Order reduction in computational inelasticity: Why it happens and how to overcome it—The ODE-case of viscoelasticity

Bernhard Eidel[1, *, †] and Charlotte Kuhn[2]

[1]*Institut für Mechanik, Fakultät für Ingenieurwissenschaften, Abteilung Bauwissenschaften,
Universität Duisburg-Essen, Universitätsstr. 15, 45117 Essen, Germany*
[2]*Lehrstuhl für Technische Mechanik, Universität Kaiserslautern, 67653 Kaiserslautern, Germany*

SUMMARY

Time integration is the numerical kernel of inelastic finite element calculations, which largely determines their accuracy and efficiency. If higher order Runge–Kutta (RK) methods, $p \geqslant 3$, are used for integration in a standard manner, they do not achieve full convergence order but fall back to second-order convergence. This deficiency called order reduction is a longstanding problem in computational inelasticity. We analyze it for viscoelasticity, where the evolution equations follow ordinary differential equations. We focus on RK methods of third order. We prove that the reason for order reduction is the (standard) linear interpolation of strain to construct data at the RK-stages within the considered time interval. We prove that quadratic interpolation of strain based on t_n, t_{n+1} and, additionally, t_{n-1} data implies consistency order three for total strain, viscoelastic strain and stress. Simulations applying the novel interpolation technique are in perfect agreement with the theoretical predictions. The present methodology is advantageous, since it preserves the common, staggered structure of finite element codes for inelastic stress calculation. Furthermore, it is easy to implement, the overhead of additional history data is small and the computation time to obtain a defined accuracy is considerably reduced compared with backward Euler. Copyright © 2011 John Wiley & Sons, Ltd.

1. INTRODUCTION

The formulation of higher order time integration schemes in computational inelasticity without order reduction is a longstanding problem of numerical analysis and computational mechanics. Order reduction in this context means that the application of higher order ($p \geqslant 3$) one-step, multistage methods of the Runge–Kutta (RK) family for integrating inelastic evolution equations fails to achieve the theoretical order of convergence. Instead, these methods result in second-order accuracy at best. Order reduction was mainly observed for elasto-(visco)-plastic constitutive models. These models typically exhibit the format of differential algebraic equations (DAE), where the evolution equations of plastic flow are described by ordinary differential equations (ODE) which are subjected to the yield condition as the algebraic constraint.

Backward Euler is well established due to its simplicity and excellent stability properties, although it is only linear, [1]. Second-order methods ($p = 2$) like midpoint rule, [2–6], achieve full order of convergence in elasto-plastic stress calculations, if the formulation is stiffly accurate.

---

*Correspondence to: Bernhard Eidel, Institut für Mechanik, Fakultät für Ingenieurwissenschaften, Abteilung Bauwissenschaften, Universität Duisburg-Essen, Universitätsstr. 15, 45117 Essen, Germany.
†E-mail: bernhard.eidel@uni-due.de

Similarly, the Backward Difference Formula (BDF-2), [7–9], equally does not exhibit order reduction. Third-order or higher order methods ($p \geqslant 3$) have been found to exhibit an order reduction to second order, [4, 5, 10–13], which seems to be inevitable for rate-independent elasto-plasticity. In more detail, the state of the art can be summarized as follows. In [5], Ellsiepen reports for Prandtl–Reuss plasticity an order reduction to second order for a stiffly accurate, diagonally implicit RK method (SDIRK) of third order [14]. Based on these results the conclusion in [5] and [10] is that for suchlike *numerical hard problems* third-order methods do not pay off. A similar result is found in [4]. However, in [13] it is found that hardening does not cure order reduction for $p \geqslant 3$. Further attempts to regularize the problem of elasto-plasticity via viscosity of Perzyna-type models failed, third-order methods achieve second order at best. An exception to this rule is the case where an unphysically large viscosity was used, which may avoid order reduction in some cases, see [13]. Similarly, order reduction is reported in [11] and [12] where a third-order Radau IIa scheme, a stiffly accurate RK method, is applied to von–Mises plasticity.

In computational viscoelasticity, linear backward Euler is mainly used for time integration and is for constant step size a good choice, cf. [15–19]. Higher order methods ($p \geqslant 2$) are rarely used so far. An exception is the work of Hartmann and coworkers, [20, 21], which follows the algorithmic structure of the method of vertical lines (MOL) to be described in Section 1.1. In [21] Rosenbrock-type methods up to order four have been successfully used for viscoelasticity. As will be shown in the present contribution, order reduction equally occurs for the ODE case of viscoelasticity, if the classical algorithmic structure of implicit finite element methods is preserved; this structure according to a partitioned or staggered ansatz will be described in Section 1.1.

Summarizing, the problem of order reduction is still an open issue. Despite considerable research efforts the reason for order reduction in computational inelasticity for both, viscoelasticity and elasto-plasticity, is not well understood, a solution of this problem within the classical finite element approach of a *staggered* algorithmic scheme is still missing. For this case, the present paper aims to analyze and solve this issue for computational viscoelasticity. In detail, the main aim is threefold:

  (i) First, the mathematical structure of inelastic stress calculation in the finite element method will be presented which is the basis for this paper.

  In Section 2, we introduce a three-parameter viscoelastic constitutive model which serves the purpose of a prototype model in geometrically non-linear finite element calculations. Based on a generalization of the initial value problem (IVP) for viscoelasticity we describe in Section 3 the application of RK methods for time integration. For stiffly accurate RK methods—here we employ a two-stage, third-order Radau IIa version—we adapt the solution algorithm to the chosen viscoelasticity model.

 (ii) Second, we solve the problem of order reduction; we explain why it occurs, propose a simple yet effective concept to overcome it, and prove that it is true, Sections 4 and 5.

  Our hypothesis, why order reduction happens, is based on an obvious fact and a very simple idea: in inelastic structures the displacement/strain path in time is in general non-linear at arbitrary material points. Multistage methods require the construction of strain at stages within the time interval where evolution equations are integrated. If, for that aim, standard linear interpolation is used, an approximation error is introduced. It is our hypothesis that this error in interpolation is passed to the order of convergence of time integration implying order reduction to second order if the RK method exhibits order $p \geqslant 3$. If this claim is true, then quadratic interpolation for strain introducing an approximation error of third order should cure the problem. The corresponding proof is based on Taylor series expansions of the numerical solution on one hand and the exact solution on the other hand. The comparison of both solutions amounts to conditions for the RK scheme, which turn out to be implicitly fulfilled by RK methods by their very construction.

(iii) Third, we assess the present methodology in representative, geometrically non-linear finite element simulations, where the viscoelastic evolution equations are integrated by the third-order RK method of Radau IIa, Section 6.

1.1. Structure of finite element methods for inelasticity

The numerical simulation of structures undergoing inelastic (viscoelastic or elasto-plastic) material behavior is a standard task in computational mechanics and engineering. For the application of the finite element method in implicit quasi-static simulations two different approaches can be distinguished.

The partitioned/staggered approach. In standard (commercial) finite element codes the corresponding numerical solution exhibits mostly the same structure: the IVP of inelastic flow is solved on a *local* (Gauss-point) level where the evolution equations are integrated by *stress-update* algorithms. The non-linear algebraic equations resulting from the spatial discretization of the weak form of the balance of momentum is typically solved by Newton's method on the *global* level. In the following we call this the *partitioned* ansatz or the *staggered* ansatz.

The method of lines (MOL)/DAE approach. Another, more recent approach of inelasticity in finite element analysis is the application of the MOL, see [10, 20–23].

Within this concept, the spatial discretization is followed by the temporal discretization. The spatial discretization using finite elements is standard and leads to a system of non-linear algebraic equations—the algebraic part of a DAE-system—with nodal displacements and internal variables at spatial integration points (Gauss-points) as unknowns. The differential part of the DAE-system consists of ODEs of first order describing the evolution of the internal variables. Both, the non-linear algebraic equations and the system of ODEs form a semi-explicit system of DAEs of first order. This applies to the broad class of rate-dependent constitutive models like viscoelasticity and viscoplasticity described by ODEs. For more details we refer to [10, 20].

The present work follows the partitioned/staggered ansatz. All statements and conclusions exclusively apply to that standard approach in classical finite element codes.

2. CONSTITUTIVE MODEL OF VISCOELASTICITY

In the following we introduce a 3D and geometrical non-linear generalization of the standard three-parameter viscoelastic material model as depicted in its 1D, geometrical linear rheological format in Figure 1. For viscoelastic constitutive models see e.g. [24, 25]. The chosen model is used to showcase the numerical solution of the corresponding evolution equations of ODE-type within the partitioned ansatz.

$$\text{decomposition of strain} \quad \boldsymbol{E} = \boldsymbol{E}^{\text{e}} + \boldsymbol{E}^{\text{v}} \tag{1}$$

$$\text{decomposition of stress} \quad \boldsymbol{S} = \boldsymbol{S}^{\text{eq}} + \boldsymbol{S}^{\text{ov}} \tag{2}$$

$$\text{equilibrium stress} \quad \boldsymbol{S}^{\text{eq}} = K_{\infty} \text{tr}(\boldsymbol{E})\boldsymbol{1} + 2\mu_{\infty}\boldsymbol{E}^{D} \tag{3}$$

$$\text{overstress} \quad \boldsymbol{S}^{\text{ov}} = 2\mu(\boldsymbol{E} - \boldsymbol{E}^{\text{v}})^{D} \tag{4}$$

$$\text{flow rule} \quad \dot{\boldsymbol{E}}^{\text{v,D}} = \frac{2\mu}{\eta}(\boldsymbol{E} - \boldsymbol{E}^{\text{v}})^{D} = \frac{1}{\eta}\boldsymbol{S}^{\text{ov}} \tag{5}$$

The model is based on the assumption that the Green–Lagrangean strain tensor $\boldsymbol{E}$ can be additively decomposed into an elastic part $\boldsymbol{E}^{\text{e}}$ and a viscous part $\boldsymbol{E}^{\text{v}}$, (1). For 3D viscoelasticity, the springs in Figure 1 are replaced by elastic bodies with *bulk modulus* K_{∞}, and *shear moduli* μ and μ_{∞}, respectively. In the decomposition

$$\boldsymbol{E} = \boldsymbol{E}^{\text{D}} + \tfrac{1}{3}\text{tr}(\boldsymbol{E})\boldsymbol{1} \tag{6}$$

$\boldsymbol{E}^{\text{D}}$ is the strain deviator, $\text{tr}(\boldsymbol{E}) = E_{ii}$ is the trace operator and $\boldsymbol{1}$ denotes the second-order unit tensor with components δ_{ij}. Assuming $\boldsymbol{E}^{\text{v}}$ to be deviatoric the constitutive relations for the *equilibrium*
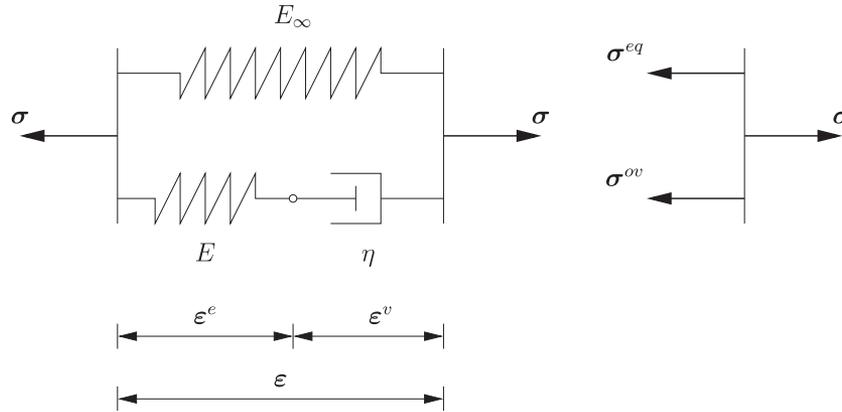
Figure 1. Rheological 1D representation of a 3-parameter viscoelasticity model.

stress S^{eq} and for the *overstress* S^{ov} read (3) and (4). With the *initial bulk modulus* $K_0 = K_\infty$ and the *initial shear modulus* $\mu_0 = \mu_\infty + \mu$ the constitutive relation for S can be rewritten as

$$S = K_0 \text{tr}(E)\mathbf{1} + 2\mu_0 E^{\text{D}} - 2\mu E^{\text{v,D}} \tag{7}$$

where the material parameter η is the *viscosity*. The reciprocal value of the scalar prefactor in (5), $\tau = \eta/2\mu$, is the *relaxation time*.

The above viscoelastic model falls into the general class of constitutive equations of the type

$$S = h(E, z) \tag{8}$$

$$\dot{z} = f(E, z), \quad z(t_0) = z_0 \tag{9}$$

where (8) denotes an elasticity relation and $z \in \mathbb{R}^{n_z}$ defines a set of internal variables describing inelastic material behavior. Here, $z = \{E^{\text{v}}\}$ with $n_z = 6$ holds for the 3D case.

