INTERNATIONAL CENTER FOR NUMERICAL METHODS IN ENGINEERING

Comparing Two Algorithms to Add Large Strains to a Small Strain Finite Element Code

A. Rodríguez Ferran
B. A. Huerta

Publication CIMNE Nº-91, November 1996

COMPARING TWO ALGORITHMS TO ADD LARGE STRAINS TO A SMALL STRAIN FINITE ELEMENT CODE

Antonio Rodríguez-Ferran,[1] and Antonio Huerta, [2†] Member, ASCE

[1] Research Assistant, Departamento de Matemática Aplicada III, E.T.S. de Ingenieros de Caminos, Universitat Politècnica de Catalunya, Campus Nord C-2, E-08034 Barcelona, Spain.

[2] Professor, Departamento de Matemática Aplicada III, E.T.S. de Ingenieros de Caminos, Universitat Politècnica de Catalunya, Campus Nord C-2, E-08034 Barcelona, Spain.

† Corresponding author, e-mail: huerta@etseccpb.upc.es

ABSTRACT

Two algorithms for the stress update (i.e., time-integration of the constitutive equation) in large strain solid mechanics are discussed, with particular emphasis on two issues: the incremental objectivity and the implementation aspects. It is shown that both algorithms are incrementally objective (i.e., they treat rigid rotations properly) and that they can be employed to add large strain capabilities to a small strain finite element code in a simple way. A set of benchmark tests, consisting of simple large deformation paths (rigid rotation, simple shear, extension, extension and compression, dilatation, extension and rotation), have been used to test and compare the two algorithms, both for elastic and plastic analysis. These tests evidence different time integration accuracy for each algorithm. However, it is demonstrated that the, in general, less accurate algorithm gives exact results for shear-free deformation paths.

Keywords: nonlinear computational mechanics, large strain solid mechanics, stress update algorithms, finite element code development, large strain benchmarks.

1. INTRODUCTION

Many problems of interest in physics and engineering are nonlinear [1,2]. Focusing on solid mechanics, [9], two basic types of nonlinearity are encountered: material and geometric. **Material nonlinearity** refers to the plastic (or more generally, inelastic) behavior shown by many engineering materials, such as metals. In some processes, moreover, the solid undergoes such large deformations that the variation in shape may not be neglected, as in a standard linear computation, thus resulting in **geometric nonlinearity**. The two kinds of nonlinearity invalidate, in many cases, a classical linear elastic analysis.

From the viewpoint of continuum mechanics, a convenient way to describe large strains in a solid is by means of a convected frame which is attached to the body and deforms with it, [9, 11]. Since the convected frame follows the body motion, it allows for a simple statement and handling of the constitutive equations, [13]. In nonlinear computational mechanics, however, the standard approach is to use a fixed Cartesian frame, [16], like in the linear case. This leads to a simpler description of motion (because the frame does not change), but, on the other hand, the treatment of constitutive laws becomes more involved (because the frame does not follow the material).

Nonlinear material behavior is often described by a rate-form constitutive equation, relating some measure of the rate of deformation to a rate of stress, [9]. In a large-strain context, the choice of a proper stress rate is a key point, because the principle of objectivity, [5, 10], should be respected: **the constitutive equation must be independent of the observer**. This is only achieved when objective quantities are employed. It can be shown that the material derivative of stress is not an objective tensor and, therefore, an alternative, objective stress rate is needed. The objectivity of the constitutive equations should be respected by the numerical algorithms employed for their time-integration. This requirement is referred to as incremental objectivity, [6].

Various stress update algorithms (i.e., algorithms for the numerical time-integration of the constitutive equations) can be found in the literature (see, for instance, [3, 5, 7]). In many cases, however, employing these algorithms to add large-strain capabilities to a small-strain FE code is a cumbersome task, because new quantities (not employed for a small strain analysis) must be computed.

This paper discusses two incrementally objective algorithms that allow to transform an existing small-strain FE code into a large-strain code **in a simple way**, [14]. Only the particular case in which the elastic part of the deformation is modeled by an hypoelastic law –a common choice in nonlinear computational mechanics– will be addressed here. The first algorithm uses the full Lagrange strain tensor, including quadratic terms to account for large strains. The second algorithm, presented in [12], employs the same strain tensor as in a small-strain analysis, but computed in the midstep configuration.

Various implementation aspects for the two algorithms are commented. It is shown, in particular, that very few additional features must be added to a code with small-strain and nonlinear material behavior to enable its use for large-strain analysis.

The two algorithms are tested and compared with the help of a set of benchmark tests, consisting of simple deformation paths (rigid rotation, simple shear, uniaxial extension, extension and compression, extension and rotation). Moreover, it is demonstrated that for some particular deformation paths the, in general, less accurate algorithm captures the exact solution.

This paper is organized as follows. Some preliminaries, including the basic equations of large strain solid mechanics and the concept of objectivity, are briefly reviewed in Section 2. The two stress update algorithms are presented in Section 3. After some introductory remarks in Section 3.1, the notion of incremental objectivity is reviewed in Section 3.2. The two algorithms are then shown in Section 3.3, and their implementation in a small-strain FE code is discussed in Section 3.4. Section 4 deals with the numerical examples. Finally, some concluding remarks are made in Section 5.

2. PRELIMINARIES

2.1. Basic equations

The first ingredient of continuum mechanics is the equation of motion, $\boldsymbol{x} = \boldsymbol{x}(\boldsymbol{X}, t)$, which yields the position $\boldsymbol{x}$ of material particles, denoted by their material coordinates $\boldsymbol{X}$, at time t, [9]. If the initial spatial coordinates are employed as material coordinates, the material displacements can be defined as $\boldsymbol{u}(t) = \boldsymbol{x}(t) - \boldsymbol{X}$. Once the displacements are defined, the kinematical description continues with strain representation. The starting point is the deformation gradient $\boldsymbol{F}$

$$\boldsymbol{F} = \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{X}}. \tag{1}$$

Various strain tensors may be defined by means of $\boldsymbol{F}$. The Lagrange strain tensor, for instance, is

$$\boldsymbol{E} = \frac{1}{2}\left(\boldsymbol{F}^T \boldsymbol{F} - \boldsymbol{I}\right), \tag{2}$$

where T means transpose and $\boldsymbol{I}$ is the identity. Another tensor representing strain is the spatial gradient of velocity $\boldsymbol{l}$. This tensor yields relevant tensors if decomposed into symmetric part (rate-of-deformation tensor, $\boldsymbol{d}$) and skew-symmetric part (spin tensor, $\boldsymbol{\omega}$),

$$\boldsymbol{l} = \frac{\partial \boldsymbol{v}}{\partial \boldsymbol{x}} = \boldsymbol{d} + \boldsymbol{\omega}. \tag{3}$$

A very common simplification in solid mechanics is that of small deformations. If displacements, rotations and strains are small enough, two important points follow: *i)* the relation between displacements and strain is linear and *ii)* the initial configuration of the body, Ω_0, can be used to solve the governing equations. Because of this, a **geometrically linear** problem results.

In some other problems, on the contrary, displacements are large when compared to the initial dimensions of the body. The relation between displacements and strains is no longer linear and, moreover, the governing equations must be solved over the current configuration Ω_t at time t, not over Ω_0. Since the motion that transforms Ω_0 into Ω_t is precisely the fundamental unknown, a **geometrically nonlinear** problem is obtained.

The balance laws of continuum mechanics state the conservation of mass, momentum and energy, [9]. For a wide range of problems in solid mechanics, three simplifying assumptions are common: *i)* mechanical and thermal effects are uncoupled, *ii)* the density is constant and *iii)* inertia forces are negligible in comparison to the other forces acting on the body (quasistatic process). The mechanical problem is then governed by the momentum balance alone, which becomes a static equilibrium equation,

$$\frac{\partial \sigma_{ij}}{\partial x_j} + b_i = 0, \tag{4}$$

where $\boldsymbol{\sigma}$ is the Cauchy stress tensor and $\boldsymbol{b}$ is the force per unit volume. Equation (4) models many problems of practical interest –including, for instance, various forming processes, see [15].

2.2. Stress tensors

The most common representation of stress is the Cauchy stress tensor $\boldsymbol{\sigma}$, defined in the current configuration Ω_t and already presented in Eq. (4). This tensor has a clear physical meaning, because it involves only forces and surfaces in the current configuration. Experimental stress measures taken in a laboratory correspond to Cauchy stresses, also known as true stresses.

