Arbitrary High Order Finite Volume Schemes for Seismic Wave Propagation on Unstructured Meshes in 2D and 3D

Michael Dumbser$^{1,2}$, Martin Käser$^1$, Josep de la Puente$^3$

$^1$ Department of Civil and Environmental Engineering, University of Trento, Italy
$^2$ Institut für Aerodynamik und Gasdynamik, Universität Stuttgart, Germany
$^3$ Department of Earth and Environmental Sciences, Geophysics Section, Ludwig-Maximilians-Universität, München, Germany

Accepted 1999 November 11. Received 1999 October 6; in original form 1999 August 3

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 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 uniform 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.

The numerical convergence analysis demonstrates that the proposed Finite Volume 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 three-dimensional wave propagation problems including realistic attenuation and complex geometry.

Key words: viscoelasticity, attenuation, Finite Volume schemes, high order accuracy, unstructured meshes, ADER approach

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 three-dimensional wave field 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, e.g. in (Braun & Sambridge 1995; Käser & Igel 2001; Käser, Igel, Sambridge & Braun 2001; Zhang 1997), provided numerical schemes with accuracies too low to be applied to realistic large scale problems. However, after the Spectral Element Methods (SEM) was introduced in the field of numerical seismology in (Priolo, Carcione & Seriani 1994; Seriani 1998), this spatially high order accurate scheme was further developed in (Komatitsch & Vilotte 1998; Komatitsch & Tromp 1999; Komatitsch & Tromp 2002). Later, a new numerical method based on a Discontinuous Galerkin (DG) approach in combination with a novel time integration scheme using Arbitrary high order DERivatives (ADER) was introduced in (Käser & Dumbser 2006a; Dumbser & Käser 2006a) to simulate elastic wave propagation of unstructured triangular and tetrahedral meshes with arbitrary high order of accuracy.

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 three-dimensional 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, Anderson & Kanamori 1976; Stein & Wyssession 2003). The correct numerical treatment of a viscoelastic medium is outlined in (Moczo, Kristek & Halada 2004). Day & Minster (1984) transformed the stress-strain relation in the time domain into a differential form and obtained additional internal variables, which replace the convolution integral. Emmerich & Korn (1987; 1992) improved this approach and showed that their method is superior in accuracy and efficiency and applied the viscoelastic models for the P-SV case. Independently, a different approach in (Carcione, Kosloff & Kosloff 1988; 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 finite-difference (FD) (Moczo & Bard 1993) methods. Finally, in (Käser, Dumbser, de la Puente & Igel 2006) viscoelastic attenuation was incorporated into the ADER-DG schemes.

In this paper, we introduce an arbitrary high order ADER Finite Volume (ADER-FV) scheme on unstructured triangular and tetrahedral meshes including elastic and viscoelastic media. In contrast to previous FV approaches, e.g. (Dormy & Tarantola 1995; Wang 2002; Wang & Liu 2002; Tadi 2004; Wang & Liu 2004; Wang, Zhang & Liu 2004) the proposed ADER-FV method is based on a new and efficient reconstruction operator for unstructured meshes in 2D and 3D developed by Dumbser and Käser in (Dumbser & Käser 2006b) and 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 2006b) is the first Finite Volume scheme on three-dimensional unstructured meshes of order higher than two. Former work on high order Finite Volume schemes on unstructured meshes was restricted to two space dimensions, see for example (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 2006a; Dumbser & Käser 2006a; Käser, Dumbser, de la Puente & Igel 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 Finite Volume 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 fine mesh. In this case, a less expensive Finite Volume 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 three-dimensional anelastic wave equations in velocity-stress formulation including attenuation due to viscoelasticity. The reconstruction operator needed for the Finite Volume scheme is briefly explained in Section 3 and the resulting ADER Finite Volume 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 in (Käser & Dumbser 2006a; Dumbser & Käser 2006a; Käser, Dumbser, de la Puente & Igel 2006). Finally, in Section 8, we present the numerical results obtained with the ADER-FV method for the two-dimensional Lamb’s problem and the three-dimensional test cases LOH.1 and LOH.3 proposed by the Pacific Earthquake Engineering Research Center (Day, Bielak, Dreger, Graves, Larsen, Olsen & Pitarka 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 in (Dumbser & Käser 2006a; Käser, Dumbser, de la Puente & Igel 2006).
\[ \sigma_i = \lambda e_{kk} \delta_{ij} + 2 \mu e_{ij} - \sum_{l=1}^{n} (\lambda Y_{i}^\lambda Y_{k}^\nu Y_{j}^\nu - 2 \mu Y_{i}^\mu Y_{k}^\nu Y_{j}^\nu) e^{\omega t} \int_{-\infty}^{t} e^{\omega (t-\tau)} d\tau. \]
The remaining problem is the evolution of the anelastic functions $\bar{\vartheta}_{ij}^\ell(t)$ in (8) in time. In fact, equation (8) is the solution of the partial differential equation

$$\frac{\partial}{\partial t} \bar{\vartheta}_{ij}^\ell(t) + \omega \varepsilon_{ij} \bar{\vartheta}_{ij}^\ell(t) = (\lambda + 2\mu) \bar{\vartheta}_{ij}^\ell(t),$$

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

However, to express the equation system in the velocity-stress formulation it is convenient to redefine the anelastic functions in the form

$$\vartheta_{ij}^\ell = \frac{\partial}{\partial t} \bar{\vartheta}_{ij}^\ell.$$  

Finally, using the equations of motion, the definition of strain $\varepsilon_{ij}$ and equations (9), (10) and (11) we can formulate the system of the anelastic wave equations as