3. TIME DISCRETIZATION BY IMPLICIT RK METHODS

3.1. Basic equations

To put things into perspective we briefly recall some basic equations for the solution of an IVP by implicit RK (IRK) methods, cf. e.g. [26–28]. The IVP exhibits the format

$$\dot{z} = f(z), \quad z(t_0) = z_0, \quad t \in [t_0, T] \tag{10}$$

which consists of an ODE, $(10)_1$, along with initial conditions, $(10)_2$. In (10) we drop for notational convenience the argument E but keep in mind that in computational inelasticity the IVP is embedded in a boundary value problem (BVP) that is solved by finite elements resulting in displacements u and total strains E. The total time interval is decomposed into subintervals $t_0 < t_1 < \cdots < t_n < \cdots < t_{n+1} < \cdots < t_N = T$, the *time steps* with time step size $\Delta t_{n+1} = t_{n+1} - t_n$, $0 \leqslant n \leqslant N$. Assuming that the exact solution at t_n is given as $z(t_n)$, the solution at t_{n+1} is searched

$$z(t_{n+1}) = z(t_n) + \int_{t_n}^{t_{n+1}} f(t, z(t)) \, dt. \tag{11}$$

For the numerical solution $z_{n+1} \approx z(t_{n+1})$ the integral is calculated by a quadrature rule consisting of s stages

$$z_{n+1} = z(t_n) + \Delta t_{n+1} \sum_{i=1}^{s} b_i f(t_n + c_i \Delta t_{n+1}, z(t_n + c_i \Delta t_{n+1})) \tag{12}$$

Table I. Butcher arrays for (left) implicit RK (IRK) methods, (center) for
backward-Euler, (right) for Radau IIa, $s = 2$.

$$
\begin{array}{c|cccc}
c_1 & a_{11} & a_{12} & \cdots & a_{1s} \\
c_2 & a_{21} & a_{22} & \cdots & a_{2s} \\
\vdots & & & \ddots & \vdots \\
c_s & a_{s1} & a_{2s} & \cdots & a_{4s} \\
\hline
& b_1 & b_1 & \ldots & b_s
\end{array}
\qquad
\begin{array}{c|c}
1 & 1 \\
\hline
& 1
\end{array}
\qquad
\begin{array}{c|cc}
\dfrac{1}{3} & \dfrac{5}{12} & -\dfrac{1}{12} \\
1 & \dfrac{3}{4} & \dfrac{1}{4} \\
\hline
& \dfrac{3}{4} & \dfrac{1}{4}
\end{array}
$$

with weighting factors b_i, $i = 1, \ldots, s$ and the coefficients c_i, $i = 1, \ldots, s$ where the latter define new time *stages* $t_i = t_n + c_i \Delta t_{n+1}$. The unknowns $z(t_n + c_i \Delta t_{n+1})$ are calculated by a second integration step in analogy to (12) employing the same stages c_i but along with the weighting factors a_{ij} building the *Runge–Kutta-Matrix* $A = (a_{ij})_{i,j=1,\ldots,s}$, hence

$$z(t_n + c_i \Delta t_{n+1}) \approx \mathbf{Z}_i = z(t_n) + \Delta t_{n+1} \sum_{j=1}^{s} a_{ij} \mathbf{f}(t_j, \mathbf{Z}_j), \quad i = 1, \ldots, s. \tag{13}$$

With $\dot{\mathbf{Z}}_j := \mathbf{f}(t_j, \mathbf{Z}_j)$, the *stage derivatives*, we can rewrite (13) as

$$\mathbf{Z}_i = z(t_n) + \Delta t_{n+1} \sum_{j=1}^{s} a_{ij} \dot{\mathbf{Z}}_j, \quad i = 1, \ldots, s. \tag{14}$$

Hence, $\mathbf{Z}_i$ and $\dot{\mathbf{Z}}_i$ are two sets of unknowns which are related by (14). After the calculation of the stage derivatives $\dot{\mathbf{Z}}_i$ the update for z_{n+1} reads as

$$z_{n+1} = z_n + \Delta t_{n+1} \sum_{i=1}^{s} b_i \dot{\mathbf{Z}}_i. \tag{15}$$

A typical representation of RK methods is the *Butcher-array*, see Table I. For RK methods the relation

$$c_i = \sum_{j=1}^{s} a_{ij} \tag{16}$$

holds between the lines $a_{i.}$ of A and the stages c_i. If the coefficients fulfil the two conditions

(i) $b_i \geqslant 0$, $i = 1, \ldots, s$,

(ii) $M := (b_i a_{ij} + b_j a_{ji} - b_i b_j)_{i,j=1,\ldots,s}$ is positive semidefinite,

the scheme is called *algebraic stable*, [26]. It can be shown that algebraic stable, implicit RK-methods are B-stable (and for that reason A-stable as well), [26]. This is especially true for schemes of the Radau IIa class, where coefficients for $s \in \{1, 2\}$ are summarized in Table I. For a survey on stability properties of time integration algorithms of elasto-plasticity we refer to [29]. An s-stage RK method of Radau IIa class is of the order $p = 2s - 1$ for the ODE case, cf. Table VII.4.1 in [26].

In summary, the solution of the IVP of the format (10) by a fully implicit RK method is obtained in the following steps:

(I) Solve the generally non-linear set of equations for the *stage solutions* $\mathbf{Z}_i$

$$\mathbf{Z}_i = z_n + \Delta t \sum_{j=1}^{s} a_{ij} \mathbf{f}(t_j, \mathbf{Z}_j), \quad i = 1, \ldots, s. \tag{17}$$

(II) Compute the *stage derivatives* $\dot{\mathbf{Z}}_i$

$$\dot{\mathbf{Z}}_i = \mathbf{f}(t_i, \mathbf{Z}_i), \quad i = 1, \ldots, s. \tag{18}$$

(III) Calculate the approximation of z at time t_{n+1}

$$z_{n+1} = z_n + \Delta t \sum_{i=1}^{s} b_i \dot{\mathbf{Z}}_i. \tag{19}$$

3.2. Stiffly accurate RK schemes

For a *stiffly accurate* RK-scheme, step (II) can be skipped (i.e. the stage derivatives $\dot{\mathbf{Z}}_i$ need not be calculated) as these schemes meet the requirement $a_{si} = b_i$ for all $i = 1, \ldots, s$ and the approximate solution z_{n+1} of step (III) coincides with the last-stage solution $\mathbf{Z}_s$

$$z_{n+1} = z_n + \Delta t \sum_{j+1}^{s} b_j \dot{\mathbf{Z}}_j = z_n + \Delta t \sum_{j=1}^{s} a_{sj} \dot{\mathbf{Z}}_j = \mathbf{Z}_s. \tag{20}$$

Next, we explain the application of stiffly accurate RK-methods like those of the Radau IIa family to the evolution equations of viscoelasticity (5). Introducing the stage solutions $\boldsymbol{E}_i^{\mathrm{v,D}}$ at $t_i = t_n + c_i \Delta t$ the system of equations for the stage solutions (17) reads as

$$\boldsymbol{E}_i^{\mathrm{v,D}} = \boldsymbol{E}_n^{\mathrm{v,D}} + \frac{\Delta t}{\tau} \sum_{j=1}^{s} a_{ij} (\boldsymbol{E}(t_j) - \boldsymbol{E}_j^{\mathrm{v}})^{\mathrm{D}}. \tag{21}$$

As the flow rule is linear, the same is true for (21). This set of linear equations exhibits $6s$ unknowns, which is an increase in numerical effort compared to e.g. backward Euler, where $\boldsymbol{E}_{n+1}^{\mathrm{v,D}}$ can be computed explicitly from $\boldsymbol{E}_n^{\mathrm{v,D}}$ and $\boldsymbol{E}_{n+1}^{\mathrm{D}}$.

Since only stiffly accurate schemes are used in this paper, data at t_{n+1} are the stage values of the last stage s

$$\boldsymbol{E}_{n+1}^{\mathrm{v,D}} = \boldsymbol{E}_s^{\mathrm{v,D}}. \tag{22}$$

Within the *partitioned/staggered approach*, see Section 1.1, displacements and strains fulfilling the weak form of the balance of momentum are calculated at t_n, t_{n+1} as well as at former times $t_{n-1}, \ldots$. At RK stages c_j in the interior of each time interval, displacements and strains have to be approximated by interpolation. These approximated strains as obtained from the *global* finite element solution serve as input for the *local* problem of time integration. In the following we will analyze in detail, how the construction of stage values of strains via linear or higher order polynomials for interpolation does affect the accuracy of time integration.

4. HYPOTHESIS, WHY ORDER REDUCTION OCCURS AND HOW TO OVERCOME IT

4.1. The standard ansatz: stage values via linear interpolation

The easiest way to determine the stage values is via linear interpolation, as applied e.g. in [12].
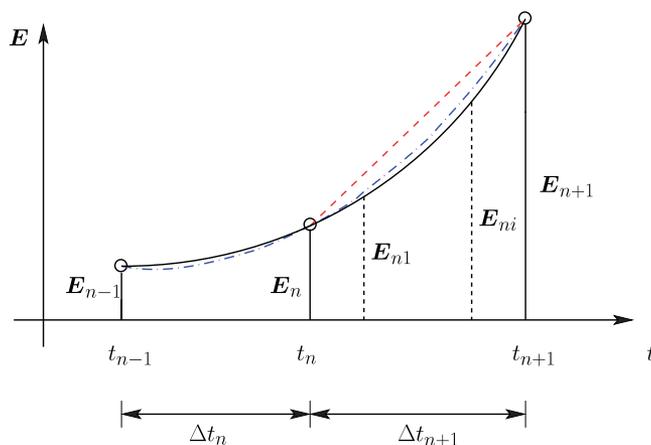

Figure 2. The exact, unknown strain path in time (full line) and its approximations via linear (dashed line) versus quadratic (dashed-dotted line) interpolation.

Introducing a new time scale $\tilde{t} := t - t_n$ in the time interval $t \in [t_n, t_{n+1}]$, the linear interpolation polynomial, supported by $(0, \boldsymbol{E}_n)$ and $(\Delta t, \boldsymbol{E}_{n+1})$, reads as

$$\boldsymbol{p}_1(\tilde{t}) = \boldsymbol{E}_n + \frac{\tilde{t}}{\Delta t}(\boldsymbol{E}_{n+1} - \boldsymbol{E}_n) \tag{23}$$

which leads to the approximation of $\boldsymbol{E}$ at time $t = t_i = t_n + c_i \Delta t$

$$\boldsymbol{E}(t = t_i) \approx \boldsymbol{p}_1(\tilde{t} = c_i \Delta t) = \boldsymbol{E}_n + c_i(\boldsymbol{E}_{n+1} - \boldsymbol{E}_n). \tag{24}$$

Under the assumption that the strains in time interval $t \in [t_n, t_{n+1}]$ and $\tilde{t} \in [0, \Delta t]$ are twice continuously differentiable, the interpolation error in the strains is

$$|\boldsymbol{E}(\tilde{t}) - \boldsymbol{p}_1(\tilde{t})| \leqslant \frac{M_2}{2}(\Delta t)^2 \tag{25}$$

with $M_2 = \max\{|\boldsymbol{E}''(\tilde{t})| : \tilde{t} \in [0, \Delta t]\}$, [30].

4.2. The novel ansatz: stage values via quadratic interpolation

Figure 2 aims to illustrate that for viscoelastic problems the exact strain path in time is non-linear in general. As a consequence, the strain path can be better approximated by means of quadratic interpolation compared with linear interpolation. Of course, this statement is based on the assumption that the strain path is smooth, which is justified for viscoelasticity but does not apply for elasto-plasticity in general.

The fact that the error for linear interpolation is of the order $\mathcal{O}(\Delta t^2)$ according to (25) suggests that this low-order approximation is a candidate to cause order reduction in viscoelastic stress computation when higher order methods, $p \geqslant 3$ are used. Based on this hypothesis we propose a quadratic interpolation polynomial $\boldsymbol{p}_2(\tilde{t})$, which is based on the data set $(-\Delta t, \boldsymbol{E}_{n-1})$, $(0, \boldsymbol{E}_n)$ and $(\Delta t, \boldsymbol{E}_{n+1})$.

Given that $\boldsymbol{E}(\tilde{t})$ is three times continuously differentiable in $[-\Delta t, \Delta t]$, it holds for the interpolation error

$$|\boldsymbol{E}(\tilde{t}) - \boldsymbol{p}_2(\tilde{t})| \leqslant \frac{M_3}{6}(\Delta t)^3 \tag{26}$$

with $M_3 = \max\{|\boldsymbol{E}'''(\tilde{t})| : \tilde{t} \in [-\Delta t, \Delta t]\}$, [30]. The interpolation polynomial $\boldsymbol{p}_2$ reads as

$$\boldsymbol{p}_2(\tilde{t}) = \frac{\boldsymbol{E}_{n-1} - 2\boldsymbol{E}_n + \boldsymbol{E}_{n+1}}{2(\Delta t)^2}\tilde{t}^2 + \frac{\boldsymbol{E}_{n+1} - \boldsymbol{E}_{n-1}}{2(\Delta t)}\tilde{t} + \boldsymbol{E}_n. \tag{27}$$

With (27) it follows for the stage values of E at $t = t_i = t_n + c_i \Delta t$ and $\tilde{t} = c_i \Delta t$

$$E(t = t_i) \approx p_2(\tilde{t} = c_i \Delta t) = \frac{c_i}{2}(c_i - 1)E_{n-1} + (1 - c_i^2)E_n + \frac{c_i}{2}(c_i + 1)E_{n+1}. \tag{28}$$

It is the core hypothesis of the present work that the order of convergence in stress computation directly depends on the approximation error in strain interpolation. We verify this hypothesis, first by a proof concerning the consistency order in Section 5, second by numerical means measuring the convergence order in Section 6.

