Geophys. J. Int. (2007) **168,** 224–242 doi: 10.1111/j.1365-246X.2006.03193.x

An arbitrary high-order Discontinuous Galerkin method for elastic waves on unstructured meshes – III. Viscoelastic attenuation

Martin Käser,[1] Michael Dumbser,[1,2] Josep de la Puente[3] and Heiner Igel[3]

[1] *Department of Civil and Environmental Engineering, University of Trento, Trento, Italy. E-mail: martin.kaeser@ing.unitn.it*
[2] *Institut für Aerodynamik und Gasdynamik, Universität Stuttgart, Germany*
[3] *Department of Earth and Environmental Sciences, Geophysics Section, Ludwig-Maximilians-Universität, München, Germany*

Accepted 2006 August 16. Received 2006 August 16; in original form 2006 May 4

SUMMARY

We present a new numerical method to solve the heterogeneous anelastic, seismic wave equations with arbitrary high order accuracy in space and time on 3-D unstructured tetrahedral meshes. Using the velocity–stress formulation provides a linear hyperbolic system of equations with source terms that is completed by additional equations for the anelastic functions including the strain history of the material. These additional equations result from the rheological model of the generalized Maxwell body and permit the incorporation of realistic attenuation properties of viscoelastic material accounting for the behaviour of elastic solids and viscous fluids. The proposed method combines the Discontinuous Galerkin (DG) finite element (FE) method with the ADER approach using Arbitrary high order DERivatives for flux calculations. The DG approach, in contrast to classical FE methods, uses a piecewise polynomial approximation of the numerical solution which allows for discontinuities at element interfaces. Therefore, the well-established theory of numerical fluxes across element interfaces obtained by the solution of Riemann problems can be applied as in the finite volume framework. The main idea of the ADER time integration approach is a Taylor expansion in time in which all time derivatives are replaced by space derivatives using the so-called Cauchy–Kovalewski procedure which makes extensive use of the governing PDE. Due to the ADER time integration technique the same approximation order in space and time is achieved automatically and the method is a one-step scheme advancing the solution for one time step without intermediate stages. To this end, we introduce a new unrolled recursive algorithm for efficiently computing the Cauchy–Kovalewski procedure by making use of the sparsity of the system matrices. The numerical convergence analysis demonstrates that the new schemes provide very high order accuracy even on unstructured tetrahedral meshes while computational cost and storage space for a desired accuracy can be reduced when applying higher degree approximation polynomials. In addition, we investigate the increase in computing time, when the number of relaxation mechanisms due to the generalized Maxwell body are increased. An application to a well-acknowledged test case and comparisons with analytic and reference solutions, obtained by different well-established numerical methods, confirm the performance of the proposed method. Therefore, the development of the highly accurate ADER–DG approach for tetrahedral meshes including viscoelastic material provides a novel, flexible and efficient numerical technique to approach 3-D wave propagation problems including realistic attenuation and complex geometry.

Key words: attenuation, Discontinuous Galerkin, high order accuracy, relaxation, unstructured meshes, viscoelasticity.

1 INTRODUCTION

Modern numerical methods for the simulation of seismic wave propagation are becoming more accurate, can handle complex 3-D geometries and provide increasingly important tools to simulate more realistic wavefields for local, regional and global wave propagation problems. Therefore, second-order effects such as attenuation and dispersion, which strongly affect the seismic wavefield, have to be incorporated to correctly model the wave amplitudes and phases of a fully 3-D seismic wavefield. A successful model for realistic attenuation is the approximation of the material as a viscoelastic medium that combines the behaviour of both, elastic solids and viscous fluids. Hereby, it is

important that the Earth's internal friction, that is, the measure of attenuation, is nearly constant over a wide seismic frequency range. This is due to the composition of the Earth's polycristalline material consisting of different minerals. The superposition of these microscopic physical attenuation (relaxation) processes leads to a flat attenuation band (Liu *et al.* 1976; Stein & Wysession 2003).

The stress–strain relation for a linear isotropic viscoelastic medium is given by the so-called Boltzmann principle (causality principle), which states that the stress at a given time t depends on the entire strain history until time t, which mathematically is represented by a time convolution of a relaxation function and the strain rate as shown, for example, by Moczo *et al.* (2004). As the integration of this stress–strain relation in the time domain is intractable in a numerical computation, Day & Minster (1984) transformed the stress–strain relation in the time domain into a differential form using a Padé approximation. They obtained n differential equations for n additional *internal variables*, which replace the convolution integral. These equations have to be solved in addition to the elastic wave equations. Furthermore, the sum of the internal variables multiplied with anelastic coefficients leads to additional viscoelastic terms for the elastic stresses. This way storage requirements and computing times were significantly increased.

Emmerich & Korn (1987) improved this approach by considering the rheology of a *generalized Maxwell body* and showed that their method is superior in accuracy and computational efficiency. They chose the relaxation frequencies logarithmically equidistant in the frequency band of interest and used a least-square method to fit arbitrary quality factor laws. Independently, a different approach (Carcione *et al.* 1988; Carcione & Cavallini 1994) assumed a *generalized Zener body* and introduced additional first-order differential equations for *memory variables*. After these revolutionary publications authors incorporating realistic viscoelastic attenuation in time domain methods used the concepts of the generalized Maxwell or generalized Zener body. A recent work by Moczo & Kristek (2005) reviewed both models and showed that both approaches are equivalent.

After Emmerich (1992) applied the viscoelastic models for the P-SV case, Moczo *et al.* (1997) presented a hybrid two-step method for simulating P-SV seismic motion in inhomogeneous viscoelastic structures with free surface topography combining discrete-wavenumber (DW) (Bouchon 1981), finite element (FE) (for example, Marfurt 1984) and finite-difference (FD) methods (for example, Moczo & Bard 1993). At first coarse spatial sampling of the anelastic functions was introduced (Day 1998; Day & Bradley 2001). In later work (Kristek & Moczo 2003) the basic theoretical and algorithmic aspects of a memory-efficient implementation of realistic attenuation was addressed based on a viscoelastic material with material discontinuities mainly for the staggered-grid finite difference approach.

In this paper, we incorporate realistic attenuation by viscoelastic material into the high order Discontinuous Galerkin (DG) approach. Originally the DG method was developed by Reed & Hill (1973). Later Dumbser (2005) and Dumbser & Munz (2005a,b) combined the DG approach with the *ADER* time integration technique (Toro *et al.* 2001; Titarev & Toro 2002; Toro & Titarev 2002) using *A*rbitrary high order *DER*ivatives. This new highly accurate numerical method was then introduced for the simulation of seismic wave propagation on unstructured meshes for two and three space dimensions (Käser & Dumbser 2006; Dumbser & Käser 2006). In this new approach we approximate the unknown solution as well as the additional anelastic functions, provided by the generalized Maxwell body, inside each tetrahedral element by a polynomial, whose coefficients—the degrees of freedom—are advanced in time. Hereby, the solution can be *discontinuous* across the element interfaces, which allows to incorporate the well-established ideas of numerical flux functions from the finite volume framework (Dumbser & Käser 2006; Dumbser *et al.* 2006; Käser & Dumbser 2006).

This paper is structured as follows. In Section 2, we introduce the system of the 3-D anelastic wave equations in velocity–stress formulation including attenuation due to viscoelasticity. The resulting DG method is briefly explained in Section 3 together with the ADER time integration approach. However, to avoid repetition we strongly refer to previous work on the purely elastic case (Dumbser & Käser 2006). Furthermore, we give a new and more efficient formulation for the required Cauchy–Kovalewski procedure. In Section 4, we illustrate the improvement of the approximation of a frequency-independent Q-law when increasing the number n of relaxation mechanisms of the generalized Maxwell body. Furthermore, we analyse the additional CPU time requirements for different orders of accuracy of the ADER–DG schemes. Results of numerical tests of convergence of the proposed ADER–DG scheme for anelastic wave propagation on tetrahedral meshes are shown in Section 5. Finally, in Section 6, we present a comparison of our results with those published after an acknowledged 3-D benchmark test of the Pacific Earthquake Engineering Research Center (Day *et al.* 2003) providing an analytic and a number of reference solutions obtained by well-established codes of other research institutions. In particular, we compare the results for different orders of accuracy combined with different numbers of relaxation mechanisms.

2 ANELASTIC WAVE EQUATIONS

The anelastic wave propagation can be described by modifying the constitutive relation, that is, Hooke's Law, as shown for example, in Moczo *et al.* (2004) and transforming it into the frequency domain. The relation between stresses $\vec{\sigma} = (\sigma_{xx}, \sigma_{yy}, \sigma_{zz}, \sigma_{xy}, \sigma_{yz}, \sigma_{xz})^T$ and strains $\vec{\varepsilon} = (\varepsilon_{xx}, \varepsilon_{yy}, \varepsilon_{zz}, \varepsilon_{xy}, \varepsilon_{yz}, \varepsilon_{xz})^T$ in the case of linear viscoelasticity can then be written as

$$\vec{\sigma}_i(\omega) = M_{ij}(\omega)\vec{\varepsilon}_j(\omega), \tag{1}$$

where M_{ij} is a matrix including complex, frequency-dependent viscoelastic moduli. In general M_{ij} has 21 independent entries, however, for the isotropic case they reduce to the two Lamé parameters $\lambda = \lambda(\omega)$ and $\mu = \mu(\omega)$.

The rheological model that defines the parameters of M_{ij} has to have a physically feasible expression that, in addition, reproduces the expected results of stress and strain damping as well as experimental observation of strain response to stress loads. Liu *et al.* (1976) propose

a superposition of different relaxation mechanisms as a way to fulfil both conditions. As introduced by Emmerich & Korn (1987) and clearly derived in Kristek & Moczo (2003) and Moczo *et al.* (2004) viscoelastic moduli of n Maxwell bodies and one spring, all connected in parallel, can be expressed as

$$\lambda(\omega) = \lambda^U \left(1 - \sum_{\ell=1}^{n} \frac{Y_\ell^\lambda \omega_\ell}{\omega_\ell + i\omega}\right), \tag{2}$$

$$\mu(\omega) = \mu^U \left(1 - \sum_{\ell=1}^{n} \frac{Y_\ell^\mu \omega_\ell}{\omega_\ell + i\omega}\right), \tag{3}$$

where $\lambda^U = \lim_{\omega\to\infty} \lambda(\omega)$ and $\mu^U = \lim_{\omega\to\infty} \mu(\omega)$ are the unrelaxed Lamé parameters as used in purely elastic media. The Y_ℓ^λ and Y_ℓ^μ are the anelastic coefficients to be determined and ω_ℓ are the relaxation frequencies of the different mechanisms. In general, given a viscoelastic modulus, for example, the shear modulus $\mu(\omega)$, the quality factor $\mathcal{Q}(\omega)$ is defined as

$$\mathcal{Q}_\mu(\omega) = \frac{Re[\mu(\omega)]}{Im[\mu(\omega)]}. \tag{4}$$

Inserting the shear modulus $\mu(\omega)$ from eq. (3) into eq. (4) leads to

$$\mathcal{Q}_\mu^{-1}(\omega) = \sum_{\ell=1}^{n} \frac{\omega_\ell \omega + \omega_\ell^2 \mathcal{Q}_\mu^{-1}(\omega)}{\omega_\ell^2 + \omega^2} Y_\ell^\mu. \tag{5}$$

Eq. (5) can be used to fit any $\mathcal{Q}(\omega)$-law (Emmerich & Korn 1987; Moczo *et al.* 2004). Observations show that the quality factor $\mathcal{Q}$ is approximately constant over a large frequency range of interest for most geophysical applications. They propose that good approximations can be obtained by choosing n relaxation frequencies ω_ℓ, $\ell = 1, \ldots, n$, that equidistantly cover the logarithmic frequency range of interest. They suggest to use $2n-1$ known values $\mathcal{Q}(\tilde{\omega}_k)$ at frequencies $\tilde{\omega}_k$, $k = 1, \ldots, 2n-1$, with $\tilde{\omega}_1 = \omega_1$ and $\tilde{\omega}_{2n-1} = \omega_n$ and solve the overdetermined system in eq. (5) for the anelastic coefficients Y_ℓ^μ by the least-square method. A more detailed discussion of the choice of frequency ranges and the corresponding sampling frequencies can be found in Graves & Day (2003).

In practice, and corresponding to the seismic P- and S-wave velocities, we have quality factors $\mathcal{Q}_P$ and $\mathcal{Q}_S$ that describe the different degree of attenuation for the different wave types. Relations as eq. (5) can be also found for anelastic coefficients Y_ℓ^P and Y_ℓ^S for viscoelastic P- and S-wave propagation and read as

$$\mathcal{Q}_\nu^{-1}(\omega_k) = \sum_{\ell=1}^{n} \frac{\omega_\ell \omega_k + \omega_\ell^2 \mathcal{Q}_\nu^{-1}(\omega_k)}{\omega_\ell^2 + \omega_k^2} Y_\ell^\nu, \qquad \text{with} \quad \nu = P, S, \quad \text{and} \quad k = 1, \ldots, 2n-1. \tag{6}$$

In the following, however, it is more convenient to express the anelastic coefficients in terms of the Lamé parameters λ and μ, which are obtained by the transformation

$$Y_\ell^\lambda = \left(1 + \frac{2\mu}{\lambda}\right) Y_\ell^P - \frac{2\mu}{\lambda} Y_\ell^S, \qquad Y_\ell^\mu = Y_\ell^S, \tag{7}$$

following directly from eqs (2) and (3) as the relation of physical parameters, for example, elastic parameters or velocities, corresponds to the purely elastic case due to the linearity of the expressions in eqs (2) and (3).

