Asociación Argentina
de Mecánica Computacional

Mecánica Computacional Vol XXVII, págs. 1003-1042 (artículo completo)
Alberto Cardona, Mario Storti, Carlos Zuppa. (Eds.)
San Luis, Argentina, 10-13 Noviembre 2008

ACCCURATE, EFFICIENT AND ROBUST EXPLICIT AND IMPLICIT INTEGRATION SCHEMES FOR THE ARRUDA-BOYCE VISCOPLATIC MODEL

Ignacio Tomaš, Adrián P. Cisilino, P. M. Frontini

Instituto de Tecnología y Ciencia de Materiales - INTEMA, Universidad Nacional de Mar del Plata, J.B. Justo 4302, B7608 FDQ, Mar del Plata. ignaciotomas@fi.mdp.edu.ar, cisilino@fi.mdp.edu.ar

Keywords: Arruda-Boyce; unified viscoplasticity; Padé approximants; numerically obtained tangent stiffness matrix; time-stepping algorithm; Reduced Backward Euler.

Abstract. Starting from the mathematical structure and kinematic framework of the Arruda-Boyce viscoplastic model, a great variety of ideas can be tested to develop advanced constitutive models for polymers, biomaterials and soft tissues. Moreover, the primitive Arruda-Boyce constitutive model has been the cornerstone for the development of quite sophisticated models to reproduce large-strain mechanical behavior of polymers and soft biological tissues.
However, it is noteworthy that it is still impossible to find a scientific article (or book) providing a detailed description of explicit or implicit integration schemes for this constitutive model. In fact, it is quite difficult to find numerical implementations of any unified viscoplastic model. For that reason, the authors present in this article two simple integration algorithms for the Arruda-Boyce viscoplastic model, one them explicit and the other one implicit.
The development of this algorithms required a thorough bibliographical review, searching and collecting the most convenient numerical strategies, revision of standard numerical practices (and eventually their avoidance) and of course the inclusion of the own ideas of the authors. The final result is a meticulous combination of elements which were carefully assembled into two numerical material routines, which up to the present date, have worked satisfactorily with all the finite element analyses carried out by the authors.

1 INTRODUCTION

1.1 Alternatives for the numerical implementation of Unified Viscoplastic models

The Arruda-Boyce constitutive model does not have a yield function like classical elastoplastic and viscoplastic constitutive models. With this formulation, elastic and inelastic deformations occur for every loading stage; therefore, the Arruda-Boyce constitutive model can be classified as a "unified viscoplastic model". Probably, the best known unified viscoplastic model is the Bodner-Partom model (Bodner and Partom 1975). This kind of models attempt to remove distinction between plasticity, creep, relaxation and viscous effect. Unified viscoplasticity should not be confused with "overstress viscoplasticity" (Perzyna's viscoplasticity). In Addition, the Arruda-Boyce constitutive model uses what some authors call a "Total-Lagrangian-Hencky" formulation or just TLH formulation (see Bathe 1996). In plain terms a TLH formulation means the following: there is a rate of plastic deformation but there is NO stress rate, the kinematic formulation is based on the multiplicative decomposition of the elastic and inelastic deformation gradients and the total stress is computed from the elastic "Hencky" strain measure (the logarithmic strain measure).

Unlike classical rate independent plasticity, there are not many works undertaking the numerical implementation of unified viscoplastic models as a central topic. An intense bibliographical revision has been carried out by the authors in the search of "similar" constitutive models (most of them for metals) and their numerical implementations. The result of this search brought up the different alternatives for their numerical implementation and the concern about additional numerical problems that might be found in this type of constitutive models. There are five basic possibilities for the numerical implementation of unified viscoplastic models:

1. **Explicit Runge-Kutta schemes in combination with automatic time-stepping algorithms.** These algorithms are very accurate, however, they must include a non-negligible "control" computational cost to work properly. Specifically, the numerical results obtained with these integration schemes is usually compared with an even more accurate solution, which is usually computed using "step doubling". See for instance Zirky 1994.
2. **Explicit Embedded Runge-Kutta schemes in combination with automatic time-stepping algorithms.** This kind of integration schemes include two solutions of different accuracy order within their mathematical structure, consequently, control and qualification of the numerical solution obtained with these integration schemes has a very low computational cost. See for instance Arya 1996 and Fritzen and Wittekindt 1997.
3. **Generalized Backward Euler Methods.** In these kind of schemes, the continuous evolution problem is transformed into a set of n non-linear scalar equations with n unknowns which are solved using a Newton-Raphson procedure. The number of unknowns is usually $n=18$ or $n=12$, occasionally, it can be reduced to $n=6$. Application of these schemes to unified viscoplasticity can be found on Saleeb 2000 and Tang 2007. Only a pithy coverage of these schemes for classical elasto-plasticity can be found on the section "General return mapping algorithms" of Simo and Hughes 1998.
4. **Reduced Backward Euler methods.** This kind of algorithms include an important number of assumptions and simplifications, some of this simplifications have their roots in geometric considerations, and feature the

reduction of the integration scheme to the solution of 1 (at most 2) nonlinear scalar equations. This kind of integration schemes are the traditional approach used in classical rate-independent elastoplasticity.
Application of these schemes to unified viscoplasticity can be found in Weber and Anand 1990, Lush et. al. 1989 and Sansour 1997.

5. **High order Backward differentiation formulas (BDF).** Backward differentiation formulas were specifically created to deal with "stiff" initial value problems. This kind of algorithms are not self starting and use information from previous time increments. Application of these schemes to unified viscoplasticity can be found in Kirchner and Simeon 1999 and Bergström 2006.
The Generalized Backward Euler method is a Backward differentiation formula of order 1 (known as BDF1). It is a particular case which is self-starting and does not use information from previous increments.

Without claiming completeness nor top relevance of the previous record of articles, the reader can appreciate that this partial list of citations is quite small. This fact reflects the actual scarcity of scientific literature covering this particular topic. It is important to mention that only Weber and Anand 1990 and Sansour 1997 report numerical procedures specifically created for models with TLH formulations. It also noteworthy that; with the exception of Tang 2007 and Bergström 2006; none of this works is related to polymeric materials.

1.2 Contents, aims, motivations and main focus of this article

Two simple numerical implementations for the Arruda-Boyce constitutive model are presented in this article, one of them explicit and the other one implicit:

- A second-order accurate embedded Runge-Kutta integration scheme, which works in combination with an automatic time-stepping algorithm. This integration algorithm belongs to the type 2 of the previous list.
- A Reduced Backward Euler scheme, featuring the reduction of the integration procedure to the solution of a system of 2 non-linear scalar equations. This algorithm belongs to the type 4 of the previous list classifying integration algorithms. In practice, this numerical implementation was not coded as a "fixed backward Euler method" but rather as a "generalized midpoint rule" allowing to use a variable θ, obtaining: a fully implicit integration scheme (Backward Euler method) for $\theta = 1$, midpoint rule for $\theta = 1/2$ and any intermediate unconditionally stable scheme for $0.5 < \theta < 1$. This algorithm is mostly inspired in the ideas found in Eterovic and Bathe 1990 and Weber and Anand 1990.

The main purpose of this article is to just present some "numerical procedures", which in the "subjective" experience of the authors have worked very well. The procedures are not analyzed in the "strict framework of numerical analysis", consequently, the presentation is quite technical. However, the authors attempted to provide the reasoning and the theoretical details underlying the development of these numerical procedures in order to give the reader insights into the potentialities and limitations of the ideas proposed in this article.

To be consistent with the idea of a "technical exposition", in all the cases, the authors tried to report with maximum simplicity and detail the recommended tolerances for residuals, criteria for the size the perturbation, choices of the order and type of approximants, initial guesses for iterative procedures, safeguard precautions and other important information of major pragmatic relevance.

It can be inferred that this article is twofold: it can be regarded collection of ideas and numerical practices or as two numerical implementations ready to be duplicated and used.

Computation of square roots, logarithms and exponentials of second order tensors constitute a non-negligible computational cost in the numerical implementation of constitutive models with TLH formulations. In order to reduce the computational cost of these numerical computations (which are pervasive in the numerical implementation of this kind of constitutive models) an appropriate methodology is presented in section 2.2. Some considerations were taken separately for the explicit and implicit integration schemes.

Both integration schemes where combined with a numerically obtained algorithmic tangent stiffness matrix. This strategy, provides great flexibility, since modifications of the integration algorithm or constitutive model only require minor modifications (or no one at all) of the subroutine that computes the tangent stiffness matrix. The details and practical considerations about the technique used to perturbate the deformation gradient are given in detail.

The main reason for building an explicit integration schemes is that they are very appealing in the context of research and development of constitutive models. Usually, the synthesis of a new constitutive model is finally accomplished after thorough numerical implementation and evaluation of many trial ideas. In this process in which trial ideas are constantly evaluated (and consequently accepted or discarded) it may not be necessary (or it may be impractical) to build a fully implicit integration scheme. In addition, explicit integration schemes (for the material routine) are quite well suited for Explicit (dynamic) finite element codes. With this purposes in mind, an efficient explicit integration scheme is presented in this article.

The major disadvantage of most explicit integration schemes is that they usually turn to be quite inefficient in "Standard" finite element codes. For instance, according to Arya 1996 and Saleeb 2000, between 10,000 and 100,000 increments are required to simulate mildly sophisticated uniaxial tests such as creep, cyclic loading or half-cycle loading (loading and unloading) when explicit integration algorithms are used for unified viscoplastic models.

As a starting point, the authors attempted the application of the time-stepping algorithms in the style of those proposed in Arya 1996, Zirky 1994, Fritzen and Wittekindt 1997 or Press 1992; which in essence, they are all equivalent since they are all based on the same idea. Application of those time-stepping algorithms to the Arruda-Boyce constitutive model did not produce satisfactory results. One of the reasons why the authors qualify those results as unsatisfactory is that too many increments were required to simulate simple uniaxial tests. However, the low efficiency of those time-stepping algorithms is just anecdotic; the most important point to be reported is that only a few finite element simulations could be completed; most of them were interrupted by the global finite element solver as a consequence of poor numerical stability or the requirement of deliberately small time increments. Consequently, the authors rejected those time-stepping algorithms and proceeded to define their own. As a result, the authors found that slight modification in the time-stepping algorithms proposed in those works improves significantly the computational efficiency of explicit integration schemes and still provide more stable solutions. The details of the time-stepping algorithm proposed in this article can be found on section 3.3.

On the other hand, a Reduced Backward Euler scheme is presented in this article. The underlying assumption of this kind of integration schemes is that the whole time-evolution problem can be reduced to the determination of a few relevant "invariants"

capable to capture the essential nature of the evolving system. In this kind of integration schemes it is still necessary to solve a system of non-linear equations, however, it is obvious that solving a system of $n=2$ is much cheaper than solving a system of $n=12$ or $n=18$ as its necessary with Generalized Backward Euler Methods, see Saleeb 2000.

The Reduced Backward Euler scheme presented in this article is supposed to be computationally cheaper than a Generalized Back Euler Method and yet unconditionally stable. However, in some publications (see for instance Arya 1996 or Weber and Anand 1990) it is suggested that unified viscoplastic constitutive models lead to a stiff system of ordinary differential equations governing the evolution of the inelastic deformations. Consequently, as the Generalized Back Euler (BDF1) can deal successfully with "stiff" initial value problems, it may be regarded as safer choice than a reduced Backward Euler method. Anyway, it is unclear if its really necessary (or imperative) to build a Generalized Backward Euler numerical implementation for the Arruda-Boyce constitutive model. The authors made no attempt to determine if there is stiff behavior (existence of significant disparity between the "principal" time rates of the initial value problem) in the Arruda-Boyce constitutive model using any theoretical device such as "conditioning numbers" or "stiffness ratios" as it has been done in Zirky 1994 for a different constitutive model .

Both numerical implementations (the explicit and the implicit one) were coded as UMAT subroutines for the finite element software ABAQUS-STANDARD.

1.3 Formulas of the modified Arruda-Boyce constitutive model and Notation

The modified Arruda-Boyce constitutive model

The total Cauchy Stress (stress of the elastic network) at the intermediate configuration is calculated using the elastic relationship:

$$\bar{T} = \frac{1}{J} \cdot \left(2\mu^e \cdot dev\left[\bar{E}^e\right] + k^e \cdot \frac{1}{3} \cdot tr\left[E^e\right] \cdot \mathbf{1} \right) \tag{1}$$

Where

$$\bar{E}^e = ln\left(U^e\right) \tag{2}$$

$$J = det\left(F\right) = det\left(F^e\right) \tag{3}$$

The total Kirchhoff stress at the intermediate configuration is just:

$$\bar{\sigma} = J \cdot \bar{T} \tag{4}$$

The stress in the Hyperelastic element is computed as follows:

$$\bar{\sigma}^B = \bar{\sigma}^B\left(\mu^L, \lambda_{lock}^L, \bar{E}_G^i\right) = \mu^p \cdot f\left(\bar{\lambda}^i, \lambda_{lock}^L\right) \cdot dev\left[B^i\right] \tag{5}$$

Where

$$B^i = F^i \cdot F^{iT} \tag{6}$$

$$f\left(\bar{\lambda}^i, \lambda_{lock}^L\right) = \frac{1}{\bar{\lambda}^i} \cdot \frac{L^{-1}\left(\dfrac{\bar{\lambda}^i}{\lambda_{lock}^L}\right)}{L^{-1}\left(\dfrac{1}{\lambda_{lock}^L}\right)} \tag{7}$$

$$\bar{\lambda}^i = \sqrt{\frac{tr\left(F^i \cdot F^{iT}\right)}{3}} \tag{8}$$

And the inverse Langevín function $L^{-1}(\bullet)$ is approximated as follows:

$$L^{-1}(x) \cong x\left(\frac{A + Bx^2}{1 + Cx^2}\right) \quad \text{with} \quad \begin{cases} A = 2.99248834685337 \\ B = \text{-}1.14365108190676 \\ C = -1 \end{cases} \tag{9}$$

The continuity, differentiability, exact (analytical) "integrability", accuracy and the fact that it's not a function by parts, make this approximation very appealing in a computational setting.

The rate of inelastic deformations at the intermediate configuration is defined as follows:

$$\bar{d}^i = \bar{d}^i\left(\bar{\sigma}^{vp}\right) = \dot{\gamma}^i \cdot \bar{N} \tag{10}$$

Where the stress driving the viscoplastic flow is the following (the notation implies that $\bar{\sigma}^{vp}$ is a deviatoric stress tensor):

$$\bar{\sigma}^{vp} = dev(\bar{\sigma}) - \bar{\sigma}^B = 2\mu^e \cdot dev\left[\bar{E}^e\right] - \bar{\sigma}^B \tag{11}$$

The magnitude of the rate of viscoplastic deformation is defined by:

$$\dot{\gamma}^i = \dot{\gamma}_0 \cdot exp\left(\frac{\tau}{\tau_{base}}\right) \tag{12}$$

$$\tau = \left\|\bar{\sigma}^{vp}\right\|_F \tag{13}$$

The direction in which the inelastic flow occurs is given by the normalized tensor $\bar{N}$, which is basically the direction of the deviatoric stress tensor $\bar{\sigma}^{vp}$:

$$\bar{N} = \frac{\bar{\sigma}^{vp}}{\left\|\bar{\sigma}^{vp}\right\|_F} = \frac{\bar{\sigma}^{vp}}{\tau} \tag{14}$$

2 GENERAL CONSIDERATIONS FOR BOTH KINDS OF ALGORITHMS

2.1 The stress update task in non-linear solid mechanics

The "incremental problem" can be summarized in terms of "given data" and "unknowns" to be computed as follows (see Bathe 1996):

> Given for the integration point χ: the stress ${}^tT$ at the beginning of the time increment, the initial deformation gradients $\left\{ {}_0^tF, {}_0^tF^e, {}_0^tF^i \right\}$, the value of all the internal state variables at the beginning of the time increment ${}^t\xi = \left({}^t\xi_1, {}^t\xi_2, ..., {}^t\xi_n\right)$ and the total deformation gradient at the end of the time

increment ${}^{t+\Delta t}_{0}F$; calculate (update) at time $t+\Delta t$ the triad $\left\{ {}^{t+\Delta t}T,\ {}^{t+\Delta t}_{0}F^{e},\ {}^{t+\Delta t}_{0}F^{i} \right\}$, all the internal state variables ${}^{t+\Delta t}\xi = \left({}^{t+\Delta t}\xi_1,\ {}^{t+\Delta t}\xi_2,...,\ {}^{t+\Delta t}\xi_n \right)$ and provide the continuum tangent stiffness matrix evaluated at the end of the increment; expression (15); or if it is possible, the algorithmic tangent stiffness matrix ; expression (16)).

$$\left. \frac{\partial \sigma}{\partial \varepsilon} \right|_{t+\Delta t} \tag{15}$$

$$\left. \frac{\partial (\Delta \sigma (\Delta \varepsilon))}{\partial \Delta \varepsilon} \right|_{t+\Delta t} \tag{16}$$

2.2 Natural logarithms, Exponentials and square roots of second order tensors

As stated and analyzed in Ortiz 2001, spectral decompositions of the type $\mathrm{A} = \Omega \cdot \Lambda_{\mathrm{A}} \cdot \Omega^{\mathrm{T}}$, where A, Ω and $\Lambda_{\mathrm{A}} \in \mathbb{R}^{3\times 3}$, constitute a non-negligible computational cost in the numerical implementation of TLH formulations. In consequence, the strategy chosen in this work is total avoidance of spectral decompositions. The main reason is that in most situations neither Ω nor Λ_{A} are really needed. For instance, to compute the elastic strain tensor $\overline{E}$ we actually do not use or need the triad $\left\{ U^{e}, R^{e}, \Lambda^{e} \right\}$. Our whole stress update procedures (the explicit and the implicit) do not need or use these tensorial entities at all.

