Geophys. J. Int. (2007) **171,** 665–694 doi: 10.1111/j.1365-246X.2007.03421.x

Arbitrary high-order finite volume schemes for seismic wave propagation on unstructured meshes in 2D and 3D

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

[1]*Department of Civil and Environmental Engineering, University of 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 2007 February 23. Received 2006 December 18; in original form 2006 July 13

SUMMARY

We present a new numerical method to solve the heterogeneous anelastic seismic wave equations with arbitrary high order of accuracy in space and time on unstructured triangular and tetrahedral meshes in two and three space dimensions, respectively. 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 relies on the finite volume (FV) approach where cell-averaged quantities are evolved in time by computing numerical fluxes at the element interfaces. The basic ingredient of the numerical flux function is the solution of Generalized Riemann Problems at the element interfaces according to the arbitrary high-order derivatives (ADER) approach of Toro *et al.*, where the initial data is piecewise polynomial instead of piecewise constant as it was in the original first-order FV scheme developed by Godunov. The ADER approach automatically produces a scheme of uniformly high order of accuracy in space and time. The high-order polynomials in space, needed as input for the numerical flux function, are obtained using a reconstruction operator acting on the cell averages. This reconstruction operator uses some techniques originally developed in the Discontinuous Galerkin (DG) Finite Element framework, namely hierarchical orthogonal basis functions in a reference element. In particular, in this article we pay special attention to underline the differences as well as the points in common with the ADER-DG schemes previously developed by the authors, especially concerning the MPI parallelization of both methods.

The numerical convergence analysis demonstrates that the proposed FV schemes provide very high order of accuracy even on unstructured tetrahedral meshes while computational cost for a desired accuracy can be reduced when applying higher order reconstructions. Applications to a series of well-acknowledged elastic and anelastic test cases and comparisons with analytic and numerical reference solutions, obtained by different well-established numerical methods, confirm the performance of the proposed method. Therefore, the development of the highly accurate ADER-FV 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: ADER approach, attenuation, finite volume schemes, high-order accuracy, tetrahedral meshes, viscoelasticity.

1 INTRODUCTION

Today numerical seismology can provide computer simulations of the propagation of seismic waves within the earth interior, that represent an invaluable tool for the understanding of the wave phenomena, their generation and their consequences. However, the simulation of a complete, highly accurate 3-D wavefield in realistic media with complex geometry is still a great challenge. After the seminal work of Madariaga (1976) and Virieux (1984, 1986) a number of different methods have been developed and a vast amount of publications on the simulation of seismic

wave propagation can be found in the literature. However, the improved knowledge of the subsurface structure and the necessity to handle geometrically complicated geological features has driven the development of numerical methods that use non-regular, unstructured meshes that provide the required geometrical flexibility.

First approaches, for example, in Braun & Sambridge (1995), Käser & Igel (2001), Käser *et al.* (2001) and Zhang (1997), provided numerical schemes with accuracies too low to be applied to realistic large-scale problems. After the spectral element method (SEM) was introduced in the field of numerical seismology in Priolo *et al.* (1994) and Seriani (1998), this spatially high-order accurate scheme was further developed in Komatitsch & Vilotte (1998), Komatitsch & Tromp (1999) and Komatitsch & Tromp (2002) and has already been applied to a very large number of realistic large-scale problems. However, the applicability of SEM is essentially limited to unstructured hexahedral meshes, whose generation may become very difficult in complex geometries and may need many manual interactions of the user. It is very important to note that the time required for manual user interaction during the mesh generation process can not be reduced like computational wallclock time that can be simply reduced by increasing the number of CPUs and that automatically will further decrease with the current and expected progress of modern supercomputer architecture. Although finite difference (FD) and SEM are very accurate as well as fast and efficient from a pure computational point of view, the bottleneck of these popular methods is the necessary mesh generation for very complex computational domains. For geometrically very complex computational domains, unstructured tetrahedral meshes are much more favourable from the point of view of mesh generation since modern commercial tetrahedral mesh generators can generate high quality tetrahedral meshes in a fully automatic manner that requires almost no manual interaction of the user. However, the main problems with tetrahedral meshes are that first there are only very few numerical methods able to run on tetrahedrons and that second the accuracy provided by those methods is usually relatively low, for example, considering second-order accurate TVD finite volume (FV) schemes.

Later, a new numerical method based on a Discontinuous Galerkin (DG) approach (Reed & Hill 1973; Cockburn & Shu 1998) in combination with a novel time integration scheme using arbitrary high-order derivatives (ADER) was introduced in Käser & Dumbser (2006) and Dumbser & Käser (2006) to simulate elastic wave propagation on unstructured triangular and tetrahedral meshes with arbitrary high order of accuracy in space and time.

Due to the increased accuracy, it became important to incorporate second-order effects such as attenuation and dispersion to correctly model the wave amplitudes and phases of a fully 3-D seismic wave field. A successful model for realistic attenuation is the approximation of the material as a viscoelastic medium. Hereby, it is important that the composition of the earth's polycrystalline material and the superposition of the microscopic physical attenuation processes leads to a flat attenuation band (see Liu *et al.* 1976; Stein & Wysession 2003). The correct numerical treatment of a viscoelastic medium is outlined in Moczo *et al.* (2004). Day & Minster (1984) transformed the stress–strain relation in the time domain into a differential form and obtained n differential equations for n additional *internal variables*, which replace the convolution integral. Emmerich & Korn (1987) and Emmerich (1992) improved this approach and showed that their method is superior in accuracy and computational efficiency and applied the viscoelastic models for the *P–SV* case. Independently, a different approach in Carcione *et al.* (1988) and Carcione & Cavallini (1994) introduced additional first-order differential equations for *memory variables*. Recent work by Moczo & Kristek (2005) reviewed both models and showed that indeed both approaches are equivalent. 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) (Marfurt 1984) and FD (Moczo & Bard 1993) methods. Finally, in Käser *et al.* (2006) viscoelastic attenuation was incorporated into the ADER-DG schemes.

In this paper, we introduce an arbitrary high-order ADER-FV scheme on unstructured triangular and tetrahedral meshes including elastic and viscoelastic media in order to improve the accuracy of the FV method which in modern commercial software packages is still limited to second order in space and time only. In contrast to previous FV approaches (e.g. Dormy & Tarantola 1995; Wang 2002; Wang & Liu 2002; Tadi 2004; Wang & Liu 2004; Wang *et al.* 2004) the proposed ADER-FV method is based on a new and efficient reconstruction operator for unstructured meshes in 2- and 3-D developed by Dumbser & Käser (2007) and on the solution of Generalized Riemann Problems (GRP) (Toro & Titarev 2002) at the element interfaces for flux computation. To our knowledge, the method presented in Dumbser & Käser (2007) is the first FV scheme on 3-D unstructured meshes of order higher than two. Former work on high-order FV schemes on unstructured meshes was restricted to two space dimensions (see e.g. Abgrall 1994; Friedrich 1998; Ollivier-Gooch & Van Altena 2002; Hu & Shu 1999; Käser & Iske 2005). The final formulation of the ADER-FV scheme differs from the ADER-DG scheme (Käser & Dumbser 2006; Dumbser & Käser 2006; Käser *et al.* 2006) only in the use of the reconstruction operator to obtain high-order spatial accuracy. Once the reconstruction is done, the implementation of the ADER-FV scheme is essentially the same as for ADER-DG methods. The advantage of FV methods, however, is that they allow considerably larger time steps than ADER-DG schemes and that one single time step is cheaper than a corresponding time step for ADER-DG schemes. The inconvenience is, that ADER-FV schemes are less accurate. However, if a fine mesh is needed for resolving small features in complex geometries, sometimes the accuracy provided by ADER-DG is not useful due to the geometrically imposed fine mesh. In this case, a less expensive FV scheme as described in this article may be the better choice.

The 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 reconstruction operator needed for the FV scheme is briefly explained in Section 3 and the resulting ADER-FV scheme based on this reconstruction is derived in Section 4. Two important boundary conditions are discussed in Section 5. In Section 6, we show numerically the convergence properties of the proposed scheme and in Section 7, we compare the ADER-FV scheme developed in this article with the ADER-DG method previously presented by the authors (Käser & Dumbser 2006; Dumbser & Käser 2006; Käser *et al.* 2006). Special attention is paid to the MPI parallelization of both methods, for which also MPI performance results are presented.

Finally, in Section 8, we present the numerical results obtained with the ADER-FV method for the 2-D Lamb's problem and the 3-D test cases LOH.1 and LOH.3 (layer over half-space) proposed by the Pacific Earthquake Engineering Research Center (Day *et al.* 2003) providing analytic and numerical reference solutions obtained by well-established codes of other research institutions. In particular, we compare the corresponding results of the ADER-FV proposed in this article with numerical results obtained with the ADER-DG method previously published by the authors (Dumbser & Käser 2006; Käser *et al.* 2006).

2 ANELASTIC WAVE EQUATIONS

The anelastic wave propagation can be described by modifying the constitutive relation, that is, Hooke's Law, as shown 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. In Liu *et al.* (1976) a superposition of different relaxation mechanisms is proposed as a way to fulfil both conditions. As introduced in Day & Minster (1984) and as also later outlined in Emmerich & Korn (1987) and Moczo & Kristek (2005), we can express the viscoelastic moduli as a combination of n mechanisms (so-called Maxwell bodies) 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, or 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 (3) into (4) leads to

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

Equation (5) can be used to fit any $\mathcal{Q}(\omega)$-law as shown in Emmerich & Korn (1987) and 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 (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 analogous 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. Therefore, from (5) we can also derive anelastic coefficients Y_ℓ^P and Y_ℓ^S for viscoelastic P- and S-wave propagation by solving the systems

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

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 (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 (2) and (3).

As shown in Kristek & Moczo (2003) and Moczo & Kristek (2005) we define a new set of variables, which are independent of the material properties, called the *anelastic functions* $\vec{\bar{\vartheta}}^\ell = (\bar{\vartheta}_{xx}^\ell, \bar{\vartheta}_{yy}^\ell, \bar{\vartheta}_{zz}^\ell, \bar{\vartheta}_{xy}^\ell, \bar{\vartheta}_{yz}^\ell, \bar{\vartheta}_{xz}^\ell)^T$, which contain the time history of the strain in the form

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

Using eq. (8) and applying the inverse Fourier transform to the viscoelastic modulus M_{ij} as outlined in detail in Moczo & Kristek (2005) the stress–strain relation (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 (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 (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 as a first-order hyperbolic PDE, it is convenient to redefine the anelastic functions in the form

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

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} -\left(\lambda Y_\ell^\lambda + 2\mu Y_\ell^\mu\right)\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 - \left(\lambda Y_\ell^\lambda + 2\mu Y_\ell^\mu\right)\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 - \left(\lambda Y_\ell^\lambda + 2\mu Y_\ell^\mu\right)\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,
\end{aligned}$$

$$
\begin{aligned}
\frac{\partial}{\partial t}\vartheta^n_{yy} - \omega_n \frac{\partial}{\partial y} v &= -\omega_n \vartheta^n_{yy},\\
\frac{\partial}{\partial t}\vartheta^n_{zz} - \omega_n \frac{\partial}{\partial z} w &= -\omega_n \vartheta^n_{zz},\\
\frac{\partial}{\partial t}\vartheta^n_{xy} - \frac{1}{2}\omega_n \left(\frac{\partial}{\partial x} v + \frac{\partial}{\partial y} u\right) &= -\omega_n \vartheta^n_{xy},\\
\frac{\partial}{\partial t}\vartheta^n_{yz} - \frac{1}{2}\omega_n \left(\frac{\partial}{\partial z} v + \frac{\partial}{\partial y} w\right) &= -\omega_n \vartheta^n_{yz},\\
\frac{\partial}{\partial t}\vartheta^n_{xz} - \frac{1}{2}\omega_n \left(\frac{\partial}{\partial z} u + \frac{\partial}{\partial x} w\right) &= -\omega_n \vartheta^n_{xz},
\end{aligned}
\tag{12}
$$

where n is the number of mechanisms used to approximate a frequency-independent Q-law and ρ is the density. For a study concerning the necessary number of mechanisms n to approximate a given Q-law with a required level of accuracy and the influence of the choice of n on the final numerical seismograms see Käser *et al.* (2006). Typically, three to five mechanisms are enough to approximate a constant value of Q on one frequency decade. Note, that each mechanism adds six further equations, that is, one for each stress component. Therefore, the system of the purely elastic 3-D wave equations consisting of nine 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 (12).

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

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

To obtain the 2-D case, we simply set $\frac{\partial}{\partial z} = 0$ and remove the corresponding equations for the stresses σ_{zz}, σ_{xz} and σ_{yz}, the memory variables ϑ^l_{zz}, ϑ^l_{xz} and ϑ^l_{yz} as well as for the velocity w from the state vector U_p and the Jacobians. Note, that the dimensions of the variable vector U_p, the Jacobian matrices $\check{A}$, $\check{B}$, $\check{C}$ and the source matrix $\check{E}$ depend on the number n of attenuation mechanisms. To keep the notation as simple as possible and without loss of generality, in the following we assume that the order of the equations in (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 (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:

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

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:

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

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 (13) representing a reaction source that couples the anelastic functions to the original elastic system can be decomposed as

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

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 ℓ-th 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 (19) 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 ℓ-th mechanism on its diagonal, that is, $E'_\ell = -\omega_\ell \cdot I$ with $I \in \mathbb{R}^{6\times 6}$ denoting the identity matrix. Since for flux computation we need to rotate the data into a coordinate system aligned with the face normal and since the numerical flux furthermore requires the absolute value $|\check{A}_{pq}|$ of matrix $\check{A}_{pq}$, in the following we will have a closer look at the absolute value matrix $|\check{A}_{pq}|$ and the rotation matrix $\check{T}_{pq}$. Similar to (14) we find that

$$|\check{A}_{pq}| = \begin{bmatrix} |A| & 0 \\ A^{||} & 0 \end{bmatrix} \in \mathbb{R}^{n_v \times n_v}, \tag{23}$$

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

$$|A| = \begin{pmatrix} c_p & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \lambda/(c_p\rho) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \lambda/(c_p\rho) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & c_s & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & c_s & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & c_p & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_s & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_s \end{pmatrix}, \tag{24}$$

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 similar block structure as in (15) of the form:

$$A^{||} = \begin{bmatrix} A_1^{||} \\ \vdots \\ A_n^{||} \end{bmatrix} \in \mathbb{R}^{6n\times 9}, \tag{25}$$

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 ℓ-th attenuation mechanism in the form:

$$A_\ell^{||} = \omega_\ell \cdot \begin{pmatrix} 1/(c_p\rho) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/(2c_s\rho) & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/(2c_s\rho) & 0 & 0 & 0 \end{pmatrix}. \tag{26}$$

To compute the rotation matrices we recall that the anelastic functions ϑ^ℓ are tensors like the stresses and thus the rotation matrix $\check{T}_{pq}$ for the full anelastic system (12) has the form

$$\check{T}_{pq} = \begin{bmatrix} T^t & 0 & 0 \\ 0 & T^v & 0 \\ 0 & 0 & T_a \end{bmatrix} \in \mathbb{R}^{n_v\times n_v}, \tag{27}$$

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

$$T^t = \begin{pmatrix} n_x^2 & s_x^2 & t_x^2 & 2n_xs_x & 2s_xt_x & 2n_xt_x \\ n_y^2 & s_y^2 & t_y^2 & 2n_ys_y & 2s_yt_y & 2n_yt_y \\ n_z^2 & s_z^2 & t_z^2 & 2n_zs_z & 2s_zt_z & 2n_zt_z \\ n_yn_x & s_ys_x & t_yt_x & n_ys_x + n_xs_y & s_yt_x + s_xt_y & n_yt_x + n_xt_y \\ n_zn_y & s_zs_y & t_zt_y & n_zs_y + n_ys_z & s_zt_y + s_yt_z & n_zt_y + n_yt_z \\ n_zn_x & s_zs_x & t_zt_x & n_zs_x + n_xs_z & s_zt_x + s_xt_z & n_zt_x + n_xt_z \end{pmatrix}, \tag{28}$$

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

$$T^v = \begin{pmatrix} n_x & s_x & t_x \\ n_y & s_y & t_y \\ n_z & s_z & t_z \end{pmatrix}. \tag{29}$$

The matrix T_a in (27) is a block diagonal matrix and has the structure

$$T_a = \begin{bmatrix} T^t & & 0 \\ & \ddots & \\ 0 & & T^t \end{bmatrix} \in \mathbb{R}^{6n\times 6n}, \tag{30}$$

where each of the n submatrices T^t is the tensor rotation matrix given in (28).

