}}\mathbf{m}_{n+1} = \frac{\partial^2\Phi(\boldsymbol{\sigma}_{n+1})}{\partial\boldsymbol{\sigma}_{n+1}\otimes\partial\boldsymbol{\sigma}_{n+1}}
\end{array}\right.
\end{aligned} \tag{26}$$

In Eq. (23), and from there on, the *algorithmic plastic multiplier*, $\Delta\lambda_{n+1}$, and the *continuum plastic multiplier*, $\dot{\lambda}$ in Eqs. (10) and (13) are related through $\Delta\lambda_{n+1} \equiv \dot{\lambda}_{n+1}\Delta t_{n+1}$.

Remark 2.1. The uniaxial equivalent stress $\Phi(\boldsymbol{\sigma})$, in Eqs. (12) and (26), is a convex function of the stresses as a requirement for the convexity of the elastic domain in the stress space, $E_{\boldsymbol{\sigma}_{n+1}} := \{\boldsymbol{\sigma}_{n+1}; \Phi(\boldsymbol{\sigma}_{n+1}) < q_{n+1}\}$, intrinsic to the use of classical radial return algorithms. This fact automatically translates into the semi-positive definite character of the symmetric tensor $\mathbf{A}_{n+1} = \frac{\partial^2\Phi(\boldsymbol{\sigma}_{n+1})}{\partial\boldsymbol{\sigma}_{n+1}\otimes\partial\boldsymbol{\sigma}_{n+1}}$, in Eq. (26) and, therefore, of the algorithmic elastic tensor [2], Ξ_{n+1}, in Eq. (26) since $\mathbf{C}^{e^{-1}}$ is isotropic and positive definite and $\Delta\lambda_{n+1} \geqslant 0$.

Remark 2.2. Also, very often, the plastic flow $\mathbf{m}(\boldsymbol{\sigma}, q)$ in Eq. (10) is, or it can be rephrased as, a quadratic function of the incremental stresses, $\boldsymbol{\sigma}_{n+1}$ (see Appendix 1 for an example). In those cases, the plastic flow tensor $\mathbf{m}_{n+1}(\boldsymbol{\sigma}_{n+1}, q_{n+1})$ is linear in the stresses i.e.

$$\mathbf{m}(\boldsymbol{\sigma}_{n+1}, q_{n+1}) = \mathbf{A}(q_{n+1}) : \boldsymbol{\sigma}_{n+1} + \mathbf{b}(q_{n+1}) \tag{27}$$

and the tensor $\mathbf{A}_{n+1}$, in Eq. (26), does not depend on the stresses ($\mathbf{A}_{n+1} = \mathbf{A}(q_{n+1})$). These facts will be appropriately recalled in Section 3.2.

2.1.1. Stability and accuracy of implicit integrations

Stability and accuracy of implicit integration algorithms for constitutive models is nowadays a very well established issue. For the two target families of constitutive models, the summary of the corresponding results is the following:

- The implicit integration of the continuum damage model in Eqs. (17)–(21) *is unconditionally stable* (in Appendix 2 a proof is given). Also, the implicit integration for the elasto-plastic models given in Eqs. (22)–(26) is unconditionally stable (proofs of the stability of implicit integrations of elasto-plastic models can be found in Refs. [7,8]).
- Both integration algorithms are first order accurate when using the backward-Euler integration procedures. Mid point integration rules increase the accuracy to second order [7].

2.2. Robustness issues: computability

In numerical solution procedures, computability [9] is associated to the ability of the considered algorithm to provide results at affordable computational costs. In this sense, terms like computability and robustness are closely related.

A typical case of lack of robustness, in non-linear solid mechanics problems, appears when standard Newton–Raphson procedures are used to linearize the resulting discrete system of Eq. (1). In this case, the *ability to provide results* is associated to the *convergence of the iterative procedure*. In turn, the convergence of a Newton–Raphson procedure depends, for instance, on the length of the time

step, but, most importantly, on the spectral properties of the resulting tangent stiffness matrix.

In fact, if the tangent matrix becomes singular, or ill conditioned, at a certain stage of the iterative procedure, the process does not converge (or it consumes an unaffordable number of iterations to converge), even for very short time steps. Then, no solution can be obtained and the process losses its robustness. In certain cases, as in modeling geometrical instabilities, remedies can be found by using appropriate continuation methods, which render the extended tangent stiffness matrix positive definite [10]. However, in many other cases, as in modeling material failure for brittle materials, these remedies fail [3] and the simulations suffer from dramatic lack of robustness.

In this context, non-linear solid mechanics problems involving crack formation and propagation constitute a paradigm of computability difficulties [9]. In *continuum settings*, i.e. when stress vs. strain constitutive models are used, the constitutive models have to be endowed with strain softening in order to model the physical loss of material strength as the crack propagates [11–13]. Then, the softening modulus in Eq. (14) is negative ($H < 0$). The situation is very similar in *discrete settings* [14–16], when material failure is modeled via traction–separation laws, which must be also equipped with strength softening. Apart from mathematical implications on the character of the corresponding boundary value problem, the algorithmic consequences of inserting negatives values of softening parameters in the implicit algorithms in Eqs. (17)–(21) and (22)–(26) are dramatic in terms of robustness.

To examine the reasons for this, let us consider the problem, discretized in time and in space in a finite element mesh of n^{elem} elements, with elemental measure $\Omega^{(e)}$. Then, the momentum Eq. (1) reads:

$$\mathbf{F}_{\text{int}}(\boldsymbol{\sigma}_{n+1}(\mathbf{a}_{n+1})) - \mathbf{F}_{\text{ext}}(t_{n+1}) = \mathbf{0} \tag{28}$$

and the tangent stiffness matrix, $\mathbf{K}^{\text{tang}}_{n+1}$, can be computed as

$$\mathbf{K}^{\text{tang}}_{n+1} = \frac{\partial \mathbf{F}_{\text{int}}(\mathbf{a}_{n+1})}{\partial \mathbf{a}_{n+1}} = \mathcal{A}_{e=1,\dots,n^{\text{elem}}} \left(\int_{\Omega^{(e)}} \nabla \mathbf{N}^{(e)} \cdot \mathbf{C}^{\text{alg.}}_{n+1} \cdot \nabla \mathbf{N}^{(e)} \, \mathrm{d}\Omega \right), \tag{29}$$

where $\mathcal{A}$ stands for the assembling operator and $\mathbf{N}^{(e)}$ are the collection of shape functions involved in element e. Negative values of the hardening/softening modulus, H, translate into loss of the robustness of the algorithmic problem through the following process:

(1) At initial stages of the analysis, the tangent constitutive operator, $\mathbf{C}^{\text{alg.}}_{n+1}$, is elastic ($\mathbf{C}^{\text{alg.}}_{n+1} = \mathbf{C}^{(e)}$) and, therefore, positive definite. So is the global tangent stiffness matrix $\mathbf{K}^{\text{tang}}_{n+1}$.
(2) In subsequent stages, the algorithmic tangent operators for loading cases, in Eqs. (21) and (26), loose positive definiteness at those points where material failure occurs and, therefore, exhibit negative eigenvalues.
(3) Consequently, the kernel of the integral in Eq. (29) losses positive definiteness, and exhibits negative eigenvalues as well.
(4) As material failure propagates through the solid, those local negative eigenvalues deteriorate, via the assembling process, the condition number of the global tangent stiffness matrix, $\mathbf{K}^{\text{tang}}_{n+1}$, whose smallest eigenvalues become progressively closer to zero.
(5) Eventually, $\mathbf{K}^{\text{tang}}_{n+1}$, becomes singular and the convergence fails. In general, there are no simple remedies for this, and the simulation process cannot be continued beyond that point.

Remark 2.3. Through the preceding reasoning it appears that the lack of positive definiteness of the algorithmic tangent operator, as a consequence of including strain softening in the model, is responsible for the observed loss of robustness. However there are other situations where lost of robustness is not directly motivated by strain softening. Modeling *very soft materials* with almost negligible elasticity (for instance in powder compaction processes [17]), involves quasi-singular algorithmic tangent operators in all the domain of the analysis, with very similar consequences in terms of robustness.

3. Implicit/explicit integration: the IMPL-EX scheme

3.1. Fundamentals of the method

The proposed implicit/explicit integration scheme, grounds on the following two stages, *to be executed at the representative time step* $n+1$:

(1) In a first stage the *explicit evaluation* of the stresses, $\tilde{\boldsymbol{\sigma}}_{n+1}$, and the stress-like variable, $\tilde{q}_{n+1}$ (notation $(\tilde{\bullet})$ will be used from now on to denote the corresponding results), is done in terms of the implicit stresses at the previous time step, $\boldsymbol{\sigma}_n$, and the extrapolated values, $\tilde{\alpha}_{n+1}$, of the strain-like internal variable, $\alpha_n, \alpha_{n-1,\dots}$, *implicitly integrated in previous time steps*. Details of this stage are given in next section.
(2) In a second stage the standard *implicit integration* of the constitutive model, according to Tables 1 or 2, is performed and the implicitly integrated stresses, $\boldsymbol{\sigma}_{n+1}$, are obtained.

In addition, fulfillment of the momentum balance Eq. (28) at time step $n+1$ is imposed in terms of these IMPL-EX stresses $\tilde{\boldsymbol{\sigma}}_{n+1}$ and *not in terms of the implicitly integrated stresses*, $\boldsymbol{\sigma}_{n+1}$.

3.2. The explicit stage

Let us, consider the strain-like internal variable, α, in Eq. (10), defining the evolution of the plastic damage (for

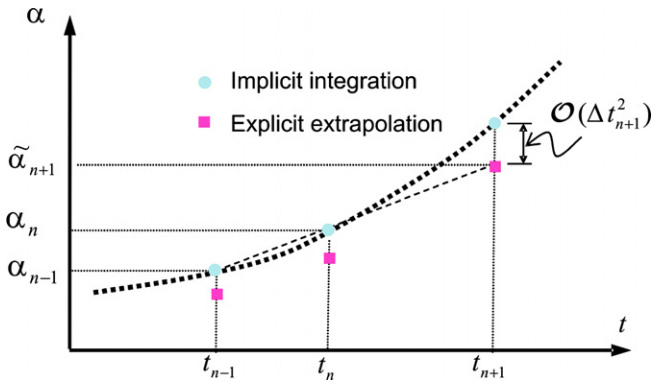

Fig. 1. Internal variable extrapolation.

the damage model) or the plastic multiplier (for the elasto-plastic model), i.e.

$$\dot{\alpha} = \dot{\lambda}. \tag{30}$$

Remark 3.1. Eq. (30), and the existence of the internal variable α, plays a fundamental role in the proposed method, but it is not limiting its application. In fact, if in the considered material model does not appear an internal variable fulfilling this equation, it can be additionally introduced to be object of the specific treatment indicated below.

Let us now consider the discrete flow of that variable obtained from the implicit integration of the constitutive model (see Fig. 1).

At the beginning of the computations at time step $n+1$, results of that integration at previous time steps are available. Taylor's expansions of that variable read:

$$\begin{cases} \alpha_{n+1} = \alpha_n + \dot{\alpha}_n \Delta t_{n+1} + \mathcal{O}(\Delta t_{n+1}^2) \\ \alpha_n = \alpha_{n-1} + \dot{\alpha}_n \Delta t_n + \mathcal{O}(\Delta t_{n+1}^2) \rightarrow \dot{\alpha}_n = \frac{\Delta \alpha_n}{\Delta t_n} - \mathcal{O}(\Delta t_n^2) \end{cases} \tag{31}$$

$$\Rightarrow \alpha_{n+1} = \alpha_n + \frac{\Delta t_{n+1}}{\Delta t_n} \Delta \alpha_n + \mathcal{O}(\Delta t_{n+1}^2), \tag{32}$$

where $\Delta t_n = t_n - t_{n-1}$, $\Delta t_{n+1} = t_{n+1} - t_n$ and $\Delta \alpha_n = \alpha_n - \alpha_{n-1}$. Therefore, truncation of the expansion (32) defines the following prediction, $\tilde{\alpha}_{n+1}$, for the internal variable α_{n+1} (see Fig. 1):

$$\tilde{\alpha}_{n+1} = \alpha_n + \frac{\Delta t_{n+1}}{\Delta t_n} \Delta \alpha_n. \tag{33}$$

Eq. (33) constitutes an *explicit extrapolation* of the internal variable, α, at time step $n+1$, in terms of the *implicit* values obtained in the previous time steps, n and $n-1$ (see Fig. 1). The algorithmic damage/plastic multiplier resulting from this extrapolation reads:

$$\Delta \tilde{\lambda}_{n+1} = \Delta \tilde{\alpha}_{n+1} = \tilde{\alpha}_{n+1} - \alpha_n = \frac{\Delta t_{n+1}}{\Delta t_n} \Delta \alpha_n = \frac{\Delta t_{n+1}}{\Delta t_n} \Delta \lambda_n. \tag{34}$$

Now, steps 3 and 4 in Tables 1 and 2 can be pursued in terms of the value $\Delta \tilde{\alpha}_{n+1}$ in Eq. (34) yielding the IMPL-EX integrated values of the remaining variables $\tilde{\boldsymbol{\sigma}}_{n+1}$ and $\tilde{q}_{n+1}$. Those IMPL-EX results are then substituted in Eq. (28) to fulfill the momentum equation. The IMPL-EX explicit stage for both cases is summarized in Tables 3 and 4.

Remark 3.2. The most specific feature of the IMPL-EX scheme is that, unlike in the standard implicit integration, the values $\tilde{\alpha}_{n+1}$ are independent of the current value of the strains, $\boldsymbol{\varepsilon}_{n+1}$, and they are known at the beginning of time step $n+1$; in other words, *they remain constant during the current time step*. This yields relevant differences in the resulting tangent algorithmic operators, in Eqs. (39) and

Table 3
IMPL-EX explicit stage for isotropic damage models

DATA: $\boldsymbol{\varepsilon}_{n+1}, \alpha_n, \alpha_{n-1}$		
1. Explicit extrapolation	$\tilde{\alpha}_{n+1} = \alpha_n + \frac{\Delta t_{n+1}}{\Delta t_n} \underbrace{\Delta \alpha_n}_{\Delta \lambda_n} = \alpha_n + \frac{\Delta t_{n+1}}{\Delta t_n} \Delta \lambda_n$ $\Delta \alpha_n = \alpha_n - \alpha_{n-1} = \Delta \lambda_n$	(35)
2. Compute damage multiplier	$\Delta \tilde{\lambda}_{n+1} = \Delta \tilde{\alpha}_{n+1} = \tilde{\alpha}_{n+1} - \alpha_n = \frac{\Delta t_{n+1}}{\Delta t_n} \Delta \lambda_n \geqslant 0$	(36)
3. Update internal & damage variables	$\tilde{q}_{n+1} = q_n + H_n \Delta \tilde{\alpha}_{n+1}; \quad \tilde{d}_{n+1} = 1 - \frac{\tilde{q}_{n+1}}{\tilde{\alpha}_{n+1}} \geqslant 0$	(37)
4. Compute stresses	$\tilde{\boldsymbol{\sigma}}_{n+1} = (1 - \tilde{d}_{n+1}) \mathbf{C}^e : \boldsymbol{\varepsilon}_{n+1} = \frac{\tilde{q}_{n+1}}{\tilde{\alpha}_{n+1}} \mathbf{C}^e : \boldsymbol{\varepsilon}_{n+1}$	(38)
5. Compute algorithmic tangent operators	$\tilde{\mathbf{C}}_{n+1}^{\text{alg}} = \frac{\partial \tilde{\boldsymbol{\sigma}}_{n+1}}{\partial \boldsymbol{\varepsilon}_{n+1}} = (1 - \tilde{d}_{n+1}) \mathbf{C}^e$	(39)

Table 4
IMPL-EX explicit stage for elasto-plastic models

DATA: $\boldsymbol{\varepsilon}_{n+1}, \alpha_n, \alpha_{n-1}, \boldsymbol{\sigma}_n, q_n$

1. Explicit extrapolation

$$\tilde{\alpha}_{n+1} = \alpha_n + \frac{\Delta t_{n+1}}{\Delta t_n} \underbrace{\Delta \alpha_n}_{\Delta \lambda_n} = \alpha_n + \frac{\Delta t_{n+1}}{\Delta t_n} \Delta \lambda_n \tag{40}$$

$$\Delta \alpha_n = \alpha_n - \alpha_{n-1} = \Delta \lambda_n$$

2. Compute plastic multiplier

$$\Delta \tilde{\lambda}_{n+1} = \Delta \tilde{\alpha}_{n+1} = \tilde{\alpha}_{n+1} - \alpha_n = \frac{\Delta t_{n+1}}{\Delta t_n} \Delta \lambda_n \geqslant 0 \tag{41}$$