The strategy chosen by the authors is massive use of high order Padé approximants, which provided a highly accurate and yet computationally efficient approach. The outstanding capabilities of the Padé approximants for this specific task seem to have been appreciated by first time in Weber and Anand 1990, however, they still remain quite unnoticed in computational inelasticity. Padé approximants have much greater accuracy and larger convergence ratio than Taylor expansions of the same accuracy order. Additional theoretical details of this technique can be found in Bender and Orszag 1978.

Square roots and natural logarithms of second order tensors

Throughout the stress update process there are two operations in which we will use square roots and logarithms, these are the following ones:

$$U = \sqrt{\left(F^{e}\right)^{T} \cdot F^{e}} = \sqrt{C^{e}} \tag{17}$$

$$\overline{E} = ln(U) \tag{18}$$

The first simplification is to fuse the operations of expressions (17) and (18) into one operator $f(\bullet) = ln\left(\sqrt{\bullet}\right)$. This last operator will be approximated with a high order Padé approximant expanded in the neighborhood of the identity matrix. Low order approximants such as $P_1^1\left(ln\sqrt{x}\right)$ are enough for stiff materials such as steel. In fact, a standard approach in metal-forming is to use the first order accurate $P_0^1\left(ln\sqrt{x}\right)$ approximant, which leads to the Cauchy-Green strain measure. However, in compliant materials (such as polymers) at large strains, principal elastic stretches can depart significantly from the unity. For instance, UHMWPE during uniaxial tests can reach a

principal elastic stretch in the order of $\lambda = 1.14 - 1.4$ at failure stress. For that reason, it is convenient to use higher order approximants such as P_2^1 or P_2^2.

$$P_2^1\left(ln\sqrt{x}\right) = 6\cdot(\mathbf{1}-x)\cdot\left(x^2-8x-5\cdot\mathbf{1}\right)^{-1} \tag{19}$$

$$P_2^2\left(ln\sqrt{x}\right) = 3\cdot(x+\mathbf{1})\cdot(x-\mathbf{1})\cdot\left(2\cdot\left(x^2+4x+\mathbf{1}\right)\right)^{-1} \tag{20}$$

Where $x \in \mathbb{R}^{3\times3}$ and $\mathbf{1} \in \mathbb{R}^{3\times3}$ is the second order identity matrix.

Since the operator $f(\bullet) = ln\left(\sqrt{\bullet}\right)$ will always be used to map a "total" tensor; not an "incremental" one; our choice for both kinds of integration schemes (the explicit and the implicit) is the fourth order accurate $P_2^2\left(ln\sqrt{x}\right)$ Padé approximant.

The accuracy of Padé approximants is astonishing. Generally speaking, when the $P_2^2\left(ln\sqrt{x}\right)$ is applied to second order tensors $\mathrm{A} = \Omega\cdot\Lambda_{\mathrm{A}}\cdot\Omega^{\mathrm{T}}$, where Ω is not necessarily the identity matrix, and the entries in the diagonal of Λ_{A} are in the range $0.6 < \lambda < 1.4$, the premature observation is that the relative error of Padé approximant; expression (21); will be in the range 10^{-3}-10^{-5} while the relative error of the "supposedly equivalent" fourth-order Taylor polynomial will always be 10-100 times larger. Padé approximants constitute a very interesting replacement for the inexact Taylor series expansions and the accurate; but computationally expensive; subroutines for eigen computations.

$$Error = \frac{\left\| P_2^2\left(ln\sqrt{A}\right) - ln\sqrt{A}\right\|_F}{\left\|ln\sqrt{A}\right\|_F} \tag{21}$$

Exponential mappings of second order tensors

Throughout both integration algorithms, the exponential mapping is only used in the update of the inelastic deformation gradient, which is detailed as follows:

$${}^{t+\theta\Delta t}_{0}F^i = exp\left({}^{t+\theta\Delta t}\overline{d}^i\cdot\Delta t\right)\cdot{}^{t}_{0}F^i \tag{22}$$

The exponential mapping of expression (22) can be approximated with the following Padé approximants expanded in the neighborhood of the null tensor

$$P_1^1\left(exp(x)\right) = (x+2\cdot\mathbf{1})\cdot(2\cdot\mathbf{1}-x)^{-1} \tag{23}$$

$$P_2^1\left(exp(x)\right) = 2\cdot(x+3\cdot\mathbf{1})\cdot\left(x^2-4x+6\cdot\mathbf{1}\right)^{-1} \tag{24}$$

$$P_2^2\left(exp(x)\right) = \left(x^2+6x+12\cdot\mathbf{1}\right)\cdot\left(x^2-6x+12\cdot\mathbf{1}\right)^{-1} \tag{25}$$

The error term of these approximations is of the form $O_{n+1}\left(\|x\|\right)^{n+1}$, where n is the accuracy order of the approximant. In our case we have that $\|x\|$ takes the following form:

$$\|x\| = \left\|{}^{t+\theta\Delta t}\overline{d}^i\cdot\theta\Delta t\right\| = {}^{t+\theta\Delta t}\gamma\cdot\theta\Delta t = \Delta\gamma_\theta \tag{26}$$

Whereas $\Delta\gamma_\theta$ is supposed to remain small (close to the null tensor) in explicit integration algorithms (because of the natural limitations in the time-increment size of

explicit algorithms); $\Delta\gamma_\theta$ may take quite large values in implicit algorithms.

For that reason, our choices are the following ones: $P_1^1\left(exp(x)\right)$ for the explicit algorithm and $P_2^2\left(exp(x)\right)$ for the implicit one. The mean guideline for this decision is the satisfaction of the incompressibility constraint on the inelastic deformations.

2.3 Computation of the tangent stiffness matrix

The computation of the tangent stiffness matrix is the whole step 18 of the explicit integration algorithm (see section 3.2) and the whole step 27 of the implicit integration algorithm (see section 4.6)

Obtaining a closed-form (exact) continuum or algorithmic tangent stiffness matrix for the Arruda-Boyce constitutive model is not a straightforward task. The strategy followed in this work is the numerical computation of an approximated algorithmic tangent stiffness matrix. This strategy seems to have been proposed by first time in Kojic and Bathe 1987.

For "Hypoelastic type" constitutive formulations, the whole stress update algorithm can be thought as a function $\Delta\sigma = \Delta\sigma(\Delta\varepsilon)$ which has $\Delta\varepsilon$ as the only input argument. The algorithmic tangent stiffness matrix is a "true derivative" of the function $\Delta\sigma(\Delta\varepsilon)$; in other words, the left hand term of expression (27). On the other hand, the perturbation strategy is truly simple: the analytical derivative is replaced by a numerical approximation, which is based on the elementary definition of differential calculus:

$$\frac{\partial\Delta\sigma}{\partial\Delta\varepsilon} \cong \frac{\delta\Delta\sigma}{\delta\Delta\varepsilon} = \frac{\Delta\sigma(\Delta\varepsilon+\delta\varepsilon)-\Delta\sigma(\Delta\varepsilon)}{\delta} \tag{27}$$

Where

- $\Delta\sigma(\Delta\varepsilon)$ is the "stress increment function", which actually depends on the integration algorithm and the stress-strain constitutive relationship of the constitutive model.
- $\Delta\varepsilon$ is the strain increment tensor, which has been applied to the material point χ (integration point in a numerical setting) during the time increment from t to $t+\Delta t$.
- $\delta\varepsilon$ is an "almost null" tensor, which only contains two small perturbations of size δ in the components $\delta\varepsilon_{ij}$ and $\delta\varepsilon_{ji}$, all the remaining entries of $\delta\varepsilon$ are zero.
- $\widetilde{\Delta\varepsilon} = \Delta\varepsilon + \delta\varepsilon$ is a perturbated strain increment tensor

Since the constitutive formulation of the Arruda-Boyce model does not use an "Hypoelastic type" formulation but a rather a "Total Lagrangian" formulation, we don't have a "stress increment function" $\Delta\sigma(\Delta\varepsilon)$, in addition, $\Delta\varepsilon$ is not a direct input of our integration algorithm. In total formulations we have a "total stress function" ${}^{t+\Delta t}\sigma = \sigma\left({}^{t}_{0}F^{e}, {}^{t}_{0}F^{i}, {}^{t+\Delta t}_{0}F\right)$ instead of a "stress increment function". Consequently, expression (27) takes the following form in our case:

$$\frac{\partial\Delta\sigma}{\partial\Delta\varepsilon} \cong \frac{\delta\Delta\sigma}{\delta\Delta\varepsilon} = \frac{{}^{t+\Delta t}\sigma\left({}^{t}_{0}F^{e}, {}^{t}_{0}F^{i}, {}^{t+\Delta t}_{0}\tilde{F}\right) - {}^{t+\Delta t}\sigma\left({}^{t}_{0}F^{e}, {}^{t}_{0}F^{i}, {}^{t+\Delta t}_{0}F\right)}{\delta} \tag{28}$$

The task involves the perturbation of the total deformation gradient. Conceptually, an infinitesimal perturbation of the stretched "fibers" in the spatial configuration can be thought as a pre-multiplication of the deformation gradient by a stretch tensor dU, see

expression (30). This tensor dU is supposed to be very close to the identity tensor, however, dU must include an "infinitesimal" perturbation of size δ in the components $(dU)_{ij}$ and $(dU)_{ji}$, see expression (29).

$$dU = \mathbf{1} + \delta\varepsilon \tag{29}$$

$${}^{t+\Delta t}_{0}\tilde{F} = dU \cdot {}^{t+\Delta t}_{0}F = (\mathbf{1} + \delta\varepsilon) \cdot {}^{t+\Delta t}_{0}F \tag{30}$$

In the context of three dimensional analysis, the deformation gradient ${}^{t+\Delta t}_{0}F$ must be perturbated 6 times. Then, each set $\left\{ {}^{t}_{0}F^{e}, {}^{t}_{0}F^{i}, {}^{t+\Delta t}_{0}\tilde{F} \right\}$ is used as input argument for the algorithm used to update the total stress. Subsequently, using the formula of expression (28) the six columns of the Jacobian matrix are generated.

This perturbation technique is ridiculously simple and yet it works outstandingly (see later Table 5 in section 5). Other more sophisticated perturbation techniques have been tested by the authors but no one provided as good results as this one in terms of computational efficiency.

The size of the perturbation δ must be small enough as to keep the quotient of expression (28) sufficiently close the value of the "true" derivative. A simple strategy to obtain an accurate approximation for the tangent stiffness matrix for every load (time) increment is to keep δ relatively small when compared to the Frobenius (Euclidean) norm of the strain increment tensor by setting a fixed value α.

$$\delta = \alpha \cdot \left\| \Delta\varepsilon \right\|_{F} \tag{31}$$

The authors do not encourage to use deliberately small values for α. First of all because it is not necessary. Second, it is important to remember that there are too many operations involved in the computation of the perturbated stress; though, it has not been observed by authors; there is always the risk that the effect of a very small α might be "dissipated" because of roundoff (the effect of finite computer accuracy).

From the numerical experience of the authors, an α as large as $\alpha = 10^{-3}$ can work quite well, however, smaller values can bring out better results. The particular choice of the authors has been $\alpha = 10^{-5}$ for all circumstances. Values for α smaller than 10^{-5} did not provide any additional benefit in terms of numbers of equilibrium iterations.

It is important to mention Miehe 1996, the only reference known by the authors, and perhaps the only one that exists, dealing with the numerical computation of algorithmic tangent stiffness matrices for constitutive models that use "Total Lagrangian" formulations. In that work a more sophisticated methodology is proposed, using a different perturbation technique which takes into account advanced kinematic considerations and a different criteria for the choice of the size of the perturbation. On the other hand, the methodology presented in this work is not as "correct" as that one of Miehe 1996, however, it presents the required essential characteristics: it is very robust, in fact, it has never been the source of failure or interruption of any finite element analysis; and provides a significant reduction in the number of equilibrium iterations when it is compared to the performance of a constant (elastic) tangent stiffness matrix, see Table 5 in section 5.

3 ACCURATE AND ROBUST EXPLICIT INTEGRATION SCHEME

3.1 Selection of the integration algorithms: Embedded Runge-Kutta schemes and "convergence" tests

It is the opinion of the authors (and perhaps that one of most practitioners) that using explicit integration algorithms without any kind of time-stepping algorithm, internal "control" or internal subroutine "qualifying" the obtained numerical solution can be a quite risky practice. The numerical solution obtained with any explicit integration algorithm must be controlled and qualified as "acceptable" or "unacceptable". The criteria of qualification is another important issue: we can use a "mild" or a "mathematically rigorous" criterion. Something important about controlling the quality of the obtained numerical solution is that "control" has a computational cost which must afforded but there are some choices which can make of control an absolutely effortless task.

In the process of selection of an appropriate integration scheme, the accuracy order is a topic of major importance. The large variety of explicit integration schemes (of any accuracy order) that exists certainly boggles the mind, see for instance Hairer 1993 and Butcher 2003. It could be possible to go directly to high order schemes such as the RK4s or RK5s, however, that can result into quite sophisticated algorithms and it is not clear if they are worth the effort. The main question is: which accuracy order constitutes the most sensible choice? In other words: which accuracy order provides a suitable balance between simplicity, efficiency and effortless control?

Following the advice of Arya 1996, only low and moderate order schemes will be used in this work. In Arya 1996 it is stated that the major limitation of these algorithms is stability rather than accuracy and that high-order schemes provide no relevant advantage. Consequently, no attempt has been carried out to implement the classical embedded Runge-Kutta schemes, such as: the Fehlberg-Runge-Kutta scheme and the Cash-Karp-Runge-Kutta scheme, both of which are fifth order accurate but require 6 functional evaluations (6 time rates to be averaged).

Some of the simplest low order Embedded Runge-Kutta schemes are the Explicit Trapezoidal Rule and the Explicit Midpoint Rule, both of which are second order accurate and both embed the first-order accurate Forward Euler (FE) approximation, see Table 1.

Trapezoidal Rule (TR)	Midpoint rule (MP)
$\begin{cases} K1 = \dot{y}(y,t) \\ K2 = \dot{y}(y+\Delta t \cdot K1 \ ; \ t+\Delta t) \end{cases}$	$\begin{cases} K1 = \dot{y}(y,t) \\ K2 = \dot{y}\left(y+\frac{\Delta t}{2}\cdot K1 ; t+\frac{\Delta t}{2}\right) \end{cases}$
$\begin{cases} {}^{t+\Delta t}y_{TR} = {}^{t}y + \frac{\Delta t}{2}(K1+K2) \\ {}^{t+\Delta t}y_{FE} = {}^{t}y + \Delta t \cdot K1 \end{cases}$	$\begin{cases} {}^{t+\Delta t}y_{MP} = {}^{t}y + \Delta t \cdot K2 \\ {}^{t+\Delta t}y_{FE} = {}^{t}y + \Delta t \cdot K1 \end{cases}$

Table 1. Midpoint and trapezoidal schemes, usually called second order Runge-Kutta.

However, the midpoint rule has an advantage over the Trapezoidal rule. That advantage is that the Midpoint Rule time rates ($K1$ and $K2$) are included into the RK3 formulas, see Table 2. Consequently, the RK3 method includes a first, a second and a third order accurate approximation without any additional computational effort. This fact is noted in Fritzen and Wittekindt 1997.