\[
\begin{align*}
\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}^{\nu} - (\lambda Y_{\ell} + 2\mu Y_{\ell}^\nu) \vartheta_{xx}^\ell - \lambda Y_{\ell} \vartheta_{yy}^\ell - \lambda Y_{\ell} \vartheta_{zz}^\ell, \\
\frac{\partial}{\partial t} \sigma_{yy} - (\lambda + 2\mu) \frac{\partial}{\partial y} v - \lambda \frac{\partial}{\partial x} u - \lambda \frac{\partial}{\partial z} w &= \sum_{\ell=1}^{\nu} - \lambda Y_{\ell} \vartheta_{xx}^\ell - (\lambda Y_{\ell} + 2\mu Y_{\ell}^\nu) \vartheta_{yy}^\ell - \lambda Y_{\ell} \vartheta_{zz}^\ell, \\
\frac{\partial}{\partial t} \sigma_{zz} - (\lambda + 2\mu) \frac{\partial}{\partial z} w - \lambda \frac{\partial}{\partial x} u - \lambda \frac{\partial}{\partial y} v &= \sum_{\ell=1}^{\nu} - \lambda Y_{\ell} \vartheta_{xx}^\ell - (\lambda Y_{\ell} + 2\mu Y_{\ell}^\nu) \vartheta_{yy}^\ell - \lambda Y_{\ell} \vartheta_{zz}^\ell, \\
\frac{\partial}{\partial t} \sigma_{xy} - \mu \left( \frac{\partial}{\partial x} u + \frac{\partial}{\partial y} v \right) &= \sum_{\ell=1}^{\nu} - 2\mu Y_{\ell} \vartheta_{xy}^\ell, \\
\frac{\partial}{\partial t} \sigma_{yz} - \mu \left( \frac{\partial}{\partial y} v + \frac{\partial}{\partial z} w \right) &= \sum_{\ell=1}^{\nu} - 2\mu Y_{\ell} \vartheta_{yz}^\ell, \\
\frac{\partial}{\partial t} \sigma_{xz} - \mu \left( \frac{\partial}{\partial z} w + \frac{\partial}{\partial x} u \right) &= \sum_{\ell=1}^{\nu} - 2\mu Y_{\ell} \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} u + \frac{\partial}{\partial y} v \right) &= -\omega_1 \vartheta_{xy}^1, \\
\frac{\partial}{\partial t} \vartheta_{yz}^1 - \frac{1}{2} \omega_1 \left( \frac{\partial}{\partial y} v + \frac{\partial}{\partial z} w \right) &= -\omega_1 \vartheta_{yz}^1, \\
\frac{\partial}{\partial t} \vartheta_{xz}^1 - \frac{1}{2} \omega_1 \left( \frac{\partial}{\partial z} w + \frac{\partial}{\partial x} u \right) &= -\omega_1 \vartheta_{xz}^1, \\
\vdots & \\
\frac{\partial}{\partial t} \vartheta_{xx}^n - \omega_n \frac{\partial}{\partial x} u &= -\omega_n \vartheta_{xx}^n, \\
\frac{\partial}{\partial t} \vartheta_{yy}^n - \omega_n \frac{\partial}{\partial y} v &= -\omega_n \vartheta_{yy}^n, \\
\frac{\partial}{\partial t} \vartheta_{zz}^n - \omega_n \frac{\partial}{\partial z} w &= -\omega_n \vartheta_{zz}^n, \\
\frac{\partial}{\partial t} \vartheta_{xy}^n - \frac{1}{2} \omega_n \left( \frac{\partial}{\partial x} u + \frac{\partial}{\partial y} v \right) &= -\omega_n \vartheta_{xy}^n, \\
\frac{\partial}{\partial t} \vartheta_{yz}^n - \frac{1}{2} \omega_n \left( \frac{\partial}{\partial y} v + \frac{\partial}{\partial z} w \right) &= -\omega_n \vartheta_{yz}^n, \\
\frac{\partial}{\partial t} \vartheta_{xz}^n - \frac{1}{2} \omega_n \left( \frac{\partial}{\partial z} w + \frac{\partial}{\partial x} u \right) &= -\omega_n \vartheta_{xz}^n,
\end{align*}
\]

where $n$ is the number of mechanisms used to approximate a frequency-independent Q-law and $\rho$ is the density. Note, that each mechanism adds 6 further equations, i.e. one for each stress component. Therefore, the system of the purely elastic three-dimensional wave equations consisting of 9 equations increases by 6$n$ 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 2006a) 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

\[
\frac{\partial U}{\partial t} + \mathbf{A} \frac{\partial U}{\partial x} + \mathbf{B} \frac{\partial U}{\partial y} + \mathbf{C} \frac{\partial U}{\partial z} = \mathbf{E} \frac{\partial U}{\partial z},
\]

To obtain the two-dimensional case, we simply set \( \frac{\partial}{\partial z} = 0 \) and remove the corresponding equations for the stresses \( \sigma_z \), \( \sigma_{xz} \) and \( \sigma_{yz} \), the memory variables \( \vartheta_{zz} \), \( \vartheta'_{xz} \) and \( \vartheta'_{yz} \) as well as for the velocity \( \nu \) from the state vector \( U_p \) and the Jacobians. Note, that the dimensions of the variable vector \( U_p \), the Jacobian matrices \( \mathbf{A}, \mathbf{B}, \mathbf{C} \) and the source matrix \( \mathbf{E} \) 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 \( p, q \in [1, ..., 9] \) denote the elastic part and \( p, q \in [10, ..., n_v] \), denote the anelastic part of the system as presented in (12). As the Jacobian matrices \( \mathbf{A}, \mathbf{B} \) and \( \mathbf{C} \) as well as the source matrix \( \mathbf{E} \) 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:

\[
\mathbf{A}_{pq} = \begin{bmatrix} A & 0 \\ A_n & 0 \end{bmatrix} \in \mathbb{R}^{n_v \times n_v}, \quad \mathbf{B}_{pq} = \begin{bmatrix} B & 0 \\ B_n & 0 \end{bmatrix} \in \mathbb{R}^{n_v \times n_v}, \quad \mathbf{C}_{pq} = \begin{bmatrix} C & 0 \\ C_n & 0 \end{bmatrix} \in \mathbb{R}^{n_v \times n_v},
\]

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

\[
A_n = \begin{bmatrix} A_1 & \vdots & A_n \end{bmatrix} \in \mathbb{R}^{6n \times 9}, \quad B_n = \begin{bmatrix} B_1 & \vdots & B_n \end{bmatrix} \in \mathbb{R}^{6n \times 9}, \quad C_n = \begin{bmatrix} C_1 & \vdots & C_n \end{bmatrix} \in \mathbb{R}^{6n \times 9},
\]

where each sub-matrix \( A_\ell, B_\ell, C_\ell \in \mathbb{R}^{6 \times 9} \), with \( \ell = 1, ..., n \), contains the relaxation frequency \( \omega_\ell \) of the \( \ell \)-th mechanism in the form:

\[
A_\ell = \omega_\ell \cdot \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},
\]

\[
B_\ell = \omega_\ell \cdot \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},
\]

\[
C_\ell = \omega_\ell \cdot \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.
\]

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

\[
\mathbf{E}_{pq} = \begin{bmatrix} 0 & E \\ 0 & E' \end{bmatrix} \in \mathbb{R}^{n_v \times n_v},
\]

with \( E \) of the block structure

\[
E = [E_1, \ldots, E_n] \in \mathbb{R}^{9 \times 6n},
\]
6  Michael Dumbser, Martin Käser, Josep de la Puente

where each matrix $E_\ell \in \mathbb{R}^{3\times 6}$, with $\ell = 1, \ldots, n$, contains the anelastic coefficients $Y_\ell^\lambda$ and $Y_\ell^\mu$ of the $\ell$-th mechanism in the form:

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

(21)

The matrix $E'$ in (19) is a diagonal matrix and has the structure

$$
E' = \begin{bmatrix}
E'_1 & 0 \\
0 & \varepsilon \\
& \varepsilon^2 & \varepsilon^n \\
& 0 & \varepsilon^n
\end{bmatrix} \in \mathbb{R}^{6n\times 6n},
$$

(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 $\omega_\ell$ of the $\ell$-th mechanism on its diagonal, i.e. $E'_\ell = -\omega_\ell \cdot I$ 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 $|A_{pq}|$ of matrix $A_{pq}$, in the following we will have a closer look at the absolute value matrix $|A_{pq}|$ and the rotation matrix $T_{pq}$. Similar to (14) we find that

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

(23)

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

$$
|A| = \begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & c_s & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & c_s & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_s \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix},
$$

(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:

$$
A^|| = \begin{bmatrix}
A^||_1 \\
\vdots \\
A^||_n
\end{bmatrix} \in \mathbb{R}^{6n\times 9},
$$

(25)

where each sub-matrix $A^||_\ell \in \mathbb{R}^{6\times 9}$, with $\ell = 1, \ldots, n$, contains the local unrelaxed material parameters and the relaxation frequency $\omega_\ell$ of the $\ell$-th attenuation mechanism in the form:

$$
A^||_\ell = \omega_\ell \cdot \begin{pmatrix}
\frac{1}{(c_p\rho)} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}.
$$

(26)