In a large strain context, other representations of stress are possible and indeed useful. The key idea, [12], is that Ω_0 and Ω_t are different configurations, so tensors defined in each configuration cannot be combined by operations such as subtraction and addition. Let ${}^0\boldsymbol{\sigma}$ and ${}^t\boldsymbol{\sigma}$ be the Cauchy stress tensors at the initial time t_0 and current time t respectively; the increment of stress may **not** be defined as ${}^t\boldsymbol{\sigma} - {}^0\boldsymbol{\sigma}$, because the two tensors are referred to different configurations. As stress increments will be needed to update stresses, a proper definition is required.

An alternative representation of stress is the second Piola-Kirchhoff tensor $\boldsymbol{S}$, defined as the pull-back of $\boldsymbol{\sigma}$

$$\boldsymbol{S} = J\boldsymbol{F}^{-1}\boldsymbol{\sigma}\boldsymbol{F}^{-T}, \tag{5}$$

where $J = \det(\boldsymbol{F})$ is the Jacobian of the motion, which reflects the variation of unit volume associated to the deformation, and the inverse of the deformation gradient $\boldsymbol{F}^{-1}$ is employed to transform $\boldsymbol{\sigma}$ from Ω_t to Ω_0, see Figure 1a. Equation (5) is called the pull-back Piola transformation. It must be remarked that $\boldsymbol{S}$ represents the state of stress at time t but referred to configuration Ω_0, and should not be confused with ${}^0\boldsymbol{\sigma}$, the stress at initial time t_0.

Equation (5) may be reversed, and then $\boldsymbol{\sigma}$ may be seen as the push-forward of $\boldsymbol{S}$

$$\boldsymbol{\sigma} = \frac{1}{J}\boldsymbol{F}\boldsymbol{S}\boldsymbol{F}^{T}, \tag{6}$$

where $\boldsymbol{F}$ transforms $\boldsymbol{S}$ from Ω_0 to Ω_t. Equation (6) is the so-called push-forward Piola transformation, Figure 1b. With the help of Eqs. (5) and (6), the stress increment may be represented either as

$${}^t\Delta\boldsymbol{\sigma} = {}^t\boldsymbol{\sigma} - J^{-1}\boldsymbol{F}\ {}^0\boldsymbol{\sigma}\ \boldsymbol{F}^{T} \tag{7a}$$

or

$$
{}^{0}\Delta\boldsymbol{\sigma} = J\boldsymbol{F}^{-1}\,{}^{t}\boldsymbol{\sigma}\,\boldsymbol{F}^{-T} - {}^{0}\boldsymbol{\sigma} \tag{7b}
$$

referred to Ω_t or Ω_0 respectively. The Piola transformations are employed to refer the two tensors to a common configuration, where the subtraction can be properly performed.

2.3. Constitutive equations and objectivity

In nonlinear solid mechanics, the material behavior is often described by a rate-form constitutive equation, relating the stress rate to velocity and/or its derivatives and the stress state (and eventually, some internal variables). The particular case of **hypoelastic materials**, where the stress rate depends linearly on the rate-of-deformation tensor $\boldsymbol{d}$, [9], will be considered here to present the two stress update algorithms. The two algorithms, however, can be extended to elastoplastic problems, by profiting from the decomposition of the rate-of-deformation $\boldsymbol{d}$ into elastic and plastic parts, [8]. The hypoelastic constitutive law is

$$
\dot{\boldsymbol{\sigma}} = \boldsymbol{C} : \boldsymbol{d}, \tag{8}
$$

where $\dot{\boldsymbol{\sigma}}$ is the material rate of stress, and $\boldsymbol{C}$ is the fourth-order modulus tensor. For isotropic materials, $\boldsymbol{C}$ can be written in terms of just two parameters, the Lamé constants λ and μ, just like in classical elasticity, [9].

In fact, Eq. (8) is only valid for small strains. As shown next, the material rate of stress $\dot{\boldsymbol{\sigma}}$ may not be employed to represent stress variation in a large strain problem, because it is not an **objective** tensor.

The **principle of objectivity** is a fundamental requirement regarding the constitutive equation in large strain solid mechanics, [5, 10]: if the constitutive equations really describe the physical behavior of the continuum, they must be independent of the observer. In other words, they must remain invariant under any change of reference frame.

This requirement is fulfilled if objective quantities appear in the constitutive equations. A quantity is said to be objective if it transforms in a proper tensorial manner under a superposed rigid-body motion. Let the rigid motion be represented by an orthogonal rotation tensor $\boldsymbol{Q}$, ($\boldsymbol{Q}^{-1} = \boldsymbol{Q}^T$) and a translation $\boldsymbol{a}$. The time-dependent relation between old and new coordinates is then $\boldsymbol{x}^{\text{new}}(t) = \boldsymbol{Q}(t)\boldsymbol{x} + \boldsymbol{a}(t)$. It is postulated that the Cauchy stress tensor $\boldsymbol{\sigma}$ is objective. As it is a second-order tensor, it transforms according to

$$\boldsymbol{\sigma}^{\text{new}}(t) = \boldsymbol{Q}(t)\boldsymbol{\sigma}(t)\boldsymbol{Q}(t)^T. \tag{9}$$

If Eq. (9) is derived with respect to time, it is readily observed that the material derivative of an objective tensor is not objective:

$$\dot{\boldsymbol{\sigma}}^{\text{new}} = \dot{\boldsymbol{Q}}\boldsymbol{\sigma}\boldsymbol{Q}^T + \boldsymbol{Q}\dot{\boldsymbol{\sigma}}\boldsymbol{Q}^T + \boldsymbol{Q}\boldsymbol{\sigma}\dot{\boldsymbol{Q}}^T \neq \boldsymbol{Q}\dot{\boldsymbol{\sigma}}\boldsymbol{Q}^T. \tag{10}$$

This invalidates the use of $\dot{\boldsymbol{\sigma}}$ as the stress rate in a rate-form constitutive equation. An alternative, objective stress rate $\boldsymbol{\sigma}^*$ is therefore needed. In fact, Eq. (8), which is valid for small strain analysis, provides unrealistic stress distributions in very simple large strain tests, as shown in Appendix 1 for a rigid rotation test.

As for the rate-of-deformation tensor $\boldsymbol{d}$, it can be shown that it is an objective tensor, so it may be employed to represent strains in a constitutive equation. Indeed, the hypoelastic constitutive equation is rewritten, in a large strain framework, as

$$\boldsymbol{\sigma}^* = \boldsymbol{C} : \boldsymbol{d}. \tag{11}$$

The stress rate $\boldsymbol{\sigma}^*$ is not uniquely determined by the objectivity principle. Some classical options reviewed in [12] are the Jaumann rate

$$\boldsymbol{\sigma}^*_J = \dot{\boldsymbol{\sigma}} + \boldsymbol{\sigma}\boldsymbol{\omega} - \boldsymbol{\omega}\boldsymbol{\sigma}, \tag{12a}$$

the Green-Naghdi rate

$$\boldsymbol{\sigma}^*_{GN} = \dot{\boldsymbol{\sigma}} + \boldsymbol{\sigma}\boldsymbol{\Omega} - \boldsymbol{\Omega}\boldsymbol{\sigma}, \tag{12b}$$

and the Truesdell rate

$$\boldsymbol{\sigma}^*_T = \dot{\boldsymbol{\sigma}} - \boldsymbol{l}\boldsymbol{\sigma} - \boldsymbol{\sigma}\boldsymbol{l}^{\boldsymbol{T}} + \text{tr}(\boldsymbol{d})\boldsymbol{\sigma}, \tag{12c}$$

where $\boldsymbol{\Omega}$ is the so-called rate-of-rotation tensor, see [9], and $\text{tr}(\boldsymbol{d})$ denotes the trace of tensor $\boldsymbol{d}$.

It can be easily checked that the terms in the RHS of Eqs. (12) additional to the material rate $\dot{\boldsymbol{\sigma}}$ ensure that the defined rates are indeed objective. Either by means of the spin rate $\boldsymbol{\omega}$, the rate-of-rotation $\boldsymbol{\Omega}$ or the gradient of velocity $\boldsymbol{l}$, the non-objectivity of $\dot{\boldsymbol{\sigma}}$ is compensated, and an objective $\boldsymbol{\sigma}^*$ is obtained.

