Efficient and low memory strain-rate independent return mapping algorithm for general yield surfaces and stress states

M.V.B. Santana, P. Keo, Mohammed Hjiaj

Version of Record: https://www.sciencedirect.com/science/article/pii/S096599782100096X
Manuscript_ee254c176d057af8f8143798c990a96b

Efficient and low memory strain-rate independent return mapping algorithm for general yield surfaces and stress states

M. V. B. Santana[a,*], P. Keo[a], M. Hjiaj[a]

[a]*INSA de Rennes – LGCGM/Structural Engineering Research Group, 20 avenue des Buttes de Coësmes, CS 70839, 35708 Rennes Cedex 7, France*

Abstract

Structural theories often consider constraints on the stresses, creating a partition into active and fixed components. Sophisticated nonlinear material models are usually described in 3D-continuum form. A reformulation of the material models is then necessary in order to enforce the constraint on the relevant stress components. In this paper, a new strain-rate independent return mapping algorithm is developed in order to deal with arbitrary partitions of the stresses considering general associative elasto-plastic material models. The developed algorithm works directly with the active stress and strain components, satisfying the constraints on the relevant stresses components at each integration point of each element and yielding optimal computational efficiency and memory costs. The tangent operator is obtain in closed-form, providing quadratic convergence to the local iterative process. A comparison of the proposed approach to a previously developed algorithm is made to highlight its properties.

*Corresponding author.
Email address: mbentosa@insa-rennes.fr (M. V. B. Santana)

Keywords: Plasticity, Implicit integration, Return mapping algorithm, General yield function, Efficiency, Memory cost

1. Introduction

Most structural theories adopt the hypothesis that only certain components of the general 3D stresses are active, that is, they directly depends on the associated strains. The remaining components are assumed to be fixed. In truss and cable theories [2], for example, usually the axial stress is active while the others stresses are set to zero. In the 2D and 3D Timoshenko's beam theory [16], the cross-section axial and shear stresses represent the active components, while in the shell theory [17] the normal stresses in the direction of the shell's thickness is set to zero. The nonlinear elasto-plastic behavior of complex materials is usually described in a 3D-continuum form and so the material models must be reformulated in order to take into account the constraints on stress components.

In a finite element context, return mapping algorithms are used to integrate the material constitutive relations between two consecutive time steps of a incremental procedure. These algorithms follow a *strain-driven* approach where, for each integration-point of each element, the strains (computed from the nodal degrees of freedom) and plastic history (stored at the integrate-point) are used to obtain the stresses, the tangent operator and the updated plastic variables. Many authors have proposed return mapping algorithms for specific partitions of the stresses [6, 10, 14, 19, 21] or specific material models [1, 3, 4, 9, 11]. The return mapping algorithm proposed by Klinkel and Govindjee [7] differs from the others in the sense that a general parti-

tion of the stresses into active and fixed components is considered. In the present work, a new version of this algorithm is developed aiming to optimize memory cost and computational efficiency.

The development of return mapping algorithms has been an active field of research in the last decades. Jetteur [5] and Simo and Taylor [21] developed return-mapping algorithms for elasto-plasticity in plane stress analysis. Simo and Govindjee [19] found an exact closed-form solution for plane stress elasto-visco-plasticity. Simo [18] studied algorithms in multiplicative plasticity. Ohno et al. [14] developed a implicit integration algorithm applicable to both plane stress and 3D stress states. Mánik [13] proposed a vector/matrix notation for return-mapping algorithm for advanced yield functions. Meng et al. [12] studied a return mapping algorithm in principal space for general isotropic elasto-plasticity with multi-surface plasticity.

In the present paper, a new return mapping algorithm is developed. The main originalities of the present contribution are:

- A return mapping algorithm capable of handling general partitions of the complete 3D stresses for arbitrary yield functions, while performing operations only on the active components of the stress and strain vectors. The remaining quantities can be optionally obtained as a by-product;

- A bit-mask implementation of the proposed algorithm, yielding advantageous computational properties such as low memory cost and high efficiency comparable to return mapping algorithms designed for specific stress partitions while maintaining a general formulation that can be applied to any structural model;

- Explicit algorithmic tangent operator in closed-form, providing quadratic convergence to the local iterative process.

The paper is structured as follows. In Section 2, the classical elasto-plastic material model with combined isotropic and kinematic hardening used in this work is introduced. In Section 3, the proposed return mapping algorithm is presented and compared against the standard approach in terms of computational efficiency and memory cost. Section 4 presents a number of numerical applications, highlighting the accuracy and efficiency of the proposed method. Finally, in Section 5, some conclusions and suggestions for future works are given. For completeness, the expressions for some classical yield surface equations, gradients and hessians are included in Appendix A.

2. Material model

Structural theories often consider that only certain stress components are relevant in the mechanical behavior of the load bearing elements, due to their shapes, support conditions and applied loads (Tab. 1). Truss elements, for example, are mainly designed with respect to the axial forces that in turn generate axial stresses (σ_{xx}). In 3D beam elements, axial (σ_{xx}) and shear (τ_{xy} and τ_{xz}) stresses are generated by the section resultants. In plane-stress elements, only the in-plane stress components (σ_{xx}, σ_{yy} and τ_{xy}) are usually considered.

In the general case, the complete 3D stress vector $\boldsymbol{\sigma}$ can be partitioned into active $\boldsymbol{\sigma}_a$ and fixed $\boldsymbol{\sigma}_z$ components. The active stress components have n_a entries, while the fixed stress components have n_z entries. For most structural theories, the fixed stress components $\boldsymbol{\sigma}_z$ are identically zero. However,

Model	σ_{xx}	σ_{yy}	σ_{zz}	τ_{xy}	τ_{xz}	τ_{yz}	n_a	n_z
Truss	✓	✗	✗	✗	✗	✗	1	5
2D Beam	✓	✗	✗	✓	✗	✗	2	4
3D Beam	✓	✗	✗	✓	✓	✗	3	3
Shell	✓	✓	✗	✓	✓	✓	5	1
Plane-stress	✓	✓	✗	✓	✗	✗	3	3
3D Solid	✓	✓	✓	✓	✓	✓	6	0

Table 1: Active and null stress components for different structural models.

to make our developments as general as possible, in the present they are assumed to have a constant (possibly non-zero) value $\boldsymbol{\sigma}_z = \overline{\boldsymbol{\sigma}}_z$. This allows, for example, the modeling of super-imposed hydro-static stress states or confinement stresses in reinforced concrete. The corresponding strain components, extracted from the full 3D strain vector $\boldsymbol{\varepsilon}$ are $\boldsymbol{\varepsilon}_a$ and $\boldsymbol{\varepsilon}_z$, respectively. It's noteworthy that the strain components $\boldsymbol{\varepsilon}_z$ are not identically fixed, but can be later computed via the fixed value condition on the stress components $\boldsymbol{\sigma}_z$. For instance, the partition of the stresses and strains into active and fixed components for the 3D beam model are (Tab. 1):

$$\boldsymbol{\sigma}_a = \begin{Bmatrix} \sigma_{xx} & \tau_{xy} & \tau_{xz} \end{Bmatrix} \tag{1}$$

$$\boldsymbol{\sigma}_z = \begin{Bmatrix} \sigma_{yy} & \sigma_{zz} & \tau_{yz} \end{Bmatrix} \tag{2}$$

$$\boldsymbol{\varepsilon}_a = \begin{Bmatrix} \varepsilon_{xx} & \gamma_{xy} & \gamma_{xz} \end{Bmatrix} \tag{3}$$

$$\boldsymbol{\varepsilon}_z = \begin{Bmatrix} \varepsilon_{yy} & \varepsilon_{zz} & \gamma_{yz} \end{Bmatrix} \tag{4}$$

In this work, only small strain elasto-plastic material behavior is considered. The total strains ($\boldsymbol{\varepsilon}_a$ and $\boldsymbol{\varepsilon}_z$) can therefore be split additivily into

elastic ($\boldsymbol{\varepsilon}_a^e$ and $\boldsymbol{\varepsilon}_z^e$) and plastic ($\boldsymbol{\varepsilon}_a^p$ and $\boldsymbol{\varepsilon}_z^p$) parts:

$$\boldsymbol{\varepsilon}_a = \boldsymbol{\varepsilon}_a^e + \boldsymbol{\varepsilon}_a^p \tag{5}$$

$$\boldsymbol{\varepsilon}_z = \boldsymbol{\varepsilon}_z^e + \boldsymbol{\varepsilon}_z^p \tag{6}$$

Considering that the material presents a linear response in the elastic regime, the complete 3D stress vector $\boldsymbol{\sigma}$ can be related to the complete 3D elastic strain vector $\boldsymbol{\varepsilon}^e$ via the elastic stiffness tensor $\mathbf{C}$ as:

$$\boldsymbol{\sigma} = \mathbf{C}\boldsymbol{\varepsilon}^e \tag{7}$$

The elastic constitutive relation can be written in terms of the active $\boldsymbol{\sigma}_a$ and fixed $\boldsymbol{\sigma}_z$ stress components as:

$$\boldsymbol{\sigma}_a = \mathbf{C}_{aa}\boldsymbol{\varepsilon}_a^e + \mathbf{C}_{az}\boldsymbol{\varepsilon}_z^e \tag{8}$$

$$\boldsymbol{\sigma}_z = \mathbf{C}_{za}\boldsymbol{\varepsilon}_a^e + \mathbf{C}_{zz}\boldsymbol{\varepsilon}_z^e \tag{9}$$