4.3. Algorithmic consistent tangent moduli

The algorithmic consistent tangent moduli $\mathbb{C}_{n+1}^{v} = \partial S_{n+1} / \partial E_{n+1}$ are derived to achieve quadratic convergence in *global* equilibrium iterations using the Newton–Raphson algorithm. These moduli incorporate the chosen interpolation scheme for strain calculation at RK stages. Let $\mathbb{P}$ be the fourth-order deviatoric tensor,

$$\mathbb{P} := \mathbb{I}d - \tfrac{1}{3}\mathbf{1} \otimes \mathbf{1} \tag{29}$$

such that $\mathbb{P} : E = E^{D}$. Then, (7) can be rewritten as

$$S = \mathbb{C}_0 : E - 2\mu E^{v,D} \tag{30}$$

with

$$\mathbb{C}_0 = K_0(\mathbf{1} \otimes \mathbf{1}) + 2\mu_0 \mathbb{P}. \tag{31}$$

The algorithmic consistent tangent moduli are given by

$$\mathbb{C}_{n+1}^{v} = \frac{\partial S_{n+1}}{\partial E_{n+1}} = \mathbb{C}_0 - 2\mu \frac{\partial E_{n+1}^{v,D}}{\partial E_{n+1}}. \tag{32}$$

The term $\partial E_{n+1}^{v,D} / \partial E_{n+1}$ can be derived from equation system (21), if it is transformed into

$$\sum_{j=1}^{s} \left(\delta_{ij} + \frac{\Delta t}{\tau} a_{ij} \right) E_j^{v,D} = E_n^{v,D} + \frac{\Delta t}{\tau} \sum_{j=1}^{s} a_{ij} E_j^{D}, \qquad i = 1, \ldots, s. \tag{33}$$

The partial derivative of (33) with respect to E_{n+1} yields

$$\sum_{j=1}^{s} \left(\delta_{ij} + \frac{\Delta t}{\tau} a_{ij} \right) \frac{\partial E_j^{v,D}}{\partial E_{n+1}} = \frac{\Delta t}{\tau} \sum_{j=1}^{s} a_{ij} \frac{\partial E_j^{D}}{\partial E_{n+1}}, \quad i = 1, \ldots, s$$

$$= \frac{\Delta t}{\tau} \sum_{j=1}^{s} a_{ij} \bar{c}_j \mathbb{P} \tag{34}$$

where $\bar{c}_j = c_j$, if the stage values E_j^{D} are computed via linear interpolation and $\bar{c}_j = (c_j / 2)(c_j + 1)$ for the novel ansatz using quadratic interpolation, respectively.

As a final step, the linear equation system (34) for $\partial E_j^{v,D} / \partial E_{n+1}$ has to be solved to obtain $\partial E_{n+1}^{v,D} / \partial E_{n+1} = \partial E_s^{v,D} / \partial E_{n+1}$.

4.4. General format of the equations within the partitioned ansatz

As already explained, the key idea of the present contribution is to replace linear interpolation for the construction of stage values for u (or equally: E) by quadratic interpolation, thus increasing the order of the corresponding approximation error to third order. The ultimate aim of this procedure is to achieve consistency order three in time integration.

Equations (35) are the general form of problem sets, where the IVP for the ODE $\dot{z} = f(u, z)$, see (9), is solved by RK methods within the partitioned ansatz.

$$
\left.
\begin{aligned}
u_{n+1} &= P(z_{n+1}) \\
z_i &= z_n + \Delta t \sum_{j=1}^{s} a_{ij} f(u_j, z_j) \\
\text{lin.: } u_i &= u_n + c_i (u_{n+1} - u_n) \\
\text{quad.: } u_i &= \frac{c_i}{2}(c_i - 1) u_{n-1} + (1 - c_i^2) u_n + \frac{c_i}{2}(c_i + 1) u_{n+1} \\
z_{n+1} &= z_n + \Delta t \sum_{i=1}^{s} b_i f(u_i, z_i)
\end{aligned}
\right\} \quad i = 1, \ldots, s \quad (35)
$$

Remark

The compact expression $(35)_1$ in terms of the operator P shall be explained. For that aim we start out with the space-discrete format of the weak form of the balance of momentum. For notational convenience we choose here the geometrical linear case.

$$
0 = \int_{\Omega^h} B^{\mathrm{T}} \mathbb{C} B \, \mathrm{d}\Omega u + \int_{\partial \Omega_u^h} B^{\mathrm{T}} \mathbb{C} L \bar{u} \, \mathrm{d}\Omega - \int_{\Omega^h} B^{\mathrm{T}} \mathbb{C} \varepsilon^{\mathrm{v}} \, \mathrm{d}\Omega - \int_{\Omega^h} N^{\mathrm{T}} f \, \mathrm{d}\Omega - \int_{\partial \Omega_\sigma^h} N^{\mathrm{T}} \bar{\sigma} \, \mathrm{d}\Omega_\sigma \quad (36)
$$

$$
= K u - R(z) \quad (37)
$$

where we use in (37) the definition of the stiffness matrix K and $R(z)$ for the remainder integrals

$$
K := \int_{\Omega^h} B^{\mathrm{T}} \mathbb{C} B \, \mathrm{d}\Omega, \quad (38)
$$

$$
R := - \int_{\Omega^h} B^{\mathrm{T}} \mathbb{C} \varepsilon^{\mathrm{v}} \, \mathrm{d}\Omega - \int_{\Omega^h} N^{\mathrm{T}} f \, \mathrm{d}\Omega - \int_{\partial \Omega_\sigma^h} N^{\mathrm{T}} \bar{\sigma} \, \mathrm{d}\Omega_\sigma + \int_{\partial \Omega_u^h} B^{\mathrm{T}} \mathbb{C} L \bar{u} \, \mathrm{d}\Omega. \quad (39)
$$

Body forces in Ω are denoted by f, external loads on boundary $\partial \Omega_\sigma^h$ by $\bar{\sigma}$, prescribed displacements on boundary $\partial \Omega_u^h$ by $\bar{u}$. Finite element shape functions are denoted by N, and L is a differential operator calculating strains from displacements, $L := \mathrm{sym}(\nabla) = 1/2(\nabla + \nabla^{\mathrm{T}})$, hence $\varepsilon = L u$. All integrals in (36) are calculated by numerical quadrature, typically by the scheme of Gauss–Legendre. Here, we focus on the first integral in (39)

$$
\int_{\Omega^h} B^{\mathrm{T}} \mathbb{C} \varepsilon^{\mathrm{v}} \mathrm{d}\Omega \approx \sum_{i=1}^{n_{\mathrm{gauss}}} \alpha_i B^{\mathrm{T}}(\xi_i) \mathbb{C} \varepsilon^{\mathrm{v}}(\xi_i) \quad (40)
$$

with Gauss-points ξ_i and Gauss-weights α_i. Hence, (40) and (36) highlight the staggered/partitioned nature in computational inelasticity where the weak form is solved for u on a global level and the solution of the evolution equations for ε^{v} is solved on a local, Gauss-point level.

Discretization in time for (36) along with $\Delta t = t_{n+1} - t_n$ as time interval of interest yields the solution for the unknown displacements u at t_{n+1} according to $(35)_1$, where $P := K^{-1} R$.

5. PROOF, WHY ORDER REDUCTION OCCURS AND HOW TO OVERCOME IT

The main aim of this section is to analyze the influence of the polynomial order of strain interpolation in time on the consistency order of the internal variables z and of the displacements u. The result of the analysis is summarized in the following theorem.

Theorem 1
For the partitioned system $(35)_{1,2}$ the consistency order for u and z is 3, if quadratic interpolation for u_i according to $(35)_4$ is used and if the RK method fulfils the following conditions:

$$\sum_{i=1}^{s} b_i = 1, \quad \sum_{i=1}^{s} b_i c_i = \frac{1}{2}, \quad \sum_{i=1}^{s} b_i c_i^2 = \frac{1}{3}, \quad \sum_{i=1}^{s} b_i \sum_{j=1}^{s} a_{ij} c_j = \frac{1}{6}. \tag{41}$$

Linear interpolation for u_i according to $(35)_3$ results in a consistency order of 2 for u and z.

Proof
The proof is based on Taylor series expansions of the exact solution and the numerical solution employing either linear or quadratic interpolation of u. A comparison of coefficients in the two solutions for the requirement of consistency order 3 amounts to conditions (41) for a_{ij}, b_i and c_j of the RK scheme. Since these conditions are basic order conditions for RK schemes ($(41)_1$ for order 1, $(41)_{1,2}$ for order 2 and $(41)_{1-4}$ for order 3), they do not impose additional requirements but are fulfilled by the very construction of any third-order RK method like the considered Radau IIa scheme with $s=2$, see e.g. [26]. We start with the proof for u, after that, for z.

For convenience we slightly simplify the notation in the following. We write u for $\boldsymbol{u}$, f for $\boldsymbol{f}$, z for $\boldsymbol{z}$, P for $\boldsymbol{P}$ and use $h := \Delta t$ and for partial derivatives the notation $P_z := \partial_z P$.

5.1. Taylor series expansion of the numerical solution

Taylor expansion of the numerical solution u_{n+1} at z_n with time increment $h := \Delta t = t_{n+1} - t_n$ yields

$$u_{n+1} = P(z_{n+1}) = P\left(z_n + h \sum_{i=1}^{s} b_i f(u_i, z_i)\right) \tag{42}$$

$$= P(z_n) + h\left(\sum_{i=1}^{s} b_i f(u_i, z_i)\right) P_z(z_n)$$

$$+ \frac{h^2}{2}\left(\sum_{i=1}^{s} b_i f(u_i, z_i)\right)^2 P_{zz}(z_n) + \frac{h^3}{6}\left(\sum_{i=1}^{s} b_i f(u_i, z_i)\right)^3 P_{zzz}(z_n) + \mathcal{O}(h^4). \tag{43}$$

An expansion of the terms $f(u_i, z_i)$ centered at (u_n, z_n) follows next. The second term in (43) is expanded to $\mathcal{O}(h^3)$, the third term in (43) to $\mathcal{O}(h^2)$ and the fourth term in (43) to order $\mathcal{O}(h)$. In the following we introduce Taylor expansions for $f := f(u_n, z_n)$ and drop for convenience the argument for function evaluations at (u_n, z_n). With $\Delta u_i := u_i - u_n = \mathcal{O}(h)$ and $\Delta z_i := z_i - z_n = \mathcal{O}(h)$ it holds

$$f(u_i, z_i) = f + \mathcal{O}(h) \tag{44}$$

$$f(u_i, z_i) = f + \Delta u_i f_u + \Delta z_i f_z + \mathcal{O}(h^2) \tag{45}$$

$$f(u_i, z_i) = f + \Delta u_i f_u + \Delta z_i f_z + \frac{\Delta u_i^2}{2} f_{uu} + \Delta u_i \Delta z_i f_{uz} + \frac{\Delta z_i^2}{2} f_{zz} + \mathcal{O}(h^3). \tag{46}$$

For (45) and (46) we need Taylor expansions of Δu_i and Δz_i up to $\mathcal{O}(h^2)$ and $\mathcal{O}(h^3)$, respectively. With

$$u_{n+1} = P(z_{n+1}) = P(z_n + \Delta z_n)$$

$$= P + \Delta z_n P_z + \frac{\Delta z_n^2}{2} P_{zz} + \mathcal{O}(h^3) \tag{47}$$

$$u_{n-1} = P(z_{n-1}) = P(z_n - \Delta z_{n-1})$$

$$= P - \Delta z_{n-1} P_z + \frac{\Delta z_{n-1}^2}{2} P_{zz} + \mathcal{O}(h^3) \tag{48}$$

we apply quadratic interpolation of u according to $(35)_4$ and arrive for Δu_i at

$$\Delta u_i = u_i - u_n$$

$$= \frac{c_i}{2}(c_i - 1)u_{n-1} + (1 - c_i^2)u_n + \frac{c_i}{2}(c_i + 1)u_{n+1} - u_n$$

$$= \frac{c_i}{2}\left[c_i(u_{n+1} + u_{n-1}) + (u_{n+1} - u_{n-1})\right] - c_i^2 u_n$$

$$= \frac{c_i}{2}\left[((c_i + 1)\Delta z_n + (1 - c_i)\Delta z_{n-1})P_z + \frac{1}{2}\left((c_i + 1)\Delta z_n^2 + (c_i - 1)\Delta z_{n-1}^2\right)P_{zz}\right]$$

$$+ \mathcal{O}(h^3) \tag{49}$$

where we used (47) and (48) for the last identity in (49). Up to now, stage solutions refer to the current time interval $[t_n, t_{n+1}]$ and are denoted by index i. For the following, we introduce superscript n for stage variables of the current time interval $[t_n, t_{n+1}]$ in order to distinguish them from stage quantities in the previous time interval $[t_{n-1}, t_n]$, which are denoted by superscript $n-1$.

$$\Delta z_{n-1} := z_n - z_{n-1} = h \sum_{i=1}^{s} b_i f_i^{n-1}$$

$$= h \sum_{i=1}^{s} b_i \{f_i^n + \mathcal{O}(h)\}$$

$$= h \sum_{i=1}^{s} b_i f_i^n + \mathcal{O}(h^2) = \Delta z_n + \mathcal{O}(h^2) \tag{50}$$

Calculating the difference of Δz_{n-1} and Δz_n yields

$$\Delta^2 z_n := \Delta z_{n-1} - \Delta z_n = \mathcal{O}(h^2) \tag{51}$$

and for Δu_i it holds (again with only one single index for the stage values in the current time interval $[t_n, t_{n+1}]$)

$$\Delta u_i = c_i \left[\left(h\left(\sum_j b_j f_j\right) + \frac{1 - c_i}{2}\Delta^2 z_n\right)P_z + c_i \frac{h^2}{2}\left(\sum_j b_j f_j\right)^2 P_{zz}\right] + \mathcal{O}(h^3).$$