RK3
$\begin{cases} K1 = \dot{y}(y,t) \\ K2 = \dot{y}\left(y + \frac{\Delta t}{2}\cdot K1 ; t + \frac{\Delta t}{2}\right) \\ K3 = \dot{y}\left(y + \Delta t\cdot(2K2 - K1) ; t + \Delta t\right) \end{cases}$ $\begin{cases} {}^{t+\Delta t}y_{FE} = {}^{t}y + \Delta t\cdot K1 \\ {}^{t+\Delta t}y_{MP} = {}^{t}y + \Delta t\cdot K2 \\ {}^{t+\Delta t}y_{RK3} = {}^{t}y + \Delta t\cdot\left(\frac{1}{6}K1 + \frac{2}{3}K2 + \frac{1}{6}K3\right) \end{cases}$

Table 2. Third order Runge-Kutta scheme

This special characteristic of the RK3 method makes it even more attractive than traditional high-order embedded Runge-Kutta schemes (such as Fehlberg, Cash-Karp, etc), because it makes it suitable to develop some "rudimentary" (but quite effective) methodologies to monitor the accuracy and particularly the "convergence" of the obtained numerical solution. Specifically, the RK3 method is "like" having access to the three the terms of the third-order Taylor series expansion. Consequently, if the RK3 method is supposed to be converging, it sounds reasonable to think that the respective Taylor series expansion is convergent too. In real life, we will never have access to the infinite terms of the Taylor series expansion nor the analytic expression of its terms to evaluate the following inequality which determines absolute convergence:

$$\lim_{n\to\infty}\left|\frac{a_{n+1}}{a_n}\right| = \lim_{n\to\infty}\left|\frac{\frac{1}{(n+1)!}y^{(n+1)}\Delta t^{n+1}}{\frac{1}{n!}y^{(n)}\Delta t^{n+1}}\right| < 1 \tag{32}$$

but we have information of the third-order truncated series, consequently, for our pragmatic purposes, we should at least be able to verify the satisfaction of the following inequalities:

$$d\left({}^{t+\Delta t}y_{MP}, {}^{t+\Delta t}y_{RK3}\right) < d\left({}^{t+\Delta t}y_{MP}, {}^{t+\Delta t}y_{FE}\right) < d\left({}^{t+\Delta t}y_{FE}, {}^{t}y\right) \tag{33}$$

Where:

$$d\left({}^{t+\Delta t}y_{MP}, {}^{t+\Delta t}y_{RK3}\right) \cong |a_3| = \left|\frac{1}{3!}{}^{t}\dddot{y}\cdot\Delta t^3\right|$$

$$d\left({}^{t+\Delta t}y_{MP}, {}^{t+\Delta t}y_{FE}\right) \cong |a_2| = \left|\frac{1}{2!}{}^{t}\ddot{y}\cdot\Delta t^2\right|$$

$$d\left({}^{t+\Delta t}y_{FE}, {}^{t}y\right) \cong |a_1| = \left|\frac{1}{1!}{}^{t}\dot{y}\cdot\Delta t\right|$$

Where $d(\bullet,\bullet)$ denotes "distance" between the two input arguments. It is important to remark that any test derived from this inequalities is not conclusive, since in general,

"local converge" does not imply "stability" of the numerical solution. However, we can derive a sort of "mild" convergence tests, which can be computed with the investment of insignificant computational effort (for the RK2 and RK3 method) and may work reasonably well. On the other hand, conclusive stability tests rely in quite expensive eigen value computations (see for instance Heath 1997).

The global update process requires the tangent stiffness matrix, which in this kind of simple numerical implementations is computed numerically. The same algorithm that is used to update the stress should be used to compute the tangent stiffness matrix. Therefore; if a high-order integration scheme is used; the final number of computations grows significantly. This argument is a good reason to use low-order Runge-Kutta schemes such as the Midpoint rule.

Finally, it is important to mention that Embedded Runge-Kutta schemes of order n requiring only n functional evaluations (n time rates to be averaged) of order greater than 3 do not seem to exist (see Hairer 1993). For instance, the Embedded Runge-Kutta schemes RK4 requires 5 time rates Ki and the Embedded Runge-Kutta schemes RK5 require 6 time rates. Consequently, the Runge-Kutta scheme RK3 constitutes a reasonable "upper ceiling" for the accuracy order in terms of "additional benefits per functional evaluation" and "effortless control".

3.2 Explicit Midpoint Integration Scheme for the Arruda-Boyce constitutive model

1. The value of tensors $\left\{ {}_0^t F, {}_0^t F^e, {}_0^t F^i, {}_0^{t+\Delta t} F \right\}$ is available at the beginning of each time increment.
2. Compute $\overline{E}^e = ln\left({}_0^t U^e \right)$ in a single operation using the $P_2^2\left(ln\sqrt{x}\right)$ Padé approximant
$$ {}^t\overline{E}^e = ln\sqrt{\left({}_0^t F^e \right)^T \cdot {}_0^t F^e}$$
3. Compute ${}^t\overline{\sigma}^B$ with expressions (5) to (9) using ${}_0^t F^i$ as input argument for those expressions.
4. Compute $dev\left({}^t\overline{\sigma}^{vp} \right)$ using the following expression
$$ {}^t\overline{\sigma}^{vp} = 2\mu^e \cdot dev\left[{}^t\overline{E}^e \right] - {}^t\overline{\sigma}^B$$
For this purpose, use the value of ${}^t\overline{E}^e$ computed in the step 2 and the value of ${}^t\overline{\sigma}^B$ computed in the step 3.
5. Compute ${}^t\overline{d}^i$: use the tensor ${}^t\overline{\sigma}^{vp}$ calculated in the previous step as input argument for expressions (10) and (12) to (14).
Store ${}^t\overline{d}^i$, not only for direct use in the next step but also for later use in the error assessment and the time-stepping algorithm (both in step 13).
6. Compute ${}_0^{t+\Delta t/2} F^i$ as follows:
$$ {}_0^{t+\Delta t/2} F^i = exp\left({}^t\overline{d}^i \cdot \Delta t/2 \right) \cdot {}_0^t F^i$$
Use the Padé approximant $P_1^1\left(exp(x)\right)$ to compute the exponential mapping within this operation.
Note that in this operation the half time increment size ($\Delta t/2$) must be used.

7. Compute ${}^{t+\Delta t/2}_{0}F$ as an average of ${}^{t}_{0}F$ and ${}^{t+\Delta t}_{0}F$ (both tensors are given data)

$$ {}^{t+\Delta t/2}_{0}F = \frac{{}^{t}_{0}F + {}^{t+\Delta t}_{0}F}{2} $$

8. Compute ${}^{t+\Delta t/2}_{0}F^{e}$ as follows

$$ {}^{t+\Delta t/2}_{0}F^{e} = {}^{t+\Delta t/2}_{0}F \cdot \left({}^{t+\Delta t/2}_{0}F^{i} \right)^{-1} $$

9. With ${}^{t+\Delta t/2}_{0}F^{e}$ calculated in the previous step compute ${}^{t+\Delta t/2}\overline{E}^{e}$

$$ {}^{t+\Delta t/2}\overline{E}^{e} = ln\sqrt{\left({}^{t+\Delta t/2}_{0}F^{e} \right)^{T} \cdot {}^{t+\Delta t/2}_{0}F^{e}} $$

   Again, use the $P_2^2\left(ln\sqrt{x}\right)$ Padé approximant for this operation.

10. With ${}^{t+\Delta t/2}\overline{E}^{e}$ computed in the previous step and ${}^{t+\Delta t/2}_{0}F^{i}$ computed in step 6 calculate ${}^{t+\Delta t/2}\overline{\sigma}^{vp}$.
11. Compute ${}^{t+\Delta t/2}\overline{d}^{i}$: use the tensor ${}^{t+\Delta t/2}\overline{\sigma}^{vp}$ computed in the previous step as input argument of expressions (10) and (12) to (14).
12. Compute ${}^{t+\Delta t}_{0}F^{i}$ as follows

$$ {}^{t+\Delta t}_{0}F^{i} = exp\left({}^{t+\Delta t/2}\overline{d}^{i} \cdot \Delta t \right) \cdot {}^{t}_{0}F^{i} $$

   Use the Padé approximant $P_1^1\left(exp(x)\right)$ to compute the exponential mapping within this operation.
   Note that in this step the full time increment size (Δt) must be used.
13. Error assessment of ${}^{t+\Delta t}_{0}F^{i}$ and time-stepping algorithm: For details of the error assessment and time-stepping algorithm: see section 3.3.

If the inelastic deformation gradient ${}^{t+\Delta t}_{0}F^{i}$ computed in the last step qualifies successfully the constraint imposed in the error assessment routine, then, the algorithm continues with the following steps, otherwise the material routine halts at the end of this step.

14. Compute ${}^{t+\Delta t}_{0}F^{e}$ as follows

$$ {}^{t+\Delta t}_{0}F^{e} = {}^{t+\Delta t}_{0}F \cdot \left({}^{t+\Delta t}_{0}F^{i} \right)^{-1} $$

15. With ${}^{t+\Delta t}_{0}F^{e}$ calculated in the last step compute ${}^{t+\Delta t}E^{e}$

$$ {}^{t+\Delta t}E^{e} = ln\sqrt{{}^{t+\Delta t}_{0}F^{e} \cdot \left({}^{t+\Delta t}_{0}F^{e} \right)^{T}} $$

   Use the $P_2^2\left(ln\sqrt{x}\right)$ Padé approximant for this operation.

16. Compute the total Cauchy stress ${}^{t+\Delta t}T$ (spatial stress) using elastic relationship and the tensor ${}^{t+\Delta t}E^{e}$ calculated in the previous step.

$$ {}^{t+\Delta t}T = \frac{1}{J} \cdot \left(2\mu^{e} \cdot dev\left[{}^{t+\Delta t}E^{e} \right] + k^{e} \cdot \frac{1}{3} \cdot tr\left[{}^{t+\Delta t}E^{e} \right] \cdot \mathbf{1} \right) $$

17. Store ${}^{t+\Delta t}_{0}F^{e}$ and ${}^{t+\Delta t}_{0}F^{i}$ and report the total stress ${}^{t+\Delta t}T$.
18. Compute of the tangent stiffness matrix: see section 2.3

It should be noted that the input argument for the operator $f(\bullet)=ln\left(\sqrt{\bullet}\right)$ it's not always a ${}^{t+\theta\Delta t}_{0}C^{e}$ (a Green deformation tensor). In the step 15, the operator $f(\bullet)=ln\left(\sqrt{\bullet}\right)$ is applied to ${}^{t+\Delta t}_{0}B^{e}$, the elastic Finger deformation tensor, which is a tensor defined in the "spatial" configuration (see Dvorkin 2006). Consequently, no rotation tensor ${}^{t+\theta\Delta t}_{0}R^{e}$ is needed in the integration algorithm.

The RK3 algorithm can be implemented similarly, in other words: using tensorial entities from the "intermediate configuration" to compute the different time rates (K1, K2 and K3), and computing spatial entities (the elastic logarithmic strain and the Cauchy stress) only at the end of the integration process, hence, avoiding the computation of rotation tensors ${}^{t+\theta\Delta t}_{0}R^{e}$.

3.3 Error estimation and time-stepping algorithm

Analytical development of the time-stepping algorithm

Actually, the error estimation and time-stepping algorithm is the whole step 13 of the integration algorithm in the previous section (section 3.2.). However, this time-stepping algorithm could also be applied to the implicit integration scheme (section 4.6) using $\theta=0.5$.

The error of a given numerical solution is the difference between this approximate solution and the exact one. As the exact solution is usually unknown, the standard approach to qualify the accuracy of the numerical solution of an initial value problem is to compare it with another numerical solution that has been obtained with an integration scheme of higher accuracy order.

For instance, to evaluate the quality of a first-order-accurate numerical solution for the inelastic deformation gradient ${}^{t+\Delta t}_{0}F^{i}$ we can compare it with the solution obtained with the explicit midpoint scheme as in expression (34).

$$E = {}^{t+\Delta t}_{0}F^{i}_{FE} - {}^{t+\Delta t}_{0}F^{i}_{MP} \tag{34}$$

Where

- ${}^{t+\Delta t}_{0}F^{i}_{FE}$ represents a numerical estimation for ${}^{t+\Delta t}_{0}F^{i}$ using the Forward Euler method (first order accurate)
- ${}^{t+\Delta t}_{0}F^{i}_{MP}$ represents a numerical estimation for ${}^{t+\Delta t}_{0}F^{i}$ using the explicit Midpoint rule (second order accurate)

The error estimation E actually applies to the first order accurate approximation. However, a standard approach implied by the embedding technique, is to keep (and report!) the high-order estimation. Consequently, it is assumed that E presents a reasonable upper bound for the error introduced by the midpoint rule approximation.

Afterward, it is imposed that the value of ε; see expression (35); which we will call "weighted error formula", should not be grater than k to keep the numerical solution accurate and stable:

$$\varepsilon = \frac{\left\| {}^{t+\Delta t}_{0}F^{i}_{FE} - {}^{t+\Delta t}_{0}F^{i}_{MP} \right\|_{E}}{\left\| {}^{t+\Delta t}_{0}F^{i}_{MP} - {}^{t}_{0}F^{i} \right\|_{E}} < k \tag{35}$$

Where

- ε is a "weighted or relative difference" between the first and second order numerical approximations.
- the denominator of expression (35) determines the "weight or importance" of this difference. It is noteworthy that the denominator proposed in expression (35) is the norm of $\Delta F^i = {}^{t+\Delta t}_{0}F^i_{MP} - {}^{t}_{0}F^i$ rather than the norm of the a total (accumulated) deformation gradient such as ${}^{t+\Delta t}_{0}F^i_{MP}$, ${}^{t+\Delta t}_{0}F^i_{FE}$ or ${}^{t}_{0}F^i$.
- $0 < k < 1$ is a value adjusted to obtain the desired quality in the numerical solution.

If $\varepsilon > k$, then, we must abort the current time increment, and restart it (make a new attempt) using a smaller time increment Δt_{new}. The problem is to decide how small Δt_{new} should be to satisfy the inequality $\varepsilon < k$ in the new attempt.

It is known that any Runge-Kutta integration scheme just represents a truncated Taylor series expansion, see expression (36).

$$ {}^{t+\Delta t}_{0}F^i = \frac{{}^{t}_{0}F^i}{0!} + \frac{{}^{t}_{0}\dot{F}^i}{1!}\cdot\Delta t + \frac{{}^{t}_{0}\ddot{F}^i}{2!}\cdot\Delta t^2 + \frac{{}^{t}_{0}\dddot{F}^i}{3!}\cdot\Delta t^3 + \frac{{}^{t}_{0}\ddddot{F}^i}{4!}\cdot\Delta t^4 + \ldots \tag{36} $$

In the case of the Forward Euler method, the numerical approximation is "equivalent" to take only the first two summands of the Taylor series, which yield an error with a dominant term $O_2\cdot\Delta t^2$; see expression (37). In the case of the Midpoint rule, the numerical approximation is equivalent to take first three summands, which yield an error with a leading term $O_3\cdot\Delta t^3$, see expression (37).

$$ {}^{t+\Delta t}_{0}F^i = \underbrace{\underbrace{\frac{{}^{t}_{0}F^i}{0!} + \frac{{}^{t}_{0}\dot{F}^i}{1!}\cdot\Delta t}_{\cong {}^{t+\Delta t}_{0}F^i_{FE}} + \frac{{}^{t}_{0}\ddot{F}^i}{2!}\cdot\Delta t^2}_{\cong {}^{t+\Delta t}_{0}F^i_{MP}} + \frac{{}^{t}_{0}\dddot{F}^i}{3!}\cdot\Delta t^3 + \frac{{}^{t}_{0}\ddddot{F}^i}{4!}\cdot\Delta t^4 + \ldots \tag{37} $$

Consequently, the difference E between the Midpoint and the Forward-Euler approximations takes the following approximate form

$$ E = {}^{t+\Delta t}_{0}F^i_{FE} - {}^{t+\Delta t}_{0}F^i_{MP} \cong O_2\cdot\Delta t^2 \tag{38} $$

And the difference between the Midpoint approximation and the initial solution; see the denominator of expression (35); takes the following form

$$ {}^{t+\Delta t}_{0}F^i_{MP} - {}^{t}_{0}F^i \cong \frac{{}^{t}_{0}\dot{F}^i}{1!}\cdot\Delta t + \frac{{}^{t}_{0}\ddot{F}^i}{2!}\cdot\Delta t^2 \tag{39} $$

However, as the leading term of this last expression is of order $O_1\cdot\Delta t$ and considering that $O_1\cdot\Delta t >> O_2\cdot\Delta t^2$ we discard the second-order terms of expression (39) to obtain the first-order approximation of expression (40).

$$ {}^{t+\Delta t}_{0}F^i_{MP} - {}^{t}_{0}F^i \cong O_1\cdot\Delta t \tag{40} $$

Note that the assumption $O_1\cdot\Delta t >> O_2\cdot\Delta t^2$, although it is not a sufficient condition, it somehow "insinuates" the idea that the numerical solution is convergent.

Replacing with the approximations of expressions (38) and (40) into the error weighting formula of expression (35) we get:

$$\varepsilon \cong \frac{\left\| O_2 \cdot \Delta t^2 \right\|_E}{\left\| O_1 \cdot \Delta t \right\|_E} = a \cdot \Delta t \tag{41}$$

If the inequality $\varepsilon < k$ is not satisfied, then, known ε; which is computed with expression (35); and the current time increment size Δt we can compute an upper bound for Δt_{new} capable to satisfy the inequality $\varepsilon < k$ in the next attempt. See the logical procedure in expression (42).

$$a = \frac{\varepsilon}{\Delta t} \quad \Rightarrow \quad \varepsilon_{new} = a \cdot \Delta t_{new} < k \quad \Rightarrow \quad \Delta t_{new} < \Delta t_{current} \frac{k}{\varepsilon} \tag{42}$$

Generally speaking, this kind of algorithms based on the estimation of truncation errors work very well. However, preliminary tests confirmed that under some particular conditions can get trapped. The problems that occurs is that the algorithm starts to suggest a $\Delta t_{new} \cong \Delta t$ repeatedly, as a consequence, futile attempts unable to satisfy error constraint are carried out over and over again. To avoid this problem, it is important to guarantee that the suggested time increment Δt_{new} is "consistently smaller" than the current one Δt; following the advice of Fritzen and Wittekindt 1997; two alternatives for Δt_{new} should be proposed, then, the smallest of those two tentative time increments should be chosen (for the specific details, see step 3.b of the time-stepping algorithm in this section).