To compute the rotation matrices we recall that the anelastic functions $\vartheta^\ell$ are tensors like the stresses and thus the rotation matrix $T_{pq}$ for the full anelastic system (12) has the form

$$
T_{pq} = \begin{bmatrix}
T^\ell & 0 & 0 \\
0 & T^\ell & 0 \\
0 & 0 & T^\ell
\end{bmatrix} \in \mathbb{R}^{n_v \times n_v},
$$

(27)
where \(T^4 \in \mathbb{R}^{6 \times 6}\) is the rotation matrix responsible for the stress tensor rotation as in the purely elastic part and is given as

\[
T^4 = \begin{pmatrix}
  n_x^2 & s^2_x & t_x^2 & 2n_y s_x & 2s_y t_x & 2n_z t_x \\
  n_y^2 & s^2_y & t_y^2 & 2n_z s_y & 2s_z t_y & 2n_y t_y \\
  n_z^2 & s^2_z & t_z^2 & 2n_y s_z & 2s_y t_z & 2n_x t_z \\
n_y n_z & s_y s_z & t_y t_z & n_y s_x + n_x s_y & s_y t_x + s_x t_y & n_y t_x + n_x t_y \\
n_z n_y & s_z s_y & t_z t_y & n_z s_x + n_x s_z & s_z t_x + s_x t_z & n_z t_x + n_x t_z \\
n_x n_z & s_x s_z & t_x t_z & n_x s_y + n_y s_x & s_z t_y + s_y t_z & n_x t_y + n_y t_z
\end{pmatrix},
\]

(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 2006a).

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

\[
T^w = \begin{pmatrix}
  n_x & s_x & t_x \\
  n_y & s_y & t_y \\
  n_z & s_z & t_z
\end{pmatrix}.
\]

(29)

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

\[
T_a = \begin{bmatrix}
  T^d & \cdots & 0 \\
  \vdots & \ddots & \vdots \\
  0 & \cdots & T^d
\end{bmatrix} \in \mathbb{R}^{6n \times 6n},
\]

(30)

where each of the \(n\) sub-matrices \(T^d\) is the tensor rotation matrix given in (28).

### 3 RECONSTRUCTION ALGORITHM

The main ingredient of the proposed arbitrary high order Finite Volume scheme is the new reconstruction algorithm proposed in (Dumbser & Käser 2006b) that makes use of techniques developed originally in the Discontinuous Galerkin (DG) framework. Whereas the method proposed in (Dumbser & Käser 2006b) 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 \(\Omega\) is discretised 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 2D and tetrahedrons in 3D. The union of all elements is called the triangulation or tetrahedrization of the domain, respectively,

\[
T_\Omega = \bigcup_{m=1}^E T^{(m)}.
\]

(31)

As usual for Finite Volume schemes, data is represented by the cell averages of the state vector \(U_p\) inside an element \(T^{(m)}\),

\[
\bar{u}_p^{(m)} = \frac{1}{|T^{(m)}|} \int_{T^{(m)}} U_p dV,
\]

(32)

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

In order to achieve high order of accuracy for a Finite Volume 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

\[
W_p^{(m)}(\xi, \eta, \zeta) = \tilde{w}^{(m)}_p \Psi_l(\xi, \eta, \zeta),
\]

(33)

where \(\xi, \eta\) and \(\zeta\) are the coordinates in a reference coordinate system, see Figure 1, where also the reference elements \(T_{\Xi}\) are defined. Throughout the whole paper we use classical tensor notation, which implies summation over each index appearing twice. Whereas the reconstructed degrees of freedom \(\tilde{w}^{(m)}_p\) are not space-dependent, the reconstruction basis functions \(\Psi_l\) 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 2D and 3D, respectively, depending on the order of the reconstruction. We use the hierarchical orthogonal reconstruction basis functions that are given e.g. in (Dubiner 1991; Cockburn, Karniadakis & Shu 2000) or in Appendix A for triangles in 2D and tetrahedrons in 3D. The transformation from the physical coordinate system \(x - y - z\) into the reference coordinate system \(\xi - \eta - \zeta\) is in three space dimensions defined by

\[
\begin{align*}
x &= X_1^{(m)} + \left( X_2^{(m)} - X_1^{(m)} \right) \xi + \left( X_3^{(m)} - X_1^{(m)} \right) \eta + \left( X_4^{(m)} - X_1^{(m)} \right) \zeta, \\
y &= Y_1^{(m)} + \left( Y_2^{(m)} - Y_1^{(m)} \right) \xi + \left( Y_3^{(m)} - Y_1^{(m)} \right) \eta + \left( Y_4^{(m)} - Y_1^{(m)} \right) \zeta, \\
z &= Z_1^{(m)} + \left( Z_2^{(m)} - Z_1^{(m)} \right) \xi + \left( Z_3^{(m)} - Z_1^{(m)} \right) \eta + \left( Z_4^{(m)} - Z_1^{(m)} \right) \zeta,
\end{align*}
\]
We note in particular that the transformed element of the following are in the following denoted as times the tetrahedron volume in 3D. For performing the reconstruction on element $T$ that contains a total number of transformation applies for from the local index this problem, whereas we use a hierarchical orthogonal transformation matrices as reported by Abgrall in (Abgrall 1994). Abgrall and Friedrich (Friedrich 1998) used barycentric coordinates in order to avoid boundaries are not considered here.

For triangles and tetrahedrons with straight edges, to which we restrict ourselves in this paper. General polyhedral elements or even curved... basis as commonly used in the Discontinuous Galerkin finite element framework.

$\begin{align*}
\vec{x} &= (x, y, z) = (\xi, \eta, \zeta)
\end{align*}$

$\begin{align*}
W^{(m)}_p \left( \xi \left( T^{(m)}, \vec{x} \right) \right) dV &= |T^{(k)}| \tilde{u}^{(k)}_p, \quad \forall T^{(k)} \in S^{(m)}.
\end{align*}$

$\begin{align*}
\int_{\tilde{T}^{(k)}} \Psi_t \left( \xi \right) d\xi d\eta d\zeta \tilde{u}^{(m)}_{pl} = |J| \tilde{T}^{(k)} \tilde{u}^{(k)}_p, \quad \forall \tilde{T}^{(k)} \in \tilde{S}^{(m)}.
\end{align*}$
Figure 2. Examples of an original stencil $S^{(m)}$ (a) and the corresponding transformed stencil $\tilde{S}^{(m)}$ (b) in 2D for the reconstruction of a polynomial of degree 3 with $n_e = 15$.