We can use a set of material-independent anelastic functions introduced by Kristek & Moczo (2003) and Moczo & Kristek (2005). They are defined in the form $\vec{\vartheta}^\ell = (\bar{\vartheta}_{xx}^\ell, \bar{\vartheta}_{yy}^\ell, \bar{\vartheta}_{zz}^\ell, \bar{\vartheta}_{xy}^\ell, \bar{\vartheta}_{yz}^\ell, \bar{\vartheta}_{xz}^\ell)^T$, and contain the time history of the strain through

$$\bar{\vartheta}_j^\ell(t) = \omega_\ell \int_{-\infty}^{t} \varepsilon_j(\tau) e^{-\omega_\ell(t-\tau)}\, d\tau. \tag{8}$$

Using eq. (8) and applying the inverse Fourier transform to the viscoelastic modulus M_{ij}, as presented in detail by Kristek & Moczo (2003), the stress–strain relation eq. (1) can be written in the time domain in the form

$$\sigma_{ij} = \lambda \varepsilon_{kk} \delta_{ij} + 2\mu \varepsilon_{ij} - \sum_{\ell=1}^{n} \left(\lambda Y_\ell^\lambda \bar{\vartheta}_{kk}^\ell \delta_{ij} + 2\mu Y_\ell^\mu \bar{\vartheta}_{ij}^\ell\right), \qquad \text{with} \quad i, j, k \in [x, y, z], \tag{9}$$

where δ_{ij} is the Kronecker Delta and the equal-index summation convention applies to the index kk. The viscoelastic constitutive relation in eq. (9) represents the elastic part minus the anelastic part depending on the anelastic coefficients Y_ℓ^λ and Y_ℓ^μ and the anelastic functions $\bar{\vartheta}_{ij}^\ell$. The remaining problem is the evolution of the anelastic functions $\bar{\vartheta}_{ij}^\ell$ in eq. (8) in time. In fact, eq. (8) is the solution of the partial differential equation

$$\frac{\partial}{\partial t} \bar{\vartheta}_j^\ell(t) + \omega_\ell \bar{\vartheta}_j^\ell(t) = \omega_\ell \varepsilon_j, \tag{10}$$

which completes the linear, hyperbolic system of the anelastic wave equations.

However, to express the equation system in the velocity–stress formulation it is convenient to redefine the anelastic functions in the form (Moczo *et al.* 2004)

$$\vartheta_j^\ell = \frac{\partial}{\partial t} \bar{\vartheta}_j^\ell. \tag{11}$$

Finally, using the equations of motion, the definition of strain ε_j and eqs (9)–(11) we can formulate the system of the anelastic wave equations as

$$
\begin{aligned}
\frac{\partial}{\partial t}\sigma_{xx} - (\lambda+2\mu)\frac{\partial}{\partial x}u - \lambda\frac{\partial}{\partial y}v - \lambda\frac{\partial}{\partial z}w &= \sum_{\ell=1}^{n} -\big(\lambda Y_\ell^\lambda + 2\mu Y_\ell^\mu\big)\vartheta_{xx}^\ell - \lambda Y_\ell^\lambda \vartheta_{yy}^\ell - \lambda Y_\ell^\lambda \vartheta_{zz}^\ell,\\
\frac{\partial}{\partial t}\sigma_{yy} - \lambda\frac{\partial}{\partial x}u - (\lambda+2\mu)\frac{\partial}{\partial y}v - \lambda\frac{\partial}{\partial z}w &= \sum_{\ell=1}^{n} -\lambda Y_\ell^\lambda \vartheta_{xx}^\ell - \big(\lambda Y_\ell^\lambda + 2\mu Y_\ell^\mu\big)\vartheta_{yy}^\ell - \lambda Y_\ell^\lambda \vartheta_{zz}^\ell,\\
\frac{\partial}{\partial t}\sigma_{zz} - \lambda\frac{\partial}{\partial x}u - \lambda\frac{\partial}{\partial y}v - (\lambda+2\mu)\frac{\partial}{\partial z}w &= \sum_{\ell=1}^{n} -\lambda Y_\ell^\lambda \vartheta_{xx}^\ell - \lambda Y_\ell^\lambda \vartheta_{yy}^\ell - \big(\lambda Y_\ell^\lambda + 2\mu Y_\ell^\mu\big)\vartheta_{zz}^\ell,\\
\frac{\partial}{\partial t}\sigma_{xy} - \mu\left(\frac{\partial}{\partial x}v + \frac{\partial}{\partial y}u\right) &= \sum_{\ell=1}^{n} -2\mu Y_\ell^\mu \vartheta_{xy}^\ell,\\
\frac{\partial}{\partial t}\sigma_{yz} - \mu\left(\frac{\partial}{\partial z}v + \frac{\partial}{\partial y}w\right) &= \sum_{\ell=1}^{n} -2\mu Y_\ell^\mu \vartheta_{yz}^\ell,\\
\frac{\partial}{\partial t}\sigma_{xz} - \mu\left(\frac{\partial}{\partial z}u + \frac{\partial}{\partial x}w\right) &= \sum_{\ell=1}^{n} -2\mu Y_\ell^\mu \vartheta_{xz}^\ell,\\
\rho\frac{\partial}{\partial t}u - \frac{\partial}{\partial x}\sigma_{xx} - \frac{\partial}{\partial y}\sigma_{xy} - \frac{\partial}{\partial z}\sigma_{xz} &= 0,\\
\rho\frac{\partial}{\partial t}v - \frac{\partial}{\partial x}\sigma_{xy} - \frac{\partial}{\partial y}\sigma_{yy} - \frac{\partial}{\partial z}\sigma_{yz} &= 0,\\
\rho\frac{\partial}{\partial t}w - \frac{\partial}{\partial x}\sigma_{xz} - \frac{\partial}{\partial y}\sigma_{yz} - \frac{\partial}{\partial z}\sigma_{zz} &= 0,\\
\frac{\partial}{\partial t}\vartheta_{xx}^1 - \omega_1\frac{\partial}{\partial x}u &= -\omega_1\vartheta_{xx}^1,\\
\frac{\partial}{\partial t}\vartheta_{yy}^1 - \omega_1\frac{\partial}{\partial y}v &= -\omega_1\vartheta_{yy}^1,\\
\frac{\partial}{\partial t}\vartheta_{zz}^1 - \omega_1\frac{\partial}{\partial z}w &= -\omega_1\vartheta_{zz}^1,\\
\frac{\partial}{\partial t}\vartheta_{xy}^1 - \frac{1}{2}\omega_1\left(\frac{\partial}{\partial x}v + \frac{\partial}{\partial y}u\right) &= -\omega_1\vartheta_{xy}^1,\\
\frac{\partial}{\partial t}\vartheta_{yz}^1 - \frac{1}{2}\omega_1\left(\frac{\partial}{\partial z}v + \frac{\partial}{\partial y}w\right) &= -\omega_1\vartheta_{yz}^1,\\
\frac{\partial}{\partial t}\vartheta_{xz}^1 - \frac{1}{2}\omega_1\left(\frac{\partial}{\partial z}u + \frac{\partial}{\partial x}w\right) &= -\omega_1\vartheta_{xz}^1,\\
&\vdots \quad \vdots\\
\frac{\partial}{\partial t}\vartheta_{xx}^n - \omega_n\frac{\partial}{\partial x}u &= -\omega_n\vartheta_{xx}^n,\\
\frac{\partial}{\partial t}\vartheta_{yy}^n - \omega_n\frac{\partial}{\partial y}v &= -\omega_n\vartheta_{yy}^n,\\
\frac{\partial}{\partial t}\vartheta_{zz}^n - \omega_n\frac{\partial}{\partial z}w &= -\omega_n\vartheta_{zz}^n,\\
\frac{\partial}{\partial t}\vartheta_{xy}^n - \frac{1}{2}\omega_n\left(\frac{\partial}{\partial x}v + \frac{\partial}{\partial y}u\right) &= -\omega_n\vartheta_{xy}^n,\\
\frac{\partial}{\partial t}\vartheta_{yz}^n - \frac{1}{2}\omega_n\left(\frac{\partial}{\partial z}v + \frac{\partial}{\partial y}w\right) &= -\omega_n\vartheta_{yz}^n,\\
\frac{\partial}{\partial t}\vartheta_{xz}^n - \frac{1}{2}\omega_n\left(\frac{\partial}{\partial z}u + \frac{\partial}{\partial x}w\right) &= -\omega_n\vartheta_{xz}^n\,,
\end{aligned}
\tag{12}
$$

where n is the number of mechanisms used to approximate a frequency-independent Q-law and ρ is the density. Note that each mechanism adds 6 further equations, that is, one for each stress component. Therefore, the system of the purely elastic 3-D wave equations consisting of 9 equations increases by $6n$ equations in the anelastic case, when n mechanisms are used. Furthermore, the anelasticity adds reactive source terms on the right-hand side of eq. (12). The inclusion of external source terms is explained in detail in previous work (Käser & Dumbser 2006).

In the following, we assume that the viscoelastic material is described with the same number n of mechanisms throughout the computational domain. Therefore, the notation will be identical to previous work (Dumbser & Käser 2006) treating the purely elastic case.

The above system (12) of $n_v = 9 + 6n$ variables and equations can be written in the more compact form

$$\frac{\partial Q_p}{\partial t} + \check{A}_{pq}\frac{\partial Q_q}{\partial x} + \check{B}_{pq}\frac{\partial Q_q}{\partial y} + \check{C}_{pq}\frac{\partial Q_q}{\partial z} = \check{E}_{pq}Q_q. \tag{13}$$

Note that the dimensions of the variable vector Q, the Jacobian matrices $\check{A}$, $\check{B}$, $\check{C}$ and the source matrix $\check{E}$ now depend on the number n of relaxation mechanisms. To keep the notation as simple as possible and without loss of generality, in the following we assume that the order of the variables in eq. (13) is such that $p, q \in [1, \ldots, 9]$ denote the elastic part and $p, q \in [10, \ldots, n_v]$, denote the anelastic part of the system as presented in eq. (12). As the Jacobian matrices $\check{A}$, $\check{B}$ and $\check{C}$ as well as the source matrix $\check{E}$ are sparse and show some particular symmetry pattern and as their dimensions may become impractical for notation, we will use the block-matrix syntax.

Therefore, we decompose the Jacobian matrices as follows:

$$\check{A} = \begin{bmatrix} A & 0 \\ A_a & 0 \end{bmatrix} \in \mathbb{R}^{n_v \times n_v}, \qquad \check{B} = \begin{bmatrix} B & 0 \\ B_a & 0 \end{bmatrix} \in \mathbb{R}^{n_v \times n_v}, \qquad \check{C} = \begin{bmatrix} C & 0 \\ C_a & 0 \end{bmatrix} \in \mathbb{R}^{n_v \times n_v}, \tag{14}$$

where $A, B, C \in \mathbb{R}^{9\times 9}$ are the Jacobians of the purely elastic part as given in Dumbser & Käser (2006). The matrices A_a, B_a and C_a include the anelastic part and exhibit themselves a block structure of the form

$$A_a = \begin{bmatrix} A_1 \\ \vdots \\ A_n \end{bmatrix} \in \mathbb{R}^{6n\times 9}, \qquad B_a = \begin{bmatrix} B_1 \\ \vdots \\ B_n \end{bmatrix} \in \mathbb{R}^{6n\times 9}, \qquad C_a = \begin{bmatrix} C_1 \\ \vdots \\ C_n \end{bmatrix} \in \mathbb{R}^{6n\times 9}, \tag{15}$$

where each submatrix $A_\ell, B_\ell, C_\ell \in \mathbb{R}^{6\times 9}$, with $\ell = 1, \ldots, n$, contains the relaxation frequency ω_ℓ of the ℓth mechanism in the form

$$A_\ell = \omega_\ell \cdot \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{2} \end{pmatrix}, \tag{16}$$

$$B_\ell = \omega_\ell \cdot \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \tag{17}$$

$$C_\ell = \omega_\ell \cdot \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{2} & 0 & 0 \end{pmatrix}. \tag{18}$$

The matrix $\check{E}$ in eq. (13) representing a reaction source that couples the anelastic functions to the original elastic system can be decomposed as

$$\check{E} = \begin{bmatrix} 0 & E \\ 0 & E' \end{bmatrix} \in \mathbb{R}^{n_v \times n_v}, \tag{19}$$

with E of the block structure

$$E = [E_1, \ldots, E_n] \in \mathbb{R}^{9\times 6n}, \tag{20}$$

where each matrix $E_\ell \in \mathbb{R}^{9\times 6}$, with $\ell = 1, \ldots, n$, contains the anelastic coefficients Y_ℓ^λ and Y_ℓ^μ of the ℓth mechanism in the form

$$E_\ell = \begin{pmatrix} -\lambda Y_\ell^\lambda - 2\mu Y_\ell^\mu & -\lambda Y_\ell^\lambda & -\lambda Y_\ell^\lambda & 0 & 0 & 0 \\ -\lambda Y_\ell^\lambda & -\lambda Y_\ell^\lambda - 2\mu Y_\ell^\mu & -\lambda Y_\ell^\lambda & 0 & 0 & 0 \\ -\lambda Y_\ell^\lambda & -\lambda Y_\ell^\lambda & -\lambda Y_\ell^\lambda - 2\mu Y_\ell^\mu & 0 & 0 & 0 \\ 0 & 0 & 0 & -2\mu Y_\ell^\mu & 0 & 0 \\ 0 & 0 & 0 & 0 & -2\mu Y_\ell^\mu & 0 \\ 0 & 0 & 0 & 0 & 0 & -2\mu Y_\ell^\mu \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \tag{21}$$

The matrix E' in eq. (13) is a diagonal matrix and has the structure

$$E' = \begin{bmatrix} E'_1 & & 0 \\ & \ddots & \\ 0 & & E'_n \end{bmatrix} \in \mathbb{R}^{6n\times 6n}, \tag{22}$$

where each matrix $E'_\ell \in \mathbb{R}^{6\times 6}$, with $\ell = 1, \ldots, n$, is itself a diagonal matrix containing only the relaxation frequency ω_ℓ of the ℓth mechanism on its diagonal, that is, $E'_\ell = -\omega_\ell \cdot I$ with $I \in \mathbb{R}^{6\times 6}$ denoting the identity matrix.