3. Update internal variables

$$\tilde{q}_{n+1} = q_n + H_n \Delta \tilde{\alpha}_{n+1} \tag{42}$$

4. Compute stresses

Solve for $\tilde{\boldsymbol{\sigma}}_{n+1}$: (43)

$$\tilde{\boldsymbol{\sigma}}_{n+1} = \underbrace{\boldsymbol{\sigma}_n + \mathbf{C}^e : (\boldsymbol{\varepsilon}_{n+1} - \boldsymbol{\varepsilon}_n)}_{\boldsymbol{\sigma}^{\text{trial}}_{n+1}} - \Delta \tilde{\lambda}_{n+1} \mathbf{C}^e : \tilde{\mathbf{m}}(\tilde{\boldsymbol{\sigma}}_{n+1}, \tilde{q}_{n+1}) \rightarrow \tilde{\boldsymbol{\sigma}}_{n+1}$$

$$\text{For linear plastic flow} : (\tilde{\mathbf{m}}_{n+1} = \widetilde{\mathbf{A}}_{n+1}(\tilde{q}_{n+1}) : \tilde{\boldsymbol{\sigma}}_{n+1} + \tilde{\mathbf{b}}_{n+1}(\tilde{q}_{n+1}))$$

$$\tilde{\boldsymbol{\sigma}}_{n+1} = \widetilde{\boldsymbol{\Xi}}_{n+1} : [\mathbf{C}^{e^{-1}} : \boldsymbol{\sigma}^{\text{trial}}_{n+1} - \Delta \tilde{\lambda}_{n+1} \tilde{\mathbf{b}}_{n+1}]; \quad \widetilde{\boldsymbol{\Xi}}_{n+1} = (\mathbf{C}^{e^{-1}} + \Delta \tilde{\lambda}_{n+1} \widetilde{\mathbf{A}}_{n+1})^{-1}$$

5. Compute algorithmic tangent operators

$$\widetilde{\mathbf{C}}^{\text{alg}}_{n+1} = \frac{\partial \tilde{\boldsymbol{\sigma}}_{n+1}}{\partial \boldsymbol{\varepsilon}_{n+1}} = \widetilde{\boldsymbol{\Xi}}_{n+1}; \quad \widetilde{\boldsymbol{\Xi}}_{n+1} = (\mathbf{C}^{e^{-1}} + \Delta \tilde{\lambda}_{n+1} \widetilde{\mathbf{A}}_{n+1})^{-1} \tag{44}$$

$$\widetilde{\mathbf{A}}_{n+1} = \partial_{\tilde{\boldsymbol{\sigma}}_{n+1}} \tilde{\mathbf{m}}_{n+1} = \frac{\partial^2 \Phi(\tilde{\boldsymbol{\sigma}}_{n+1})}{\partial \tilde{\boldsymbol{\sigma}}_{n+1} \otimes \partial \tilde{\boldsymbol{\sigma}}_{n+1}}$$

(44), when compared with the ones resulting from the implicit integration in Eqs. (21) and (26). The key differences are:

(1) *The IMPL-EX algorithmic tangent operators are in all cases symmetric and semi-positive definite*. The arguments are trivial for the damage model (Eq. (39)), since $(1 - \tilde{d}_{n+1}) \geqslant 0$ and $\mathbf{C}^e$ is symmetric and positive definite. For the elasto-plastic model, in Eq. (44), they follow from the reasoning in Remark 2.1, about the symmetric and semi-positive definite character of tensor $\widetilde{\mathbf{A}}_{n+1}$, and the positive character of $\Delta \tilde{\lambda}_{n+1}$ and $\mathbf{C}^e$. Therefore, those *problems of robustness,* referred to in Section 2.2, due to the negative character of the algorithmic tangent operators in Eq. (29) *should disappear.*

(2) The algorithmic tangent operator, in Eq. (39) *is constant* (independent of the current strain, $\boldsymbol{\varepsilon}_{n+1}$) *for the damage model*. For the elasto-plastic model, *the algorithmic tangent operator,* in Eq. (44), *is also constant* for the linear plastic flow cases mentioned in Remark 2.2 (see Eq. (27)). Otherwise, the plastic flow can be expanded around $\boldsymbol{\sigma}_n$, for the explicit stage in Table 4 as (see Appendix 1 for a specific example):

$$\begin{aligned} \mathbf{m}(\tilde{\boldsymbol{\sigma}}_{n+1}) &= \mathbf{m}(\boldsymbol{\sigma}_n + \Delta \tilde{\boldsymbol{\sigma}}_{n+1}) \\ &= \mathbf{m}(\boldsymbol{\sigma}_n) + \mathbf{A}(\boldsymbol{\sigma}_n) : \Delta \tilde{\boldsymbol{\sigma}}_{n+1} + \mathcal{O}(\Delta \tilde{\boldsymbol{\sigma}}_{n+1})^2, \\ \Delta \tilde{\boldsymbol{\sigma}}_{n+1} &= \tilde{\boldsymbol{\sigma}}_{n+1} - \boldsymbol{\sigma}_n. \end{aligned} \tag{45}$$

Truncation up to linear terms gives rise to the linearized plastic flow:

$$\begin{aligned} \tilde{\mathbf{m}}_{n+1} &= \mathbf{A}(\boldsymbol{\sigma}_n) : \Delta \tilde{\boldsymbol{\sigma}}_{n+1} + \mathbf{m}(\boldsymbol{\sigma}_n), \\ \widetilde{\mathbf{A}}_{n+1} &= \frac{\partial \tilde{\mathbf{m}}_{n+1}}{\partial \tilde{\boldsymbol{\sigma}}_{n+1}} = \mathbf{A}(\boldsymbol{\sigma}_n) \end{aligned} \tag{46}$$

and, again, the algorithmic tangent operator, in Eq. (44), is constant during the time step $n+1$. In all those cases, the Newton–Raphson linearization of Eq. (28) yields *a constant tangent matrix,* $\widetilde{\mathbf{K}}^{\text{tang}}_{n+1}$, the iterative process should converge in a unique iteration per time step, and *the solving procedure becomes step-linear*.

Remark 3.3. The explicit stage of the IMPL-EX algorithm reduces to the simple algebraic computations displayed in Tables 3 and 4. Therefore, *the IMPL-EX scheme does not involve relevant additional computational costs*, other than storing in memory some values computed in the implicit stage, typically the stress field, $\boldsymbol{\sigma}$, and the algorithmic plastic/damage multiplier, $\Delta\lambda$, to be used in subsequent time steps.

3.3. The IMPL-EX algorithm as a predictor/corrector scheme

In order to provide an interpretation of the IMPL-EX integration scheme, it is illustrative to decompose the stresses, $\tilde{\boldsymbol{\sigma}}_{n+1}$, in Eqs. (38) and (43), as the sum of a predictor

stress, $\tilde{\boldsymbol{\sigma}}^{(0)}_{n+1}$, and the remaining corrector stress. Therefore, for the damage model:

$$\begin{aligned}\tilde{\boldsymbol{\sigma}}_{n+1}(\boldsymbol{\varepsilon}_{n+1}) &= (1-\tilde{d}_{n+1})\mathbf{C}^e : \boldsymbol{\varepsilon}_{n+1} \\ &= \underbrace{\tilde{\boldsymbol{\sigma}}^{(0)}_{n+1}}_{\text{predictor}} + \underbrace{(1-\tilde{d}_{n+1})\mathbf{C}^e : \Delta\boldsymbol{\varepsilon}_{n+1}}_{\text{corrector}},\end{aligned} \tag{47}$$

$$\tilde{\boldsymbol{\sigma}}^{(0)}_{n+1} \cong \boldsymbol{\sigma}_{n+1}|_{\Delta\boldsymbol{\varepsilon}_{n+1}=0} = (1-\tilde{d}_{n+1})\mathbf{C}^e : \boldsymbol{\varepsilon}_n = \frac{\tilde{q}_{n+1}}{\tilde{\alpha}_{n+1}}\mathbf{C}^e : \boldsymbol{\varepsilon}_n$$

and for the elasto-plastic model (and the linear plastic flow case):

$$\begin{aligned}\tilde{\sigma}_{n+1} &= \tilde{\Xi}_{n+1} : \lfloor \mathbf{C}^{e^{-1}} : \boldsymbol{\sigma}_n + (\boldsymbol{\varepsilon}_{n+1} - \boldsymbol{\varepsilon}_n) - \Delta\tilde{\lambda}_{n+1}\tilde{\mathbf{b}}_{n+1} \rfloor \\ &= \underbrace{\tilde{\boldsymbol{\sigma}}^{(0)}_{n+1}}_{\text{predictor}} + \underbrace{\tilde{\Xi}_{n+1} : \Delta\boldsymbol{\varepsilon}_{n+1}}_{\text{corrector}},\end{aligned} \tag{48}$$

$$\tilde{\boldsymbol{\sigma}}^{(0)}_{n+1} = \tilde{\boldsymbol{\sigma}}_{n+1}|_{\boldsymbol{\varepsilon}_{n+1}=\boldsymbol{\varepsilon}_n} = \tilde{\Xi}_{n+1} : [\mathbf{C}^{e^{-1}} : \boldsymbol{\sigma}_n - \Delta\tilde{\lambda}_{n+1}\tilde{\mathbf{b}}_{n+1}].$$

The interpretation of that decomposition can be seen in Fig. 2. It should be noticed that the predictor stresses, $\tilde{\boldsymbol{\sigma}}^{(0)}_{n+1}$, in Eqs. (47) and (48), can be calculated, without any further information, at the beginning of the time step $n+1$. They define the initial value for the internal forces $\mathbf{F}_{\text{int}}(\tilde{\boldsymbol{\sigma}}^{(0)}_{n+1})$ in the momentum Eq. (28). Then, the first iteration of the Newton–Raphson procedure, corrects them, across a linear correction path, to fulfill the momentum Eq. (28).

Remark 3.4. Like in regular integration procedures, the IMPL-EX scheme, yields null unbalanced (residual) forces at the end of the time step n i.e.:

$$\widetilde{\mathbf{R}}_n \equiv \mathbf{F}_{\text{int}}(\tilde{\boldsymbol{\sigma}}_n) - \mathbf{F}_{\text{ext}}(t_n) = \mathbf{0}, \tag{49}$$

but, unlike in standard integration procedures, the aforementioned predictor–corrector character, yields an unbalanced residue $\widetilde{\mathbf{R}}^{(0)}_{n+1}$ at the beginning of the time step $n+1$ (before any increment of the external actions):

$$\widetilde{\mathbf{R}}^{(0)}_{n+1} \equiv \mathbf{F}_{\text{int}}(\tilde{\boldsymbol{\sigma}}^{(0)}_{n+1}) - \mathbf{F}_{\text{ext}}(t_n) \neq \mathbf{0}, \tag{50}$$

since $\tilde{\boldsymbol{\sigma}}^{(0)}_{n+1} \neq \tilde{\boldsymbol{\sigma}}_n$, according to Eqs. (47) and (48). The null character of the initial residual forces, $\widetilde{\mathbf{R}}^{(0)}_{n+1}$ in Eq. (49), is implicitly assumed in many continuation (arc-length) methods [10]. Therefore, in order to use them together with the IMPL-EX integration scheme, they have to be slightly (and trivially) modified. In Appendix 3, the modified version of the classical *updated normal plane*, to be used with the proposed integration method, is described.

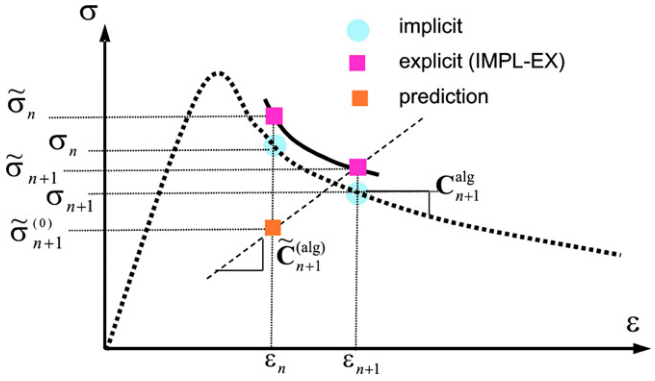

Fig. 2. Prediction/correction stages of the IMPL-EX integration scheme.

3.4. The IMPL-EX algorithm as a fractional time-step scheme

Let us consider the global time evolution problem, giving raise to the time-discretized problem in Eqs. (1)–(3):

Find:

$$\mathbf{a}(t); \alpha(t); \boldsymbol{\sigma}(t)$$

Such that:

$$\mathbf{F}_{\text{int}}(\mathbf{a}, \boldsymbol{\sigma}, t) - \mathbf{F}_{\text{ext}}(t) = \mathbf{G}(\mathbf{a}, \boldsymbol{\sigma}, t) = \mathbf{0} \quad \text{(balance of forces)}, \tag{51}$$

$$g(\alpha, \boldsymbol{\sigma}, t) = \mathbf{0} \quad \text{(state/evolution equation)}, \tag{52}$$

$$\dot{\boldsymbol{\sigma}} = \Sigma(\boldsymbol{\varepsilon}(\mathbf{a}), \alpha, \boldsymbol{\sigma}) \quad \text{(constitutive equation)}. \tag{53}$$

Eqs. (51)–(53) can be regarded as a set of differential-algebraic equations (DAE) [18], in the unknowns $(\mathbf{a}, \alpha, \boldsymbol{\sigma})$ where integration of the *ordinary differential equation* (ODE) (53), is constrained by the fulfillment of the *algebraic equations* (51) and (52).

The IMPL-EX integration scheme outlined above can be rewritten as a *fractional step method* [18–20] for integration of the global time evolution problem in Eqs. (1)–(3) based on the steps shown in Table 5.

In Step 1 (Eq. (55)) the extrapolation of the internal variable $\tilde{\alpha}_{n+1}$, as in Eq. (33), is done. Then, the stresses $\tilde{\boldsymbol{\sigma}}_{n+1}$ are computed, in Eq. (56), via a backward-Euler integration of Eq. (53), and the balance of forces Eq. (54) is solved providing the nodal displacements, $\mathbf{a}_{n+1}$. Intrinsically, Eqs. (54)–(56) in Step 1 constitute a non-linear system, which has to be iteratively solved repeating Step 1 until achieving convergence. However, for the damage model and the linear plastic flow cases, it becomes a linear system of equations (see Remark 3.2). This constitutes the IMPL-EX explicit stage described in Tables 3 and 4.

Then, in Step 2 the values $\mathbf{a}_{n+1}$ remain frozen, and Eqs. (55) and (56) correspond to the *IMPL-EX implicit stage* i.e.: the implicit integration of the constitutive models in Tables 1 and 2.

The consistency of the scheme in Table 5, with the differential algebraic system (51)–(53), is proven by checking that, for $\Delta t_{n+1} = 0$:

$$\alpha_{n+1} = \tilde{\alpha}_{n+1} = \alpha_n; \quad \boldsymbol{\sigma}_{n+1} = \tilde{\boldsymbol{\sigma}}_{n+1} = \boldsymbol{\sigma}_n \tag{57}$$

and that the solution $(\mathbf{a}_{n+1}, \alpha_{n+1}, \boldsymbol{\sigma}_{n+1})$ fulfills the time-discretized version, in Eqs. (1)–(3), of the flow problem in Eqs. (54)–(56).

3.5. Accuracy analysis

The IMPL-EX scheme introduces a specific error associated to the extrapolation of the internal variable in Eqs.

Table 5
The IMPL-EX scheme as a constrained fractional step integration method

Step 1	Step 2	
$\mathbf{F}_{\text{int}}(\mathbf{a}_{n+1},\tilde{\boldsymbol{\sigma}}_{n+1},t_{n+1})-\mathbf{F}_{\text{ext}}(t_{n+1})=\mathbf{0}$	$\dot{\mathbf{a}}_{n+1}=0$	(54)
$\tilde{\alpha}_{n+1}=\alpha_n+\dfrac{\Delta t_{n+1}}{\Delta t_n}\Delta\alpha_n$	$g(\alpha_{n+1},\boldsymbol{\sigma}_{n+1},t_{n+1})=0$	(55)
$\dot{\tilde{\boldsymbol{\sigma}}}_{n+1}\equiv\dfrac{\tilde{\boldsymbol{\sigma}}_{n+1}(\mathbf{a}_{n+1})-\boldsymbol{\sigma}_n}{\Delta t_{n+1}}=\boldsymbol{\Sigma}(\boldsymbol{\varepsilon}(\mathbf{a}_{n+1}),\tilde{\alpha}_{n+1},\tilde{\boldsymbol{\sigma}}_{n+1})$	$\dot{\boldsymbol{\sigma}}_{n+1}\equiv\dfrac{\boldsymbol{\sigma}_{n+1}-\boldsymbol{\sigma}_n}{\Delta t_{n+1}}=\Sigma(\boldsymbol{\varepsilon}(\mathbf{a}_{n+1}),\alpha_{n+1},\boldsymbol{\sigma}_{n+1})$	(56)
IMPL-EX explicit stage: $\mathbf{a}_{n+1}$	IMPL-EX implicit stage: $(\alpha_{n+1},\boldsymbol{\sigma}_{n+1})$	

(32) and (33). Therefore, the accuracy of the algorithm will depend crucially on the type of this extrapolation.