The time-stepping algorithm

1. Compute the forward Euler approximation ${}^{t+\Delta t}_{0}F^i_{FE}$: for this purpose we use ${}^{t}\bar{d}^{\,i}$ (already computed in step 5 of the main material routine, see section 3.2.) and $P^1_1\left(exp(x)\right)$ to compute the exponential mapping.
$$ {}^{t+\Delta t}_{0}F^i_{FE} = exp\left({}^{t}\bar{d}^{\,i} \cdot \Delta t\right) \cdot {}^{t}_{0}F^i $$
2. Compute ε using the formula of expression (35). ${}^{t+\Delta t}_{0}F^i_{FE}$ is taken from the previous step and the midpoint estimation ${}^{t+\Delta t}_{0}F^i_{MP}$ is supposed to be already computed in step 12 of the main material routine.
3. If the condition $\varepsilon < k$ is satisfied, then, the main integration routine continues with steps 14 to 18 (see section 3.2), otherwise it continues computing the $\Delta t_{suggested}$ as follows:
   a. First we compute
   $$\Delta t_{new} = b \cdot \Delta t_{current} \frac{k}{\varepsilon}$$
   Where $0 < b \leq 1$ is a constant to guarantee that $\Delta t_{new} < \Delta t_{current} \frac{k}{\varepsilon}$.
   b. The final time increment size to be used in the next attempt is the following one:
   $$\Delta t_{suggested} = min\left[\Delta t_{new}; c \cdot \Delta t\right]$$

Where $min[\bullet;\bullet]$ is the smallest of the two arguments separated by the semicolon. This final step saves the algorithm of repeated iterations using the almost the same size for the time increment, leading to infinite attempts unable to satisfy error requirements and a fatal crash of the algorithm.

Values for b slightly smaller that 1 provide the best overall performance. In all circumstances, the authors used $b = 0.95$.

The main parameter limiting the maximum allowable size for the time increment is k. The larger it is the larger the time increments will be. It can take values as high as $k = 0.4$ or $k = 0.3$. The value $k = 0.5$ works at the beginning of most analysis but finally blows up. Although with $k = 0.4$ the algorithm is still capable to work; keeping the numerical solution in the "right track"; the obtained numerical results turn to be quite "rough". The authors do not recommend such large values for k. The authors used $k = 0.05 - 0.1$ in all the finite element analyses.

An appropriate value for c can be 0.7, 0.8 or 0.9 (see Fritzen and Wittekindt 1997). In all circumstances the authors used $c = 0.8$

Conceptual differences

The classical time-stepping algorithms use an error weighting formula (a "constraint imposed to the distance" between two numerical solutions) like that one in expression (43). In early stages, the authors attempted to use the classical time-stepping algorithms which rely on error weighting formulas similar to expression (43) and could not obtain satisfactory results.

$$\varepsilon = \frac{\left\| {}^{t+\Delta t}_{0}F^{i}_{FE} - {}^{t+\Delta t}_{0}F^{i}_{MP} \right\|_E}{\left\| {}^{t+\Delta t}_{0}F^{i}_{MP} \right\|_E} < k \tag{43}$$

This typical constraint used in classical time-stepping algorithms is quite restrictive at the beginning of the deformation process when $\left\| {}^{t+\Delta t}_{0}F^{i}_{MP} \right\|_E \cong \sqrt{3}$, however, when the deformation increases (imagine for instance an uniaxial traction test), that constraint is gradually relaxed since the denominator of expression (43) increases too. Consequently, if it's desired to obtain an accurate and stable numerical solution throughout all the analysis using the classical error weighting formula; such as that one in expression (43); it is necessary to use a very small value for k, which force to use deliberately small values for Δt at the beginning of the analysis but small-moderate sized time increments at large strains, which produce a relatively unstable numerical solutions.

On the other hand, the "distance constraint" between two numerical solutions presented in this section does not have much difference with those presented in Arya 1996, Zirky 1994 and Fritzen and Wittekindt 1997 or in classical books of numerical methods such as Press 1992 or in numerical analysis books highly specialized in ordinary differential equations such as Hairer 1993 and Butcher 2003. The only difference is the denominator of expression (35). The difference is slight, however, it reduces significantly the number of increments and still provides more stable solutions.

Generally speaking, expression (35) does not care about the total accumulated error. The condition imposed by expression (35) only cares about what is happening from

time t to $t+\Delta t$. We could say it is a "local constraint" between two numerical solutions ${}^{t+\Delta t}_{0}F^{i}_{FE}$ and ${}^{t+\Delta t}_{0}F^{i}_{MP}$, because it does not care about the size of the total accumulated tensors ${}^{t+\Delta t}_{0}F^{i}_{MP}$, ${}^{t+\Delta t}_{0}F^{i}_{MP}$, ${}^{t}_{0}F^{i}$ nor their associated errors. This local constraint imposes a "maximum admissible distance" between two numerical solution weighted over the magnitude of the local (from time t to $t+\Delta t$) deformation increment $\Delta F^{i} = {}^{t+\Delta t}_{0}F^{i}_{MP} - {}^{t}_{0}F^{i}$.

However, there is a more precise reason why this weighting error formula works better than that one expression (43). Basically, up to some point we can find that the analytical development relies in the assumption that $O_1 \cdot \Delta t >> O_2 \cdot \Delta t^2$. Under this assumption, we could arrive to the idea that ${}^{t+\Delta t}_{0}F^{i}_{FE} \cong {}^{t+\Delta t}_{0}F^{i}_{MP}$. Replacing with this last approximation into the denominator of expression (35) we get that the weighting formula becomes:

$$\varepsilon = \frac{\left\| {}^{t+\Delta t}_{0}F^{i}_{FE} - {}^{t+\Delta t}_{0}F^{i}_{MP} \right\|_E}{\left\| {}^{t+\Delta t}_{0}F^{i}_{FE} - {}^{t}_{0}F^{i} \right\|_E} < k \tag{44}$$

Looking closely at the this last formula we have that it expresses the same idea of expression (33). In other words, it enforces the satisfaction of the following inequality:

$$\left\| {}^{t+\Delta t}_{0}F^{i}_{FE} - {}^{t+\Delta t}_{0}F^{i}_{MP} \right\|_E < \left\| {}^{t+\Delta t}_{0}F^{i}_{FE} - {}^{t}_{0}F^{i} \right\|_E \tag{45}$$

The constraints imposed in expression (35) and expression (44) are quite equivalent; in fact, they can be used indistinctly producing very similar numerical results. These distance constraints do not work to enforce a certain level accuracy, these distance constraints "intend" to enforce "local convergence" of the Runge-Kutta method by using sufficiently small k.

4 ACCURATE AND EFFICIENT IMPLICIT SCHEME

4.1 Step 1: Application of the backward-Euler operator

The starting point of this scheme is the determination (computation) of ${}^{t+\theta\Delta t}_{0}F$ and the application of the backward Euler operator to elastic and inelastic deformation gradients.

First compute/define ${}^{t+\theta\Delta t}_{0}F$ as follows;

$${}^{t+\theta\Delta t}_{0}F = \theta \cdot {}^{t+\Delta t}_{0}F + (1-\theta)\, {}^{t}_{0}F \tag{46}$$

Where $0.5 \le \theta \le 1$.

Proceeding as if ${}^{t+\theta\Delta t}\overline{d}^{\,i}$ is already known; apply the backward Euler operator from time $t+\theta\Delta t$ to the inelastic and the elastic deformation gradients, see expressions (47) and (48).

$${}^{t+\Delta t}_{0}F^{i} = exp\left(\theta\Delta t \cdot {}^{t+\theta\Delta t}\overline{d}^{\,i}\right) \cdot {}^{t}_{0}F^{i} \tag{47}$$

$${}^{t+\theta\Delta t}_{0}F^{e} = \left({}^{t+\theta\Delta t}_{0}F\right) \cdot \left({}^{t}F^{i}\right)^{-1} \cdot exp\left(-\theta\Delta t \cdot {}^{t+\theta\Delta t}\overline{d}^{\,i}\right) \tag{48}$$

Where

$${}^{t+\theta\Delta t}\overline{d}^{\,i} = \gamma\left({}^{t+\theta\Delta t}\tau\right) \cdot {}^{t+\theta\Delta t}N \tag{49}$$

$$ {}^{t+\theta\Delta t}N = \frac{{}^{t+\theta\Delta t}\overline{\sigma}^{vp}}{\left\| {}^{t+\theta\Delta t}\overline{\sigma}^{vp} \right\|} = \frac{{}^{t+\theta\Delta t}\overline{\sigma}^{vp}}{{}^{t+\theta\Delta t}\tau} \tag{50} $$

It is noteworthy that expression (48) can be rewritten as follows

$$ {}^{t+\Delta t}_{0}F^{e} = F^{e}_{trial} \cdot exp\left(-\theta\Delta t \cdot {}^{t+\theta\Delta t}\overline{d}^{\,i}\right) \tag{51} $$

Where

$$ F^{e}_{trial} = \left({}^{t+\theta\Delta t}_{0}F\right) \cdot \left({}^{t}_{0}F^{i}\right)^{-1} \tag{52} $$

This last tensor can be computed directly from the given data at the beginning of each time increment. Additionally, as the inelastic deformations are supposed to occur without change of volume (${}^{t}J^{i} = 1 \;\; \forall t$), then, the elastic volumetric deformation ${}^{t+\theta\Delta}J^{e}$ at time $t+\theta\Delta t$ can be computed as follows

$$ {}^{t+\theta\Delta t}J^{e} = J^{e}_{trial} = det\left(F^{e}_{trial}\right) \tag{53} $$

Dividing both terms (left and right terms) of expression (51) by $\left({}^{t+\theta\Delta}J^{e}\right)^{1/3}$, the isochoric deformation gradient ${}^{t+\theta\Delta t}_{0}\mathbb{F}^{e}$ is obtained:

$$ {}^{t+\theta\Delta t}_{0}\mathbb{F}^{e} = \mathbb{F}^{e}_{trial} \cdot exp\left(-\theta\Delta t \cdot {}^{t+\theta\Delta t}\overline{d}^{\,i}\right) \tag{54} $$

$$ \mathbb{F}^{e}_{trial} = \frac{F^{e}_{trial}}{\left({}^{t+\theta\Delta t}J^{e}\right)^{1/3}} \tag{55} $$

The Arruda-Boyce constitutive model implemented in this work does not have any scalar internal state variable. However, this kind of models usually have a set of scalar state variables ${}^{t+\Delta t}\xi = \left({}^{t+\Delta t}\xi_1, {}^{t+\Delta t}\xi_2, ..., {}^{t+\Delta t}\xi_n\right)$. If that is the case, the backward Euler operator should be applied to these scalar state variables too.

$$ \begin{cases} {}^{t+\Delta t}\xi_1 = {}^{t}\xi_1 + \Delta t \cdot {}^{t+\theta\Delta t}\dot{\xi}_1 \\ {}^{t+\Delta t}\xi_2 = {}^{t}\xi_2 + \Delta t \cdot {}^{t+\theta\Delta t}\dot{\xi}_2 \\ \ldots \\ {}^{t+\Delta t}\xi_n = {}^{t}\xi_n + \Delta t \cdot {}^{t+\theta\Delta t}\dot{\xi}_n \end{cases} \tag{56} $$

4.2 Step 2: analytical expressions for the stress driving the viscoplastic flow at time $t+\theta\Delta t$

The mathematical formulas for the stress driving the viscoplastic flow ${}^{t+\theta\Delta t}\overline{\sigma}^{vp}$; expressions (11) and (5) to (8); is rewritten as follows:

$$ {}^{t+\theta\Delta t}\overline{\sigma}^{vp} = 2\mu^{e} \cdot dev\left[{}^{t+\theta\Delta t}\overline{E}^{e}\right] - \mu^{p} \cdot f\left({}^{t+\theta\Delta t}\overline{\lambda}^{i}, \lambda^{L}_{lock}\right) \cdot dev\left[{}^{t+\theta\Delta t}B^{i}\right] \tag{57} $$

Using the expression (47) the finger deformation tensor ${}^{t+\theta\Delta t}B^{i}$ is rewritten in terms of ${}^{t+\theta\Delta t}\overline{d}^{\,i}$:

$$ {}^{t+\theta\Delta t}B^{i} = exp\left(\theta\Delta t \cdot {}^{t+\theta\Delta t}\overline{d}^{\,i}\right) \cdot {}^{t}_{0}B^{i} \cdot exp\left(\theta\Delta t \cdot {}^{t+\theta\Delta t}\overline{d}^{\,i}\right) \tag{58} $$

Additionally, we know that deviatoric part of the Hencky strain tensor can be computed directly from the isochoric elastic deformation gradient as follows:

$$dev\left({}^{t+\theta\Delta t}\overline{E}^{e}\right) = ln\left(\sqrt{\mathbb{F}^{eT}\cdot\mathbb{F}^{e}}\right) \tag{59}$$

Replacing with expression (54) into this last expression we obtain:

$$dev\left({}^{t+\theta\Delta t}\overline{E}^{e}\right) = ln\left(\sqrt{exp\left(-\theta\Delta t\cdot{}^{t+\theta\Delta t}\overline{d}^{i}\right)\cdot\mathbb{C}^{e}_{trial}\cdot exp\left(-\theta\Delta t\cdot{}^{t+\theta\Delta t}\overline{d}^{i}\right)}\right) \tag{60}$$

$$\mathbb{C}^{e}_{trial} = \mathbb{F}^{e\ T}_{trial}\cdot\mathbb{F}^{e}_{trial} \tag{61}$$

It is noteworthy that in order to compute the hyperelastic stress at time $t+\theta\Delta t$; second summand of expression (57); we also need the effective stretch measure ${}^{t+\theta\Delta t}\overline{\lambda}^{i}$, which is computed from finger deformation tensor ${}^{t+\theta\Delta t}B^{i}$. The effective stretch measure ${}^{t+\theta\Delta t}\overline{\lambda}^{i}$ can be computed taking the trace of expression (58); see expression (62):

$$3\left({}^{t+\theta\Delta t}\overline{\lambda}^{i}\right)^{2} = trace\left[exp\left(\theta\Delta t\cdot{}^{t+\theta\Delta t}\overline{d}^{i}\right)\cdot{}^{t}_{0}B^{i}\cdot exp\left(\theta\Delta t\cdot{}^{t+\theta\Delta t}\overline{d}^{i}\right)\right] \tag{62}$$

4.3 Step 3: suitable approximations and linearizations

Approximations and linearizations to be included

Since expressions (58), (60) and (62) are quite intricate, it is necessary to add some simplifications and approximations to make the problem mathematically tractable.

The exponential mappings of expressions (58) to (62) will be replaced with a first order accurate approximation, see expression (63).

$$exp(x) = \mathbf{1} + x + O_2\left(\|x\|\right)^2 \tag{63}$$

On the other hand; as proposed in Eterovic and Bathe 1990; expression (60) can be replaced with the following first order accurate approximation:

$$dev\left({}^{t+\theta\Delta t}\overline{E}^{e}\right) = ln\left(\sqrt{{}^{t+\theta\Delta t}\mathbb{C}^{e}}\right) = dev\left(\overline{E}^{e}_{trial}\right) - \theta\Delta t\,{}^{t+\theta\Delta t}\overline{d}^{i} + O_2\cdot\left(\left\|\theta\Delta t\,{}^{t+\theta\Delta t}\overline{d}^{i}\right\|\right)^2 \tag{64}$$

This is a quite good approximation, however, comparatively, it can be verified that expression (64) generally includes greater errors than the approximation of expression (63).

It is noteworthy that expression (64) constitutes a forceful attempt to write the elastic deviatoric strain in a "rate-type form". This approximation is eventually used to write the total deviatoric stress in "rate-type form". It is recalled that in this type of constitutive formulations there is no stress rate. Expression (64) is just a mathematical artifice that makes the problem more tractable but up to some point corrupts the original nature of the formulation.

In addition, our implicit integration scheme will make use and "abuse" of the following classical approximation:

$${}^{t+\theta\Delta t}N \cong N_{trial} \tag{65}$$

This approximation is somehow a "prediction" of the behavior of the constitutive model at time $t+\theta\Delta t$ rather than a "simplification".

Summarizing, the total number of approximations to be included are three: expressions (63), (64) and (65). It is important to remark that any approximation or simplification has a price to be paid, which could be negligible or not. In other words, some of these approximations might introduce some limitation or restriction in the integration algorithm while others will not cause any detrimental effect at all. The effect

of these approximations is discussed in section 4.5.