3 THE NUMERICAL SCHEME

The computational domain $\Omega \in \mathbb{R}^3$ is divided into conforming tetrahedral elements $\mathcal{T}^{(m)}$ being addressed by a unique index (m). Furthermore, we suppose the matrices $\check{A}_{pq}$, $\check{B}_{pq}$, $\check{C}_{pq}$ and $\check{E}_{pq}$ in eq. (13) to be piecewise constant inside an element $\mathcal{T}^{(m)}$.

The construction of the ADER–DG scheme including viscoelastic attenuation remains basically the same as that presented in detail by Dumbser & Käser (2006) for the purely elastic case. The difference is in a larger linear hyperbolic system of equations and additional source terms as given in eqs (12) or (13).

The numerical solution Q_h of eq. (13) is approximated inside a tetrahedron $\mathcal{T}^{(m)}$ by a linear combination of space-dependent but time-independent polynomial basis functions $\Phi_l(\xi, \eta, \zeta)$ with support $\mathcal{T}^{(m)}$ and with only time-dependent degrees of freedom $\hat{Q}_{pl}^{(m)}(t)$:

$$\left(Q_h^{(m)}\right)_p(\xi, \eta, \zeta, t) = \hat{Q}_{pl}^{(m)}(t)\Phi_l(\xi, \eta, \zeta), \tag{23}$$

where ξ, η and ζ are the coordinates in a reference element $\mathcal{T}_E$ and are defined in the appendix of Dumbser & Käser (2006). The index h denotes the numerical solution, the index p stands for the number of unknowns in the vector Q and l indicates the lth basis function. The maximum polynomial degree N of the basis functions Φ_l in eq. (23) determines the number of necessary basis functions, that is, $l = 1, \ldots, (N+1)(N+2)(N+3)/6$, and leads to a numerical scheme of order $N+1$ in space and time (Dumbser & Käser 2006). We use the orthogonal basis functions Φ_k of Cockburn *et al.* (2000) given in their explicit form in Appendix A.

Following Dumbser & Käser (2006) and not repeating the tedious derivation of the numerical scheme, we get the fully discrete one-step ADER–DG scheme of the form

$$\begin{aligned} &\left[\left(\hat{Q}_{pl}^{(m)}\right)^{n+1} - \left(\hat{Q}_{pl}^{(m)}\right)^{n}\right]|J|M_{kl} \\ &\quad + \frac{1}{2}\sum_{j=1}^{4} \check{T}_{pq}^{j}\left(\check{A}_{qr}^{(m)} + \left|\check{A}_{qr}^{(m)}\right|\right)\left(\check{T}_{rs}^{j}\right)^{-1}|S_j|F_{kl}^{-,j} \cdot I_{slmn}(\Delta t)\left(\hat{Q}_{mn}^{(m)}\right)^{n} \\ &\quad + \frac{1}{2}\sum_{j=1}^{4} \check{T}_{pq}^{j}\left(\check{A}_{qr}^{(m)} - \left|\check{A}_{qr}^{(m)}\right|\right)\left(\check{T}_{rs}^{j}\right)^{-1}|S_j|F_{kl}^{+,j,i,h} \cdot I_{slmn}(\Delta t)\left(\hat{Q}_{mn}^{(m_j)}\right)^{n} \\ &\quad - \check{A}_{pq}^{*}|J|K_{kl}^{\xi} \cdot I_{qlmn}(\Delta t)\left(\hat{Q}_{mn}^{(m)}\right)^{n} - \check{B}_{pq}^{*}|J|K_{kl}^{\eta} \cdot I_{qlmn}(\Delta t)\left(\hat{Q}_{mn}^{(m)}\right)^{n} - \check{C}_{pq}^{*}|J|K_{kl}^{\zeta} \cdot I_{qlmn}(\Delta t)\left(\hat{Q}_{mn}^{(m)}\right)^{n} \\ &\quad = |J|\check{E}_{pq} \cdot I_{qlmn}(\Delta t)\left(\hat{Q}_{mn}^{(m)}\right)^{n} M_{kl} \end{aligned} \tag{24}$$

to evolve the degrees of freedom $\hat{Q}_{pl}^{(m)}$ in element $\mathcal{T}^{(m)}$ from time level t^n to t^{n+1}. We remark that the total number of unknowns in the viscoelastic case now depends on the number of relaxation mechanisms n and $p, q, r, s = 1, \ldots, n_v$. As in Dumbser & Käser (2006) M_{kl} is the mass matrix, $F_{kl}^{-,j}$, $F_{kl}^{+,j,i,h}$ are the flux matrices, K_{kl}^{ξ}, K_{kl}^{η} and K_{kl}^{ζ} are the stiffness matrices and $I_{qlmn}(\Delta t)$ is the tensor responsible for the high order time integration over one time step Δt. $\check{A}_{pq}^{*}$, $\check{B}_{pq}^{*}$ and $\check{C}_{pq}^{*}$ are the Jacobian matrices transformed into the reference tetrahedron $\mathcal{T}_E$. Furthermore, $|J|$ is the determinant of the Jacobian matrix of this transformation, see the appendix in Dumbser & Käser (2006), and $|S_j|$ denotes the area of the jth face of tetrahedron $\mathcal{T}^{(m)}$.

However, in the following let us skip the element index (m) and have a closer look at the viscosity matrix $|\check{A}_{qr}|$ introduced by the numerical flux (Toro 1999) and the rotation matrix $\check{T}_{pq}^{j}$ in eq. (24), which are used for the flux computation in normal direction to element interfaces.

Similarly to eq. (14), we find that

$$|\check{A}| = \begin{bmatrix} |A| & 0 \\ A^{||} & 0 \end{bmatrix} \in \mathbb{R}^{n_v \times n_v}, \tag{25}$$

where $|A| \in \mathbb{R}^{9\times 9}$ is identical to the one of the purely elastic part as given in Dumbser & Käser (2006) and has the form

$$|A| = \begin{pmatrix} c_p & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \lambda/(c_p\rho) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \lambda/(c_p\rho) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & c_s & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & c_s & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & c_p & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_s & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_s \end{pmatrix}, \tag{26}$$

with $c_p = \sqrt{\frac{\lambda+2\mu}{\rho}}$ and $c_s = \sqrt{\frac{\mu}{\rho}}$, the P- and S-wave velocities of the unrelaxed purely elastic material.

The matrix $A^{||}$ includes the anelastic part and exhibits itself a block structure similar to that in eq. (15) of the form

$$A^{||} = \begin{bmatrix} A_1^{||} \\ \vdots \\ A_n^{||} \end{bmatrix} \in \mathbb{R}^{6n\times 9}, \tag{27}$$

where each submatrix $A_\ell^{||} \in \mathbb{R}^{6\times 9}$, with $\ell = 1, \ldots, n$, contains the local unrelaxed material parameters and the relaxation frequency ω_ℓ of the ℓth relaxation mechanism in the form

$$A_\ell^{||} = \omega_\ell \cdot \begin{pmatrix} 1/(c_p\rho) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/(2c_s\rho) & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/(2c_s\rho) & 0 & 0 & 0 \end{pmatrix}. \tag{28}$$

Skipping the index j for the jth face of a tetrahedral element, and recalling that the anelastic functions ϑ^ℓ are tensors like the stresses the rotation matrix $\check{T}_{pq}$ for the full anelastic system in eq. (24) has the form

$$\check{T} = \begin{bmatrix} T^t & 0 & 0 \\ 0 & T^v & 0 \\ 0 & 0 & T_a \end{bmatrix} \in \mathbb{R}^{n_v \times n_v}, \tag{29}$$

where $T^t \in \mathbb{R}^{6\times 6}$ is the rotation matrix responsible for the stress tensor rotation as in the purely elastic part and is given as

$$T^t = \begin{pmatrix} n_x^2 & s_x^2 & t_x^2 & 2n_x s_x & 2s_x t_x & 2n_x t_x \\ n_y^2 & s_y^2 & t_y^2 & 2n_y s_y & 2s_y t_y & 2n_y t_y \\ n_z^2 & s_z^2 & t_z^2 & 2n_z s_z & 2s_z t_z & 2n_z t_z \\ n_y n_x & s_y s_x & t_y t_x & n_y s_x + n_x s_y & s_y t_x + s_x t_y & n_y t_x + n_x t_y \\ n_z n_y & s_z s_y & t_z t_y & n_z s_y + n_y s_z & s_z t_y + s_y t_z & n_z t_y + n_y t_z \\ n_z n_x & s_z s_x & t_z t_x & n_z s_x + n_x s_z & s_z t_x + s_x t_z & n_z t_x + n_x t_z \end{pmatrix}, \tag{30}$$

with the components of the normal vector $\vec{n} = (n_x, n_y, n_z)^T$ and the two tangential vectors $\vec{s} = (s_x, s_y, s_z)^T$ and $\vec{t} = (t_x, t_y, t_z)^T$, which lie in the plane determined by the boundary face of the tetrahedron and are orthogonal to each other and the normal vector $\vec{n}$ as shown in Dumbser & Käser (2006).

The matrix $T^v \in \mathbb{R}^{3\times 3}$ is the rotation matrix responsible for the velocity vector rotation as in the purely elastic part and is given as

$$T^v = \begin{pmatrix} n_x & s_x & t_x \\ n_y & s_y & t_y \\ n_z & s_z & t_z \end{pmatrix}. \tag{31}$$

The matrix T_a in eq. (29) is a block diagonal matrix and has the structure

$$T_a = \begin{bmatrix} T^t & & 0 \\ & \ddots & \\ 0 & & T^t \end{bmatrix} \in \mathbb{R}^{6n \times 6n}, \tag{32}$$

where each of the n submatrices T^t is the tensor rotation matrix given in eq. (30).

Using the symmetries of $\check{A}$, $|\check{A}|$ and $\check{T}$ and the particular composition of the source term matrix $\check{E}$ as given in eq. (13)–(22), we can separate the full system in eq. (13) into two parts. We call the first 9 equations the *elastic part* and the remaining equations 10 to n_v the *anelastic part*. Therefore, the fluxes and volume integrals appearing in the discrete formulation of the DG approach in eq. (24) can be computed separately for each part. Furthermore, the computation of the flux and stiffness contributions of the anelastic part can be reduced to 6 instead of $6n$, as for each mechanisms the corresponding matrices remain the same. Only the multiplication with the relaxation frequency ω_ℓ depends on the ℓth mechanism. However, both parts are still coupled via the Cauchy–Kovalewski procedure of the ADER time integration approach and the source terms $\check{E}$ in eq. (13).

In the following Section 3.1, we present in detail how this coupling is accomplished with a new, more efficient time integration approach in order to replace the costly multiplication with the 4-D tensor $I_{qlmn}(\Delta t)$ in eq. (24).

3.1 The ADER time discretization

One could use a similar algorithm as presented in Käser & Dumbser (2006) and Dumbser & Käser (2006) to compute the Cauchy–Kovalewski procedure explicitly using the tensor $I_{plqm}(\Delta t)$, but for huge systems expressed through eq. (13) that arise when using a large number of relaxation mechanisms, this approach would be too slow because of the many matrix–matrix multiplications involved. Therefore, in this article we present a different approach that turned out to be equal to the previous one, however, it is much faster. The unrolled recursive algorithm described in the following becomes especially efficient because the matrices $\check{A}_{pq}$, $\check{B}_{pq}$, $\check{C}_{pq}$ and $\check{E}_{pq}$ are usually very sparse as shown in Section 2.