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

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 $\tilde{S}^{(m)}$. 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 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 $\Xi = (\tilde{\xi}, \tilde{\eta}, \tilde{\zeta})$. The mapping and its inverse are then denoted as

$$\vec{\xi} = \tilde{\xi}(\tilde{T}^{(k)}, \Xi), \quad \vec{\Xi} = \Xi(\tilde{T}^{(k)}, \vec{\xi}),$$

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

$$|J| |\tilde{J}| \int_{T_E} \Psi_l(\vec{\xi}(\tilde{T}^{(k)}, \Xi)) d\tilde{\xi} d\tilde{\eta} d\tilde{\zeta} \tilde{w}_{pl}^{(m)} = |J| |\tilde{J}| |T_E| \tilde{u}_{pl}^{(k)}, \quad \forall \tilde{T}^{(k)} \in \tilde{S}^{(m)},$$

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

$$\int_{T_E} \Psi_l(\vec{\xi}(\tilde{T}^{(k)}, \Xi)) d\tilde{\xi} d\tilde{\eta} d\tilde{\zeta} \tilde{w}_{pl}^{(m)} = |T_E| \tilde{u}_{pl}^{(k)}, \quad \forall \tilde{T}^{(k)} \in \tilde{S}^{(m)},$$

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

\[ A_{ij} \tilde{u}_{pi} = \bar{u}_{pj}, \]  

with

\[ A_{ij} = \frac{1}{|T_i|} \left( \int_{E_i} \Psi_i \left( \xi \left( \tilde{T}^{(kj)} \right) \right) d\xi d\eta d\zeta \right) \quad \text{and} \quad \bar{u}_{pj} = \tilde{u}_{p}^{(kj)}. \]  

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 \( A_{ij} \) 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 \( A_{ij} \) 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 2D and \( n_e = 2L \) in 3D. This means that we take between 50% and 100% 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 Figure 2 (a) and (b) in two dimensions and Figure 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 \( \bar{u}_{pq} = \tilde{u}_{p}^{(e1)} = \tilde{u}_{p}^{(m)} \), which is written in tensor notation

\[ C_l \hat{u}_{pl} = R_l \bar{u}_{p} \]  

The vectors \( C_l \) and \( R_l \) contain only zeros except of the entries \( C_1 = 1 \) and \( R_1 = 1 \). The least-squares solution of (42) with the constraint (44) coupled via a Lagrangian multiplier \( \lambda_p \) is obtained according to (Dumbser & Käser 2006b) as

\[ \begin{pmatrix} 2A_{jj}A_{jk} & -C_l \\ C_l \delta_{lk} & 0 \end{pmatrix} \begin{pmatrix} \hat{u}_{pk} \\ \lambda_p \end{pmatrix} = \begin{pmatrix} 2A_{jj} \bar{u}_{pj} \\ R_l \bar{u}_{pl} \end{pmatrix}. \]  

Here, \( \delta_{lk} \) 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 \( \Omega \) so that the unknown vector of the reconstructed degrees of freedom \( \hat{u}_{pl} \) 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 \( \bar{u}_{pl} \) of the stencil \( S^{(m)} \). We repeat that reconstruction is done component-wise for each variable \( p \) of the governing equations (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 DISCRETISATION OF THE ANELASTIC WAVE EQUATIONS

4.1 Semi-Discrete Finite Volume Scheme

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

\[ \int_{\partial T^{(m)}} \frac{\partial U}{\partial t} dV - \int_{\partial T^{(m)}} F_p^h (W_q^-, W_q^+) \bar{n} dS = \int_{\partial T^{(m)}} E_{pq} U_q dV. \]  

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 Finite Volume scheme, these values correspond to the cell averages in the element \( T^{(m)} \) and the neighbours, respectively. In the case of a higher order Finite Volume 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 \( \bar{n} = (n_x, n_y, n_z)^T \) 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 \( T^{(kj)} \):

\[ F_p^h (W_q^-, W_q^+) \bar{n} = \frac{1}{2} \left[ \bar{T}_{pq} \left( \hat{A}_{rs} + |\hat{A}_{rs}| \right) (\bar{T}_{rq})^{-1} W_q^- + \bar{T}_{pq} \left( \hat{A}_{rs} - |\hat{A}_{rs}| \right) (\bar{T}_{rq})^{-1} W_q^+ \right], \]  

where \( \hat{A} \) is the Jacobian matrix of the flux function and \( \bar{T} \) is the numerical flux. The exact Riemann solver is given by

\[ \bar{T}_{ij} = \left\{ \begin{array}{ll} T_{ij} & \text{if } i = j \\ -T_{ji} & \text{if } i \neq j \end{array} \right. \]  

where \( T_{ij} \) is the Jacobian matrix of the flux function.
where $W_q^- = \tilde{u}_q(l)^{(m)} \Psi(l)^{(m)}$ and $W_q^+ = \hat{w}_{ql}^{(k_j)} \Psi(l)^{(k_j)}$ are the boundary extrapolated values of the reconstructed numerical solution $W_q$ from element $(m)$ and the $j$-th 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 $A_{pq} = A_{pq}^{(m)} = A_{pq}(T^{(m)})$ as function of the element $T^{(m)}$. Inserting (47) into (46) and splitting the boundary integral into the contributions of each face $1 \leq j \leq N_E$ of the element $T^{(m)}$, we obtain

$$
\frac{\partial}{\partial t} \tilde{u}_q(l)^{(m)} \big|_{T^{(m)}} = + \frac{1}{2} \sum_{j=1}^{N_E} T_{pr} \left( \bar{A}_{x}^{(m)} \right) \left( \bar{T}_{sq} \right)^{-1} \hat{w}_{ql}^{(k_j)} \int_{\partial T^{(m)}} \Psi(l)^{(k_j)} dS = \frac{1}{2} \sum_{j=1}^{N_E} T_{pr} \left( \bar{A}_{x}^{(m)} \right) \left( \bar{T}_{sq} \right)^{-1} \hat{w}_{ql}^{(k_j)} \int_{\partial T^{(m)}} \Psi(l)^{(k_j)} dV.
$$

(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 Figure 1), the method can be implemented much more efficiently since all integrals can be precomputed 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 2D and 3D then reads as

$$
\frac{\partial}{\partial t} \bar{u}_p(l)^{(m)} \big|_{T^{(m)}} = + \frac{1}{2} \sum_{j=1}^{N_E} \bar{T}_{pr} \left( \bar{A}_{x}^{(m)} \right) \left( \bar{T}_{sq} \right)^{-1} |S_j| F_{i,-}^{-1} \hat{w}_{ql}^{(m)} + \frac{1}{2} \sum_{j=1}^{N_E} \bar{T}_{pr} \left( \bar{A}_{x}^{(m)} \right) \left( \bar{T}_{sq} \right)^{-1} |S_j| F_{i,+}^{-1,h} \hat{w}_{ql}^{(h)} = E_{pq}^{(m)} \hat{u}_q^{(m)} \big|_{T^{(m)}}^T (50)
$$

(49)

where $|S_j|$ denotes the surface of face $j$ in 3D and the edge length of edge $j$ in 2D. In (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 2006a; Dumbser & Käser 2006a; Käser, Dumbser, de la Puente & Igel 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 then be stored. 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 2D as well as in 3D. Then, the vector of volume coordinates $\tilde{\xi}$ is given on the faces via mapping functions from the face parameters $\chi$ and $\tau$, 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 face parameters $\chi$ and $\tau$ inside the element to the corresponding face parameters $\tilde{\chi}$ and $\tilde{\tau}$ in the neighbour face. Whereas in 2D this transformation is always $\tilde{\chi} = 1 - \chi$, in 3D 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

$$
F_{i,-}^{-j} = \int_{\partial(T_E)} \Psi(l) \left( \tilde{\xi}(j) \right) d\chi, \quad \forall 1 \leq j \leq 3,
$$

(50)

$$
F_{i,+}^{-i,h} = \int_{\partial(T_E)} \Psi(l) \left( \tilde{\xi}(j) \left( 1 - \chi \right) \right) d\chi, \quad \forall 1 \leq i \leq 3.
$$

(51)

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

$$
F_{i,-}^{-j} = \int_{\partial(T_E)} \Psi(l) \left( \tilde{\xi}(j) \left( \chi, \tau \right) \right) d\chi d\tau, \quad \forall 1 \leq j \leq 4,
$$

(52)

$$
F_{i,+}^{-i,h} = \int_{\partial(T_E)} \Psi(l) \left( \tilde{\xi}(j) \left( \tilde{\chi}(h), \tilde{\tau}(h) \right) \right) d\chi d\tau, \quad \forall 1 \leq i \leq 4, \quad \forall 1 \leq h \leq 3.
$$