This result is quite similar to the expression which follows from linear interpolation. Introducing the notation

$$\tilde{c}_i := \begin{cases} 1 & \text{for linear interpolation} \\ c_i & \text{for quadratic interpolation} \end{cases}$$

allows to embrace both cases all at once, thus

$$\Delta u_i = c_i \left[\left(h\left(\sum_j b_j f_j\right) + \frac{1-\tilde{c}_i}{2}\Delta^2 z_n \right) P_z + \tilde{c}_i \frac{h^2}{2}\left(\sum_j b_j f_j\right)^2 P_{zz} \right] + \mathcal{O}(h^3). \tag{52}$$

For Δz_i it holds

$$\Delta z_i = z_i - z_n = h\sum_j a_{ij} f_j$$

$$= h\sum_j a_{ij}\{f + \Delta u_j f_u + \Delta z_j f_z + \mathcal{O}(h^2)\}$$

$$= h\sum_j a_{ij}\{f + \Delta u_j f_u + \Delta z_j f_z\} + \mathcal{O}(h^3). \tag{53}$$

Taylor series expansions up to $\mathcal{O}(h^2)$ read for Δu_i and for Δz_i

$$\Delta u_i = c_i h\left(\sum_j b_j\{f + \mathcal{O}(h)\}\right) P_z + \mathcal{O}(h^2)$$

$$= c_i h\left(\sum_j b_j\right) f P_z + \mathcal{O}(h^2) \tag{54}$$

$$\Delta z_i = h\sum_j a_{ij}\{f + \mathcal{O}(h)\} + \mathcal{O}(h^2)$$

$$= h\left(\sum_j a_{ij}\right) f + \mathcal{O}(h^2). \tag{55}$$

Series expansion to $\mathcal{O}(h^3)$ yields: (The term with $\Delta^2 z_n$ will be expanded at the end.)

$$\Delta u_i = c_i \left[h\left(\sum_j b_j\{f + \Delta u_j f_u + \Delta z_j f_z + \mathcal{O}(h^2)\}\right) P_z + \frac{1-\tilde{c}_i}{2}\Delta^2 z_n P_z \right.$$

$$\left. + \tilde{c}_i \frac{h^2}{2}\left(\sum_j b_j\{f + \mathcal{O}(h)\}\right)^2 P_{zz} \right] + \mathcal{O}(h^3)$$

$$= c_i \left[h\left(\sum_j b_j\left\{f + \left(c_j h\left(\sum_r b_r\right) f P_z + \mathcal{O}(h^2)\right) f_u + \left(h\left(\sum_r a_{jr}\right) f + \mathcal{O}(h^2)\right) f_z\right\}\right) P_z \right.$$

$$\left. + \frac{1-\tilde{c}_i}{2}\Delta^2 z_n P_z + \tilde{c}_i \frac{h^2}{2}\left(\sum_j b_j f\right)^2 P_{zz} \right] + \mathcal{O}(h^3)$$

$$= c_i \left[h\left(\sum_j b_j\left\{f + c_j h\left(\sum_r b_r\right) f f_u P_z + h\left(\sum_r a_{jr}\right) f f_z\right\}\right) P_z + \frac{1-\tilde{c}_i}{2}\Delta^2 z_n P_z \right.$$

$$\left. + \tilde{c}_i \frac{h^2}{2}\left(\sum_j b_j\right)^2 f^2 P_{zz} \right] + \mathcal{O}(h^3) \tag{56}$$

and

$$\Delta z_i = h \sum_j a_{ij} \left\{ f + \left(c_j h \left(\sum_r b_r \right) f P_z + \mathcal{O}(h^2) \right) f_u + \left(h \left(\sum_r a_{jr} \right) f + \mathcal{O}(h^2) \right) f_z \right\} + \mathcal{O}(h^3)$$

$$= h \sum_j a_{ij} \left\{ f + c_j h \left(\sum_r b_r \right) f f_u P_z + h \left(\sum_r a_{jr} \right) f f_z \right\} + \mathcal{O}(h^3). \tag{57}$$

Inserting the above expressions into the Taylor expansion of the numerical solution u_{n+1}, we finally arrive at

$$u_{n+1} = u_n$$

$$+ h \sum_i b_i \left\{ f + h \left[\sum_j a_{ij} \left\{ f + c_j h \left(\sum_r b_r \right) f f_u P_z + h \left(\sum_r a_{jr} \right) f f_z \right\} \right] f_z \right.$$

$$+ c_i \left[h \sum_j b_j \left\{ f + c_j h \left(\sum_r b_r \right) f f_u P_z + h \left(\sum_r a_{jr} \right) f f_z \right\} P_z + \tilde{c}_i \frac{h^2}{2} \left(\sum_r b_r \right) f^2 P_{zz} \right] f_u$$

$$+ c_i \left[\frac{1 - \tilde{c}_i}{2} \Delta^2 z_n P_z \right] f_u$$

$$+ c_i^2 \frac{h^2}{2} \left(\sum_r b_r \right)^2 f^2 f_{uu} P_z^2 + c_i h^2 \left(\sum_r b_r \right) \left(\sum_r a_{ir} \right) f^2 f_{uz} P_z + \frac{h^2}{2} \left(\sum_r a_{ir} \right)^2 f^2 f_{zz} \right\} P_z$$

$$+ \frac{h^2}{2} \left(\left(\sum_i b_i \right)^2 f^2 + 2h \left(\sum_i b_i \right) \left(\sum_i b_i c_i \right) \left(f^2 f_u P_z + f^2 f_z \right) \right) P_{zz}$$

$$+ \frac{h^3}{6} \left(\sum_r b_r \right)^3 f^3 P_{zzz} + \mathcal{O}(h^4). \tag{58}$$

5.2. Taylor series expansion of the exact solution

The Taylor series expansion of the exact solution up to $\mathcal{O}(h^4)$ reads as

$$u(t_n + h) = u + h \dot{u} + \frac{h^2}{2} \ddot{u} + \frac{h^3}{6} \dddot{u} + \mathcal{O}(h^4). \tag{59}$$

Applying the chain rule and the product rule yields the time derivatives of u in (59).

$$\dot{u} = P_z f \tag{60}$$

$$\ddot{u} = P_{zz} f^2 + P_z^2 f f_u + P_z f f_z \tag{61}$$

$$\dddot{u} = P_{zzz} f^3 + 4 P_{zz} P_z f^2 f_u + 3 P_{zz} f^2 f_z + 2 P_z^2 f f_u f_z + 2 P_z^2 f^2 f_{uz}$$

$$+ P_z^3 f f_u^2 + P_z^3 f^2 f_{uu} + P_z f f_z^2 + P_z f^2 f_{zz} \tag{62}$$

A comparison of coefficients in Taylor expansions (58) and (59) will result in conditions for the RK method to achieve consistency order 2 and order 3, respectively.

Conditions for consistency order 2. A comparison of coefficients of h-terms and h^2-terms yields the same conditions for consistency order 2 as in [12], since $\tilde{c}_i$ (stemming from quadratic

interpolation) merely appears in h^3-terms. Hence, the conditions stemming from a comparison of h- and h^2-terms read as

$$\sum_{i=1}^{s} b_i = 1 \quad \text{and} \quad \sum_{i=1}^{s} b_i c_i = \frac{1}{2}. \tag{63}$$

Conditions for consistency order 3. Next, a comparison of coefficients of h^3-terms is due. First of all, we list all terms which do not impose additional conditions—going beyond (63)—for the coefficients in the RK scheme. Then, we consider terms without $\tilde{c}_i$, which are equal for linear and quadratic interpolation and which impose additional conditions. Finally, we consider terms which contain $\tilde{c}_i$.

$$h^3 P_{zzz} f^3 : \quad \frac{1}{6} \left(\sum_i b_i \right)^3 = \frac{1}{6} \;\Rightarrow\; \sum_i b_i = 1 \quad \text{in agreement with (63)}$$

$$h^3 P_{zz} f^2 f_z : \quad \frac{1}{2} 2 \sum_i b_i \sum_i b_i c_i = \frac{3}{6} \;\Rightarrow\; \left(\sum_i b_i \right) \left(\sum_i b_i c_i \right) = \frac{1}{2} \quad \text{in agreement with (63)}$$

Additional conditions are generated by the following terms:

$$h^3 P_z^3 f^2 f_{uu} : \quad \left(\sum_i b_i c_i^2 \right) \frac{1}{2} \left(\sum_r b_r \right) = \frac{1}{6} \quad \text{with (63)} \Rightarrow \sum_i b_i c_i^2 = \frac{1}{3}$$

$$h^3 P_z^2 f^2 f_{uz} : \quad \sum_i b_i c_i \left(\sum_r b_r \right) \left(\sum_r a_{ir} \right) = \frac{2}{6} \quad \text{with (16), (63)} \Rightarrow \sum_i b_i c_i^2 = \frac{1}{3}$$

$$h^3 P_z f^2 f_{zz} : \quad \sum_i b_i \frac{1}{2} \left(\sum_r a_{ir} \right)^2 = \frac{1}{6} \quad \text{with (16)} \Rightarrow \sum_i b_i c_i^2 = \frac{1}{3}$$

$$h^3 P_z f f_z^2 : \quad \sum_i b_i \sum_j a_{ij} \sum_r a_{jr} = \frac{1}{6} \quad \text{with (16)} \Rightarrow \sum_i b_i \sum_j a_{ij} c_j = \frac{1}{6}$$

Summarizing, compared with (63), two additional conditions for a consistency order of 3 must be fulfilled by the RK method:

$$\sum_{i=1}^{s} b_i c_i^2 = \frac{1}{3} \quad \text{and} \quad \sum_{i=1}^{s} b_i \sum_{j=1}^{s} a_{ij} c_j = \frac{1}{6}. \tag{64}$$

These conditions are basic conditions for the construction of third-order RK methods, see [26], and therefore are in particular fulfilled by the coefficients of Radau IIa-schemes with $s \geqslant 2$, see the Butcher tableau on the right of Table I.

Next, terms containing $\tilde{c}_i$ are compared:

$$h^3 P_{zz} P_z f^2 f_u : \quad \frac{1}{2} \left(\sum_i b_i c_i \tilde{c}_i \right) + \frac{1}{2} 2 \left(\sum_i b_i \right) \left(\sum_i b_i c_i \right) = \frac{4}{6} \quad \text{with (63)} \Rightarrow \sum_i b_i c_i \tilde{c}_i = \frac{1}{3} \tag{65}$$

For the case of linear interpolation ($\tilde{c}_i = 1$), (65) is in contradiction to the second requirement in (63); for the case of quadratic interpolation ($\tilde{c}_i = c_i$); however, (65) coincides with the first condition for third order in (64).

In the case of linear interpolation, a comparison of coefficients of terms containing $h^3 P_z^3 f f_u^2$ and $h^3 P_z^2 f f_u f_z$ results in conditions which are in contradiction to (63) and to (64):

$$h^3 P_z^3 f f_u^2: \quad \sum_i b_i c_i \sum_j b_j c_j \sum_r b_r = \frac{1}{6} \Rightarrow \left(\sum_i b_i c_i\right)^2 = \frac{1}{6} \quad \text{in contradiction to (63)}_2$$

$$h^3 P_z^2 f f_u f_z: \quad \sum_i b_i \sum_j a_{ij} c_j \left(\sum_r b_r\right) + \sum_i b_i c_i \sum_j b_j \sum_r a_{jr} = \frac{2}{6}$$

$$\Rightarrow \sum_i b_i \sum_j a_{ij} c_j + \left(\sum_i b_i c_i\right)^2 = \frac{1}{3}$$

$$\Rightarrow \sum_i b_i \sum_j a_{ij} c_j = \frac{1}{12} \quad \text{in contradiction to (64)}_2$$

In order to examine the case of quadratic interpolation, the expressions $\Delta^2 z_n = \Delta z_{n-1} - \Delta z_n$ have to be expanded up to $\mathcal{O}(h^3)$.

$$\Delta z_n = z_{n+1} - z_n = h \sum_j b_j f_j$$

$$= h \sum_j b_j \left\{ f + \Delta u_j f_u + \Delta z_j f_z \right\} + \mathcal{O}(h^3)$$

$$= h \sum_j b_j \left\{ f + c_j h \left(\sum_r b_r\right) f f_u P_z + h \left(\sum_r a_{jr}\right) f f_z \right\} + \mathcal{O}(h^3) \tag{66}$$

Very similar to this result we obtain for Δz_{n-1}

$$\Delta z_{n-1} = h \sum_j b_j \left\{ f^{n-1} + c_j h \left(\sum_r b_r\right) f^{n-1} f_u^{n-1} P_z^{n-1} + h \left(\sum_r a_{jr}\right) f^{n-1} f_z^{n-1} \right\} + \mathcal{O}(h^3)$$

$$= h \sum_j b_j \left\{ f^{n-1} + c_j h \left(\sum_r b_r\right) f f_u P_z + h \left(\sum_r a_{jr}\right) f f_z \right\} + \mathcal{O}(h^3) \tag{67}$$

with

$$f^{n-1} = f + \mathcal{O}(h)$$
$$f_u^{n-1} = f_u + \mathcal{O}(h)$$
$$f_z^{n-1} = f_z + \mathcal{O}(h)$$
$$P_z^{n-1} = P_z + \mathcal{O}(h)$$