As in Käser & Dumbser (2006) and Dumbser & Käser (2006) we first write the governing PDE (13) in the reference system as

$$\frac{\partial Q_p}{\partial t} + \check{A}^*_{pq} \frac{\partial Q_q}{\partial \xi} + \check{B}^*_{pq} \frac{\partial Q_q}{\partial \eta} + \check{C}^*_{pq} \frac{\partial Q_q}{\partial \zeta} - \check{E}_{pq}\, Q_q = 0, \tag{33}$$

with

$$\check{A}^*_{pq} = \check{A}_{pq} \frac{\partial \xi}{\partial x} + \check{B}_{pq} \frac{\partial \xi}{\partial y} + \check{C}_{pq} \frac{\partial \xi}{\partial z}, \qquad \check{B}^*_{pq} = \check{A}_{pq} \frac{\partial \eta}{\partial x} + \check{B}_{pq} \frac{\partial \eta}{\partial y} + \check{C}_{pq} \frac{\partial \eta}{\partial z}, \qquad \check{C}^*_{pq} = \check{A}_{pq} \frac{\partial \zeta}{\partial x} + \check{B}_{pq} \frac{\partial \zeta}{\partial y} + \check{C}_{pq} \frac{\partial \zeta}{\partial z}. \tag{34}$$

In contrast to the approach in Käser & Dumbser (2006) and Dumbser & Käser (2006), we now immediately project the modified governing equation (33) onto the DG basis functions and insert the spatial DG approximation. We obtain

$$\langle \Phi_k, \Phi_l \rangle \frac{\partial}{\partial t} \hat{Q}_{pl}(t) + \left\langle \Phi_k, \frac{\partial \Phi_l}{\partial \xi} \right\rangle \check{A}^*_{pq} \hat{Q}_{ql}(t) + \left\langle \Phi_k, \frac{\partial \Phi_l}{\partial \eta} \right\rangle \check{B}^*_{pq} \hat{Q}_{ql}(t) + \left\langle \Phi_k, \frac{\partial \Phi_l}{\partial \zeta} \right\rangle \check{C}^*_{pq} \hat{Q}_{ql}(t) - \langle \Phi_k, \Phi_l \rangle \check{E}_{pq} \hat{Q}_{ql}(t) = 0, \tag{35}$$

where $\langle a, b \rangle = \int_{\mathcal{T}_E} a \cdot b \, dV$ denotes the inner product over the reference tetrahedron $\mathcal{T}_E$. Eq. (35) can be reformulated using the definitions of the mass matrix $M_{kl} = \langle \Phi_k, \Phi_l \rangle$, the stiffness matrices $K^\xi_{kl} = \langle \frac{\partial \Phi_k}{\partial \xi}, \Phi_l \rangle$, $K^\eta_{kl} = \langle \frac{\partial \Phi_k}{\partial \eta}, \Phi_l \rangle$ and $K^\zeta_{kl} = \langle \frac{\partial \Phi_k}{\partial \zeta}, \Phi_l \rangle$, see Dumbser & Käser (2006), and the Kronecker symbol δ_{n1} as follows:

$$\frac{\partial}{\partial t} \hat{Q}_{pn}(t) = \left(- M^{-1}_{nk} K^\xi_{lk} \check{A}^*_{pq} - M^{-1}_{nk} K^\eta_{lk} \check{B}^*_{pq} - M^{-1}_{nk} K^\zeta_{lk} \check{C}^*_{pq} + \delta_{nl} \check{E}_{pq} \right) \hat{Q}_{ql}(t) \tag{36}$$

Eq. (36) is a system of ordinary differential equations that governs the time evolution of the degrees of freedom $\hat{Q}_{pn}(t)$ without taking into account effects from the element boundaries. However, it can be used in order to estimate the time evolution during one time step. The mth time derivative of $\hat{Q}_{pn}(t)$ is then given recursively by

$$\frac{\partial^m}{\partial t^m} \hat{Q}_{pn}(t) = \left(- M^{-1}_{nk} K^\xi_{lk} \check{A}^*_{pq} - M^{-1}_{nk} K^\eta_{lk} \check{B}^*_{pq} - M^{-1}_{nk} K^\zeta_{lk} \check{C}^*_{pq} + \delta_{nl} \check{E}_{pq} \right) \frac{\partial^{m-1}}{\partial t^{m-1}} \hat{Q}_{ql}(t), \qquad \forall m \geq 1. \tag{37}$$

The Taylor series for the degrees of freedom

$$\hat{Q}_{pn}(t) = \sum_{m=0}^{N} \frac{(t - t^n)^m}{m!} \frac{\partial^m}{\partial t^m} \hat{Q}_{pn}(t^n) \tag{38}$$

can be integrated analytically in time, and with $\Delta t = t^{n+1} - t^n$ we obtain:

$$\int_{t^n}^{t^{n+1}} \hat{Q}_{pn}(t) dt = \sum_{m=0}^{N} \frac{\Delta t^{m+1}}{(m+1)!} \frac{\partial^m}{\partial t^m} \hat{Q}_{pn}(t^n) := I_{pnql}(\Delta t)\, \hat{Q}_{ql}(t^n). \tag{39}$$

Eq. (39) together with eq. (37) can be seen as a discrete Cauchy–Kovalewski procedure for the system (33). Due to the linearity of the governing system, this new approach is equal to the use of the 4-D tensor as presented previously in Käser & Dumbser (2006) and Dumbser & Käser (2006). To clarify this new procedure of the high order ADER time integration, an example of a FORTRAN code is presented in Appendix B. There, we also show how to make use of the sparsity of the system matrices and stiffness matrices.

4 QUALITY FACTOR APPROXIMATION

In this section, we investigate the accuracy of the approximation of a given, generally constant, Q-law with respect to the number of relaxation mechanisms as described in Section 2. In fact, we clearly illustrate the improvement of the approximation of a frequency-independent Q-law as recognized by Liu *et al.* (1976) when increasing the number n of relaxation mechanisms. Furthermore, we analyse the corresponding, additional CPU time requirements, when different orders of accuracy of the ADER–DG schemes are used in combination with an increasing number of such mechanisms. Fig. 1 shows how a constant Q-law can be fitted by using (a) 2, (b) 3, (c) 5 or (d) 10 relaxation mechanisms on a frequency band of (0.1, 10) Hz. We point out that following Emmerich & Korn (1987) already 3 relaxation mechanisms approximate a constant, frequency-independent Q-law with a maximum deviation of around 5 per cent. Using only 2 relaxation mechanisms seems to be a too rough approximation whereas 5 or more mechanisms already lead to a Q-law approximations which might not even be necessary in most cases. The influence of the number of used relaxation mechanisms on seismograms recorded for an anelastic subsurface model is studied in Section 6.

Table 1 shows the increasing CPU time, when the number n of mechanisms is increased. The CPU times are normalized with respect to the purely elastic case, where no attenuation is incorporated, that is, $n = 0$. We remark that 3 mechanisms, as typically suggested in the literature for example, by Emmerich & Korn (1987) or Moczo *et al.* (1997), only increase the computational effort between a factor of 1.6 and 3.0 depending on the order of the used ADER–DG scheme. This efficiency is quite remarkable, in particular, as we treat the anelastic functions, that is, the anelastic part of eq. (13) as described in 3, with the same (full) order of accuracy. The results of a convergence study in the following Section 5 confirm that this way the scheme maintains its high order approximation properties.

5 CONVERGENCE STUDY

In this section, we present the results of a numerical convergence study to confirm the very high accuracy of the proposed ADER–DG method on tetrahedral meshes considering viscoelastic attenuation. We show results from second- to seventh-order ADER–DG schemes, which are

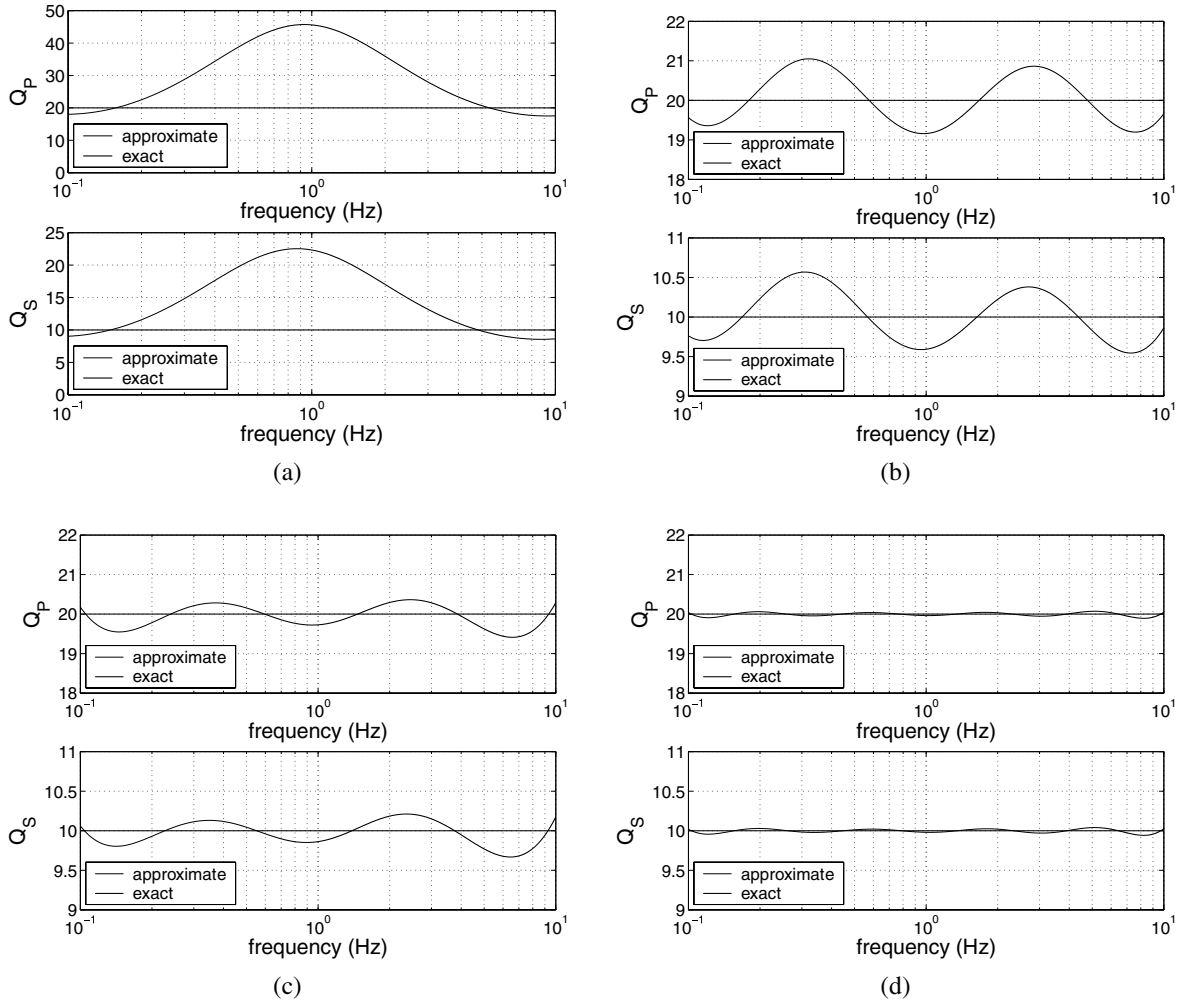

Figure 1. Approximation of frequency independent Q-factors using: (a) 2; (b) 3; (c) 5 or (d) 10 mechanisms on a frequency band of (0.1, 10) Hz.

Table 1. Evolution of the computational effort with respect to the purely elastic case ($n = 0$) with increasing number of mechanisms n for ADER–DG schemes from second to sixth order.

n	0	1	2	3	4	5	6	7	8	9	10
$\mathcal{O}2$	1.00	1.39	1.46	1.54	1.65	1.78	1.90	1.99	2.11	2.17	2.31
$\mathcal{O}3$	1.00	1.52	1.67	1.82	1.98	2.13	2.28	2.44	2.61	2.75	2.91
$\mathcal{O}4$	1.00	1.72	1.91	2.05	2.26	2.44	2.65	2.83	3.04	3.21	3.41
$\mathcal{O}5$	1.00	1.84	2.04	2.23	2.45	2.68	2.91	3.08	3.29	3.53	3.73
$\mathcal{O}6$	1.00	1.91	2.13	2.32	2.53	2.78	3.01	3.20	3.43	3.64	3.84

denoted by ADER–DG $\mathcal{O}2$ to ADER–DG $\mathcal{O}7$, respectively. Furthermore, we remark that the proposed ADER–DG schemes automatically obtain the same order for space and time.

To determine the convergence orders we solve the 3-D seismic wave equations (12) with viscoelastic attenuation in the unit-cube, that is, in a computational domain $\Omega = [-1, 1] \times [-1, 1] \times [-1, 1] \in \mathbb{R}^3$, as sketched in Fig. 2. At the boundaries of Ω we use periodic boundary conditions. The homogeneous material parameters are set to

$$\lambda = 2, \qquad \mu = 1, \qquad \rho = 1, \qquad Q_P = 20, \qquad Q_S = 10, \tag{40}$$

throughout the computational domain Ω. The $\mathcal{Q}$-factors are assumed to be frequency independent over the frequency band (0.1, 10) Hz. To this end, we are using 5 relaxation mechanisms, as introduced in Section 2, that lead to a satisfying approximation of a constant $\mathcal{Q}$-law as shown in Fig. 1(c). These material properties introduce damping and dispersion of the P and S waves.

For the convergence test, we use an initial condition representing a plane P-wave travelling along the space diagonal $\vec{d} = (1, 1, 1)^T$ of the domain Ω and a plane S-wave travelling in opposite direction as already shown in Dumbser & Käser (2006). The total simulation time T is set to $T = 0.1$ s. The CFL number is set in all computations to 50 per cent of the stability limit $\frac{1}{2N+1}$ of Runge–Kutta DG schemes. For a thorough investigation of the linear stability properties of the ADER–DG schemes based on a *von Neumann* stability analysis see Dumbser (2005).