In order to work in a more general setting, let us consider the following family of extrapolations of the internal variable, $\tilde{\alpha}_{n+1}$, in terms of the implicitly integrated values at previous time steps α_n,α_{n-1} and α_{n-2}:

$$\Delta\tilde{\alpha}_{n+1}=\tilde{\alpha}_{n+1}-\alpha_n=\beta\frac{\Delta t_{n+1}}{\Delta t_n}\Delta\alpha_n+(1-\beta)\frac{\Delta t_{n+1}}{\Delta t_{n-1}}\Delta\alpha_{n-1},$$
$$\Delta\alpha_n=\alpha_n-\alpha_{n-1};\quad \Delta\alpha_{n-1}=\alpha_{n-1}-\alpha_{n-2}. \tag{58}$$

In order to find out the accuracy properties of that extrapolation family, and assuming, when needed, for mathematical simplification that $\Delta t_{n-1}=\Delta t_n=\Delta t_{n+1}$ let us consider the following Taylor's expansion of α_{n-1} around t_n:

$$\begin{aligned}\alpha_{n-1}&=\alpha(t_n-\Delta t_n)\\&=\alpha_n-\dot{\alpha}_n\Delta t_n+\frac{1}{2}\ddot{\alpha}_n(\Delta t_n)^2-\frac{1}{6}\dddot{\alpha}_n(\Delta t_n)^3\\&\quad+\mathcal{O}(\Delta t_{n+1}^4),\end{aligned} \tag{59}$$

where $\dot{\alpha}_n,\ddot{\alpha}_n$ and $\dddot{\alpha}_n$ stand for increasing time derivatives at time step n. Therefore, from Eq. (59):

$$\begin{aligned}\Delta\alpha_n&=\alpha_n-\alpha_{n-1}\\&=\dot{\alpha}_n\Delta t_n-\frac{1}{2}\ddot{\alpha}_n(\Delta t_n)^2+\frac{1}{6}\dddot{\alpha}_n(\Delta t_n)^3+\mathcal{O}(\Delta t_{n+1}^4).\end{aligned} \tag{60}$$

Similarly to Eq. (60), for the time interval $[t_{n-2},t_{n-1}]$ one gets:

$$\begin{aligned}\Delta\alpha_{n-1}&=\alpha_{n-1}-\alpha_{n-2}\\&=\dot{\alpha}_{n-1}\Delta t_{n-1}-\frac{1}{2}\ddot{\alpha}_{n-1}(\Delta t_{n-1})^2+\frac{1}{6}\dddot{\alpha}_{n-1}(\Delta t_{n-1})^3\\&\quad+\mathcal{O}(\Delta t_{n+1}^4).\end{aligned} \tag{61}$$

In turn, time derivatives in Eq. (61) can be approximated by the following expansions:

$$\begin{aligned}\dot{\alpha}_{n-1}&=\dot{\alpha}(t_n-\Delta t_{n-1})=\dot{\alpha}_n-\ddot{\alpha}_n\Delta t_{n-1}+\frac{1}{2}\dddot{\alpha}_n(\Delta t_{n-1})^2+\mathcal{O}(\Delta t_{n+1}^3),\\\ddot{\alpha}_{n-1}&=\ddot{\alpha}(t_n-\Delta t_{n-1})=\ddot{\alpha}_n-\dddot{\alpha}_n\Delta t_{n-1}+\mathcal{O}(\Delta t_{n+1}^2),\\\dddot{\alpha}_{n-1}&=\dddot{\alpha}(t_n-\Delta t_{n-1})=\dddot{\alpha}_n+\mathcal{O}(\Delta t_{n-1}).\end{aligned} \tag{62}$$

Substitution of Eq. (62) into Eq. (61), provides, after some straight-forward operations, the following expansion of $\Delta\alpha_{n-1}$ around t_n:

$$\begin{aligned}\Delta\alpha_{n-1}&=\alpha_{n-1}-\alpha_{n-2}\\&=\dot{\alpha}_n\Delta t_{n-1}-\frac{3}{2}\ddot{\alpha}_n(\Delta t_{n-1})^2+\frac{7}{6}\dddot{\alpha}_n(\Delta t_{n-1})^3\\&\quad+\mathcal{O}(\Delta t_{n+1}^4).\end{aligned} \tag{63}$$

Then, substitution of Eqs. (60) and (63) into (58) yields:

$$\begin{aligned}\Delta\tilde{\alpha}_{n+1}&=[\beta+(1-\beta)]\dot{\alpha}_n\Delta t_{n+1}\\&\quad-\left[\frac{1}{2}\beta+\frac{3}{2}(1-\beta)\right]\ddot{\alpha}_n(\Delta t_{n+1})^2\\&\quad+\left[\frac{1}{6}\beta+\frac{7}{6}(1-\beta)\right]\dddot{\alpha}_n(\Delta t_{n+1})^3+\mathcal{O}(\Delta t_{n+1}^4)\end{aligned} \tag{64}$$

and, after performing the bracketed operations:

$$\begin{aligned}\Delta\tilde{\alpha}_{n+1}&=\tilde{\alpha}_{n+1}-\alpha_n\\&=\dot{\alpha}_n\Delta t_{n+1}-\left(\frac{3}{2}-\beta\right)\ddot{\alpha}_n(\Delta t_{n+1})^2\\&\quad+\left(\frac{7}{6}-\beta\right)\dddot{\alpha}_n(\Delta t_{n+1})^3+\mathcal{O}(\Delta t_{n+1}^4).\end{aligned} \tag{65}$$

Eq. (65) provides the resulting expansion of the *IMPL-EX extrapolated increment*, $\Delta\tilde{\alpha}_{n+1}$, around t_n. On the other hand, the Taylor's expansion of the *implicitly integrated increment*, $\Delta\alpha_{n+1}$, reads:

$$\begin{aligned}\Delta\alpha_{n+1}&=\alpha_{n+1}-\alpha_n\\&=\dot{\alpha}_n\Delta t_{n+1}+\frac{1}{2}\ddot{\alpha}_n(\Delta t_{n+1})^2+\frac{1}{6}\dddot{\alpha}_n(\Delta t_{n+1})^3\\&\quad+\mathcal{O}(\Delta t_{n+1}^4).\end{aligned} \tag{66}$$

Subtraction of Eq. (66) from (65) yields:

$$\begin{aligned}\tilde{\alpha}_{n+1} - \alpha_{n+1} &= \Delta\tilde{\alpha}_{n+1} - \Delta\alpha_{n+1} \\ &= -(2-\beta)\ddot{\alpha}_n(\Delta t_{n+1})^2 + (1-\beta)\dddot{\alpha}_n(\Delta t_{n+1})^3 \\ &\quad + \mathcal{O}(\Delta t_{n+1}^4),\end{aligned} \tag{67}$$

which establishes the error of the IMPL-EX results with respect to the implicit ones. Eq. (67) proves the order one accuracy of the extrapolation (58) for any value of β excepting the particular choice $\beta = 2$ that provides second order accuracy. For this value, the extrapolation in Eq. (58) yields:

$$\Delta\tilde{\alpha}_{n+1} = 2\frac{\Delta t_{n+1}}{\Delta t_n}\Delta\alpha_n - \frac{\Delta t_{n+1}}{\Delta t_{n-1}}\Delta\alpha_{n-1} \tag{68}$$

as the optimal (in terms of accuracy) linear extrapolation for the IMPL-EX scheme.

In order to corroborate these results, in Fig. 3 an accuracy analysis, proposed in [2,7] but here based on the IMPL-EX integration of a J2 plasticity model, using different values of β in Eq. (58), is presented.

A single material point (Gauss-point) is subjected to uniaxial stress and driven to yielding. Then, the required number of, constant and arbitrary, strain increments are given, and the corresponding stresses are integrated using the IMPL-EX algorithm of Table 4. The error is then computed with respect to the implicit integration, on the same strain path, using a very large number of increments, which is considered the *exact* solution (see Fig. 3).

For all values $\beta \neq 2$ the slope, γ, of the error curve in a log–log diagram is $\gamma = 2$, corresponding to a first order accuracy in Eq. (67). Only for $\beta = 2$ the slope of the error curve is $\gamma = 3$, agreeing with the second order accuracy expected from Eq. (67).

Remark 3.5. It is well known that accuracy analyzes apply only to the limit case of infinitesimal time increments ($\Delta t \to 0$) and monotonic loading. Therefore, higher accuracy orders could not translate into smaller errors for finite time step lengths. This has already been observed in implicit integration schemes for constitutive equations [7]. The experience of the authors, in a large number of simulations, shows that, for typical time step lengths, the extrapolation scheme in Eq. (68) ($\beta = 2$ in Eq. (58)) is not the optimal, in terms of the global error, in comparison with the one in Eq. (33) ($\beta = 1$), in spite of the higher order of accuracy of the former. Therefore, in all the examples, presented in the remaining of this work, the linear extrapolation scheme, in Eq. (33), has been used.

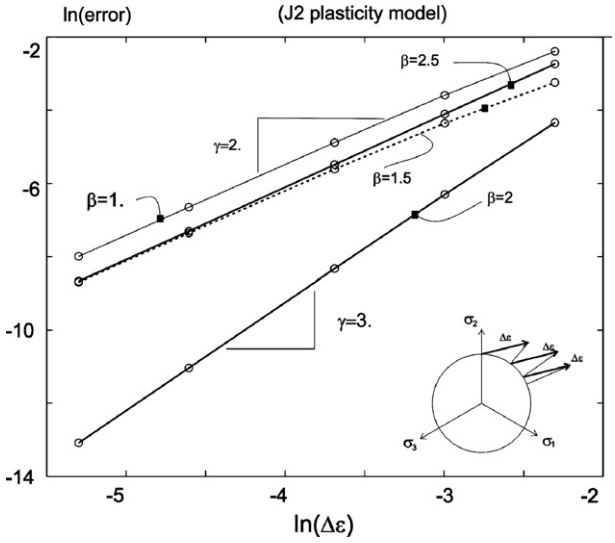

Fig. 3. Error (log–log) diagram for the IMPL-EX integration of a J2 plasticity model.

3.6. Error control: adaptive time stepping

In Section 3.5 the additional error, with respect to the implicitly integrated values, resulting from the extrapolation in Eq. (58) has been analyzed. It appears that there are two different sources for this error:

- The intrinsic error, resulting from the IMPL-EX linear extrapolation of the internal variable α in Eq. (35) or (40), *at the smooth parts* of the internal variable evolution curves (see Fig. 4).
- The error resulting from that extrapolation *at points where this curve is not smooth* (typically when passing from elastic to inelastic stages or the opposite. In fact, at these points the subdiferential of function $\alpha(t)$ has multiple subderivatives, $\partial\alpha = \lfloor\dot{\lambda}^-, \dot{\lambda}^+\rfloor$, and the IMPL-EX extrapolation is done on the basis of the first of those subderivatives, $\dot{\alpha}^-$ (see Eqs. (35) and (36) or (40) and (41)) i.e.:

$$\tilde{\alpha}_{n+1} = \alpha_n + \frac{\Delta\lambda_n}{\Delta t_n}\Delta t_{n+1} \approx \alpha_n + \dot{\lambda}_n^- \Delta t_{n+1}. \tag{69}$$

The consequences of those errors on the results will be examined in Section 4.1. In spite of the order of accuracy determined by Eq. (67), it is clear that the extrapolation error becomes larger as the length of the time step, Δt, increases. Therefore, the necessity to control this length,

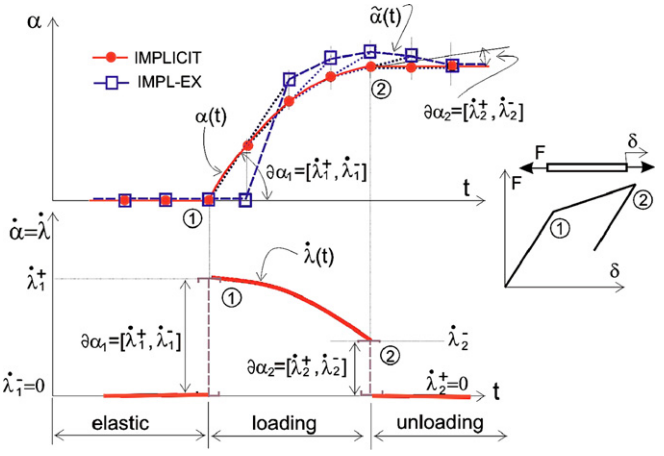

Fig. 4. Errors in the IMPL-EX integration scheme. In points 1 and 2 the evolution of α is not smooth.

in order to keep that error into appropriate bounds, arises. In this section, an adaptive time step scheme developed for this purpose is presented. The scheme has an *a priori* character, so that the time step length at the time step $n+1$ is calculated in terms of values determined at previous time steps. The goal of the scheme is, then, to keep the error due to the IMPL-EX extrapolation, $e_{\alpha_{n+1}}(\mathbf{x})$, bounded at all material points $\mathbf{x} \in \Omega^h$ i.e.:

$$e_{\alpha_{n+1}}(\mathbf{x}) = |\tilde{\alpha}_{n+1} - \alpha_{n+1}|(\mathbf{x}) \leqslant \xi\alpha^{\text{ref}} \quad \forall \mathbf{x} \in \Omega^h, \tag{70}$$

where $\xi \in (0,1)$ stands for the tolerance of the relative error, with respect to α^{ref} a, model dependent, reference value. For the damage and elasto-plastic models in Eqs. (9)–(15) the following values have been considered:

$$\alpha^{\text{ref}} = \begin{cases} \text{damage model} \rightarrow \frac{\sigma_u}{\sqrt{E}} (= \alpha_0 = q_0) \\ \text{elasto-plastic model} \rightarrow \frac{\sigma_u}{E} (= \text{uniaxial elastic strain}). \end{cases} \tag{71}$$

Let us then consider the Taylor's expansion in Eq. (67) for the considered case $\beta = 1$ (see Remark 3.5):

$$\tilde{\alpha}_{n+1} - \alpha_{n+1} = -\ddot{\alpha}_n \Delta t_{n+1}^2 + \mathcal{O}(\Delta t_{n+1}^4). \tag{72}$$

Therefore, the absolute error $e_{\alpha_{n+1}}$, in Eq. (70), can be evaluated as

$$e_{\alpha_{n+1}} = |\tilde{\alpha}_{n+1} - \alpha_{n+1}| \cong |\ddot{\alpha}_n| \Delta t_{n+1}^2. \tag{73}$$

Additional Taylor's expansions yield the values:

$$\begin{aligned} \dot{\alpha}_n &= \frac{\overbrace{\alpha_n - \alpha_{n-1}}^{\Delta\alpha_n}}{\Delta t_n} + \mathcal{O}(\Delta t) \cong \frac{\Delta\alpha_n}{\Delta t_n} = \frac{\Delta\lambda_n}{\Delta t_n}, \\ \dot{\alpha}_{n-1} &= \frac{\overbrace{\alpha_{n-1} - \alpha_{n-2}}^{\Delta\alpha_{n-1}}}{\Delta t_{n-1}} + \mathcal{O}(\Delta t) \cong \frac{\Delta\alpha_{n-1}}{\Delta t_{n-1}} = \frac{\Delta\lambda_{n-1}}{\Delta t_{n-1}}, \\ \ddot{\alpha}_n &= \frac{\dot{\alpha}_n - \dot{\alpha}_{n-1}}{\Delta t_n} + \mathcal{O}(\Delta t) \cong \frac{\dot{\alpha}_n - \dot{\alpha}_{n-1}}{\Delta t_n} = \frac{1}{\Delta t_n}\left(\frac{\Delta\lambda_n}{\Delta t_n} - \frac{\Delta\lambda_{n-1}}{\Delta t_{n-1}}\right), \end{aligned} \tag{74}$$

where the algorithmic result $\Delta\alpha_n = \Delta\lambda_n \forall n$, in Eqs. (19) and (24), has been considered. Substitution of the last Eq. (74) into Eq. (73) yields:

$$e_{\alpha_{n+1}} \cong |\ddot{\alpha}_n| \Delta t_{n+1}^2 = \frac{1}{\Delta t_n}\left|\frac{\Delta\lambda_n}{\Delta t_n} - \frac{\Delta\lambda_{n-1}}{\Delta t_{n-1}}\right| \Delta t_{n+1}^2. \tag{75}$$