(53)

The left state flux matrix (superscript ‘−’ $F_{i,-}^{-j}$) accounts for the contribution of the element $(m)$ itself to the fluxes over face $j$ and the right state flux matrix (superscript ‘+’ $F_{i,+}^{-i,h}$) accounts for the contribution of the element’s direct side neighbours $(k_j)$ to the fluxes over the face $j$. Index $1 \leq i \leq N_E$ indicates the local number of the common face as it is seen from neighbour $(k_j)$ and depends on the mesh generator. Index $1 \leq h \leq 3$ accounts for the three possible orientations of the face due to rotation and denotes the number of the local node
Table 1. Face definition on triangles and tetrahedrons

<table>
<thead>
<tr>
<th>Triangles (2D)</th>
<th>Tetrahedrons (3D)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Face Points</td>
<td>Face Points</td>
</tr>
<tr>
<td>1 1 2</td>
<td>1 1 3 2</td>
</tr>
<tr>
<td>2 2 3</td>
<td>2 1 2 4</td>
</tr>
<tr>
<td>3 3 1</td>
<td>3 1 4 3</td>
</tr>
<tr>
<td></td>
<td>4 2 3 4</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>$j$</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\xi^{(j)}(\chi)$</td>
<td>$1-\chi$</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>$\eta^{(j)}(\chi)$</td>
<td>$\chi$</td>
<td>$1-\chi$</td>
<td></td>
</tr>
</tbody>
</table>

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 $F^{i,h}_{l}$ 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; Titarev & Toro 2005) can be used for high order time integration of the Finite Volume method on unstructured meshes, called ADER-FV method in the following, for general linear hyperbolic systems. For linear systems, a particular simplification can be introduced: time-integration and flux computation can be exchanged, i.e. instead of solving the Riemann Problems for all spatial derivatives on the interface and doing then the Cauchy-Kovalewski procedure with the obtained derivatives, we can integrate the reconstructed solution in time separately in each element using the Cauchy-Kovalewski procedure on the reconstructed solution and then plug the time-integrated values on the boundaries into the numerical flux function, which then takes correctly into account the discontinuity at the interface. We emphasize that the pure application of the Cauchy-Kovalewski procedure requires the solution to be analytic, whereas the ADER approach uses the solution of Generalized Riemann Problems with piecewise polynomial initial data. This requires only that the solution is piecewise analytic on both side of the element interfaces. Note that the GRPs are always solved along the face-normal direction.

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

$$\frac{\partial U_p}{\partial t} + \frac{\partial U_p}{\partial \xi} + \frac{\partial U_p}{\partial \eta} + \frac{\partial U_p}{\partial \zeta} - \hat{E}_{pq} U_q = 0,$$

(54)

with

$$\hat{A}^{pq}_{\xi} = \hat{A}_{pq} \frac{\partial}{\partial x} + \hat{B}_{pq} \frac{\partial}{\partial y} + \hat{C}_{pq} \frac{\partial}{\partial z}$$

(55)

$$\hat{B}^{pq}_{\eta} = \hat{A}_{pq} \frac{\partial}{\partial x} + \hat{B}_{pq} \frac{\partial}{\partial y} + \hat{C}_{pq} \frac{\partial}{\partial z}$$

(56)

$$\hat{C}^{pq}_{\zeta} = \hat{A}_{pq} \frac{\partial}{\partial x} + \hat{B}_{pq} \frac{\partial}{\partial y} + \hat{C}_{pq} \frac{\partial}{\partial z}.$$  

(57)

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

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

(58)

which can be proven by complete induction.

Table 3. 3D volume coordinates $\xi^{(j)}$ in function of the face parameters $\chi$ and $\tau$

<table>
<thead>
<tr>
<th>$j$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\xi^{(j)}(\chi, \tau)$</td>
<td>$\tau$</td>
<td>$\chi$</td>
<td>$0$</td>
<td>$1-\chi-\tau$</td>
</tr>
<tr>
<td>$\eta^{(j)}(\chi, \tau)$</td>
<td>$\chi$</td>
<td>$\tau$</td>
<td>$\chi$</td>
<td>$\tau$</td>
</tr>
<tr>
<td>$\zeta^{(j)}(\chi, \tau)$</td>
<td>$0$</td>
<td>$\tau$</td>
<td>$\chi$</td>
<td>$\tau$</td>
</tr>
</tbody>
</table>
We now introduce the approximation (33) and obtain

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

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

\[ W_p(\xi, \eta, \zeta, t^n) = \sum_{k=0}^{M} \frac{(t - t^n)^k}{k!} \left( A^x_{qy} \frac{\partial}{\partial \xi} + B^y_{pq} \frac{\partial}{\partial \eta} + C^z_{pq} \frac{\partial}{\partial \zeta} - E_{pq} \right)^k W_p(\xi, \eta, \zeta, t^n). \]  

We now introduce the approximation (33) and obtain

\[ W_p(\xi, \eta, \zeta, t^n) = \sum_{k=0}^{M} \frac{(t - t^n)^k}{k!} \left( A^x_{qy} \frac{\partial}{\partial \xi} + B^y_{pq} \frac{\partial}{\partial \eta} + C^z_{pq} \frac{\partial}{\partial \zeta} - E_{pq} \right)^k \Psi_l(\xi, \eta, \zeta, t^n). \]  

This approximation can now be projected onto the reconstruction basis functions \( \Psi_l \) 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

\[ \bar{w}_{pl}(t) = \left\langle \Psi_n, \sum_{k=0}^{M} \frac{(t - t^n)^k}{k!} \left( A^x_{qy} \frac{\partial}{\partial \xi} + B^y_{pq} \frac{\partial}{\partial \eta} + C^z_{pq} \frac{\partial}{\partial \zeta} - E_{pq} \right)^k \Psi_l \right\rangle \bar{w}_{qm}(t^n), \]  

where \( \langle \ldots, \ldots \rangle \) 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

\[ \int_{t^n}^{t^n + \Delta t} \bar{w}_{pl}(t) dt = \left\langle \Psi_n, \sum_{k=0}^{M} \frac{(t - t^n)^{k+1}}{(k+1)!} \left( A^x_{qy} \frac{\partial}{\partial \xi} + B^y_{pq} \frac{\partial}{\partial \eta} + C^z_{pq} \frac{\partial}{\partial \zeta} - E_{pq} \right)^k \Psi_l \right\rangle \bar{w}_{qm}(t^n). \]  

With the definition

\[ I_{plqm}(\Delta t) = \left\langle \Psi_n, \sum_{k=0}^{M} \frac{(t - t^n)^{k+1}}{(k+1)!} \left( A^x_{qy} \frac{\partial}{\partial \xi} + B^y_{pq} \frac{\partial}{\partial \eta} + C^z_{pq} \frac{\partial}{\partial \zeta} - E_{pq} \right)^k \Psi_l \right\rangle, \]  

equation (63) becomes simply

\[ \int_{t^n}^{t^n + \Delta t} \bar{w}_{pl}(t) dt = I_{plqm}(\Delta t) \bar{w}_{qm}(t^n). \]  