3 RECONSTRUCTION ALGORITHM

The main ingredient of the proposed arbitrary high-order FV scheme is the new reconstruction algorithm proposed in Dumbser & Käser (2007) that makes use of techniques developed originally in the DG framework. Whereas the method proposed by Dumbser & Käser (2007) even includes a non-linear WENO reconstruction algorithm based on several stencils to ensure monotonicity of discontinuous solutions, we will restrict the reconstruction operator in this article to a linear one, based only on one central stencil. The computational domain Ω is discretized by conforming elements $T^{(m)}$, indexed by a unique mono-index m ranging from 1 to the total number of elements E. The elements are chosen to be triangles in 2-D and tetrahedrons in 3-D. The union of all elements is called the triangulation or tetrahedrization of the domain,

$$\mathcal{T}_\Omega = \bigcup_{m=1}^{E} T^{(m)}. \tag{31}$$

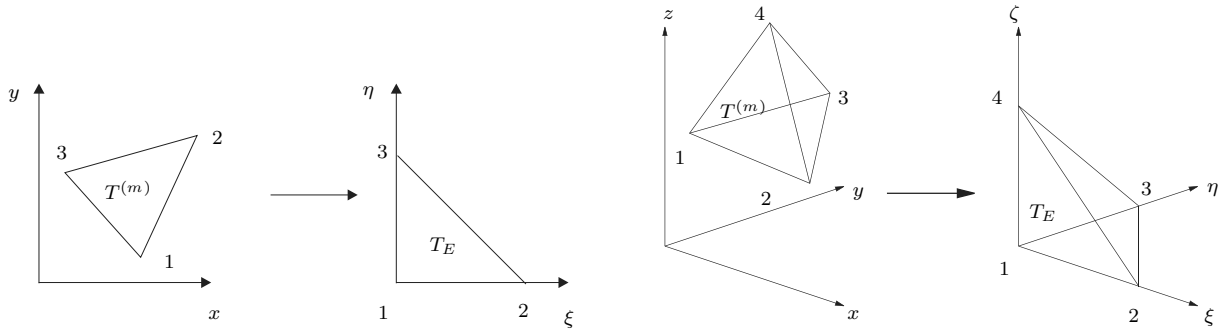

Figure 1. Transformation from the physical triangle and tetrahedron $T^{(m)}$ to the canonical reference triangle T_E with nodes (0,0), (1, 0) and (0, 1) and the canonical reference tetrahedron T_E with nodes (0, 0, 0), (1, 0, 0), (0, 1, 0) and (0, 0, 1).

As usual for FV schemes, data are represented by the cell averages of the state vector U_p inside an element $T^{(m)}$,

$$\bar{u}_p^{(m)} = \frac{1}{|T^{(m)}|} \int_{T^{(m)}} U_p \, dV, \tag{32}$$

where $|T^{(m)}|$ denotes the volume of the element.

In order to achieve high order of accuracy for a FV scheme, we need to reconstruct higher order polynomials W_p from the given cell averages $\bar{u}_p$. We write the reconstruction polynomial for element $T^{(m)}$ as

$$W_p^{(m)}(\xi, \eta, \zeta) = \hat{w}_{pl}^{(m)} \Psi_l(\xi, \eta, \zeta), \tag{33}$$

where ξ, η and ζ are the coordinates in a reference coordinate system, see Fig. 1, where also the reference elements T_E are defined. Throughout the paper we use classical tensor notation, which implies summation over each index appearing twice. Whereas the reconstructed degrees of freedom $\hat{w}_{pl}^{(m)}$ are not space-dependent, the reconstruction basis functions Ψ_l are polynomials of degree M and depend on space. The index l ranges from 0 to its maximum value $L - 1$, where $L = \frac{1}{2}(M+1)(M+2)$ and $L = \frac{1}{6}(M+1)(M+2)(M+3)$ are the numbers of reconstructed degrees of freedom in 2- and 3-D, respectively, depending on the order of the reconstruction. We use the hierarchical orthogonal reconstruction basis functions that are given, for example, in Dubiner (1991) and Cockburn *et al.* (2000) or in Appendix A for triangles in 2-D and tetrahedrons in 3-D. The transformation from the physical coordinate system x–y–z into the reference coordinate system ξ–η–ζ is in three space dimensions defined by

$$\begin{aligned}
x &= X_1^{(m)} + \left(X_2^{(m)} - X_1^{(m)}\right)\xi + \left(X_3^{(m)} - X_1^{(m)}\right)\eta + \left(X_4^{(m)} - X_1^{(m)}\right)\zeta, \\
y &= Y_1^{(m)} + \left(Y_2^{(m)} - Y_1^{(m)}\right)\xi + \left(Y_3^{(m)} - Y_1^{(m)}\right)\eta + \left(Y_4^{(m)} - Y_1^{(m)}\right)\zeta, \\
z &= Z_1^{(m)} + \left(Z_2^{(m)} - Z_1^{(m)}\right)\xi + \left(Z_3^{(m)} - Z_1^{(m)}\right)\eta + \left(Z_4^{(m)} - Z_1^{(m)}\right)\zeta,
\end{aligned}$$

where $X_i^{(m)}$, $Y_i^{(m)}$ and $Z_i^{(m)}$ denote the physical vertex coordinates of the considered element $T^{(m)}$. In two space dimensions, the same transformation applies for x and y, setting $\zeta = 0$. As short notation for the mapping and its inverse mapping from $\vec{\xi} = (\xi, \eta, \zeta)$ to $\vec{x} = (x, y, z)$ and vice versa with respect to the element $T^{(m)}$, we simply write

$$\vec{x} = \vec{x}(T^{(m)}, \vec{\xi}), \qquad \vec{\xi} = \vec{\xi}(T^{(m)}, \vec{x}). \tag{34}$$

Via the inverse mapping given in (34) for the vector $\vec{\xi}$, the element $T^{(m)}$ is transformed to the unit element T_E, whose volume is $|T_E| = \frac{1}{2}$ in two dimensions and $|T_E| = \frac{1}{6}$ in three space dimensions, respectively. Furthermore,

$$J_{ij} = \frac{\partial x_i}{\partial \xi_j} \tag{35}$$

is the Jacobian matrix of the transformation and $|J| = |J_{ij}|$ its determinant, being equal to twice the triangle surface in 2-D and equal to six times the tetrahedron volume in 3-D. For performing the reconstruction on element $T^{(m)}$, we now choose a reconstruction stencil

$$\mathcal{S}^{(m)} = \bigcup_{j=1}^{n_e} T^{(k_j)} \tag{36}$$

that contains a total number of n_e elements. Here $1 \le j \le n_e$ is a local index, counting the elements in the stencil, and (k_j) is the mapping from the local index j to the global indexation of the elements in $\mathcal{T}_\Omega$. We set by definition $(k_1) = (m)$ and thus the first element in the stencil is always the considered element $T^{(m)}$ for which reconstruction is to be done. For ease of notation, we write in the following only (k), meaning (k_j).

We then apply the inverse mapping (34) with respect to element $T^{(m)}$ to all the elements $T^{(k)} \in \mathcal{S}^{(m)}$, where the transformed elements are in the following denoted as $\tilde{T}^{(k)}$. We emphasize that for all elements $T^{(k)} \in \mathcal{S}^{(m)}$ the mapping with respect to the first element in the stencil is applied, so (m) is constant for each stencil and, therefore, the applied mapping formula is the *same* for all elements in $\mathcal{S}^{(m)}$. We note in particular that the transformed element of the first element in the stencil is of course the canonical reference element, hence $\tilde{T}^{(k_1)} = \tilde{T}^{(m)} = T_E$. The stencil transformed in that way is denoted $\tilde{\mathcal{S}}^{(m)} = \bigcup \tilde{T}^{(k)}$, see a two- and 3-D example in Figs 2 and 3.

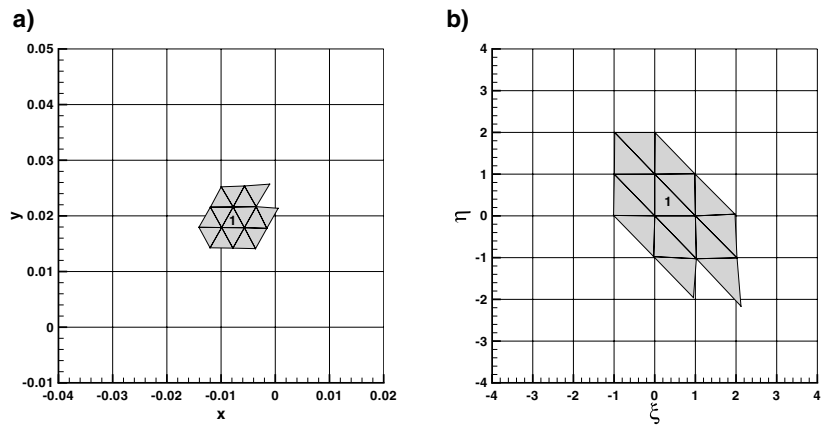

Figure 2. Examples of an original stencil $\mathcal{S}^{(m)}$ (a) and the corresponding transformed stencil $\tilde{\mathcal{S}}^{(m)}$ (b) in 2-D for the reconstruction of a polynomial of degree 3 with $n_e = 15$.

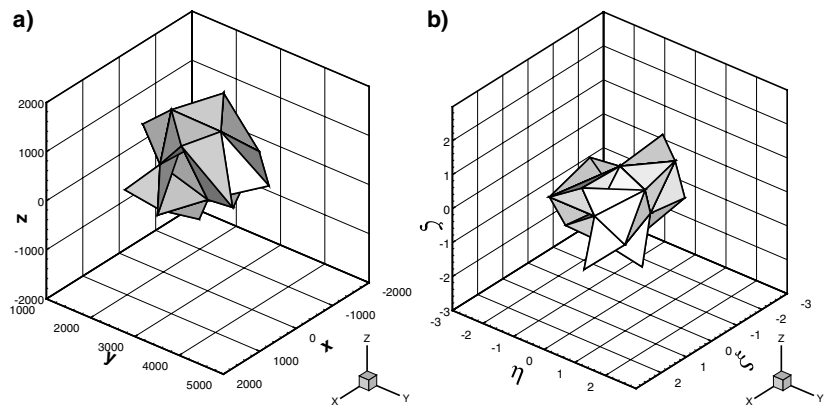

Figure 3. Examples of an original stencil $\mathcal{S}^{(m)}$ (a) and the corresponding transformed stencil $\tilde{\mathcal{S}}^{(m)}$ (b) in 3-D for the reconstruction of a polynomial of degree 2 with $n_e = 20$.

The reconstruction must be conservative, at least in the element considered for reconstruction. Initially, we even require integral conservation for $W_p^{(m)}$ in all elements $T^{(k)} \in \mathcal{S}^{(m)}$. In the physical coordinate system we thus have

$$\int_{T^{(k)}} W_p^{(m)}\big(\vec{\xi}\big(T^{(m)}, \vec{x}\big)\big)\, dV = \big|T^{(k)}\big|\bar{u}_p^{(k)}, \qquad \forall T^{(k)} \in \mathcal{S}^{(m)}. \tag{37}$$

After transforming all elements of the stencil using (34) and taking into account that the degrees of freedom $\hat{w}_{pl}^{(m)}$ do not depend on space, we obtain the intermediate result

$$|J|\left[\int_{\tilde{T}^{(k)}} \Psi_l(\vec{\xi}\,)\, d\xi\, d\eta\, d\zeta\right]\hat{w}_{pl}^{(m)} = |J|\big|\tilde{T}^{(k)}\big|\bar{u}_p^{(k)}, \qquad \forall \tilde{T}^{(k)} \in \tilde{\mathcal{S}}^{(m)}. \tag{38}$$

The Jacobian determinant appears on both sides of eq. (38) and thus cancels out. Please note that in the general case this is only possible for triangles and tetrahedrons with straight edges, to which we restrict ourselves in this paper. General polyhedral elements or even curved boundaries are not considered here.

The canceling of the Jacobian determinants automatically cancels scaling effects of the problem and avoids ill-conditioned reconstruction matrices as reported by Abgrall (1994). Abgrall and Friedrich (Friedrich 1998) used barycentric coordinates in order to avoid this problem, whereas we use a hierarchical *orthogonal* basis as commonly used in the DG-FE framework. During the reconstruction step, the basis polynomials are continuously extended over the whole stencil. In more detail, this extension means that during reconstruction the polynomial term given by $\Psi_l(\vec{\xi})$ is not only valid inside the reference element T_E, but also in all the other elements in the transformed stencil $\tilde{\mathcal{S}}^{(m)}$. After the reconstructed polynomial for element $T^{(m)}$ has been obtained, the basis polynomials are again restricted to the considered element $T^{(m)}$. We emphasize that the integration on the left-hand side of (38) has to be done over the *transformed* elements $\tilde{T}^{(k)}$. In order to do this integration, the trick now consists in doing *another* coordinate transformation to a second reference coordinate system using the vertices of the transformed element $\tilde{T}^{(k)}$ as parameter of another mapping from the first $\xi - \eta - \zeta$ reference system to the second $\tilde{\xi} - \tilde{\eta} - \tilde{\zeta}$ reference coordinate system. For convenience, we denote $\vec{\Xi} = (\tilde{\xi}, \tilde{\eta}, \tilde{\zeta})$. The mapping and its inverse are then denoted as

$$\vec{\xi} = \vec{\xi}\big(\tilde{T}^{(k)}, \vec{\Xi}\big), \qquad \vec{\Xi} = \vec{\Xi}\big(\tilde{T}^{(k)}, \vec{\xi}\big), \tag{39}$$

and the Jacobian determinant of this mapping is called $|\tilde{J}|$. Thus, eq. (38) becomes after the second transformation

$$|J||\tilde{J}|\left\{\int_{T_E}\Psi_l\big(\vec{\xi}(\tilde{T}^{(k)},\vec{\Xi})\big)d\tilde{\xi}d\tilde{\eta}d\tilde{\zeta}\right\}\hat{w}_{pl}^{(m)}=|J||\tilde{J}||T_E|\bar{u}_p^{(k)},\qquad \forall\tilde{T}^{(k)}\in\tilde{\mathcal{S}}^{(m)}, \tag{40}$$

where again all Jacobian determinants cancel out! The final set of reconstruction equations is

$$\left\{\int_{T_E}\Psi_l\big(\vec{\xi}(\tilde{T}^{(k)},\vec{\Xi})\big)\,d\tilde{\xi}\,d\tilde{\eta}\,d\tilde{\zeta}\right\}\hat{w}_{pl}^{(m)}=|T_E|\bar{u}_p^{(k)},\qquad \forall\tilde{T}^{(k)}\in\tilde{\mathcal{S}}^{(m)}. \tag{41}$$

In order to compute the integral on the left hand side of (41), we use classical multidimensional Gaussian quadrature of appropriate order. For an exhaustive overview of such multidimensional quadrature formulae see Stroud (1971).

For convenience, we introduce the simplified tensor notation

$$A_{jl}\hat{w}_{pl}=\bar{u}_{pj}, \tag{42}$$

with

$$A_{jl}=\frac{1}{|T_E|}\left\{\int_{T_E}\Psi_l\big(\vec{\xi}(\tilde{T}^{(k_j)},\vec{\Xi})\big)\,d\tilde{\xi}\,d\tilde{\eta}\,d\tilde{\zeta}\right\}\qquad\text{and}\qquad \bar{u}_{pj}=\bar{u}_p^{(k_j)}. \tag{43}$$

The number of reconstructed degrees of freedom is L and, therefore, we need at least $n_e = L$ elements in the stencil. Unfortunately, if we choose $n_e = L$ so that the matrix A_{jl} becomes square, the resulting scheme may become unstable on general meshes. Therefore, we are forced to use *more* elements than the necessary minimum. The use of enlarged reconstruction stencils for robustness purposes has already been reported previously in the literature (see e.g. Barth & Frederickson 1990; Ollivier-Gooch & Van Altena 2002; Käser & Iske 2005).

Furthermore, due to geometrical issues, the reconstruction matrix may be not invertible. This may happen, for example, when all elements are aligned on a straight line. Therefore, the stencil construction algorithm should avoid such cases. In our particular implementation, we compute the singular values of the matrix A_{jl} and check if some of them are zero. If so, we continue adding elements until none of the singular values is zero.

In order to fix parameters once and for all, since we are interested in a very general algorithm, we usually choose $n_e = 1.5L$ in 2-D and $n_e = 2L$ in 3-D. This means that we take between 50 and 100 per cent more elements than the minimum necessary for reconstruction.

The reconstruction stencil is generated for each element $T^{(m)}$ according to the following algorithm: We recursively add successively the Neumann neighbours (i.e. the direct side neighbours) of the element $T^{(m)}$ and all Neumann neighbours of the elements added to the stencil so far, until the desired number of stencil elements n_e is reached. This procedure guarantees a rather central reconstruction stencil which is needed for linear stability issues of the scheme. For an example of central stencils see Figs 2(a) and (b) in two dimensions and Figs 3(a) and (b) in three dimensions. As confirmed by the numerical results in Section 8, this algorithm works equally well at the boundaries of the computational domain, where the stencils are biased to one side.

Since (42) becomes overdetermined with our choice $n_e > L$ we use a constrained least-squares technique in order to solve (42) respecting conservation in the first element $T^{(m)}$ of the stencil. Due to the special choice of the reconstruction basis functions, the equality constraint becomes simply $\hat{w}_{p0} = \bar{u}_p^{(k_1)} = \bar{u}_p^{(m)}$, which is written in tensor notation

$$C_l\hat{w}_{pl}=R_i\bar{u}_{pi}. \tag{44}$$

The vectors C_l and R_i contain only zeros except for the entries $C_0 = 1$ and $R_1 = 1$. The least-squares solution of (42) with the constraint (44) coupled via a Lagrangian multiplier λ_p is obtained according to Dumbser & Käser (2007) as

$$\begin{pmatrix}2A_{jl}A_{jk} & -C_l\\ C_l\delta_{lk} & 0\end{pmatrix}\cdot\begin{pmatrix}\hat{w}_{pk}\\ \lambda_p\end{pmatrix}=\begin{pmatrix}2A_{jl}\bar{u}_{pj}\\ R_i\bar{u}_{pi}\end{pmatrix}. \tag{45}$$

Here, δ_{lk} is the Kronecker symbol. The matrix on the left hand side of (45) will be called reconstruction matrix in the following and in order to increase the speed of the algorithm, it is inverted and stored for each element of $\mathcal{T}_\Omega$ so that the unknown vector of the reconstructed degrees of freedom $\hat{w}_{pl}$ can be easily calculated for each component p by a simple matrix–vector multiplication of the inverse reconstruction matrix and the vector of known cell averages $\bar{u}_{pj}$ of the stencil $\mathcal{S}^{(m)}$. We repeat that reconstruction is done componentwise for each variable p of the governing eq. (13). We note that most of the memory requirements of our proposed scheme are due to the storage of the inverse reconstruction matrices.