Regarding the Truesdell rate, it has been defined in Eq. (12c) in terms of the **Eulerian** tensors $\boldsymbol{\sigma}$, $\boldsymbol{l}$ and $\boldsymbol{d}$, referred to the current configuration. An alternative expression, which provides insight into its physical meaning and is useful from an algorithmic viewpoint, see [12], is

$$\boldsymbol{\sigma}^*_T = \frac{1}{J}\boldsymbol{F}\dot{\boldsymbol{S}}\boldsymbol{F}^T. \tag{13}$$

In this equation, the Truesdell rate can be interpreted as the push-forward Piola transformation of the material derivative of the second Piola-Kirchhoff stress tensor $\boldsymbol{S}$. Thus, instead of using the time derivative of the Cauchy stress tensor which yields the non-objective material rate, see Eq. (10), the Truesdell rate is preferred because it is by construction an objective rate. In Eq. (13) it is easily observed that the Truesdell rate proceeds in three steps: *i)* $\boldsymbol{\sigma}$ is pulled-back into $\boldsymbol{S}$, *ii)* the material derivative of $\boldsymbol{S}$ is performed and *iii)* the resulting rate is pushed-forward into the current configuration. Where the rationale is that the material derivative of a material tensor (i.e., a tensor referred to the initial configuration) yields an objective tensor.

3. TWO STRESS UPDATE ALGORITHMS FOR LARGE STRAINS

3.1. Introductory remarks

If the Finite Element Method [4, 16] is employed, the partial differential equation (4) is transformed into the nonlinear system of equations

$$\boldsymbol{r}(\boldsymbol{u}) = \boldsymbol{f}_{\text{int}}(\boldsymbol{u}) - \boldsymbol{f}_{\text{ext}}(\boldsymbol{u}) = \boldsymbol{0}, \tag{14}$$

where $\boldsymbol{f}_{\text{int}}$ is the internal force vector, $\boldsymbol{f}_{\text{ext}}$ is the external load vector and $\boldsymbol{r}$ are the residual forces, which are null if equilibrium is attained. Equation (14) is typically solved incrementally with a displacement-based implicit method, [1]. The fundamental unknowns are then the incremental displacements $\Delta\boldsymbol{u} = {}^{n+1}\boldsymbol{x} - {}^{n}\boldsymbol{x}$ from one (known) equilibrium configuration Ω_n at time t_n to a new (unknown) equilibrium configuration Ω_{n+1} at time $t_{n+1} = t_n + \Delta t$.

Nonlinear systems of equations like Eq. (14) may be solved by a number of iterative techniques, [2]. The two key ideas are that *i)* a linearized form of Eq. (14) is used to predict and then iteratively correct $\Delta \boldsymbol{u}$, and *ii)* the constitutive equation (11) must be integrated after each iteration to check equilibrium. Indeed, each iteration i yields a candidate configuration Ω^i_{n+1}. To check whether it is the equilibrium configuration at time t_{n+1}, stresses must be updated from the previous configuration Ω_n by time-integrating the rate-form constitutive equation. By doing so, the internal forces and the residual forces, Eq. (14), may be computed.

3.2. Incremental objectivity

As remarked in [5], stress update is the central problem in nonlinear solid mechanics and affects fundamentally the accuracy of the overall algorithm.

In the context of the incremental build-up of the solution, it is useful to define the incremental versions of the tensors presented in Eqs. (1) and (2). Let ${}^{n}\boldsymbol{F}$ and ${}^{n+1}\boldsymbol{F}$ be the deformation gradients relating Ω_n and Ω_{n+1} respectively to the reference configuration Ω_0, see Figure 2. The incremental deformation gradient ${}^{n}\boldsymbol{\Lambda}$ is

$$ {}^{n}\boldsymbol{\Lambda} = {}^{n+1}\boldsymbol{F} \; {}^{n}\boldsymbol{F}^{-1}, \tag{15} $$

which refers configuration Ω_{n+1} to configuration Ω_n. The corresponding incremental Lagrange strain tensor is then

$$ {}^{n}\Delta \boldsymbol{E} = \frac{1}{2}\left({}^{n}\boldsymbol{\Lambda}^T \; {}^{n}\boldsymbol{\Lambda} - \boldsymbol{I} \right). \tag{16} $$

The objectivity of the constitutive equation (11) is attained through the definition of objective stress rates, Eqs. (12). **Incremental objectivity** is a requirement on the algorithm for the numerical time-integration of the constitutive equation, which is often presented as the discrete counterpart of the principle of objectivity, [6]. Let the incremental deformation gradient ${}^{n}\boldsymbol{\Lambda}$ relating configurations Ω_n and Ω_{n+1} be an orthogonal tensor ${}^{n}\boldsymbol{R}$. The numerical algorithm is said to be **incrementally objective** if it predicts a stress state at t_{n+1} that is simply a rotation of the stress state at t_n,

$$ {}^{n+1}\boldsymbol{\sigma} = {}^{n}\boldsymbol{R} \; {}^{n}\boldsymbol{\sigma} \; {}^{n}\boldsymbol{R}^T. \tag{17} $$

In other words, an incrementally objective algorithm **assumes** that the body motion between t_n and t_{n+1} is a rigid rotation and rotates stresses in accordance with that assumption, with no spurious stress variations, Eq. (17). It must be remarked, however, that the incremental deformation gradient ${}^n\boldsymbol{\Lambda}$ being an orthogonal tensor ${}^n\boldsymbol{R}$ does not necessarily imply that the true (unknown) body motion between t_n and t_{n+1} is a rigid rotation. For this reason, incremental objectivity is just a reasonable property of the numerical algorithm (i.e., a rigid rotation is assumed when possible) rather than a physical requirement like the principle of objectivity, [13,14].

3.3. Two stress update algorithms for large strains

First stress update algorithm

It is possible to employ the incremental Lagrange strain tensor defined in Eq. (16) as the strain measure in the increment Δt. The stress increment is then

$$ {}^n\Delta\boldsymbol{\sigma} = \boldsymbol{C} : {}^n\Delta\boldsymbol{E}, \tag{18} $$

where the superscript n in $\Delta\boldsymbol{\sigma}$ indicates that this tensorial quantity is, like ${}^n\Delta\boldsymbol{E}$, referred to configuration Ω_n.

In a large-strain context it is no longer valid to compute the new stresses ${}^{n+1}\boldsymbol{\sigma}$ by simply adding the stress increment ${}^n\Delta\boldsymbol{\sigma}$ to the old stresses ${}^n\boldsymbol{\sigma}$, because these latter two tensors are in the configuration Ω_n and ${}^{n+1}\boldsymbol{\sigma}$ is sought in the configuration Ω_{n+1}. It is necessary to transform the tensors adequately by means of the push-forward Piola transformation, Eq. (6). The numerical algorithm for stress update is then

$$ {}^{n+1}\boldsymbol{\sigma} = {}^nJ^{-1}\;{}^n\boldsymbol{\Lambda}\;{}^n\boldsymbol{\sigma}\;{}^n\boldsymbol{\Lambda}^T + {}^nJ^{-1}\;{}^n\boldsymbol{\Lambda}\,({}^n\Delta\boldsymbol{\sigma})\,{}^n\boldsymbol{\Lambda}^T, \tag{19} $$

where the Jacobian ${}^nJ$ is defined as $\det({}^n\boldsymbol{\Lambda})$ and the incremental deformation gradient, Eq. (15), is employed to push-forward both ${}^n\boldsymbol{\sigma}$ and ${}^n\Delta\boldsymbol{\sigma}$ into the new configuration Ω_{n+1}.

This algorithm is incrementally objective: if ${}^n\boldsymbol{\Lambda}$ is an orthogonal tensor, Eq. (16) yields a null strain tensor ${}^n\Delta\boldsymbol{E}$ and Eq. (19) reduces to Eq. (17), thus predicting a rigid rotation of stresses, with no spurious stress variations. Note that the use of the full incremental Lagrange tensor, including quadratic terms, is essential for the incremental objectivity of the algorithm.