For efficient algorithms to do the Cauchy-Kovalewski procedure, we refer to (Käser & Dumbser 2006a; Dumbser & Käser 2006a; Käser, Dumbser, de la Puente & Igel 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:

\[ \left[ \bar{w}_{pl}^{(m)}(t^{n+1}) - \bar{w}_{pl}^{(m)}(t^n) \right] \left[ t^{(m)} \right] = \]  

\[ + \frac{1}{2} \sum_{j=1}^{N_{E}} \mathcal{T}_{ij} \left[ \bar{A}_{ij}^{(m)} \right] \left( \bar{T}_{ij} \right)^{-1} |S_j| F_{i-j}^{t,\Delta t} \cdot I_{q_{ij}n}^{\Delta t} \cdot \bar{w}_{ij}^{(m)}(t^n) \]  

\[ + \frac{1}{2} \sum_{j=1}^{N_{E}} \mathcal{T}_{ij} \left[ \bar{A}_{ij}^{(m)} + \bar{A}_{ij}^{(m)} \right] \left( \bar{T}_{ij} \right)^{-1} |S_j| F_{i-j}^{t,\Delta t} \cdot I_{q_{ij}n}^{\Delta t} \cdot \bar{w}_{ij}^{(m)}(t^n) = \]  

\[ E_{pq}^{(m)} \cdot I_{q_{ij}n}^{\Delta t} \cdot \bar{w}_{ij}^{(m)}(t^n) \left[ t^{(m)} \right]. \]  

From the structure of eqn. (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 \( \bar{w}_{ij}^{(m)}(t^n) \) 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 Finite Volume 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, Dumbser, de la Puente & Igel 2006), except of the following two differences: First, for Finite Volume schemes only the cell averages $\overline{u}_n$ have to be evolved in time, whereas for DG schemes all degrees of freedom $\hat{u}_m$ must be updated. Second, the fluxes of ADER-DG schemes are computed directly with the degrees of freedom $\hat{u}_m$, whereas the fluxes for ADER-FV schemes are computed using the reconstructed degrees of freedom $\hat{u}_{pl}$, which are obtained from the cell averages $\overline{u}_p$ 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 Finite Volume schemes in contrast to Discontinuous Galerkin methods is the generation of appropriate reconstruction stencils at the boundary of the computational domain. In this article, we choose one-sided stencils, i.e. 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 $n_n$ 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 traveling 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, i.e. 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:

$$F_p^{AbsorbBC} = \frac{1}{2} \mathcal{F}_{pq} \left( \hat{A}^{(m)}_{qr} + \hat{A}^{(m)}_{rqr} \right) \left( \mathcal{T}_{rs} \right)^{-1} \hat{w}_{pl}^{(m)} \psi_l^{(m)}. \tag{67}$$

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, i.e. 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 (Bérenger1994) and applied in (Collino & Tsogka 2001; 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 retrace 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 Finite Element methods such as the Spectral Element method (SEM), we have no direct control on the values at the boundaries within the Finite Volume 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 $\hat{U}_p$, 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,

$$F_p^{FreeBC} = \frac{1}{2} \mathcal{F}_{pq} \left( \hat{A}^{(m)}_{qr} + \hat{A}^{(m)}_{rqr} \right) \left( \mathcal{T}_{rs} \right)^{-1} \hat{w}_{pl}^{(m)} \psi_l^{(m)} + \frac{1}{2} \mathcal{F}_{pq} \left( \hat{A}^{(m)}_{qr} - \hat{A}^{(m)}_{rqr} \right) \Gamma_{rs} \left( \mathcal{T}_{rs} \right)^{-1} \hat{u}_{pl}^{(m)} \psi_l^{(m)}, \tag{68}$$

where the matrix $\Gamma_{rs} = \text{diag}(-1, 1, 1, -1, 1, -1, 1, 1, 0, ..., 0)$ accounts for the mirroring of normal and shear stresses with respect to the face-normal direction. The visco-elasticity memory variables do not enter the flux. We remark, that the solution of the inverse Riemann Problem is not equivalent to the FD approach of adding fictitious ghost points, but comes out naturally from the FV framework and provides the
Figure 4. Sequence of discretisations of the computational domain $\Omega$ via regularly refined tetrahedral meshes used for the numerical convergence analysis.

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 Finite Difference 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 order to sixth order ADER-FV schemes, which are denoted by ADER-FV $O^2$ to ADER-FV $O^6$ respectively. Note, that the same order for time and space accuracy is automatically obtained.

To determine the convergence orders we solve the three-dimensional seismic wave equations with viscoelastic attenuation in (12) on the unit-cube as sketched in Figure 4, i.e. 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 $\lambda = 2, \mu = 1, \rho = 1, Q_P = 20, Q_S = 10$, throughout the computational domain $\Omega$. 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, Dumbser, de la Puente & Igel 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, e.g. from (Stein & Wysession 2003), that a space-time harmonic solution to this problem can be found under the form

$$ U_p(x, y, z, t) = U_0^p \cdot e^{(\omega t - k_x x - k_y y - k_z z)}, \quad p = 1, \ldots, n_v $$

where $U_0^p$ is the initial amplitude vector, $\omega$ the wave frequencies to determine, and $k = (k_x, k_y, k_z)^T = (\pi, \pi, \pi)^T$. (70)

is the wave number 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 $\omega$:

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

$$ (A_{pq} k_x + B_{pq} k_y + C_{pq} k_z - i \cdot E_{pq} ) \cdot U_q = \omega \cdot U_q^p, \quad p, q = 1, \ldots, n_v, $$

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 $\omega_p$. Recalling, e.g. from (Toro 1999), that each solution of the linear hyperbolic system (13) is given by a linear combination of the right
sequence is obtained by dividing the computational domain $\Omega = 0$ as shown in Figure 4. This way, the refinement is controlled by changing the number of sub-cubes in each space dimension.

The numerical analysis to determine the convergence orders is performed on a sequence of tetrahedral meshes as shown in Figure 4. The mesh is generated by subdividing the space diagonal $\mathbf{r} = (1, 1, 1)^T$ of the domain $\Omega$ at $t = 0$ is given by (73) using the combination of only two right eigenvectors $(R_{p2}, ..., R_{p6})$ with the coefficients $\nu_2 = \nu_9 = 1$ and zero otherwise. The total simulation time $T$ is set to $T = 0.1s$. For a thorough investigation of the linear stability properties of ADER-FV schemes via differential approximation and via von Neumann stability analysis see (Dumbser & Käser 2006a). Therefore, the initial condition at $t = 0$ is given by (73) using the combination of only two right eigenvectors $(R_{p2}, ..., R_{p6})$ with the coefficients $\nu_2 = \nu_9 = 1$ and zero otherwise. The total simulation time $T$ is set to $T = 0.1s$. For a thorough investigation of the linear stability properties of ADER-FV schemes via differential approximation and via von Neumann stability analysis see (Dumbser, Schwartzkopff & Munz 2006).

The numerical analysis to determine the convergence orders is performed on a sequence of tetrahedral meshes as shown in Figure 4. The mesh sequence is obtained by dividing the computational domain $\Omega$ into a number of sub-cubes, which are then subdivided into five tetrahedrons as shown in Figure 4. This way, the refinement is controlled by changing the number of sub-cubes in each space dimension.

We can arbitrarily pick one of the variables of the system of the seismic wave equations (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 $L^\infty$-norm and the continuous $L^2$-norm

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

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 $L^\infty$-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 $L^\infty$ and $L^2$ errors with the corresponding convergence orders $O_{L^\infty}$ and $O_{L^2}$ 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 $T = 0.1s$ on one Pentium Xeon 3.6 GHz processor with 4GB of RAM.