4 FINITE VOLUME DISCRETIZATION OF THE ANELASTIC WAVE EQUATIONS

4.1 Semi-discrete finite volume scheme

The general semi-discrete form of the FV scheme is obtained by integration of (13) over an element $T^{(m)}$, integration by parts and inserting a numerical flux $F_p^h(W_p^+, W_p^-)\vec{n}$ in normal direction,

$$\int_{T^{(m)}}\frac{\partial U_p}{\partial t}\,dV+\int_{\partial T^{(m)}}F_p^h\left(W_q^-,W_q^+\right)\vec{n}\,dS=\int_{T^{(m)}}\check{E}_{pq}U_q\,dV. \tag{46}$$

The numerical flux is a function of the boundary extrapolated values W_p^+ and W_p^- at the element interfaces. In the case of a first-order FV scheme, these values correspond to the cell averages in the element $T^{(m)}$ and the neighbours, respectively. In the case of a higher order FV scheme, the values W_p^+ and W_p^- are obtained from a high-order polynomial reconstruction as shown in the previous Section 3.

The flux can be written very easily in a coordinate system which is aligned with the outward pointing unit normal vector $\vec{n} = (n_x, n_y, n_z)^T$ on the boundary making use of a variable rotation.

We use the exact Riemann solver as numerical flux in normal direction between two elements $T^{(m)}$ and $T^{(k_j)}$:

$$F_p^h(W_q^-, W_q^+)\vec{n} = \frac{1}{2}[\check{T}_{pr}(\check{A}_{rs} + |\check{A}_{rs}|)(\check{T}_{sq})^{-1} W_q^- + \check{T}_{pr}(\check{A}_{rs} - |\check{A}_{rs}|)(\check{T}_{sq})^{-1} W_q^+], \tag{47}$$

where $W_q^- = \hat{w}_{ql}^{(m)}\Psi_l^{(m)}$ and $W_q^+ = \hat{w}_{ql}^{(k_j)}\Psi_l^{(k_j)}$ are the boundary extrapolated values of the reconstructed numerical solution W_h from element (m) and the jth side neighbour (k_j), respectively, since both elements adjacent to a boundary contribute to the numerical flux. For the case of non-conservative linear systems with piecewise constant varying coefficients, the flux has to be evaluated in each element with the corresponding coefficient matrix $\check{A}_{pq} = \check{A}_{pq}^{(m)} = \check{A}_{pq}(T^{(m)})$ as function of the element $T^{(m)}$. Inserting (47) into (46) and splitting the boundary integral into the contributions of each face $1 \le j \le N_E$ of the element $T^{(m)}$, we obtain

$$\begin{aligned}\frac{\partial}{\partial t}\bar{u}_p^{(m)} \int_{T^{(m)}} dV &+ \frac{1}{2}\sum_{j=1}^{N_E} \check{T}_{pr}\big(\check{A}_{rs}^{(m)} + |\check{A}_{rs}^{(m)}|\big)(\check{T}_{sq})^{-1}\hat{w}_{ql}^{(m)} \int_{(\partial T^{(m)})_j} \Psi_l^{(m)}\, dS \\ &+ \frac{1}{2}\sum_{j=1}^{N_E} \check{T}_{pr}\big(\check{A}_{rs}^{(m)} - |\check{A}_{rs}^{(m)}|\big)(\check{T}_{sq})^{-1}\hat{w}_{ql}^{(k_j)} \int_{(\partial T^{(m)})_j} \Psi_l^{(k_j)}\, dS = \check{E}_{pq}^{(m)}\hat{w}_{ql}^{(m)} \int_{T^{(m)}} \Psi_l^{(m)}\, dV.\end{aligned} \tag{48}$$

Equation (48) is written in the physical x–y–z system, but if we transform each physical element $T^{(m)}$ to a canonical reference element T_E in a $\xi - \eta - \zeta$ reference system (see Fig. 1), the method can be implemented much more efficiently since all integrals can be pre-computed beforehand in the reference system.

After integration in the reference system and taking into account the orthogonality of the basis functions for the source term integral on the right hand side, the semi-discrete formulation in 2- and 3-D then reads as

$$\begin{aligned}\frac{\partial}{\partial t}\bar{u}_p^{(m)}\left|T^{(m)}\right| &+ \frac{1}{2}\sum_{j=1}^{N_E} \check{T}_{pr}\big(\check{A}_{rs}^{(m)} + \left|\check{A}_{rs}^{(m)}\right|\big)(\check{T}_{sq})^{-1}|S_j|F_l^{-,j}\hat{w}_{ql}^{(m)} \\ &+ \frac{1}{2}\sum_{j=1}^{N_E} \check{T}_{pr}\big(\check{A}_{rs}^{(m)} + \left|\check{A}_{rs}^{(m)}\right|\big)(\check{T}_{sq})^{-1}|S_j|F_l^{+,i,h}\hat{w}_{ql}^{(k_j)} = \check{E}_{pq}^{(m)}\bar{u}_q^{(m)}|T^{(m)}|,\end{aligned} \tag{49}$$

where $|S_j|$ denotes the surface of face j in 3-D and the edge length of edge j in 2-D. In eq. (49), we use flux matrices acting on the degrees of freedom of the reconstructed polynomials similar to the flux matrices for ADER-DG schemes introduced in Dumbser (2005), Käser & Dumbser (2006), Dumbser & Käser (2006) and Käser *et al.* (2006), which act on the degrees of freedom of the DG basis polynomials. The flux matrices can be calculated analytically once on the reference element and are then stored. We emphasize that these matrices are universal and do not depend on the mesh. Therefore, they have to be stored only once for the reference element and hence do not contribute to the total memory requirements of the scheme. In the following, we give the details of calculating those flux matrices on triangles and tetrahedrons in two and three space dimensions. First, we define the local faces with their local vertex ordering according to Table 1, where the vertex numbering is strictly counter-clockwise in 2-D as well as in 3-D. Then, the vector of volume coordinates $\vec{\xi}$ is given on the faces via mapping functions from the face parameters χ and τ, see Tables 2 and 3. Last but not least, for flux computation over the face, we have to integrate along the face inside the element as well as in the neighbour. This is done consistently by the transformation from the

Table 1. Face definition on triangles and tetrahedrons.

Face	Points		
	Triangles (2-D)		
1	1	2	
2	2	3	
3	3	1	
	Tetrahedrons (3-D)		
1	1	3	2
2	1	2	4
3	1	4	3
4	2	3	4

Table 2. 2-D volume coordinates $\vec{\xi}^{(j)}$ in function of the edge parameter χ.

j	1	2	3
$\xi^{(j)}(\chi)$	χ	$1-\chi$	0
$\eta^{(j)}(\chi)$	0	χ	$1-\chi$

Table 3. 3-D volume coordinates $\vec{\xi}^{(j)}$ in function of the face parameters χ and τ

j	1	2	3	4
$\xi^{(j)}(\chi,\tau)$	τ	χ	0	$1-\chi-\tau$
$\eta^{(j)}(\chi,\tau)$	χ	0	τ	χ
$\zeta^{(j)}(\chi,\tau)$	0	τ	χ	τ

Table 4. Transformation of the face parameters χ and τ of the tetrahedron's face to the face parameters $\tilde{\chi}$ and $\tilde{\tau}$ in the neighbour tetrahedron according to the three possible orientations (h) of the neighbour face.

h	1	2	3
$\tilde{\chi}^{(h)}(\chi,\tau)$	τ	$1-\chi-\tau$	χ
$\tilde{\tau}^{(h)}(\chi,\tau)$	χ	τ	$1-\chi-\tau$

face parameters χ and τ inside the element to the corresponding face parameters $\tilde{\chi}$ and $\tilde{\tau}$ in the neighbour face. Whereas in 2-D this transformation is always $\tilde{\chi}=1-\chi$, in 3-D the transformation depends on the orientation of the neighbour face respect to the local face of the considered element, since via rotation of the triangular faces there may be three possible orientations. The corresponding mappings are given in Table 4.

In two space dimensions, all possible flux matrices are

$$F_l^{-,j}=\int_{\partial(T_E)_j}\Psi_l[\vec{\xi}^{(j)}(\chi)]\,d\chi,\qquad \forall 1\le j\le 3, \tag{50}$$

$$F_l^{+,i,h}=\int_{\partial(T_E)_j}\Psi_l[\vec{\xi}^{(i)}(1-\chi)]\,d\chi,\qquad \forall 1\le i\le 3. \tag{51}$$

Index h is not used in 2-D. In three dimensions, all possible flux matrices are

$$F_l^{-,j}=\int_{\partial(T_E)_j}\Psi_l\big[\vec{\xi}^{(j)}(\chi,\tau)\big]d\chi\,d\tau,\qquad \forall 1\le j\le 4, \tag{52}$$

$$F_l^{+,i,h}=\int_{\partial(T_E)_j}\Psi_l\big[\vec{\xi}^{(i)}\big(\tilde{\chi}^{(h)},\tilde{\tau}^{(h)}\big)\big]\,d\chi\,d\tau,\qquad \forall 1\le i\le 4,\quad \forall 1\le h\le 3. \tag{53}$$

The left state flux matrix (superscript '$-$') $F_l^{-,j}$ accounts for the contribution of the element (m) itself to the fluxes over face j and the right state flux matrix (superscript '$+$') $F_l^{+,i,h}$ accounts for the contribution of the element's direct side neighbours (k_j) to the fluxes over the face j. Index $1\le i\le N_E$ indicates the local number of the common face as it is seen from neighbour (k_j) and depends on the mesh generator. Index $1\le h\le 3$ accounts for the three possible orientations of the face due to rotation and denotes the number of the local node in the neighbour's face which lies on the local vertex 1 of face j in tetrahedron number (m). Index h also depends on the mesh generator. On a given tetrahedral mesh, where indices i and h are known, only four of the 12 possible matrices $F_l^{+,i,h}$ are used per element.

4.2 The fully discrete formulation of the ADER-FV scheme

In this section, we show how the ADER approach (Toro & Titarev 2002; Titarev & Toro 2002, 2005) can be used for high-order time integration of the FV method on unstructured meshes, called ADER-FV method in the following, for general linear hyperbolic systems. It is mainly based on the so-called Cauchy-Kovalewski procedure which expresses time-derivatives in terms of purely spatial derivatives by successively differentiating the governing PDE. For linear systems of PDE, a particular simplification can be introduced: space–time integration and flux computation can be exchanged, which leads to a completely quadrature free approach in which all integrals are computed analytically.

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

$$\frac{\partial U_p}{\partial t} + \check{A}^*_{pq}\frac{\partial U_q}{\partial \xi} + \check{B}^*_{pq}\frac{\partial U_q}{\partial \eta} + \check{C}^*_{pq}\frac{\partial U_q}{\partial \zeta} - \check{E}_{pq}\, U_q = 0, \tag{54}$$

with

$$\check{A}^*_{pq} = \check{A}_{pq}\frac{\partial \xi}{\partial x} + \check{B}_{pq}\frac{\partial \xi}{\partial y} + \check{C}_{pq}\frac{\partial \xi}{\partial z}, \tag{55}$$

$$\check{B}^*_{pq} = \check{A}_{pq}\frac{\partial \eta}{\partial x} + \check{B}_{pq}\frac{\partial \eta}{\partial y} + \check{C}_{pq}\frac{\partial \eta}{\partial z}, \tag{56}$$

$$\check{C}^*_{pq} = \check{A}_{pq}\frac{\partial \zeta}{\partial x} + \check{B}_{pq}\frac{\partial \zeta}{\partial y} + \check{C}_{pq}\frac{\partial \zeta}{\partial z}. \tag{57}$$

The kth time derivative of the entire state vector U_p is obtained via the Cauchy-Kovalewski procedure applied to the governing eq. (13) in the reference system (54), and reads as

$$\frac{\partial^k}{\partial t^k}U_p = \left(\check{A}^*_{pq}\frac{\partial}{\partial \xi} + \check{B}^*_{pq}\frac{\partial}{\partial \eta} + \check{C}^*_{pq}\frac{\partial}{\partial \zeta} - \check{E}_{pq}\right)^k U_q, \tag{58}$$

which can be proven by complete induction.

We expand the reconstructed solution of (54) in a Taylor series in time about the current time level t^n up to degree M,

$$W_p(\xi,\eta,\zeta,t) = \sum_{k=0}^{M}\frac{(t-t^n)^k}{k!}\frac{\partial^k}{\partial t^k}W_p(\xi,\eta,\zeta,t^n), \tag{59}$$

and replace time derivatives by space derivatives, using eq. (58):

$$W_p(\xi,\eta,\zeta,t) = \sum_{k=0}^{M}\frac{(t-t^n)^k}{k!}(-1)^k\left(\check{A}^*_{pq}\frac{\partial}{\partial \xi} + \check{B}^*_{pq}\frac{\partial}{\partial \eta} + \check{C}^*_{pq}\frac{\partial}{\partial \zeta} - \check{E}_{pq}\right)^k W_q(\xi,\eta,\zeta,t^n). \tag{60}$$

We now introduce the approximation (33) and obtain

$$W_p(\xi,\eta,\zeta,t) = \sum_{k=0}^{M}\frac{(t-t^n)^k}{k!}(-1)^k\left(\check{A}^*_{pq}\frac{\partial}{\partial \xi} + \check{B}^*_{pq}\frac{\partial}{\partial \eta} + \check{C}^*_{pq}\frac{\partial}{\partial \zeta} - \check{E}_{pq}\right)^k \Psi_l(\vec{\xi})\hat{w}_{ql}(t^n). \tag{61}$$

This approximation can now be projected onto the reconstruction basis functions Ψ_k in order to get an approximation of the evolution of the reconstructed degrees of freedom during one time step from time level t^n to time level t^{n+1}. We obtain

$$\hat{w}_{pl}(t) = \frac{\left\langle \Psi_n, \sum_{k=0}^{M}\frac{(t-t^n)^k}{k!}(-1)^k\left(\check{A}^*_{pq}\frac{\partial}{\partial \xi} + \check{B}^*_{pq}\frac{\partial}{\partial \eta} + \check{C}^*_{pq}\frac{\partial}{\partial \zeta} - \check{E}_{pq}\right)^k \Psi_m\right\rangle}{\langle \Psi_n, \Psi_l\rangle}\hat{w}_{qm}(t^n), \tag{62}$$

where $\langle .\,,.\rangle$ denotes the inner product over the reference element T_E and the division by $\langle \Psi_n, \Psi_l\rangle$ denotes the multiplication with the inverse of the mass matrix. This reduces indeed to division by its diagonal entries since the mass matrix is diagonal due to the supposed orthogonality of the basis functions. Equation (62) can be integrated analytically in time from the current time level t^n to the next time level $t^{n+1} = t^n + \Delta t$. We obtain

$$\int_{t^n}^{t^n+\Delta t}\hat{w}_{pl}(t)\,dt = \frac{\left\langle \Psi_n, \sum_{k=0}^{M}\frac{\Delta t^{(k+1)}}{(k+1)!}(-1)^k\left(\check{A}^*_{pq}\frac{\partial}{\partial \xi} + \check{B}^*_{pq}\frac{\partial}{\partial \eta} + \check{C}^*_{pq}\frac{\partial}{\partial \zeta} - \check{E}_{pq}\right)^k \Psi_m\right\rangle}{\langle \Psi_n, \Psi_l\rangle}\hat{w}_{qm}(t^n). \tag{63}$$

With the definition

$$I_{plqm}(\Delta t) = \frac{\left\langle \Psi_n, \sum_{k=0}^{M}\frac{\Delta t^{(k+1)}}{(k+1)!}(-1)^k\left(\check{A}^*_{pq}\frac{\partial}{\partial \xi} + \check{B}^*_{pq}\frac{\partial}{\partial \eta} + \check{C}^*_{pq}\frac{\partial}{\partial \zeta} - \check{E}_{pq}\right)^k \Psi_m\right\rangle}{\langle \Psi_n, \Psi_l\rangle} \tag{64}$$

eq. (63) becomes simply

$$\int_{t^n}^{t^n+\Delta t}\hat{w}_{pl}(t)\,dt = I_{plqm}(\Delta t)\hat{w}_{qm}(t^n). \tag{65}$$

For efficient algorithms to do the Cauchy-Kovalewski procedure, we refer to Käser & Dumbser (2006), Dumbser & Käser (2006) and Käser *et al.* (2006). We finally obtain the fully discrete ADER-FV scheme by integration of (49) in time, where t^n and t^{n+1} denote the current and the successive time level:

$$\begin{aligned}
&\left(\bar{u}_p^{(m)}(t^{n+1}) - \bar{u}_p^{(m)}(t^n)\right)|T^{(m)}| \\
&\quad + \frac{1}{2}\sum_{j=1}^{N_E}\check{T}_{pr}\left(\check{A}^{(m)}_{rs} + \left|\check{A}^{(m)}_{rs}\right|\right)(\check{T}_{sq})^{-1}|S_j|F_l^{-,j}\cdot I_{qlnm}(\Delta t)\cdot\hat{w}^{(m)}_{nm}(t^n) \\
&\quad + \frac{1}{2}\sum_{j=1}^{N_E}\check{T}_{pr}\left(\check{A}^{(m)}_{rs} + \left|\check{A}^{(m)}_{rs}\right|\right)(\check{T}_{sq})^{-1}|S_j|F_l^{+,i,h}\cdot I_{qlnm}(\Delta t)\cdot\hat{w}^{(k_j)}_{nm}(t^n) = \check{E}^{(m)}_{pq}\cdot I_{q1nm}(\Delta t)\cdot\hat{w}^{(m)}_{nm}(t^n)\left|T^{(m)}\right|.
\end{aligned} \tag{66}$$

From the structure of eq. (66) we see that the space–time-integrated values on the boundaries enter the exact Riemann solver in order to give the space–time integral of the solution of the GRP at the interface. We emphasize that this can only be done for linear systems.