The sub-tensors $\mathbf{C}_{aa}$, $\mathbf{C}_{az}$, $\mathbf{C}_{za}$ and $\mathbf{C}_{zz}$ are created from the complete 3D elastic stiffness tensor $\mathbf{C}$ by extracting the associated rows and columns. It's noteworthy that, as the elastic stiffness tensor $\mathbf{C}$ has a full rank, the sub-tensors $\mathbf{C}_{aa}$ and $\mathbf{C}_{zz}$ are inversible for any partition of active and fixed stress components. Therefore, given the elastic active strains $\boldsymbol{\varepsilon}_a^e$ and the fixed stress $\overline{\boldsymbol{\sigma}}_z$ components, the elastic fixed strains $\boldsymbol{\varepsilon}_z^e$ can be obtained from Eq. (9) as:

$$\boldsymbol{\varepsilon}_z^e = \mathbf{C}_{zz}^{-1}\left(\overline{\boldsymbol{\sigma}}_z - \mathbf{C}_{za}\boldsymbol{\varepsilon}_a^e\right) \tag{10}$$

Combining Eqs. (8) and (10), the active stress $\boldsymbol{\sigma}_a$ and strain $\boldsymbol{\varepsilon}_a$ components can be directly related as:

$$\boldsymbol{\sigma}_a = \mathbf{C}_a\boldsymbol{\varepsilon}_a^e + \boldsymbol{\sigma}_r \tag{11}$$

where:

$$\boldsymbol{\sigma}_r = \mathbf{C}_{az}\mathbf{C}_{zz}^{-1}\overline{\boldsymbol{\sigma}}_z \tag{12}$$

$$\mathbf{C}_a = \mathbf{C}_{aa} - \mathbf{C}_{az}\mathbf{C}_{zz}^{-1}\mathbf{C}_{za} \tag{13}$$

The combined active elastic stiffness tensor $\mathbf{C}_a$ expression in Eq. (13) involves a matrix inversion of size $(n_z \times n_z)$ and two matrix multiplications of size $(n_a \times n_z)$ per $(n_z \times n_z)$ per $(n_z \times n_a)$. For structural models where n_a is small (e.g. truss and beams), this operations can be computationally expensive. The combined active elastic stiffness tensor $\mathbf{C}_a$ can be obtained in a more efficient way via the elastic flexibility tensor $\mathbf{D} = \mathbf{C}^{-1}$, relating the elastic strains $\boldsymbol{\varepsilon}^e$ to the stresses $\boldsymbol{\sigma}$.

Analogously to the stresses in Eqs. (8) and (9), the elastic constitutive relation can be written in terms of the elastic active $\boldsymbol{\varepsilon}_a^e$ and fixed $\boldsymbol{\varepsilon}_z^e$ strain components as:

$$\boldsymbol{\varepsilon}_a^e = \mathbf{D}_{aa}\boldsymbol{\sigma}_a + \mathbf{D}_{az}\overline{\boldsymbol{\sigma}}_z \tag{14}$$

$$\boldsymbol{\varepsilon}_z^e = \mathbf{D}_{za}\boldsymbol{\sigma}_a + \mathbf{D}_{zz}\overline{\boldsymbol{\sigma}}_z \tag{15}$$

Solving Eq. (14) for the active stress components $\boldsymbol{\sigma}_a$, the combined active elastic stiffness tensor $\mathbf{C}_a$ can be obtained by simply inverting the $(n_a \times n_a)$ sub-tensor $\mathbf{D}_{aa}$:

$$\mathbf{C}_a = \mathbf{D}_{aa}^{-1} \tag{16}$$

The stress limit is represented via a yield surface $f(\boldsymbol{\sigma}_a, \boldsymbol{\sigma}_z, \mathbf{q}, \lambda)$ with kinematic hardening (back-stress) $\mathbf{q}$ and isotropic hardening λ. The elastic domain is characterized by stress states in which $f(\boldsymbol{\sigma}_a, \boldsymbol{\sigma}_z, \mathbf{q}, \lambda) < 0$, while the

plastic domain is characterized by stress states in which $f(\boldsymbol{\sigma}_a, \boldsymbol{\sigma}_z, \mathbf{q}, \lambda) = 0$. Stress states in which $f(\boldsymbol{\sigma}_a, \boldsymbol{\sigma}_z, \mathbf{q}, \lambda) > 0$ are inadmissible and don't represent a physical configuration. The yield surface has a fixed shape $f^*(\boldsymbol{\sigma}_a - \mathbf{q}, \boldsymbol{\sigma}_z)$ that is translated via the kinematic hardening variable $\mathbf{q}$ and scales according to the isotropic hardening λ via a yield stress function $\sigma_y(\lambda)$. The yield surface equation can then be written in a general form as:

$$f(\boldsymbol{\sigma}_a, \boldsymbol{\sigma}_z, \mathbf{q}, \lambda) = f^*(\boldsymbol{\sigma}_a - \mathbf{q}, \boldsymbol{\sigma}_z) - \sigma_y(\lambda) \leq 0 \tag{17}$$

The yield surface's gradients $\mathbf{g}_a(\boldsymbol{\sigma}_a, \boldsymbol{\sigma}_z, \mathbf{q})$ and $\mathbf{g}_z(\boldsymbol{\sigma}_a, \boldsymbol{\sigma}_z, \mathbf{q})$, with respect to the active $\boldsymbol{\sigma}_a$ and fixed $\boldsymbol{\sigma}_z$ stress components respectively, can be defined as:

$$\mathbf{g}_a(\boldsymbol{\sigma}_a, \boldsymbol{\sigma}_z, \mathbf{q}) = \frac{\partial f^*(\boldsymbol{\sigma}_a - \mathbf{q}, \boldsymbol{\sigma}_z)}{\partial \boldsymbol{\sigma}_a} \tag{18}$$

$$\mathbf{g}_z(\boldsymbol{\sigma}_a, \boldsymbol{\sigma}_z, \mathbf{q}) = \frac{\partial f^*(\boldsymbol{\sigma}_a - \mathbf{q}, \boldsymbol{\sigma}_z)}{\partial \boldsymbol{\sigma}_z} \tag{19}$$

The yield surface hessian $\mathbf{H}_a(\boldsymbol{\sigma}_a, \boldsymbol{\sigma}_z, \mathbf{q})$ and plastic modulus $K_p(\lambda)$ are used in the return-mapping algorithm of Sec. 3 and can be defined as:

$$K_p(\lambda) = \frac{\partial \sigma_y(\lambda)}{\partial \lambda} \tag{20}$$

$$\mathbf{H}_a(\boldsymbol{\sigma}_a, \boldsymbol{\sigma}_z, \mathbf{q}) = \frac{\partial^2 f^*(\boldsymbol{\sigma}_a - \mathbf{q}, \boldsymbol{\sigma}_z)}{\partial \boldsymbol{\sigma}_a \partial \boldsymbol{\sigma}_a} \tag{21}$$

In the present work, an associative flow rule is adopted, and so, the time evolution of the plastic deformations $\boldsymbol{\varepsilon}_a^p$ and $\boldsymbol{\varepsilon}_z^p$ can then be described via the yield surface gradients $\mathbf{g}_a(\boldsymbol{\sigma}_a, \boldsymbol{\sigma}_z, \mathbf{q})$ and $\mathbf{g}_z(\boldsymbol{\sigma}_a, \boldsymbol{\sigma}_z, \mathbf{q})$ as:

$$\dot{\boldsymbol{\varepsilon}}_a^p = \dot{\lambda} \mathbf{g}_a(\boldsymbol{\sigma}_a, \boldsymbol{\sigma}_z, \mathbf{q}) \tag{22}$$

$$\dot{\boldsymbol{\varepsilon}}_z^p = \dot{\lambda} \mathbf{g}_z(\boldsymbol{\sigma}_a, \boldsymbol{\sigma}_z, \mathbf{q}) \tag{23}$$

The back-stress evolution is also described via an associative flow rule:

$$\dot{\mathbf{q}} = \dot{\lambda} U \mathbf{g}_a(\boldsymbol{\sigma}_a, \boldsymbol{\sigma}_z, \mathbf{q}) \tag{24}$$

where U is the kinematic hardening modulus.

Finally, the model is complemented with the following Khun-Tucker consistency conditions:

$$\dot{\lambda} \geq 0 \tag{25}$$

$$\dot{\lambda} f(\boldsymbol{\sigma}_a, \boldsymbol{\sigma}_z, \mathbf{q}, \lambda) = 0 \tag{26}$$

The first consistency condition (Eq. 25) implies that the isotropic hardening λ increases monotonically. The second condition (Eq. 26) implies that, while in the elastic domain ($f(\boldsymbol{\sigma}_a, \boldsymbol{\sigma}_z, \mathbf{q}, \lambda) < 0$) the isotropic hardening is constant ($\dot{\lambda} = 0$). However, in the plastic domain ($f(\boldsymbol{\sigma}_a, \boldsymbol{\sigma}_z, \mathbf{q}, \lambda) = 0$), the plastic state can evolve ($\dot{\lambda} > 0$).