Second stress update algorithm

An alternative, more accurate numerical algorithm will be shown next. Following [12], the hypoelastic constitutive equation is written in terms of the Truesdell objective stress rate, Eq. (12c),

$$\boldsymbol{\sigma}^*_T = \boldsymbol{C} : \boldsymbol{d}. \tag{20}$$

A basic ingredient of this algorithm is that $\boldsymbol{d}$ is evaluated in the midstep configuration $\Omega_{n+\frac{1}{2}}$, defined through linear interpolation between Ω_n and Ω_{n+1}. The midstep spatial coordinates are

$$^{n+\frac{1}{2}}\boldsymbol{x} = \frac{1}{2}\left({}^{n}\boldsymbol{x} + {}^{n+1}\boldsymbol{x}\right) = {}^{n}\boldsymbol{x} + \frac{1}{2}\Delta\boldsymbol{u}, \tag{21}$$

the associated deformation gradient is

$$^{n+\frac{1}{2}}\boldsymbol{F} = \frac{1}{2}\left({}^{n}\boldsymbol{F} + {}^{n+1}\boldsymbol{F}\right), \tag{22}$$

and, similarly to Eq. (15), the incremental deformation gradient relating the midstep and final configurations is

$$^{n+\frac{1}{2}}\boldsymbol{\Lambda} = {}^{n+1}\boldsymbol{F}\; {}^{n+\frac{1}{2}}\boldsymbol{F}^{-1}. \tag{23}$$

The different deformation gradient tensors are summarized in Figure 2. Using a midpoint rule algorithm to integrate Eq. (20), the stress update becomes

$$^{n+1}\boldsymbol{\sigma} = {}^{n}J^{-1}\; {}^{n}\boldsymbol{\Lambda}\; {}^{n}\boldsymbol{\sigma}\; {}^{n}\boldsymbol{\Lambda}^T + {}^{n+\frac{1}{2}}J^{-1}\; {}^{n+\frac{1}{2}}\boldsymbol{\Lambda}\left(\Delta t\; \boldsymbol{C} : \boldsymbol{d}\right)\big|_{n+\frac{1}{2}}\; {}^{n+\frac{1}{2}}\boldsymbol{\Lambda}^T. \tag{24}$$

with the Jacobian $^{n+\frac{1}{2}}J$ defined as $\det\left(^{\mathrm{n}+\frac{1}{2}}\boldsymbol{\Lambda}\right)$. As in Eq. (19), tensors referred to the initial and midstep configurations are pushed-forward into the final one by means of the appropriate incremental deformation gradients $^{n}\boldsymbol{\Lambda}$ and $^{n+\frac{1}{2}}\boldsymbol{\Lambda}$.

Recalling the definition of $\boldsymbol{d}$, Eq. (3), the approximation to $^{n+\frac{1}{2}}\boldsymbol{d}$ needed in Eq. (24) will be

$$^{n+\frac{1}{2}}\boldsymbol{d} = \frac{^{n+\frac{1}{2}}\Delta\boldsymbol{\varepsilon}}{\Delta t} = \frac{1}{2\Delta t}\left\{\left[\frac{\partial(\Delta\boldsymbol{u})}{\partial(^{n+\frac{1}{2}}\boldsymbol{x})}\right] + \left[\frac{\partial(\Delta\boldsymbol{u})}{\partial(^{n+\frac{1}{2}}\boldsymbol{x})}\right]^T\right\}. \tag{25}$$

The stress increment is then

$$ {}^{n+\frac{1}{2}}\Delta\boldsymbol{\sigma} = (\Delta t\ \boldsymbol{C} : \boldsymbol{d})\,|_{n+\frac{1}{2}} = \boldsymbol{C} : \frac{1}{2}\left\{ \left[\frac{\partial(\Delta \boldsymbol{u})}{\partial({}^{n+\frac{1}{2}}\boldsymbol{x})}\right] + \left[\frac{\partial(\Delta \boldsymbol{u})}{\partial({}^{n+\frac{1}{2}}\boldsymbol{x})}\right]^T \right\}. \tag{26} $$

It must be noted that in Eq. (25) the strain increment ${}^{n+\frac{1}{2}}\Delta\boldsymbol{\varepsilon}$ is represented by the symmetrized gradient of the incremental displacements, like in a small strain analysis. No additional quadratic terms are needed in Eq. (26), because large strains are properly modeled by employing the midstep configuration to compute the gradient of displacements. As discussed in [12], this algorithm is also incrementally objective. Moreover, the numerical tests of next Section show that its numerical performance (in terms of accuracy) is superior to that of the first algorithm. An accuracy analysis of the two algorithms, showing that the first one is first-order accurate in time and the second one is second-order accurate, can be found in [13].

The two stress update algorithms can also be employed in elastoplasticity. The basic idea is to model the elastic part of the deformation with an hypoelastic law, and use any of the two algorithms to compute the elastic trial stress, [5, 12]. After that, a plastic corrector –a radial return algorithm, for instance–, is required to account for material nonlinearity, [5].

3.4. Implementation aspects

It is shown in this Section that any of the two stress update algorithms can be employed to add large-strain capabilities to a small-strain FE code in a simple way.

The basic idea is that the incremental deformation gradients required in Eqs. (19) and (24) can be computed in a straightforward manner by using quantities that are available in a small strain code. Consider, for instance, the incremental deformation gradient ${}^{n}\boldsymbol{\Lambda}$ relating Ω_n to Ω_{n+1}, Eq. (15). Recalling the definition of $\boldsymbol{F}$ in Eq. (1) and the expression of incremental displacements, it can be easily checked that ${}^{n}\boldsymbol{\Lambda}$ can be put as

$$ {}^{n}\boldsymbol{\Lambda} = \frac{\partial({}^{n+1}\boldsymbol{x})}{\partial({}^{n}\boldsymbol{x})} = \boldsymbol{I} + \frac{\partial(\Delta \boldsymbol{u})}{\partial({}^{n}\boldsymbol{x})}. \tag{27} $$

If an Updated Lagrangian formulation is used, [1], the configuration Ω_n is taken as a reference to compute the incremental displacements. In such a context, ${}^{n}\boldsymbol{\Lambda}$ can be computed from Eq. (27) with the aid of standard nodal shape functions, by expressing $\Delta\boldsymbol{u}$ in terms of the nodal values of incremental displacements. Since the derivatives of shape functions are available in a standard FE code, [4, 16], no new quantities must be computed to obtain ${}^{n}\boldsymbol{\Lambda}$.

Recalling the expression of the incremental Lagrange strain tensor ${}^{n}\Delta\boldsymbol{E}$ in terms of ${}^{n}\boldsymbol{\Lambda}$, Eq. (16), it can written as

$$
{}^{n}\Delta\boldsymbol{E} = \frac{1}{2}\left\{ \left[\frac{\partial(\Delta\boldsymbol{u})}{\partial({}^{n}\boldsymbol{x})}\right] + \left[\frac{\partial(\Delta\boldsymbol{u})}{\partial({}^{n}\boldsymbol{x})}\right]^{T} + \left[\frac{\partial(\Delta\boldsymbol{u})}{\partial({}^{n}\boldsymbol{x})}\right]^{T}\left[\frac{\partial(\Delta\boldsymbol{u})}{\partial({}^{n}\boldsymbol{x})}\right] \right\}. \tag{28}
$$

As for ${}^{n+\frac{1}{2}}\boldsymbol{\Lambda}$, combining Eqs. (21), (22) and (23) renders

$$
{}^{n+\frac{1}{2}}\boldsymbol{\Lambda} = \frac{\partial({}^{n+1}\boldsymbol{x})}{\partial({}^{n+\frac{1}{2}}\boldsymbol{x})} = \boldsymbol{I} + \frac{1}{2}\frac{\partial(\Delta\boldsymbol{u})}{\partial({}^{n+\frac{1}{2}}\boldsymbol{x})}, \tag{29}
$$

so ${}^{n+\frac{1}{2}}\boldsymbol{\Lambda}$ can also be directly computed with the aid of the shape functions, once the configuration of the mesh has been updated from Ω_n to $\Omega_{n+\frac{1}{2}}$.

As a result, the only two additional features that are required to handle large strains are **1.** the updating of mesh configuration and **2.** the computation of incremental gradient gradients, Eqs. (27) and (29). This can be seen by comparing the schematic algorithm for a small-strain analysis with nonlinear material behavior, shown in Box 1 with the large-strain versions, depicted in Box 2a (first stress update algorithm) and Box 2b (second stress update algorithm). In Boxes 2a and 2b (large strains), the modifications with respect to Box 1 (small strains) are highlighted with boldface, and the symbol • is employed to designate additional steps.