The reconstructed degrees of freedom $\hat{w}_{nm}^{(m)}(t^n)$ at time level t^n are obtained for each element at the beginning of a time step using the reconstruction operator described in Section 3. The proposed FV scheme is quadrature-free since no Gaussian integration is used in space and time. It performs high-order time-integration from t^n to t^{n+1} in one single step. It thus needs the same memory as a first-order explicit Euler time stepping scheme. The scheme looks almost the same as the ADER-DG scheme presented in Käser *et al.* (2006), except for the following two differences: First, for FV schemes only the cell averages $\bar{u}_p$ have to be evolved in time, whereas for DG schemes all degrees of freedom $\hat{u}_{pl}$ must be updated. Second, the fluxes of ADER-DG schemes are computed directly with the degrees of freedom $\hat{u}_{pl}$, whereas the fluxes for ADER-FV schemes are computed using the reconstructed degrees of freedom $\hat{w}_{pl}$, which are obtained from the cell averages $\bar{u}_p$ in the separate reconstruction step.

5 BOUNDARY CONDITIONS

There is a variety of physically meaningful boundary conditions of an elastic medium. However, the two most important types of boundaries are *absorbing* and *free surface* boundaries, which will be discussed in the framework of the ADER-FV method in the following. An important difficulty in the context of FV schemes in contrast to DG methods is the generation of appropriate reconstruction stencils at the boundary of the computational domain. In this article, we choose one-sided stencils, that is, stencils that lie completely in the computational domain. Without changing the stencil search algorithm described in Section 3 the stencils at the boundary are simply generated by adding to the stencil recursively the direct Neumann neighbours of the elements already in the stencil until the required number of elements n_e is reached, starting always with the central element for which reconstruction is to be performed. Since at the boundary several elements do not have a full set of direct Neumann neighbours, only the existing neighbours can be added. For the flux computation, we then solve inverse Riemann Problems, as for ADER-DG methods.

5.1 Absorbing boundaries

At absorbing boundaries, no waves are supposed to enter the computational domain and the waves travelling outward should pass the boundary without reflections. In this section we present a very simple approach, that so far yielded satisfactory results, at least for our purposes. The numerical flux (47) is based on the solution of the Riemann Problem given by the jump across the element interface. It is a strict upwind method, that is, outgoing waves at an element interface are only influenced by the state in the inside element itself. In contrast, the flux contribution of incoming waves is purely due to the state in the neighbour element. Thus, a simple implementation of absorbing boundary conditions is to use the following numerical flux in (66) at all those tetrahedral faces that coincide with an absorbing boundary:

$$F_p^{\mathrm{AbsorbBC}}(W_q^-, W_q^+)\vec{n} = \frac{1}{2}\check{T}_{pr}\left(\check{A}_{rs}^{(m)} + \left|\check{A}_{rs}^{(m)}\right|\right)(\check{T}_{sq})^{-1}W_q^-. \tag{67}$$

The flux function (67) allows only for outgoing waves, which are merely defined by the state in the element due to upwinding. Since incoming waves are not allowed, the respective flux contribution must vanish, that is, it is set to zero in the implementation of the method. We are aware that these absorbing boundary conditions have some problems at corners or for *grazing* incidence of waves. Therefore, in future work, approaches like the perfectly matched layer (PML) technique, as introduced in Bérenger (1994) and applied in Collino & Tsogka (2001) and Komatitsch & Tromp (2003) should be incorporated to improve the performance of the proposed scheme for such boundaries.

5.2 Free-surface boundaries

On the free surface of an elastic medium, the normal stress and the shear stresses with respect to the boundary are determined by physical constraints. Outside the elastic medium, there are no external forces that retract the particles into their original position. Therefore, the normal stress and the shear stress values at the free surface have to be zero. In contrast to classical continuous FE methods such as the SEM, we have no direct control on the values at the boundaries within the FV framework. However, the boundary conditions can be imposed correctly via the numerical flux. Considering that the numerical flux is based on the solution of a Riemann Problem at an element interface and given some boundary extrapolated values from inside the computational domain on a free surface, we must solve a so-called inverse Riemann Problem such that its solution yields exactly the free-surface boundary conditions *at* the domain boundary. In the particular case of the free surface, the solution of the inverse Riemann Problem can be obtained via symmetry considerations. For those components of the state vector U_p, that we want to be zero at the domain boundary, we prescribe a virtual boundary extrapolated component on the outside of the interface that has the same magnitude but opposite sign. For the other components we just copy the inside values to the outside. For the free-surface boundary condition the resulting numerical flux function in (66) can then be formulated as follows,

$$F_p^{\mathrm{FreeSurfBC}}(W_q^-, W_q^+)\vec{n} = \frac{1}{2}\check{T}_{pr}\left(\check{A}_{rs}^{(m)} + \left|\check{A}_{rs}^{(m)}\right|\right)(\check{T}_{sq})^{-1}W_q^- + \frac{1}{2}\check{T}_{pr}\left(\check{A}_{rs}^{(m)} - \left|\check{A}_{rs}^{(m)}\right|\right)\Gamma_{st}\,(\check{T}_{tq})^{-1}W_q^+, \tag{68}$$

where the matrix $\Gamma_{\mathrm{st}} = \mathrm{diag}(-1, 1, 1, -1, 1, -1, 1, 1, 1, 0, \ldots, 0)$ accounts for the mirroring of normal and shear stresses with respect to the face-normal direction. The viscoelastic memory variables do not enter the flux. We remark that the solution of the inverse Riemann

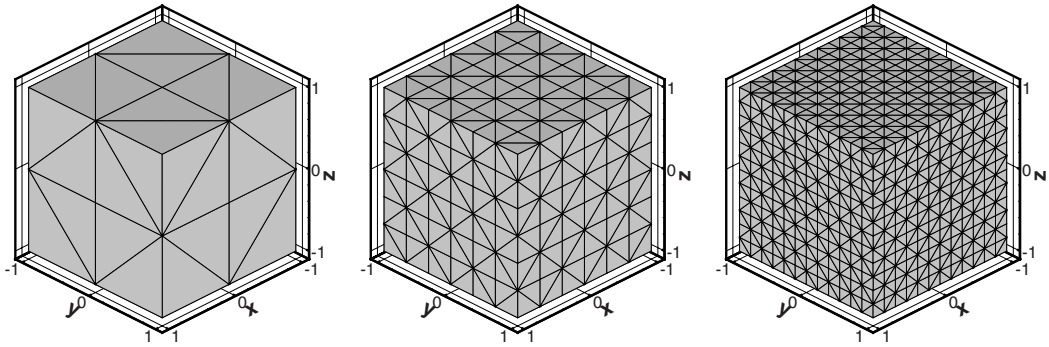

Figure 4. Sequence of discretizations of the computational domain Ω via regularly refined tetrahedral meshes used for the numerical convergence analysis.

Problem is *not* equivalent to the FD approach of adding fictitious *ghost* points, but comes out naturally from the FV framework and provides the exact values of the normal and shear stresses as required by the free surface boundary condition. Numerical tests such as Lamb's problem in two space dimensions and the LOH test cases in three space dimensions confirm the performance and accuracy of this approach, especially compared to conventional FD schemes, as shown in Section 8.2.

6 CONVERGENCE ANALYSIS

In this section, we present the results of a numerical convergence analysis to demonstrate the very high accuracy that can be obtained with the proposed ADER-FV method on unstructured tetrahedral meshes considering viscoelastic attenuation. We show results from second- to sixth-order ADER-FV schemes, which are denoted by ADER-FV $\mathcal{O}2$ to ADER-FV $\mathcal{O}6$, respectively. Note, that the same order for time and space accuracy is automatically obtained.

To determine the convergence order of our method we solve the 3-D seismic wave equations with viscoelastic attenuation in (12) on the unit-cube as sketched in Fig. 4, that is, on a computational domain $\Omega = [-1, 1] \times [-1, 1] \times [-1, 1] \in \mathbb{R}^3$ with periodic boundary conditions.

The homogeneous material parameters are set to

$$\lambda = 2, \qquad \mu = 1, \qquad \rho = 1, \qquad Q_P = 20, \qquad Q_S = 10, \tag{69}$$

throughout the computational domain Ω. The Q-factors are assumed to be frequency independent over the frequency band [0.1, 10] Hz. To this end, we are using 5 mechanisms as outlined in Section 2. For the quality of the approximation of the frequency independent Q factors and the associated computational effort in function of the number of mechanisms used see Käser *et al.* (2006). Approximately, the same CPU time growth as shown there for ADER-DG schemes also holds for the ADER-FV method presented in this article. These attenuation properties introduce damping and dispersion of the P and S waves. We know, for example, from Stein & Wysession (2003), that a space–time harmonic solution to this problem can be found under the form

$$U_p(x, y, z, t) = U_p^0 \cdot e^{i\cdot(\omega t - k_x x - k_y y - k_z z)}, \qquad p = 1, \ldots, n_v \tag{70}$$

where U_p^0 is the initial amplitude vector, ω the wave frequencies to determine, and

$$\vec{k} = (k_x, k_y, k_z)^T = (\pi, \pi, \pi)^T. \tag{71}$$

is the wavenumber vector leading to a periodic, plane sinusoidal wave in the unit-cube with the wave front perpendicular to the cube's space diagonal. In the following, we briefly describe how we determine the wave frequencies ω:

With the assumption, that eq. (70) is the analytic solution of the governing eq. (13), we calculate the first time and space derivatives of eq. (70) analytically and plug them into eq. (13). From there, we can derive the so-called dispersion relation, which is the following eigenproblem

$$(\check{A}_{pq} k_x + \check{B}_{pq} k_y + \check{C}_{pq} k_z - i \cdot \check{E}_{pq}) \cdot U_q^0 = \omega \cdot U_q^0, \qquad p, q = 1, \ldots, n_v, \tag{72}$$

with $i^2 = -1$. Solving the eigenproblem (72) gives us the matrix R_{pq} of right eigenvectors $R_{p1}, \ldots, R_{pn_v}$ and the associated eigenvalues ω_p.

Recalling, for example, from Toro (1999), that each solution of the linear hyperbolic system (13) is given by a linear combination of the right eigenvectors, that is, $U_p = R_{pq} \nu_q$, we can compute the coefficients as $\nu_p = R_{pq}^{-1} U_q^0$ via the initial amplitude vector. Now, we can synthesize the exact solution of the attenuated plane wave in the form

$$U_p(x, y, z, t) = R_{pq} \nu_q \cdot e^{i\cdot(\omega_q t - k_x x - k_y y - k_z z)}. \tag{73}$$

In the convergence test, we use a plane P wave and a plane S-wave travelling in opposite directions along the space diagonal $\vec{n} = (1, 1, 1)^T$ of the domain Ω as already shown in Dumbser & Käser (2006). Therefore, the initial condition at $t = 0$ is given by (73) using the combination of only two right eigenvectors $(R_{p2}, \ldots, R_{p9})$ with the coefficients $\nu_2 = \nu_9 = 1$ and zero otherwise.

Table 5. Convergence orders for the velocity component v for ADER-FV $\mathcal{O}2$ to ADER-FV $\mathcal{O}6$ schemes in ascending order. Viscoelastic attenuation is modeled using five mechanisms.

h	L^∞	$\mathcal{O}_{L^\infty}$	L^2	$\mathcal{O}_{L^2}$	I	CPU (s)
2.16×10^{-1}	1.5622×10^{-2}	–	1.7101×10^{-2}	–	4	1
1.08×10^{-1}	3.7724×10^{-3}	2.1	3.9186×10^{-3}	2.1	8	16
7.21×10^{-2}	1.6544×10^{-3}	2.0	1.6899×10^{-3}	2.1	12	72
5.41×10^{-2}	9.8176×10^{-4}	1.8	9.8126×10^{-4}	1.9	16	228
2.16×10^{-1}	2.5053×10^{-2}	–	2.4776×10^{-2}	–	4	1
1.08×10^{-1}	3.9698×10^{-3}	2.7	3.1450×10^{-3}	3.0	8	21
7.21×10^{-2}	1.2139×10^{-3}	2.9	9.3247×10^{-4}	3.0	12	100
5.41×10^{-2}	5.0839×10^{-4}	3.0	3.9337×10^{-4}	3.0	16	316
2.16×10^{-1}	5.0283×10^{-3}	–	5.0372×10^{-3}	–	4	3
1.44×10^{-1}	9.8627×10^{-4}	4.0	1.0197×10^{-3}	3.9	6	13
1.08×10^{-1}	3.5887×10^{-4}	3.5	3.3024×10^{-4}	3.9	8	40
7.21×10^{-2}	7.1172×10^{-5}	4.0	6.6368×10^{-5}	4.0	12	203
2.16×10^{-1}	3.4920×10^{-3}	–	3.4793×10^{-3}	–	4	7
1.44×10^{-1}	5.0073×10^{-4}	4.8	4.6677×10^{-4}	5.0	6	36
1.08×10^{-1}	1.1954×10^{-4}	5.0	1.1227×10^{-4}	5.0	8	99
7.21×10^{-2}	1.5758×10^{-5}	5.0	1.4849×10^{-5}	5.0	12	501
2.16×10^{-1}	1.0705×10^{-3}	–	7.3333×10^{-4}	–	4	15
1.44×10^{-1}	1.0996×10^{-4}	5.6	6.6501×10^{-5}	5.9	6	73
1.08×10^{-1}	2.2398×10^{-5}	5.5	1.2250×10^{-5}	5.9	8	219
7.21×10^{-2}	2.0320×10^{-6}	5.9	1.0900×10^{-6}	6.0	12	1110

The total simulation time T is set to $T = 0.1$s. The CFL number is set in all computations to 0.5. For a thorough investigation of the linear stability properties of ADER-FV schemes via differential approximation and via von Neumann stability analysis (see Dumbser *et al.* 2006). In the same reference, also the dispersion and dissipation errors of ADER-FV schemes are investigated theoretically and numerically.

The numerical analysis to determine the convergence orders is performed on a sequence of tetrahedral meshes as shown in Fig. 4. 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. 4. This way, the refinement is controlled by changing the number of subcubes in each space dimension.

We can arbitrarily pick one of the variables of the system of the seismic wave eq. (13) to numerically determine the convergence order of the used ADER-FV schemes. In Table 5, we show the error for the velocity component v. The error of the reconstructed numerical solution W_h with respect to the exact solution U_e is measured in the L^∞-norm and the continuous L^2-norm

$$\|W_h - U_e\|_{L^2(\Omega)} = \left(\int_\Omega |W_h - U_e|^2 \, dV \right)^{1/2}, \tag{74}$$

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 first column in Table 5 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. 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.

7 COMPARISON OF ADER-FV AND ADER-DG SCHEMES

In this section, we provide a thorough comparison of the ADER-FV schemes presented in this paper for the anelastic wave equations and the ADER-DG method proposed for the same equations previously in Käser *et al.* (2006). First, we recall the fully discrete versions of both schemes in eqs (75) and (76), respectively.