In order to keep the extrapolation error at the material point $\mathbf{x} \in \Omega^h$ equal to the bound in Eq. (70) (i.e.: $e_{\alpha_{n+1}}(\mathbf{x}) = \xi\alpha^{\text{ref}}$), from Eq. (75) the critical time step length, $\Delta t_{n+1}^{\text{crit}}$, has to fulfill:

$$\begin{aligned} (\Delta t_{n+1}^{\text{crit}}(\mathbf{x}))^2 &= \frac{e_{\alpha_{n+1}}(\mathbf{x}) \Delta t_n}{\left|\frac{\Delta\lambda_n(\mathbf{x})}{\Delta t_n} - \frac{\Delta\lambda_{n-1}(\mathbf{x})}{\Delta t_{n-1}}\right|} \\ &= \frac{\xi\alpha^{\text{ref}}}{\left|\Delta\lambda_n(\mathbf{x}) - \frac{\Delta t_n}{\Delta t_{n-1}}\Delta\lambda_{n-1}(\mathbf{x})\right|} \Delta t_n^2. \end{aligned} \tag{76}$$

Therefore, in order to keep the relative error *below* the tolerance ξ *at all points of the discretized domain* Ω^h, an estimation of the time step length reads:

$$\begin{aligned} \Delta t_{n+1}^2 &\leqslant \underset{\mathbf{x}\in\Omega^h}{\text{MIN}} (\Delta t_{n+1}^{\text{crit}}(\mathbf{x}))^2 \\ &= \xi\alpha^{\text{ref}} \Delta t_n^2 \underset{\mathbf{x}\in\Omega^h}{\text{MIN}} \frac{1}{\left|\Delta\lambda_n(\mathbf{x}) - \frac{\Delta t_n}{\Delta t_{n-1}}\Delta\lambda_{n-1}(\mathbf{x})\right|}. \end{aligned} \tag{77}$$

An alternative expression, which is the one used by the authors for practical purposes, consists of considering the factor $\Delta t_n / \Delta t_{n-1}$ in, Eq. (77), close to one and, then, estimating the time step length as

$$\Delta t_{n+1}^2 = \xi\alpha^{\text{ref}} \Delta t_n^2 \underset{\mathbf{x}\in\Omega^h}{\text{MIN}} \frac{1}{|\Delta\lambda_n(\mathbf{x}) - \Delta\lambda_{n-1}(\mathbf{x})|}. \tag{78}$$

The result in Eq. (78) is subjected to the following restrictions:

$$\begin{aligned} \Delta t_{n+1} &\leqslant \eta \Delta t_n, \\ \Delta t_{n+1} &\leqslant \Delta t^{\max}, \end{aligned} \tag{79}$$

where the first Eq. (79) imposes an *ad hoc* limitation on the growth of the time step length, at two consecutive time steps, in terms of the acceleration factor η (for practical simulations purposes a value $\eta = 1.3$ has been used in the results presented here). The second Eq. (79) limits the maximum value of the time step length to the value, $\Delta t^{\max}$, determined by considerations on the precise tracing of the action–response curve.

Remark 3.6. Eq. (78) provides a *prediction* of the length of the time step at time step $n+1$ in terms of values at previous time steps n and $n-1$. This gives rise to a variable time-step length scheme (adaptive time stepping) where the estimation of the time step length does not imply any recalculation, apart from the one in Eq. (78), and does not translates into relevant additional computational costs. Therefore, Eq. (70) might not be exactly, but approximately, fulfilled. In Section 4.1 the degree of efficiency of the proposed scheme will be assessed.

Remark 3.7. The result in Eq. (78) admits a geometrical interpretation. In fact, taking into account Eqs. (74), (76) can be written:

$$\begin{aligned} (\Delta t_{n+1}^{\text{crit}}(\mathbf{x}))^2 &= \frac{e_{\alpha_{n+1}}(\mathbf{x}) \Delta t_n}{\left|\frac{\Delta\lambda_n(\mathbf{x})}{\Delta t_n} - \frac{\Delta\lambda_{n-1}(\mathbf{x})}{\Delta t_{n-1}}\right|} \\ &\cong \frac{\xi\alpha^{\text{ref}}}{|\dot{\alpha}_n(\mathbf{x}) - \dot{\alpha}_{n-1}(\mathbf{x})|} \Delta t_n \left(\cong \frac{\xi\alpha^{\text{ref}}}{|\ddot{\alpha}_n(\mathbf{x})|}\right). \end{aligned} \tag{80}$$

The value $|\dot{\alpha}_n - \dot{\alpha}_{n-1}| \cong |\ddot{\alpha}_n| \Delta t_n$ is a measure of the *apparent* curvature and smoothness of the algorithmic curve describing the internal variable evolution (see Fig. 4). Therefore, at non-smooth points it becomes larger ($|\dot{\alpha}_n - \dot{\alpha}_{n-1}| \neq 0$), resulting in a time step shortening from Eq. (80). For smooth points $|\dot{\alpha}_n - \dot{\alpha}_{n-1}| \cong 0$, and the time step length can be amplified (at a speed limited by the second Eq. (78)) up to the maximum value $\Delta t^{\max}$.

4. Numerical assessment

Some simple, but representative, simulations are used to assess numerically the properties of the IMPL-EX integration scheme, in terms of: (1) accuracy, (2) computational cost and (3) robustness.

4.1. Accuracy and computational cost analyzes

The target example consists of the plastic loading of a rectangular plate, with a center hole, which is vertically stretched (see Fig. 5a). For symmetry reasons only one fourth of the plate is analyzed. The assumed material model is a J2 plasticity model, in plane strain, with linear strain hardening. In order to prevent locking effects, a mixed quad element, with constant pressure and linear displacements [21,22], is used.

In Fig. 5a, the action–response curves, F–δ, obtained with both, implicit and IMPL-EX, integration schemes is presented. For large time steps, the IMPL-EX result exhibits an *overstress* effect, at the initiation of the plastic regime, which can be almost completely removed by reducing the time step length (see also Section 4.2). Plastic yielding spreads in a surface manner as it is shown in

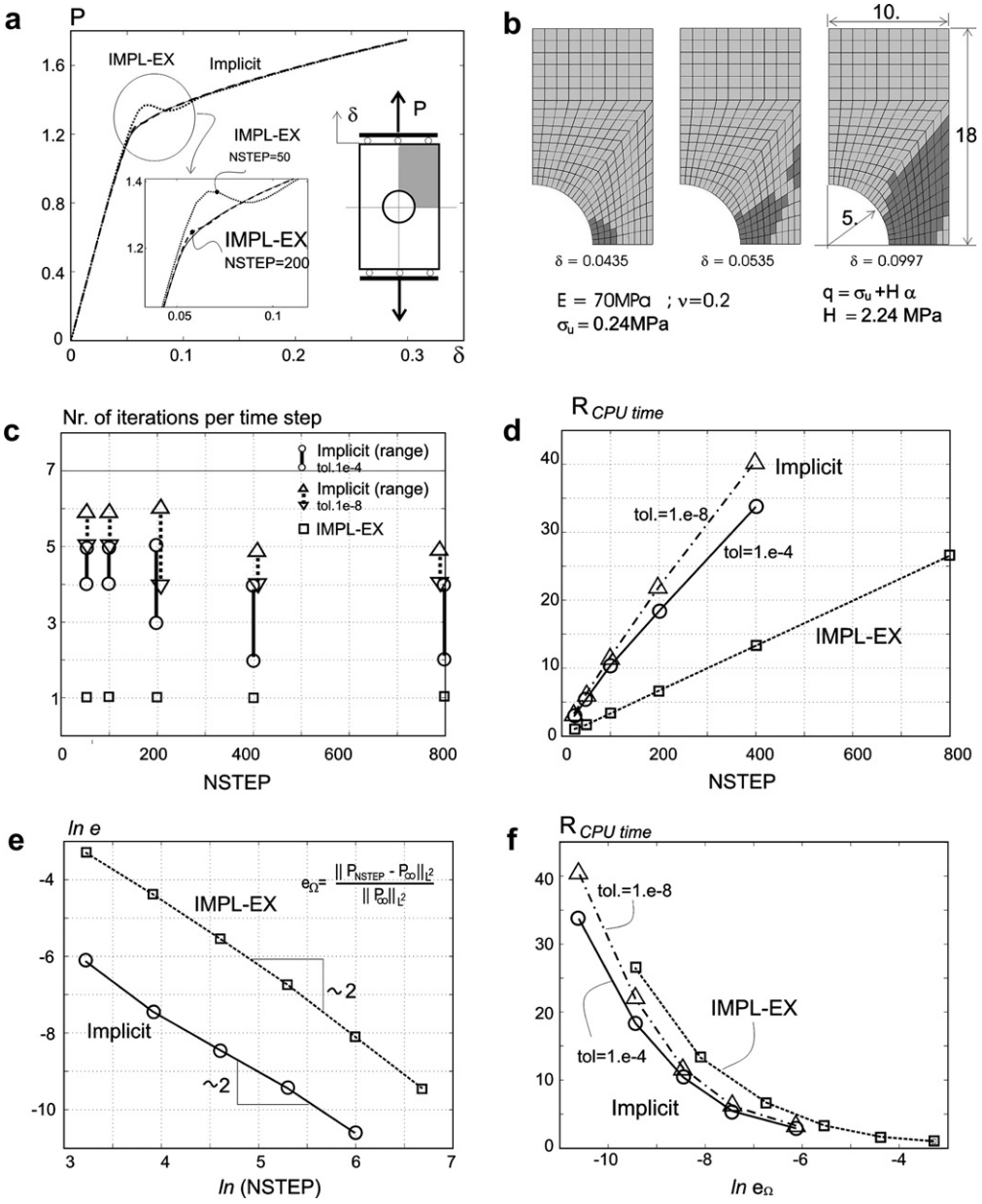

Fig. 5. Square plate with a center hole (strain hardening): (a) action–response, P–δ, curves, (b) propagation of plastic yielding for increasing stages of the analysis (darker elements are in plastic loading), (c) comparison of the (in average) required number of iterations per time step in terms of the considered number of times steps, *NSTEP*, (d) relative computational cost in terms of the number of time steps, (e) error (log–log) diagram and (f) relative computational cost in terms of the accepted integration error.

Fig. 5b and, as a result, the curve P–δ has a smooth aspect, regardless the number of elements of the finite element mesh.

In Fig. 5c, a comparison, in terms of the required number of iterations (average) per time step, in terms of the length of the time step (or, equivalently, the number of steps used for the complete analysis, NSTEP) is presented. As expected, the IMPL-EX scheme is step-linear and a unique iteration per time step is required, regardless the value of NSTEP. For the implicit case the number of iterations depends, naturally, on the adopted tolerance (tol.) for convergence of the Newton–Raphson procedure and it ranges from 4–6 (for large time step lengths) to 2–5 (for minor step lengths). This translates into a proportionally higher computational cost for the implicit scheme, when compared with the IMPL-EX scheme, for the same number of used time steps NSTEP, as displayed in Fig. 5d.

In Fig. 5e the analysis of the error provided by both integration procedures, in terms of the L_2 norms of the corresponding P–δ curves, referenced to the *exact* solution, obtained with a large value of NSTEP, is presented in a log–log diagram. There, it can be checked the linear accuracy of both schemes (slope of the curves $\cong 2$) and the higher absolute error associated to the IMPL-EX procedure. Combination of the results in Fig. 5d and e allows comparing the relative computational cost of both integration schemes, now in terms of the committed error (see Fig. 5f). Here the computational cost is slightly higher for the IMPL-EX scheme, although certainly depending on the adopted value of the tolerance, tol., in the implicit scheme.

Remark 4.1. The results in Fig. 5(c)–(f), show that, for this type of *robust problems*, i.e. as the implicitly integrated problem yields an *easy and robust* convergence of the Newton–Raphson procedure all along the analysis, the superior robustness of IMPL-EX is not a determinant issue. However, the superior computational cost vs. error of the implicit scheme might not always be the most determinant factor. In fact, if the length of the time-step is determined by the maximum acceptable error, probably there would not be any advantage in using the IMPL-EX scheme, since results in Fig. 5f will be determinant. However, if, as it often happens, the required number of time steps in the analysis is determined for the precise tracing of the action–response curve (i.e., obtaining a sufficiently high number of points) rather than for error considerations, then the IMPL-EX scheme can lead to substantial reduction, up to 3–4 times, of the final computational costs, as indicated by results in Fig. 5d.

4.2. Assessment of the error control

The rectangular plate problem in Fig. 5 is now used to assess the performance of the adaptive time stepping algorithm presented in Section 3.6.

Fig. 6a displays the maximum local relative error, $\mathrm{MAX}_{\mathbf{x}\in\Omega^h}(e_\alpha(\mathbf{x})/\alpha^{\mathrm{ref}})$, for different values of the tolerance, ξ. According to Eq. (70) the *ideal* value for this variable (implying a perfect error control) would keep that error equal to the value ξ at all times. As we normalize this error as $\mathrm{MAX}_{\mathbf{x}\in\Omega^h}(e_\alpha(\mathbf{x})/(\xi\alpha^{\mathrm{ref}}))$, like in Fig. 6b, that ideal value should be one, regardless the value of ξ.

Due to the predictive character of the proposed error control method, and the restrictions set in Eq. (79), this goal is not completely achieved, but the method shows a good efficiency in keeping the error below the tolerance limit, ξ, for most of analysis time interval. Only at initial times, where a large number of material points switch from elastic to inelastic regimes and the IMPL-EX extrapolation induces the aforementioned overstress (see Sections 3.6 and 4.1), at some points, and for a reduced number of time steps, the tolerance is surpassed; but soon after, the time step shortening reduces the maximum *local* error to the accepted limit.

As for the *global* error, e_Ω, in terms of the area under the force–displacement curve in Fig. 5a, Figs. 6c and d shows a clear correlation between the adopted tolerance, ξ, for the local error, e_α, and the global error, e_Ω, and, therefore, the capability of the proposed time stepping algorithm to control the global error as well.

4.3. Robustness and computability analyzes

Let us now consider the example displayed in Fig. 7. The rectangular plate in Fig. 7a is discretized in five rows of quads and pushed horizontally along the upper rows, inducing a pure shear stress state in the lowest row of elements. Two different finite element meshes are used, keeping constant the number of rows but using different number of columns: $n_{\mathrm{col}} = 10$, for the coarse mesh, and $n_{\mathrm{col}} = 80$ for the fine one. The material is assumed to follow a J_2 plasticity model in plane strain but, this time, it is equipped with exponential softening. Consequently, plastic yielding localizes in the lowest row and, unlike in the hardening case of Section 4.1, propagates linearly, in a sequential, element-by-element manner, along that row from the left to the right hand side (see Fig. 7b).

The action–response, F–δ, curve exhibits the peculiar shape displayed in Fig. 7c. On a global trend, exhibiting first a hardening branch, then a softening one and, in the last stretch, a strong snap-back, a large number of local oscillations and snap-backs are superposed. Indeed, every oscillation corresponds to the propagation of the plastic yielding from one Gauss-point to the next one, in the plastic path of elements in Fig. 7b. This is a simple, but very representative, case of a behavior that is frequently found in the context of more general and real-life problems in computational material failure mechanics aiming at simulation of propagation of cracks (in brittle materials [3]) or shear bands (in more ductile materials).