FOR EVERY TIME-INCREMENT $[t_n, t_{n+1}]$;

FOR EVERY ITERATION k WITHIN THE TIME-INCREMENT;

1.– Compute the incremental displacements $\Delta \boldsymbol{u}^k$ by solving a linearized form of Eq. (14)

2.– Compute the incremental strains $\Delta \boldsymbol{\varepsilon}^k$ as the symmetrized gradient of displacements:

$$\Delta \boldsymbol{\varepsilon}^k = \frac{1}{2} \left\{ \left[\frac{\partial (\Delta \boldsymbol{u}^k)}{\partial \boldsymbol{X}} \right] + \left[\frac{\partial (\Delta \boldsymbol{u}^k)}{\partial \boldsymbol{X}} \right]^T \right\}$$

3.– Compute the elastic trial incremental stresses $\Delta \boldsymbol{\sigma}^k_{\text{trial}}$ via the elastic modulus tensor:

$$\Delta \boldsymbol{\sigma}^k_{\text{trial}} = \boldsymbol{C} : \Delta \boldsymbol{\varepsilon}^k$$

4.– Compute the elastic trial stresses at t_{n+1}:

$$\boldsymbol{\sigma}^k_{\text{trial}} = {}^n\boldsymbol{\sigma} + \Delta \boldsymbol{\sigma}^k_{\text{trial}}$$

5.– Compute the final stresses $\boldsymbol{\sigma}^k$ at t_{n+1} by performing the plastic correction

6.– Compute the internal forces $\boldsymbol{f}^k_{\text{int}}$ by integrating the stresses $\boldsymbol{\sigma}^k$

7.– Check convergence. If it is not attained, go back to step **1.**

Box 1: Small-strain analysis with nonlinear material behavior

FOR EVERY TIME-INCREMENT $[t_n, t_{n+1}]$;

FOR EVERY ITERATION k WITHIN THE TIME-INCREMENT;

1.– Compute the incremental displacements $\Delta \boldsymbol{u}^k$ by solving a linearized form of Eq. (14)

2.– Compute the incremental strains $\Delta \boldsymbol{\varepsilon}^k$ **accounting for quadratic terms, Eq. (28)**:

$$\Delta \boldsymbol{\varepsilon}^k = \frac{1}{2} \left\{ \left[\frac{\partial(\Delta \boldsymbol{u}^k)}{\partial({}^n\boldsymbol{x})} \right] + \left[\frac{\partial(\Delta \boldsymbol{u}^k)}{\partial({}^n\boldsymbol{x})} \right]^T + \left[\frac{\partial(\Delta \boldsymbol{u}^k)}{\partial({}^n\boldsymbol{x})} \right]^T \left[\frac{\partial(\Delta \boldsymbol{u}^k)}{\partial({}^n\boldsymbol{x})} \right] \right\}$$

3.– Compute the elastic trial incremental stresses $\Delta \boldsymbol{\sigma}^k_{\text{trial}}$ via the elastic modulus tensor:

$$\Delta \boldsymbol{\sigma}^k_{\text{trial}} = \boldsymbol{C} : \Delta \boldsymbol{\varepsilon}^k$$

•.– Compute the incremental deformation gradient ${}^n\boldsymbol{\Lambda}$, Eq. (27), and its determinant ${}^nJ$

4.– Compute the elastic trial stresses at t_{n+1}, **pushing forward both ${}^n\boldsymbol{\sigma}$ and $\Delta \boldsymbol{\sigma}^k_{\text{trial}}$ from configuration Ω_n to Ω_{n+1}, Eq. (19):**

$$\boldsymbol{\sigma}^k_{\text{trial}} = {}^nJ^{-1}\ {}^n\boldsymbol{\Lambda}\ {}^n\boldsymbol{\sigma}\ {}^n\boldsymbol{\Lambda}^T + {}^nJ^{-1}\ {}^n\boldsymbol{\Lambda} \left(\Delta \boldsymbol{\sigma}^k_{\text{trial}} \right) {}^n\boldsymbol{\Lambda}^T$$

•.– Update the configuration from Ω_n to Ω_{n+1} by using the incremental displacements of the current step

5.– Compute the final stresses $\boldsymbol{\sigma}^k$ at t_{n+1} by performing the plastic correction

6.– Compute the internal forces $\boldsymbol{f}^k_{\text{int}}$ by integrating the stresses $\boldsymbol{\sigma}^k$

7.– Check convergence. If it is not attained, **recover configuration** Ω_n and go back to step **1.**

Box 2a: First stress update algorithm

FOR EVERY TIME-INCREMENT $[t_n, t_{n+1}]$;

FOR EVERY ITERATION k WITHIN THE TIME-INCREMENT;

1.– Compute the incremental displacements $\Delta \boldsymbol{u}^k$ by solving a linearized form of Eq. (14)

•.– Update the configuration from Ω_n to $\Omega_{n+\frac{1}{2}}$, Eq. (21)

2.– Compute the incremental strains $\Delta \boldsymbol{\varepsilon}^k$ as the symmetrized gradient of displacements:

$$\Delta \boldsymbol{\varepsilon}^k = \frac{1}{2} \left\{ \left[\frac{\partial(\Delta \boldsymbol{u}^k)}{\partial({}^{n+\frac{1}{2}}x)} \right] + \left[\frac{\partial(\Delta \boldsymbol{u}^k)}{\partial({}^{n+\frac{1}{2}}x)} \right]^T \right\}$$

3.– Compute the elastic trial incremental stresses $\Delta \boldsymbol{\sigma}^k_{\text{trial}}$ via the elastic modulus tensor:

$$\Delta \boldsymbol{\sigma}^k_{\text{trial}} = \boldsymbol{C} : \Delta \boldsymbol{\varepsilon}^k$$

•.– Compute the incremental deformation gradients ${}^{n}\boldsymbol{\Lambda}$ and ${}^{n+\frac{1}{2}}\boldsymbol{\Lambda}$, Eqs. (27) and (29), and its determinants ${}^{n}J$ and ${}^{n+\frac{1}{2}}J$

4.– Compute the elastic trial stresses at t_{n+1}, **pushing forward ${}^{n}\boldsymbol{\sigma}$ and $\Delta \boldsymbol{\sigma}^k_{\text{trial}}$ to Ω_{n+1}, Eq. (24):**

$$\boldsymbol{\sigma}^k_{\text{trial}} = {}^{n}J^{-1}\ {}^{n}\boldsymbol{\Lambda}\ {}^{n}\boldsymbol{\sigma}\ {}^{n}\boldsymbol{\Lambda}^T + {}^{n+\frac{1}{2}}J^{-1}\ {}^{n+\frac{1}{2}}\boldsymbol{\Lambda} \Delta \boldsymbol{\sigma}^k_{\text{trial}}\ {}^{n+\frac{1}{2}}\boldsymbol{\Lambda}^T$$

•.– Update the configuration from $\Omega_{n+\frac{1}{2}}$ to Ω_{n+1}

5.– Compute the final stresses $\boldsymbol{\sigma}^k$ at t_{n+1} by performing the plastic correction

6.– Compute the internal forces $\boldsymbol{f}^k_{\text{int}}$ by integrating the stresses $\boldsymbol{\sigma}^k$

7.– Check convergence. If it is not attained, **recover configuration** Ω_n and go back to step **1.**

Box 2b: Second stress update algorithm

4. NUMERICAL TESTS

4.1. Introductory remarks

The two stress update algorithms presented in Section 3.3 are compared with the help of various simple, large-strain, deformation paths: rigid rotation, simple shear, uniaxial extension, extension and compression, extension and rotation. Both elastic and elastoplastic cases are considered.

Elastic behavior will be represented by Young's modulus E and Poisson's coefficient ν or, equivalently, by the Lamé constants λ and μ. All the tests have been performed with $\nu = 0$, resulting in $\lambda = 0$ and $\mu = E/2$. For the plastic cases, the constitutive law is assumed to be bilinear, with a plastic modulus of $E_p = E/100$ and a yield stress of $\sigma_y = E/2$.

The rigid rotation test demonstrates that, as predicted by theory, both algorithms are incrementally objective. Differences between the two algorithms arise for the other, non-rigid, test cases. The first algorithm is first-order accurate in time, while the second one is second-order accurate, see the proof in [13]. As a consequence, the general behavior is that results depend heavily on the number of time-increments if the first algorithm is employed, and only slightly with the second one. However, the first algorithm shows a superior performance for some stress components in various tests, as shown in Section 4.7 and Appendix 2. For comparison purposes, a small-strain analysis is also performed. The results show that neglecting the large strains lead to qualitatively different responses. The need for a large-strain analysis is clearly demonstrated.

4.2. Rigid rotation

A rectangle $ABCD$ is initially subjected to a uniaxial stress field

$$\left.\begin{aligned} \sigma_{xx} &= 1. \\ \sigma_{xy} &= 0. \\ \sigma_{yy} &= 0. \end{aligned}\right\}, \qquad (30a)$$

and rotated 270° around vertex A, see Appendix 1. Both algorithms are capable of rotating the stress field accordingly,

$$\left.\begin{aligned}\sigma_{xx} &= 0.\\ \sigma_{xy} &= 0.\\ \sigma_{yy} &= 1.\end{aligned}\right\}, \tag{30b}$$

even if only one time-step is employed. This result illustrates the incremental objectivity of the two algorithms.