The ADER-FV scheme for the anelastic wave equations developed in this article reads as

$$\begin{aligned} &|T^{(m)}| \left[\bar{u}_p^{(m)}(t^{n+1}) - \bar{u}_p^{(m)}(t^n) \right] \\ &\quad + \frac{1}{2} \sum_{j=1}^{N_E} \check{T}_{pr} \left(\check{A}_{rs}^{(m)} + \left| \check{A}_{rs}^{(m)} \right| \right) (\check{T}_{sq})^{-1} |S_j| F_l^{-,j} \cdot I_{qlnm}(\Delta t) \cdot \hat{w}_{nm}^{(m)}(t^n) \\ &\quad + \frac{1}{2} \sum_{j=1}^{N_E} \check{T}_{pr} \left(\check{A}_{rs}^{(m)} + \left| \check{A}_{rs}^{(m)} \right| \right) (\check{T}_{sq})^{-1} |S_j| F_l^{+,i,h} \cdot I_{qlnm}(\Delta t) \cdot \hat{w}_{nm}^{(k_j)}(t^n) = \left| T^{(m)} \right| \check{E}_{pq}^{(m)} \cdot I_{q1nm}(\Delta t) \cdot \hat{w}_{nm}^{(m)}(t^n) . \end{aligned} \tag{75}$$

We recall that the ADER-DG method for the anelastic wave equations presented in Käser *et al.* (2006) has the following form:

$$
\begin{aligned}
&|J| M_{kl}\big[\hat{u}_{pl}^{(m)}(t^{n+1}) - \hat{u}_{pl}^{(m)}(t^{n})\big] \\
&\quad + \frac{1}{2}\sum_{j=1}^{N_E} \check{T}_{pr}^{j}\big(\check{A}_{rs}^{(m)} + \big|\check{A}_{rs}^{(m)}\big|\big)(\check{T}_{sq}^{j})^{-1}|S_j| F_{kl}^{-,j} \cdot I_{qlnm}(\Delta t) \cdot \hat{u}_{nm}^{(m)}(t^{n}) \\
&\quad + \frac{1}{2}\sum_{j=1}^{N_E} \check{T}_{pr}^{j}\big(\check{A}_{rs}^{(m)} - \big|\check{A}_{rs}^{(m)}\big|\big)(\check{T}_{sq}^{j})^{-1}|S_j| F_{kl}^{+,j,i,h} \cdot I_{qlnm}(\Delta t) \cdot \hat{u}_{nm}^{(k_j)}(t^{n}) \\
&\quad - \check{A}_{pq}^{*}\,|J|\,K_{kl}^{\xi} \cdot I_{qlnm}(\Delta t) \cdot \hat{u}_{nm}^{(m)}(t^{n}) - \check{B}_{pq}^{*}\,|J|\,K_{kl}^{\eta} \cdot I_{qlnm}(\Delta t) \cdot \hat{u}_{nm}^{(m)}(t^{n}) \\
&\quad \check{C}_{pq}^{*}\,|J|\,K_{kl}^{\zeta} \cdot I_{qlnm}(\Delta t) \cdot \hat{u}_{nm}^{(m)}(t^{n}) = |J|\,M_{kl}\check{E}_{pq}^{(m)} \cdot I_{qlmn}(\Delta t) \cdot \hat{u}_{mn}^{(m)}(t^{n}).
\end{aligned}
\tag{76}
$$

From the fully discrete version of both schemes we can immediately deduce those features that both schemes have in common as well as their differences, which will have an important impact on CPU time and memory requirements of both methods. Both schemes are one-step methods, that is, they directly integrate the governing eq. (13) from time level t^n to time level t^{n+1} without any intermediate stages. This is possible thanks to the Cauchy-Kovalewski procedure that is identical in both methods. However, in the ADER-FV scheme (75) the Cauchy-Kovalewski procedure is applied to the reconstructed degrees of freedom $\hat{w}_{nm}$ whereas in the ADER-DG scheme (76) this procedure can be directly applied to the degrees of freedom $\hat{u}_{nm}$ already given by the spatial DG discretization. Except for this difference, the method for carrying out the Cauchy-Kovalewski procedure is the same in both schemes, see the tensor $I_{qlnm}(\Delta t)$ appearing in both methods. Furthermore, both methods are completely quadrature-free since all spatial integrals are computed analytically and then stored in the flux matrices. However, the flux matrices do *not* have the same size for ADER-DG and ADER-FV schemes. This is due to a very important difference that distinguishes both approaches: whereas in the ADER-DG scheme all polynomial coefficients $\hat{u}_{pl}$ (matrix of size $n_v \times L$) are evolved in time, the ADER-FV scheme only advances the cell averages $\bar{u}_p$ (vector of length n_v), see (75) and (76). We repeat that n_v is the number of variables in the system of governing equations and $L = \frac{1}{2}(M+1)(M+2)$ and $L = \frac{1}{6}(M+1)(M+2)(M+3)$ are the number of degrees of freedom in two and three space dimensions, respectively. This means, that the fully discrete system for the ADER-DG scheme is by a factor of L *larger* than the corresponding system for ADER-FV schemes. This is also reflected in the size of the flux matrices, which are simple vectors of length L in the case of ADER-FV schemes and matrices of size $L \times L$ for ADER-DG schemes. The large flux matrices are the first key factor leading to the much larger CPU time observed for ADER-DG in comparison to ADER-FV. As a side note we remark that due to the representation of the reconstructed solution W of the FV scheme in terms of the same basis functions as for the DG schemes, the flux matrix of ADER-FV schemes is identical to the first row of the corresponding ADER-DG flux matrix. In addition to the larger flux matrices, in the ADER-DG algorithm also stiffness matrices appear due to the non-vanishing volume integral after the integration by parts operation in the derivation of the scheme (see e.g. Käser & Dumbser 2006; Dumbser & Käser 2006). We see that the stiffness matrices K_{kl}^{ξ},K_{kl}^{η} and K_{kl}^{ζ} appear in (76) but not in (75). This adds an additional CPU effort for the ADER-DG scheme, however, it is very small compared to the flux computation since the stiffness matrices are very sparse and, therefore, can be multiplied very efficiently.

The second and even more important key factor leading to the much larger CPU times of ADER-DG compared to ADER-FV is the unfavourable time step restriction that comes out of a von Neumann stability analysis (see e.g. Dumbser 2005). In general, one can roughly say that the ADER-DG time step limit decreases proportional to $\frac{1}{2M+1}$, where M is the degree of the DG basis polynomials. For the FV scheme, the time step limit is *independent* of the degree M of the reconstruction basis polynomials, or even becomes larger, at least on Cartesian grids in multiple space dimensions. For details on this topic see Dumbser *et al.* (2006). This means that, for example, a fifth-order ADER-FV scheme has a time step limit that is nine times larger than the time step limit of a fifth-order ADER-DG scheme on the same mesh. Although this disadvantage seems to be so heavy that ADER-DG schemes may suddenly appear very unattractive from this point of view, we will explain later in this section that ADER-DG still has very strong advantages over ADER-FV which makes them at least competitive against ADER-FV, if not superior.

Before discussing the inconveniences of ADER-FV schemes, we would like to make some remarks on the MPI parallelization of both schemes on modern massively parallel systems. Since both methods are one-step schemes, the total communication overhead is considerably low compared to methods using high-order Runge-Kutta time integration since in the ADER approach data has to be exchanged only *once* at the beginning of each time step. Then, each subdomain can evolve the solution independently of its neighbour domains. Unstructured mesh partitioning is done for both schemes with the free METIS software package described in Karypis & Kumar (1998). Whereas ADER-DG schemes must exchange only the degrees of freedom $\hat{u}_{pl}$ of the direct neighbours of a subdomain boundary, ADER-FV schemes must exchange also all the cell averages $\bar{u}_p$ necessary for the reconstruction procedure. In our implementation, we decided to split the MPI communications for ADER-FV into two parts. First, all the necessary cell averages $\bar{u}_p$ needed for reconstruction in each subdomain are exchanged. Then, each subdomain performs the reconstruction and finally all subdomains exchange the reconstructed degrees of freedom $\hat{w}_{pl}$ only at the direct neighbours of a subdomain boundary, exactly as in the ADER-DG case. At this stage we would like to give some more details on the domain decomposition for high order FV methods where the reconstruction stencils may become quite large and, therefore, the domains will overlap by more than one row of elements. In a pre-processing step, we run the standard METIS software package to provide a non-overlapping domain decomposition of the entire computational domain

$$
\Omega = \bigcup_{i=1}^{n_c} \Omega_i, \qquad \text{with} \quad \Omega_i \cap \Omega_j = \emptyset \qquad \text{for} \quad i \neq j, \tag{77}
$$

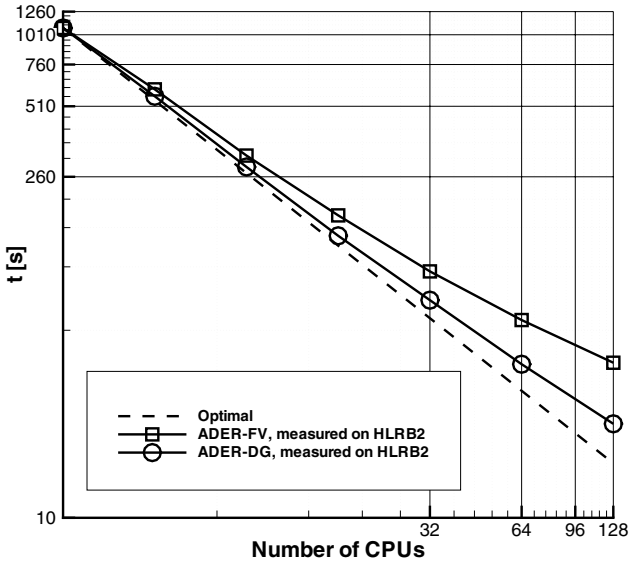

Figure 5. MPI speedup measured on the HLRB2 supercomputed of the Leibniz Rechenzentrum in München, Germany, for ADER-FV and ADER-DG schemes and theoretical optimum for 100 per cent MPI efficiency.

into n_c subdomains Ω_i for the n_c CPUs. After starting the parallel FV code, the whole mesh is read by each processor and then each CPU i searches only for those reconstruction stencils needed to perform reconstruction for all elements in its own subdomain Ω_i. These reconstruction stencils may reach, however, into the subdomains Ω_j of other CPUs. The lists of those elements which are necessary for reconstruction and that are *not* in the own subdomain Ω_i but that lie in some neighbour subdomain Ω_j are sent once at the beginning to all neighbour CPUs j before the start of the computation. In this manner, the CPUs communicate to each other the lists of those elements needed for the reconstruction procedure that lie in a neighbour subdomain (reconstruction lists). After the communication of the reconstruction lists, each CPU i keeps in memory *only* the data and the part of the mesh that is associated to its own subdomain Ω_i. At the beginning of a time step, each CPU j sends to its neighbour i the cell averages of those elements that have been requested by CPU i through the reconstruction list. Recapitulating, for our implementation of the ADER-FV method, the user initially just has to provide a non-overlapping domain decomposition that can be generated with any standard software packages, such as, for example, METIS, and then the overlapping of the reconstruction stencils is handled internally by the code in a pre-processing step via communication of the reconstruction lists.

Due to the reconstruction, the communication overhead is higher for ADER-FV compared to ADER-DG and we thus expect the MPI scalability to be better for ADER-DG schemes. This is indeed confirmed by our numerical experiments carried out on the HLRB2 supercomputer (4096 Intel Itanium2 Madison 9M CPUs, each with 1.6 GHz clock speed and 4GB of RAM) of the Leibniz Rechenzentrum (LRZ) in München, Germany. We ran a fixed test problem with the same number of 69 120 tetrahedral elements and fifth order of accuracy in space and time for both the ADER-FV and the ADER-DG scheme using 2, 4, 8, 16, 32, 64 and 128 CPUs. The measured MPI speedup and the theoretical optimum supposing 100 per cent efficiency are plotted in Fig. 5, where the times of ADER-DG are normalized with respect to the CPU times needed by the ADER-FV scheme on two processors. On 128 CPUs, the ADER-DG method still reaches a parallel efficiency of 69 per cent, whereas the parallel efficiency of the ADER-FV scheme drops down to 38 per cent. We note that for both codes we only used a very basic MPI implementation using non-blocking communication without any special optimizations. Typical runs of the parallel ADER-FV and ADER-DG codes on the HLRB2 supercomputer of the LRZ supercomputing centre in München and on the NEC Linux cluster (400 Intel Xeon EM64T CPUs, each with 3.2 GHz of clock speed and 1 GB of RAM) of the HLRS supercomputing centre in Stuttgart use between 64 and 128 processors.

The main inconvenience of ADER-FV schemes is the necessary reconstruction procedure in order to provide high-order accuracy in space from the given cell averages. This step is not necessary for ADER-DG schemes since they directly evolve all polynomial coefficients in time. Although reconstruction can become quite cumbersome on unstructured tetrahedral meshes in three space dimensions, the systematic framework presented in Dumbser & Käser (2007) and outlined also in this article is still quite easy to implement due to the use of hierarchical orthogonal reconstruction basis functions Ψ and the transformation of the reconstruction stencils to a reference coordinate system aligned with the central element. The resulting algorithm is robust and in particular it is also cost-efficient. As we will see later when analyzing in detail the CPU times of all steps required by ADER-DG and ADER-FV, the reconstruction operator is still less expensive than the ADER-DG flux computation. To increase computational efficiency, we store the inverse of the reconstruction matrix in (45) for each element and then multiply the vector of cell averages in the stencil $S^{(m)}$ with this inverse matrix. Although this speeds up reconstruction considerably, the associated memory load is quite high. Most of the memory requirements of our proposed ADER-FV scheme are due to the storage of these inverse reconstruction matrices. Due to the least-squares reconstruction approach using twice the number of necessary elements in 3-D, the

Table 6. Detailed CPU time comparison for the individual steps necessary for ADER-FV and ADER-DG schemes in 3-D per element and time step for the purely elastic wave equations, without attenuation. Note that no compiler optimization has been used.

	ADER-FV $\mathcal{O}4(\Delta t_{FV} = 2.64E-002)$		ADER-DG $\mathcal{O}4(\Delta t_{DG} = 3.77E-003)$	
	CPU time	Percentage	CPU time	Percentage
Reconstruction	0.10 ms	20 per cent	0 ms	0 per cent
Cauchy-Kovalewski procedure	0.12 ms	24 per cent	0.11 ms	10 per cent
Flux & stiffness computation	0.27 ms	56 per cent	1.03 ms	90 per cent
Total	**0.48 ms**	100 per cent	**1.14 ms**	100 per cent
	ADER-FV $\mathcal{O}6(\Delta t_{FV} = 2.64\mathrm{E}-002)$		ADER-DG $\mathcal{O}6(\Delta t_{DG} = 2.40\mathrm{E}-003)$	
	CPU time	Percentage	CPU time	Percentage
Reconstruction	0.53 ms	35 per cent	0 ms	0 per cent
Cauchy-Kovalewski procedure	0.53 ms	35 per cent	0.56 ms	12 per cent
Flux & stiffness computation	0.46 ms	30 per cent	3.99 ms	88 per cent
Total	**1.52 ms**	100 per cent	**4.54 ms**	100 per cent

inverse reconstruction matrices have the size $L \times 2L$. Fortunately, they have to be stored only once for each element, independent of the number of variables n_v of the system (13).

After this general discussion, we would like to show the reader in more detail the difference in CPU time requirements of ADER-DG and ADER-FV. On this behalf we run the same convergence test as shown in the previous Section 6 on the same mesh once with an ADER-DG scheme and once with an ADER-FV scheme. The computer system is in both cases the same Intel Xeon workstation with 3.6 GHz and 4 GB of RAM. Both codes run in their serial version, without making use of the sparsity of the flux matrices and without using compiler optimizations. We then take the detailed function timing profile generated by the profiler after the run. The CPU times needed for the reconstruction, the Cauchy-Kovalewski procedure and the flux computation are presented in Tables 6 and 7 per element and time step, without making use of compiler optimization. We note that for ADER-DG no reconstruction is necessary and that the times for flux computation also include the stiffness matrices and the evaluation of the source term on the right hand side of (76). All the points mentioned in the previous general discussion are confirmed by the results presented in Tables 6 and 7. The reconstruction is only used for ADER-FV schemes and the Cauchy-Kovalewski procedure is the same in both methods and, therefore, also needs roughly the same CPU time. Furthermore, the flux computation of ADER-DG is much more expensive than the one for the ADER-FV scheme due to the larger flux matrices and the presence of the stiffness terms. However, we would like to note that for ADER-DG schemes some CPU time can be saved making use of the sparsity of the flux matrices. The CPU times presented in Dumbser & Käser (2006) and Käser *et al.* (2006) and in Table 8 use this property and are obtained with maximum compiler optimizations in order to speed up the code. With the optimized ADER-FV and ADER-DG codes, we obtain the following CPU time needed per tetrahedral element and time step by the fourth-order ADER-FV and ADER-DG scheme for the 3-D anelastic wave equations with 5 attenuation mechanisms: 0.24 ms for ADER-FV and 0.45 ms for ADER-DG, respectively. We finally have to consider the difference in the time step limit, which explains the different number of iterations and the large discrepancy in CPU times shown in Tables 5 and 8. However, especially having a look at the error norms presented in Tables 5 and 8 we have to emphasize that ADER-DG schemes are much more accurate than ADER-FV methods. This allows considerably coarser meshes for ADER-DG to reach the same precision as ADER-FV which at the same time reduces computational effort due to the reduced number of elements and due to the larger time step induced by the coarser mesh.

Comparing the CPU times given in this article for ADER-FV schemes, see Table 5 and comparing with the CPU times obtained for the same test problem using ADER-DG schemes (see Käser *et al.* 2006) and Table 8, we note that on the same mesh the fourth-order FV

Table 7. Detailed CPU time comparison for the individual steps necessary for ADER-FV and ADER-DG schemes in 3-D per element and time step for the anelastic wave equations using 5 attenuation mechanisms. Note that no compiler optimization has been used.

	ADER-FV $\mathcal{O}4(\Delta t_{FV} = 2.64E-002)$		ADER-DG $\mathcal{O}4(\Delta t_{DG} = 3.77E-003)$	
	CPU time	Percentage	CPU time	Percentage
Reconstruction	0.34 ms	25 per cent	0 ms	0 per cent
Cauchy-Kovalewski procedure	0.55 ms	40 per cent	0.56 ms	27 per cent
Flux & stiffness computation	0.48 ms	35 per cent	1.47 ms	73 per cent
Total	**1.36 ms**	100 per cent	**2.03 ms**	100 per cent
	ADER-FV $\mathcal{O}6(\Delta t_{FV} = 2.64E-002)$		ADER-DG $\mathcal{O}6(\Delta t_{DG} = 2.40E-003)$	
	CPU time	Percentage	CPU time	Percentage
Reconstruction	2.06 ms	28 per cent	0 ms	0 per cent
Cauchy-Kovalewski procedure	4.09 ms	57 per cent	4.13 ms	39 per cent
Flux & stiffness computation	1.07 ms	15 per cent	6.36 ms	61 per cent
Total	**7.22 ms**	100 per cent	**10.49 ms**	100 per cent

Table 8. Convergence orders for the velocity component v for the ADER-DG $\mathcal{O}4$ scheme. Viscoelastic attenuation is modeled using five mechanisms.

h	L^∞	$\mathcal{O}_{L^\infty}$	L^2	$\mathcal{O}_{L^2}$	I	CPU (s)
2.16×10^{-1}	5.4011×10^{-4}	–	3.2609×10^{-4}	–	28	35
1.44×10^{-1}	1.4012×10^{-4}	3.3	5.7198×10^{-5}	4.3	40	168
1.08×10^{-1}	4.3978×10^{-5}	4.0	1.7152×10^{-5}	4.2	54	504
7.21×10^{-2}	9.0642×10^{-6}	3.9	3.2404×10^{-6}	4.1	80	2514

schemes are about twelve times faster than the corresponding ADER-DG schemes. However, the ADER-DG method is also about ten times more accurate. At the end, the considerable advantage in accuracy makes the ADER-DG scheme slightly superior to ADER-FV schemes comparing the CPU time needed by both methods at the same level of accuracy in L^2 norm. In L^∞ norm, ADER-FV is slightly more efficient. However, there may be important realistic applications where the coarse meshes needed by ADER-DG to be competitive with ADER-FV are not realizable. This is the case, for example, when small geometrical features have to be resolved by the mesh, such as complex surface topography, complex layered sediment structures embedded in the model or also thin layers with different material properties. In all these cases, the final mesh resolution is almost fixed by the requirement of resolving all the small features. Since ADER-FV is much faster than ADER-DG on the same mesh, though less accurate, it may be the preferable method of choice in such cases. Since both methods are able to treat unstructured meshes and since both schemes have many common parts, we are running the ADER-FV and ADER-DG schemes in the same software package in order to be flexible to opt either for ADER-DG or ADER-FV, depending on the requirements of the test case.