### Table 5. Convergence rates of velocity component $v$ for ADER-FV $O^2$ to ADER-FV $O^6$ schemes. Viscoelastic attenuation is modeled using five mechanisms.

<table>
<thead>
<tr>
<th>$h$</th>
<th>$L^\infty$</th>
<th>$O_{L^\infty}$</th>
<th>$L^2$</th>
<th>$O_{L^2}$</th>
<th>$I$</th>
<th>CPU [s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.16 · 10^{-1}</td>
<td>1.5622 · 10^{-2}</td>
<td>–</td>
<td>1.7101 · 10^{-2}</td>
<td>–</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>1.08 · 10^{-1}</td>
<td>3.7724 · 10^{-3}</td>
<td>2.1</td>
<td>3.9186 · 10^{-3}</td>
<td>2.1</td>
<td>8</td>
<td>21</td>
</tr>
<tr>
<td>7.21 · 10^{-2}</td>
<td>1.6544 · 10^{-3}</td>
<td>2.0</td>
<td>1.6899 · 10^{-3}</td>
<td>2.1</td>
<td>12</td>
<td>103</td>
</tr>
<tr>
<td>5.41 · 10^{-2}</td>
<td>9.8176 · 10^{-4}</td>
<td>1.8</td>
<td>9.8126 · 10^{-4}</td>
<td>1.9</td>
<td>16</td>
<td>324</td>
</tr>
<tr>
<td>2.16 · 10^{-1}</td>
<td>2.5053 · 10^{-2}</td>
<td>–</td>
<td>2.4776 · 10^{-2}</td>
<td>–</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>1.08 · 10^{-1}</td>
<td>3.9698 · 10^{-3}</td>
<td>2.7</td>
<td>3.1450 · 10^{-3}</td>
<td>3.0</td>
<td>8</td>
<td>31</td>
</tr>
<tr>
<td>7.21 · 10^{-2}</td>
<td>1.2139 · 10^{-3}</td>
<td>2.9</td>
<td>9.3247 · 10^{-4}</td>
<td>3.0</td>
<td>12</td>
<td>154</td>
</tr>
<tr>
<td>5.41 · 10^{-2}</td>
<td>5.0839 · 10^{-4}</td>
<td>3.0</td>
<td>3.9337 · 10^{-4}</td>
<td>3.0</td>
<td>16</td>
<td>488</td>
</tr>
<tr>
<td>2.16 · 10^{-1}</td>
<td>5.0283 · 10^{-3}</td>
<td>–</td>
<td>5.0372 · 10^{-3}</td>
<td>–</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>1.44 · 10^{-1}</td>
<td>9.8627 · 10^{-4}</td>
<td>4.0</td>
<td>1.0197 · 10^{-3}</td>
<td>3.9</td>
<td>6</td>
<td>20</td>
</tr>
<tr>
<td>1.08 · 10^{-1}</td>
<td>3.5887 · 10^{-4}</td>
<td>3.5</td>
<td>3.3024 · 10^{-4}</td>
<td>3.9</td>
<td>8</td>
<td>61</td>
</tr>
<tr>
<td>7.21 · 10^{-2}</td>
<td>7.1172 · 10^{-5}</td>
<td>4.0</td>
<td>6.6368 · 10^{-5}</td>
<td>4.0</td>
<td>12</td>
<td>307</td>
</tr>
<tr>
<td>2.16 · 10^{-1}</td>
<td>3.4920 · 10^{-3}</td>
<td>–</td>
<td>3.4793 · 10^{-3}</td>
<td>–</td>
<td>4</td>
<td>9</td>
</tr>
<tr>
<td>1.44 · 10^{-1}</td>
<td>5.0073 · 10^{-4}</td>
<td>4.8</td>
<td>4.6677 · 10^{-4}</td>
<td>5.0</td>
<td>6</td>
<td>46</td>
</tr>
<tr>
<td>1.08 · 10^{-1}</td>
<td>1.1954 · 10^{-4}</td>
<td>5.0</td>
<td>1.1227 · 10^{-4}</td>
<td>5.0</td>
<td>8</td>
<td>144</td>
</tr>
<tr>
<td>7.21 · 10^{-2}</td>
<td>1.5758 · 10^{-5}</td>
<td>5.0</td>
<td>1.4849 · 10^{-5}</td>
<td>5.0</td>
<td>12</td>
<td>727</td>
</tr>
<tr>
<td>2.16 · 10^{-1}</td>
<td>1.0705 · 10^{-3}</td>
<td>–</td>
<td>7.3333 · 10^{-4}</td>
<td>–</td>
<td>4</td>
<td>21</td>
</tr>
<tr>
<td>1.44 · 10^{-1}</td>
<td>1.0996 · 10^{-4}</td>
<td>5.6</td>
<td>6.6501 · 10^{-5}</td>
<td>5.9</td>
<td>6</td>
<td>106</td>
</tr>
<tr>
<td>1.08 · 10^{-1}</td>
<td>2.2398 · 10^{-5}</td>
<td>5.5</td>
<td>1.2250 · 10^{-5}</td>
<td>5.9</td>
<td>8</td>
<td>319</td>
</tr>
<tr>
<td>7.21 · 10^{-2}</td>
<td>2.0320 · 10^{-6}</td>
<td>5.9</td>
<td>1.0900 · 10^{-6}</td>
<td>6.0</td>
<td>12</td>
<td>1616</td>
</tr>
</tbody>
</table>

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, Dumbser, de la Puente & Igel 2006). First, we recall the fully discrete versions of both schemes in equations (75) and (76), respectively.
evolved in time, the ADER-FV scheme only advances the cell averages

The ADER Finite Volume scheme for the anelastic wave equations developed in this article reads as

\[ T^{(m)} \left[ \hat{u}_p^{(m)} (t^{n+1}) - \hat{u}_p^{(m)} (t^n) \right] + \]

\[ + \frac{1}{2} \sum_{j=1}^{N_E} T_{pr} \left( A_{rs}^{(m)} + \frac{1}{2} A_{rs}^{(m)} \right) \left( T_{eq} \right)^{-1} |S_j| F_{t}^{j,h} \cdot I_{q_{1nm}}(\Delta t) \cdot \hat{w}^{(k)}_{nm} (t^n) - \]

\[ = \left[ T^{(m)} \right] T_{pq} F_{pq}^{(m)} \cdot I_{q_{1nm}}(\Delta t) \cdot \hat{u}_{nm}^{(m)} (t^n). \] (75)