Inclusion of the linearized expressions

Replacing the exponential mappings of expression (58) with its corresponding first order accurate approximation; expression (63); and ${}^{t+\theta\Delta t}\overline{d}^{\,i}$ with the right-hand term of expression (49), the following approximation for ${}^{t+\theta\Delta t}B^i$ is obtained:

$$\begin{aligned} {}^{t+\theta\Delta t}B^i \cong {}_0^tB^i + \gamma\left({}^{t+\theta\Delta t}\tau\right)\cdot\left[\Delta t_\theta \cdot 2\cdot sym\left({}_0^tB^i\cdot {}^{t+\theta\Delta t}N\right)\right]+ \\ +\left(\gamma\left({}^{t+\theta\Delta t}\tau\right)\right)^2\cdot\left[{\Delta t_\theta}^2\cdot\left({}^{t+\theta\Delta t}N\cdot {}_0^tB^i\cdot {}^{t+\theta\Delta t}N\right)\right] \end{aligned} \tag{66}$$

Where

$$\Delta t_\theta = \theta\cdot\Delta t \tag{67}$$

Then, incorporating the approximations (64) and (66) into the expression (57), and replacing ${}^{t+\theta\Delta t}\overline{d}^{\,i}$ with the right-hand term of expression (49), we finally obtain an approximation for the stress driving the viscoplastic flow ${}^{t+\theta\Delta t}\overline{\sigma}^{vp}$:

$$\begin{aligned} {}^{t+\theta\Delta t}\overline{\sigma}^{vp} \cong & \left[2\mu^e dev\left(\overline{E}^e_{trial}\right)\right]-\gamma\left({}^{t+\theta\Delta t}\tau\right)\cdot\left[\Delta t_\theta\cdot 2\mu^e\cdot {}^{t+\theta\Delta t}N\right] \\ & -\mu^L_{eff}\left({}^{t+\theta\Delta t}\overline{\lambda}^{\,i}\right)\cdot dev\left[{}_0^tB^i\right] \\ & -\mu^L_{eff}\left({}^{t+\theta\Delta t}\overline{\lambda}^{\,i}\right)\cdot\gamma\left({}^{t+\theta\Delta t}\tau\right)\cdot\left[\Delta t_\theta\cdot 2\cdot dev\left[sym\left({}_0^tB^i\cdot {}^{t+\theta\Delta t}N\right)\right]\right] \\ & -\mu^L_{eff}\left({}^{t+\theta\Delta t}\overline{\lambda}^{\,i}\right)\cdot\left(\gamma\left({}^{t+\theta\Delta t}\tau\right)\right)^2\cdot\left[{\Delta t_\theta}^2\cdot dev\left[{}^{t+\theta\Delta t}N\cdot {}_0^tB^i\cdot {}^{t+\theta\Delta t}N\right]\right] \end{aligned} \tag{68}$$

Where

$$\mu^L_{eff}\left({}^{t+\theta\Delta t}\overline{\lambda}^{\,i}\right) = \mu^p\cdot f\left({}^{t+\theta\Delta t}\overline{\lambda}^{\,i},\lambda^L_{lock}\right) \tag{69}$$

Reduction of the problem to 2 non-linear scalar equations

Since $tr\left[{}^{t+\theta\Delta t}B^i\right] = 3\left({}^{t+\theta\Delta t}\overline{\lambda}^{\,i}\right)^2$, applying the trace to expression (66) a non-linear scalar equation in terms of ${}^{t+\theta\Delta t}\overline{\lambda}^{\,i}$ and ${}^{t+\theta\Delta t}\tau$ is obtained:

$$\begin{aligned} & \left(\gamma\left({}^{t+\theta\Delta t}\tau\right)\right)^2\cdot\left({\Delta t_\theta}^2\cdot tr\left[{}^{t+\theta\Delta t}N\cdot {}_0^tB^i\cdot {}^{t+\theta\Delta t}N\right]\right)+ \\ & +\gamma\left({}^{t+\theta\Delta t}\tau\right)\cdot\left(2\cdot\Delta t_\theta\cdot tr\left[{}_0^tB^i\cdot {}^{t+\theta\Delta t}N\right]\right)+tr\left[{}_0^tB^i\right]-3\left({}^{t+\theta\Delta t}\overline{\lambda}^{\,i}\right)^2=0 \end{aligned} \tag{70}$$

Then, by taking the dot product of expression (68) with ${}^{t+\theta\Delta t}N$ another non-linear equation in terms of ${}^{t+\theta\Delta t}\overline{\lambda}^{\,i}$ and ${}^{t+\theta\Delta t}\tau$ is obtained:

$$\begin{aligned} & 2\mu^e\left[dev\left(\overline{E}^e_{trial}\right):{}^{t+\theta\Delta t}N\right]+\mu^L_{eff}\left({}^{t+\theta\Delta t}\overline{\lambda}^{\,i}\right)\cdot\left[-dev\left[{}_0^tB^i\right]:{}^{t+\theta\Delta t}N\right]+\gamma\left({}^{t+\theta\Delta t}\tau\right)\cdot\left[-2\mu^e\cdot\Delta t_\theta\right]+ \\ & +\mu^L_{eff}\left({}^{t+\theta\Delta t}\overline{\lambda}^{\,i}\right)\cdot\gamma\left({}^{t+\theta\Delta t}\tau\right)\cdot\left[-2\cdot\Delta t_\theta\cdot dev\left[sym\left({}_0^tB^i\cdot {}^{t+\theta\Delta t}N\right)\right]:{}^{t+\theta\Delta t}N\right]+ \\ & +\mu^L_{eff}\left({}^{t+\theta\Delta t}\overline{\lambda}^{\,i}\right)\cdot\left(\gamma\left({}^{t+\theta\Delta t}\tau\right)\right)^2\cdot\left[-{\Delta t_\theta}^2\cdot dev\left[{}^{t+\theta\Delta t}N\cdot {}_0^tB^i\cdot {}^{t+\theta\Delta t}N\right]:{}^{t+\theta\Delta t}N\right]-{}^{t+\theta\Delta t}\tau=0 \end{aligned} \tag{71}$$

Finally, we have that expressions (70) and (71) constitute a set of two non-linear equations with two unknowns: ${}^{t+\theta\Delta t}\overline{\lambda}^{\,i}$ and ${}^{t+\theta\Delta t}\tau$. However, it is noteworthy that ${}^{t+\theta\Delta t}N$ is unknown and without ${}^{t+\theta\Delta t}N$ we can't compute the coefficients of these non-linear

equations. To solve this problem we just replace ${}^{t+\theta\Delta t}N$ with N_{trial}. The consequences of this approximation in a numerical setting are discussed in section 4.5.

4.4 Numerical strategy to solve the system of equations

Solving the systems of equations defined by expressions (70) and (71) is not a trivial task. This system of equation has the following form:

$$\begin{cases} f_1(\tau,\lambda)=0 \\ f_2(\tau,\lambda)=0 \end{cases} \tag{72}$$

In which expressions (70) and (71) are represented by expressions $f_1(\tau,\lambda)=0$ and $f_2(\tau,\lambda)=0$ respectively. These 2 functions can be plotted in the $\tau\lambda-plane$. The "physically meaningful" solution to this system of equations is the intersection point defined by $\left({}^{t+\theta\Delta t}\bar{\lambda}^{i},\ {}^{t+\theta\Delta t}\tau\right)$. However, this system of equations does not have a unique intersection. The specific behavior of this system in the $\tau\lambda-plane$ depends on the functions $f\left(\bar{\lambda},\lambda_{lock}\right)$ and $\gamma(\tau)$, expressions (7) and (12) respectively for the Arruda-Boyce viscoplastic model. In our particular case, four intersection points exist between and only one of them has physical meaning. In Figure 1 the reader can appreciate the qualitative behavior of this system of equations. The equations plotted in Figure 1 were taken from an arbitrary integration point and loading conditions.

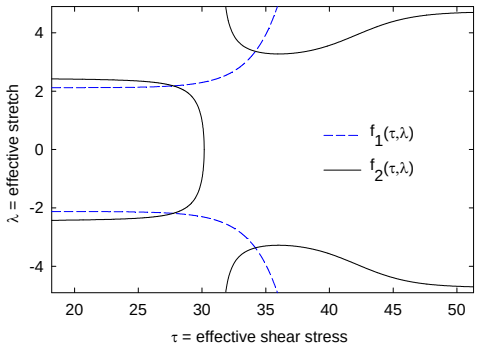

Figure 1. Graph of the functions $f_1(\tau,\lambda)$ and $f_2(\tau,\lambda)$ in the $\tau\lambda$-plane showing the 4 intersections.

Two possible strategies to solve this problem are the following ones:

1. Using a 2-variable Newton-Raphson procedure.
2. Using a 2-level Newton-Raphson procedure capable to "bracket" the correct solution. This strategy is suggested and used in Lush et. al. 1989.

The authors found that a hasty attempt to use the 2-variable Newton-Raphson procedure in combination with a "supposedly reasonable" initial guess works very well in a number of situations but also fails catastrophically in a few circumstances. Generally speaking, the 2-variable Newton-Raphson procedure works well, but some additional refinement must be added to make it more reliable.

On the other hand, the 2-level (2 stages) Newton-Raphson procedure splits the problem in two halves: it combines the "bracketing capabilities" of the bisection scheme and the efficiency of the Newton-Raphson procedure for scalar equations with only one unknown. The main objective of this numerical strategy is to guarantee than only the physically meaningful solution is obtained. For further details see Lush et. al. 1989.

The final methodology implemented in this work lies somehow between the two

methodologies previously described. The two stages of our procedure are the following ones:

1. Stage 1: the main goal of this stage is determining upper and lower bounds for ${}^{t+\theta\Delta t}\overline{\lambda}^{i}$ and ${}^{t+\theta\Delta t}\tau$, then, with this bounds we can determine a accurate initial guess $\left(\tau_0, \lambda_0\right)$.
2. Stage 2: using the accurate initial guess $\left(\tau_0, \lambda_0\right)$ developed in the previous stage, the system of two equations with two unknowns is solved using the 2-variable Newton-Raphson procedure.

Conceptually, this procedure is similar to that one used in Lush et. al. 1989 because it splits the problem in two halves and uses only one of the equations of the system in the first stage. However, computationally and mathematically it is quite different.

Most of the efforts were concentrated in developing good estimations for the upper and lower bounds of ${}^{t+\theta\Delta t}\overline{\lambda}^{i}$ and ${}^{t+\theta\Delta t}\tau$, which is the stage 1 of our procedure. In an ideal circumstance, the upper and lower bounds for ${}^{t+\theta\Delta t}\tau$ and ${}^{t+\theta\Delta t}\overline{\lambda}^{i}$ should define a small "window" (a rectangular subset) in the $\tau\lambda - plane$ (see Figure 2b) which only contains the physically meaningful solution (intersection). Then, taking any point of this small "rectangle" (for instance, the point in the geometric center of this rectangle) as an initial guess, the 2-variable Newton-Raphson procedure will converge to the closest intersection point (see step 17 of the implicit integration algorithm is section 4.6).

It is important to develop "narrow estimations" for ${}^{t+\theta\Delta t}\tau$ and ${}^{t+\theta\Delta t}\overline{\lambda}^{i}$, in other words, we should be able to verify that the "gap" defined by the upper and lower bounds is a small when compared to the average of them. Mathematically, we should be able to verify the following:

$$\frac{{}^{t+\theta\Delta t}\tau_{upper} - {}^{t+\theta\Delta t}\tau_{lower}}{\left(\dfrac{{}^{t+\theta\Delta t}\tau_{upper} + {}^{t+\theta\Delta t}\tau_{lower}}{2}\right)} << 1 \quad \text{and} \quad \frac{{}^{t+\theta\Delta t}\lambda_{upper} - {}^{t+\theta\Delta t}\lambda_{lower}}{\left(\dfrac{{}^{t+\theta\Delta t}\lambda_{upper} + {}^{t+\theta\Delta t}\lambda_{lower}}{2}\right)} << 1 \tag{73}$$

Finding upper and lower bounds for ${}^{t+\theta\Delta t}\overline{\lambda}^{i}$ satisfying this previous requirement is quite straightforward, the procedure relies in the computation of two trial states. However, computing upper and lower bounds for ${}^{t+\theta\Delta t}\tau$ is a bit more sophisticated.

Using two trial states to define upper and lower bounds for ${}^{t+\theta\Delta t}\overline{\lambda}^{i}$

Usually, "trial state" means that from time t to $t+\theta\Delta t$ only elastic deformations occur and inelastic deformations remain unchanged. We will call this trial state "trial state 1" or just "trial 1".

$${}^{t+\Delta t}_{0}F^{e}_{trial\ 1} = {}^{t+\theta\Delta t}_{0}F \cdot \left({}^{t}_{0}F^{i}\right)^{-1} \tag{74}$$

$${}^{t+\Delta t}_{0}F^{i}_{trial\ 1} = {}^{t}_{0}F^{i} \tag{75}$$

We can consider another trial state, meaning that from time t to $t+\theta\Delta t$ only inelastic deformations occur and elastic deformations remain unchanged. We will call this trial state "trial state 2".

$${}^{t+\Delta t}_{0}F^{e}_{trial\ 2} = {}^{t}_{0}F^{e} \tag{76}$$

$${}^{t+\Delta t}_{0}F^{i}_{trial\ 2} = \left({}^{t}_{0}F^{e}\right)^{-1} \cdot {}^{t+\theta\Delta t}_{0}F \tag{77}$$

Then using the trial inelastic deformation gradients ${}^{t+\Delta t}_{0}F^{i}_{trial\,1}$ and ${}^{t+\Delta t}_{0}F^{i}_{trial\,2}$ we can compute the corresponding scalars ${}^{t+\theta\Delta t}\overline{\lambda}^{i}_{trial\,1}$ and ${}^{t+\theta\Delta t}\overline{\lambda}^{i}_{trial\,2}$. In practice, in can be verified that the scalars ${}^{t+\theta\Delta t}\overline{\lambda}^{i}_{trial\,1}$ and ${}^{t+\theta\Delta t}\overline{\lambda}^{i}_{trial\,2}$ define a very "narrow range" for the effective stretch measure.

Consequently, the values ${}^{t+\theta\Delta t}\overline{\lambda}^{i}_{trial\,1}$ and ${}^{t+\theta\Delta t}\overline{\lambda}^{i}_{trial\,2}$ are used to define an upper and lower bound for ${}^{t+\theta\Delta t}\overline{\lambda}^{i}$. It is noteworthy that no assumption can be made about which one is the "upper bound" and which one is the "lower bound", that depends on the loading conditions. Therefore, ${}^{t+\theta\Delta t}\overline{\lambda}^{i}_{trial\,1}$ and ${}^{t+\theta\Delta t}\overline{\lambda}^{i}_{trial\,2}$ must be sorted according to their size, finally, the largest value is assigned to ${}^{t+\theta\Delta t}\lambda_{upper}$ and the smallest value is assigned to ${}^{t+\theta\Delta t}\lambda_{lower}$.

Computing upper an lower bounds for ${}^{t+\theta\Delta t}\tau$

We can continue using each pair $\left({}^{t+\Delta t}_{0}F^{e}_{trial\,1}, {}^{t+\Delta t}_{0}F^{i}_{trial\,1}\right)$ and $\left({}^{t+\Delta t}_{0}F^{e}_{trial\,2}, {}^{t+\Delta t}_{0}F^{i}_{trial\,2}\right)$ to compute the corresponding scalars ${}^{t+\theta\Delta t}\tau_{trial\,1}$ and ${}^{t+\theta\Delta t}\tau_{trial\,2}$. The physically meaningful intersection point will lies between the vertical lines $\tau = {}^{t+\theta\Delta t}\tau_{trial\,1}$ and $\tau = {}^{t+\theta\Delta t}\tau_{trial\,2}$, however, the "gap" defined by this bounds will usually be greater than the average of them. Consequently, the values ${}^{t+\theta\Delta t}\tau_{trial\,1}$ and ${}^{t+\theta\Delta t}\tau_{trial\,2}$ cannot be directly used to establish directly useful upper and lower bounds for ${}^{t+\theta\Delta t}\tau$. Anyway, these two values; ${}^{t+\theta\Delta t}\tau_{trial\,1}$ and ${}^{t+\theta\Delta t}\tau_{trial\,2}$; are computed because they might be useful in intermediate steps, particularly, as initial guesses for intermediate iterative procedures (see steps 12 and 13 of the implicit integration scheme in section 4.6).

Conceptually, the methodology to obtain accurate upper and lower bounds for ${}^{t+\theta\Delta t}\tau$ is described as follows:

- $\lambda = {}^{t+\theta\Delta t}\lambda_{upper}$ and $\lambda = {}^{t+\theta\Delta t}\lambda_{lower}$ define two horizontal lines in the $\tau\lambda - plane$, see Figure 2a. The actual solution point $\left({}^{t+\theta\Delta t}\overline{\lambda}^{i}, {}^{t+\theta\Delta t}\tau\right)$ lies between these two horizontal lines; see Figure 2a and the enlarged detail in Figure 2b.
- Both equations of the system $f_1(\tau,\lambda)=0$ and $f_2(\tau,\lambda)=0$ cross these horizontal lines. However, the points crossed by the function $f_2(\tau,\lambda)=0$ are closer than the points crossed by the function $f_1(\tau,\lambda)=0$, see Figure 2. The horizontal coordinates of the two intersection points between $f_2(\tau,\lambda)=0$ and the horizontal lines $\lambda = {}^{t+\theta\Delta t}\lambda_{upper}$ and $\lambda = {}^{t+\theta\Delta t}\lambda_{lower}$ define very accurate upper and lower bounds for ${}^{t+\theta\Delta t}\tau$. See Figure 2b.
- Then, to find the horizontal coordinates of these 2 intersection points it is necessary to solve two non-linear equation with one unknown for each one.

$$f_2\left(\tau, {}^{t+\theta\Delta t}\lambda_{lower}\right)=0$$

$$f_2\left(\tau, {}^{t+\theta\Delta t}\lambda_{upper}\right)=0$$

The solution of one of these equations will be ${}^{t+\theta\Delta t}\tau_{lower}$ and the solution from the other will be ${}^{t+\theta\Delta t}\tau_{upper}$.