3. Return mapping algorithm

In a traditional finite element analysis, at each time-step t_{n+1}, the total active strains $\boldsymbol{\varepsilon}_{a,n+1}$, at each integration-point of each element, are computed from the nodal displacements. For such *strain-driven* approach, a return-mapping algorithm is necessary to obtain the active stresses $\boldsymbol{\sigma}_{a,n+1}$, tangent operator $\boldsymbol{\Pi}_{a,n+1} = \partial \boldsymbol{\sigma}_{a,n+1} / \partial \boldsymbol{\varepsilon}_{a,n+1}$ and plastic state ($\boldsymbol{\varepsilon}^p_{a,n+1}$, $\mathbf{q}_{n+1}$ and λ_{n+1}) using the isotropic elasto-plastic material model described in Sec. 2. The active stress $\boldsymbol{\sigma}_{a,n+1}$ and tangent modulus $\boldsymbol{\Pi}_{a,n+1}$ can then be used to compute the elements internal force vector and tangent stiffness matrix. In this Section, the standard method, proposed by Klinkel and Govindjee [7],

is first reviewed in Sec. 3.1 for later comparison with the proposed method, presented in sec. 3.2. The computational time and memory cost of the two approaches are investigated, clearly highlighting the advantages of the proposed method.

3.1. Standard method

Many researchers have proposed return-mapping algorithms for specific structural models, mostly focusing on uni-dimensional (truss) and plane-stress formulations. The return-mapping algorithm proposed by Klinkel and Govindjee [7], referred here as *standard method*, considers a general partition of the stresses ($\boldsymbol{\sigma}_a$ and $\boldsymbol{\sigma}_z$) and, therefore, is ideal for comparison with the methodology developed in the present work.

In the standard method, the active plastic strain components $\boldsymbol{\varepsilon}^p_{a,n}$, kinematic hardening $\mathbf{q}_n$ and isotropic hardening λ_n at the previous time step n are stored at each integration-point, together with the fixed total $\boldsymbol{\varepsilon}_{z,n}$ and plastic $\boldsymbol{\varepsilon}^p_{z,n}$ strain components. The total active strain components $\boldsymbol{\varepsilon}_{a,n+1}$ at the integration-point in the current time step t_{n+1} are computed from the nodal displacements and the fixed stresses are considered null ($\overline{\boldsymbol{\sigma}}_z = \mathbf{0}$).

At first, the current total fixed strain components are set equal to the previous step ($\boldsymbol{\varepsilon}^0_{z,n+1} = \boldsymbol{\varepsilon}_{z,n}$). Then, an iterative process starts in order to fulfill the elasto-plastic constitutive equations and the fixed (null) stress condition simultaneously. At iteration k, the fixed total and plastic strain components ($\boldsymbol{\varepsilon}^k_{z,n+1}$ and $\boldsymbol{\varepsilon}^{p,k}_{z,n+1}$) are combined with the active total and plastic strain components ($\boldsymbol{\varepsilon}_{a,n+1}$ and $\boldsymbol{\varepsilon}^p_{a,n+1}$) to form the 3D total and plastic strain vector ($\boldsymbol{\varepsilon}^k_{n+1}$ and $\boldsymbol{\varepsilon}^{p,k}_{n+1}$). A complete 3D return mapping algorithm is used to determine the 3D stresses $\boldsymbol{\sigma}^k_{n+1}$ and tangent modulus $\mathbf{\Pi}^k_{n+1} = \partial\boldsymbol{\sigma}_{n+1}/\partial\boldsymbol{\varepsilon}_{n+1}$ that

satisfies the elasto-plastic model. Then, the fixed stresses components $\boldsymbol{\sigma}^k_{z,n+1}$ are extracted from the 3D stress vector $\boldsymbol{\sigma}^k_{n+1}$ and the condition $\|\boldsymbol{\sigma}^k_{z,n+1}\| < tol$ is checked for a given tolerance *tol*. If the fixed stress components $\boldsymbol{\sigma}^k_{z,n+1}$ are sufficiently small the iterative process stops, otherwise the fixed strain components $\boldsymbol{\varepsilon}_{z,n+1}$ are updated as:

$$\boldsymbol{\varepsilon}^{k+1}_{z,n+1} = \boldsymbol{\varepsilon}^k_{z,n+1} - \left(\boldsymbol{\Pi}^k_{zz,n+1}\right)^{-1} \boldsymbol{\sigma}^k_{z,n+1} \tag{27}$$

When the algorithm converges, the active elasto-plastic tangent operator $\boldsymbol{\Pi}_{a,n+1}$ can be computed as:

$$\boldsymbol{\Pi}_{a,n+1} = \boldsymbol{\Pi}_{aa,n+1} - \boldsymbol{\Pi}_{az,n+1}\boldsymbol{\Pi}^{-1}_{zz,n+1}\boldsymbol{\Pi}_{za,n+1} \tag{28}$$

where $\boldsymbol{\Pi}^k_{aa,n+1}$, $\boldsymbol{\Pi}^k_{az,n+1}$, $\boldsymbol{\Pi}^k_{za,n+1}$ and $\boldsymbol{\Pi}_{zz,n+1}$ are the sub-tensors obtained by extracting the rows and columns relative to the active and fixed stress components form the complete 3D tangent operator $\boldsymbol{\Pi}_{n+1} = \partial\boldsymbol{\sigma}_{n+1}/\partial\boldsymbol{\varepsilon}_{n+1}$.

A flowchart of the standard method is shown in Fig. 1. A main drawback of the standard method is that, not only the fixed total $\boldsymbol{\varepsilon}_z$ and plastic $\boldsymbol{\varepsilon}^p_z$ strain components must be computed, but they must also be stored at each integration-point. This considerably increases the required memory usage, considering that the values at the previous t_n and current t_{n+1} time steps must be stored (in order to return to an equilibrium configuration if convergence fails) and specially for structural models where $n_z \geq n_a$. The memory usage per integration-point (for a machine with 8 *bytes* float point precision) is shown in Tab. 2.

Another important drawback of the standard method is the required computational time. In order to achieve convergence, a complete 3D return

mapping algorithm must be called multiple times. For elasto-plastic material models with linear hardening behavior, convergence can be achieved instantaneously if further loading on a plastic branch, while there may be several structural level iterations, but for strongly nonlinear plastic behavior more iterations may be required even at the local level. Moreover, the repeated operations of assembly and extraction of the active and fixed strain and stress components can be considerably time consuming.

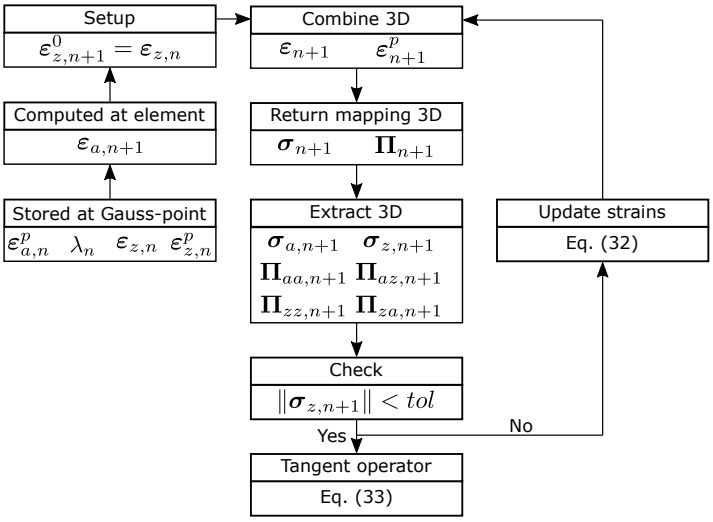

Figure 1: Flowchart of the standard return mapping algorithm.

3.2. Proposed method

The proposed method focus on working (storing and performing operations) only on the active part of the stress $\boldsymbol{\sigma}_a$ and strain $\boldsymbol{\varepsilon}_a$ vectors. Therefore, it's important to identify and extract the active components in an efficient way. Bit-wise operations are among the fastest procedures a central processing unit (CPU) can perform [8, 15]. In terms of the implementation

Model	$\boldsymbol{\varepsilon}_a^p$	$\boldsymbol{\varepsilon}_z$	$\boldsymbol{\varepsilon}_z^p$	$\mathbf{q}$	λ	Total
Truss	16	80	80	16	16	208 bytes
2D Beam	32	64	64	32	16	208 bytes
3D Beam	48	48	48	48	16	208 bytes
Shell	80	16	16	80	16	208 bytes
Plane-stress	48	48	48	48	16	208 bytes
3D Solid	96	0	0	96	16	208 bytes

Table 2: Standard algorithm memory cost per integration-point for different structural models.

procedure, optimal efficiency and memory cost are achieved in the present work by storing the active stress components via a bit-mask format (Tab. 3). A single byte variable is stored for each *element type*. In this format, each bit represents a power of 2, and the presence of a stress type is obtained by simply checking whether a bit is set or not in the bit mask. Bit operations are usually intrinsic routines/functions of most programming languages and are thoroughly optimized in order to achieve minimal computational time. The active and fixed stresses, strains and stiffness components can then be efficiently extracted from their complete 3D counterparts. Moreover, specific stress components can be obtained from the active $\boldsymbol{\sigma}_a$ and fixed $\boldsymbol{\sigma}_z$ stress vectors by checking the position of a bit on the bit-mask.

