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
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