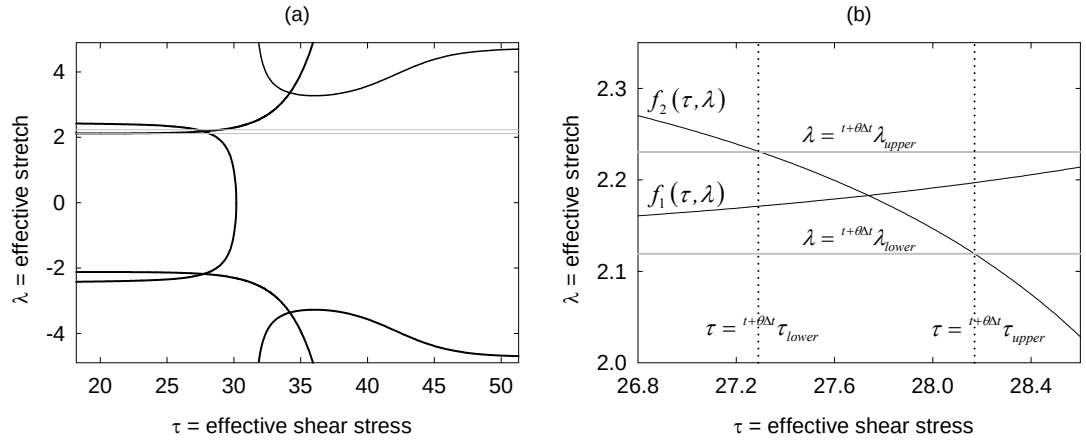

Figure 2. Figure 2a: Graph of the functions $f_1(\tau, \lambda)$ and $f_2(\tau, \lambda)$ showing the 4 intersections and the physically meaningful intersection bounded by the horizontal lines $\lambda = {}^{t+\theta\Delta t}\lambda_{upper}$ and $\lambda = {}^{t+\theta\Delta t}\lambda_{lower}$. Figure 2b: enlarged detail of the physically meaningful intersection point; the upper and lower bounds for the intersection point are defined by the horizontal (continuum) and vertical (dotted) lines. The vertical doted lines are defined by the intersections of $f_2(\tau, \lambda)$ with the horizontal lines.

4.5 Safeguard Precautions and Sources of error

The main advantages of this integration algorithm are its simplicity and efficiency, since you only have to solve a system of 2 non-linear equations with 2 unknowns. The main disadvantage is that everything depends on it. In other words, the whole reliability of this algorithm depends on the routine used to solve the system of two equations with two unknowns. The key question is: what can go wrong with the solution to this system?

One of the things that can happen is that the Newton-Raphson procedures spend too many iterations to find the solutions. To solve the this problem, the authors just imposed limits (maximum number iterations) for the Newton-Raphson Procedures. If the maximum number of iterations is reached, then, the integration procedure is aborted and reattempted using half the current time increment size. That's all what is needed solve the problem. This is a problem of minor importance.

A totally different, and more serious, kind of problem is when we obtain a "supposedly" physically meaningful solution; in other words, the correct intersection point; but the values for its coordinates $\left({}^{t+\theta\Delta t}\overline{\lambda}^{i}, {}^{t+\theta\Delta t}\tau\right)$ are absolutely unreasonable. Actually, under some particular circumstances, it is possible to obtain a solution which is "the correct one"; but which is not truly representative of the evolution law imposed by the constitutive model. In what follows, the authors will explain why can that happen and the measures that must be taken to avoid that kind of "non-representative" solutions.

How can be possible that the correct solution to the system of equations defined by equations (70) and (71) returns absolutely non-reasonable values for ${}^{t+\theta\Delta t}\overline{\lambda}^{i}$ and ${}^{t+\theta\Delta t}\tau$? The answer is simple, the two non-linear equations (70) and (71) include the approximations of expressions (63), (64) and (65), consequently, they incorporate their truncation errors too.

These two non-linear equations represent the constraint that the material point should satisfy at time $t+\theta\Delta t$. This constraint constitutes the satisfaction of the evolution law defined by the constitutive model, which is captured with the time evolution of the invariants $\overline{\lambda}^{i}$ and τ. But whether expressions (70) and (71) accurately approximate that constraint imposed by evolution law or just and aberration of it depends on the

accuracy of the approximations (63), (64) and (65).

Therefore, if we want to keep the two non-linear equations (70) and (71) truly representative of the evolution law defined by the constitutive model, we just need to keep the error of approximations (63), (64) and (65) sufficiently small. The error term of approximations (63) and (64) is of the form $\left|O_2 \cdot \Delta\gamma_\theta{}^2\right|$. In addition, the three approximations become exact expressions if $\Delta\gamma_\theta = {}^{t+\theta\Delta t}\gamma \cdot \theta\Delta t = 0$. In other words, expressions (63), (64) and (65) become exact expressions if there isn't inelastic flow from time t to $t+\theta\Delta t$.

Therefore, a maximum allowable size for the value of $\Delta\gamma_\theta$ should be imposed. Going to pragmatic measures: values such as $\Delta\gamma_\theta \cong 1$ clearly define a malfunction and plausible failure of the numerical procedure; values such as $\Delta\gamma_\theta \cong 0.5$ work well; the choice taken by the authors is that $\Delta\gamma_\theta$ should never violate the constraint $\Delta\gamma_\theta < 0.15$ (see step 19 of the implicit integration algorithm in section 4.6).

By imposing $\Delta\gamma_\theta < 0.15$, the errors associated to approximations (63) and (64) turns to be quite low. For instance, with $\Delta\gamma_\theta = 0.15$ the error included in the first order accurate approximation of the exponential mapping is in the order of the 1%. It is important to mention that $\Delta\gamma_\theta < 0.15$ is a quite conservative choice, since larger values such as $\Delta\gamma_\theta \cong 0.3 - 0.5$ were tested without any signal of malfunction.

Though, a simple limitation in the maximum admissible value of $\Delta\gamma_\theta$ is enough to solve the problem, it is still important to understand which are the greatest sources of error. Whereas approximation (63) and specially approximation (64) include large errors for large values of $\Delta\gamma_\theta$; the error of the approximation ${}^{t+\theta\Delta t}N \cong N_{trial}$ is absolutely negligible.

As stated in section 4.3, the simplification ${}^{t+\theta\Delta t}N \cong N_{trial}$, is an hypothesis or prediction about the direction of the plastic flow at time $t+\theta\Delta t$ rather than a simplification. Then, if it's a reasonable hypothesis, it should be confirmed with great accuracy when the direction of the plastic flow ${}^{t+\theta\Delta t}N$ is computed at the end of the integration procedure. Measuring the error in terms of a "distance", see expression (78), it can be verified the high accuracy of this simplification.

$$error = d\left(N_{trial\,1}, {}^{t+\theta\Delta t}N\right) = \left\| N_{trial\,1} - {}^{t+\theta\Delta t}N \right\|_F \tag{78}$$

The error of approximation ${}^{t+\theta\Delta t}N \cong N_{trial}$ is in the range of 10^{-12} and 10^{-17} for quite large time increments. Taking into account that for many numerical practitioners 10^{-16} is considered "computer accuracy", we are taking about a very small error. If deliberately large time (load) increments are attempted, then, the error of the approximation ${}^{t+\theta\Delta t}N \cong N_{trial}$ will be in the order of 10^{-10}, which is still negligible. Generally speaking, the approximation ${}^{t+\theta\Delta t}N \cong N_{trial}$ can be used with impunity, even for time increments larger than those presented in this article. It is suspected that the underlying "geometric structure" of the initial value problem posed by the inelastic flow may justify the quality of this approximation.

Basically, all the limitations for the size of the maximum allowable size for the time increment come from the expressions (63) and specially (64), which always included the largest errors. In addition, it is important to point out that any numerical implementation using any of these two approximations may suffer the same limitations.

For instance, in Tang 2007 a generalized backward Euler method is proposed for a viscoplastic constitutive model (which is very similar to the Arruda-Boyce constitutive model) which uses the approximation of expression (64) in its mathematical formulation. That kind of integration algorithms may have the same limitations that the algorithm presented in this work has.

4.6 Implicit integration scheme

1. The value of the tensors $\left\{ {}_0^t F, {}_0^t F^e, {}_0^t F^i, {}_0^{t+\Delta t} F \right\}$ is known at the beginning of each time increment. Additionally the value for θ; with $0.5 < \theta < 1$; has been set up by the user.
2. Define ${}_0^{t+\theta\Delta t} F$ as ${}_0^{t+\theta\Delta t} F := \theta \cdot {}_0^{t+\Delta t} F + (1-\theta)\, {}_0^t F$
3. Define Δt_θ as $\Delta t_\theta := \theta \cdot \Delta t$
4. Compute $F^e_{trial\,1}$; see expression (52); and compute $\bar{E}^e_{trial\,1}$ in a single operation using the Padé approximant $P_2^2\left(ln\sqrt{x}\right)$:
$$\bar{E}^e_{trial\,1} := ln\sqrt{\left(F^e_{trial\,1}\right)^T \cdot F^e_{trial\,1}}$$
5. Compute $\bar{\lambda}^i_{trial\,1}$ and $\bar{\sigma}^B_{trial\,1}$ with expressions (5) to (9) using ${}_0^t F^i$ as input argument for those expressions.
6. Compute $\bar{\sigma}^{vp}_{trial\,1}$ using the following expression
$$\bar{\sigma}^{vp}_{trial\,1} := 2\mu^e \cdot dev\left[\bar{E}^e_{trial\,1}\right] - \bar{\sigma}^B_{trial\,1}$$
For this purpose, use the value of $\bar{E}^e_{trial\,1}$ computed in the step 2 and the value of $\bar{\sigma}^B_{trial\,1}$ computed in the step 3.
7. From $\bar{\sigma}^{vp}_{trial\,1}$ computed in the last step, compute: $\tau_{trial\,1}$ and $N_{trial\,1}$, see expressions (13) and (14).
8. As a by-product of the operations 4 to 7 it is necessary to compute the following tensors
$$T1 := {}_0^t B^i$$
$$T2 := T1 \cdot N_{trial\,1} = {}_0^t B^i \cdot N_{trial\,1}$$
$$T3 := N_{trial\,1} \cdot T2 = N_{trial\,1} \cdot {}_0^t B^i \cdot N_{trial\,1}$$
$$T4 := dev\left(\bar{E}^e_{trial\,1}\right)$$
$$T5 := dev\left[T1\right] = dev\left[{}_0^t B^i\right]$$
These five tensors $\{T1, T2, T3, T4, T5\}$ together with $N_{trial\,1}$, are all the tensorial identities that are necessary to compute the 8 coefficients of the non-linear equations $f_1(\tau,\lambda) = 0$ and $f_2(\tau,\lambda) = 0$, expressions (70) and (71) respectively.
9. Compute the following coefficients:

$$C11 := \left[tr(T1) \right] \qquad C21 := \left[2\mu^e \cdot T4 : N_{trial\,1} \right]$$

$$C12 := \left[\Delta t_\theta \cdot 2 \cdot tr\left(sym(T2) \right) \right] \qquad C22 := \left[-2\mu^e \cdot \Delta t_\theta \right]$$

$$C13 := \left[\Delta t_\theta{}^2 \cdot tr(T3) \right] \qquad C23 := \left[-T5 : N_{trial\,1} \right]$$

$$C24 := \left[-2 \cdot \Delta t_\theta \cdot dev\left[sym(T2) \right] : N_{trial\,1} \right]$$

$$C25 := \left[-\Delta t_\theta{}^2 \cdot dev\left[T3 \right] : N_{trial\,1} \right]$$

The coefficients $C11$, $C12$, $C13$ belong to the non-linear equation of expression (70). On the other hand, the coefficients $C21$, $C22$, $C23$, $C24$, $C25$ belong to the non-linear equation of expression (71).

10. Compute $\overline{\lambda}^i_{trial\,2}$ and $\tau_{trial\,2}$ (scalar identities from the "trial state 2") from $F^e_{trial\,2}$ and $F^i_{trial\,2}$; see expressions (76) and (77).
11. Sort $\overline{\lambda}^i_{trial\,1}$ (computed in step 5) and $\overline{\lambda}^i_{trial\,2}$ (computed in the previous step) Assign the largest value to ${}^{t+\theta\Delta t}\lambda_{upper}$ and the smallest value to ${}^{t+\theta\Delta t}\lambda_{lower}$.
12. From $\tau_{trial\,1}$ (computed in step 7) and $\tau_{trial\,2}$ (computed in step 10) keep the smallest one, which we will call τ_{low}, this value will be used as an initial guess in the first stage of the non-linear equation solver.
13. Replace with ${}^{t+\theta\Delta t}\lambda_{upper}$ into $f_2(\tau,\lambda)=0$, and find the τ that satisfies this non-linear equation using the Newton-Raphson method. Use $\tau_0 = \tau_{low}$ as an initial guess.
14. Replace with ${}^{t+\theta\Delta t}\lambda_{lower}$ into $f_2(\tau,\lambda)=0$, and find the τ that satisfies this non-linear equation using the Newton-Raphson method. Use the value of τ obtained in the solution of the non-linear equation of the previous step as an initial guess.
15. Sort the two values of τ obtained in the steps 13 and 14. Assign the largest value to ${}^{t+\theta\Delta t}\tau_{upper}$ and the smallest value to ${}^{t+\theta\Delta t}\tau_{lower}$.
16. In steps 11 and 15 we have determined the accurate upper and lower bounds for ${}^{t+\theta\Delta t}\tau$ and ${}^{t+\theta\Delta t}\overline{\lambda}^i$, now, we define the initial guess for the two-variable Newton-Raphson procedure as an average of the upper and lower bounds:

$$\left[\tau_0, \lambda_0 \right]^T := \left[\frac{{}^{t+\theta\Delta t}\tau_{upper} + {}^{t+\theta\Delta t}\tau_{lower}}{2} ; \frac{{}^{t+\theta\Delta t}\lambda_{upper} + {}^{t+\theta\Delta t}\lambda_{lower}}{2} \right]^T$$

17. Solve the system of two non-linear equations with two unknowns $f_1(\tau,\lambda)=0$ and $f_2(\tau,\lambda)=0$ using the Newton-Raphson method and the initial guess $\left[\tau_0, \lambda_0 \right]^T$ computed in step 16.

    Each k^{th} iteration of the Newton-Raphson procedure will require the evaluation of the vector of residuals $\left[f_1(\tau,\lambda); f_2(\tau,\lambda) \right]^T$:

$$\begin{bmatrix} f_1(\tau_k,\lambda_k) \\ f_2(\tau_k,\lambda_k) \end{bmatrix} = \begin{bmatrix} \left(\gamma(\tau_k)\right)^2 \cdot C13 + \gamma(\tau_k) \cdot C12 + C11 - 3\left(\lambda_k\right)^2 \\ C21 + C22 \cdot \gamma(\tau_k) + C23 \cdot \mu^L_{eff}(\lambda_k) + C24 \cdot \mu^L_{eff}(\lambda_k) \cdot \gamma(\tau_k) + C25 \cdot \mu^L_{eff}(\lambda_k) \cdot \left(\gamma(\tau_k)\right)^2 - \tau_k \end{bmatrix}$$

    This two last formulas for $f_1(\tau_k,\lambda_k)$ and $f_2(\tau_k,\lambda_k)$ are expressions (70) and (71) but rewritten in a more compact fashion, using the coefficients $C11$, $C12$,

$C13$, $C21$, $C22$, $C23$, $C24$ and $C25$ computed in step 9.

In addition, each iteration will require the assembly, evaluation and inversion of the following Jacobian:

$$J(\tau_k, \lambda_k) = \begin{bmatrix} \frac{\partial f_1}{\partial \tau} & \frac{\partial f_1}{\partial \lambda} \\ \frac{\partial f_2}{\partial \tau} & \frac{\partial f_2}{\partial \lambda} \end{bmatrix}$$

Which again, uses the coefficients $C11$, $C12$, $C13$, $C21$, $C22$, $C23$, $C24$ and $C25$ computed in step 9.

The result of this step is supposed to be the solution "vector" $\left[{}^{t+\theta\Delta t}\tau, {}^{t+\theta\Delta t}\overline{\lambda}^{i} \right]^T$

18. With the value of ${}^{t+\theta\Delta t}\tau$; obtained from the previous step; compute ${}^{t+\theta\Delta t}\gamma$ using expressions (12).
19. If ($\Delta\gamma = {}^{t+\theta\Delta t}\gamma \cdot \Delta t_\theta < 0.15$) then

    GOTO step 20.