For example, a 3D beam model has active stresses $\boldsymbol{\sigma}_a = \{\sigma_{xx}, \tau_{xy}, \tau_{xz}\}$. If the complete 3D stresses are ordered as $\boldsymbol{\sigma} = \{\sigma_{xx}, \sigma_{yy}, \sigma_{zz}, \tau_{xy}, \tau_{xz}, \tau_{yz}\}$, the indexes of the active stresses are {0, 3, 4} (index count starts at 0) and the desired bit-mask will be $2^0 + 2^3 + 2^4 = 0b011001 = 25$. In a computation,

Model	τ_{yz}	τ_{xz}	τ_{xy}	σ_{zz}	σ_{yy}	σ_{xx}	bit-mask
Truss	0	0	0	0	0	1	1
2D Beam	0	0	1	0	0	1	9
3D Beam	0	1	1	0	0	1	25
Shell	1	1	1	0	1	1	59
Plane-stress	0	0	1	0	1	1	11
3D Solid	1	1	1	1	1	1	63

Table 3: Bit-mask representation of the active stress components for different structural models.

if the stress component σ_{yy} (with index 1) is required, its isolated bit-mask ($0b000010$) is compared to the structural model bit-mask ($0b011001$) via a bit-wise AND operation [15] resulting in a null bit-mask ($0b000000$). The stress component σ_{yy} then belongs to the fixed stress vector $\boldsymbol{\sigma}_z$, with inversed bit-mask ($0b100110$). When the stress component τ_{xy} is required, again its isolated bit-mask ($0b001000$) is compared with the structural model bit-mask, now resulting in a non-null bit-mask ($0b001000$). In both cases, a bit-wise COUNT operation [15] is performed to identify the index of the stress component with respect to the respective stress vector and finally the entry of the stress vector in the obtained index is returned. It's worthy reinforcing that bit-wise operations are extremely fast, rendering a reduced computational time to the return mapping routine when compared to the manipulation of the full 3D vectors with six components.

In the present work, the Euler backward method is used to integrate the

associative flow rule relations (Eqs. 22 and 24):

$$\boldsymbol{\varepsilon}^{p}_{a,n+1} = \boldsymbol{\varepsilon}^{p}_{a,n} + \Delta\lambda_{n+1}\mathbf{g}_{a,n+1} \tag{29}$$

$$\mathbf{q}_{n+1} = \mathbf{q}_{n} + \Delta\lambda_{n+1}U\mathbf{g}_{a,n+1} \tag{30}$$

where $\mathbf{g}_{a,n+1} = \mathbf{g}_{a}\left(\boldsymbol{\sigma}_{a,n+1}, \overline{\boldsymbol{\sigma}}_{z}, \mathbf{q}_{n+1}\right)$ and $\Delta\lambda_{n+1} = \lambda_{n+1} - \lambda_{n}$ is the finite increment of the isotropic hardening state variable between the time steps t_n and t_{n+1}.

Initially, the plastic state is frozen ($\lambda^{0}_{n+1} = \lambda_n$ implying that $\mathbf{q}^{0}_{n+1} = \mathbf{q}_n$ and $\boldsymbol{\varepsilon}^{p,0}_{a,n+1} = \boldsymbol{\varepsilon}^{p}_{a,n}$) in a elastic predictor step. The trial active stresses $\boldsymbol{\sigma}^{0}_{a,n+1}$ can then be computed form Eq. (11) as:

$$\boldsymbol{\sigma}^{0}_{a,n+1} = \mathbf{C}_{a}\left(\boldsymbol{\varepsilon}_{a,n+1} - \boldsymbol{\varepsilon}^{p}_{a,n}\right) + \boldsymbol{\sigma}_{r} \tag{31}$$

If the elastic predictor state is admissible, i.e. $f(\boldsymbol{\sigma}^{0}_{a,n+1}, \overline{\boldsymbol{\sigma}}_{z}, \mathbf{q}^{0}_{n+1}, \lambda^{0}_{n+1}) \leq 0$, the elasto-plastic tangent operator is simply set to the elastic tangent operator, i.e. $\boldsymbol{\Pi}_{a,n+1} = \mathbf{C}_{a}$ and the return mapping algorithm reaches its end. Otherwise, the isotropic hardening λ_{n+1}, the kinematic hardening $\mathbf{q}_{n+1}$ and the active stresses $\boldsymbol{\sigma}_{a,n+1}$ must be adjusted so that the resulting stress state falls on the yield surface. The resulting nonlinear system of equations can be obtained combining Eqs. (11), (29) and (30) and recalling the consistency condition in Eq. (26):

$$f_{n+1} = 0 \tag{32}$$

$$\mathbf{h}_{n+1} = \mathbf{q}_{n} - \mathbf{q}_{n+1} + \Delta\lambda_{n+1}U\mathbf{g}_{a,n+1} = \mathbf{0} \tag{33}$$

$$\mathbf{r}_{n+1} = \boldsymbol{\sigma}^{0}_{n+1} - \boldsymbol{\sigma}_{n+1} - \Delta\lambda_{n+1}\mathbf{C}_{a}\mathbf{g}_{a,n+1} = \mathbf{0} \tag{34}$$

where $\mathbf{h}_{n+1}$ and $\mathbf{r}_{n+1}$ are the residual back-stress and stress, respectively.

The Newton-Raphson method is then applied to find the desired stress state. Taking the derivative of Eqs. (32) and (34) with respect to λ_{n+1}, $\mathbf{q}_{n+1}$ and $\boldsymbol{\sigma}_{a,n+1}$ at iteration k, provides the plastic corrector as:

$$f^k_{n+1} + \mathbf{g}^k_{a,n+1} \cdot \left(\delta\boldsymbol{\sigma}^k_{n+1} - \delta\mathbf{q}^k_{n+1}\right) - K^k_{p,n+1}\delta\lambda^k_{n+1} = 0 \tag{35}$$

$$\mathbf{h}^k_{n+1} - \left(\mathbf{I} + \Delta\lambda^k_{n+1} U \mathbf{H}^k_{a,n+1}\right)\delta\mathbf{q}^k_{n+1} + \Delta\lambda^k_{n+1} U \mathbf{H}^k_{a,n+1}\delta\boldsymbol{\sigma}^k_{a,n+1} + U\mathbf{g}^k_{a,n+1}\delta\lambda^k_{n+1} = \mathbf{0} \tag{36}$$

$$\mathbf{r}^k_{n+1} - \left(\mathbf{I} + \Delta\lambda^k_{n+1}\mathbf{C}_a\mathbf{H}^k_{a,n+1}\right)\delta\boldsymbol{\sigma}^k_{a,n+1} + \Delta\lambda^k_{n+1}\mathbf{C}_a\mathbf{H}^k_{a,n+1}\delta\mathbf{q}^k_{n+1} - \mathbf{C}_a\mathbf{g}^k_{a,n+1}\delta\lambda^k_{n+1} = \mathbf{0} \tag{37}$$

where:

$$\delta\lambda^k_{n+1} = \lambda^{k+1}_{n+1} - \lambda^k_{n+1} \tag{38}$$

$$\delta\mathbf{q}^k_{n+1} = \mathbf{q}^{k+1}_{n+1} - \mathbf{q}^k_{n+1} \tag{39}$$

$$\delta\boldsymbol{\sigma}^k_{n+1} = \boldsymbol{\sigma}^{k+1}_{n+1} - \boldsymbol{\sigma}^k_{n+1} \tag{40}$$

Solving Eq. (36) for the kinematic hardening update $\delta\mathbf{q}^k_{n+1}$, we have:

$$\delta\mathbf{q}^k_{n+1} = \mathbf{V}^k_{n+1}\left[\mathbf{h}^k_{n+1} + U\left(\Delta\lambda^k_{n+1}\mathbf{H}^k_{a,n+1}\delta\boldsymbol{\sigma}^k_{a,n+1} + \mathbf{g}^k_{a,n+1}\delta\lambda^k_{n+1}\right)\right] \tag{41}$$

where:

$$\mathbf{V}^k_{n+1} = \left(\mathbf{I} + \Delta\lambda^k_{n+1} U \mathbf{H}^k_{a,n+1}\right)^{-1} \tag{42}$$

Combining Eqs. (37) and (41), the active stress update $\delta\boldsymbol{\sigma}^k_{a,n+1}$ can be obtained as:

$$\delta\boldsymbol{\sigma}^k_{a,n+1} = \mathbf{D}^k_{n+1}\left(\mathbf{z}^k_{n+1} - \mathbf{C}_a\mathbf{Q}^k_{n+1}\mathbf{g}^k_{a,n+1}\delta\lambda^k_{n+1}\right) \tag{43}$$

where:

$$\mathbf{Q}_{n+1}^{k} = \mathbf{I} - \Delta\lambda_{n+1}^{k} U \mathbf{H}_{a,n+1}^{k} \mathbf{V}_{n+1}^{k} \tag{44}$$

$$\mathbf{D}_{n+1}^{k} = \left(\mathbf{I} + \Delta\lambda_{n+1}^{k} \mathbf{C}_{a} \mathbf{H}_{a,n+1}^{k} \mathbf{Q}_{n+1}^{k}\right)^{-1} \tag{45}$$

$$\mathbf{z}_{n+1}^{k} = \mathbf{r}_{n+1}^{k} + \Delta\lambda_{n+1}^{k} \mathbf{C}_{a} \mathbf{H}_{a,n+1}^{k} \mathbf{V}_{n+1}^{k} \mathbf{h}_{n+1}^{k} \tag{46}$$