4.3. Simple shear

The problem statement for the simple shear test is shown in Figure 3, where the initial ($t = 0$) and final ($t = 1$) configurations are presented. The initial stress field is zero. The equations of motion are

$$\left.\begin{aligned}x(t) &= X + Yt\\ y(t) &= Y\end{aligned}\right\}, \tag{31}$$

and the deformation gradient is

$$\boldsymbol{F} = \begin{bmatrix}1 & t\\ 0 & 1\end{bmatrix}. \tag{32}$$

If the Truesdell objective rate is employed, Eq. (20) yields the system of ordinary differential equations

$$\left.\begin{aligned}\dot{\sigma}_{xx} - 2\sigma_{xy} &= 0\\ \dot{\sigma}_{xy} - \sigma_{yy} &= E/2\\ \dot{\sigma}_{yy} &= 0\end{aligned}\right\}, \tag{33}$$

which can be complemented with null initial conditions and solved to provide the analytical solution

$$\left.\begin{aligned}\sigma_{xx}(t) &= \frac{E}{2}t^2\\ \sigma_{xy}(t) &= \frac{E}{2}t\\ \sigma_{yy}(t) &= 0\end{aligned}\right\}. \tag{34}$$

Both algorithms have been employed, with different values of the time-step (1, 2, 3, 5, 10, 20, 50 increments), to integrate the constitutive equation for the given deformation path, Eq. (31), from $t = 0$ to $t = 1$.

Figure 4 presents the results for the elastic case (1, 5, 50 increments). It can be seen that the second algorithm gets, for the three components of stress, the exact analytical values, whereas the first one grossly overestimates stresses when not enough increments are employed, and demands a small time-step to get close to the analytical solution. The output of a small strain analysis (1 increment), is also presented in Figure 4: the null σ_{yy} and linear σ_{xy} are correctly predicted, but this linear analysis is not able to reproduce the quadratic σ_{xx}, Eq. (34).

Since the analytical solution is known, Eq. (34), the error in the final stress ($t = 1$) can be computed and plotted versus the number of time-increments. In a log-log scale, an straight line with a slope equal to the order of the algorithm is expected (that is, 1 for the first algorithm and 2 for the second, see [13]). An average slope can be computed by fitting a straight line via a linear regression. Figure 5 depicts the results for the shear test. Only the first algorithm is shown, because the second one provides the exact analytical solution (except for rounding errors) for any number of time-steps. It can be seen that for the three components of stress, Figures 5a, 5b and 5c, the observed slopes are very close to the expected value of 1 (1.09 for σ_{xx}, 1.16 for σ_{xy} and 1.00 for σ_{yy}). The first-order accuracy of the first algorithm is thus corroborated by this simple test.

The shear test has also been carried out in the elastoplastic case. The results are presented in Figure 6 (1, 10, 50 increments). As in the elastic problem, the first algorithm grossly overestimates the final stress if only one increment is employed, while the second one predicts much more accurate values. With a higher number of time-steps, both algorithms converge to the same response. It may be observed that, of course, when plastification starts (around $t = 0.6$), the curves differ from their elastic counterparts of Figure 4.

A small strain analysis, this time with 50 increments to account for material nonlinearity, is again not satisfactory. As commented above, σ_{xx} is incorrectly kept at its zero initial value throughout the first steps of the computation, with elastic behavior. As σ_{yy} is also zero, the only non-zero stress is σ_{xy}, and the elastic stress tensor is

$$\boldsymbol{\sigma} = \begin{bmatrix} 0 & \sigma_{xy} \\ \sigma_{xy} & 0 \end{bmatrix}, \tag{35}$$

that is, a fully deviatoric tensor. As the plastic correction, once plastification begins, is carried precisely along the deviatoric part of the stress tensor, σ_{xy} is the only component of stress affected by plastification, while σ_{xx} and σ_{yy} keep their elastic behavior. The need for a large strain analysis is again demonstrated.

4.4. Uniaxial extension

A unit square is subjected to uniaxial extension in the x direction, see Figure 7. The equations of motion are

$$\left. \begin{aligned} x(t) &= X(1+t) \\ y(t) &= Y \end{aligned} \right\}, \tag{36}$$

and the deformation gradient is

$$\boldsymbol{F} = \begin{bmatrix} 1+t & 0 \\ 0 & 1 \end{bmatrix}. \tag{37}$$

The analytical solution of Eq. (20) is, this case

$$\left. \begin{aligned} \sigma_{xx}(t) &= Et \\ \sigma_{xy}(t) &= 0 \\ \sigma_{yy}(t) &= 0 \end{aligned} \right\}. \tag{38}$$

Both algorithms correctly predict null values for σ_{xy} and σ_{yy}. Differences are found, on the contrary, for σ_{xx}: while the first algorithm grossly overestimates it, the second one slightly underestimates it, see Figure 8. It can be seen in Figure 8 that, for this particular test, a small strain analysis, with one time-step, produces the exact solution. This is due to the fact that the two sources of error (neglecting quadratic terms in the strain tensor and the time-discretization error) compensate each other. Of course, this is not the situation in a general case, as shown for the other tests.

The error in the final value of σ_{xx} versus the number of time-steps is presented in Figure 9. The observed slopes for the two algorithms are in this case 1.13 for the first algorithm and 1.95 for the second one, in very good agreement with expected values of 1 and 2 respectively.

The plastic response is presented in Figure 10. When plastification begins at $t = 0.5$, the plastic correction is performed along the deviatoric part of the stress tensor, thus resulting in an increase of σ_{yy} at the expense of σ_{xx}. Again, the second algorithm behaves much better than the first one if a small number of time-increments is employed, especially for σ_{xx}, see Figure 10a. With a large number of time-steps (50), the two algorithms provide very similar results, different from the small strain analysis, see Figure 10b.

4.5. Extension and compression

A unit square undergoes extension in the x direction and compression in the y direction, with no change in volume, see Figure 11. The equations of motion are

$$\left.\begin{aligned} x(t) &= X(1+t) \\ y(t) &= Y/(1+t) \end{aligned}\right\}, \tag{39}$$

the deformation gradient is

$$\boldsymbol{F} = \begin{bmatrix} 1+t & 0 \\ 0 & 1/(1+t) \end{bmatrix}, \tag{40}$$

and the analytical solution of Eq. (20) is

$$\left.\begin{aligned} \sigma_{xx}(t) &= E\left(t + \frac{t^2}{2}\right) \\ \sigma_{xy}(t) &= 0 \\ \sigma_{yy}(t) &= \frac{E}{2}\left[\frac{1}{(1+t)^2} - 1\right] \end{aligned}\right\}. \tag{41}$$

Again, both algorithms are capable of predicting null σ_{xy} for the elastic test, but differences appear for σ_{xx} and σ_{yy}, see Figure 12. The observed orders are in this case 1.15 (first algorithm) and 1.93 (second algorithm) for σ_{xx}, see Figure 13a, and 0.87 (first algorithm) and 1.99 (second algorithm) for σ_{yy}, see Figure 13b.

As for the plastic test, results are shown in Figure 14. Once again, convergence to the "reference" solution (with 50 time-increments) is faster with the second algorithm than with the first one, while the small strain analysis yields a qualitatively different response.

4.6. Dilatation

A unit square undergoes biaxial extension, see Figure 15. The equations of motion are

$$\left.\begin{aligned} x(t) &= X(1+t) \\ y(t) &= Y(1+t) \end{aligned}\right\}, \tag{42}$$

resulting in the deformation gradient

$$\boldsymbol{F} = \begin{bmatrix} 1+t & 0 \\ 0 & 1+t \end{bmatrix}. \tag{43}$$

The analytical solution of Eq. (20) is

$$\left.\begin{aligned} \sigma_{xx}(t) &= E \ \ln(1+t) \\ \sigma_{xy}(t) &= 0 \\ \sigma_{yy}(t) &= E \ \ln(1+t) \end{aligned}\right\}. \tag{44}$$

As in the two previous tests, both algorithms provide qualitatively correct results, in the sense that σ_{xy} is zero and σ_{xx} equals σ_{yy} for any number of time-steps and in both elastic and plastic modes. There are sharp differences, however, concerning convergence behavior. For the elastic case, for instance, the second algorithm provides a better prediction with one time-increment than the first one with five, see Figure 16. It can be seen in Figure 17 that, once again, the algorithms behave as expected. The computed slopes are in this case 1.11 for the first algorithm and 1.97 for the second one.

A similar comparison is valid in the plastic test, where one increment with the second algorithm gets closer to the reference solution than ten steps of the first algorithm, see Figure 18.