Using a finer mesh, $n_{\mathrm{col}} = 80$, alleviates the amplitude of the oscillations, but increases the number of them, as it can

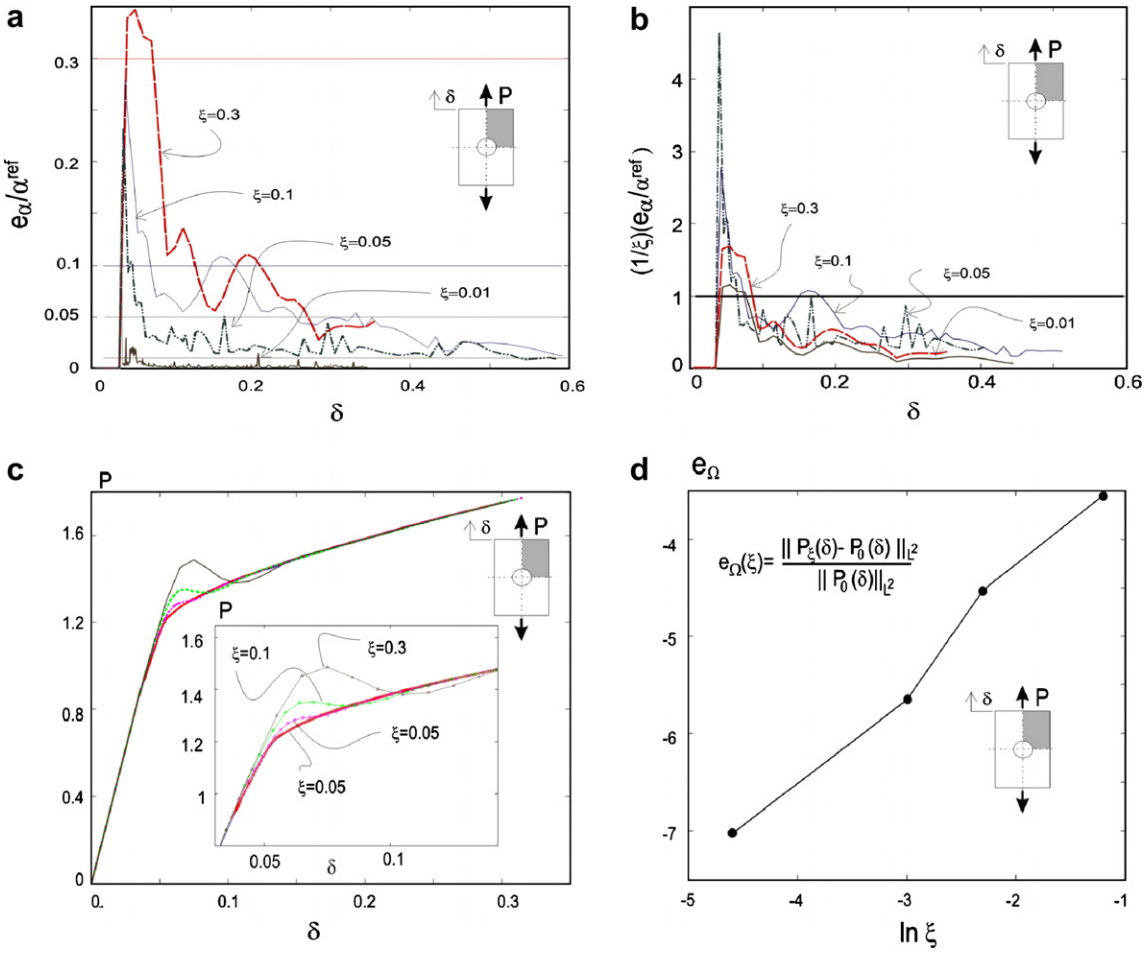

Fig. 6. Errol control assessment: (a) evolution of the maximum relative error for different values of the tolerance ξ, (b) normalized maximum error evolution, (c) and (d) correlation between the structural error and the tolerance parameter ξ.

be observed in Fig. 7d. The consequence of such a rough action–response path is a tremendous loss of computability. Indeed, those types of curves are not frequently displayed in the literature (although they have already been reported some times [23,24]) because they are extremely difficult to obtain by using classical implicit integration schemes. Only skillful *ad hoc* procedures [24], or a precise and *a priori* knowledge of the structural response allowing devising a specific load control procedure, allow tracing the complete action–response $F-\delta$.

In order to trace structural responses displaying snap-backs in the equilibrium curve, one has to resort to general continuation methods [10,25], like the *updated normal-plane method* described in Appendix 3. Using this procedure, the results presented in Figs. 7c and d have been obtained.

Using the implicit integration scheme, the Newton–Raphson iterative procedure losses convergence very soon (see Fig. 7c), before reaching the critical load. Neither reduction of the time step length nor the use of a finer mesh remedies that situation: the analysis could not be driven beyond the point shown in Fig. 7c. Reasons for this loss of convergence, were given in Section 2.2 and, Remark 2.3, on the basis of the loss of the positive definite character of the tangent algorithmic operator, which, after propagation, translates into singularity of the numerical tangent stiffness matrix. This is corroborated in Fig. 7e, displaying the condition number of the structural tangent stiffness matrix of the problem (after removing the prescribed degrees of freedom) along the iteration procedure. There it can be checked that the condition number of that matrix (defined as the ratio of the minimum and maximum eigenvalues) becomes zero, precisely at the point were the iteration procedure losses its convergence. Using more sophisticated displacement control methods does not substantially change the situation.

On the contrary, using the IMPL-EX integration scheme, the positive definite character of the tangent algorithmic operator is guaranteed (see Remark 3.2) and the condition number of the resulting algorithmic stiffness matrix remains positive all along the analysis, as it is displayed in Fig. 7f.

The consequence is that, even for this, very difficult, type of problems, the action response curve can be traced, *at no*

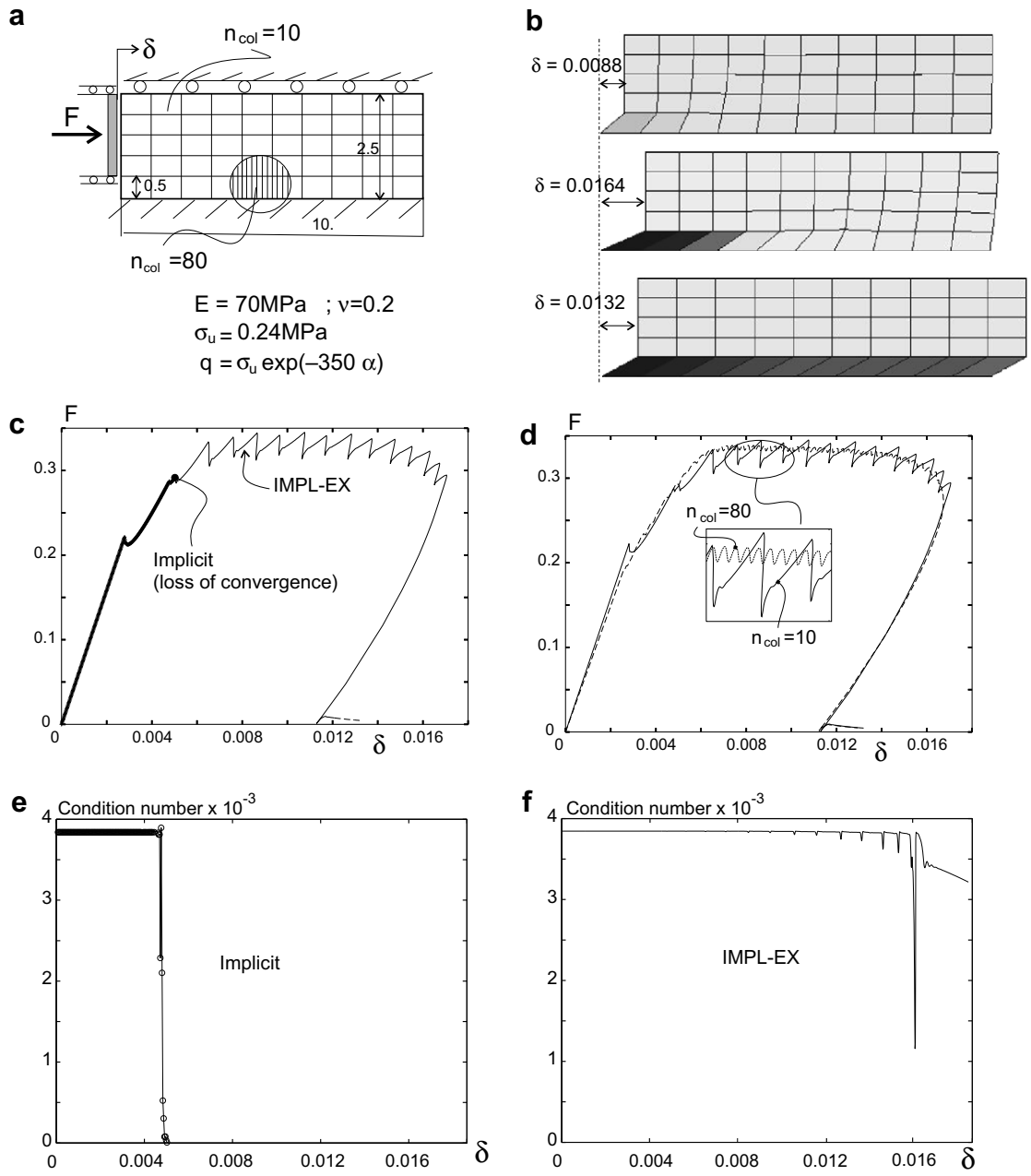

Fig. 7. Rectangular plate in shear (strain softening): (a) problem description and material properties, (b) propagation of plastic yielding for increasing stages of the analysis (in darker tones the plastic strain levels), (c) and (d) action–response, F–δ, curves and (e) and (f) evolution of the algorithmic stiffness matrix condition number.

difficulty, and at the only cost of a unique iteration per time step.

Indeed, the preceding example is a very simple one, selected to exemplify the computability problems found in numerical simulations involving soft and brittle materials. It is just a little sample of the difficulties found when facing real problems in two to three dimensions. However, according to the experiences of the authors in solving a large number of these problems in different fields [3], the benefits of the IMPL-EX integration scheme shown here, extend to general problems affected by similar computability difficulties and provide dramatic improvements in terms of the robustness of the numerical simulations and of the resulting computational costs.

5. Application to contact/friction interfaces

Contact-friction phenomena are responsible for the nonlinear behavior in many solid mechanics problems. Very often, the contact and friction conditions for the contacting

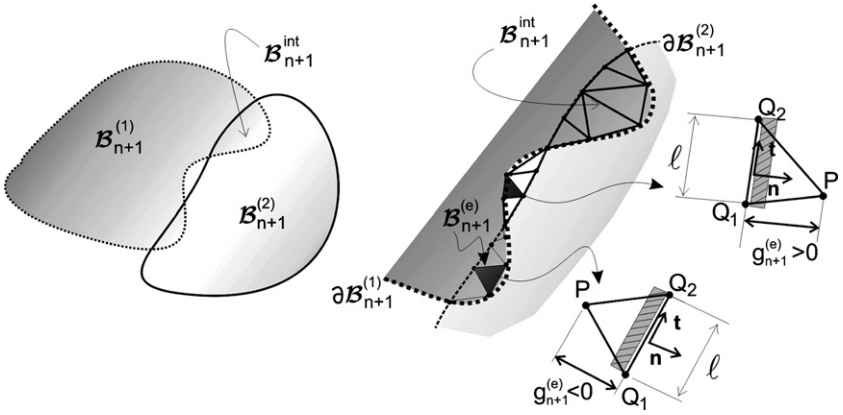

Fig. 8. Contact-friction interface.

bodies, are imposed at specific interfaces, via non-linear constitutive models [26] which, in turn, can be integrated using the previous IMPL-EX schemes. Again, this allows enjoying the aforementioned advantages of robustness and, if there is no other source of non-linearity, the iterative solving algorithm becomes step-linear.

5.1. Contact-friction model

For the sake of simplicity, let us restrict the problem to the two-dimensional case. Let us, then, consider, at time step $n+1$, two contacting bodies, $\mathscr{B}^{(1)}_{n+1}$ and $\mathscr{B}^{(2)}_{n+1}$, whose boundaries, $\partial\mathscr{B}^{(1)}_{n+1}$ and $\partial\mathscr{B}^{(2)}_{n+1}$, are connected by a contact interface, $\mathscr{B}^{\text{int}}_{n+1}$, which is assumed to remain constant along the time step (see Fig. 8). This contact interface can be either a line, discretized in one-dimensional finite elements [26], or a surface, discretized in two-dimensional elements [27]. The way that this contact interface is constructed is not relevant for our purposes, whenever it defines the distances (gaps) from points of one boundary with respect to the other (i.e. $g^{(e)}_{n+1}(\mathbf{x})\forall\mathbf{x}\in\mathscr{B}^{\text{int}(e)}_{n+1}$ for the interface element, e, occupying the interface elemental domain $\mathscr{B}^{\text{int}(e)}_{n+1}$), and the incremental relative displacements of both boundaries in the tangential direction, $\Delta v^{(e)}_{n+1}(\mathbf{x}) = v^{(e)}_{n+1}(\mathbf{x}) - v^{(e)}_{n}(\mathbf{x})\ \forall\mathbf{x}\in\mathscr{B}^{\text{int}(e)}_{n+1}$.

Let us also assume that both $g^{(e)}_{n+1}$ and $\Delta v^{(e)}_{n+1}$ can be appropriately rephrased into strain-like measures by dividing them by a representative length, ℓ (see Fig. 8), i.e.:

$$\varepsilon^{(e)}_{n+1} = \frac{g^{(e)}_{n+1}}{\ell}; \quad \Delta\gamma^{(e)}_{n+1} = \frac{\Delta v^{(e)}_{n+1}}{\ell}. \tag{81}$$

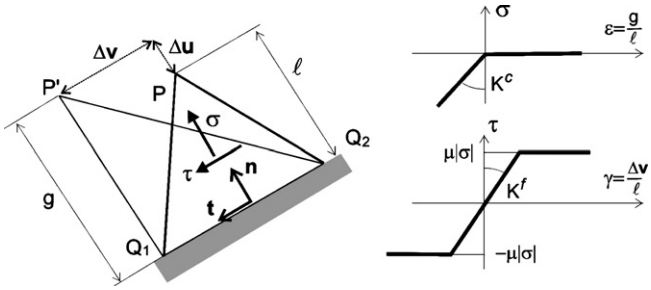

Fig. 9. Contact-friction constitutive models at the interface.

Then, the contact condition, $g^{(e)}_{n+1}(\mathbf{x}) \geqslant 0\ \forall\mathbf{x}\in\mathscr{B}^{\text{int}(e)}_{n+1}$, precluding boundary interpenetration, and the friction effects between both bodies, can be imposed via an appropriate continuum (stress vs. strain) contact-friction model, defined at the interface, and relating the normal and tangential stresses, (σ,τ) with the normal and tangential strains, (ε,γ), i.e.:

$$\{\boldsymbol{\sigma}\} = \begin{Bmatrix}\sigma\\ \tau\end{Bmatrix}; \quad \{\boldsymbol{\varepsilon}\} = \begin{Bmatrix}\varepsilon\\ \gamma\end{Bmatrix},$$
$$\{\dot{\boldsymbol{\sigma}}\} = [\mathbf{C}^{cf}]\cdot\{\dot{\boldsymbol{\varepsilon}}\}; \quad [\mathbf{C}^{cf}] = \begin{bmatrix}C_{\sigma\sigma} & C_{\sigma\tau}\\ C_{\tau\sigma} & C_{\tau\tau}\end{bmatrix}. \tag{82}$$

Those stresses, once spatially integrated, translate into a, self-equilibrated, nodal forces at the contact/friction interface, $\mathscr{B}^{\text{int}}_{n+1}$, and, then, into contact/friction nodal reactions at the contacting boundaries $\partial\mathscr{B}^{(1)}_{n+1}$ and $\partial\mathscr{B}^{(2)}_{n+1}$.

Using a penalty-like method, to impose contact, and a Coulomb model, to account for the frictional effects, that interface continuum constitutive model reads (see Fig. 9):

	Contact model	Friction model
Constitutive equation	$\sigma = E(\beta)\varepsilon$	$\dot{\tau} = K^f(\dot{\gamma} - \dot{\gamma}^{\text{slip}})$ (83)
	$E(\beta) = \begin{cases} 0 & \text{for } \beta \geqslant 0 \\ K_c & \text{for } \beta < 0 \end{cases}$	

(continued on next page)

	Contact model	Friction model	
Internal variables evolution	$\dot{\beta} = \dot{\varepsilon}(g) \quad \beta\|_{t=0} = \varepsilon(g^{(0)})$	$\dot{\gamma}^{\text{slip}} = \dot{\tilde{\lambda}} m; \quad m = \dfrac{\partial f(\tau)}{\partial \tau} = \text{sign}(\tau) = \dfrac{\tau}{\|\tau\|}$ $\dot{\gamma}^{\text{slip}} = \dot{\lambda}\tau; \quad \dot{\lambda} = \dot{\tilde{\lambda}}\|\tau\| \begin{cases} \text{rephrased} \\ \text{slip flow} \end{cases}$ $\dot{\alpha} = \dot{\lambda}; \quad \alpha\|_{t=0} = 0$	(84)
Slip function		$f(\tau) \equiv \|\tau\| - \tau_y; \quad \tau_y = \mu\|\sigma\|$	(85)
Stick–slip condition		$\dot{\lambda} \geqslant 0; \quad f \leqslant 0; \quad \dot{\lambda} f = 0$	(86)
Constitutive tangent tensor	$C_{\sigma\sigma} = E^{\tan}(\beta); \quad C_{\boldsymbol{\sigma}\tau} = 0$ $E^{\tan}(\beta) = \begin{cases} 0 & \text{for } \beta \geqslant 0 \\ K^c & \text{for } \beta < 0 \end{cases}$	$C_{\tau\tau} = G^{\tan}; \quad C_{\tau\sigma} = \mu \text{sign}(\sigma) E^{\tan}$ $G^{\tan}(\gamma) \to \begin{cases} G^{(\text{stick})} \equiv K^f \\ G^{(\text{slip})} \equiv 0 \end{cases}$	(87)

where K^c and K^f are, respectively, the contact and the friction-stick penalty parameters. In the contact model, a dummy internal variable β ($\beta \equiv \varepsilon$) has been introduced to make the model ready for the IMPL-EX integration. On the other hand, the friction model displays a complete elasto-plastic format which makes it completely comparable to the one in Eqs. (9)–(15).