Finally, the isotropic hardening update $\delta\lambda_{n+1}^{k}$ can be obtained combining Eqs. (35), (41) and (43):

$$\delta\lambda_{n+1}^{k} = \frac{f_{n+1}^{k} + \mathbf{g}_{a,n+1}^{k} \cdot \left(\mathbf{D}_{n+1}^{k} \mathbf{z}_{n+1}^{k} - \mathbf{V}_{n+1}^{k} \mathbf{h}_{n+1}^{k}\right)}{K_{p,n+1}^{k} + \mathbf{g}_{a,n+1}^{k} \cdot \left(\mathbf{Q}_{n+1}^{k} \mathbf{D}_{n+1}^{k} \mathbf{C}_{a} \mathbf{Q}_{n+1}^{k} + U \mathbf{V}_{n+1}^{k}\right) \mathbf{g}_{a,n+1}^{k}} \tag{47}$$

The active stresses $\boldsymbol{\sigma}_{a,n+1}$, kinematic hardening $\mathbf{q}_{n+1}^{k}$ and the isotropic hardening λ_{n+1} can be updated via Eqs. (38)-(40). The absolute value of the yield equation $|f_{n+1}|$ and the norms of the residual stress $\|\mathbf{r}_{n+1}\|$ and back-stress $\|\mathbf{h}_{n+1}\|$ are then checked against a given tolerance tol. Once this iterative process converges, the exact elasto-plastic tangent operator $\boldsymbol{\Pi}_{a,n+1}$ can be computed, providing quadratic convergence to the local iterative process. From the consistency condition (Eq. 26), the increment of the yield function f_{n+1} (Eq. 17) must be null:

$$df_{n+1} = \mathbf{g}_{a,n+1} \cdot \left(d\boldsymbol{\sigma}_{a,n+1} - d\mathbf{q}_{n+1}\right) - K_{p,n+1} d\lambda_{n+1} = 0 \tag{48}$$

Taking the infinitesimal increment of the finite evolution-rule at the current time step t_{n+1} in Eq. (30) gives:

$$d\mathbf{q}_{n+1} = \Delta\lambda_{n+1} U \mathbf{H}_{a,n+1} \left(d\boldsymbol{\sigma}_{a,n+1} - d\mathbf{q}_{n+1}\right) + U \mathbf{g}_{a,n+1} d\lambda_{n+1} \tag{49}$$

The infinitesimal increment of the kinematic hardening $d\mathbf{q}_{n+1}$ can then be obtained as:

$$d\mathbf{q}_{n+1} = U \mathbf{V}_{n+1} \left(\Delta\lambda_{n+1} \mathbf{H}_{a,n+1} d\boldsymbol{\sigma}_{a,n+1} + \mathbf{g}_{a,n+1} d\lambda_{n+1}\right) \tag{50}$$

where:

$$\mathbf{V}_{n+1} = \left(\mathbf{I} + \Delta\lambda_{n+1} U \mathbf{H}_{a,n+1}\right)^{-1} \tag{51}$$

Taking the infinitesimal increment of the elasto-plastic constitutive relation at the current time step t_{n+1} in Eq. (11) and recalling Eq. (29), we have:

$$d\boldsymbol{\sigma}_{a,n+1} = \mathbf{C}_a \left[d\boldsymbol{\varepsilon}_{a,n+1} - \Delta\lambda_{n+1} \mathbf{H}_{a,n+1} \left(d\boldsymbol{\sigma}_{a,n+1} - d\mathbf{q}_{n+1} \right) - \mathbf{g}_{a,n+1} d\lambda_{n+1} \right] \tag{52}$$

The infinitesimal increment of the active stresses $d\boldsymbol{\sigma}_{a,n+1}$ can then be obtained as:

$$d\boldsymbol{\sigma}_{a,n+1} = \mathbf{D}_{n+1} \left(\mathbf{C}_a d\boldsymbol{\varepsilon}_{a,n+1} - \mathbf{C}_a \mathbf{Q}_{n+1} \mathbf{g}_{a,n+1} d\lambda_{n+1} \right) \tag{53}$$

where:

$$\mathbf{Q}_{n+1} = \mathbf{I} - \Delta\lambda_{n+1} U \mathbf{H}_{a,n+1} \mathbf{V}_{n+1} \tag{54}$$

$$\mathbf{D}_{n+1} = \left(\mathbf{I} + \Delta\lambda_{n+1} \mathbf{C}_a \mathbf{H}_{a,n+1} \mathbf{Q}_{n+1} \right)^{-1} \tag{55}$$

The hardening infinitesimal increment $d\lambda_{n+1}$ can then be obtained combining Eqs. (48), (50) and (53) as:

$$d\lambda_{n+1} = \frac{\mathbf{g}_{a,n+1} \cdot \mathbf{Q}_{n+1} \mathbf{D}_{n+1} \mathbf{C}_a d\boldsymbol{\varepsilon}_{a,n+1}}{B_{n+1}} \tag{56}$$

where:

$$B_{n+1} = K_{p,n+1} + \mathbf{g}_{a,n+1} \cdot \left(\mathbf{Q}_{n+1} \mathbf{D}_{n+1} \mathbf{C}_a \mathbf{Q}_{n+1} + U \mathbf{V}_{n+1} \right) \mathbf{g}_{a,n+1} \tag{57}$$

Finally, replacing $d\lambda_{n+1}$ from Eq. (56) in Eq. (53), the active stress infinitesimal increment $d\boldsymbol{\sigma}_{a,n+1}$ can be directly related to the active total

strain infinitesimal increment $d\boldsymbol{\varepsilon}_{a,n+1}$ as:

$$d\boldsymbol{\sigma}_{a,n+1} = \boldsymbol{\Pi}_{a,n+1} d\boldsymbol{\varepsilon}_{a,n+1} \tag{58}$$

where the exact elasto-plastic tangent operator $\boldsymbol{\Pi}_{a,n+1}$ is given in closed-form by:

$$\boldsymbol{\Pi}_{a,n+1} = \mathbf{D}_{n+1}\mathbf{C}_a - \frac{(\mathbf{D}_{n+1}\mathbf{C}_a\mathbf{Q}_{n+1}\mathbf{g}_{a,n+1}) \otimes (\mathbf{D}_{n+1}\mathbf{C}_a\mathbf{Q}_{n+1}\mathbf{g}_{a,n+1})}{B_{n+1}} \tag{59}$$

The fixed strains can be optionally computed for post-processing. The elastic fixed strains $\boldsymbol{\varepsilon}^e_{z,n+1}$ can be obtained from Eq. (10) provided the elastic active strains $\boldsymbol{\varepsilon}^e_{a,n+1} = \boldsymbol{\varepsilon}_{a,n+1} - \boldsymbol{\varepsilon}^p_{a,n+1}$. The total fixed strains can be computed as $\boldsymbol{\varepsilon}_{z,n+1} = \boldsymbol{\varepsilon}^e_{z,n+1} + \boldsymbol{\varepsilon}^p_{z,n+1}$, where the plastic fixed strains $\boldsymbol{\varepsilon}^p_{z,n+1}$ are updated via the discrete form of the flow-rule Eq. (23) as:

$$\boldsymbol{\varepsilon}^p_{z,n+1} = \boldsymbol{\varepsilon}^p_{z,n} + \Delta\lambda_{n+1}\mathbf{g}_{z,n+1} \tag{60}$$

A flowchart of the proposed method is shown in Fig. 2. As described above, with the proposed method only the active plastic strains $\boldsymbol{\varepsilon}^p_a$, the kinematic hardening $\mathbf{q}_{n+1}$ and the isotropic hardening are stored at each integration-point. This considerably reduces the memory cost when compared to the standard method. The memory usage per integration-point with the proposed method is shown in Tab. 4. The memory cost reduction between the two methods is illustrated for different structural models in Fig. 3. As expected, the memory cost reduction is larger for structural models with n_a small, as only the required quantities are stored at the integration-points; showing that the proposed method is as efficient as return mapping

algorithms designed to specific stress states, while maintaining its flexibility through a general approach.

Another important feature of the proposed method is its computational efficiency. In order to achieve convergence, only one return mapping loop is executed, involving uniquely with the active stress and strain components. As it will be shown in the numerical examples of Sec. 4, this considerably reduces the required computational time. Aligned with the implemented bit-mask tools for fast extraction and manipulation of the required stress components, the proposed method yields a fast algorithm for general stresses states.