A final consideration is devoted to an estimation of the memory requirements of ADER-DG and ADER-FV schemes without considering the memory requirements of the unstructured mesh information. We compare ADER-FV schemes and ADER-DG methods with polynomial degree M for a number of n attenuation mechanisms. The number of reals that has to be stored per element in three space dimensions is for ADER-DG schemes

$$N_{\mathrm{DOF}}^{\mathrm{DG}} = (9+6n)\cdot 2\cdot \frac{1}{6}(M+1)(M+2)(M+3) \tag{78}$$

and for ADER-FV schemes

$$N_{\mathrm{DOF}}^{\mathrm{FV}} = (9+6n)\left(1+\frac{1}{6}(M+1)(M+2)(M+3)\right)+\frac{1}{3}(M+1)^2(M+2)^2(M+3)^2\,. \tag{79}$$

For ADER-DG methods, we only have to store two sets of degrees of freedom for each variable of the governing PDE, namely for the time-integrated degrees of freedom and for the new time level, see (78). For ADER-FV schemes, data are stored in each element in form of cell averages, that is, we store one set of cell averages for the new time level and then we also need to compute and store for each element the set of the time-integrated reconstructed degrees of freedom, see the first term in (79). However, the ADER-FV approach also requires the storage of the inverse of the reconstruction matrix in (45), which grows rapidly with the degree M of the method but which does not depend on the number of variables $n_v = 9 + 6n$ of the system. All other local matrices appearing in the two methods (75) and (76), such as, for example, $\check{T}^j_{pq}$ and $\check{A}^*_{pq}$, $\check{B}^*_{pq}$, $\check{C}^*_{pq}$, $\check{E}^*_{pq}$, etc. are not stored but recomputed in each time step. We emphasize that the flux matrices $F^{-,j}_{kl}$ and $F^{+,j,i,h}_{kl}$ are the same for all elements since they are computed only once on the reference tetrahedron and thus do *not contribute* to the memory requirements. Furthermore, the tensor I_{pqlm} is needed only formally since the time integration via the Cauchy-Kovalewski procedure is done very efficiently in each time step applying the algorithm described in Käser *et al.* (2006). To give an example for eqs (78) and (79), a third-order ADER-DG scheme for $n = 5$ attenuation mechanisms would need to store 780 reals per element whereas the third-order FV method for $n = 5$ already has to store 1629 reals. For the fourth-order version of the schemes, we already need to store 1560 reals for the DG method and even 5619 reals for the finite volume scheme. Considering the important fact that DG schemes reach the same accuracy than FV schemes on a much coarser grid, the DG schemes are much less memory intensive than the FV methods. It is, however, very important to mention that these large storage requirements for the FV schemes are mainly due to the unstructured tetrahedral formulation where we need to store the inverse of the reconstruction matrices for all elements. On regular Cartesian grids the fourth-order ADER-FV approach would have to store only 819 reals per element instead of 5619 reals.

In the following section, we will present the results obtained with ADER FV schemes for several standard benchmark problems.

8 APPLICATION EXAMPLES

We apply the proposed ADER-FV method on well-defined 2- and 3-D test problems for which also analytic reference solutions are available. The 2-D benchmark is the well-known Lamb's problem (Lamb 1904) in the same setup as given in Komatitsch & Vilotte (1998) and Käser & Dumbser (2006) to verify the accuracy of the scheme at the free-surface boundary. The two 3-D benchmark problems LOH.1 and LOH.3 were published in the final report of the *LIFELINES PROGRAM TASK 1A02* (Day *et al.* 2003) of the Pacific Earthquake Engineering Research Center and are part of a multi-institutional code validation project of a series of different numerical methods employed in numerical modeling 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. Furthermore, reference solutions are provided by the ADER-DG scheme proposed by the authors (Käser & Dumbser 2006; Dumbser & Käser 2006; Käser *et al.* 2006). Both LOH test cases contain a heterogeneous

layered medium with a free surface boundary condition. Whereas LOH.1 solves only the purely elastic wave equations without attenuation, the LOH.3 benchmark also includes viscoelastic behaviour with its associated attenuation and dispersion mechanisms. The results of the four reference codes given in Day *et al.* (2003) 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 FD 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 FE. 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.

8.1 Lamb's problem in two space dimensions

A classical test case to validate the implementation of free surface boundary conditions and point sources is Lamb's problem (Lamb 1904), consisting in a vertical (with respect to the surface) point force acting *on* the free surface. Details on how to implement point sources in ADER-DG schemes are given in Käser & Dumbser (2006). Due to the similar mathematical formulation, the same implementation can also be used for ADER-FV schemes. The solution of Lamb's problem for a plane surface can be computed analytically (see e.g. Pilant 1979) and can hence be used to assess the quality of numerical methods. In this paper, we use the FORTRAN code `EX2DDIR` of Berg *et al.* (1994) to compute the exact 2-D solution of the seismic response from a vertical directional point source in an elastic half-space with a free surface. The code `EX2DDIR` is based on the Cagniard-de Hoop technique (de Hoop 1960) and allows the use of an arbitrary source time function for displacements or velocities. Considering the accuracy of a numerical method with respect to the correct treatment of sources and the free-surface boundary condition, Lamb's problem poses a challenging test case in particular because of the Rayleigh waves propagating along the free surface.

The setup of the physical problem is chosen as in the paper of Komatitsch & Vilotte (1998), who solved this problem using the SEM (see e.g. Komatitsch & Tromp 1999, 2002). Furthermore, this problem was solved in Käser & Dumbser (2006) on very coarse triangular meshes using a 10th-order ADER-DG scheme.

We use a homogeneous elastic medium with a *P*-wave velocity of $c_p = 3200\,\mathrm{m\,s^{-1}}$, an *S*-wave velocity of $c_s = 1847.5\,\mathrm{m\,s^{-1}}$ and a mass density of $\rho = 2200\,\mathrm{kg\,m^{-3}}$. The numerical model with origin (0, 0) at the left bottom corner is 4000 m wide and has a height of 2000 m on the left boundary. The tilt angle of the free surface is $\phi = 10^\circ$. The directional point source, acting as a force perpendicular to this tilted surface, is located at the free surface at $\vec{x}_s = (1720.00,\ 2303.28)^T$. The two receivers are located at (2694.96, 2475.18) and (3400.08, 2599.52) such that their distances from the source along the surface are 990 m and 1706 m, respectively. On the left, right and bottom boundaries of the model we use then absorbing boundary conditions as described in Section 5. We use a sixth-order ADER-FV scheme on a triangular mesh built in such a way that the free surface boundary at the top is resolved with 800 triangles, the left and right boundaries of the model are discretized using 75 triangles refined towards the surface, and at the bottom 50 elements are used. The resulting mesh consists of 194 130 triangles, which is about 57 times more than the mesh used in Käser & Dumbser (2006) for the 10th-order ADER-DG scheme using only 3416 triangles. In order to avoid undesired effects of possibly reflected wave energy at the right model boundary, we extended the mesh up to a width of 4700 m for the numerical computations. The source time function that specifies the temporal variation of the point source is a Ricker wavelet given by

$$S^T(t) = a_1 \left(0.5 + a_2 (t - t_D)^2\right) e^{a_2 (t - t_D)^2}, \tag{80}$$

where $t_D = 0.08\,\mathrm{s}$ is the source delay time and $a_1 = -2000\,\mathrm{kg\,m^{-2}\,s^{-2}}$ and $a_2 = -(\pi f_c)^2$ are constants determining the amplitude and frequency of the Ricker wavelet of central frequency $f_c = 14.5$ Hz.

The wave propagation is simulated until time $T_{\mathrm{end}} = 1.3$ s when all waves have already passed the two receivers. We repeat that for the results shown in this paper, a sixth-order ADER-FV $\mathcal{O}6$ scheme with a Courant number of CFL = 0.5 is used. In order to reach the final simulation time $T_{\mathrm{end}} = 1.3$ we need 3233 time steps. In Fig. 6, we present the snapshots of the absolute value of the velocity vector of the seismic wave field at $t = 0.48$ s. In Fig. 7, we present the unscaled seismograms obtained from our numerical simulations, as recorded by receiver 1 and 2, respectively, together with the analytic solution provided by `EX2DDIR`. The analytic and numerical solutions match very well, such that the lines are hardly distinguishable on this scale. Therefore, the difference between analytic and numerical solution is also plotted. The maximum relative error on the regular mesh remains always less than 15 per cent. We conclude from this example that the accurate solution of Lamb's problem with the ADER-FV method proposed in this article confirms that the implementation of free surface boundary conditions as suggested in Section 5.2 leads to the correct physical behaviour of elastic surface waves.

We emphasize that the 10th-order ADER-DG scheme presented in Käser & Dumbser (2006) was able to produce errors of less than 1 per cent on the much coarser mesh for the same test problem. This underlines the important trade-off between speed and accuracy of ADER-FV and ADER-DG schemes, respectively, that always has to be taken into account.

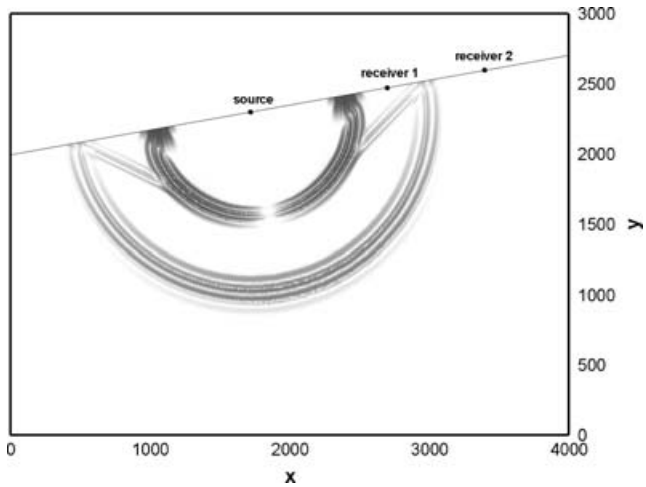

Figure 6. Absolute value of the velocity vector ($\sqrt{u^2+v^2}$) at $t = 0.48$ s for Lamb's problem obtained on an unstructured triangular mesh using an ADER-FV $\mathcal{O}6$ scheme.

8.2 Layer over half-space test cases LOH.1 and LOH.3 in three space dimensions

The setup of the test problems LOH.1 and LOH.3 is shown in Fig. 8(a), where for clarity only one of four symmetrical quarters of the complete computational domain $\Omega = [-15\,000\,\text{m}, 15\,000\,\text{m}] \times [-15\,000\,\text{m}, 15\,000\,\text{m}] \times [0\,\text{m}, 17\,000\,\text{m}]$ is plotted. The material parameters of the thin layer (Medium 1) with thickness 1000 m over the half-space (Medium 2) are given in Table 9. For the elastic case LOH.1, the quality factors are set to infinity. 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 $M_{xy} = M_{yx} = M_0 = 10^{18}$ N m. The location of the point source is $(x_s, y_s, z_s) = (0\,\text{m}, 0\,\text{m}, 2000\,\text{m})$, that is, in the centre of the xy plane of the domain Ω in 2000 m depth. The moment-rate time history is given through the source time function

$$S^T(t) = \frac{t}{T^2}\exp\left(-\frac{t}{T}\right), \tag{81}$$

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 outlined in previous work (Käser & Dumbser 2006). The same methodology applies also to ADER-FV schemes.

The signals are recorded up to a simulation time of 9 s by 10 receivers on the free surface as indicated in Fig. 8(a). The receiver locations are $(x_i, y_i, z_i) = (i \cdot 600\,\text{m}, i \cdot 800\,\text{m}, 0\,\text{m})$, for $i = 1, \ldots, 10$. The computational domain Ω is discretized by an unstructured, tetrahedral mesh as shown in Fig. 8(b) using 2100 114 elements. For comparison, the mesh used for the ADER-DG computations shown in Dumbser & Käser (2006) and Käser *et al.* (2006) contained only 249 338 elements. 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 100 m. This mesh spacing was also required in the setup of the benchmark for FD schemes. In other zones the mesh is coarsened up to average edge lengths of 1000 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, since also in the ADER-FV framework the reconstructed numerical solution is available in form of polynomials within each element and, therefore, can be evaluated at any position within an element. 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 2 as the faces of the tetrahedral elements are aligned with the material interface as shown in Figs 8(a) and (b).

In the following, we present the comparison of our results obtained with a fourth-order ADER-FV scheme and the four results of the reference codes (UCBL, UCSB, WCC2 and CMUN) against the analytic solution. Furthermore, the results obtained in Dumbser & Käser (2006) and Käser *et al.* (2006) with a fourth-order ADER-DG method are shown. Analogous to the LOH.1 and LOH.3 test cases in the *LIFELINES PROGRAM TASK 1A02* the visual comparisons in Figs 9 and 10 show the radial, transversal and vertical components of the seismic velocity field recorded at receiver 10 at $(x_{10}, y_{10}, z_{10}) = (6000\,\text{m}, 8000\,\text{m}, 0\,\text{m})$. Additionally, each plot gives the relative seismogram misfit

$$E = \sum_{j=1}^{n_t}\left(s_j - s_j^a\right)^2 \Big/ \sum_{j=1}^{n_t}\left(s_j^a\right)^2, \tag{82}$$

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 analytic value. 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).

Whereas all solutions agree quite well for the LOH.1 test case, see Fig. 9, the four reference solutions shown in Figs 10(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

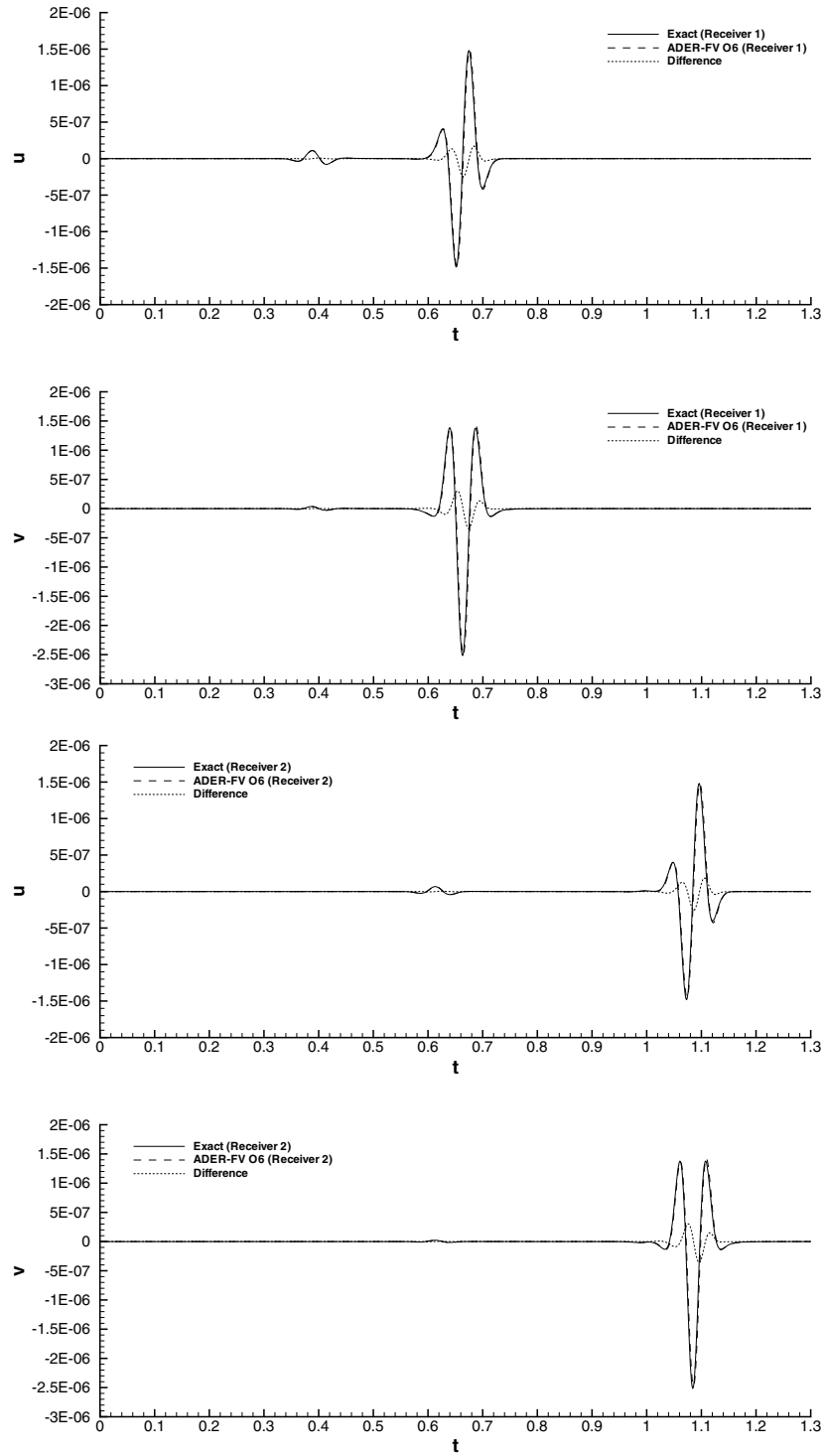

Figure 7. Seismograms of the normal and tangential velocity components with respect to the surface at the two receivers 1 and 2 for Lamb's problem.