In the following, we explain in detail how the initial condition and the analytic solution for the convergence test problem are found. We know, for example, from Stein & Wysession (2003), that the analytic solution to the plane wave problem has the form

$$Q_p(x, y, z, t) = Q_p^0 \cdot e^{i \cdot (\omega t - k_x x - k_y y - k_z z)}, \qquad p = 1, \ldots, n_v, \tag{41}$$

where Q_p^0 is the initial amplitude vector ω the angular frequency to determine, and

$$\vec{k} = (k_x, k_y, k_z)^T = (\pi, \pi, \pi)^T. \tag{42}$$

is the wavenumber vector. Eqs (41) and (42) lead to a periodic, plane sinusoidal wave in the unit-cube Ω with the wave front perpendicular to the cube's space diagonal.

In the following, we briefly explain how we determine the angular frequencies ω.

With the assumption that eq. (41) is the analytic solution of the governing equation (13), we calculate the first time and space derivatives of eq. (41) analytically and insert them into eq. (13). From there, we can derive an eigenproblem of the general form $M\vec{v} = \alpha\vec{v}$. Here, in particular, we get

$$(\check{A}_{pq} k_x + \check{B}_{pq} k_y + \check{C}_{pq} k_z - i \cdot \check{E}_{pq}) \cdot Q_q^0 = \omega \cdot Q_q^0, \qquad p, q = 1, \ldots, n_v. \tag{43}$$

Solving an eigenproblem means to find the p eigenvalues $\alpha^{(j)}$ and eigenvectors $\vec{v}^{(j)}$, $j = 1, \ldots, p$, of the square matrix $M \in \mathbb{R}^{p \times q}$. In our particular case of eq. (43), we need to find the eigenvalues $\omega^{(j)}$ and the matrix R_{pq} of right eigenvectors $\vec{r}_p^{(1)}, \ldots, \vec{r}_p^{(n_v)} \in \mathbb{R}^{n_v}$, with $p = 1, \ldots, n_v$.

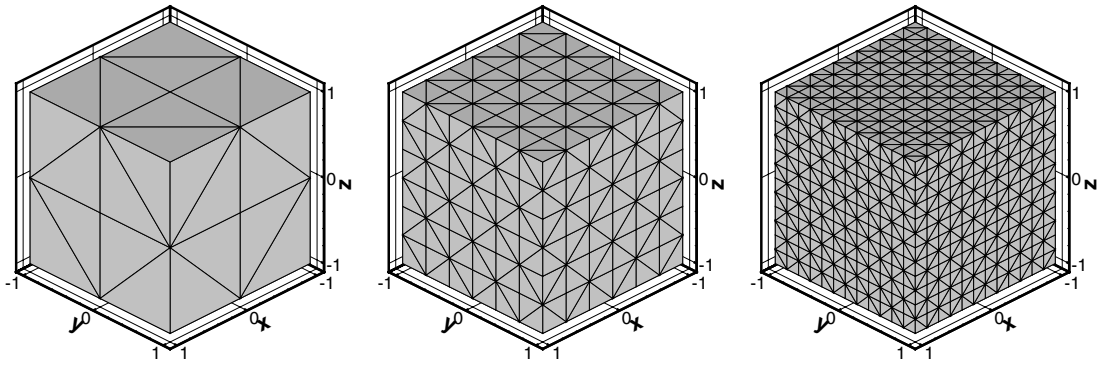

Figure 2. Sequence of discretizations of the computational domain Ω via regularly refined tetrahedral meshes, which are used for the numerical convergence analysis.

Table 2. Convergence rates of velocity component v of the ADER–DG $\mathcal{O}2$ up to ADER–DG $\mathcal{O}7$ schemes on tetrahedral meshes with viscoelastic attenuation.

h	L^∞	$\mathcal{O}_{L^\infty}$	L^2	$\mathcal{O}_{L^2}$	N_d	I	CPU (s)
1.08×10^{-1}	5.8094×10^{-3}	–	4.8622×10^{-3}	–	81 920	24	67
7.21×10^{-2}	2.5990×10^{-3}	2.0	2.1265×10^{-3}	2.0	276 480	36	341
5.41×10^{-2}	1.5287×10^{-3}	1.8	1.1775×10^{-3}	2.1	655 360	46	1043
4.33×10^{-2}	9.6624×10^{-4}	2.1	7.4891×10^{-4}	2.0	1280 000	58	2546
2.16×10^{-1}	5.1803×10^{-3}	–	3.2846×10^{-3}	–	25 600	20	12
1.08×10^{-1}	6.1874×10^{-4}	3.1	3.4224×10^{-4}	3.3	204 800	38	175
7.21×10^{-2}	1.6487×10^{-4}	3.3	1.0294×10^{-4}	3.0	691 200	58	857
5.41×10^{-2}	7.9007×10^{-5}	2.6	4.2568×10^{-5}	3.1	1638 400	76	2708
2.16×10^{-1}	5.4011×10^{-4}	–	3.2609×10^{-4}	–	51 200	28	35
1.44×10^{-1}	1.4012×10^{-4}	3.3	5.7198×10^{-5}	4.3	172 800	40	168
1.08×10^{-1}	4.3978×10^{-5}	4.0	1.7152×10^{-5}	4.2	409 600	54	504
7.21×10^{-2}	9.0642×10^{-6}	3.9	3.2404×10^{-6}	4.1	1382 400	80	2514
4.33×10^{-1}	1.8736×10^{-3}	–	8.2689×10^{-4}	–	11 200	18	7
2.16×10^{-1}	7.6374×10^{-5}	4.6	2.2952×10^{-5}	5.2	89 600	36	98
1.44×10^{-1}	9.2562×10^{-6}	5.2	2.8210×10^{-6}	5.2	302 400	52	482
1.08×10^{-1}	2.4829×10^{-6}	4.6	6.5480×10^{-7}	5.1	716 800	70	1483
8.66×10^{-1}	2.2965×10^{-2}	–	5.5321×10^{-3}	–	2240	12	1
4.33×10^{-1}	3.4744×10^{-4}	6.0	9.2044×10^{-5}	5.9	17 920	22	17
2.16×10^{-1}	6.4859×10^{-6}	5.7	1.3871×10^{-6}	6.1	143 360	42	259
1.44×10^{-1}	5.8794×10^{-7}	5.9	1.1658×10^{-7}	6.1	483 840	64	1318
8.66×10^{-1}	4.4014×10^{-3}	–	1.3209×10^{-3}	–	3360	14	3
4.33×10^{-1}	4.7643×10^{-5}	6.5	1.2218×10^{-5}	6.8	26 880	26	43
2.88×10^{-1}	3.2770×10^{-6}	6.6	5.8054×10^{-7}	7.5	90 720	38	213
2.16×10^{-1}	4.4764×10^{-7}	6.9	7.6709×10^{-8}	7.0	215 040	50	673

Recall that the solution of a linear hyperbolic system, as for example in eq. (13), is given by a linear combination of the right eigenvectors (Toro 1999). Therefore, the analytic solution $Q_p(x, y, z, t)$ in eq. (41) can be written as $Q_p = v_q \cdot R_{pq}$. The coefficients v_p can be computed via $v_q = R^{-1}_{qp}\ Q^0_p$ via the initial amplitude vector Q^0_p. Now, we can synthesize the analytic solution $Q_p(x, y, z, t)$ of the convergence test problem in the form

$$Q_p(x, y, z, t) = \sum_{j=1}^{n_v} v_j \cdot \vec{r}_p^{(j)}\, e^{i\cdot(\omega^{(j)}\cdot t - k_x x - k_y y - k_z z)} \qquad p = 1, \ldots, n_v. \tag{44}$$

In the special case of our initial condition, where one plane *P*-wave travels along the space diagonal $\vec{d} = (1, 1, 1)^T$ and one plane *S*-wave travels in the opposite direction, we only need two right eigenvectors. The initial condition for the convergence test problem is, therefore, given by eq. (44) using the two right eigenvectors $\vec{r}_p^{(2)}$ and $\vec{r}_p^{(9)}$, that is, $v_2 = v_9 = 1$ and zero otherwise.

To determine the convergence orders we calculate the solution of the same convergence test problem on a sequence of tetrahedral meshes as shown in Fig. 2. The mesh sequence is obtained by dividing the computational domain Ω into a number of subcubes, which are then subdivided into five tetrahedrons as shown in Fig. 2. We remark that this subdivision leads to four equal tetrahedrons with $1/6$ of the cube's volume and one regular central tetrahedron of $1/3$ of the cube's volume. This way, the refinement level is controlled by changing the number of subcubes in each space dimension.

Now, we can arbitrarily pick one of the variables of the vector Q_p of the seismic wave equations (13) to numerically determine the convergence order of the used ADER–DG scheme. In Table 2, we show the errors for the shear stress component σ_{yz}. The errors of the numerical solution Q_h with respect to the exact solution Q_e obtained from eq. (44) is measured in the L^∞-norm and the continuous L^2-norm

$$E^s_{L^2} = \|Q_h - Q_e\|_{L^2(\Omega)} = \left(\int_\Omega |Q_h - Q_e|^2\, dV \right)^{\frac{1}{2}}, \tag{45}$$

where the integration is approximated by Gaussian integration which is exact for a polynomial degree twice that of the basis functions of the numerical scheme. The L^∞-norm is approximated by the maximum error arising at any of these Gaussian integration points. The convergence orders are then computed through

$$\mathcal{O}_{L^\nu} = \log\left(\frac{E^s_{L^\nu}}{E^{s-1}_{L^\nu}}\right) \Big/ \log\left(\frac{h^s}{h^{s-1}}\right), \qquad \text{with} \qquad \nu = 2, \infty, \tag{46}$$

where h^s indicates the mesh spacing h of mesh number s in the sequence of meshes.

The first column in Table 2 shows the mesh spacing h, represented by the maximum diameter of the circumscribed spheres of the tetrahedrons. The following four columns show the L^∞ and L^2 errors with the corresponding convergence orders $\mathcal{O}_{L^\infty}$ and $\mathcal{O}_{L^2}$ determined by successively refined meshes. Furthermore, we present the total number N_d of degrees of freedom, which is a measure of required storage

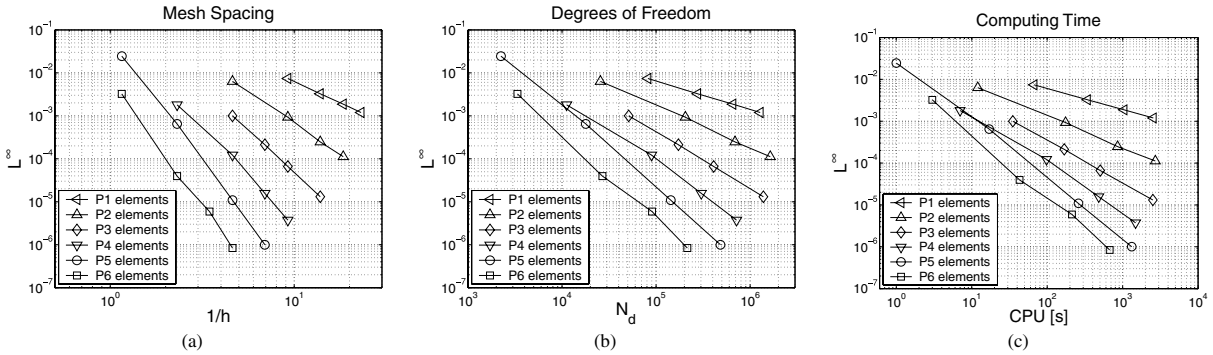

Figure 3. Convergence rates of velocity component v of Table 2. The L^∞ error is plotted versus (a) the mesh spacing h, (b) the number of degrees of freedom N_d and (c) the CPU time.

space during run time and is given through the product of the number of total mesh elements and the number N_e of degrees of freedom per element. N_e depends on the order of the scheme, that is, the degree N of the polynomial basis functions via $N_e(N) = \frac{1}{6}(N+1)(N+2)(N+3)$. In the last two columns we give the number I of iterations and the CPU times in seconds needed to reach the simulation time $T = 0.1$s on one Pentium Xeon 3.6 GHz processor with 4GB of RAM.

In Fig. 3, we visualize the convergence results of Table 2 to demonstrate the dependences of the L^∞ error with respect to (a) mesh width h, (b) number of degrees of freedom N_d and (c) CPU time. With mesh refinement, the higher order schemes converge faster towards the analytic solution as shown in Fig. 3(a). Furthermore, Fig. 3(b) illustrates that higher order schemes reach a desired accuracy requiring a lower number of total degrees of freedom. The total number of degrees of freedom is the product of the number of mesh elements and the degrees of freedom per element. Therefore, obviously the increasing number of degrees of freedom per element is overcompensated by the dramatic decrease of the number of required elements to reach a certain error level. The CPU time comparisons in Fig. 3(c) also illustrate that for higher accuracy, that is, smaller errors, are reached in less computational time when using a higher order ADER–DG scheme. We remark that in all three plots of Fig. 3 we clearly show that for very high accuracy, the higher order schemes pay off due to their superior convergence properties.