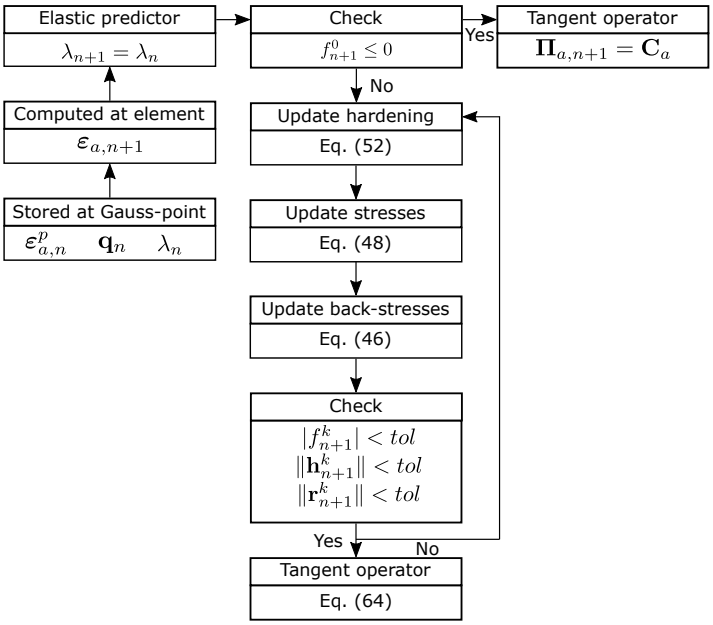

Figure 2: Flowchart of the proposed return mapping algorithm.

The extension of the proposed algorithm for finite strains can be achieved straightforwardly. The elastic constitutive relations must then be modified

Model	$\boldsymbol{\varepsilon}_a^p$	$\mathbf{q}$	λ	Total	Relative
Truss	16	16	16	48 bytes	23.08%
2D Beam	32	32	16	80 bytes	38.46%
3D Beam	48	48	16	112 bytes	53.85%
Shell	80	80	16	176 bytes	84.62%
Plane-stress	48	48	16	112 bytes	53.85%
3D Solid	96	96	16	208 bytes	100.00%

Table 4: Proposed algorithm memory cost per integration-point for different structural models.

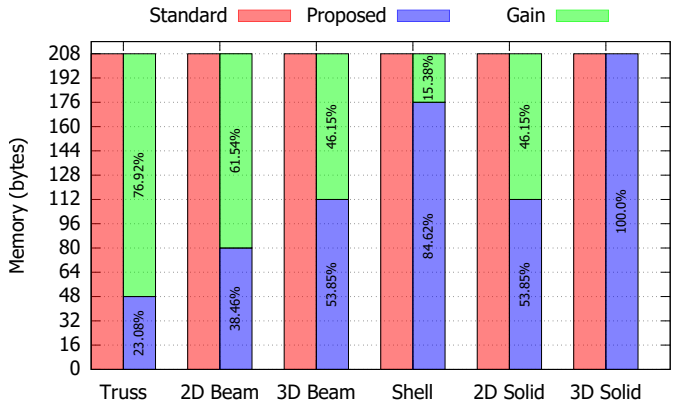

Figure 3: Memory cost comparison between the standard and proposed algorithms for different structural models.

accordingly and the strains must be split in a multiplicative manner. The methodology proposed by Simo [18] can be used to transform the plastic model via a group of special transformations, achieving an additive split of the strains and finally using the proposed return mapping algorithm.

4. Numerical examples

In this section, the proposed algorithm is validated with respect to different structural models and yield surfaces, in order to highlight the general potential of the proposed formulation. Accuracy and computational cost are studied with respect to the standard algorithm, highlighting the advantages of the proposed method. For reference, both the standard and proposed algorithms were implemented using the *C++* programming language [22]. Special care was taken to minimize overhead and perform memory allocation and data setup at initialization so that only the computational time of the return mapping routines was measured. The relative computational time between the two algorithms should be approximately the same for any machine. In this work, the numerical analyses were performed in a Dell Precision 5540 with 64 GB of RAM memory and processor Intel Core I9-9880H (2.30 GHz). Also, to minimize the influence of system applications running in the background, all analysis were performed in a single core.

In all examples, the stress-strain curves were computed with 10^2 steps. In order to evaluate the computational time, in a single test, the curves are constructed 10^5 times, summing up to 10^7 calls of the used return mapping routine. The average value of 10 tests is then used to evaluate the performance of the adopted method. As the proposed algorithm depends only on the strains computed and the data stored at each integration-point, the FE discretization has no influence on the computational efficiency or on the memory cost of the return mapping routine.

4.1. Truss

In this first example, the von Mises material model (Appendix A.1) is used with a integration-point in uni-axial stress state ($\boldsymbol{\sigma}_a = \{\sigma_{xx}\}$), commonly assumed in truss and cable finite element formulations. This simple example is used to illustrate how the proposed algorithm can be used for low-order structural models while maintaining its performance. It can be easily shown that, in this case, the tangent operator of Eq. (59) reduces to the tangent modulus used in uni-dimensional plasticity [20], i.e. $\Pi_{a,n+1} = E(K_p + U)/(E + K_p + U)$. The numerical analysis was performed for a steel with elastic modulus $E = 200$ GPa and Poisson's ratio $\nu = 0.30$.

A linear hardening law was adopted, i.e. $\sigma_y(\lambda) = \sigma_0 + K_p\lambda$ with $\sigma_0 = 400$ MPa. The algorithm was tested in perfect plasticity ($K_p = 0$), hardening ($K_p = E$) and softening ($K_p = -E/3$) with no kinematic hardening ($U = 0$). The analysis was carried out incrementing the axial strain until $\varepsilon_{xx} = 2E/\sigma_0$. The results with the two algorithms match exactly as shown in Fig. 4. The average computational times for the standard and proposed algorithms were 36.4 s and 4.69 s, respectively, showing that the proposed method was 7.76 times faster than the standard method.

The case where the fixed stresses $\overline{\boldsymbol{\sigma}}_z$ are non-zero, with $\sigma_{yy} = \sigma_{zz} = \sigma_z$, is considered next in order to illustrate the capabilities of the proposed formulation. The results obtained with the proposed algorithm and the closed-form solution [20] for different values of σ_z are shown in Fig. 5, where an exact match is observed.

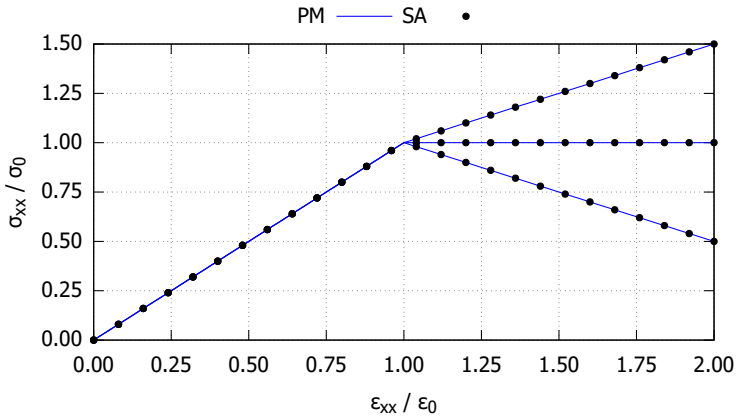

Figure 4: Uni-axial stress-strain relation obtained with the standard algorithm (SA) and proposed algorithm (PA).

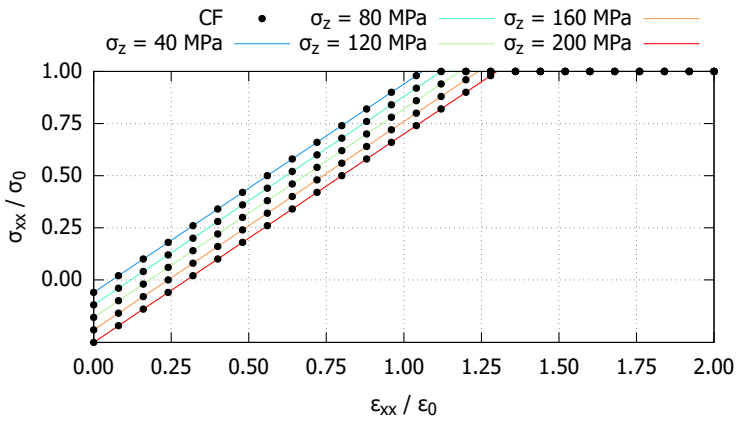

Figure 5: Uni-axial stress-strain relation with non-zero fixed stresses obtained with the proposed algorithm (PA) and closed-form solution [20].

4.2. 2D Beam

In the next example, the Drucker-Prager material model (Appendix A.2) is used with a integration-point in a 2D shear stress state ($\boldsymbol{\sigma}_a = \{\sigma_{xx}, \tau_{xy}\}$),

commonly assumed in 2D Timoshenko's beam finite element formulations. The numerical analysis was performed for a concrete with elastic modulus $E = 40$ GPa, Poisson's ratio $\nu = 0.25$, compressive strength $\sigma_c = 30$ MPa and tension strength $\sigma_t = 3$ MPa.

The analysis was carried out incrementing the axial strain until $\varepsilon_{xx} = 2E/\sigma_c$ and taking $\gamma_{xy} = \beta\varepsilon_{xx}$ for different values of the parameter β within the interval $[0 : 4]$. The results for perfect plasticity ($K_p = 0$ and $U = 0$) are shown in Fig. 6, where the standard and proposed methods provide an exact match. The average computational times for the standard and proposed algorithms were 119.0 s and 13.3 s, respectively, showing that the proposed method was 8.95 times faster than the standard method.