(e.g. for UCSB) are quite notable. In addition, the results of UCBL, WCC2 and CMUN produce strong, unphysical oscillations in the transverse component.

The results with the fourth-order ADER-FV scheme in Figs 9(f) and 10(f) using three attenuation mechanisms for the LOH.3 case clearly match the analytic solution much better than the FD schemes UCBL, UCSB and WCC using the *same maximal mesh resolution* of 100 m and using the *same order of accuracy* of four. The results obtained with the ADER-FV scheme show lower numbers for the misfit E, indicating that the FV approach is more appropriate to discretize the free-surface boundary condition and the discontinuous material jump between Medium 1 and 2 than the FD staggered grid approach. The accuracy of the ADER-FV method is in both cases (LOH.1 and LOH.3) comparable to the ADER-DG $\mathcal{O}4$ scheme shown in Figs 9(e) and 10(e). We note, however, that the mesh used for the ADER-FV computation presented in

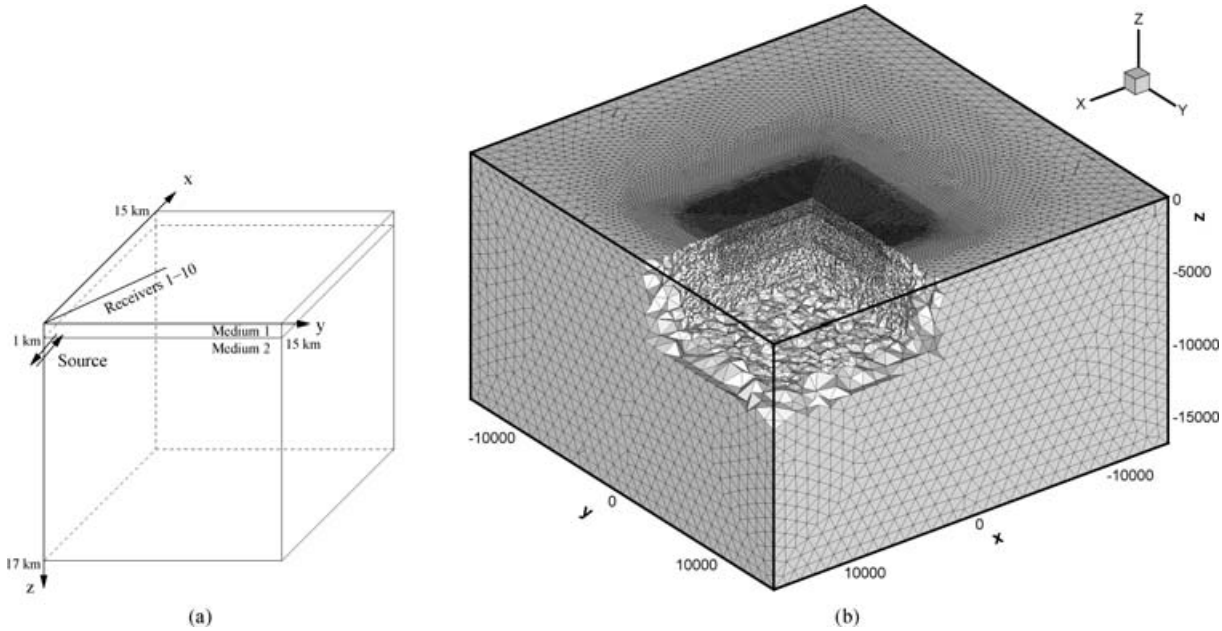

Figure 8. (a) One of four symmetric quarters is shown for the LOH test cases, where a layer of 1000 m (Medium 1) is lying on top of another layer (Medium 2). The source is a point dislocation at 2000 m depth represented by a moment tensor with the only non-zero components $M_{xy} = M_{yx}$. (b) Cut into the discretization of the LOH.1 and LOH.3 model to visualize the problem-adapted tetrahedral mesh, which is refined under the receiver line.

Table 9. Material parameters for the LOH.1 and LOH.3 test cases. Note, that attenuation will cause dispersion of the *P* and *S* waves such that the given wave speeds and material parameters for LOH.3 refer to a reference frequency $f_r = 2.5$ Hz. For the LOH.1 test case λ and μ are the *unrelaxed* material parameters and Q_p and Q_s are set to infinity.

	$c_p(f_r)$ (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

this article is about two to three times finer in each space dimension with respect to the mesh used for the ADER-DG scheme in Dumbser & Käser (2006) and Käser *et al.* (2006). We finally would like to emphasize that our solution was computed on an unstructured tetrahedral mesh, whereas the fourth-order FD reference solutions have all been computed on a regular Cartesian grid. The different underlying mesh topology makes of course our high-order FV approach much more expensive for this particular test case compared to the FD schemes. However, we emphasize that the geometry of this test case is very simple and that mesh generation does not pose any problem. The capability of our schemes to run on tetrahedral meshes, which can be generated in an almost fully automatic manner even for very complex geometries, makes them especially attractive for test cases with a very complex geometry where Cartesian or even hexahedral grid generation can become very difficult or even completely impossible. For a final performance comparison between our new unstructured ADER-FV approach and classical FD schemes, it is either necessary to take into account also the time needed for mesh generation, especially considering that for tetrahedral mesh generation almost no manual interaction of the user is necessary, whereas structured hexahedral grids are very difficult to generate for complex geometries and may need a lot of time-consuming manual user interaction. Or, it is necessary to compare ADER-FV and FD schemes on the same Cartesian mesh topology, which has been already done for the 2-D case in Schwartzkopff *et al.* (2004).

9 CONCLUSION

We have presented a new class of quadrature-free FV schemes of arbitrary order of accuracy in space and time (ADER-FV schemes) on unstructured triangular and tetrahedral meshes to simulate the propagation of seismic waves, incorporating the effect of attenuation and dispersion observed in realistic anelastic material. The convergence results demonstrate the high accuracy obtained with ADER-FV schemes on tetrahedral meshes. A thorough comparison of ADER-FV with the ADER-DG schemes previously published by the authors was presented, emphasizing especially the common parts of both schemes as well as the differences. Although ADER-FV is much faster than ADER-DG on the same mesh, the FV method is less accurate than DG. Therefore, ADER-FV schemes might be more useful in those cases where very fine meshes are needed in order to capture small geometrical features.

The solution of well-established benchmark tests using the proposed ADER-FV method and the comparison of the obtained results against analytic solutions clearly shows the increase in accuracy with respect to reference solutions obtained by other methods. We emphasize

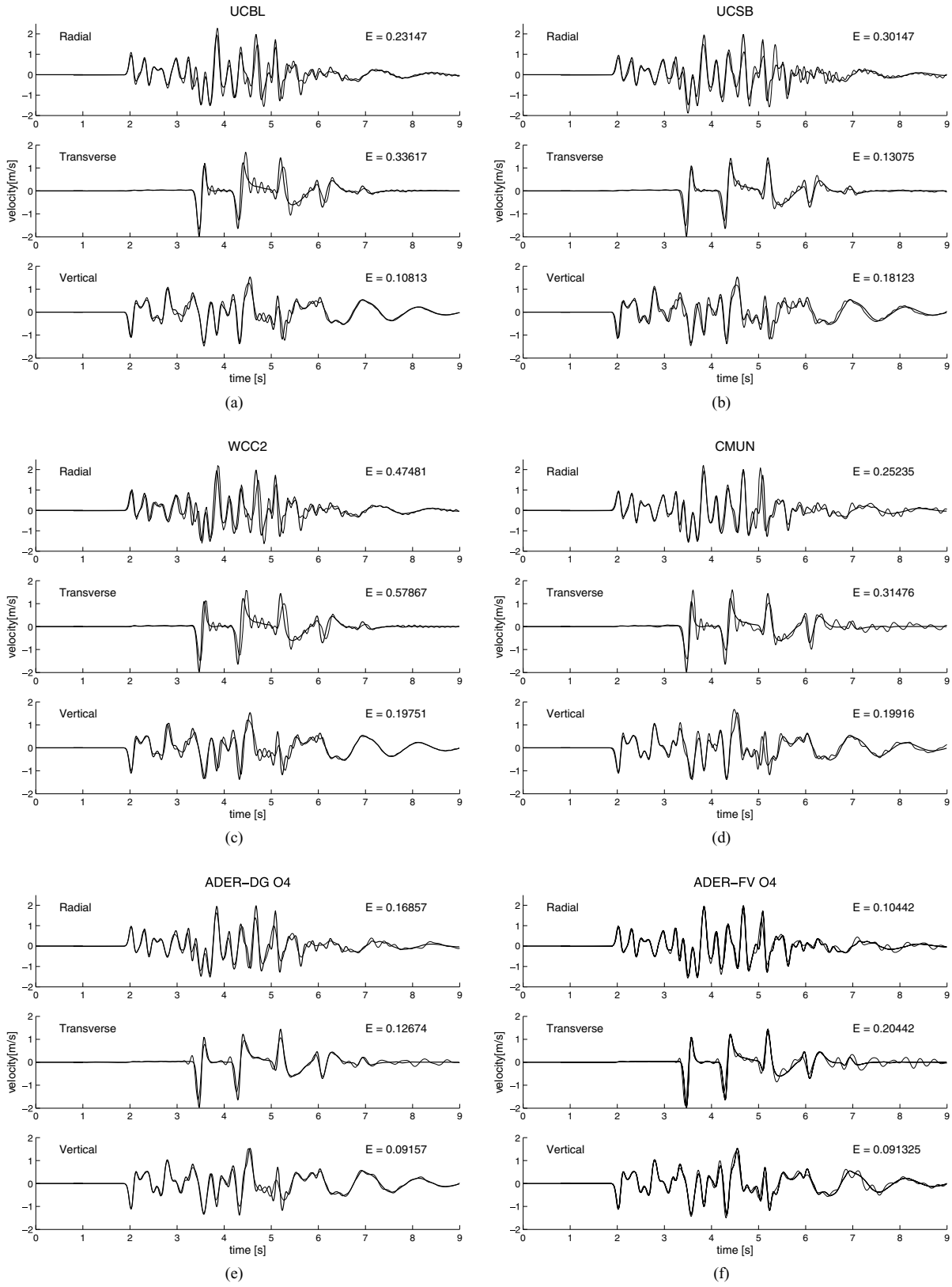

Figure 9. Comparison of the radial, transverse and vertical velocity components for the LOH.1 test case on receiver 10. The analytic solution (thick line) is plotted against the numerical one (thin line) obtained by (a) UCBL, (b) UCSB, (c) WCC2, (d) CMUN, (e) ADER-DG $\mathcal{O}4$ and (f) ADER-FV $\mathcal{O}4$. The relative seismogram misfit E from eq. (82) is given for each trace.

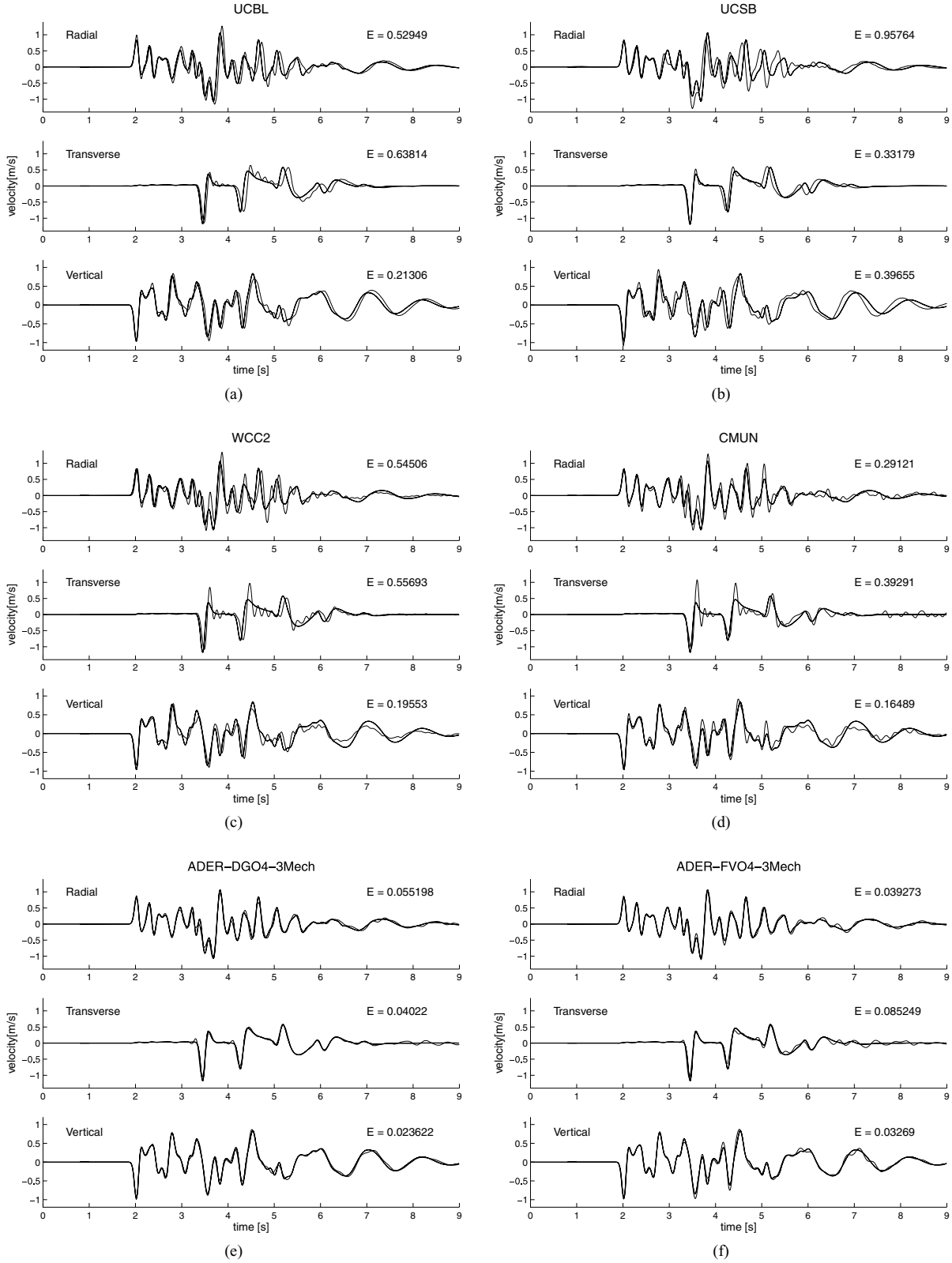

Figure 10. Comparison of the radial, transverse and vertical velocity components for the LOH.3 test case on receiver 10. The analytic solution (thick line) is plotted against the numerical one (thin line) obtained by (a) UCBL, (b) UCSB, (c) WCC2, (d) CMUN, (e) ADER-DG $\mathcal{O}4$ with three attenuation mechanisms and (f) ADER-FV $\mathcal{O}4$ with three attenuation mechanisms. The relative seismogram misfit E from eq. (82) is given for each trace.

that in both the LOH.1 and the LOH.3 test case the mesh spacing used was similar to the one used by the FD reference codes. However, our ADER-FV results are more accurate. We think that this is due to the fact that the free-surface boundary condition and the discontinuous jump in material properties can be handled much better by the flux formulation of the FV framework, using the solution of (Generalized) Riemann Problems at the element interfaces. Furthermore, realistic attenuation due to viscoelasticity is thoroughly included in our approach since also the viscoelastic properties are discretized with the same order of accuracy in space and time as the purely elastic case. Therefore, the proposed ADER-FV methods represents a new numerical scheme simulating accurately seismic wave propagation with realistic viscoelastic attenuation on unstructured 3-D tetrahedral meshes. Due to the lower CPU times with respect to the expensive ADER-DG schemes, it may be of interest for realistic applications in complex geometries, where the level of accuracy provided by ADER-DG is not needed.

ACKNOWLEDGMENTS

The presented work was supported by the *Deutsche Forschungsgemeinschaft* through the *DFG Forschungsstipendium* (DU 1107/1-1) and the *Emmy Noether-Programm* (KA 2281/1-1), the 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 to set up the LOH.1 and LOH.3 test cases and providing the analytic 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. Finally, we would like to thank explicitly the three referees of this paper who helped to improve the quality and the clarity of this article.

REFERENCES

Abgrall, R., 1994. On essentially non-oscillatory schemes on unstructured meshes: analysis and implementation, *J. Computat. Phys.,* **144,** 45–58.

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.

Barth, T.J. & Frederickson, P.O., 1990. Higher order solution of the Euler equations on unstructured grids using quadratic reconstruction, *AIAA paper no. 90-0013,* 28th Aerospace Sciences Meeting, January 1990.

Bérenger, 1994. A perfectly matched layer for the absorption of electromagnetic waves, *J. Comput. Phys.,* **114,** 185–200.