Next, the Taylor series of f^{n-1} up to order $\mathcal{O}(h^2)$ must be calculated. It holds

$$f^{n-1} = f - \Delta u_{n-1} f_u - \Delta z_{n-1} f_z + \mathcal{O}(h^2).$$

Inserting

$$\Delta z_{n-1} = h \sum_r b_r f^{n-1} = h \sum_r b_r \{f + \mathcal{O}(h)\} = h \left(\sum_r b_r\right) f + \mathcal{O}(h^2)$$

$$\Delta u_{n-1} = u_n - u_{n-1}$$

$$= [P(z_{n-1}) + \Delta z_{n-1} P_z(z_{n-1}) + \mathcal{O}(h^2)] - P(z_{n-1})$$

$$= \left(h \left(\sum_r b_r \right) f + \mathcal{O}(h^2) \right) (P_z + \mathcal{O}(h)) + \mathcal{O}(h^2)$$

$$= h \left(\sum_r b_r \right) f P_z + \mathcal{O}(h^2)$$

yields

$$f^{n-1} = f - h \left(\sum_r b_r \right) f f_u P_z - h \left(\sum_r b_r \right) f f_z + \mathcal{O}(h^2)$$

and with this result it follows for the difference

$$\Delta^2 z_n = h \sum_j b_j \left\{ -h \left(\sum_r b_r \right) f f_u P_z - h \left(\sum_r b_r \right) f f_z \right\} + \mathcal{O}(h^3). \tag{68}$$

A comparison of coefficients with $h^3 P_z^3 f f_u^2$ and $h^3 P_z^2 f f_u f_z$ for quadratic interpolation finally results in the following conditions.

$$h^3 P_z^3 f f_u^2 : \quad \left(\sum_i b_i c_i \right)^2 + \sum_i b_i c_i \frac{1-c_i}{2} \sum_j b_j \left(-\sum_r b_r \right) = \frac{1}{6}$$

$$\Rightarrow \left(\sum_i b_i c_i \right)^2 + \frac{1}{2} \left(\sum_i b_i c_i^2 - \sum_i b_i c_i \right) = \frac{1}{6} \tag{69}$$

$$h^3 P_z^2 f f_u f_z : \quad \sum_i b_i \sum_j a_{ij} c_j + \sum_i b_i c_i \frac{1-c_i}{2} \sum_j b_j \left(-\sum_r b_r \right) = \frac{2}{6}$$

$$\Rightarrow \sum_i b_i \sum_j a_{ij} c_j + \left(\sum_i b_i c_i \right)^2 + \frac{1}{2} \left(\sum_i b_i c_i^2 - \sum_i b_i c_i \right) = \frac{1}{3} \tag{70}$$

It can be easily verified that RK methods, which fulfil standard order 3 conditions, (63) and (64), equally fulfil conditions (69) and (70), which completes the proof for consistency order 3 of u for quadratic interpolation. $\square$

The proof for the consistency order of z is similar to that for u and is readily obtained by virtue of the preliminary results.

The Taylor series expansion of the numerical solution of $z_{n+1} = z_n + \Delta z_n$ reads as

$$z_{n+1} = z_n + h \sum_i b_i \left(f + \Delta u_i f_u + \Delta z_i f_z + \frac{\Delta u_i^2}{2} f_{uu} + \Delta u_i \Delta z_i f_{uz} + \frac{\Delta z_i^2}{2} f_{zz} + \mathcal{O}(h^3) \right). \tag{71}$$

Taylor series expansion of the exact solution of $z(t_n + h)$ yields

$$z(t_n + h) = z + h \dot{z} + \frac{h^2}{2} \ddot{z} + \frac{h^3}{6} \dddot{z} + \mathcal{O}(h^4). \tag{72}$$

Applying the chain rule and the product rule yields the time derivatives of z in (72)

$$\dot{z} = f \tag{73}$$

$$\ddot{z} = P_z f f_u + f f_z \tag{74}$$

$$\dddot{z} = P_z^2 f^2 f_{uu} + 2 P_z f^2 f_{uz} + f^2 f_{zz} + P_{zz} f^2 f_u + P_z^2 f f_u^2 + 2 P_z f f_u f_z + f f_z^2 \tag{75}$$

Inserting equations (54), (55), (56), (57) and (68) into (71) yields

$$
\begin{aligned}
z_{n+1} = z_n \\
+ h \sum_i b_i \Bigg\{ f + c_i \Bigg[h \Bigg(\sum_j b_j \Bigg\{ f + c_j h \Bigg(\sum_r b_r \Bigg) f f_u P_z + h \Bigg(\sum_r a_{jr} \Bigg) f f_z \Bigg\} \Bigg) P_z \\
+ \frac{1 - \tilde{c}_i}{2} h \sum_j b_j \Bigg\{ -h \Bigg(\sum_r b_r \Bigg) f f_u P_z - h \Bigg(\sum_r b_r \Bigg) f f_z \Bigg\} + \tilde{c}_i \frac{h^2}{2} \Bigg(\sum_j b_j \Bigg)^2 f^2 P_{zz} \Bigg] f_u \\
+ h \sum_j a_{ij} \Bigg\{ f + c_j h \Bigg(\sum_r b_r \Bigg) f f_u P_z + h \Bigg(\sum_r a_{jr} \Bigg) f f_z \Bigg\} f_z + \frac{1}{2} \Bigg(c_i h \Bigg(\sum_j b_j \Bigg) f P_z \Bigg)^2 f_{uu} \\
+ \frac{1}{2} \Bigg(h \Bigg(\sum_j a_{ij} \Bigg) f \Bigg)^2 f_{zz} + \Bigg(c_i h \Bigg(\sum_j b_j \Bigg) f P_z \Bigg) \Bigg(h \Bigg(\sum_j a_{ij} \Bigg) f \Bigg) f_{uz} \Bigg\} + \mathcal{O}(h^4).
\end{aligned} \tag{76}
$$

A comparison of coefficients in (76) with coefficients in (72) yields for the

- h-term:

$$
f: \quad \sum_i b_i = 1 \tag{77}
$$

- h^2-terms:

$$
P_z f f_u: \quad \sum_i b_i c_i \sum_j b_j = \frac{1}{2} \tag{78}
$$

$$
f f_z: \quad \sum_i b_i \sum_j a_{ij} = \frac{1}{2}
$$

$$
\Rightarrow \sum_i b_i c_i = \frac{1}{2} \tag{79}
$$

- h^3-terms:

$$
P_z^2 f^2 f_{uu}: \quad \sum_i b_i c_i^2 \Bigg(\sum_j b_j \Bigg)^2 = \frac{1}{3} \tag{80}
$$

$$
P_z f^2 f_{uz}: \quad \sum_i b_i c_i \sum_j b_j \sum_j a_{ij} = \frac{1}{3}
$$

$$
\Rightarrow \sum_i b_i c_i^2 \sum_j b_j = \frac{1}{3} \tag{81}
$$

$$
f^2 f_{zz}: \quad \sum_i b_i \Bigg(\sum_j a_{ij} \Bigg)^2 = \frac{1}{3}
$$

$$
\Rightarrow \sum_i b_i c_i^2 = \frac{1}{3} \tag{82}
$$

$$
P_{zz} f^2 f_u: \quad \sum_i b_i c_i \tilde{c}_i \Bigg(\sum_j b_j \Bigg)^2 = \frac{1}{3} \tag{83}
$$

$$P_z^2 ff_u^2: \quad \sum_i b_i c_i \sum_j b_j c_j \sum_r b_r - \sum_i b_i c_i \left(\frac{1-\tilde{c}_i}{2} \right) \sum_j b_j \sum_r b_r = \frac{1}{6}$$

$$\Rightarrow \left(\sum_i b_i c_i \right)^2 \left(\sum_r b_r \right) - \left(\sum b_i c_i (\frac{1-\tilde{c}_i}{2}) \right) \left(\sum_r b_r \right)^2 = \frac{1}{6} \tag{84}$$

$$ff_z^2: \quad \sum_i b_i \sum_j a_{ij} \sum_r a_{jr} = \frac{1}{6}$$

$$\Rightarrow \sum_i b_i \sum_j a_{ij} c_j = \frac{1}{6} \tag{85}$$

$$P_z ff_u f_z: \quad \sum_i b_i c_i \sum_j b_j \sum_r a_{jr} + \sum_i b_i \sum_j a_{ij} c_j \sum_r b_r + \sum_i b_i c_i \left(\frac{1-\tilde{c}_i}{2} \right) \sum_j b_j (-1) \sum_r b_r = \frac{1}{3}$$

$$\Rightarrow \left(\sum_i b_i c_i \right)^2 + \sum_i b_i \sum_j a_{ij} c_j \sum_r b_r - \sum_i b_i c_i \left(\frac{1-\tilde{c}_i}{2} \right) \left(\sum_r b_r \right)^2 = \frac{1}{3} \tag{86}$$

It can be readily verified that conditions (77)–(86) are fulfilled by RK-standard order conditions $(41)_{1-4}$. For conditions (83), (84) and (86) however, this is true, if $\tilde{c}_i = c_i$ holds, i.e. if quadratic interpolation of strains is applied. Hence, in that case z exhibits consistency order 3. In contrast, consistency order 3 cannot be obtained for z, if $\tilde{c}_i = 1$, i.e. if linear interpolation is used. □

Several conclusions and remarks are due.

 (i) Since the total strain tensor E is derived from displacements u by differentiation with respect to spatial variables, E exhibits the same consistency order as u in time; here, for quadratic interpolation, it is of order 3. As a consequence of this fact and the statement of Theorem 1, that the differential variable z, here: viscous strain E^v exhibits consistency order 3, it can be concluded from Equations (3) and (4) that equilibrium stress S^{eq} as well as the overstress S^{ov} equally exhibit order 3.

 (ii) The above proof deals with the consistency order and therefore addresses the local error in time. The conclusion from local consistency order to global convergence order must be based on stability considerations. Here we have shown that the use of linear interpolation for the strains implies a reduction in consistency order to 2. As a consequence, a convergence order beyond 2 cannot be expected. For quadratic interpolation however, consistency order 3 has been shown, which is necessary for convergence order 3. It is left to the numerical assessment in Section 6, whether convergence order 3 for the considered ODE of viscoelasticity is obtained.

 (iii) The proof is based on problems of the format (10), which is an ODE. For the broad class of computational inelasticity this applies to constitutive equations either of (rate dependent) viscoelasticity or to models of elasto-plasticity *without* yield surfaces as an algebraic constraint like the model of Chan–Bodmer–Lindholm etc. For models of (rate-independent) elasto-plasticity entailing a yield surface and thus forming a DAE this proof does not apply.

6. NUMERICAL ASSESSMENT

In this section, the third-order (two-stage) Radau IIa scheme is used to investigate the influence of quadratic versus linear interpolation of strain on the convergence order. Hence, we verify by numerical means whether the convergence order of the global errors is the same as the proven consistency order of the local errors.

Table II. Uni- and biaxial stretch: viscoelastic material parameters.

K_∞ (N/mm^2)	μ_∞ (N/mm^2)	μ (N/mm^2)	η (N s/mm^2)
10 000	6.598	28.03	0.075

In order to achieve reliable conclusions concerning the validity of our proposal we carry out simulations for a variety of test sets: (i) uniaxial and biaxial stretch, i.e. homogeneous deformations, (ii) radial contraction of an annulus, (iii) stretching of a rectangular plate with a hole.

Since the predictions concerning the consistency order apply to *primary* quantities such as displacements $\boldsymbol{u}$ and total strains $\boldsymbol{E}$ but also to *derived* quantities such as viscoelastic strain $\boldsymbol{E}^{\mathrm{v}}$ and stress $\boldsymbol{S}$, we carry out the convergence analysis for each of these quantities in problem sets (ii) and (iii).

For all test sets an accurate reference solution $\boldsymbol{X}^{ex}$ for tensor $\boldsymbol{X}$ with $\boldsymbol{X} \in \{\boldsymbol{E}, \boldsymbol{E}^{\mathrm{v}}, \boldsymbol{S}\}$ is calculated by *numerical overkill* using a very small time step size where the accuracy of the results is in the range of machine precision. Based on this reference solution, a relative, global error for finite time step sizes is calculated according to

$$e(\boldsymbol{X}) = \frac{1}{N_{\mathrm{el}} \cdot N_{\mathrm{gauss}}} \sum_{i=1}^{N_{\mathrm{el}}} \sum_{j=1}^{N_{\mathrm{gauss}}} \frac{||\boldsymbol{X}^{(ij)}(\Delta t) - \boldsymbol{X}^{(ij)ex}||}{||\boldsymbol{X}^{(ij)ex}||}, \tag{87}$$

where $\boldsymbol{X}(\Delta t)$ is the tensor for a time step size Δt, N_{el} is the number of elements in the domain and N_{gauss} is the number of Gauss-points per element.

In the following, the relative error versus the time step size Δt is displayed in double logarithmic scaling. For uniform convergence, the mean order of convergence will be calculated by means of linear regression.

The Radau IIa scheme has been implemented into an eight-node volume element within an extended version of FEAP, a general purpose finite element code [31]. The implementation of the three-parameter viscoelastic material model of Section 2 is simplified employing implicit RK methods, since the generally non-linear set of equations (21) for the stage solutions becomes linear in this special case.

Remarks

In viscoelasticity—opposed to elasto-plasticity—loaded structures always exhibit nonzero inelastic strains from the very first loading step of the deformation history, such that evolution equations are to be integrated from the very beginning.

Note that the first time step for quadratic interpolation must rely on linear interpolation using t_n and t_{n+1} data, since t_{n-1} data are not available at that stage.