6 APPLICATION EXAMPLE

We apply the proposed ADER–DG method to a well-defined 3-D test problem, which was published in the final report of the *LIFELINES PROGRAM TASK 1A02* (Day *et al.* 2003) of the Pacific Earthquake Engineering Research Center. The test case is part of a multi-institutional code validation project of a series of different numerical methods employed in numerical modelling of earthquake ground motion in 3-D Earth models. Therefore, besides a quasi-analytic solution, simulation results from four different well-established codes exist and serve as additional reference solutions. The results of these four codes are denoted by four-character abbreviations indicating the respective institutions:

UCBL (Doug Dreger and Shawn Larsen, University of California, Berkeley/Lawrence Livermore National Laboratory),
UCSB (Kim Olsen, University of California, Santa Barbara),
WCC2 (Arben Pitarka, URS Corporation), and
CMUN (Jacobo Bielak, Carnegie-Mellon University).

The first three codes use the finite-difference method on uniform, structured grids with staggered locations of the velocity and stress components and fourth order accuracy in space. The CMUN code uses piecewise linear interpolation on unstructured tetrahedral FEs.

The quasi-analytic solution is a frequency-wavenumber solution obtained by a modification of the method presented in Luco & Apsel (1983) and Apsel & Luco (1983) and is compared to all numerical solutions to evaluate their accuracy. The setup of the test problem LOH.3 (layer over half-space) is shown in Fig. 4(a), where for clarity only one of four symmetrical quarters of the complete computational domain $\Omega = [-15\,000m, 15\,000m] \times [-15\,000m, 15\,000m] \times [0m, 17\,000m]$ is plotted. The material parameters of the attenuating layer (Medium 1) of the top $1000m$ over the attenuating half-space (Medium 2) are given in Table 3.

The seismic source is a point dislocation, represented by a double-couple source, where the only non-zero entries of the seismic moment tensor are $M_{xy} = M_{yx} = M_0 = 10^{18}Nm$. The location of the point source is $(x_s, y_s, z_s) = (0m, 0m, 2000m)$, that is, in the centre of the xy plane of the domain Ω in $2000m$ depth.

The moment-rate time history is given through the source time function

$$S^T(t) = \frac{t}{T^2} \exp\left(-\frac{t}{T}\right), \tag{47}$$

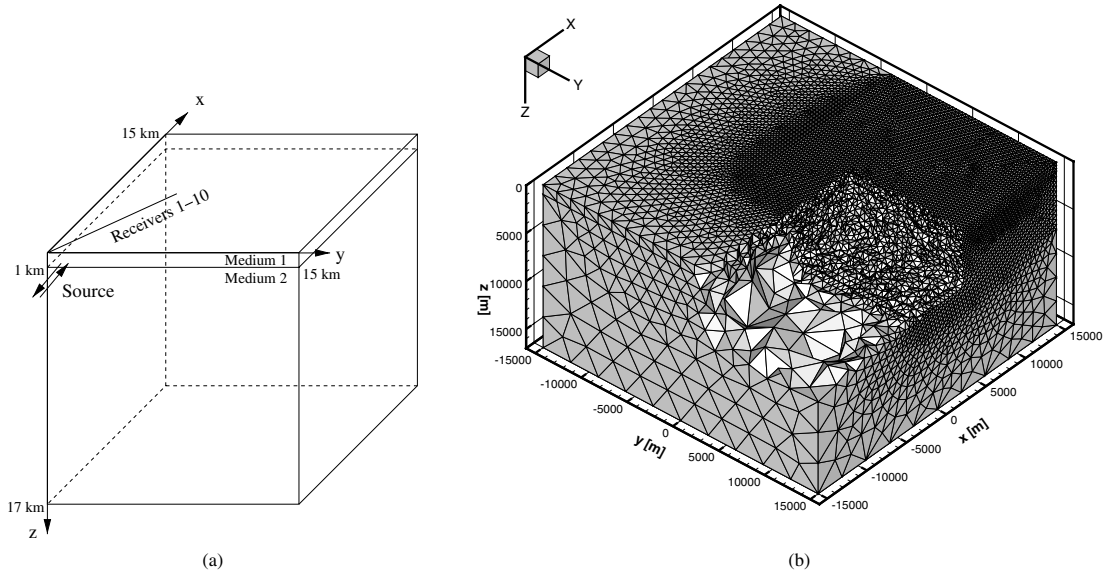

Figure 4. (a) One of four symmetric quarters is shown for the LOH.3 test case, where a layer of 1 km (Material 1) is lying on top of another layer (Material 2). The source is a point dislocation at 2000 km depth represented by a moment tensor with the only non-zero components $M_{xy} = M_{yx}$. (b) Cut into the discretization of the LOH.3 model to visualize the problem-adapted tetrahedral mesh, which is refined in the quarter under the receiver line to a depth of 6000 m.

Table 3. Material parameters for the LOH.3 test case. Note that attenuation will cause dispersion of the P and S waves such that the given wave speeds refer to a reference frequency $f_r = 2.5$ Hz.

	$c_p(f_r)$ (m s^{-1})	$c_s(f_r)$ (m s^{-1})	ρ (kg m^{-3})	$\lambda(f_r)$ (GPa)	$\mu(f_r)$ (GPa)	Q_p	Q_s
Medium 1	4000	2000	2600	20.8	10.4	120	40
Medium 2	6000	3464	2700	32.4	32.4	155.9	69.3

where the smoothness parameter T, controlling the frequency content and amplitude of the source time function, is set to $T = 0.1$ s. We remark that details of the discretization of external source terms in the framework of ADER–DG methods are shown in previous work (Käser & Dumbser 2006).

The signals are calculated up to a simulation time of 9 s by 10 receivers on the free surface as indicated in Fig. 4(a). The receiver locations $(x_i, y_i, z_i) = (i \cdot 600m, i \cdot 800m, 0m)$, for $i = 1, \ldots, 10$.

The computational domain Ω is discretized by an unstructured, tetrahedral mesh of 249 338 elements. The mesh is illustrated in Fig. 4(b). Furthermore, the mesh is generated in a problem-adapted manner. To this end, in the zone of interest the waves travelling from the source to the receivers pass through tetrahedral elements with an average edge length of 350 m, whereas in other zones the mesh is coarsened up to average edge lengths of 3000 m to reduce the number of total elements and, therefore, computational cost. We remark that neither the source location nor the receiver locations have to coincide with nodes of the tetrahedral mesh, as in the ADER–DG framework the numerical solution is represented by polynomials within each element and, therefore, can be evaluated at any position within an element as shown in Käser & Dumbser (2006). This greatly simplifies the process of mesh generation and does not restrict the desired flexibility provided by unstructured meshes. However, the mesh respects the material interface between Medium 1 and Medium 2 as the faces of the tetrahedral elements are aligned with the material interface as shown in Figs 4(a) and (b).

In the following, we present the comparison of our results obtained by an ADER–DG $\mathcal{O}4$ and ADER–DG $\mathcal{O}5$ scheme and the four results of the reference codes (UCBL, UCSB, WCC2 and CMUN) against the analytical solution. Analogous to the LOH.3 test case in the *LIFELINES PROGRAM TASK* 1*A*02 the visual comparisons in Fig. 5 show the radial, transversal and vertical components of the seismic velocity field recorded at receiver 10 at $(x_{10}, y_{10}, z_{10}) = (6000m, 8000m, 0m)$. Additionally, each plot gives the relative seismogram misfit

$$E = \sum_{j=1}^{n_t} \left(s_j - s_j^a\right)^2 \Big/ \sum_{j=1}^{n_t} \left(s_j^a\right)^2, \tag{48}$$

where n_t is number of time samples of the seismogram, s_j is the numerical value of the particular seismogram at sample j and s_j^a is the corresponding analytical value. As shown by Kristeková *et al.* (2006) our misfit criterion only quantifies the misfit between two signals in the case of a pure amplitude modification of the whole signal. However, we use this simple criterion given by eq. (48) as it is not crucial in this case to assess the accuracy of the proposed method. Furthermore, it allows for the direct comparison to results of previous work by Dumbser & Käser (2006) as well as to the results obtained by the ADER Finite Volume scheme (Dumbser *et al.* 2006) for the same test problem.

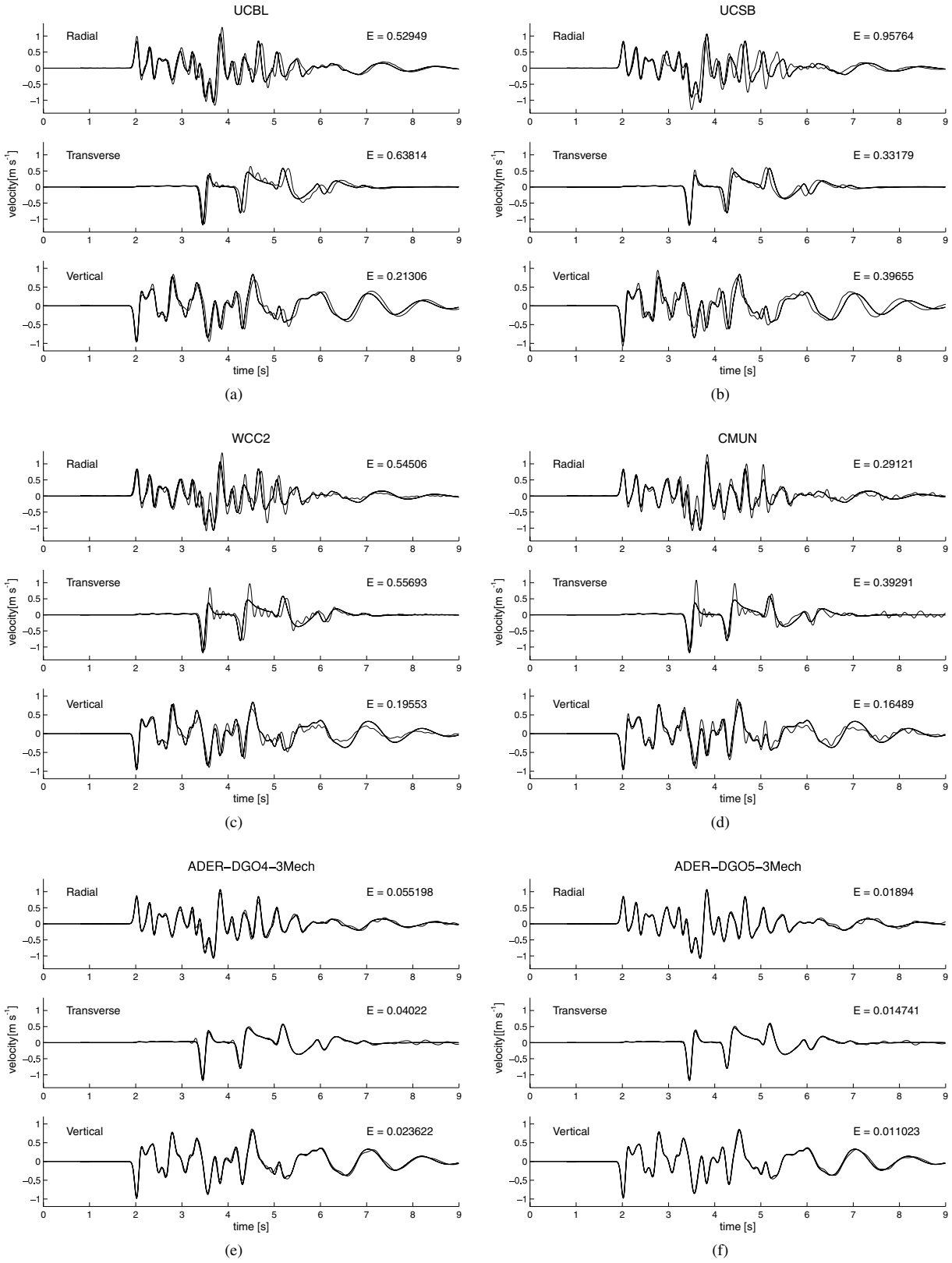

Figure 5. Comparison of the radial, transverse and vertical velocity components for the LOH. 3 test case on receiver 10. The analytical solution (thick line) is plotted against the numerical one (thin line) obtained by (a) UCBL, (b) UCSB, (c) WCC2, (d) CMUN, (e) ADER–DG $\mathcal{O}4$ with 3 relaxation mechanisms and (f) ADER–DG $\mathcal{O}5$ with 3 relaxation mechanisms. The relative seismogram misfit E from eq. (48) is given for each trace.

We remark that for all shown seismograms, the original source was deconvolved and replaced by a Gaussian of spread 0.05 as described in Day *et al.* (2003).

The four reference solutions shown in Figs 5(a)–(d) remarkably differ from each other due to the different ways of incorporating viscoelastic attenuation. Amplitude errors (e.g. for CMUN) and phase errors (e.g. for UCSB) are quite noticeable. In addition, the results of UCBL, WCC2 and CMUN produce strong, unphysical oscillations in the transverse component.

The results with the fourth-order ADER–DG $\mathcal{O}4$ scheme in Fig. 5(e) using three relaxation mechanisms clearly match the analytic solution better and show lower numbers for the misfit E. Furthermore, the appearing but rather small errors can even be dramatically decreased by using the higher order ADER–DG $\mathcal{O}5$ scheme in Fig. 5(f). As seen also in the convergence results of Table 2, the ADER–DG $\mathcal{O}5$ simulation took about four times longer than the ADER–DG $\mathcal{O}4$. Unfortunately, run times for the reference solutions are neither given in the report (Day *et al.* 2003) nor could they be found elsewhere. We also remark that the weak oscillations after 6 s in our results might be due to reflections from the model boundaries as we are not using sophisticated conditions for absorbing, non-reflecting boundaries but control them via fluxes as described in detail in Käser & Dumbser (2006) and Dumbser & Käser (2006).