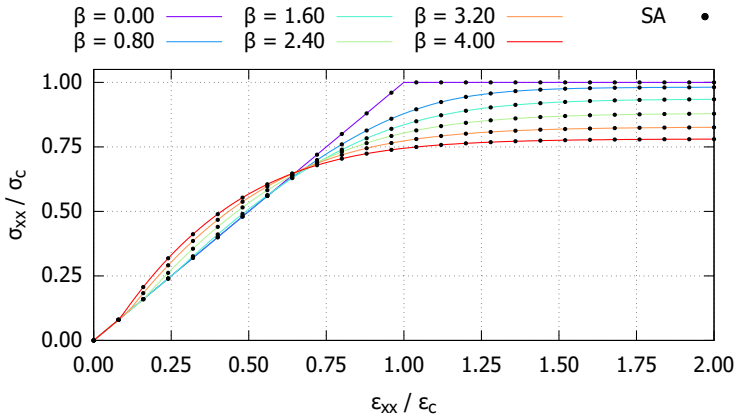

Figure 6: 2D beam stress-strain relation obtained with the standard algorithm (SA) and proposed algorithm.

4.3. 3D Beam

In the next example, the Bresler-Pister material model (Appendix A.3) is used with a integration-point in a 3D shear stress state ($\boldsymbol{\sigma}_a = \{\sigma_{xx}, \tau_{xy}, \tau_{xz}\}$),

commonly assumed in 3D Timoshenko's and Bernoulli's beam finite element formulations. The numerical analysis was performed for a concrete with elastic modulus $E = 40$ GPa, Poisson's ratio $\nu = 0.25$, compressive strength $\sigma_c = 30$ MPa, tension strength $\sigma_t = 3$ MPa and bi-axial compressive strength $\sigma_b = 40$ MPa.

The analysis was carried out incrementing the axial strain until $\varepsilon_{xx} = 2E/\sigma_c$ and taking $\gamma_{xy} = \gamma_{xz} = \beta\varepsilon_{xx}$ for different values of the parameter β within the interval $[0 : 4]$. The results for perfect plasticity ($K_p = 0$ and $U = 0$) are shown in Fig. 7, where the standard and proposed methods provide an exact match. The average computational times for the standard and proposed algorithms were 208.0 s and 19.9 s, respectively, showing that in this case the proposed method was 10.45 times faster than the standard method.

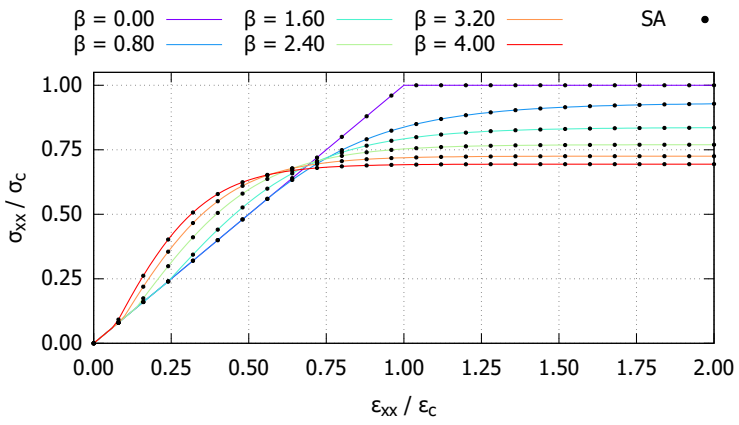

Figure 7: 3D beam stress-strain relation obtained with the standard algorithm (SA) and proposed algorithm.

4.4. Plane-stress

In the next example, a square plate with circular hole (Fig. 8) is studied. The system is under plane-stress, with active stresses $\boldsymbol{\sigma}_a = \{\sigma_{xx}, \sigma_{yy}, \tau_{xy}\}$. The plate is made of steel and the von Mises material model is adopted with elastic modulus $E = 200$ GPa, Poisson's ratio $\nu = 0.30$ and yield stress $\sigma_0 = 400$ MPa. The square has size $a = 2.00$ m, the inner circle has a radius $r = 0.50$ m and the plate thickness is $t = 1.00$ cm. The plate bottom edge is fixed in the vertical direction, while the top edge is subjected to an uniformly distributed load.

The integration-point where plastic deformations first appears is selected for comparison. The active components of the deformation history of this integration-point are used as input for the algorithm. The results for isotropic plasticity ($K_p = 2$ GPa and $U = 0$) are shown in Fig. 9, where the standard and proposed methods provide an exact match. For completeness, the projection algorithm of Simo and Taylor [21] was also used to compare the results, providing an excellent match as well. The average computational times for the standard and proposed algorithms were 51.8 s and 8.04 s, respectively, showing that in this case the proposed method was 6.44 times faster than the standard method.

5. Conclusions

In this paper, a new return mapping algorithm that considers a general partition of the 3D stresses into active and fixed components for arbitrary yield functions is developed. The algorithm operates exclusively on the active stress and strain components, yielding a low memory cost and high compu-

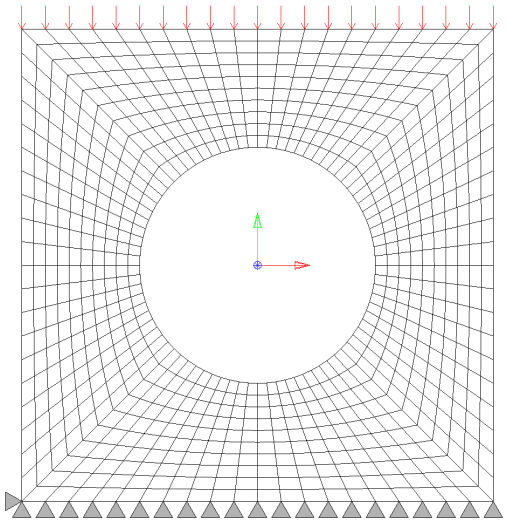

Figure 8: Square with circular under plane-stress loading.

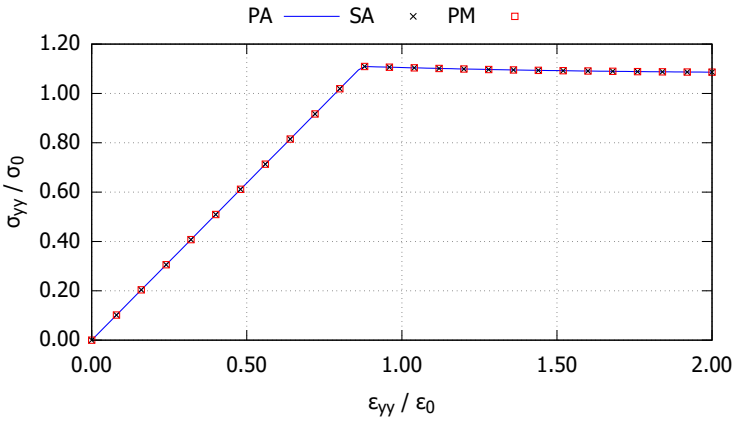

Figure 9: Plane-stress stress-strain relation obtained with the standard algorithm (SA), proposed algorithm (PA) and projection method (PM).

tational efficiency. The exact tangent operator is obtained in closed-form, providing quadratic convergence to the local iterative process. A comparison with the standard approach shows its performance gains (in terms of

memory and computational time), while maintaining the general format of the formulation. Numerical examples demonstrate the applicability of the proposed method for different structural models.

A bit-mask format is used to store the active stress types and perform operations (extraction and update) of the relative strain and stress components. This approach yields an optimal memory cost and simultaneously allows to obtain the stress tensors (yield function, gradient and hessian) required in the return mapping algorithm with minimal computational time.

In a future work, a modification of the algorithm to deal with finite strains can be envisioned. As proposed by Simo [18], a group of special transformations can be applied so that traditional algorithms with additive split can be used. Damage models can also be added to the present return mapping algorithm by modification the constitutive relations accordingly.

Acknowledgements

The authors gratefully acknowledge financial support by the Walloon Region of Belgium and bureau d'études Greisch through the research grant n°8096, "FINELG2020".

References

[1] R. D. Borst and A. E. Groen. A note on the calculation of consistent tangent operators for von mises and drucker-prager plasticity. *Communications in Numerical Methods in Engineering*, 10(12):1021–1025, 1994.

[2] M. Crusells-Girona, F. C. Filippou, and R. L. Taylor. A mixed formulation for nonlinear analysis of cable structures. *Computers and Structures*, 186:50–61, 2017.

[3] O. Giraldo-Londoño and G. H. Paulino. A unified approach for topology optimization with local stress constraints considering various failure criteria: von mises, drucker–prager, tresca, mohr–coulomb, bresler– pister and willam–warnke. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 476(2238):20190861, 2020.

[4] G. Hofstetter and R. L. Taylor. Non-associative drucker-prager plasticity at finite strains. *Communications in Applied Numerical Methods*, 6(8): 583–589, 1990.

[5] P. Jetteur. Implicit integration algorithm for elastoplasticity in plane stress analysis. *Engineering Computations*, 1986.

[6] E. Kirchner, S. Reese, and P. Wriggers. A finite element method for plane stress problems with large elastic and plastic deformations. *Communications in Numerical Methods in Engineering*, 13(12):963–976, 1997.

[7] S. Klinkel and S. Govindjee. Using finite strain 3d-material models in beam and shell elements. *Engineering Computations*, 19(3):254–271, 2002.

[8] C. Lecoutre and J. Vion. Enforcing Arc Consistency using Bitwise Operations. *Constraint Programming Letters (CPL)*, 2:21–35, 2008.

[9] J. Lee and G. L. Fenves. A return-mapping algorithm for plastic-damage

models: 3-d and plane stress formulation. *International Journal for Numerical Methods in Engineering*, 50(2):487–506, 2001.