In Eq. (84), the original slip flow, $\dot{\gamma}^{\text{slip}} = \dot{\tilde{\lambda}} m = \dot{\tilde{\lambda}} \text{sign}(\tau)$, has been rephrased to, $\dot{\gamma}^{\text{slip}} = \dot{\lambda}\tau$ in order to generate a linear plastic flow, amenable to enjoy the step-linear character from the IMPL-EX integration scheme (see Remarks 2.2, 3.2 and Appendix 1).

Remark 5.1. The contact model in Eqs. (83)–(87) imposes, via the contact penalty constant, K_c, the condition $\varepsilon(\mathbf{x}) \geqslant 0 \ \forall \mathbf{x} \in \mathscr{B}^{\text{int}}$ and, therefore, through Eq. (81), the contact condition $g(\mathbf{x}) \geqslant 0 \ \forall \mathbf{x} \in \mathscr{B}^{\text{int}}$. On the other hand, the contact and friction models are coupled, in one direction, since the normal stress, σ determined in the contact model, appears in the slip function, g (in Eq. (85) for the friction model). Therefore, the contact model has to be firstly integrated and, then, the friction model can be solved.

5.1.1. Implicit integration

The contact-friction model in Eqs. (83)–(87) can be integrated using backward-Euler integration according to Table 6.

In Eq. (93), the result $\frac{\partial \|(\bullet)\|}{\partial(\bullet)} = \text{sign}(\bullet)$ has been considered.

5.1.2. IMPL-EX integration

The implicit–explicit integration of the above contact-friction model follows the scheme described in Section 3. The implicit stage has been described in Table 6. Then, the explicit stage involves the steps as in Table 7.

Remark 5.2. Unlike in the implicit integration case, the non-diagonal terms $\widetilde{C}^{\text{alg}}_{\tau\boldsymbol{\sigma}_{n+1}}$ and $\widetilde{C}^{\text{alg}}_{\boldsymbol{\sigma}\tau_{n+1}}$, in the tangent operator in Eq. (82), are null. Besides, the term $\widetilde{C}^{\text{alg}}_{\tau\tau_{n+1}} = \widetilde{G}^{\text{alg}}_{n+1} = K^f/(1 + K^f \Delta\tilde{\lambda}_{n+1})$ is positive according to Eq. (98). Therefore the algorithmic tangent operator obtained from the IMPL-EX integration:

$$[\widetilde{\mathbf{C}}^{cf}_{n+1}] = \begin{bmatrix} \widetilde{C}^{\text{alg}}_{\sigma\sigma_{n+1}} & \widetilde{C}^{\text{alg}}_{\sigma\tau_{n+1}} \\ \widetilde{C}^{\text{alg}}_{\tau\sigma_{n+1}} & \widetilde{C}^{\text{alg}}_{\tau\tau_{n+1}} \end{bmatrix} = \begin{bmatrix} E(\tilde{\beta}_{n+1}) & 0 \\ 0 & \frac{K^f}{1+K^f\Delta\tilde{\lambda}_{n+1}} \end{bmatrix} \tag{99}$$

is step-constant, symmetric and semi-positive definite. Through arguments similar to the ones in Section 2.2 and Remark 2.3, this should translate into a larger robustness of the numerical simulations.

5.2. A representative simulation: 2D cylinder roller contact

A classical contact/friction benchmark [28] is here used to assess the performance of the IMPL-EX algorithm in Table 7. A steel cylinder is pressed against an aluminum block, as shown in Fig. 10a, by means of a vertical force: $F = 37.0$ kN. The cylinder and the block are assumed to behave as linear elastic materials with parameters described in the same Figure. A 2D plane strain case is considered with a friction coefficient: $\mu = 0.1$.

Fig. 10b shows the corresponding contact pressure and frictional stress along the x-coordinate. There, the contact pressure analytical solution corresponds to the classical Hertz's solution [29] for the frictionless case.

Table 6
Implicit integration scheme for the contact-friction model

	Contact model	Friction model	
DATA	ε_{n+1}	$\Delta\gamma_{n+1,}, \tau_n, \alpha_n$	(88)
1. Compute trial values		$\tau_{n+1}^{\text{trial}} = \tau_n + K^f \Delta\gamma_{n+1}$ $f_{n+1}^{\text{trial}} \equiv \|\tau_{n+1}^{\text{trial}}\| - \tau_y; \quad \tau_y = \mu\|\boldsymbol{\sigma}_{n+1}\|$	(89)
2. Compute slip multiplier		if $f_{n+1}^{\text{trial}} < 0 \rightarrow \Delta\lambda_{n+1} = 0 \rightarrow$ stick else if $f_{n+1}^{\text{trial}} \geqslant 0 \rightarrow$ $\lambda_{n+1} = \dfrac{f_{n+1}^{\text{trial}}}{K^f \tau_y} \rightarrow$ slip	(90)
3. Update internal variables	$\beta_{n+1} = \underbrace{\beta_n}_{\varepsilon_n} + (\varepsilon_{n+1} - \varepsilon_n)$ $= \boldsymbol{\varepsilon}_{n+1}$	$\alpha_{n+1} = \alpha_n + \Delta\lambda_{n+1}$	(91)
4. Update stresses	$\sigma_{n+1} = \begin{cases} 0 & \text{for } \varepsilon_{n+1} \geqslant 0 \\ K^c \varepsilon_{n+1} & \text{for } \varepsilon_{n+1} < 0 \end{cases}$	$\tau_{n+1} = \begin{cases} \tau_{n+1}^{\text{trial}}(\text{stick}) \\ \frac{\tau_{n+1}^{\text{trial}}}{1+K^f\Delta\lambda_{n+1}} = \text{sign}(\tau_{n+1}^{\text{trial}})\tau_y(\text{slip}) \end{cases}$	(92)
5. Compute algorithmic tangent operator	$C_{\sigma\sigma_{n+1}}^{\text{alg}} = E_{n+1}^{\text{alg}}; \quad C_{\sigma\tau_{n+1}}^{\text{alg}} = 0$ $E_{n+1}^{\text{alg}}(\beta_{n+1}) = \dfrac{\partial\boldsymbol{\sigma}_{n+1}}{\partial\boldsymbol{\varepsilon}_{n+1}}$ $= \begin{cases} 0 & \text{for } \boldsymbol{\varepsilon}_{n+1} \geqslant 0 \\ K^c & \text{for } \boldsymbol{\varepsilon}_{n+1} < 0 \end{cases}$	$C_{\tau\tau_{n+1}}^{\text{alg}} = G_{n+1}^{\text{alg}}$ $C_{\tau\sigma_{n+1}}^{\text{alg}} = \mu\text{sign}(\tau_{n+1}^{\text{trial}})\text{sign}(\sigma_{n+1})E_{n+1}^{\text{alg}}$ $G_{n+1}^{\text{alg}}(\gamma_{n+1}) = \dfrac{\partial\tau_{n+1}}{\partial\gamma_{n+1}} = \begin{cases} G^{(\text{stick})} = K^f \\ G^{(\text{slip})} = 0 \end{cases}$	(93)

For the frictional case, Fig. 10c compares also the frictional stress, τ, along the x-coordinate, obtained for a certain, low, value of the friction/stick penalty parameter, K^f, in Eq. (87).

It is well known that contact/friction constraints are better imposed for high values of the contact and friction penalties, K^c and K^f. Therefore, working with these high values is generally intended.

However, it is also known that, for implicit integrations, robustness issues set clear limitations on those values. In order to asses the additional robustness provided by the proposed IMPL-EX algorithm, in front of the classical implicit integration, in Table 8 comparative results, of the relative computational cost and convergence, for increasing values of the ratio K^f/E_c, are presented. There, it is displayed that, as that ratio increases, the number of required iterations for the implicit integration scheme also grows and, therefore, so does the ratio implicit/IMPL-EX, of the computational costs. Most importantly, it appears that there is an upper limit for which the implicit integration scheme no longer converges, whereas that limitation for the IMPL-EX scheme can be much further extended.

6. Concluding remarks

Along the previous sections new implicit/explicit integration schemes to increase computability in non-linear solid mechanics problems have been analyzed. The main conclusions, obtained from that study, are:

- In many cases, when implemented in a classical Newton–Raphson iterative method, the proposed integration scheme render the resulting non-linear problem step-linear (i.e. the problem takes a unique iteration per time step to converge). For other cases, some additional modifications (rephrasing the plastic multiplier or linearization of the plastic flow) retrieve that property for the iterative method.
- The order of accuracy of implicit integration schemes is also kept (generally first order accuracy). However, the committed error is larger. A method to *a priori* control

Table 7
IMPL-EX explicit stage for the contact friction model

	Contact model	Friction model	
DATA	$\boldsymbol{\varepsilon}_{n+1}, \beta_n, \Delta\beta_n$	$\Delta\gamma_{n+1}, \tau_n, \alpha_n, \Delta\alpha_n$	(94)
1. Explicit extrapolation	$\tilde{\beta}_{n+1} = \beta_n + \frac{\Delta t_{n+1}}{\Delta t_n}\Delta\beta_n$ $\Delta\beta_n = \beta_n - \beta_{n-1}$	$\tilde{\alpha}_{n+1} = \alpha_n + \frac{\Delta t_{n+1}}{\Delta t_n}\Delta\alpha_n$ $\Delta\alpha_n = \alpha_n - \alpha_{n-1} = \Delta\lambda_n$	(95)
2. Compute slip multiplier		$\Delta\tilde{\lambda}_{n+1} = \tilde{\alpha}_{n+1} - \alpha_n = \frac{\Delta t_{n+1}}{\Delta t_n}\Delta\lambda_n \geqslant 0$	(96)
3. Update stresses	$\tilde{\sigma}_{n+1} = E(\tilde{\beta}_{n+1})\varepsilon_{n+1}$	$\tilde{\tau}_{n+1} = \tau_{n+1}^{\text{trial}} - K^f\Delta\tilde{\lambda}_{n+1}\tilde{\tau}_{n+1}^{\rightarrow}$ $\tilde{\tau}_{n+1} = \frac{\tau_{n+1}^{\text{trial}}}{1 + K^f\Delta\tilde{\lambda}_{n+1}}$	(97)
4. Compute algorithmic tangent operator	$\widetilde{C}^{\text{alg}}_{\sigma\sigma_{n+1}} = E(\tilde{\beta}_{n+1})$ $\widetilde{C}^{\text{alg}}_{\sigma\tau_{n+1}} = 0$ $E(\tilde{\beta}_{n+1}) = \begin{cases} 0 & \text{for } \tilde{\beta}_{n+1} \geqslant 0 \\ K^c & \text{for } \tilde{\beta}_{n+1} < 0 \end{cases}$	$\widetilde{C}^{\text{alg}}_{\tau\tau_{n+1}} = \widetilde{G}^{\text{alg}}_{n+1};\ \widetilde{C}^{\text{alg}}_{\tau\sigma_{n+1}} = 0$ $\widetilde{G}^{\text{alg}}_{n+1} = \frac{\partial\tilde{\tau}_{n+1}}{\partial\gamma_{n+1}} = \frac{K^f}{1 + K^f\Delta\tilde{\lambda}_{n+1}}$	(98)

that error, and to keep it below the desired bounds (automatic time stepping algorithm) has been presented and assessed.

- The algorithm supplies additional computability capabilities with respect to more classical integration schemes. Robustness is dramatically increased for those solid material problems where alternative implicit schemes exhibit lack of computability: i.e. constitutive models equipped with strain softening, soft materials models, contact/friction interfaces for high friction penalty parameters, etc. Moreover, the symmetric character of the resulting algorithmic constitutive tensor, provides additional savings, as for the required computational times when non-symmetric models are considered, since symmetric solvers can be used.
- For those cases where the traditional implicit schemes are robust enough, the proposed schemes can sometimes provide savings in the computational cost too, whenever the requirements on the precise tracing of the action–response curve determine the maximum time step length.
- Although not considered here, the inclusion of non-linear kinematics in the solid mechanics problem (finite strains) does not change the central conclusions of this work, in terms of the computability and computational robustness provided by the IMPL-EX scheme. However, due to the geometrical non-linearity, the step linear character of the iterative solution no longer holds in this case, though the number of required iterations to converge is considerably reduced.
- As for stability properties: analytical proofs of the unconditional stability of the proposed scheme are not available yet, and they are object of current research. However, after using the proposed methodology in a large number of numerical simulations, with linear and non-linear kinematics and with a wide number of constitutive models, the authors can report that classical numerical instabilities due to error propagation have not been found, even when large time step lengths, typical of unconditionally stable implicit methods, have been used.

Although, for the sake of brevity, in this work the method has been numerically assessed by simple, two-dimensional, cases, the authors have applied it to many different problems in computational solid mechanics [3,27,30] either in 2D or 3D cases and for infinitesimal and finite strains. These experiences have confirmed the conclusions reported above. Moreover, the method is not restricted to the families of constitutive models tackled here, which have been chosen for exemplification reasons. Indeed, the authors have applied it successfully to elasto-plastic pressure dependent models [31], and they believe that, keeping the essentials presented here, it can be used for a large

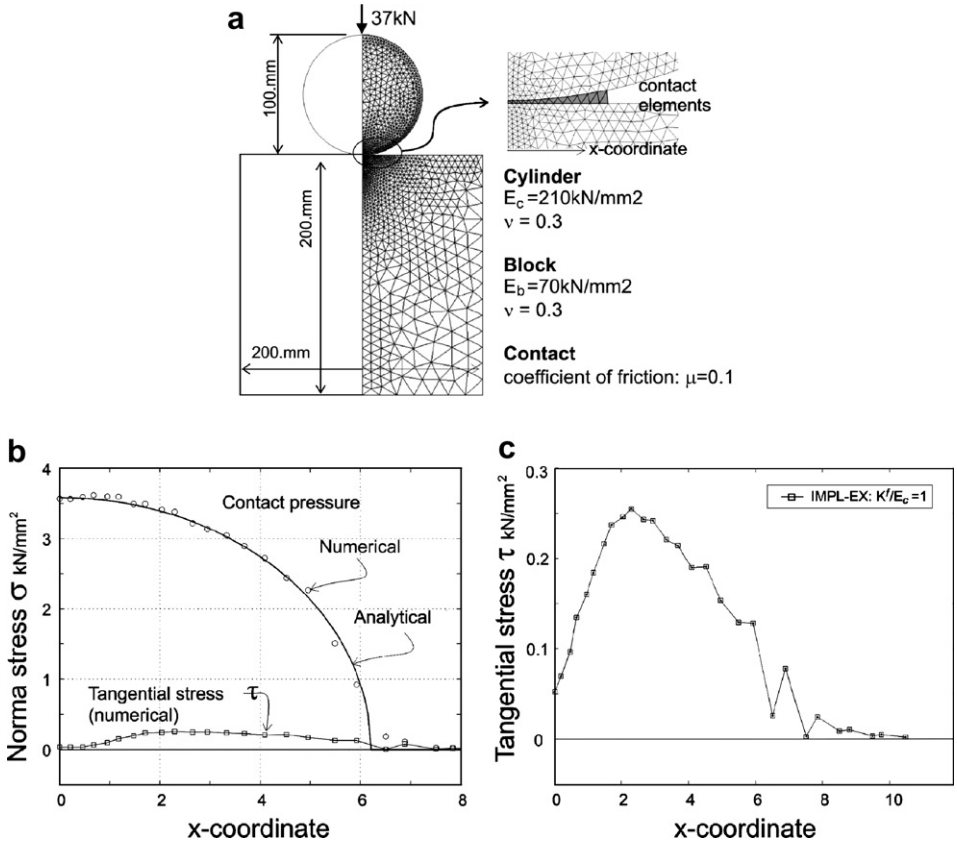

Fig. 10. 2D cylinder roller contact with friction.