6.1. Uni- and biaxial stretch

We study the relaxation of a viscoelastic cube subject to either uni- or biaxial stretch. The cube is discretized by one eight-node brick-type element of side length L. Material parameters are given in Table II. Displacement boundary conditions at $Y = L/2$, for biaxial stretch also at $X = -L/2$, are chosen to avoid rigid body motions and to ensure a homogeneous deformation state. The displacement parameter is set to $u = 0.003\,L$ at $t = 0$ s, then held fixed for $t > 0$ s, see Figure 3.

The temporal evolution of nonzero stress components is depicted in Figure 4. Strain components that are not directly given by prescribed displacements exhibit qualitatively the same behavior as the stress components in Figure 4; they relax from a maximum value—corresponding to the maximum overstress—to the equilibrium values, see Figure 5. The non-linear strain path in time seems to be well suited to check the convergence behavior of time integration, when linear or quadratic interpolation is used. For the convergence analysis we employ eight different time step sizes Δt. The error is calculated at $t = 0.005$ s and at $t = 0.01$ s. The reference solution is calculated by Radau
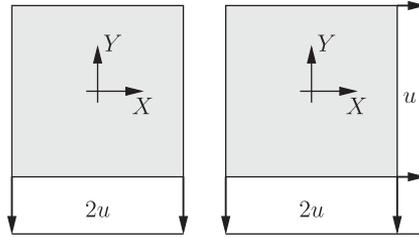
Figure 3. Displacement control for the relaxation of a cube subject to
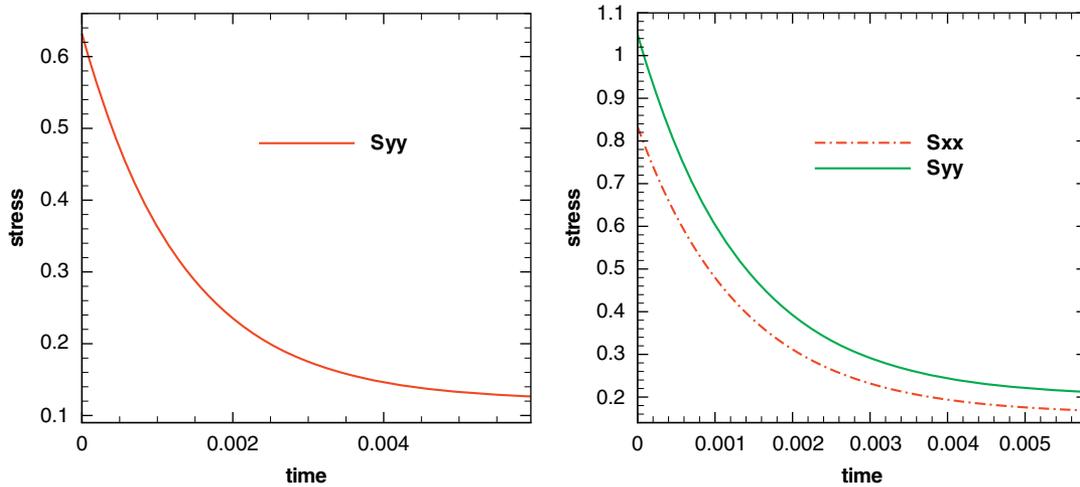(left:) uni- and (right:) biaxial stretch.

Figure 4. Relaxation of nonzero stress components in (left:) uniaxial and (right:) biaxial stretch.
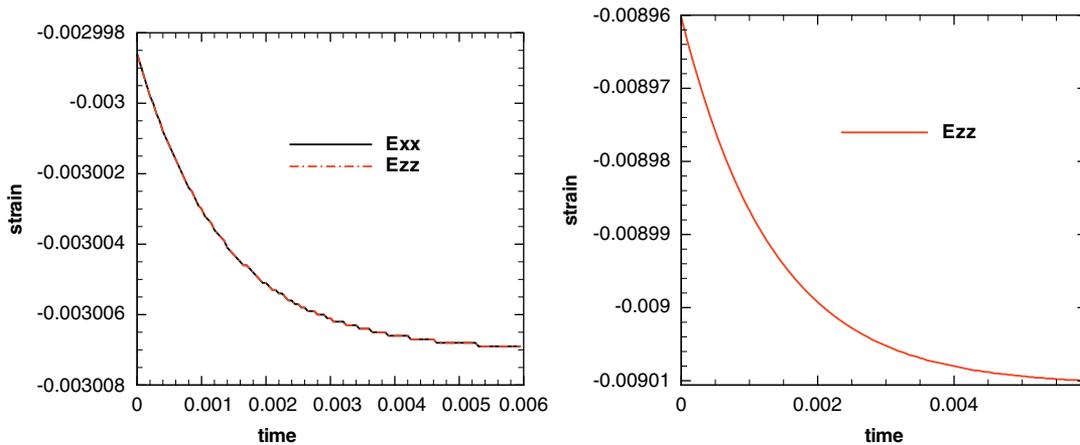

Figure 5. Nonzero strain components in the tension tests exhibit a non-linear relaxation behavior.

IIa along with quadratic interpolation and with time step size $\Delta t = 5.0E{-}07$. The considered time step sizes are $\Delta t = \{0.0001, 0.000125, 0.00025, 0.0005, 0.001, 0.00125, 0.0025, 0.005\}$s.

At $t = 0.005$ s and at $t = 0.01$ s (in uni- as well as in biaxial stretch) Radau IIa achieves third order for both, linear interpolation as well as quadratic interpolation, Tables III, IV and Figures 6 and 7. There is no order reduction.

Table III. Relaxation of a cube in uniaxial stretch: order of convergence of $e(S)$ for different methods.

Method	Abbrev.	$t = 0.005\,s$	$t = 0.01\,s$
Backward Euler	E	1.04	1.40
Radau IIa, $s = 2$, linear interpolation	R2l	2.99	2.83
Radau IIa, $s = 2$, quadratic interpolation	R2q	2.97	2.80

Table IV. Relaxation of a cube in biaxial stretch: order of convergence of $e(S)$ for different methods.

Method	Abbrev.	$t = 0.005\,s$	$t = 0.01\,s$
Backward Euler	E	1.03	1.40
Radau IIa, $s = 2$, linear interpolation	R2l	3.05	2.88
Radau IIa, $s = 2$, quadratic interpolation	R2q	2.97	2.76
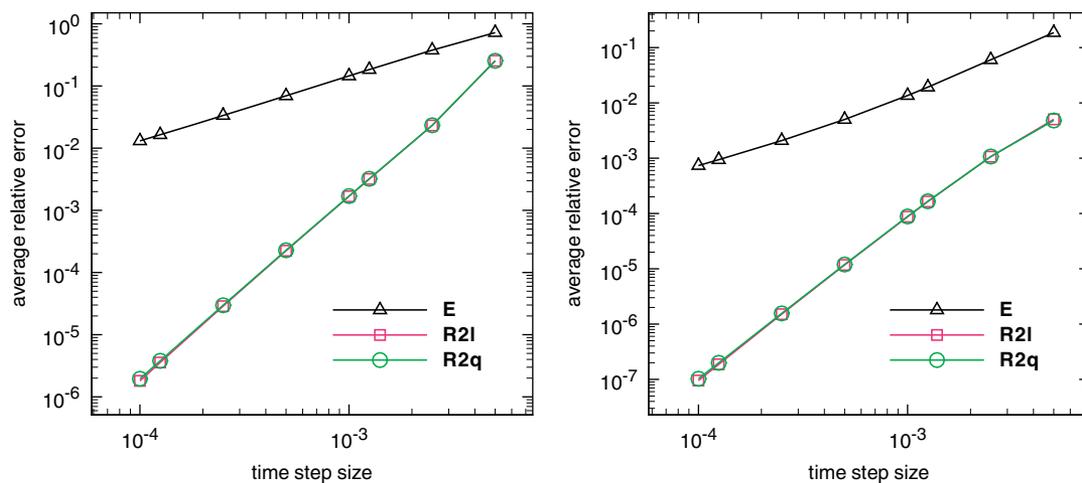
Figure 6. Uniaxial stretch: convergence diagrams for $e(S)$ at $t = 0.005\,s$ (left) and at $t = 0.01\,s$ (right) for E:backward Euler, R2l:Radau IIa with linear interpolation, R2q:Radau IIa with quadratic interpolation. $\Delta t_{max} = 0.005\,s$, $\Delta t_{min} = 0.0001\,s$.

The reason for the full convergence order using linear interpolation might be due to the fact that the strain path to be interpolated is not strongly non-linear, such that linear interpolation is sufficient. Remarkably, even linear backward Euler exhibits super-linear convergence, most notably at $t = 0.01\,s$. At this stage, we anticipate the results of all further tests in this paper; for linear interpolation Radau IIa suffers from order reduction to second-order convergence. This fact strongly suggests that homogeneous deformation states like uni- or biaxial stretch are not sufficient to arrive at general and reliable conclusions concerning the convergence behavior of time integration algorithms.

6.2. Radial contraction of an annulus

In the second test set we consider the radial contraction of an annulus, which exhibits radii $r_i = 20\,mm$, $r_o = 40\,mm$ and thickness $t = 1\,mm$ as displayed in the left of Figure 8. Two symmetry planes are exploited in the simulation, such that the simulation is performed at a quarter system. As shown in the right of Figure 8, the quarter system is discretized by 10 elements in circumferential direction, 10 elements in radial direction and one element over the thickness.

At $t = 0\,s$ the inner rim is radially pulled inwards by displacement $\Delta u_r = 1\,mm$ and then, for $t > 0\,s$, is kept fixed. The corresponding maximum strain amounts to 0.009. After that, the viscoelastic structure with material parameters according to Table II relaxes into its equilibrium configuration.
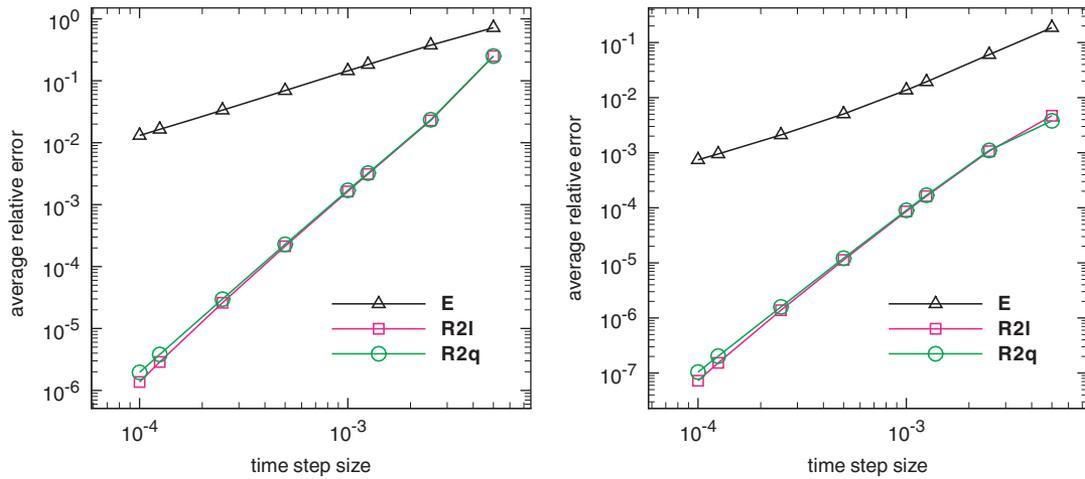
Figure 7. Biaxial stretch: convergence diagrams for $e(\boldsymbol{S})$ at $t = 0.005\,\mathrm{s}$ (left) and at $t = 0.01\,\mathrm{s}$ (right) for E-backward Euler, R2l-Radau IIa with linear interpolation, R2q-Radau IIa with quadratic interpolation. $\Delta t_{\max} = 0.005\,\mathrm{s}$, $\Delta t_{\min} = 0.0001\,\mathrm{s}$.
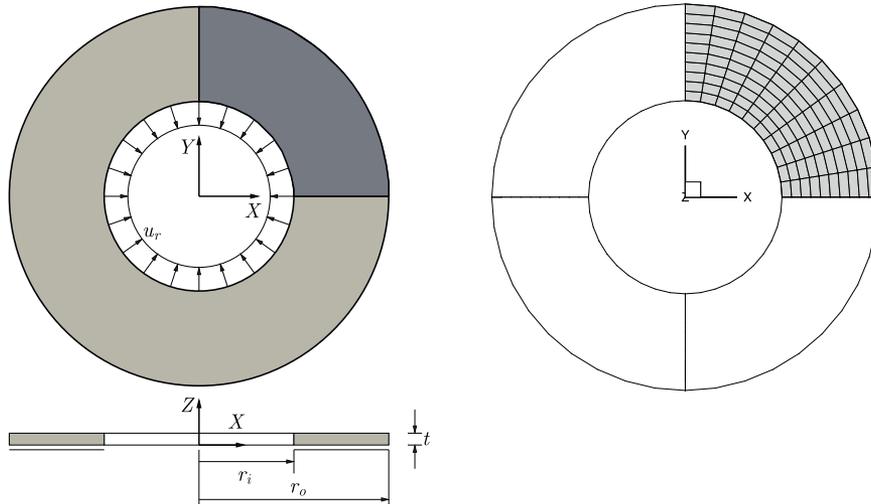

Figure 8. Radial contraction of an annulus by displacement control for u_r applied to the inner rim. Left: geometry and loading, right: finite element mesh.