    Else If ($\Delta\gamma > 0.15$) then

    Interrupt the integration algorithm a make a new attempt using half the current time increment size.

    EndIf
20. Compute the rate of inelastic deformation as ${}^{t+\theta\Delta t}\overline{d}^{i} := {}^{t+\theta\Delta t}\gamma \cdot N_{trial\,1}$, where ${}^{t+\theta\Delta t}\gamma$ has been computed in step 18 and $N_{trial\,1}$ has been computed in step 7
21. Compute ${}^{t+\theta\Delta t}_{0}F^{i}$ using the exponential mapping:

$$ {}^{t+\theta\Delta t}_{0}F^{i} := exp\left({}^{t+\theta\Delta t}\overline{d}^{i} \cdot \Delta t_\theta \right) \cdot {}^{t}_{0}F^{i} $$

    Use the Padé approximant $P_2^2\left(exp(x)\right)$ to compute the exponential mapping within this operations, see expression (25).
22. Compute ${}^{t+\theta\Delta t}_{0}F^{e}$ as ${}^{t+\theta\Delta t}_{0}F^{e} := {}^{t+\theta\Delta t}_{0}F \cdot \left({}^{t+\theta\Delta t}_{0}F^{i} \right)^{-1}$
23. If ($\theta = 1$) then

$$ {}^{t+\Delta t}_{0}F^{e} := {}^{t+\theta\Delta t}_{0}F^{e} $$

$$ {}^{t+\Delta t}_{0}F^{i} := {}^{t+\theta\Delta t}_{0}F^{i} $$

    Else If ($\theta \neq 1$) then

    Re-compute ${}^{t+\theta\Delta t}\overline{d}^{i}$ from ${}^{t+\theta\Delta t}_{0}F^{e}$ and ${}^{t+\theta\Delta t}_{0}F^{i}$.

    Using this "new" rate of deformation gradient ${}^{t+\theta\Delta t}\overline{d}^{i}$, compute ${}^{t+\Delta t}_{0}F^{i}$:

$$ {}^{t+\Delta t}_{0}F^{i} := exp\left({}^{t+\theta\Delta t}\overline{d}^{i} \cdot \Delta t \right) \cdot {}^{t}_{0}F^{i} $$

    Subsequently, compute ${}^{t+\Delta t}_{0}F^{e}$:

$$ {}^{t+\Delta t}_{0}F^{e} := {}^{t+\Delta t}_{0}F \cdot \left({}^{t+\Delta t}_{0}F^{i} \right)^{-1} $$

    End If
24. With ${}^{t+\Delta t}_{0}F^{e}$ calculated (or defined) in step 23 compute ${}^{t+\Delta t}E^{e}$

$$ {}^{t+\Delta t}E^{e} := ln\sqrt{ {}^{t+\Delta t}_{0}F^{e} \cdot \left({}^{t+\Delta t}_{0}F^{e} \right)^{T} } $$

    Use the Padé approximant $P_2^2\left(ln\sqrt{x}\right)$ for this operation.
25. Compute the total Cauchy stress ${}^{t+\Delta t}T$ (spatial stress) using elastic relationship and the tensor ${}^{t+\Delta t}E^{e}$ calculated in the previous step:

$$ {}^{t+\Delta t}T := \frac{1}{J}\cdot\left(2\mu^e \cdot dev\left[{}^{t+\Delta t}E^e\right] + k^e \cdot \frac{1}{3}\cdot tr\left[{}^{t+\Delta t}E^e\right]\cdot\mathbf{1}\right) $$

26. Store ${}^{t+\Delta t}_{0}F^e$ and ${}^{t+\Delta t}_{0}F^i$ and report the total stress ${}^{t+\Delta t}T$.
27. Compute of the tangent stiffness matrix: see section 2.3

It is important to remark that this implicit integration scheme makes use of 3 Newton-Raphson procedures: see steps 13, 14 and 17. Two of this procedures are for a single equation with only one unknown (steps 13 and 14), and step 17 is for the system of two non-linear equations with two unknowns which requires the evaluation of a 2×2 Jacobian.

Analytical derivation of the jacobians may not be recommendable, since it's very easy to introduce an error in the process, in addition, it can be a quite time consuming task. To avoid the introduction of any error, in all the cases, the jacobians were derived analytically using GPL symbolic calculus software and directly exported to FORTRAN. Adaptation of the implicit numerical scheme presented in this article to different "variants" of the Arruda-Boyce constitutive model; which use different functions $f\left(\overline{\lambda},\lambda_{lock}\right)$ and $\gamma(\tau)$; is quite straightforward using this computational strategy, in fact, it only takes a few minutes.

It is noteworthy that the analytical derivation of the 2×2 jacobian requires the derivation of the scaling function $f\left(\overline{\lambda},\lambda_{lock}\right)$, consequently, it requires the derivation of the Inverse Langevín function. At this point, is where it turns be quite convenient, or at least practical, to have the inverse Langevín function defined as a single function and not as a function by parts.

Apart from the procedures to solve the non-linear equations there are no other iterative processes, since logarithms, exponentials and square roots of second order tensors are computed using Padé approximants. In addition, no rotation matrix R^e is used at any stage of the integration scheme.

It is important to mention that, in 3-dimensional analysis, it is necessary to evaluate the stress tensor a total of 7 times. The first time, it is evaluated compute the stress which will be reported to the finite element program. The other 6 times it is evaluated to compute the tangent stiffness matrix. The computational cost of these 6 re-computations can be reduced by using some information generated during the first stress-update procedure. More precisely, steps 10 to 16 can be eliminated during the 6 re-computations of the stress tensor because we don't need to obtain an accurate initial guess for the 2-variable Newton-Raphson procedure. The reason is that we already have a very accurate initial guess: when solving the systems $f_1(\tau,\lambda)=0$ and $f_2(\tau,\lambda)=0$ in each of the 6 re-computations of the stress tensor, we just use the vector $\left[{}^{t+\theta\Delta t}\tau,\ {}^{t+\theta\Delta t}\overline{\lambda}^i\right]^T$ computed in the step 17 of the first stress update as an initial guess.

5 NUMERICAL EXPERIMENTS

Three kinds of numerical experiments will be reported in this article:

- Displacement-controlled uniaxial test: displacement ramp and unload ramp returning to the initial length, see Figure 3a and Figure 3b.
- Load-controlled uniaxial test: load ramp starting from zero uniaxial stress and unload ramp returning to a zero stress state, see Figure 3c and Figure 3d.
- A displacement controlled plane-stress indentation, see Figure 4 and Figure 5

For each numerical experiment it will be reported:

- The total number of increments which were required to complete the finite element analysis.
- The total number of global equilibrium iterations required in the during the whole finite element analysis

In addition, we will report the numerical performance of the Newton-Raphson procedures within the material routine for the Load-controlled uniaxial test, see Table 4.

All the numerical results presented in this article for the implicit integration scheme were carried out with $\theta = 1$, in other words, Backward Euler scheme. Other very interesting results were obtained for $0.5 \le \theta \le 1$ but for obvious reasons of space are not presented in this article.

In all the cases, the set of 6 mechanical parameters of the model was the following one:

$$\left\{\mu^e; \lambda^e; \mu^p; \lambda_{lock}^L; \dot{\gamma}_0; \tau_{base}\right\} = \left\{251.7MPa; 2898MPa; 6.52MPa; 2.92; 1.284 \cdot 10^{-7}\,1/\mathrm{s}; 0.962MPa\right\}$$

which was taken from Bergström 2002.

Displacement and load controlled and Uniaxial tests

The goal of these two numerical experiments was to check the accuracy, stability and efficiency of the both kinds of integration algorithms. Particularly, the main interest was to find any weakness of the schemes which could eventually lead to the complete interruption of the computational process or blow up of the numerical solution.

Displacement and load-controlled tests can be regarded as very similar experiments. However, that is not true. Actually, displacement and load controlled tests behave quite differently in a computational setting. In spite of its deceptively simple aspect, the load-controlled test proved to be toughest numerical experiment carried out by the authors.

It is much difficult to find the equilibrium of the momentum equation in the load-controlled tests than in the displacement-controlled tests. For instance, while a constant tangent stiffness matrix (the elastic tangent stiffness matrix) can provisionally work in displacement-controlled uniaxial tests, it may not work at all in any load-controlled test.

Furthermore, not only the constant tangent stiffness matrix failed in load-controlled uniaxial tests but also any symmetric (or symmetrized) tangent stiffness matrix. Since nonsymmetric equation solution is as much as four times as expensive as the corresponding symmetric system, unsymmetric tangent stiffness matrices are usually avoided (see ABAQUS theory manual). Consequently, by default, ABAQUS-Standard only takes the symmetric part of the tangent stiffness matrix provided by the material routine (see ABAQUS Analysis manual). Customary practice tells that the symmetric part of the tangent stiffness matrix can be good enough for most finite element procedures, the only penalty might be a mild increase in the number of equilibrium iterations. However, for the specific case of the Arruda-Boyce constitutive model in load-controlled uniaxial tests, the symmetrized-numerically-obtained tangent stiffness

matrix did not provide any satisfactory result at all. Actually, no load-controlled uniaxial tests could be completed using a symmetrized tangent stiffness matrix. It is important to mention that only the load-controlled uniaxial tests exhibited such a marked characteristic, for all the other numerical experiments presented in this article a symmetrized tangent stiffness matrix was enough. Nevertheless, to "play safe", generalized use of the unsymmetric tangent stiffness matrix is always recommendable.

All the uniaxial tests presented in this article were performed using a single element of unitary dimensions. The maximum length reached by the element was in the order of 5 unities, in other words, the maximum "nominal strain" was in the order of 400%. The tangent stiffness matrix used by the global solver was the complete (the unsymmetric) numerically obtained tangent stiffness matrix. The methodology used for the numerical computation of the tangent stiffness matrix was presented in section 2.3.

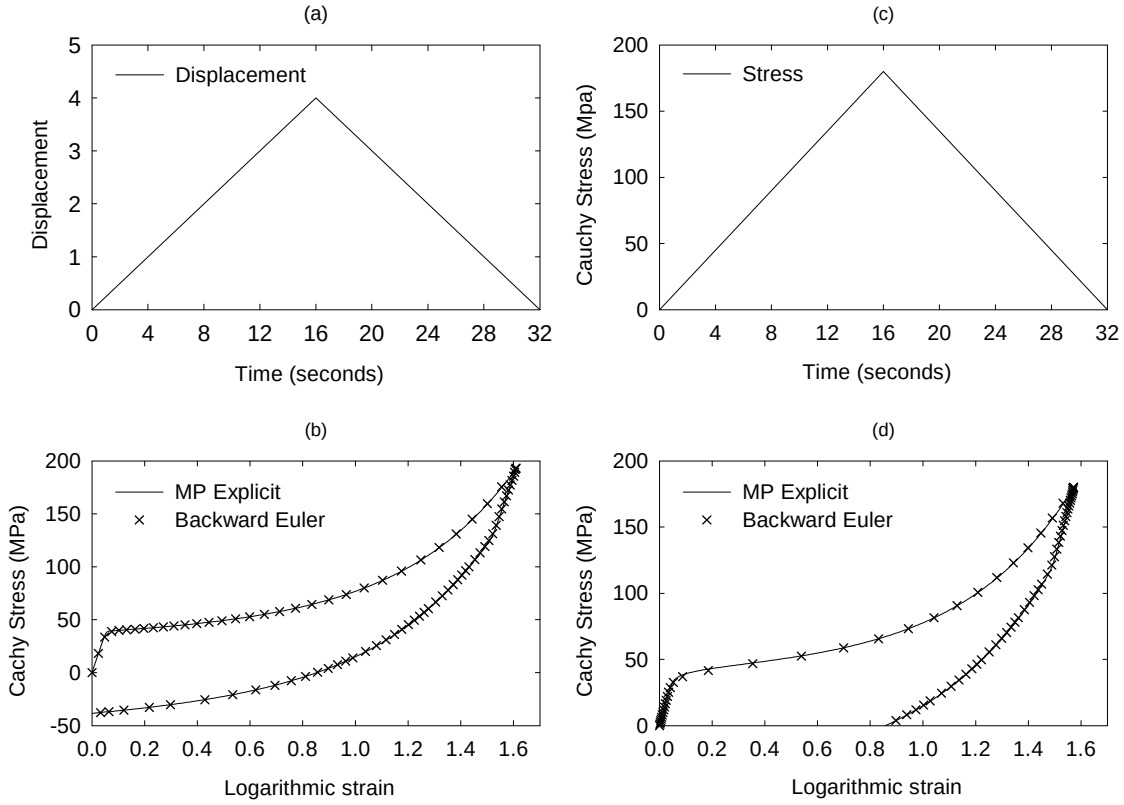

Figure 3. Figure 3a: external load for the displacement-controlled uniaxial. Figure 3b: corresponding stress-strain results for the displacement-controlled test. Figure 3c: external load for the load-controlled uniaxial test. Figure 3d: corresponding stress-strain results for the stress controlled test.

	Displacement controlled uniaxial test		Load controlled uniaxial test	
	Explicit Midpoint rule	Backward Euler	Explicit Midpoint rule	Backward Euler
Total Number of increments	908	83	1591	79
Total Number of equilibrium iterations	911	117	1605	125

Table 3. Numerical performance of the Explicit and Implicit integration schemes in the displacement and stress controlled uniaxial tests

It can be appreciated (see Table 3) that the relation between the number of increments used by the implicit and explicit integration schemes is approximately one order of magnitude (≈1/10) for the displacement controlled uniaxial test. For the case load-controlled uniaxial test, the difference is even larger: the implicit integration

scheme only requires ≈1/16 of the increments required by the explicit integration scheme. To evaluate the CPU time required for the computation with each integration algorithm, consider the total number of equilibrium iterations as directly proportional to the time consumed.

In terms of accuracy, the explicit integration scheme is the reference, this algorithm performs very accurately. On the other hand, the implicit integration scheme does an outstanding job too: it can be appreciated that the crosses (the points provided by the results of the implicit integration scheme) lie very well over the curves provided by the explicit integration scheme, sees Figure 3b and Figure 3d.

At internal level, it can be verified that the steps used to refine the initial guess (steps 13 and 14 of the implicit algorithm described in section 4.6) use a relevant number of Newton-Raphson iterations, specially the step 13. However, they yield a very accurate initial guess. With that initial guess, only 4 iterations of the two-variable Newton-Raphson procedure are required to reach a residual in the order of 10^{-16} for each function $f_1(\tau,\lambda)$ and $f_2(\tau,\lambda)$.

Additionally, it is important to comment a result that it's not in the tables nor in the figures. In section 4.5, the authors discussed about the limitations that should be imposed in the maximum strain increment $\Delta\gamma$ to avoid undesired "non-representative" solutions. In this sense, the action taken by the authors at computational level is the step 19 of the implicit integration scheme, which imposed the constraint $\Delta\gamma < 0.15$. During the whole uniaxial tests the constraint $\Delta\gamma < 0.15$ was violated in only 3 occasions, consequently, the UMAT requested 3 time increment cutbacks. Those cutbacks occurred during the unloading process, when the greatest rates of inelastic deformation occur because both the total stress (the elastic stress) and the backstress "push" in the same direction. Without those cutbacks the integration algorithm and the whole finite element model, might have crashed. Three cutbacks is a quite "acceptable price" paid for the use and abuse of the approximations (63), (64) and (65) into the formulation of the expressions for $f_1(\tau,\lambda)=0$ and $f_2(\tau,\lambda)=0$. In addition, taking into account that these pair of uniaxial tests constitute quite severe numerical experiments, it can be gathered that for general finite element analyses (not as severe as these ones) the UMAT may request not cutback at all.

	Range (Min-Max)	Average
Newton iterations for step 13	2-30	9
Newton iterations for step 14	1-5	4
Newton iterations for the system of two equations (step 17)	1-5	4

Table 4. Performance of the Newton-Raphson procedures within the material routine. The data were taken from the load-controlled uniaxial test.

Plane stress indentation test

The goal of this numerical experiment was to check the behavior of the explicit and implicit integration schemes in the context of inhomogeneous distribution of strains and strain rates and contact. In Figure 4, the reader can appreciate the geometric setup and boundary conditions of the finite element model.

Again, the difference between the explicit and implicit integration algorithm schemes is approximately one order of magnitude (1/10) for the indentation test, see Table 5. In spite of the large time increments, the accuracy of the Backward Euler method is remarkable, the crosses (the points provided by the results of the implicit integration

scheme) lie very well over the "reference curve" provided by the explicit integration scheme, see Figure 5.