Table 8
Comparative analysis between implicit and IMPL-EX algorithms for solving the 2D cylinder roller contact with friction

	CPU time cost: Implicit (tol.10^{-4})/ IMPL-EX	Maximum number of iterations required by the implicit scheme
$K^f/E_c = 10^0$	115./36. ≅ 3.2	6
$K^f/E_c = 10^1$	144./36. ≅ 4.0	9
$K^f/E_c = 10^2$	150./36. ≅ 4.2	18
$K^f/E_c = 10^3$	−/36	Implicit not converged after 50 iterations
$K^f/E_c = 10^4$	−/36	Implicit not converged after 50 iterations

Total number of time steps: 200.
E_c : Young's modulus of the cylindrical punch ($E_c = 210\,\mathrm{kN/mm^2}$).
K^f : stick-penalty parameter for the friction model ($K^c = K^f$).

variety, of rate-dependent and rate independent, inelastic constitutive models and solid mechanics problems.

Acknowledgements

Financial support from the Spanish Ministry of Science and Technology through grants BIA2005-09250-C03-03 and DPI 2004-0766-C02-02 is gratefully acknowledged.

The authors wish also to thank Prof. R. Codina, from the Technical University of Catalonia (UPC), for the technical discussions related to some aspects of this paper.

Appendix 1. Rephrasing and linearizing the plastic flow: J2 plasticity example

Let us consider the classical J2 plasticity, the yield function, $g(\boldsymbol{\sigma}, q)$ and the following three different options for the plastic flow formulation:

A.1. Original flow

The yield function, $g(\boldsymbol{\sigma}, q)$, and plastic flow tensor, $\mathbf{m}$, in Eqs. (12) and (10) are defined as

$$\begin{aligned} g(\boldsymbol{\sigma}, q) &= \Phi(\boldsymbol{\sigma}) - q \equiv \sqrt{\frac{3}{2}}\underbrace{\sqrt{\mathbf{S}:\mathbf{S}}}_{\|\mathbf{S}\|} - q, \\ \mathbf{S} &= \mathrm{dev}(\boldsymbol{\sigma}) = \mathbf{I}^{\mathrm{dev}} : \boldsymbol{\sigma}, \\ \mathbf{m}(\boldsymbol{\sigma}) &= \frac{\partial g(\boldsymbol{\sigma}, q)}{\partial \boldsymbol{\sigma}} = \frac{\partial \Phi(\boldsymbol{\sigma})}{\partial \boldsymbol{\sigma}} = \sqrt{\frac{3}{2}}\frac{\mathbf{S}}{\|\mathbf{S}\|}, \\ \mathbf{A}(\boldsymbol{\sigma}) &= \frac{\partial \mathbf{m}(\boldsymbol{\sigma})}{\partial \boldsymbol{\sigma}} = \sqrt{\frac{3}{2}}\frac{1}{\|\mathbf{S}\|}\left(\mathbf{I}^{\mathrm{dev}} - \frac{\mathbf{S}}{\|\mathbf{S}\|}\otimes\frac{\mathbf{S}}{\|\mathbf{S}\|}\right), \end{aligned} \tag{100}$$

where $\mathbf{A}$ is the Hessian of the equivalent uniaxial stress, $\Phi(\boldsymbol{\sigma})$ (Von Misses stress), $\mathbf{I}^{\mathrm{dev}} = \mathbf{I} - \frac{1}{3}(\mathbf{1}\otimes\mathbf{1})$ is the deviatoric operator, and $\mathbf{I}$ and $\mathbf{1}$ stand, respectively, for the fourth and second order symmetric unit tensors. Clearly $\mathbf{A}(\boldsymbol{\sigma})$, which can be proven to be semi-positive definite, is not constant and, therefore, $\mathbf{m}(\boldsymbol{\sigma})$ cannot be included in the linear plastic flow case of Eq. (27). Therefore its

IMPL-EX integration *will not yield a step linear problem* (see Remark 3.2). The flow equations to be integrated are (see Eqs. (10) and (12)):

$$\begin{aligned}
&\dot{\boldsymbol{\sigma}} = \mathbf{C} : (\dot{\boldsymbol{\varepsilon}} - \dot{\boldsymbol{\varepsilon}}^p),\\
&\dot{\boldsymbol{\varepsilon}}^p = \dot{\lambda}\mathbf{m}(\boldsymbol{\sigma}) = \dot{\lambda}\sqrt{\frac{3}{2}}\frac{\mathbf{S}}{\|\mathbf{S}\|},\\
&\dot{\alpha} = \dot{\lambda},\\
&\dot{q} = H\dot{\alpha},
\end{aligned} \tag{101}$$

where the plastic multiplier is solved by imposing plastic consistency i.e.: $g(\boldsymbol{\sigma}(\lambda), q(\lambda)) = 0$ (see Eq. (23)).

A.2. Rephrased plastic flow (linear plastic flow)

Let us, now, rephrase the flow Eq. (101) as

$$\begin{aligned}
&\dot{\boldsymbol{\sigma}} = \mathbf{C} : (\dot{\boldsymbol{\varepsilon}} - \dot{\boldsymbol{\varepsilon}}^p),\\
&\dot{\boldsymbol{\varepsilon}}^p = \underbrace{\dot{\lambda}\sqrt{\frac{3}{2}}\frac{1}{\|\mathbf{S}\|}}_{\dot{\bar{\lambda}}}\mathbf{S} = \dot{\bar{\lambda}}\bar{\mathbf{m}}(\boldsymbol{\sigma}),\\
&\bar{\mathbf{m}}(\boldsymbol{\sigma}) = \mathbf{S}; \quad \dot{\bar{\lambda}} = \sqrt{\frac{3}{2}}\frac{\dot{\lambda}}{\|\mathbf{S}\|},\\
&\dot{\bar{\alpha}} = \dot{\bar{\lambda}}; \quad \dot{\bar{\alpha}} = \sqrt{\frac{3}{2}}\frac{\dot{\alpha}}{\|\mathbf{S}\|},\\
&\dot{q} = \overline{H}\dot{\bar{\alpha}}; \quad \overline{H} = H\sqrt{\frac{2}{3}}\|\mathbf{S}\|,
\end{aligned} \tag{102}$$

where, again, the plastic multiplier, $\bar{\lambda}$, is solved by imposing plastic consistency i.e.: $g(\boldsymbol{\sigma}(\bar{\lambda}), q(\bar{\lambda}) = 0$ (see Eq. (23)).

Clearly, the only difference between Eqs. (101) and (102) is the change of variables $(\alpha, \dot{\lambda})$ by $(\bar{\alpha}, \dot{\bar{\lambda}})$. Therefore, the integrated results in terms of the generalized stresses, $(\boldsymbol{\sigma}, q)$, must be the same in both cases.

However, as for the IMPL-EX integration procedure the change is substantial. From Eq. (102):

$$\overline{\mathbf{A}}(\boldsymbol{\sigma}) = \frac{\partial \bar{\mathbf{m}}(\boldsymbol{\sigma})}{\partial \boldsymbol{\sigma}} = \frac{\partial \mathbf{S}}{\partial \boldsymbol{\sigma}} = \mathbf{I}^{\text{dev}} \tag{103}$$

and, now, $\overline{\mathbf{A}}(\boldsymbol{\sigma})$, is constant. Therefore, $\bar{\mathbf{m}}(\boldsymbol{\sigma})$ falls into the linear plastic flow case of Eq. (27) and the IMPL-EX integration *will yield a step linear problem* enjoying the advantages mentioned in Remark 3.2.

A.3. Linearization of the plastic flow

Let us now consider the linearization of the plastic flow, in Eq. (100), at the explicit stage of the IMPL-EX scheme, at time step $n+1$, according to Eq. (46):

$$\begin{aligned}
&\tilde{\mathbf{m}}(\boldsymbol{\sigma}_{n+1}) = \mathbf{m}(\boldsymbol{\sigma}_n) + \mathbf{A}(\boldsymbol{\sigma}_n) : \Delta\tilde{\boldsymbol{\sigma}}_{n+1}; \quad \Delta\tilde{\boldsymbol{\sigma}}_{n+1} = \tilde{\boldsymbol{\sigma}}_{n+1} - \boldsymbol{\sigma}_n,\\
&\widetilde{\mathbf{A}}_{n+1} = \frac{\partial \tilde{\mathbf{m}}_{n+1}}{\partial \tilde{\boldsymbol{\sigma}}_{n+1}} = \mathbf{A}(\boldsymbol{\sigma}_n) = \mathbf{A}_n.
\end{aligned} \tag{104}$$

Substitution of Eq. (100) into Eq. (104) yields:

$$\begin{aligned}
\tilde{\mathbf{m}}_{n+1} &= \sqrt{\frac{3}{2}}\frac{\mathbf{S}_n}{\|\mathbf{S}_n\|} + \mathbf{A}_n : (\widetilde{\mathbf{S}}_{n+1} - \mathbf{S}_n)\\
&= \mathbf{A}_n : \mathbf{S}_{n+1} + \sqrt{\frac{3}{2}}\frac{\mathbf{S}_n}{\|\mathbf{S}_n\|},
\end{aligned} \tag{105}$$

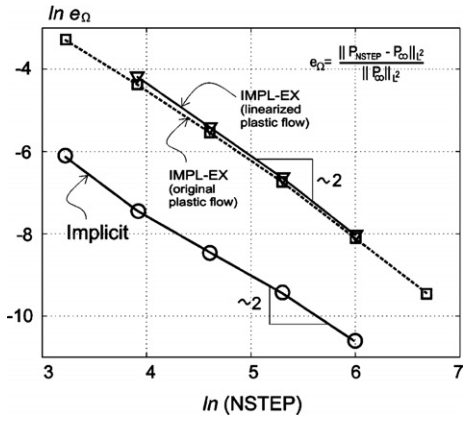

Fig. 11. Error diagram for the J2 plasticity model in Fig. 5. Comparison of different integration schemes.

where the result $\mathbf{A}_n : \mathbf{S}_n = \mathbf{0}$ has been used. Eq. (105) is a specific case of a linear plastic flow in Eq. (27). Therefore, the IMPL-EX integration*yields a step-linear problem*.

More specifically, for this J2 plasticity model, after some algebraic manipulation, Eqs. (43) and (44), read:

$$\begin{aligned}
&\tilde{\boldsymbol{\sigma}}_{n+1} = \sigma^m_{n+1}\mathbf{1} + \widetilde{\mathbf{S}}_{n+1} \rightarrow\\
&\begin{cases}
\sigma^m_{n+1} = \sigma^m_n + K(\text{tr}(\boldsymbol{\varepsilon}_{n+1}) - \text{tr}(\boldsymbol{\varepsilon}_n)),\\
\widetilde{\mathbf{S}}_{n+1} = (\mathbf{I} + 2\mu\Delta\tilde{\bar{\lambda}}_{n+1}\mathbf{A}_n)^{-1} : \left(\mathbf{S}^{\text{trial}}_{n+1} - 2\mu\Delta\tilde{\bar{\lambda}}_{n+1}\sqrt{\frac{3}{2}}\frac{\mathbf{S}_n}{\|\mathbf{S}_n\|}\right),\\
\mathbf{S}^{\text{trial}}_{n+1} = \mathbf{S}_n + 2\mu\text{dev}(\boldsymbol{\varepsilon}_{n+1} - \boldsymbol{\varepsilon}_n),
\end{cases}\\
&\widetilde{\mathbf{C}}^{\text{alg}(\text{load.})}_{n+1} = \frac{\partial\tilde{\boldsymbol{\sigma}}_{n+1}}{\partial\boldsymbol{\varepsilon}_{n+1}} = K(\mathbf{1}\otimes\mathbf{1}) + 2\mu(\mathbf{I} + 2\mu\Delta\tilde{\bar{\lambda}}_{n+1}\mathbf{A}_n)^{-1} : \mathbf{I}^{\text{dev}},
\end{aligned} \tag{106}$$

where σ^m_{n+1} stands for the mean stress, and K is the bulk modulus. In Fig. 11a comparison of the results obtained with the above options is presented. There, it can be checked that the IMPL-EX integration, for cases (A1) and (A2) (IMPL-EX original or rephrased plastic flow) and (A3) (linearized plastic flow) above, have the same order of accuracy and converge to the implicitly integrated results.

Appendix 2. Stability analysis of the damage model implicit integration

Stability of numerical integration algorithms is essentially related to the way that the numerical integration error, committed at a given time step of the analysis, propagates and amplifies in subsequent time steps [21,22]. In the context of linear constitutive models, the stability of an integration scheme can be analyzed via its spectral properties but, for non-linear models as the ones considered here, more sophisticated tools must be used. Classical works on the topic are the ones in [7,8], where the concepts of small and large scale instability are considered.

For a standard problem of evolution:

$$\begin{cases}
\frac{\text{d}}{\text{d}t}\boldsymbol{\Sigma}(t) = f(\boldsymbol{\Sigma}(t), t); \quad t \in [0, T],\\
\boldsymbol{\Sigma}(0) = \boldsymbol{\Sigma}_0
\end{cases} \tag{107}$$

and considering the time domain of interest, $[0,T]$, subdivided in N disjoint time intervals $[t_n, t_{n+1}]$ ($n \in [0,N]$) and any two possible discrete flows of the solution, $\boldsymbol{\Sigma}_n$ and $\widehat{\boldsymbol{\Sigma}}_n$ $\forall n \in [0,N]$, generated by two different initial conditions, $\boldsymbol{\Sigma}_0$ and $\widehat{\boldsymbol{\Sigma}}_0$, respectively, the integration scheme is said to be *B-stable in the large scale* if:

$$\underbrace{\|\boldsymbol{\Sigma}_{n+1} - \widehat{\boldsymbol{\Sigma}}_{n+1}\|_{\mathbf{G}}}_{e_{n+1}} \leqslant \underbrace{\|\boldsymbol{\Sigma}_n - \widehat{\boldsymbol{\Sigma}}_n\|_{\mathbf{G}}}_{e_n} \quad \forall n \in [0,N], \tag{108}$$

where $\|\bullet\|_{\mathbf{G}}$ stands for a norm in the appropriate metric $\mathbf{G}$. Eq. (108) states that the (relative) error of both flows, $e_{(\bullet)}$, motivated by the error in the initial values, e_0, does not amplify in subsequent time steps.

For the damage model in Eqs. (9)–(15), $\boldsymbol{\Sigma}(t)$ in Eq. (107) can be taken as the generalized stress tensor: $\boldsymbol{\Sigma}(t) = [\boldsymbol{\sigma}(t), q(t)]$.

Now, let us consider a *prescribed* strain flow, $\boldsymbol{\varepsilon}(t)$ $\forall t \in [0,T]$, driving the evolution of the stresses through the corresponding integration scheme, and its corresponding discrete counterparts, $[\boldsymbol{\varepsilon}_n, \boldsymbol{\varepsilon}_{n+1}]$, at the ends of the time interval $[t_n, t_{n+1}]$.

The internal strain-like variable, α, in Eqs. (16) and (18) is integrated exactly according to Eq. (16) and, therefore the flows $\alpha(\boldsymbol{\varepsilon}_n)$ and $\hat{\alpha}(\boldsymbol{\varepsilon}_n)$ are identical:

$$\alpha_n = \hat{\alpha}_n \quad \forall n \in [0,N]. \tag{109}$$

For the integrated internal variable q, in Eq. (19), both flows are:

$$\begin{cases} q_{n+1} = q_n + H_n(\alpha_n)(\alpha_{n+1} - \alpha_n), \\ \hat{q}_{n+1} = \hat{q}_n + H_n(\alpha_n)(\alpha_{n+1} - \alpha_n), \\ \Rightarrow \|q_{n+1} - \hat{q}_{n+1}\| = \|q_n - \hat{q}_n\|, \end{cases} \tag{110}$$

which proves the large scale stability for the integration of the internal variable. As for the integrated stresses in Eq. (20) the two flows, $\boldsymbol{\sigma}$ and $\hat{\boldsymbol{\sigma}}$, are:

$$\begin{cases} \boldsymbol{\sigma}_{n+1} = \frac{q_{n+1}}{\alpha_{n+1}} \mathbf{C}^e : \boldsymbol{\varepsilon}_{n+1} \\ \hat{\boldsymbol{\sigma}}_{n+1} = \frac{\tilde{q}_{n+1}}{\alpha_{n+1}} \mathbf{C}^e : \boldsymbol{\varepsilon}_{n+1} \end{cases} \Rightarrow \frac{\boldsymbol{\sigma}_{n+1}}{q_{n+1}} - \frac{\hat{\boldsymbol{\sigma}}_{n+1}}{\tilde{q}_{n+1}} = 0,$$
$$\Rightarrow \left\| \frac{\boldsymbol{\sigma}_{n+1}}{q_{n+1}} - \frac{\hat{\boldsymbol{\sigma}}_{n+1}}{\hat{q}_{n+1}} \right\| = 0 \Rightarrow 0 = \left\| \frac{\boldsymbol{\sigma}_{n+1}}{q_{n+1}} - \frac{\hat{\boldsymbol{\sigma}}_{n+1}}{\hat{q}_{n+1}} \right\| = \left\| \frac{\boldsymbol{\sigma}_n}{q_n} - \frac{\hat{\boldsymbol{\sigma}}_n}{\hat{q}_n} \right\|, \tag{111}$$

which proves the large scale stability of the implicitly integrated variable $(\boldsymbol{\sigma}/q)$. Then, the product:

$$\underbrace{\frac{\boldsymbol{\sigma}}{q}}_{\text{stable}} \cdot \underbrace{q}_{\text{stable}} = \boldsymbol{\sigma} \tag{112}$$

is stable, which proves the stability of the implicitly integrated stresses $\boldsymbol{\sigma}$.