[10] S. W. Lee, J. W. Yoon, and D. Y. Yang. A stress integration algorithm for plane stress elastoplasticity and its applications to explicit finite element analysis of sheet metal forming processes. *Computers and Structures*, 66(2):301–311, 1998.

[11] P. B. Lourénço, R. de Borst, and J. G. Rots. A plane stress softening plasticity model for orthotropic materials. *International Journal for Numerical Methods in Engineering*, 40(21):4033–4057, 1997.

[12] C. Meng, Z. Tang, M. Chen, and Q. Peng. Return mapping algorithm in principal space for general isotropic elastoplasticity involving multi-surface plasticity and combined isotropic-kinematic hardening within finite deformation framework. *Finite Elements in Analysis and Design*, 150:1–19, 2018.

[13] T. Mánik. A natural vector/matrix notation applied in an efficient and robust return-mapping algorithm for advanced yield functions. *European Journal of Mechanics - A/Solids*, 90:104357, 2021.

[14] N. Ohno, M. Tsuda, and T. Kamei. Elastoplastic implicit integration algorithm applicable to both plane stress and three-dimensional stress states. *Finite Elements in Analysis and Design*, 66:1–11, 2013.

[15] W. Press. *Numerical recipes : the art of scientific computing*. Cambridge University Press, Cambridge, UK New York, 2007.

[16] J. Reddy. On locking-free shear deformable beam finite elements. *Computer Methods in Applied Mechanics and Engineering*, 149(1):113–132, 1997.

[17] C. Sansour. Large strain deformations of elastic shells constitutive modelling and finite element analysis. *Computer Methods in Applied Mechanics and Engineering*, 161(1):1–18, 1998.

[18] J. Simo. Algorithms for static and dynamic multiplicative plasticity that preserve the classical return mapping schemes of the infinitesimal theory. *Computer Methods in Applied Mechanics and Engineering*, 99 (1):61–112, 1992.

[19] J. Simo and S. Govindjee. Exact closed-form solution of the return mapping algorithm in plane stress elasto-viscoplasticity. *Engineering Computations*, 5(3):254–258, 1988.

[20] J. C. Simo and T. J. R. Hughes. *Computational Inelasticity*. Springer-Verlag, 1998.

[21] J. C. Simo and R. L. Taylor. A return mapping algorithm for plane stress elastoplasticity. *International Journal for Numerical Methods in Engineering*, 22(3):649–670, 1986.

[22] B. Stroustrup. *Programming: Principles and Practice Using C++*. Pearson Education, 2014. ISBN 9780133796742.

Appendix A. Yield surface gradients and hessians

In this appendix, the yield surface equation, gradient and hessian for the von Mises, Drucker-Prager and Bresler-Pister material models are presented for completeness.

Appendix A.1. von Mises

The von Mises material model [1] is usually applied to metals and considers that only the deviatoric part of the stresses generates plastic strains. The yield surface equation is given by:

$$f(\boldsymbol{\sigma}) = \sqrt{3J_2(\boldsymbol{\sigma})} - \sigma_y(\lambda) \tag{A.1}$$

where the second deviatoric invariant J_2 is defined as:

$$J_2(\boldsymbol{\sigma}) = \frac{(\sigma_{xx} - \sigma_{yy})^2 + (\sigma_{xx} - \sigma_{zz})^2 + (\sigma_{yy} - \sigma_{zz})^2}{6} + \tau_{xy}^2 + \tau_{xz}^2 + \tau_{yz}^2 \tag{A.2}$$

The term $\sigma_v = \sqrt{3J_2(\boldsymbol{\sigma})}$ is referred as equivalent stress. The yield surface gradient can be computed as:

$$\mathbf{g}(\boldsymbol{\sigma}) = \frac{\partial \sigma_v}{\partial \boldsymbol{\sigma}} = \frac{1}{\sigma_v} \begin{Bmatrix} \sigma_{xx} - (\sigma_{yy} + \sigma_{zz})/2 \\ \sigma_{yy} - (\sigma_{xx} + \sigma_{zz})/2 \\ \sigma_{zz} - (\sigma_{xx} + \sigma_{yy})/2 \\ 3\tau_{xy} \\ 3\tau_{xz} \\ 3\tau_{yz} \end{Bmatrix} \tag{A.3}$$

Finally, the surface hessian is given by:

$$\mathbf{H}(\boldsymbol{\sigma}) = \frac{\partial^2 \sigma_v}{\partial \boldsymbol{\sigma} \partial \boldsymbol{\sigma}} = \frac{1}{\sigma_v} \begin{bmatrix} 1 & -1/2 & -1/2 & 0 & 0 & 0 \\ -1/2 & 1 & -1/2 & 0 & 0 & 0 \\ -1/2 & -1/2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 \end{bmatrix} - \frac{1}{\sigma_v^2} \frac{\partial \sigma_v}{\partial \boldsymbol{\sigma}} \otimes \frac{\partial \sigma_v}{\partial \boldsymbol{\sigma}} \tag{A.4}$$

Appendix A.2. Drucker-Prager

The Drucker-Prager material model [1] is usually applied to fragile materials, e.g. rock and concrete. The yield surface equation depends not only on the second deviatoric invariant J_2 but also on the first stress invariant I_1 (hydrostatic pressure):

$$f(\boldsymbol{\sigma}) = \sqrt{3J_2(\boldsymbol{\sigma})} - A - BI_1(\boldsymbol{\sigma}) \tag{A.5}$$

where:

$$I_1(\boldsymbol{\sigma}) = \sigma_{xx} + \sigma_{yy} + \sigma_{zz} \tag{A.6}$$

The elastic limits are distinct in tension σ_t and compression σ_c and the yield surface has a cap in tension. The parameters A and B can obtained as:

$$A = \frac{2\sigma_t \sigma_c}{\sigma_c + \sigma_t} \tag{A.7}$$

$$B = \frac{\sigma_t - \sigma_c}{\sigma_c + \sigma_t} \tag{A.8}$$

The yield surface gradient can be computed as:

$$\mathbf{g}(\boldsymbol{\sigma}) = \frac{\partial \sigma_v}{\partial \boldsymbol{\sigma}} - B \frac{\partial I_1}{\partial \boldsymbol{\sigma}} \tag{A.9}$$

where:

$$\frac{\partial I_1}{\partial \boldsymbol{\sigma}} = \begin{Bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \end{Bmatrix}^T \tag{A.10}$$

Finally, the yield surface hessian is given by:

$$\mathbf{H}(\boldsymbol{\sigma}) = \frac{\partial^2 \sigma_v}{\partial \boldsymbol{\sigma} \partial \boldsymbol{\sigma}} \tag{A.11}$$

Appendix A.3. Bresler-Pister

The Bresler-Pister material model [3] is an extension of the Drucker-Prager yield surface, including a cap in compression as well. The yield surface equation is given by:

$$f(\boldsymbol{\sigma}) = \sqrt{3J_2(\boldsymbol{\sigma})} - A - BI_1(\boldsymbol{\sigma}) - CI_1^2(\boldsymbol{\sigma}) \tag{A.12}$$

The parameters A, B and C can be related to the elastic limits in uni-axial tension σ_t and compression σ_c and bi-axial compression σ_b as:

$$A = \frac{\sigma_c \sigma_b \sigma_t (\sigma_t + 8\sigma_b - 3\sigma_c)}{(\sigma_t + \sigma_c)(\sigma_t + 2\sigma_b)(2\sigma_b - \sigma_c)} \tag{A.13}$$

$$C = \frac{3\sigma_b \sigma_t - \sigma_b \sigma_c - 2\sigma_c \sigma_t}{(\sigma_t + \sigma_c)(\sigma_t + 2\sigma_b)(2\sigma_b - \sigma_c)} \tag{A.14}$$

$$B = \frac{(\sigma_t - \sigma_c)[4\sigma_b^2 + \sigma_c \sigma_t - \sigma_b(\sigma_c + \sigma_t)]}{(\sigma_t + \sigma_c)(\sigma_t + 2\sigma_b)(2\sigma_b - \sigma_c)} \tag{A.15}$$

The yield surface gradient can be computed as:

$$\mathbf{g}(\boldsymbol{\sigma}) = \frac{\partial \sigma_v}{\partial \boldsymbol{\sigma}} - (B + 2CI_1)\frac{\partial I_1}{\partial \boldsymbol{\sigma}} \tag{A.16}$$

Finally, the yield surface hessian is given by:

$$\mathbf{H}(\boldsymbol{\sigma}) = \frac{\partial^2 \sigma_v}{\partial \boldsymbol{\sigma} \partial \boldsymbol{\sigma}} - 2C\frac{\partial I_1}{\partial \boldsymbol{\sigma}} \otimes \frac{\partial I_1}{\partial \boldsymbol{\sigma}} \tag{A.17}$$

Credit Author Statement

Category 1
Conception and design of study: M. V. B. Santana
Acquisition of data: M. V. B. Santana, P. Keo
Analysis and/or interpretation of data: M. V. B. Santana
Category 2
Drafting the manuscript: M. V. B. Santana
Revising the manuscript critically for important intellectual content: M. V. B. Santana, M. Hjiaj
Category 3
Approval of the version of the manuscript to be published: M. V. B. Santana, P. Keo, M. Hjiaj

Declaration of Competing Interests

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