Again, the calculation of relaxation is geometrically non-linear. The time step sizes are $\Delta t = \{0.0001, 0.000125, 0.00025, 0.0005, 0.001, 0.00125, 0.0025\}\mathrm{s}$. The reference solution is calculated by Radau IIa along with quadratic interpolation using a time step size of $\Delta t = 5.0\mathrm{E}{-}07$.

Figure 9 and Table V reveal that Radau IIa with linear interpolation, R2l, exhibits order reduction resulting in quadratic convergence, whereas Radau IIa with quadratic interpolation, R2q, achieves full order of convergence, order 3.

The numerical results concerning the convergence order are in agreement with the theoretical predictions concerning the consistency order. This is true for all analyzed quantities, for total strains $\boldsymbol{E}$, for viscoelastic strains $\boldsymbol{E}^{\mathrm{v}}$ and for the stresses $\boldsymbol{S}$.

Figure 9 indicates that the error of backward Euler employing the smallest time step considered, $\Delta t_{\min} = 0.0001\,\mathrm{s}$, is in the range of the error of Radau IIa for a time step which is 25 times larger than $\Delta t_{\min}$.

Finally, the overall performance of the time integration methods shall be measured and compared by means of the accuracy of stresses versus the total computation time. The diagrams in Figure 10
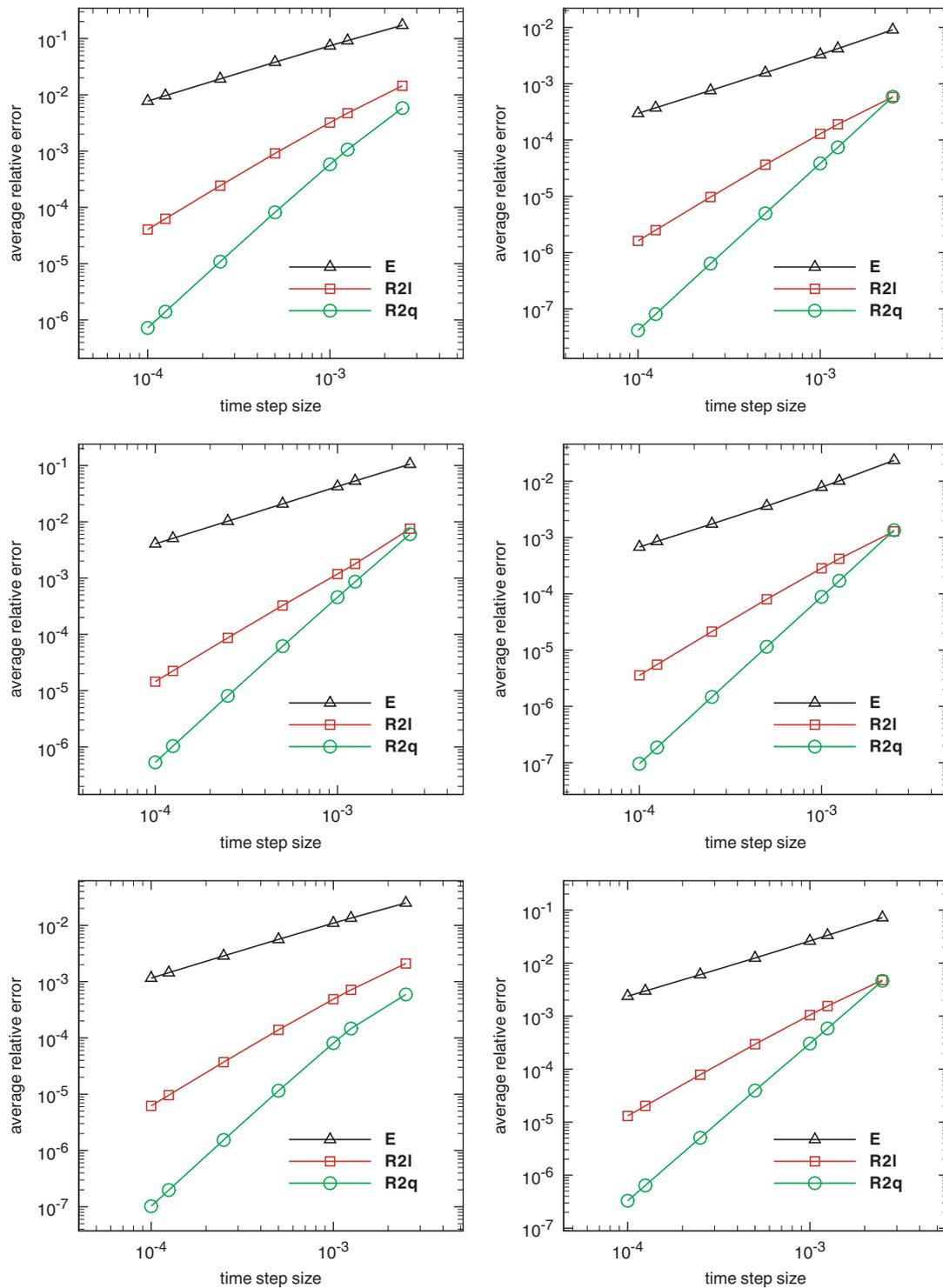
Figure 9. Radial contraction of an annulus: error versus time step-size evaluated at $t = 0.005\,\mathrm{s}$ (left column) and at $t = 0.01\,\mathrm{s}$ (right column) for different time integrators, $\Delta t_{\max} = 0.0025\,\mathrm{s}$, $\Delta t_{\min} = 0.0001\,\mathrm{s}$. First row: $e(\boldsymbol{E})$, second row: $e(\boldsymbol{E}^{\mathrm{v}})$, third row: $e(\boldsymbol{S})$.

Table V. Radial contraction of an annulus: order of convergence for different methods.

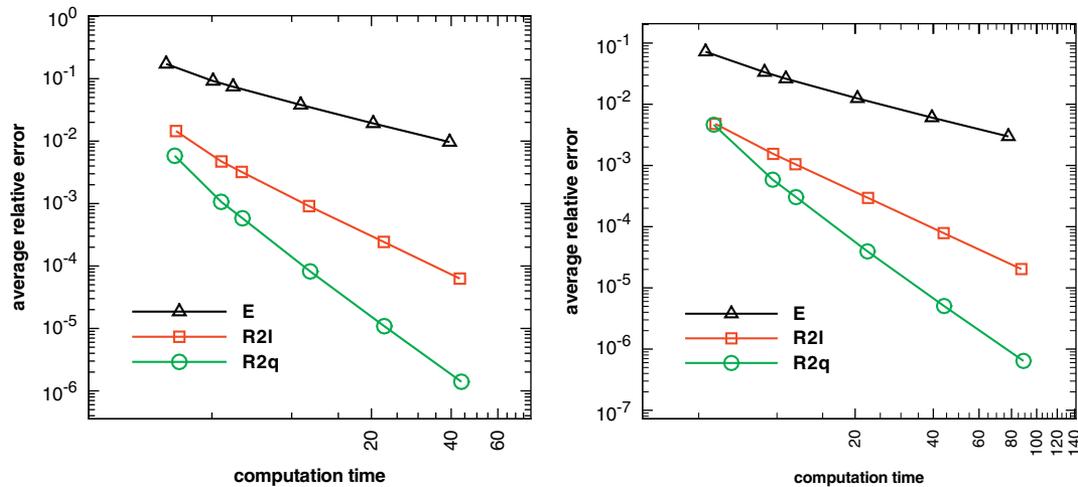| Error	$e(\boldsymbol{E})$		$e(\boldsymbol{E}^{\mathrm{v}})$		$e(\boldsymbol{S})$	
	$t = 0.005\,\mathrm{s}$	$t = 0.01\,\mathrm{s}$	$t = 0.005\,\mathrm{s}$	$t = 0.01\,\mathrm{s}$	$t = 0.005\,\mathrm{s}$	$t = 0.01\,\mathrm{s}$
E	0.96	1.06	1.02	1.09	0.97	1.06
R2l	1.83	1.85	1.93	1.85	1.85	1.85
R2q	2.76	2.97	2.91	2.97	2.83	2.96

Figure 10. Biaxial stretch: error $e(\boldsymbol{S})$ versus computation time (s); evaluation at $t = 0.005\,\mathrm{s}$ (left) and at $t = 0.01\,\mathrm{s}$ (right) for E-backward Euler, R2l-Radau IIa with linear interpolation, R2q-Radau IIa with quadratic interpolation.

display the results for time step sizes of $\Delta t = \{0.0001, 0.000125, 0.00025, 0.0005, 0.001, 0.00125, 0.0025\}\,\mathrm{s}$. It turns out that Radau IIa is significantly faster than backward Euler to obtain a defined accuracy, which is already visible for the version suffering of order reduction, but becomes even more prominent for Radau IIa with full order of convergence.

Furthermore, the diagrams in Figure 10 show that the overall computation time for a certain time step size is roughly the same for all methods, in particular, that Radau IIa is somewhat more expensive than backward Euler, but the difference between linear and quadratic interpolation for Radau IIa is negligible.

6.3. Tension strip with hole

In the present example we consider the creep of a quadratic strip with a hole. The geometry of the structure is given by $40 \times 40 \times 1$ mm for the length l, width l and thickness t, respectively. The plate exhibits a hole of radius $r = 3$ mm. The traction load $\bar{\sigma} = 0.1$ (N/mm) is applied at $t = 0$ s and then kept fixed for $t > 0$ s in geometrically non-linear calculations.

The strip is supported in its midsurface in thickness-direction, and supported orthogonal to the symmetry planes in the width direction and in the length direction, respectively. Exploiting three symmetry planes allows a simulation of one-eighth system, the resulting structure is discretized by $30 \times 20 \times 2$ elements, see Figure 11.

The three-parameter viscoelastic material law of Section 2 is used along with the same material parameters as in the former two examples, see Table II. Time step sizes considered in the simulations are $\Delta t = \{0.0003, 0.000375, 0.00075, 0.0015, 0.003, 0.00375\}\,\mathrm{s}$, reference values are calculated using Radau IIa with quadratic interpolation for $\Delta t = 5.0\mathrm{E}{-}07$.

Similar to the previous example, time integration applying the two-stage Radau IIa-method along with linear interpolation obtains not more than second order. For quadratic interpolation of
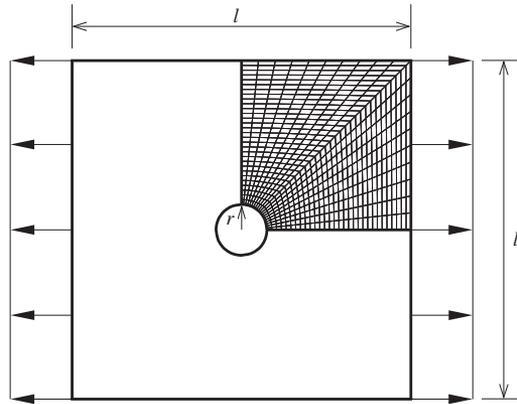
Figure 11. FE-model of the quadratic tension strip with a hole, loading, symmetry conditions applied.

Table VI. Tension strip with hole: order of convergence for different methods.

	$e(\boldsymbol{E})$		$e(\boldsymbol{E}^{\mathrm{v}})$		$e(\boldsymbol{S})$	
Error	$t=0.015\,\mathrm{s}$	$t=0.03\,\mathrm{s}$	$t=0.015\,\mathrm{s}$	$t=0.03\,\mathrm{s}$	$t=0.015\,\mathrm{s}$	$t=0.03\,\mathrm{s}$
E	0.97	1.06	0.97	1.06	0.99	1.13
R2l	1.83	1.86	1.83	1.86	1.83	1.87
R2q	2.62	2.90	2.62	2.90	2.85	2.94

the strain tensor however, results close to third order are achieved. This is true for all analyzed quantities, $\boldsymbol{E}$, $\boldsymbol{E}^{\mathrm{v}}$ and $\boldsymbol{S}$, see Table VI and Figure 12. These results are in agreement with the analytical predictions of Theorem 1.

Moreover, the diagrams in Figure 12 show that the error of backward Euler employing the smallest time step considered, $\Delta t_{\min}=0.0003\,\mathrm{s}$, is still larger than the error of Radau IIa for a time step which is 12 times larger than $\Delta t_{\min}$.

For a fair comparison of the methods their performance shall be compared, i.e. their accuracy versus the numerical cost in terms of the overall computation time. The diagrams in Figure 13 underpin that Radau IIa is significantly faster than linear backward Euler to achieve a defined accuracy.

7. SUMMARY AND CONCLUSIONS

In this paper we have analyzed the problem of order reduction in computational inelasticity, when evolution equations exhibit the format of an ODE as it is the case for the broad class of viscoelastic material models. For the classical, staggered structure of finite element algorithms dealing with inelasticity we have identified the reason and have proposed a solution to the problem by simple and effective means. The main findings shall be summarized:

(i) The reason for order reduction in computational viscoelasticity within the classical structure of finite element algorithms is a low-order (typically: linear) approximation of the displacement/strain path in time for the calculation of stage values in the RK scheme.