In this last deformation path, a unit square undergoes a uniaxial extension and a superposed rigid rotation, see Figure 19. The equations of motion are

$$\left.\begin{aligned} x(t) &= X(1+t)\cos(2\pi t) - Y\sin(2\pi t) \\ y(t) &= X(1+t)\sin(2\pi t) + Y\cos(2\pi t) \end{aligned}\right\}, \tag{45}$$

the deformation gradient is

$$\boldsymbol{F} = \begin{bmatrix} (1+t)\cos(2\pi t) & -\sin(2\pi t) \\ (1+t)\sin(2\pi t) & \cos(2\pi t) \end{bmatrix}, \tag{46}$$

and the analytical solution of Eq. (20) is

$$\left.\begin{aligned} \sigma_{xx}(t) &= Et\ \cos^2(2\pi t) \\ \sigma_{xy}(t) &= Et\ \sin^2(2\pi t) \\ \sigma_{yy}(t) &= Et\ \sin(2\pi t)\cos(2\pi t) \end{aligned}\right\}, \tag{47}$$

which is, as expected, a rotation of the solution to the uniaxial extension test, Eq. (38).

The output of the elastic analysis with 1, 5 and 50 time-steps can be seen in Figure 20. If only one increment is employed, the rotation part of the motion is not captured and the predicted stress is identical to that of the uniaxial extension test. With a higher number of steps, the comparative performance of the two algorithms is different from that of the previous tests. For the stress components σ_{xy} and σ_{yy}, for instance, the first algorithm is the one which predicts correct null values for any number of time-increments. In fact, this result illustrates a more general behavior. A unified treatment of shear-free deformation paths can be found in Appendix 2. It is shown that the first algorithm correctly predicts null shear stresses for any number of time-steps, while the second one does not.

As for the stress component σ_{xx}, the first algorithm performs better if a reduced number of time-steps (5) is employed, and a larger number is required for the second algorithm to produce more accurate results. This behavior is illustrated by Figure 21a, where the error curves for σ_{xx} of the two algorithms intersect each other. Again, the observed order of both schemes (1.07 for the first one and 2.30 for the second one) is in accordance with the expected values. Figures 21b and 21c show the convergence behavior of the second algorithm for the other two stress components. It can be seen that the convergence to the exact analytical value is very fast, especially for σ_{yy}, Figure 21c.

The outcome of the plastic analysis is depicted in Figure 22. Once again, a small-strain analysis with nonlinear material behavior turns out to be completely unsatisfactory, providing a solution which is qualitatively different from that of a large-strain analysis.

5. CONCLUDING REMARKS

Two numerical stress update algorithms for large strains have been discussed. The first one uses the full incremental Lagrange strain tensor, including quadratic terms, as the strain measure. The second one works in the midstep configuration, where the symmetrized gradient of the displacement increment can be employed as the strain measure, as in a small strain analysis.

Any of the two algorithms may be employed to enhance a small strain finite element code into a large strain code in a simple way. In particular, it has been shown that if the code already handles material nonlinearity, adding large strains only involves two additional features: the updating of the mesh configuration and the computation of incremental deformation gradients, which are readily available from standard shape functions.

These two algorithms have been compared with the aid of a set of simple deformation paths (rigid rotation, simple shear, uniaxial extension, extension and compression, extension and rotation), both in elasticity and elastoplasticity. The rigid rotation test shows that the two schemes are incrementally objective, that is, they rotate the stresses accordingly and yield no spurious stress variations. For the other, non-rigid tests, the results show that the second algorithm is generally more accurate that the first one, the main reason being the use of the midstep configuration as a reference. In fact, it can be shown that the first algorithm is first-order accurate, and the second one is second-order accurate. Accuracy, however, is not the only relevant point: for shear-free deformation paths, there is no error in the shear component of the stress tensor computed with the first algorithm. In the numerical tests, the two algorithms behave in very good agreement with their expected behavior. Analytical solutions are provided for the elastic analyses, which can be employed to verify the implementation of these (or any other) stress update algorithms.

ACKNOWLEDGEMENTS

The authors thank the graduate student A. Vila for carrying out the various numerical tests. The authors also want to express their appreciation to Prof. A. Millard for fruitful discussions on the topic of large strains. The financial support from DGICYT PB94-1200 and the Human Capital and Mobility ERB-4050-PL-92-2002 is gratefully acknowledged.

APPENDIX 1: MATERIAL STRESS RATE VERSUS OBJECTIVE STRESS RATE

It has been commented in Section 2.3 that the material stress rate cannot be employed in rate-form constitutive equations for large strains, because it is not objective. This fact is illustrated here with a simple test.

Consider the rigid body rotation problem depicted in Figure A.1. The rectangle $ABCD$ is initially subjected to a uniaxial stress σ^o_{xx} and rotates around vertex A with a constant angular velocity ω. The velocity field is

$$(v_x, v_y) = (-\omega\, y, \omega\, x)\,, \tag{A.1}$$

and the spatial gradient of velocity $\boldsymbol{l}$ is a skew-symmetric tensor, thus coincident with the spin rate $\boldsymbol{\omega}$,

$$\boldsymbol{l} = \begin{bmatrix} 0 & -\omega \\ \omega & 0 \end{bmatrix} = \boldsymbol{\omega}. \tag{A.2}$$

As expected, the rate-of-deformation $\boldsymbol{d}$ is a null tensor in this rigid-body, strain-free motion. If $\dot{\boldsymbol{\sigma}}$ is selected as stress rate, Eq. (8), then

$$\boldsymbol{d} = \boldsymbol{0} \quad \Rightarrow \quad \dot{\boldsymbol{\sigma}} = \boldsymbol{C} : \boldsymbol{d} = \boldsymbol{0}. \tag{A.3}$$

Since $\dot{\boldsymbol{\sigma}} = \mathbf{0}$, the stress distribution is constantly equal to the initial one and does not follow the rotation. After a 270° turn, for example, the stress of Figure A.2 is predicted, where the uniaxial stress is now through the short dimension of the rectangle $ABCD$. This result is fully unsatisfactory; a simple rotation of the initial stress should be obtained, see Figure A.3. This correct answer can be achieved by using an objective stress rate, as commented in Section 4.2.

APPENDIX 2: STRESS UPDATE FOR SHEAR-FREE DEFORMATION PATHS

Here, a particular behavior of the first stress update algorithm is demonstrated. This algorithm correctly predicts null shear stresses for any shear-free deformation path. The second algorithm, on the contrary, only provides similar results for some particular deformation paths of this type.

The general expression for shear-free deformation paths is

$$\left.\begin{aligned} x(t) &= X a_x(t) \cos\theta(t) - Y a_y(t) \sin\theta(t) \\ y(t) &= X a_x(t) \sin\theta(t) + Y a_y(t) \cos\theta(t) \end{aligned}\right\}, \qquad (A.4)$$

where $a_x(t)$ and $a_y(t)$ represent the axial deformations in the x and y directions respectively, and $\theta(t) = 2\pi t$ accounts for the rigid rotation. Taking $a_x(t) = 1 + t$ and $a_y(t) = 1$ yields the extension-rotation test of Section 4.7. An extension-compression-rotation test with no change in volume can be obtained with $a_x(t) = \frac{1}{a_y(t)} = 1 + t$, and choosing $a_x(t) = a_y(t) = 1 + t$ renders a dilatation-rotation test.

The deformation gradient of motion (A.4) is

$$\boldsymbol{F} = \begin{bmatrix} a_x(t)\cos\theta(t) & -a_y(t)\sin\theta(t) \\ a_x(t)\sin\theta(t) & a_y(t)\cos\theta(t) \end{bmatrix}, \qquad (A.5)$$

and the Jacobian is $J = a_x(t) a_y(t)$.

To assess the behavior of the two stress update algorithms, it is enough to study the stress predicted after one time-step, and to check if the shear stress component (in rotated axes, to account for the rigid rotation) is null or not.

First stress update algorithm

For the first time-step Δt, the incremental Lagrange strain tensor is, Eq. (16),

$$\Delta \boldsymbol{E} = \frac{1}{2}\begin{bmatrix} (A_x^2 - 1) & 0 \\ 0 & (A_y^2 - 1) \end{bmatrix}, \tag{A.6}$$

with $A_x = a_x(\Delta t)$ and $A_y = a_y(\Delta t)$. Recalling the expression of the first algorithm, Eq. (19), the stress after one time-step is

$$\boldsymbol{\sigma} = \frac{E}{2A_xA_y}\begin{bmatrix} \left(B_x\cos^2\Theta + B_y\sin^2\Theta\right) & \left(B_x\sin\Theta\cos\Theta - B_y\sin\Theta\cos\Theta\right) \\ \left(B_x\sin\Theta\cos\Theta - B_y\sin\Theta\cos\Theta\right) & \left(B_x\sin^2\Theta + B_y\cos^2\Theta\right) \end{bmatrix}, \tag{A.7}$$

with $\Theta = 2\pi\Delta t$, $B_x = A_x^2(A_x^2 - 1)$ and $B_y = A_y^2(A_y^2 - 1)$.