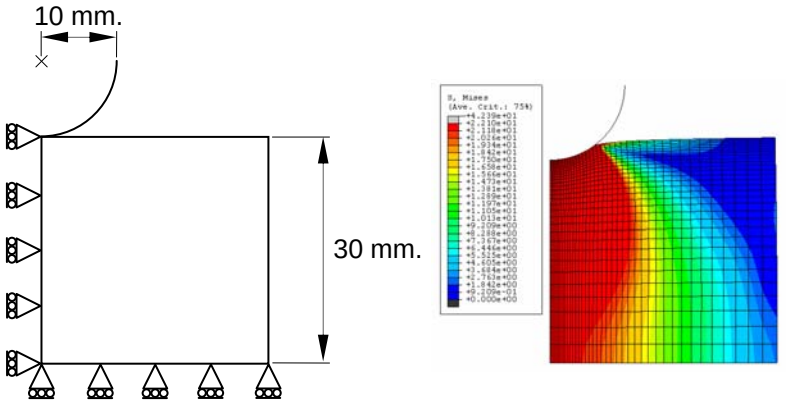

Figure 4. Indentation test: underformed geometric setup of the model and deformed shape showing Von Mises stress contours when the indenter reaches maximum depth (3 mm.). The indenter is an analytic rigid body (absolutely undeformable) with its rotational degrees of freedom constrained to zero.

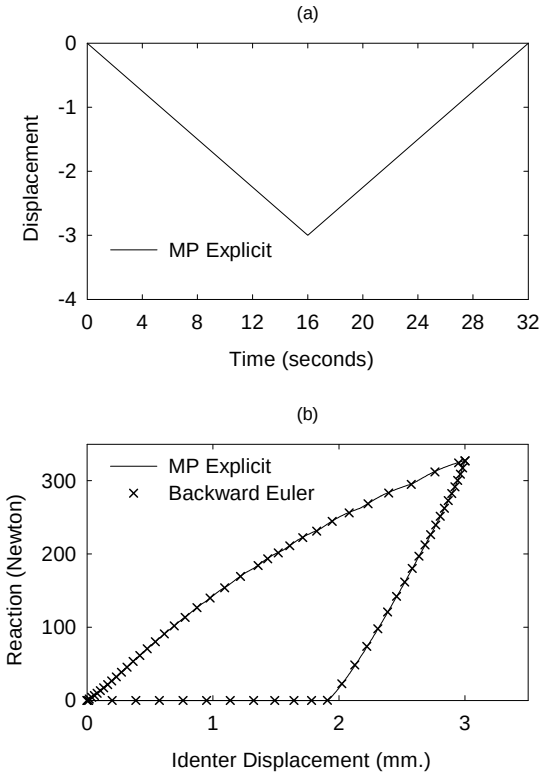

Figure 5. Figure 5a: displacement of the indenter. Figure 5b: numerically obtained force–displacement curves for the indentation test.

In the case of the indentation model, three alternative tangent stiffness matrices were tested for the implicit integration scheme: the numerically obtained tangent stiffness matrix, the symmetric part of the numerically obtained tangent stiffness matrix and the elastic tangent stiffness. The numerical performance for these three alternatives is presented Table 5. It can be appreciated that for "general" finite element analyses, the symmetric part of numerically obtained tangent stiffness matrix is enough. Furthermore,

the complete numerically obtained tangent stiffness matrix did not provide any additional benefit at all.

Additionally, the constraint $\Delta\gamma < 0.15$ was not violated during the whole finite element analysis of the indentation test. Consequently, the UMAT routine didn't request any time increment cutback.

	Explicit midpoint rule	Backward Euler		
		Symmetrized-Numerically obtained tangent stiffness matrix	Complete Numerically obtained tangent stiffness matrix	Constant tangent stiffness matrix
Total number of increments	1433	69	69	132
Total number of equilibrium iterations	1592	176	176	612

Table 5. Numerical performance of the Explicit and Implicit integration schemes in the plane stress indentation

6 GENERAL OBSERVATIONS DISCUSSION AND COMMENTS

The explicit midpoint rule combined with the time-stepping algorithm presented in this article turned to be quite reliable. In fact, this combination always did its work, completing all the finite element analysis which were tested. This is primarily a merit of the time-stepping algorithm presented in this article, which is based on the error formula of expression (35) or alternatively expression (44).

The efficiency of the explicit integration scheme is quite notorious too. For instance, the displacement-controlled test in Figure 3b required 400 increments to apply the total displacement of 4 units and 508 increments to return the element to its initial length. This is clearly much less than the minimum of 10.000 increments suggested for instance in Saleeb 2000 or the 100.000 suggested in Arya 1996. Again, the merit belongs to the time-stepping algorithm.

Some of the possible applications for this explicit integration scheme are the following ones:

- A simple and fast approach to implement unified viscoplastic models using TLH formulations in the context of constitutive model development.
- Computation of reference solutions. A good reference solution is not supposed to just be accurate, it must also provide many "discrete points" capable to resolve the most subtle details. Some details are difficult to analyze if there are only a few points in a curve, for instance, the transition from elastic behavior to plastic behavior in uniaxial tests, post-yielding strain softening, pressure dependence and fitting capabilities among others. In this sense, the time-stepping algorithm presented in this article increases the resolution (adds more "points") only when it counts, in other words, when the interesting phenomena happens and helps to keep the numerical solution deliberately smooth. In addition, reference solutions are used to qualify the accuracy of efficient integration schemes such as the implicit scheme presented in this article.
- Numerical implementation of unified viscoplastic models in within Explicit (Dynamic) finite element codes. Clearly, Explicit finite element codes base their power in the fast computation of hundreds of thousands increments dealing with the most severe non-linearities. In this context, prompt computation of the material routine is a very important factor, perhaps, more important than stability of the numerical results provided by the material routine. Stability may not be important for simple reason: the time increments in Explicit finite

element analyses are so small that it is unlikely that the material routine (UMAT) could require (and hence request) smaller time increments than those provided by the Explicit Finite Element Solver.

From the point of view of "numerical experimentation", it is quite interesting the fact that enforcement of "local convergence" generally provided very stable solutions. On the other hand, from numerical analysis theory, it is known that "consistency + stability $\Rightarrow$ convergence" is a just a one way relation, consequently, the converse; "convergence $\Rightarrow$ consistency + stability"; is not generally true. However, for pragmatic purposes, the converse worked quite well.

On the other hand, we have that the implicit numerical scheme for the Arruda-Boyce constitutive model turned to be particularly efficient allowing to use large time increments.

The number of approximations and simplifications included in the analytical derivation of the conditions $f_1(\tau,\lambda)=0$ and $f_2(\tau,\lambda)=0$ should not be ignored. As a consequence of those approximations, the conditions $f_1=0$ and $f_2=0$ have a finite range of validity. Basically, these conditions loose its validity (and physical sense) if $\Delta\gamma$ takes large values. In a computational setting, the constraint imposed to $\Delta\gamma$ was violated in rare occasions. Consequently, it can be said that the massive use of the approximations (63), (64) and (65) only added minor limitations to the implicit integration scheme. However, the constraint imposed to the value of $\Delta\gamma$ should not be considered as an "accessory routine" of the integration scheme, the inclusion of that constraint is mandatory to guarantee robust analysis procedures.

It is important to note that the implicit integration scheme presented in this article is not limited to the original A-B model. Considering the A-B model just as an elementary basis, we can modify (or change) it's flow rule $\gamma(\tau)$, modify (or change) its scaling $f\left(\overline{\lambda},\lambda_{lock}\right)$ and add some scalar internal state variables in order to adapt it to specific applications. Playing with these aspects, it is possible to develop advanced constitutive models (capable to reproduce specific phenomena of polymers and biomaterials) and their corresponding numerical implementations. For instance, it is quite easy to adapt the implicit integration scheme presented in this article to a constitutive model with the following form:

- Magnitude of the rate of inelastic deformation defined by the power law; see expression (79):

$$\gamma(\tau)=\dot{\gamma}_0\cdot\left(\frac{\tau}{\tau_{vp}}\right)^m \tag{79}$$

- The shear resistance τ_{vp} of the viscoplastic element does not need to be a constant, it can be modeled in rate-form as a scalar state variable as it has been done in Arruda 1993b, Bergström 2002 or Anand and Gurtin 2003. See expressions (80) and (81) depicting the functional dependencies and the application of the backward Euler-operator applied to the shear resistance τ_{vp}.

$$\dot{\tau}_{vp}=g\left(\gamma(\tau)\right) \tag{80}$$

$$^{t+\Delta t}\tau_{vp}={}^{t}\tau_{vp}+\Delta t\cdot g\left(\gamma\left({}^{t+\Delta t}\tau\right)\right) \tag{81}$$

- We could add one modification more: keeping the original scaling function $f\left(\bar{\lambda},\lambda_{lock}\right)$ for the Hyperelastic model, the mechanical parameters μ^p and λ_{lock}^L can be modeled in rate-form becoming scalar state variables:

$$\dot{\mu}^p = h\left(\gamma(\tau)\right) \tag{82}$$

$$^{t+\Delta t}\mu^p = {}^t\mu^p + \Delta t \cdot h\left(\gamma\left({}^{t+\Delta t}\tau\right)\right) \tag{83}$$

  This last possibility is very interesting to model the evolution of "isotropic" damage ("degradation" of the elastic constants) in the hyperlastic element. This feature might be necessary to model cyclic behavior.

In spite of these modifications, such a constitutive model can be implemented using the same integration algorithm presented in this article. The mathematical problem can be reduced to the determination of $\left({}^{t+\Delta t}\bar{\lambda}^i, {}^{t+\Delta t}\tau\right)$ from a system of two non-linear equations, just like it was done for the original A-B model.

Additionally, the numerical procedure presented to compute the algorithmic tangent stiffness matrix turned to be quite efficient. A good measure of the computational efficiency (CPU time) is the total number of equilibrium iterations. Specifically looking at Table 5 it can be appreciated that the implicit integration scheme in combination with the elastic tangent stiffness matrix required 612 increments to complete the analysis, while the same integration scheme combined with the numerically obtained tangent stiffness matrix required just 176 increments. This is a considerable reduction in the global computational cost and CPU time.

In other circumstances, the major concern related to the tangent-stiffness matrix is not efficiency, but rather obtaining a solution or no one at all. For instance, the complete numerically-obtained tangent-stiffness matrix is mandatory for the simulation of load (stress) controlled deformations processes in which occur relevant orientation of the of the material fibers (anisotropic evolution processes). On the other hand, we have that the symmetrized tangent stiffness matrix is quite enough for general displacement-controlled multiaxial loading conditions in which only moderate strains and minor orientation of the material fibers occur. Any intermediate condition, between the two cases previously described, must be adjusted on case by case basis.

Finally, it is important to remark that both numerical implementations do not use any eigen value computation nor rotation tensor. Not even the perturbation of the deformation gradient requires the application of eigen-decomposition.

7 CONCLUSIONS

Two simple numerical implementations for the Arruda-Boyce constitutive models have been presented in this article, one of them explicit and the other one implicit. Both material routines were coded as an UMAT for the software ABAQUS-STANDARD.

The explicit integration scheme turned to be very stable and quite fast in spite of its natural limitations for maximum allowable size of the time increment size. Most the merits of its performance are due to the time-stepping presented in this article. The time-stepping algorithms presented in this article "intends" to enforce local converge of the numerical solution.

In addition, this explicit integration algorithm is deliberately accurate and ideally suited for testing trial ideas, constitutive model development, computation of reference solutions and may work quite well in Explicit finite element codes.

On the other hand, the implicit numerical implementation turned to be very efficient and quite accurate too. On the basis of the Arruda-Boyce constitutive model and its

kinematic framework, this numerical implementation can be easily modified to test other flow rules and hyperelastic models for the back stress which can include scalar state variables.

Both computational implementations worked in combination with numerically obtained tangent stiffness matrix and the Padé approximants for the computation of square roots, natural logarithms and exponentials of second order tensors.

8 ACKNOWLEDGEMENTS

This work has been partially supported the grants PICT-12-14114; "Biomecánica de Miembro superior" from the "Agencia Nacional de Promoción científica y Tecnológica", and project PIP 6253 "Biomateriales para implantes óseos y cartílagos, evaluación y análisis de las propiedades de transporte y del comportamiento mecánico a escala macro y nanométrica" from CONICET. This support is gratefully acknowledged.

The first author wants to thank all the people who helped him in what should be the "easy task" of getting specific papers and articles. He would also like to thank MIT libraries services for providing free access to all their graduate theses. Those thesis have been an very useful source of information for the development if this article.

REFERENCES

ABAQUS Analysis user's manual, ABAQUS 6.5 ; *ABAQUS*, Inc; (2004)

ABAQUS theory manual, ABAQUS 6.5 ; *ABAQUS*, Inc; 2004.

Anand, L. and Gurtin, M.E.; "A theory of amorphous solids undergoing large deformations with applications to polymeric glasses"; International Journal of Solids and structures; 2003

Arruda, E.M. and Boyce, M.C.; "A Three-Dimensional Constitutive Model for the Large Stretch Behavior of Rubber Elastic Materials"; *J. Mech. Phys. Sol.*; 41:389,1993a.

Arruda, E.M. and Boyce, M.C.; "Evolution of plastic anisotropy in amorphous polymers during finite straining"; *International Journal of Plasticity*; 9:697–720, 1993b.

Arya, V.K.; "Efficient and accurate explicit integration algorithms with applications to viscoplastic models"; *International Journal for Numerical Methods in Engineering*; 39:261-279, 1996.

Bathe, K.J.; "Finite element procedures"; Prentice Hall; 1996

Bender, C.M and Orszag, S.A.; "Advanced mathematical methods for scientists and engineers"; McGraw-Hill; 1978

Bergström, J.S.; Kurtz, S.M.; Rimmac, C.M.; Edidin, AA; "Constitutive modeling of ultra-high molecular weight polyethylene under large-deformation and cyclic loading conditions"; *Biomaterials*; 23:2329-2343, 2002.

Bergström, J.S.; Bowden A.E.; Rimmac, C.M.; Kurtz, S.M.; "Development and Implementation of an Advanced User Material Model for UHMWPE"; *9th International LS-DYNA Users Conference*; 2006.

Bodner, S.R. and Partom, Y.; "Constitutive equations for elastic–viscoplastic strain-hardening materials"; ASME Journal of Applied Mechanics; 1975;42:385–9

Butcher, J.C.; "Numerical Methods for ordinary differential equations"; Wiley; 2003

Dvorkin, E.N.; Goldschmit, M.B.; "Nonlinear Continua"; Springer; 2006.

Eterovic, L.A. and Bathe, K.J.; "A Hyperelastic-Based Large Strain Elasto-Plastic Constitutive Formulation with Combined Isotropic-Kinematic Hardening Using the Logarithmic Stress and Strain Measures"; *International Journal for Numerical Methods in Engineering*; 30:1099-1114, 1990.

Fritzen, P. and Wittekindt, J.; "Numerical Solution of Viscoplastic Constitutive

Equations with Internal State Variables, Part I"; *Mathematical Methods in the Applied Sciences*, John Wiley & Sons; 20:1411-1425, 1997.

Hairer, E., Norsett, S.P., Wanner, G.; "Solving ordinary differential equations I"; Springer; 1993

Heath, M.T.; "Scientific computing, An Introductory Survey"; McGraw-Hill; 1997.

Kirchner, E. and Simeon, B.; "A higher-order time integration method for viscoplasticity"; *Computer Methods in Applied Mechanics and Engineering*; 1999

Kojic, M. and Bathe, K.J.; "The 'Effective-Stress-Function' Algorithm for Thermo-Elasto-Plasticity and Creep"; *International Journal for Numerical Methods in Engineering*, 24:1509-1532, 1987.

Lush, A.M., Weber, G., Anand, L.; "An implicit time integration procedure for a set of internal variable constitutive equations for isotropic elasto-viscoplasticity"; *International Journal of Plasticity*; 5:521-549; 1989

Miehe, C.; "Numerical computation of algorithmic (consistent) tangent moduli in large-strain computational inelasticity"; *Computer Methods in Applied Mechanics and Engineering*; 134:223-240, 1996.

Ortiz, M., Radovitzky, R.A., Repetto E.A.; "The computation of the exponential and logarithmic mappings and the their first and second order linearizations"; *International Journal for Numerical Methods in Engineering*"; 52:1431-1441; 2001

Press, W.H., Teukolsky, S.A., Vetterling W.T. and Flannery, B.P.; "Numerical Recipes in Fortran 77"; Cambridge University Press; 1986-1992.

Saleeb, A.F., Wilt, T.E., Li, W; "Robust integration schemes for generalized viscoplasticity with internal-state variables"; *Computers & Structures*; 74:601–628; 2000

Sansour, C. and Kollmann, F.G.; "On theory and numerics of large viscoplastic deformation"; *Computer Methods in Applied Mechanics and Engineering*; 146:351-369, 1997.

Simo J.C. & Hughes T.J.R.; "Computational inelasticity"; 1998

Tang, H., Barthelat, F., Espinosa, H.D; "An elasto-viscoplastic interface model for investigating the constitutive behavior of nacre"; *Journal of the Mechanics and Physics of solids*; 2007

Weber, G. and Anand, L.; "Finite deformation constitutive equations and a time integration procedure for isotropic, hyperelastic-viscoplastic solids"; *Computer Methods in Applied Mechanics and Engineering*; 79:173-202, 1990.

Zirky, M.A.; "An accurate and stable algorithm for high strain-rate finite strain plasticity"; *Computers & Structures*; 50:337-350, 1994.