Berg, P., If, F., Nielsen, P. & Skovgaard, O., 1994. Analytical reference solutions, in *Modeling the Earth for Oil Exploration,* pp. 421–427, ed. Helbig, K., Pergamon Press, Brussels.

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

Braun, J. & Sambridge, M., 1995. A numerical method for solving partial differential equations on highly irregular evolving grids, *Nature,* **376,** 655–660.

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.

Collino, F. & Tsogka, C., 2001. Application of the PML absorbing layer model to the linear elastodynamic problem in anisotropic heterogeneous media, *Geophysics,* **66,** 294–307.

Cockburn, B. & Shu, C.W., 1998. The Runge–Kutta discontinuous Galerkin method for conservation laws V: multidimensional systems, *J. Comput. Phys.,* **141,** 199–224.

Cockburn, B., Karniadakis, G.E. & Shu, C.W., 2000. *Discontinuous Galerkin Methods,* Lecture Notes in Computational Science and Engineering, Springer.

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

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.

de Hoop, A.T., 1960. A modification of Cagniard's method for solving seismic pulse problems, *Appl. Sci. Res.,* **B8,** 349–356.

Dormy, E. & Tarantola, A., 1995. Numerical simulation of elastic wave propagation using a finite volume method, *J. geophys. Res.,* **100,** 2123–2133.

Dubiner, M., 1991. Spectral methods on triangles and other domains, *J. Sci. Comput.,* **6,** 345–390.

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., 2007. Arbitrary high order non-oscillatory Finite Volume schemes on unstructured meshes for linear hyperbolic systems, *J.Computat. Phys.,* **221,** 693–723.

Dumbser, M., Schwartzkopff, T. & Munz, C.D., 2006. Arbitrary High Order Finite Volume Schemes for Linear Wave Propagation, in *Computational Science and High Performance Computing II,* pp. 129–144, eds Krause, E., Shokin, Y., Resch, M. & Shokina, N., Notes on Numerical Fluid Mechanics and Multidisciplinary Design (NNFM), Springer, Berlin, Heidelberg.

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.

Friedrich, O., 1998. Weighted essentially non-oscillatory schemes for the interpolation of mean values on unstructured grids, *J. Computat. Phys.,* **144,** 194–212.

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

Hu, C. & Shu, C.W., 1999. Weighted essentially non-oscillatory schemes on triangular meshes, *J. Computat. Phys.,* **150,** 97–127.

Karniadakis, G.E. & Sherwin, S.J., 1999. *Spectral/hp Element Methods in CFD,* Oxford University Press, Oxford.

Karypis, G. & Kumar, V., 1998. Multilevel k-way Partitioning Scheme for Irregular Graphs, *J. Parallel Distrib. Comput.,* **48**(1), 96–129.

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.

Käser, M. & Igel, H., 2001. Numerical simulation of 2D wave propagation on unstructured grids using explicit differential operators, *Geophys. Prospect.,* **49,** 607–619.

Käser, M. & Iske, A., 2005. ADER schemes on adaptive triangular meshes for scalar conservation laws, *J. Computat. Phys.,* **205,** 486–508.

Käser, M., Igel, H., Sambridge, M. & Braun, J., 2001. A comparative study of explicit differential operators on arbitrary grids, *J. Comput. Acoustics,* **9,** 1111–1125.

Käser, M., Dumbser, M., de la Puente, J. & Igel, H., 2006. An arbitrary high order discontinuous galerkin method for elastic waves on unstructured meshes III: viscoelastic attenuation. *Geophys. J. Int.,* **168,** 224–242.

Komatitsch, D. & Tromp, J., 1999. Introduction to the spectral-element method for 3-D seismic wave propagation, *Geophys. J. Int.,* **139,** 806–822.

Komatitsch, D. & Tromp, J., 2002. Spectral-element simulations of global seismic wave propagation—I. Validation, *Geophys. J. Int.,* **149,** 390–412.

Komatitsch, D. & Tromp, J., 2003. A perfectly matched layer absorbing boundary condition for the second-order seismic wave equation, *Geophys. J. Int.,* **154,** 146–153.

Komatitsch, D. & Vilotte, J.P., 1998. The spectral-element method: an efficient tool to simulate the seismic response of 2D and 3D geological structures, *Bull. seism. Soc. Am.,* **88,** 368–392.

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.

Lamb, H., 1904. On the propagation of tremors over the surface of an elastic solid, *Phil. Trans. R. Soc. Lond. Ser. A,* **203,** 1–42.

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.

Madariaga, R., 1976. Dynamics of an expanding circular fault. *Bull. seism. Soc. Am.,* **65,** 163–182.

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

Ollivier-Gooch, C. & Van Altena, M., 2002. A high-order-accurate unstructured mesh finite-volume scheme for the advection-diffusion equation, *J. Computat. Phys.,* **181,** 729–752.

Pilant, W.L., 1979. *Elastic Waves in the Earth,* Elsevier Scientific Publishing, Amsterdam.

Priolo, E., Carcione, J.M. & Seriani, G., 1994. Numerical simulation of interface waves by high-order spectral modeling techniques, *J. acoust. Soc. Am.,* **95,** 681–693.

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.

Schwartzkopff, T., Dumbser, M. & Munz, C.D., 2004. Fast high order ADER schemes for linear hyperbolic equations, *J. Computat. Phys.,* **197,** 532–539.

Seriani, G., 1998. 3-D large-scale wave propagation modeling by a spectral-element method on a Cray T3E multiprocessor, *Comput. Methods Appl. Mech. Eng.,* **164,** 235–247.

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

Stroud, A.H., 1971. *Approximate Calculation of Multiple Integrals,* Prentice-Hall Inc., Englewood Cliffs, New Jersey.

Tadi, M., 2004. Finite Volume Method for 2D Elastic Wave Propagation, *Bull. seism. Soc. Am.,* **94,** 1500–1509.

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

Titarev, V.A. & Toro, E.F., 2005. ADER schemes for three-dimensional nonlinear hyperbolic systems, *J. Computat. Phys.,* **204,** 715–736.

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

Toro, E.F. & Titarev, V.A., 2002. Solution of the generalized Riemann problem for advection-reaction equations, *Proc. Roy. Soc. London,* **458**(2018), 271–281.

Virieux, J., 1984. SH-wave propagation in heterogeneous media: velocity-stress finite-diffe4ence method, *Geophysics,* **49,** 1933–1942.

Virieux, J., 1986. P-SV wave propagation in heterogeneous media: Velocity-stress finite-difference method, *Geophysics,* **51,** 889–901.

Wang, Z.J., 2002. Spectral (finite) volume method for conservation laws on unstructured grids: basic formulation, *J. Comput. Phys.,* **178,** 210–251.

Wang, Z.J. & Liu, Y., 2002. Spectral (finite) volume method for conservation laws on unstructured grids II: extension to two-dimensional scalar equation, *J. Comput. Phys.,* **179,** 665–697.

Wang, Z.J. & Liu, Y., 2004. Spectral (finite) volume method for conservation laws on unstructured grids III: one-dimensional systems and partition optimization, *J. Sci. Comput.,* **20,** 137–157.

Wang, Z.J., Zhang, L. & Liu, Y., 2004. Spectral (finite) volume method for conservation laws on unstructured grids IV: extension to two-dimensional systems, *J. Comput. Phys.,* **194,** 716–741.

Zhang, J., 1997. Quadrangle-grid velocity-stress finite-difference method for elastic-wave-propagation simulation, *Geophys. J. Int.,* **131,** 127–134.

APPENDIX A: RECONSTRUCTION BASIS FUNCTIONS

We use orthogonal hierarchical basis functions as given in Cockburn *et al.* (2000) and Karniadakis & Sherwin (1999), which extend the original ideas of Dubiner (1991). 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 reconstruction 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 reconstruction basis functions Ψ_k defined by (A5) and (A9) constitute orthogonal basis systems with respect to the inner product on their respective reference elements.

A1 Two space dimensions

For triangles the reference element T_E is defined as

$$T_E = \left\{(\xi, \eta) \in \mathbb{R}^2 \mid 0 \le \xi \le 1 \ \wedge \ 0 \le \eta \le 1 - \xi\right\}. \tag{A4}$$

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

$$\Psi_{k(p,q)}(\xi, \eta) = \Theta_p^a(\delta) \cdot \Theta_{pq}^b(2\eta - 1) \tag{A5}$$

with

$$\delta = \frac{2\xi}{1 - \eta} - 1. \tag{A6}$$

The mono-index $k = k(p, q)$ is given as a function of the index pair (p, q). Our ordering for the reconstruction basis functions up to degree three (for a fourth order FV scheme) is, for example,

$$\begin{aligned}
\Psi_0 &= 1,\\
\Psi_1 &= \eta - 1 + 2\,\xi,\\
\Psi_2 &= -1 + 3\,\eta,\\
\Psi_3 &= \eta^2 - 2\,\eta + 6\,\xi\,\eta + 6\,\xi^2 - 6\,\xi + 1,\\
\Psi_4 &= 5\,\eta^2 + 10\,\xi\,\eta - 6\,\eta - 2\,\xi + 1,\\
\Psi_5 &= 1 - 8\,\eta + 10\,\eta^2,\\
\Psi_6 &= \eta^3 - 3\,\eta^2 + 12\,\eta^2\xi - 24\,\xi\,\eta + 3\,\eta + 30\,\eta\,\xi^2 + 12\,\xi + 20\,\xi^3 - 30\,\xi^2 - 1,\\
\Psi_7 &= 7\,\eta^3 - 15\,\eta^2 + 42\,\eta^2\xi + 9\,\eta - 48\,\xi\,\eta + 42\,\eta\,\xi^2 - 6\,\xi^2 - 1 + 6\,\xi,\\
\Psi_8 &= 21\,\eta^3 + 42\,\eta^2\xi - 33\,\eta^2 - 24\,\xi\,\eta + 13\,\eta - 1 + 2\,\xi,\\
\Psi_9 &= -1 + 15\,\eta - 45\,\eta^2 + 35\,\eta^3.
\end{aligned} \tag{A7}$$

A2 Three space dimensions

For tetrahedrons the reference element T_E is defined as

$$T_E = \left\{(\xi, \eta, \zeta) \in \mathbb{R}^3 \mid 0 \le \xi \le 1 \ \wedge \ 0 \le \eta \le 1 - \xi \ \wedge \ 0 \le \zeta \le 1 - \xi - \eta\right\}. \tag{A8}$$

Furthermore, the edge parameters χ and τ that parametrize the four triangular faces are defined in the reference triangle (A5). The basis functions $\Psi_k(\xi, \eta, \zeta)$ are defined on this reference element as the following product of the primal functions:

$$\Psi_{k(p,q,r)}(\xi, \eta, \zeta) = \Theta_p^a(\alpha) \cdot \Theta_{pq}^b(\beta) \cdot \Theta_{pqr}^c(\gamma). \tag{A9}$$

with

$$\alpha = \frac{\eta - 1 + \zeta + 2\,\xi}{1 - \eta - \zeta}, \quad \beta = \frac{2\,\eta - 1 + \zeta}{1 - \zeta}, \quad \gamma = -1 + 2\,\zeta. \tag{A10}$$

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

$$\begin{aligned}
\Psi_0 &= 1,\\
\Psi_1 &= \zeta - 1 + \eta + 2\,\xi,\\
\Psi_2 &= \zeta - 1 + 3\,\eta,\\
\Psi_3 &= -1 + 4\,\zeta,\\
\Psi_4 &= \zeta^2 + 6\,\zeta\,\xi - 2\,\zeta + 2\,\zeta\,\eta + 6\,\eta\,\xi - 2\,\eta + 1 - 6\,\xi + \eta^2 + 6\,\xi^2,\\
\Psi_5 &= \zeta^2 + 6\,\zeta\,\eta - 2\,\zeta + 2\,\zeta\,\xi + 1 - 2\,\xi + 10\,\eta\,\xi - 6\,\eta + 5\,\eta^2,\\
\Psi_6 &= \zeta^2 + 8\,\zeta\,\eta - 2\,\zeta + 1 - 8\,\eta + 10\,\eta^2,\\
\Psi_7 &= 6\,\zeta^2 + 6\,\zeta\,\eta - 7\,\zeta + 12\,\zeta\,\xi - 2\,\xi + 1 - \eta,\\
\Psi_8 &= 6\,\zeta^2 - 7\,\zeta + 18\,\zeta\,\eta + 1 - 3\,\eta,\\
\Psi_9 &= 1 - 10\,\zeta + 15\,\zeta^2.
\end{aligned} \tag{A11}$$

APPENDIX B: EXAMPLES OF FLUX MATRICES IN 2- AND 3-D

Flux matrices for a 2-D reconstruction of degree three (fourth order FV scheme) in two space dimensions:

$$
\begin{aligned}
F_l^{-,1} &= \begin{pmatrix} 1 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & -1 \end{pmatrix} \\
F_l^{-,2} &= \begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{2} & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \end{pmatrix} \\
F_l^{-,3} &= \begin{pmatrix} 1 & -\frac{1}{2} & \frac{1}{2} & \frac{1}{3} & -\frac{1}{3} & \frac{1}{3} & -\frac{1}{4} & \frac{1}{4} & -\frac{1}{4} & \frac{1}{4} \end{pmatrix} \\
F_l^{+,1,h} &= \begin{pmatrix} 1 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & -1 \end{pmatrix} \\
F_l^{+,2,h} &= \begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{2} & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \end{pmatrix} \\
F_l^{+,3,h} &= \begin{pmatrix} 1 & -\frac{1}{2} & \frac{1}{2} & \frac{1}{3} & -\frac{1}{3} & \frac{1}{3} & -\frac{1}{4} & \frac{1}{4} & -\frac{1}{4} & \frac{1}{4} \end{pmatrix}.
\end{aligned}
\tag{B1}
$$

Flux matrices for a 3-D reconstruction of degree two (third order FV scheme) in three space dimensions:

$$
\begin{aligned}
F_l^{-,1} &= \begin{pmatrix} \frac{1}{2} & 0 & 0 & -\frac{1}{2} & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} \end{pmatrix} \\
F_l^{-,2} &= \begin{pmatrix} \frac{1}{2} & 0 & -\frac{1}{3} & \frac{1}{6} & 0 & 0 & \frac{1}{4} & 0 & -\frac{1}{6} & \frac{1}{12} \end{pmatrix} \\
F_l^{-,3} &= \begin{pmatrix} \frac{1}{2} & -\frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{12} & -\frac{1}{12} & \frac{1}{12} & -\frac{1}{12} & \frac{1}{12} & \frac{1}{12} \end{pmatrix} \\
F_l^{-,4} &= \begin{pmatrix} \frac{1}{2} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{12} & \frac{1}{12} & \frac{1}{12} & \frac{1}{12} & \frac{1}{12} & \frac{1}{12} \end{pmatrix} \\
F_l^{+,1,1} &= \begin{pmatrix} \frac{1}{2} & 0 & 0 & -\frac{1}{2} & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} \end{pmatrix} \\
F_l^{+,1,2} &= \begin{pmatrix} \frac{1}{2} & 0 & 0 & -\frac{1}{2} & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} \end{pmatrix} \\
F_l^{+,1,3} &= \begin{pmatrix} \frac{1}{2} & 0 & 0 & -\frac{1}{2} & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} \end{pmatrix} \\
F_l^{+,2,1} &= \begin{pmatrix} \frac{1}{2} & 0 & -\frac{1}{3} & \frac{1}{6} & 0 & 0 & \frac{1}{4} & 0 & -\frac{1}{6} & \frac{1}{12} \end{pmatrix} \\
F_l^{+,2,2} &= \begin{pmatrix} \frac{1}{2} & 0 & -\frac{1}{3} & \frac{1}{6} & 0 & 0 & \frac{1}{4} & 0 & -\frac{1}{6} & \frac{1}{12} \end{pmatrix} \\
F_l^{+,2,3} &= \begin{pmatrix} \frac{1}{2} & 0 & -\frac{1}{3} & \frac{1}{6} & 0 & 0 & \frac{1}{4} & 0 & -\frac{1}{6} & \frac{1}{12} \end{pmatrix} \\
F_l^{+,3,1} &= \begin{pmatrix} \frac{1}{2} & -\frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{12} & -\frac{1}{12} & \frac{1}{12} & -\frac{1}{12} & \frac{1}{12} & \frac{1}{12} \end{pmatrix} \\
F_l^{+,3,2} &= \begin{pmatrix} \frac{1}{2} & -\frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{12} & -\frac{1}{12} & \frac{1}{12} & -\frac{1}{12} & \frac{1}{12} & \frac{1}{12} \end{pmatrix} \\
F_l^{+,3,3} &= \begin{pmatrix} \frac{1}{2} & -\frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{12} & -\frac{1}{12} & \frac{1}{12} & -\frac{1}{12} & \frac{1}{12} & \frac{1}{12} \end{pmatrix} \\
F_l^{+,4,1} &= \begin{pmatrix} \frac{1}{2} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{12} & \frac{1}{12} & \frac{1}{12} & \frac{1}{12} & \frac{1}{12} & \frac{1}{12} \end{pmatrix} \\
F_l^{+,4,2} &= \begin{pmatrix} \frac{1}{2} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{12} & \frac{1}{12} & \frac{1}{12} & \frac{1}{12} & \frac{1}{12} & \frac{1}{12} \end{pmatrix} \\
F_l^{+,4,3} &= \begin{pmatrix} \frac{1}{2} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{12} & \frac{1}{12} & \frac{1}{12} & \frac{1}{12} & \frac{1}{12} & \frac{1}{12} \end{pmatrix}.
\end{aligned}
\tag{B2}
$$