If the stress $\boldsymbol{\sigma}$ is transformed into the rotated stress $\boldsymbol{\sigma}' = \boldsymbol{R}\boldsymbol{\sigma}\boldsymbol{R}^T$, where $\boldsymbol{R}$ is a orthogonal tensor that accounts for the rigid rotation, the result is

$$\boldsymbol{\sigma}' = \frac{E}{2}\begin{bmatrix} \frac{A_x}{A_y}(A_x^2 - 1) & 0 \\ 0 & \frac{A_y}{A_x}(A_y^2 - 1) \end{bmatrix}. \tag{A.8}$$

Equation (A.8) shows that null shear stresses $\sigma_{x'y'}$ are predicted, independently of the axial deformations A_x and A_y. It can be easily checked that the analysis just performed for the first time-step can be extended to the successive increments. In conclusion, the first algorithm predicts null shear stresses (in the rotated axes) $\sigma_{x'y'}$ for any number of time-steps and for any choice of $a_x(t)$ and $a_y(t)$.

Second stress update algorithm

A similar analysis has been performed for the second algorithm. Since the midstep configuration is employed, the required computations are rather cumbersome in this case. The incremental displacements for the first time-step can be computed from Eq. (A.4). After that, the midstep coordinates are obtained following Eq. (21). Then the incremental displacements are written in terms of the midstep coordinates, so the strain increment can be written as, Eq. (25),

$$^{n+\frac{1}{2}}\Delta\boldsymbol{\varepsilon} = \frac{2}{A_xA_y + (A_x + A_y)\cos\Theta + 1} \begin{bmatrix} A_xA_y + (A_x - A_y)\cos\Theta - 1 & (A_x - A_y)\sin\Theta \\ (A_x - A_y)\sin\Theta & A_xA_y - (A_x - A_y)\cos\Theta - 1 \end{bmatrix} \tag{A.9}$$

Once the strain increment is known, the stress $\boldsymbol{\sigma}$ is computed by employing the incremental deformation gradient, Eqs. (24) and (29), and then transformed into the rotated stress $\boldsymbol{\sigma}'$. The final result is that the shear stress $\sigma_{x'y'}$ is proportional to a factor that depends on axial deformations A_x and A_y:

$$\sigma_{x'y'} \sim -A_x^3 A_y^2 + A_x^2 A_y^3 + A_x^2 A_y - A_x A_y^2. \qquad (A.10)$$

It can be seen from Eq. (A.10) that $\sigma_{x'y'}$ is not null for any choice of $a_x(t)$ and $a_y(t)$. Taking $a_x(t) = 1 + t$ and $a_y(t) = 1$, for instance, yields a non-null $\sigma_{x'y'}$, as shown for the extension-rotation test of Section 4.7. On the contrary, for $a_x(t) = a_y(t)$ (dilatation-rotation test) and for $a_x(t)a_y(t) = 1$ (extension-compression-rotation test, with no volume change), null values for $\sigma_{x'y'}$ are predicted. In conclusion, the second stress update algorithm predicts correct null shear stresses (in the rotated axes) for some particular deformation paths of the form (A.4), but not for every shear-free path.

APPENDIX 3: REFERENCES

[1] Bathe, K.J. (1982), Finite Element Procedures in Engineering Analysis, Prentice Hall, New Jersey, USA.

[2] Crisfield, M.A. (1991), Non-linear Finite Element Analysis of Solids and Structures, John Wiley & Sons Ltd., England.

[3] Healy, B.E. and Dodds, R.H. (1992), "A Large Strain Plasticity Model for Implicit Finite Element Analyses", *Computational Mechanics*, Vol. 9, pp. 95-112.

[4] Hughes, T.J.R. (1987), The Finite Element Method, Prentice Hall International, Stanford, USA.

[5] Hughes, T.J.R. (1984), Numerical Implementation of Constitutive Models: Rate-Independent Deviatoric Plasticity, Chapter 2 of Theoretical Foundation for Large-scale Computations for Non-linear Material Behavior, Eds. S. Nemat-Nasser, R.J. Asaro and G.A. Hegemier, Martinus Nijhoff Publishers, Dordrecht.

[6] Hughes, T.J.R. and Winget, J. (1980), "Finite Rotation Effects in Numerical Integration of Rate Constitutive Equations Arising in Large-Deformation Analysis", *Int. J. Num. Meth. Engrg.*, Vol. 15, pp. 1862-1867.

[7] Key, S.W. and Krieg, R.D. (1982), "On the Numerical Implementation of Inelastic Time Dependent and Time Independent, Finite Strain Constitutive Equations in Structural Mechanics", *Comp. Meths. Appl. Mech. Engrg.*, Vol. 33, pp. 439-452.

[8] Khan, A.K. and Huang, S. (1995), Continuum Theory of Plasticity, John Wiley & Sons Inc., New York.

[9] Malvern, L.W. (1969), Introduction to the Mechanics of a Continuous Medium, Prentice-Hall Series in Engineering of the Physical Sciences, Englewood Cliffs, New Jersey.

[10] Marsden, J.E. and Hughes, T.J.R. (1983), Mathematical Foundations of Elasticity, Prentice Hall, USA.

[11] Pegon, P. and Guélin, P. (1986), "Finite Strain Plasticity in Convected Frames", *Int. J. Num. Meth. Engrg.*, Vol. 22, pp. 521-545.

[12] Pinsky, P.M., Ortiz, M. and Pister, K.S. (1983), "Numerical Integration of Rate Constitutive Equations in Finite Deformation Analysis", *Comp. Meths. Appl. Mech. Engrg.*, Vol. 40, pp. 137-158.

[13] Rodríguez-Ferran, A., Pegon, P. and Huerta, A. (1996), "Accuracy Analysis of Two Stress Update Algorithms for Large Strain Solid Mechanics", Research Report n. 92, Int. Centre for Numerical Methods in Engng. (CIMNE), Barcelona.

[14] Rodríguez-Ferran, A. and Huerta, A. (1994), "A Comparison of Two Objective Stress Rates in Object-Oriented Codes", Monograph CIMNE No. 26, ISBN: 84-87867-52-9, Barcelona.

[15] Simulation of Materials Processing: Theory, Methods and Applications, Proc. of the Fifth Int. Conf. on Numer. Meths. in Industrial Forming Processes (NUMIFORM'95), Ithaca, New York, 1995.

[16] Zienkiewicz, O.C. and Taylor, R.L. (1991), The Finite Element Method, Vols. 1 and 2, Mc Graw Hill, London.

Figure 1.– Piola transformations. a) Pull-back Piola transformation. b) Push-forward Piola transformation

Figure 2.– Deformation gradients in an incremental analysis

Figure 3.– Simple shear test. Initial and final configurations

Figure 4.– Simple shear test, elastic analysis. Stress vs. time curves computed with the two algorithms (1, 5 and 50 time-steps) and small-strain analysis (1 time-step). a) σ_{xx}. b) σ_{xy}. c) σ_{yy}

Figure 5.– Simple shear test, elastic analysis. Error in the final value of stress ($t = 1$) vs. number of time-steps (1, 2, 3, 5, 10, 20, 50). a) σ_{xx}. b) σ_{xy}. c) σ_{yy}

Figure 6.– Simple shear test, elastoplastic analysis. Stress vs. time curves computed with the two algorithms (1, 10 and 50 time-steps) and small-strain analysis (50 time-steps). a) σ_{xx}. b) σ_{xy}. c) σ_{yy}

Figure 7.– Uniaxial extension test. Initial and final configurations

Figure 8.– Uniaxial extension test, elastic analysis. Horizontal normal stress σ_{xx} vs. time curves computed with the two algorithms (1, 5 and 50 time-steps) and small-strain analysis (1 time-step)

Figure 9.– Uniaxial extension test, elastic analysis. Error in the final value of horizontal normal stress σ_{xx} ($t = 1$) vs. number of time-steps (1, 2, 3, 5, 10, 20, 50).

Figure 10.– Uniaxial extension test, elastoplastic analysis. Stress vs. time curves computed with the two algorithms (1, 10 and 50 time-steps) and small-strain analysis (50 time-steps). a) σ_{xx}. b) σ_{yy}

Figure 11.– Extension and compression test. Initial and final configurations

Figure 12.– Extension and compression test, elastic analysis. Stress vs. time curves computed with the two algorithms (1, 5 and 50 time-steps) and small-strain analysis (1 time-step). a) σ_{xx}. b) σ_{yy}