Fig. 6(a)–(d) presents the influence of the number of relaxation mechanisms on the seismograms. Using an ADER–DG $\mathcal{O}4$ scheme with 3 or 5 mechanisms basically does not affect the seismogram accuracy as shown in Figs 6(a) and (c). Therefore, 3 mechanisms already seem to approximate the constant Q-law with sufficient accuracy, which agrees with observations made in the literature (for example, Emmerich & Korn 1987; Moczo *et al.* 1997). However, results of the ADER–DG $\mathcal{O}5$ schemes with 2 or 3 mechanisms in Figs 6(b) and (d) clearly illustrate that 2 mechanisms do not accurately represent a constant Q-law as already mentioned in Section 4. Hereby we confirm that amplitude and phase errors visible in Fig. 6(d) have to be due to the insufficient Q-law approximation instead of lacking accuracy of the numerical scheme itself. Therefore, we claim that in general 3 relaxation mechanisms might be enough to incorporate viscoelastic attenuation in many cases.

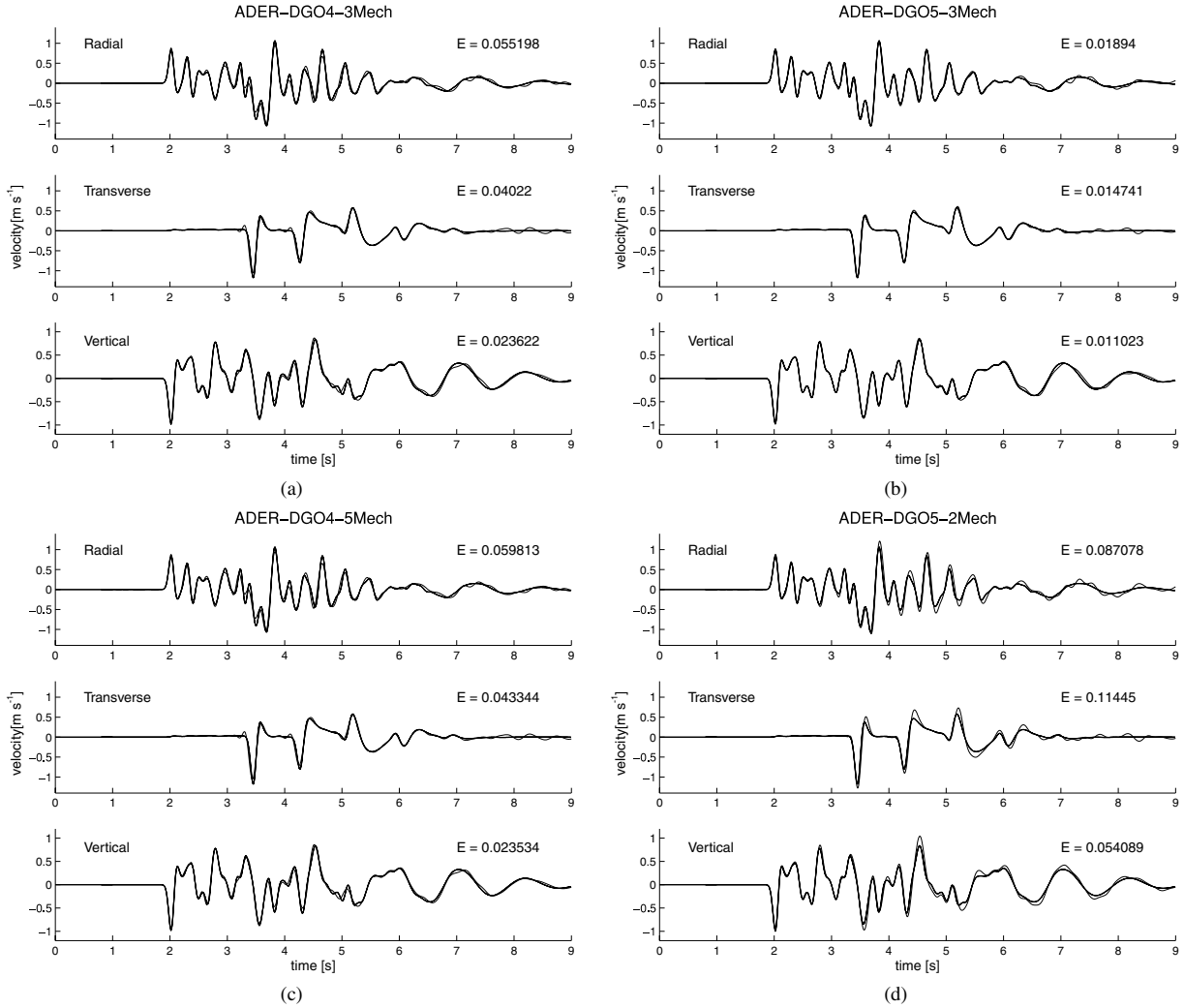

Figure 6. Comparison of the radial, transverse and vertical velocity components for the LOH.3 test case on receiver 10. The analytical solution (thick line) is plotted against the numerical one (thin line) obtained by (a) ADER–DG $\mathcal{O}4$ with 3 relaxation mechanisms, (b) ADER–DG $\mathcal{O}5$ with 3 relaxation mechanisms, (c) ADER–DG $\mathcal{O}4$ with 5 relaxation mechanisms and (d) ADER–DG $\mathcal{O}5$ with 2 relaxation mechanisms. The relative seismogram misfit E from eq. (48) is given for each trace.

7 CONCLUSION

We have presented the incorporation of realistic attenuation of seismic waves into the new ADER–Discontinuous Galerkin (ADER–DG) schemes using viscoelastic material. The proposed numerical method can approximate the seismic wavefield, that is, stresses and velocities, with arbitrarily high order of accuracy in space and time on unstructured 3-D tetrahedral meshes. The additional variables, the anelastic functions, can be treated similarly in the case of viscoelastic material. Therefore, the linear hyperbolic system of the seismic wave equations increases with the number of relaxation mechanisms and includes source terms resulting from the approximating viscoelastic material behaviour by a generalized Maxwell body. However, the introduction of a new Cauchy–Kovalewski procedure for the high order ADER time integration results in a more efficient implementation and, therefore, does not increase the computation time dramatically when incorporating viscoelastic attenuation. The convergence results demonstrate the high accuracy of the ADER–DG schemes on tetrahedral meshes. In addition, the detailed investigation of the required number of relaxation mechanisms agrees with the suggestions in the literature, that 3 mechanisms seem to be sufficient for the accurate incorporation of realistic attenuation. The solution of a well-established benchmark test with different orders of accuracy of the ADER–DG schemes and the comparison of the obtained results against analytic solutions clearly shows the dramatic increase in accuracy with respect to reference solutions obtained by other methods. Therefore, the proposed ADER–DG methods represents a new numerical scheme simulating seismic wave propagation with unprecedented accuracy on unstructured 3-D tetrahedral meshes thoroughly including realistic attenuation due to viscoelasticity.

ACKNOWLEDGMENTS

The work was supported by the DFG (Deutsche Forschungsgemeinschaft) through the Emmy Noether-program (KA 2281/1-1) and the DFG-CNRS research group FOR 508, Noise Generation in Turbulent Flows and by the European Marie-Curie Research and Training Network SPICE (Seismic Wave Propagation in Complex Media: a European Network) within the Human Mobility Programme. The support and comments by Steven Day in setting up the LOH.3 test case and providing the analytical and reference solutions are highly acknowledged. We also thank Peter Moczo and Geza Seriani for the helpful support when incorporate viscoelastic attenuation. The support of the super-computing facilities of the HLRS in Stuttgart and the LRZ in München are highly appreciated for performing the calculations presented in this article.

REFERENCES

Apsel, R.J. & Luco, J.E., 1983. On the Green's functions for a layered half-space. Part II, *Bull. seism. Soc. Am.,* **73,** 931–951.

Bouchon, M., 1981. A simple method to calculate Green's functions for elastic layered media, *Bull. seism. Soc. Am.,* **71,** 959–971.

Carcione, J.M. & Cavallini, F., 1994. A rheological model for anelastic anisotropic media with applications to seismic wave propagation, *Geophys. J. Int.,* **119,** 338–348.

Carcione, J.M., Kosloff, D. & Kosloff, R., 1988. Wave propagation simulation in a linear viscoelastic medium, *Geophys. J.,* **95,** 597–611.

Cockburn, B., Karniadakis, G.E. & Shu, C.W., 2000. *Discontinuous Galerkin Methods, Theory, Computation and Applications*, LNCSE, Vol. 11, Springer, Berlin, Germany.

Day, S.M., 1998. Efficient simulation of constant Q using coarse-grained memory variables, *Bull. seism. Soc. Am.,* **88,** 1051–1062.

Day, S.M., Bielak, J., Dreger, D., Graves, R., Larsen, S., Olsen, K.B. & Pitarka, A., 2003. *Tests of 3D Elastodynamic Codes: Final Report for Lifelines Project 1A02,* Pacific Earthquake Engineering Research Center, October 10.

Day, S.M. & Bradley, C., 2001. Memory-efficient simulation of anelastic wave propagation, *Bull. seism. Soc. Am.,* **91,** 520–531.

Day, S.M. & Minster, J.B., 1984. Numerical simulation of wave-fields using a Padé approximant method, *Geophys. J. R. astr. Soc.,* **78,** 105–118.

Dumbser, M., 2005. *Arbitrary High Order Schemes for the Solution of Hyperbolic Conservation Laws in Complex Domains,* Shaker Verlag, Aachen.

Dumbser, M. & Käser, M., 2006. An arbitrary high order Discontinuous Galerkin method for elastic waves on unstructured meshes II: the three-dimensional isotropic case, *Geophys. J. Int.,* **167,** 319–336.

Dumbser, M., Käser, M. & de la Puente, J., 2006. Arbitrary high order finite volume schemes for seismic wave propagation on unstructured meshes in 2D and 3D, submitted.

Dumbser, M. & Munz, C.D., 2005a. Arbitrary high order Discontinuous Galerkin schemes, in *Numerical Methods for Hyperbolic and Kinetic Problems,* pp. 295–333, eds. Cordier, S., Goudon, T., Gutnic, M. & Sonnendrucker, E., IRMA series in mathematics and theoretical physics, EMS Publishing House, Zürich.

Dumbser, M. & Munz, C.D., 2005b. ADER Discontinuous Galerkin Schemes for Aeroacoustics, in *Comptes Rendus Mécanique,* **333,** 683–687.

Dumbser, M. & Munz, C.D., 2006. Building blocks for arbitrary high order Discontinuous Galerkin schemes, *J. Sci. Comput.,* **27,** 215–230.

Emmerich, H., 1992. PSV-wave propagation in a medium with local heterogeneities: a hybrid formulation and its application, *Geophys. J. Int.,* **109,** 54–64.

Emmerich, H. & Korn, M., 1987. Incorporation of attenuation into time-domain computations of seismic wave fields, *Geophysics,* **52,** 1252–1264.

Graves, R.W. & Day, S.M., 2003. Stability and accuracy analysis of coarse-grain viscoelastic simulations, *Bull. seism. Soc. Am.,* **93,** 283–300.

Käser, M. & Dumbser, M., 2006. An arbitrary high order Discontinuous Galerkin method for elastic waves on unstructured meshes I: the two-dimensional isotropic case with external source terms, *Geophys. J. Int.,* **166,** 855–877.

Kristek, J. & Moczo, P., 2003. Seismic-wave propagation in viscoelastic media with material discontinuities: a 3D fourth-order staggered-grid finite-difference modeling, *Bull. seism. Soc. Am.,* **93,** 2273–2280.

Kristeková, M., Kristek, J., Moczo, P. & Day S.M., 2006. Misfit criteria for quantitative comparison of seismograms, *Bull. seism. Soc. Am.,* in press.

Liu, H.P., Anderson, D.L. & Kanamori, H., 1976. Velocity dispersion due to anelasticity; implications for seismology and mantle composition, *Geophys. J. R. astr. Soc.,* **47,** 41–58.

Luco, J.E. & Apsel, R.J., 1983. On the Green's functions for a layered half-space. Part I, *Bull. seism. Soc. Am.,* **73,** 909–929.

Marfurt, K.J., 1984. Accuracy of finite-difference and finite-element modeling of the scalar and elastic wave equations, *Geophysics,* **49,** 533–549.

Moczo, P. & Bard, P.Y., 1993. Wave diffraction, amplification and differential motion near strong lateral discontinuities, *Bull. seism. Soc. Am.,* **83,** 85–106.

Moczo, P., Bystricky, E., Kristek, J., Carcione, J.M. & Bouchon, M., 1997. Hybrid modeling of P-{SV} seismic motion at inhomogeneous viscoelastic topographic structures, *Bull. seism. Soc. Am.,* **87,** 1305–1323.

Moczo, P. & Kristek, 2005. On the rheological models used for time-domain methods of seismic wave propagation, *Geophys. Res. Lett.,* **32,** L01306, doi:10.1029/2004GL021598.

Moczo, P., Kristek, J. & Halada, L., 2004. *The Finite Difference Method for Seismologists. An Introduction,* Comenius University, Bratislava, available at http://www.spice-rtn.org/

Reed, W.H. & Hill, T.R., 1973. Triangular mesh methods for the neutron transport equation, Technical Report, LA-UR-73-479, Los Alamos Scientific Laboratory.