Appendix 3. Continuation (updated normal plane) method for the IMPL-EX scheme

The classical continuation methods, for quasi-static problems, are based on the iterative resolution of the *extended* system of equations at time step $n+1$ [10]:

$$\mathbf{R}(\mathbf{a}_{n+1}, \mu_{n+1}) \equiv \mathbf{F}_{\text{int}}(\mathbf{a}_{n+1}) - \mu_{n+1}(s_{n+1})\mathbf{f}^{\text{ref}} = \mathbf{0}, \tag{113}$$

$$c(\mathbf{a}_{n+1}, \Delta s_{n+1}) = 0; \quad \Delta s_{n+1} \geqslant 0, \tag{114}$$

where $\mathbf{R}_{n+1}$ are the residual forces and $\mathbf{f}^{\text{ref}}$ are the reference external forces, which are incremented (or decremented) according to the load factor $\mu(s)$, evolving in terms of the pseudo-time parameter $s \geqslant 0 (\Delta s \geqslant 0)$. Eq. (114) constitutes a scalar additional constraint, allowing to solve for the incremental load factor, $\Delta\mu_{n+1}$, for a given value of Δs_{n+1}.

Then, the typical Newton–Raphson scheme for resolution of Eqs. (113) and (114) is the following:

Box 1. Continuation method scheme

Repeat, until convergence, the following steps:

1. Residual forces linearization

$$\mathbf{K}_{n+1}^{(i)} \cdot \delta\mathbf{a}_{n+1}^{(i+1)} = -\underbrace{\mathbf{R}(\mathbf{a}_{n+1}^{(i)}, \mu_{n+1}^{(i)})}_{\mathbf{R}_{n+1}^{(i)}} + \delta\mu_{n+1}^{(i+1)}\mathbf{f}^{\text{ref}} = \mathbf{0},$$
$$\mathbf{K}_{n+1}^{(i)} = \frac{\partial \mathbf{R}_{n+1}^{(i)}}{\partial \mathbf{a}_{n+1}^{(i)}}. \tag{115}$$

2. Iterative displacement decomposition

$$\delta\mathbf{a}_{n+1}^{(i+1)} = \delta\mathbf{a}^a + \delta\mu_{n+1}^{(i+1)}\mathbf{q}_{n+1}^{(i)},$$
$$\delta\mathbf{a}^a = -[\mathbf{K}_{n+1}^{(i)}]^{-1} \cdot \mathbf{R}_{n+1}^{(i)}; \quad \mathbf{q}_{n+1}^{(i)} = [\mathbf{K}_{n+1}^{(i)}]^{-1} \cdot \mathbf{f}^{\text{ref}}. \tag{116}$$

3. Solve for the iterative load increment

$$c(\mathbf{a}_{n+1}^{(i)} + \delta\mathbf{a}^a + \delta\mu_{n+1}^{(i+1)}\mathbf{q}_{n+1}^{(i)}, \Delta s_{n+1}) = 0 \rightarrow \delta\mu_{n+1}^{i+1}. \tag{117}$$

4. Update displacements and load factor

$$\mathbf{a}_{n+1}^{(i+1)} = \mathbf{a}_n + \Delta\mathbf{a}_{n+1}^{(i+1)}; \quad \Delta\mathbf{a}_{n+1}^{(i+1)} = \Delta\mathbf{a}_{n+1}^{(i)} + \delta\mathbf{a}_{n+1}^{(i+1)},$$
$$\mu_{n+1}^{(i+1)} = \mu_n + \Delta\mu_{n+1}^{(i+1)}; \quad \Delta\mu_{n+1}^{(i+1)} = \Delta\mu_{n+1}^{(i)} + \delta\mu_{n+1}^{(i+1)}. \tag{118}$$

where superscript $(\bullet)^{(i)}$ refers to iteration i. In Eq. (116) $\mathbf{q}_{n+1}^{(i)}$ is the so-called *reference displacement vector,* which can be understood as the linear displacement response of the solid, endowed with an structural stiffness, $\mathbf{K}_{n+1}^{(i)}$ and subjected to the reference external actions, $\mathbf{f}^{\text{ref}}$. The scalar Eq. (117) is generally solved in a closed form.

Let, for instance, consider the following format of the constraint (114):

$$c(\mathbf{a}_{n+1},\Delta s_{n+1}) \equiv \Delta\mathbf{a}_{n+1}\cdot\underbrace{\frac{\mathbf{q}_{n+1}^{(i)}}{\|\mathbf{q}_{n+1}^{(i)}\|}}_{\mathbf{n}_{n+1}^{(i)}} - \Delta s_{n+1} = \Delta\mathbf{a}_{n+1}\cdot\mathbf{n}_{n+1}^{(i)} - \Delta s_{n+1} = 0, \tag{119}$$

which establishes Δs_{n+1} as the projection of the actual incremental displacements, $\Delta\mathbf{a}_{n+1}$, onto the reference direction, $\mathbf{n}_{n+1}^{(i)}$. Then Eq. (117) reads:

$$\delta\mu_{n+1}^{(i+1)} = \frac{\Delta s_{n+1} - (\Delta\mathbf{a}_{n+1}^{(i)} + \delta\mathbf{a}^a)\cdot\mathbf{n}_{n+1}^{(i)}}{\mathbf{q}_{n+1}^{(i)}\cdot\mathbf{n}_{n+1}^{(i)}}. \tag{121}$$

At the beginning of the time step $\Delta\mathbf{a}_{n+1}^{(i)} = 0$ and Eq. (121) reads:

$$\delta\mu_{n+1}^{(0)} = \frac{\Delta s_{n+1} - \delta\mathbf{a}^a\cdot\mathbf{n}_{n+1}^{(0)}}{\mathbf{q}_{n+1}^{(0)}\cdot\mathbf{n}_{n+1}^{(0)}}\delta\mathbf{a}^a = -[\mathbf{K}_{n+1}^{(0)}]^{-1}\cdot\mathbf{R}_{n+1}^{(0)};\quad \mathbf{q}_{n+1}^{(0)} = [\mathbf{K}_{n+1}^{(0)}]^{-1}\cdot\mathbf{f}^{\text{ref}}. \tag{122}$$

Unlike in standard integration methods, in the IMPL-EX integration scheme the initial residue, $\mathbf{R}_{n+1}^{(0)}$, is not zero (see Remark 3.4 and Eq. (50)). Moreover, for the step-linear cases mentioned in Remark 3.2, the algorithmic effective stiffness $\widetilde{\mathbf{K}}_{n+1}^{\text{eff}}$ is constant and the Newton–Raphson procedure takes a unique iteration per time step to converge. For those cases, the continuation method in BOX 1 reduces to:

Box 2. Continuation method for the IMPL-EX integration scheme and the step-linear case

1. Residual forces linearization

$$\widetilde{\mathbf{K}}_{n+1}^{\text{tang}}\cdot\Delta\mathbf{a}_{n+1} = -\underbrace{\mathbf{R}^{(0)}(\mathbf{a}_n,\mu_{n+1}^{(O)},\tilde{\alpha}_{n+1})}_{\mathbf{R}_{n+1}^{(0)}} + \Delta\mu_{n+1}\mathbf{f}^{\text{ref}},\qquad \widetilde{\mathbf{K}}_{n+1}^{\text{tang}} = \frac{\partial\mathbf{R}_{n+1}^{(0)}}{\partial\mathbf{a}_n}. \tag{123}$$

2. Incremental displacement decomposition

$$\Delta\mathbf{a}_{n+1} = \Delta\mathbf{a}^a + \Delta\mu_{n+1}\mathbf{q}_{n+1},\qquad \Delta\mathbf{a}^a = -\widetilde{\mathbf{K}}_{n+1}^{\text{tang}^{-1}}\cdot\mathbf{R}_{n+1}^{(0)};\quad \mathbf{q}_{n+1} = [\widetilde{\mathbf{K}}_{n+1}^{\text{tang}}]^{-1}\cdot\mathbf{f}^{\text{ref}}. \tag{124}$$

3. Solving for the load increment

$$\Delta\mu_{n+1} = \frac{\Delta s_{n+1} - \Delta\mathbf{a}^a\cdot\mathbf{n}_{n+1}}{\mathbf{q}_{n+1}\cdot\mathbf{n}_{n+1}},\qquad \Delta\mathbf{a}^a = -[\widetilde{\mathbf{K}}_{n+1}^{\text{tang}}]^{-1}\cdot\mathbf{R}_{n+1}^{(0)};\quad \mathbf{q}_{n+1} = \widetilde{\mathbf{K}}_{n+1}^{\text{tang}^{-1}}\cdot\mathbf{f}^{\text{ref}},\qquad \mathbf{n}_{n+1} = \frac{\mathbf{q}_{n+1}}{\|\mathbf{q}_{n+1}\|}. \tag{125}$$

4. Update displacements and load factor

$$\mathbf{a}_{n+1} = \mathbf{a}_n + \Delta\mathbf{a}_{n+1},\qquad \mu_{n+1} = \mu_n + \Delta\mu_{n+1}. \tag{126}$$

$$\begin{cases} c(\mathbf{a}_{n+1}^{(i+1)},\Delta s_{n+1}) \equiv \Delta\mathbf{a}_{n+1}^{(i+1)}\cdot\mathbf{n}_{n+1}^{(i)} - \Delta s_{n+1} = 0, \\ \Delta\mathbf{a}_{n+1}^{(i+1)} = \Delta\mathbf{a}_{n+1}^{(i)} + \delta\mathbf{a}_{n+1}^{(i+1)} = \Delta\mathbf{a}_{n+1}^{(i)} + \delta\mathbf{a}^a + \delta\mu_{n+1}^{(i+1)}\mathbf{q}_{n+1}^{(i)}. \end{cases} \tag{120}$$

Eq. (120) can be solved for $\delta\mu_{n+1}^{(i+1)}$ yielding:

References

[1] J. Lubliner, Plasticity Theory, Mcmillan Publishing Company, 1990.
[2] J.C. Simo, T.J.R. Hughes, Computational Inelasticity, Springer, 1998.
[3] J. Oliver, A.E. Huespe, S. Blanco, D.L. Linero, Stability and robustness issues in numerical modeling of material failure in the

strong discontinuity approach, Comput. Methods Appl. Mech. Engrg. 195 (2006) 7093–7114.
[4] J. Oliver, Topics on failure mechanics, International Center for Numerical Methods in Engineering (CIMNE) Monograph no. 68, 2002.
[5] J. Oliver, A.E. Huespe, E. Samaniego, E.W.V. Chaves, Continuum approach to the numerical simulation of material failure in concrete, Int. J. Numer. Anal. Methods Geomech. 28 (2004) 609–632.
[6] J.C. Simo, J.W. Ju, Stress and strain based continuum damage models: I. Formulation, Int. J. Solids Struct. 15 (1987) 821–840.
[7] M. Ortiz, E.P. Popov, Accuracy and stability of integration algorithms for elastoplastic constitutive relations, Int. J Numer. Methods Engrg. 21 (1985) 1561–1576.
[8] J.C. Simo, S. Govindjee, Nonlinear B-stability and symmetry preserving return mapping algorithms for plasticity and viscoplasticity, Int. J. Numer. Methods Engrg. 31 (1991) 151–176.
[9] T. Belytschko, K. Mish, Computability in non-linear solid mechanics, Int. J. Numer. Methods Engrg. 52 (2001) 3–21.
[10] M.A. Crisfield, Non-linear Finite Element Analysis of Solids and Structures, Advanced Topics, vol. 2, John Wiley & Sons, 1998.
[11] J. Oliver, Modelling strong discontinuities in solid mechanics via strain softening constitutive equations. 1. Fundamentals, Int. J. Numer. Methods Engrg. 39 (1996) 3575–3600.
[12] F. Armero, K. Garikipati, An analysis of strong discontinuities in multiplicative finite strain plasticity and their relation with the numerical simulation of strain localization in solids, Int. J. Solids Struct. 33 (1996) 2863–2885.
[13] J. Oliver, A.E. Huespe, Theoretical and computational issues in modelling material failure in strong discontinuity scenarios, Comput. Methods Appl. Mech. Engrg. 193 (2004) 2987–3014.
[14] J. Oliver, On the discrete constitutive models induced by strong discontinuity kinematics and continuum constitutive equations, Int. J. Solids Struct. 37 (2000) 7207–7229.
[15] T. Belytschko, N. Moes, S. Usui, C. Parimi, Arbitrary discontinuities in finite elements, Int. J. Numer. Methods Engrg. 50 (2001) 993–1013.
[16] J.M. Sancho, J. Planas, D.A. Cendón, E. Reyes, J.C. Gálvez, An embedded crack model for finite element analysis of concrete fracture, Engrg. Fract. Mech. 74 (2007) 75–86.
[17] J. Oliver, S. Oller, J.C. Cante, A plasticity model for simulation of industrial powder compaction processes, Int. J. Solids Struct. 33 (1996) 3161–3178.
[18] P.K. Vijalapura, J. Strain, S. Govindjee, Fractional step methods for index-1 differential algebraic equations, J. Comput. Phys. 203 (2005) 305–320.
[19] F. Armero, J.C. Simo, A new unconditionally stable fractional step method for non-linear coupled thermomechanical problems, Int. J. Numer. Methods Engrg. 35 (2005) 737–776.
[20] J. Blasco, R. Codina, A. Huerta, A fractional-step method for the incompressible Navier–Stokes equations related to a predictor–multicorrector algorithm, Int. J. Numer. Methods Fluids 28 (1998) 1391–1419.
[21] T.J.R. Hughes, The Finite Element Method, Prentice-Hall, Englewood Cliffs, NJ, 1987.
[22] T. Belytschko, W.K. Liu, B. Moran, Nonlinear Finite Elements for Continua and Structures, John Wiley & Sons Ltd., New York, 2001.
[23] J.G. Rots, S. Invernizzi, Regularized saw-tooth softening, in: N. Bicanic, R.D. Borst, H. Mang, G. Meschke (Eds.), Computational Modelling of Concrete Structures, A.A. Balkema, Liss, The Netherlands, 2003, pp. 599–617.
[24] J.G. Rots, S. Invernizzi, B. Belletti, Saw-tooth softening/stiffening – a stable computational procedure for RC structures, Computers and Concrete 3 (2006) 213–233.
[25] C. Felippa, Nonlinear finite element methods in notes of course (ASEN 5107), Department of Aerospace Engineering Sciences University of Colorado at Boulder, 2007 (Chapter 18).
[26] P. Wriggers, Computational Contact Mechanics, Wiley, 2002.
[27] J. Oliver, J.C. Cante, R. Weyler, C. Gonzalez, Particle finite element methods in solid mechanics problems, Springer-Verlag, 2007, pp. 87–104.
[28] A.W.A. Konter, Advanced finite element contact benchmarks, Netherlands Institute for Metals Research FENET-UNOTT-DLE-09, 2005.
[29] H. Hertz, Ueber die Beruehrung fester elastische Koerper, J. Reine Angew. Math. 92 (1881) 156–171.
[30] J. Oliver, D.L. Linero, A.E. Huespe, O.L. Manzoli, Two-dimensional modeling of material failure in reinforced concrete by means of a continuum strong discontinuity approach, Comput. Methods Appl. Mech. and Engrg. 197 (2008) 332–348.
[31] P.J. Sánchez, A.E. Huespe, J. Oliver, On some topics for the numerical simulation of ductile fracture, International Journal of Plasticity, in press, available online, 2007.