(ii) For RK methods with $p\geqslant 3$ we have proven that quadratic interpolation of the displacements/strains for the construction of stage values within the considered time interval is a sufficient condition to achieve a consistency order of 3. Here, we have chosen t_n, t_{n+1}, and additionally, t_{n-1} data for the construction of a quadratic polynomial. Complementarily, we have proven that standard linear approximation inevitably leads to a reduced consistency order of 2. These order statements apply for different quantities, for displacements/total strains, for viscoelastic strains and for stresses.
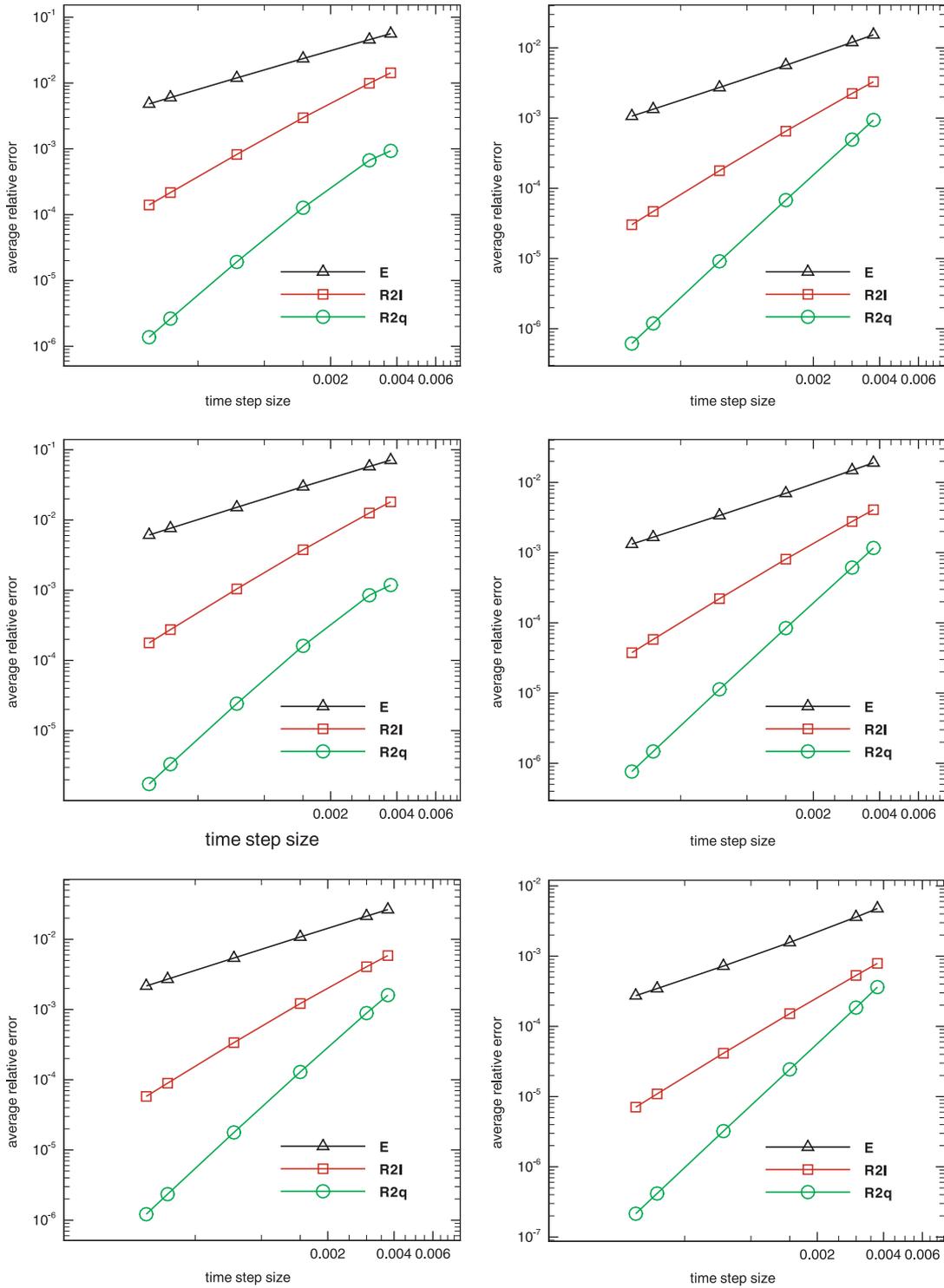
Figure 12. Tension strip with hole: error versus time step size evaluated at $t = 0.015\,\text{s}$ (left column) and for $t = 0.03\,\text{s}$ (right column) for different time integrators, $\Delta t_{\max} = 0.00375\,\text{s}$, $\Delta t_{\min} = 0.0003\,\text{s}$. First row: $e(\boldsymbol{E})$, second row: $e(\boldsymbol{E}^{\mathrm{v}})$, third row: $e(\boldsymbol{S})$.
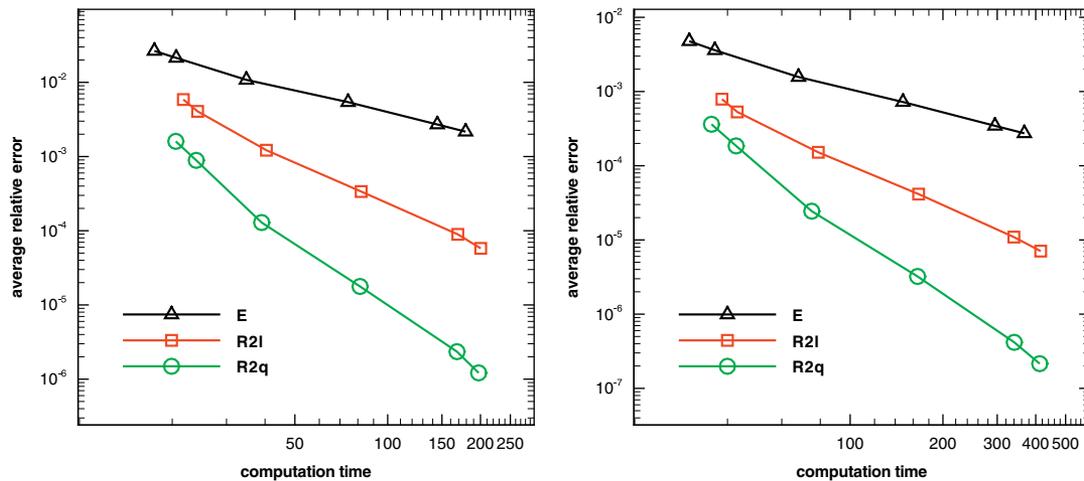
Figure 13. Tension strip with hole: error $e(S)$ versus computation time (s); evaluation at $t = 0.015$ s (left) and at $t = 0.03$ s (right) for E-backward Euler, R2l-Radau IIa with linear interpolation, R2q-Radau IIa with quadratic interpolation.

(iii) Numerical tests have been carried out applying a fully implicit, two-stage, third-order RK-version of Radau IIa class. For the newly proposed quadratic interpolation scheme all simulations result in full order of convergence for displacements/total strains, viscoelastic strains and stresses. For linear interpolation we observed an order reduction to second order for the same quantities. Hence, the numerical results concerning the convergence order of the global error are in agreement with the theoretical predictions for the consistency order, i.e. the local error.

(iv) Compared with standard backward Euler, a main advantage of the present RK integration algorithm is its considerable speed-up of simulations to achieve the same level of accuracy, which has been shown in simulations for viscoelastic structures subject to creep as well as relaxation.

(v) Another key advantage of the present approach is that it neatly fits into the standard finite element approach to deal with inelastic constitutive models, i.e. the staggered solution algorithm where the weak form of the BVP is solved on a global level and the IVP is solved on the local level of quadrature points. As a consequence, the present approach can easily be implemented into finite element codes. The additional computational overhead is restricted to history data at time t_{n-1}, which equals the number of the strain components along with eventually additional variables depending on the material model.

In conclusion, we believe that the present analysis of the reason for order reduction and its solution opens the door to the usage of accurate and efficient higher order RK methods in computational viscoelasticity while maintaining the staggered algorithmic structure as realized in lots of research and commercial finite element codes.

REFERENCES

1. Simó JC. Modelling and simulation of plasticity. In *Handbook of Numerical Analysis*, Ciarlet PG, Lions JL (eds). Numerical Methods for Solids (Part 3), vol. 6. Elsevier: Amsterdam, 1998; 183–499.
2. Ortiz M, Popov EP. Accuracy and stability of integration algorithms for elastoplastic constitutive relations. *International Journal for Numerical Methods in Engineering* 1985; **21**:1561–1576.

3. Govindjee S, Simó JC. Non-linear B-stability and symmetry preserving return mapping algorithms for plasticity and viscoplasticity. *International Journal for Numerical Methods in Engineering* 1991; **31**:151–176.
4. Wieners C. Multigrid methods for Prandtl–Reuß plasticity. *Numerical Linear Algebra with Applications* 1999; **6**:457–478.
5. Ellsiepen P. Zeit—und ortsadaptive Verfahren angewandt auf Mehrphasenprobleme poröser Medien (in German). *Ph.D. Thesis*, Universität Stuttgart, 1999.
6. Artioli E, Auricchio F, Beirão da Veiga L. Second-order accurate integration algorithms for von-Mises plasticity with a nonlinear kinematic hardening mechanism. *Computer Methods in Applied Mechanics and Engineering* 2007; **196**:1827–1846.
7. Papadopoulos P, Taylor RL. On the application of multi-step integration methods to infinitesimal elastoplasticity. *International Journal for Numerical Methods in Engineering* 1994; **37**:3169–3184.
8. Scherf O. Numerische Simulation inelastischer Körper (in German). *Ph.D. Thesis*, *Fortschritts-Berichte VDI*, (321), 2000.
9. Eckert S, Baaser H, Gross D. A BDF2 integration method with stepsize control for elastoplasticity. *Computational Mechanics* 2004; **34**:377–386.
10. Ellsiepen P, Hartmann P. Remarks on the interpretation of current nonlinear finite element analyses as differential-algebraic equations. *International Journal for Numerical Methods in Engineering* 2001; **51**:679–707.
11. Büttner J, Simeon B. Runge–Kutta methods in elastoplasticity. *Applied Numerical Mathematics* 2002; **41**:443–458.
12. Büttner J. Numerische Simulation zeitabhängiger Materialgleichungen mitFließfläche (in German). *Ph.D. Thesis*, *Fortschritts-Berichte VDI* (366), 2003.
13. Hartmann S, Bier J. High-order time integration applied to metal powder plasticity. *International Journal of Plasticity* 2008; **24**:17–54.
14. Alexander R. Diagonally implicit Runge–Kutta methods for stiff O.D.E.'s. *SIAM Journal on Numerical Analysis* 1977; **14**:1006–1021.
15. Holzapfel GA. On large strain viscoelasticity: continuum formulation and finite element applications to elastomeric structures. *International Journal for Numerical Methods in Engineering* 1996; **39**:3903–3926.
16. Kaliske N, Rothert H. Formulation and implementation of three-dimensional viscoelasticity at small and finite strains. *Computational Mechanics* 1997; **19**:228–239.
17. Reese S, Govindjee S. A theory of finite viscoelasticity and numerical aspects. *International Journal of Solids and Structures* 1998; **35**:3455–3482.
18. Huber N, Tsakmakis C. Finite deformation viscoelasticity laws. *Mechanics of Materials* 2000; **32**:1–18.
19. Govindjee S. Numerical issues in finite elasticity and viscoelasticity. In *Mechanics and Thermomechanics of Rubberlike Solids*, Saccomandi G, Ogden RW (eds). CISM Courses and Lectures, vol. 452. Springer: Berlin, 2004; 187–232.
20. Hartmann S. Computation in finite strain viscoelasticity: finite elements based on the interpretation as differential-algebraic equations. *Computer Methods in Applied Mechanics and Engineering* 2002; **191**:1439–1470.
21. Hartmann S, Wensch J. Finite element analysis of viscoelastic structures using Rosenbrock-type methods. *Computational Mechanics* 2007; **40**:383–398.
22. Hartmann S. On displacement control within the DIRK/MLNA approach in non-linear finite element analysis. In *Computational Fluid and Solid Mechanics*, Bathe K-J (ed.), vol. 1. Elsevier: Amsterdam, 2003; 316–319.
23. Hartmann S. A remark on the application of the Newton–Raphson method in non-linear finite element analysis. *Computer Methods in Applied Mechanics and Engineering* 2005; **36**:100–116.
24. Lakes RS. *Viscoelastic Solids*. CRC Press: Boca Raton, 1998.
25. Haupt P. *Continuum Mechanics and Theory of Materials* (2nd edn). Springer: Berlin, Heidelberg, New York, 2002.
26. Hairer E, Nørsett SP, Wanner G. *Solving Ordinary Differential Equations I, Nonstiff Problems* (corr. 3rd printing). Springer: Berlin, Heidelberg, 2008.
27. Hairer E, Lubich Ch, Roch M. *The Numerical Solution of Differential-Algebraic Systems by Runge-Kutta Methods*. Lecture Notes in Mathematics, vol. 1409. Springer: Berlin, Heidelberg, 1989.
28. Hairer E, Wanner G. *Solving Ordinary Differential Equations II, Stiff and Differential-Algebraic Problems* (corr. 2nd printing). Springer: Berlin, Heidelberg, 2002.
29. Hackl K. A survey on time-integration algorithms for convex and nonconvex elastoplasticity. In *From Convexity to Nonconvexity*, Gilbert RP, Panagiotopoulos PD, Pardalos P (eds). Nonsmooth Optimization and its Applications. Kluwer: Boston, 2001; 123–136.
30. Bronstein IN, Semendjajew KA, Musiol G, Mühlig H. *Taschenbuch der Mathematik*. Verlag Harri Deutsch AG: Thun, 1998.
31. Zienkiewicz OC, Taylor RL. *The Finite Element Method* (5th edn), vols. 1–3. Butterworth-Heinemann: Oxford, 2000.