Stein, S. & Wysession, M., 2003. *An Introduction to Seismology, Earthquakes and Earth Structure,* Blackwell Publishing, Malden, MA.

Titarev, V.A. & Toro, E.F., 2002. ADER: Arbitrary high order Godunov approach, *J. Sci. Comput.,* **17,** 609–618.

Toro, E.F., 1999. *Riemann Solvers and Numerical Methods for Fluid Dynamics,* Springer, Berlin.

Toro, E.F., Millington, A.C. & Nejad, L.A., 2001. Towards very high order Godunov schemes, in *Godunov Methods; Theory and Applications,* pp. 907–940, Kluwer Academic Plenum Publishers, Oxford.

Toro, E.F. & Titarev, V.A., 2002. Solution of the generalized Riemann problem for advection-reaction equations, *Proc. R. Soc. London,* **458,** 271–281.

APPENDIX A: ORTHOGONAL BASIS FUNCTIONS

We use orthogonal hierarchical basis functions as given in Cockburn *et al.* (2000). The basis functions are given in terms of the Jacobi polynomials $P_n^{\alpha,\beta}(x)$, which are solutions of the Jacobi differential equation:

$$(1-x^2)y'' + [\beta - \alpha - (\alpha+\beta+2)\,x]\,y' + n\,(n+\alpha+\beta+1)\,y = 0. \tag{A1}$$

They are given on the interval $[-1; 1]$ by

$$P_n^{\alpha,\beta}(x) = \frac{(-1)^n}{2^n n!}\,(1-x)^{-\alpha}\,(1+x)^{-\beta}\,\frac{d^n}{dx^n}\left[(1-x)^{\alpha+n}\,(1+x)^{\beta+n}\right]. \tag{A2}$$

For $\alpha = \beta = 0$ the Jacobi polynomials $P_n^{0,0}(x)$ reduce to the Legendre polynomials. The DG basis functions are then constructed using the three primal functions

$$\Theta_i^a(x) = P_i^{0,0}(x), \qquad \Theta_{ij}^b(x) = \left(\frac{1-x}{2}\right)^i P_j^{2i+1,0}(x), \qquad \Theta_{ijk}^c(x) = \left(\frac{1-x}{2}\right)^{i+j} P_k^{2i+2j+2,0}(x). \tag{A3}$$

The sets of basis functions Φ_k constitute orthogonal basis systems with respect to the inner product on the respective reference elements $\mathcal{T}_E$.

For tetrahedrons the reference element $\mathcal{T}_E$ is defined as

$$\mathcal{T}_E = \{(\xi,\eta,\zeta) \in \mathbb{R}^3 \mid 0 \le \xi \le 1 \ \wedge\ 0 \le \eta \le 1-\xi \ \wedge\ 0 \le \zeta \le 1-\xi-\eta\}. \tag{A4}$$

The basis functions $\Phi_k(\xi,\eta,\zeta)$ are defined on this reference element as the following product of the primal functions:

$$\Phi_{k(p,q,r)}(\xi,\eta,\zeta) = \Theta_p^a(\alpha)\cdot\Theta_{pq}^b(\beta)\cdot\Theta_{pqr}^c(\gamma). \tag{A5}$$

with

$$\alpha = \frac{\eta-1+\zeta+2\xi}{1-\eta-\zeta}, \quad \beta = \frac{2\eta-1+\zeta}{1-\zeta}, \quad \gamma = -1+2\zeta. \tag{A6}$$

The mono-index $k = k(p,q,r)$ is again a function of the index triple (p,q,r). The 3-D basis functions up to degree two for a third order scheme are:

$$\begin{aligned}
\Phi_0 &= 1,\\
\Phi_1 &= -1+2\xi+\eta+\zeta,\\
\Phi_2 &= -1+3\eta+\zeta,\\
\Phi_3 &= -1+4\zeta,\\
\Phi_4 &= 1-6\xi+6\xi^2-2\eta+6\xi\eta+\eta^2-2\zeta+6\xi\zeta+2\eta\zeta+\zeta^2,\\
\Phi_5 &= 1-2\xi-6\eta+10\xi\eta+5\eta^2-2\zeta+2\xi\zeta+6\eta\zeta+\zeta^2,\\
\Phi_6 &= 1-8\eta+10\eta^2-2\zeta+8\eta\zeta+\zeta^2,\\
\Phi_7 &= 1-2\xi-\eta-7\zeta+12\xi\zeta+6\eta\zeta+6\zeta^2,\\
\Phi_8 &= 1-3\eta-7\zeta+18\eta\zeta+6\zeta^2,\\
\Phi_9 &= 1-10\zeta+15\zeta^2.
\end{aligned} \tag{A7}$$

APPENDIX B: FORTRAN EXAMPLE CODE FOR THE CAUCHY–KOVALEWSKI PROCEDURE USING SPARSE MATRICES

```
TYPE tSparseMatrix                                                    ! Type, defining a sparse matrix
  INTEGER, ALLOCATABLE :: nNonZero                                    ! Number of non-zero entries
  INTEGER, ALLOCATABLE :: NonZeroIndex1(:),NonZeroIndex2(:)           ! Row and column indices
  REAL, ALLOCATABLE :: NonZero(:)                                     ! Non-zero values
END TYPE tSparseMatrix                                                !
!                                                                     !
! The subroutine SPT_MATMUL multiplies a sparse matrix A from the left to the transpose  !
! of a full matrix B and saves the negative result in the transposed matrix C:           !
! C_ki = - A_ij .B_kj                                                 !
!                                                                     !
PURE SUBROUTINE SPT_MATMUL(C, A, B, n, m)                             !
  IMPLICIT NONE                                                       !
  ! Argument list declaration                                         !
  TYPE(tSparseMatrix) :: A                                            ! Sparse matrix of dimension (n,n)
  INTEGER :: n,m                                                      ! Matrix dimensions n and m
  REAL :: C(m,n)                                                      ! Result matrix of dimension (m,n)
  REAL :: B(m,n)                                                      ! Full matrix of dimension (m,n)
  ! Local variable declaration                                        !
  INTEGER :: i,j,iNonZero                                             ! Loop counters
  INTENT(IN) :: n,m,A,B                                               ! Intent definition for the arguments
  INTENT(OUT) :: C                                                    ! Intent definition for the arguments
  C = 0.                                                              ! Initialize C
  DO iNonZero = 1, A per centnNonZero                                 ! Loop over all non-zero entries of A
    i = A per centNonZeroIndex1(iNonZero)                             ! Get row index
    j = A per centNonZeroIndex2(iNonZero)                             ! Get column index
    C(:,i) = - B(:,j)*A per centNonZero(iNonZero)                     ! Multiply and subtract
  ENDDO                                                               !
END SUBROUTINE SPT_MATMUL                                             !
!                                                                     !
! The subroutine SPL_MATMUL multiplies a sparse matrix A from the left to a  !
! full matrix B and adds the result to the matrix C:                  !
! C_ik = C_ik + A_ij .B_jk                                            !
!                                                                     !
PURE SUBROUTINE SPL_MATMUL(C, A, B, n, m)                             !
  IMPLICIT NONE                                                       !
  ! Argument list declaration                                         !
  TYPE(tSparseMatrix) :: A                                            ! Sparse matrix of dimension (n,n)
  INTEGER :: n,m                                                      ! Matrix dimensions n and m
  REAL :: C(m,n)                                                      ! Result matrix of dimension (m,n)
  REAL :: B(m,n)                                                      ! Full matrix of dimension (m,n)
  ! Local variable declaration                                        !
  INTEGER :: i,j,iNonZero                                             ! Loop counters
  REAL :: CT(n,m)                                                     ! Transpose of matrix C
  REAL :: BT(n,m)                                                     ! Transpose of matrix B
  INTENT(IN) :: n,m,A,B                                               ! Intent definition for the arguments
  INTENT(INOUT) :: C                                                  ! Intent definition for the arguments
  CT = 0.                                                             ! Initialize CT
  BT = TRANSPOSE(B)                                                   ! Transpose B to have fast index first
  DO iNonZero = 1, A per centnNonZero                                 ! Loop over all non-zero entries of A
    i = A per centNonZeroIndex1(iNonZero)                             ! Get row index
    j = A per centNonZeroIndex2(iNonZero)                             ! Get column index
    CT(:,i) = CT(:,i) - BT(:,j)*A per centNonZero(iNonZero)           ! Multiply and subtract
  ENDDO                                                               !
  C = C + TRANSPOSE(CT)                                               ! Add transpose of CT to C
END SUBROUTINE SPL_MATMUL                                             !
! Subroutine doing the CK Procedure for the linear system with source term:  !
! u_t + Au_x + Bu_y + Cu_z = - (- E) u                                !
!                                                                     !
```

```
SUBROUTINE CauchyKovalewskiLinear3D(TimeIntDOF,TaylorDOF,DOF,t1,t2, &          !
nVar, nPoly,nDegFr,A,B,C,E )                                                    !
  IMPLICIT NONE                                                                 !
  ! Argument list declaration                                                   !
  REAL :: t1, t2                                                                ! t^n, t^{n+1}
  REAL :: nVar                                                                  ! Number of variables of the system
  REAL :: nPoly                                                                 ! Degree of the basis polynomials
  REAL :: nDegFr                                                                ! Number of DOF per element
  TYPE(tSparseMatrix) :: KxiTM,KetaTM,KzetaTM                                   ! Sparse M_nk^-1 K_lk^xi, M_nk^-1 K_lk^eta, M_nk^-1 K_lk^zeta
  TYPE(tSparseMatrix) :: A,B,C,E                                                ! Sparse Ǎ*_pq, B̌*_pq, Č*_pq and -Ě_pq
  REAL :: TimeIntDOF(nDegFr,nVar)                                               ! I_pnql(Δt) Q̂_ql(t^n)
  REAL :: TaylorDOF(nDegFr,nVar,0:nPoly)                                        ! ∂^m/∂t^m Q̂_pn(t)dt
  REAL :: DOF(nDegFr,nVar)                                                      ! Q̂_pl(t^n)
  ! Local variable declaration                                                  !
  INTEGER :: m                                                                  ! Derivative order
  INTEGER :: iVar,iDegFr                                                        ! Index of degree of freedom
  REAL :: Temp(nDegFr,nVar)                                                     ! Temporary variable
  REAL :: dtk(0:LocPoly)                                                        ! Auxiliary variables for Taylor series
  INTENT(IN) :: nVar,nPoly,nDegFr,KxiTM,KetaTM,KzetaTM,A,B,C,E,DOF,t1,t2       ! Intent definition for the arguments
  INTENT(OUT) :: TimeIntDOF, TaylorDOF                                          ! Intent definition for the arguments
  !                                                                             !
  TaylorDOF(:,:,0) = DOF(:,:)                                                   ! Zeroth time derivative
  DO m = 0, nPoly - 1                                                           !
  ! Start with source term contribution -δ_nl(-Ě_pq)                            !
    CALL SPT_MATMUL(Temp, E, TaylorDOF(:,:,m), nVar, nDegFr)                    ! Multiply Ě_pq with ∂^m/∂t^m Q̂_pl(t^n)
    TaylorDOF(:,:,m + 1) = Temp(:,:)                                            ! Store result in derivative of order m + 1
    ! Contribution -M_nk^-1 K_lk^xi Ǎ*_pq                                       !
    CALL SPT_MATMUL(Temp, A, TaylorDOF(:,:,m), nVar, nDegFr)                    ! Multiply -Ǎ*_pq with ∂^m/∂t^m Q̂_pl(t^n)
    CALL SPL_MATMUL(TaylorDOF(:,:,m+1), KxiTM, Temp, nDegFr, nVar)              ! Multiply result with M_nk^-1 K_lk^xi and add
    ! Contribution -M_nk^-1 K_lk^eta B̌*_pq                                      !
    CALL SPT_MATMUL(Temp, B, TaylorDOF(:,:,m), nVar, nDegFr)                    ! Multiply -B̌*_pq with ∂^m/∂t^m Q̂_pl(t^n)
    CALL SPL_MATMUL(TaylorDOF(:,:,m+1), KetaTM, Temp, nDegFr, nVar)             ! Multiply result with M_nk^-1 K_lk^eta and add
    ! Contribution -M_nk^-1 K_lk^zeta Č*_pq                                      !
    CALL SPT_MATMUL(Temp, C, TaylorDOF(:,:,m), nVar, nDegFr)                    ! Multiply -Č*_pq with ∂^m/∂t^m Q̂_pl(t^n)
    CALL SPL_MATMUL(TaylorDOF(:,:,m+1), KzetaTM, Temp, nDegFr, nVar)            ! Multiply result with M_nk^-1 K_lk^zeta and add
  ENDDO                                                                         !
  dtk(0) = t2-t1                                                                ! t^{n+1} - t^n = Δt
  DO m = 1, nPoly                                                               ! Compute the factors
    dtk(m) = dtk(m-1)*dtk(0)/REAL(m+1)                                          ! (t^{n+1} - t^n)^{(m+1)}/(m + 1)!
  ENDDO                                                                         ! for time-integrated Taylor series recursively
    TimeIntDOF(:,:) = 0.                                                        !
  DO m = 0, nPoly                                                               !
  TimeIntDOF(:,:) = TimeIntDOF(:,:) + TaylorDOF(:,:,m)*dtk(m)                   ! Time-integrated degrees of freedom
  ENDDO                                                                         !
END SUBROUTINE CauchyKovalewskiLinear3D                                         !
```