We recall that the ADER Discontinuous Galerkin method for the anelastic wave equations presented in (Käser, Dumbser, de la Puente & Igel 2006) has the following form:

\[ |J| M_K \left[ \hat{u}_p^{(m)} (t^{n+1}) - \hat{u}_p^{(m)} (t^n) \right] + \]

\[ + \frac{1}{2} \sum_{j=1}^{N_E} T_{pr} \left( A_{rs}^{(m)} + \frac{1}{2} A_{rs}^{(m)} \right) \left( T_{eq} \right)^{-1} |S_j| F_{t}^{j,h} \cdot I_{q_{1nm}}(\Delta t) \cdot \hat{w}^{(k)}_{nm} (t^n) - \]

\[ = \left[ T^{(m)} \right] T_{pq} F_{pq}^{(m)} \cdot I_{q_{1nm}}(\Delta t) \cdot \hat{u}_{nm}^{(m)} (t^n). \] (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, i.e., they directly integrate the governing equation (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 \( u_{1nm} \), whereas in the ADER-DG scheme (76) this procedure can be directly applied to the degrees of freedom \( u_{1nm} \) already given by the spatial DG discretisation. Except of this difference, the method for carrying out the Cauchy-Kovalewski procedure is the same in both schemes, see the tensor \( I_{q_{1nm}}(\Delta t) \) 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 \( \hat{u}_p \) (matrix of size \( n_v \times L \)) are evolved in time, the ADER-FV scheme only advances the cell averages \( \hat{u}_p \) (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 \( L = \frac{1}{2}(M + 1)(M + 2)(M + 3) \) 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 \( \hat{W} \) of the Finite Volume scheme in terms of the same basis functions as for the Discontinuous Galerkin 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. (Cockburn, Karniadakis & Shu 2000; Käser & Dumbser 2006a; Dumbser & Käser 2006a). We see that the stiffness matrices \( K_{\xi}^{\xi}, K_{\eta}^{\eta} \) and \( K_{\nu}^{\nu} \) 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-FV time step limit decreases proportional to \( \frac{1}{\sqrt{L}} \), where \( M \) is the degree of the DG basis polynomials. For the Finite Volume 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, Schwarczkopf & Munz 2006). This means that for example a fifth order ADER-FV scheme has a time step limit that is eleven 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 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 $u_{pl}$ 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 $\hat{u}_{pl}$ only at the direct neighbours of a subdomain boundary, exactly as in the ADER-DG case. Therefore, the communication overhead is higher for ADER-FV, but since many of the reconstruction stencils are overlapping, the overhead is not larger than a factor of two. Profiling has shown that both parallel codes scale quite well with the number of processors. Typical runs of the parallel ADER-FV and ADER-DG codes on the SGI Altix cluster of the LRZ supercomputing center in München and on the NEC Linux cluster of the HLRS supercomputing center 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 & Kaiser 2006b) and outlined also in this article is still quite easy to implement due to the use of hierarchical orthogonal reconstruction basis functions $\Psi$ 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 3D, the inverse reconstruction matrices have the size $L × 2L$. 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, and once with an ADER-FV scheme and once with an ADER-DG. On this behalf we run the same convergence test as shown in the previous Section 6 on the same mesh once with an ADER-FV scheme and once with an ADER-DG 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, and once with an ADER-FV scheme and once with an ADER-DG scheme.

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, and once with an ADER-FV scheme and once with an ADER-DG 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, and once with an ADER-FV scheme and once with an ADER-DG scheme.

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 6. However, especially having a look at the error norms presented in Tables 5 and 6 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, Dumbser, de la Puente & Igel 2006) and Table 6, we note that on the same mesh the fourth order Finite Volume schemes are about ten 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 superior to ADER-FV schemes comparing the CPU time needed by both methods at the same level of accuracy. 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 more or less given already at the beginning of the simulation due to 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. In the following section, we present the results obtained with ADER-FV for several standard benchmark problems.

8 APPLICATION EXAMPLES

We apply the proposed ADER-FV method on well-defined two- and three-dimensional test problems for which also analytic reference solutions are available. The 2D benchmark is the well-known Lamb’s problem (Lamb 1904) in the same setup as given in (Komatitsch & Vilotte 1998) and (Käser & Dumbser 2006a) to verify the accuracy of the scheme at the free-surface boundary. The two 3D benchmark problems LOH.1 and LOH.3 were published in the final report of the LIFELINES PROGRAM TASK 1A02 (Day, Bielak, Dreger, Graves, Larsen, Olsen & Pitarka 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 three-dimensional 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 in (Käser & Dumbser 2006a; Dumbser & Käser 2006a; Käser, Dumbser, de la Puente & Igel 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, Bielak, Dreger, Graves, Larsen, Olsen & Pitarka 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

The first three codes use Finite Differences 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 Finite Elements. The quasi-analytic solution is a frequency-wavenumber solution obtained by a modification of the method presented in (Luco & Apsel 1983; 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. 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 and H (1994) to compute the exact two-dimensional 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 and Vilotte (1998), who solved this problem using the Spectral Element method, see e.g. (Komatitsch & Tromp 1999; Komatitsch & Tromp 2002). Furthermore, this problem was solved in (Käser & Dumbser 2006a) on very coarse triangular meshes using a tenth order ADER-DG scheme.

We use a homogeneous elastic medium with a P-wave velocity of \(c_p = 3200 \text{ m s}^{-1}\), an S-wave velocity of \(c_s = 1847.5 \text{ m s}^{-1}\) and a mass density of \(\rho = 2200 \text{ kg m}^{-3}\). 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 \(\bar{x}_s = (1720.00, 2363.28)^T\). 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 194130 triangles, which is about 57 times more than the mesh used in (Käser & Dumbser 2006a) for the tenth 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

\[
S^T(t) = a_1 (0.5 + a_2 (t - t_D)^2) e^{a_2 (t - t_D)^2} ,
\]

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

The wave propagation is simulated until time \(T_{end} = 1.3 \text{ s}\) 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 \(\text{CFL} = 0.5\) is used. In order to reach the final simulation time \(T_{end} = 1.3\) we need 3233 time steps. In Figure 5 we present the snapshots of the absolute value of the velocity vector of the seismic wave field at \(t = 0.48 \text{ s}\). In Figure 6 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%. 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 tenth order ADER-DG scheme presented in (Käser & Dumbser 2006a) was able to produce errors of less than 1% 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.

8.2 Layer Over Halfspace Test Cases LOH.1 and LOH.3 in Three Space Dimensions

The setup of the test problems LOH.1 and LOH.3 (Layer Over Halfspace) is shown in Figure 7(a), where for clarity only one of four symmetrical quarters of the complete computational domain \(\Omega = [-15000m, 15000m] \times [-15000m, 15000m] \times [0m, 17000m]\) is plotted.
The material parameters of the thin layer (Medium 1) with thickness 1000m over the halfspace (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 $M_{xy} = M_{yx} = M_0 = 10^{18} \text{Nm}$. The location of the point source is $(x_s, y_s, z_s) = (0m, 0m, 2000m)$, i.e. in the center of the $xy$ plane of the domain $\Omega$ in 2000m depth. The moment-rate time history is given through the source time function

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

(78)

where the smoothness parameter $T$, controlling the frequency content and amplitude of the source time function, is set to $T = 0.1s$. We remark, that details of the discretisation of external source terms in the framework of ADER-DG methods are outlined in previous work (Käser & Dumbser 2006a). The same methodology applies also to ADER-FV schemes.

The signals are recorded up to a simulation time of 9s by 10 receivers on the free surface as indicated in Figure 7(a). The receiver locations are $(x_i, y_i, z_i) = (i \cdot 600m, i \cdot 800m, 0m)$, for $i = 1, \ldots, 10$. The computational domain $\Omega$ is discretised by an unstructured, tetrahedral mesh as shown in Figure 7(b) using 2100114 elements. For comparison, the mesh used for the ADER-DG computations shown in (Dumbser & Käser 2006a) and (Käser, Dumbser, de la Puente & Igel 2006) contained only 249338 elements. The mesh is generated in a problem-adapted manner. To this end, in the zone of interest the waves traveling from the source to the receivers pass through tetrahedral elements with an average edge length of 100m. This mesh spacing was also required in the setup of the benchmark for Finite Difference schemes. In other zones the mesh is coarsened up to average edge lengths of 1000m 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 Medium 2 as the faces of the tetrahedral elements are aligned with the material interface as shown in Figure 7(a) and (b).
Figure 6. Seismograms of the normal and tangential velocity components w.r.t. the surface at the two receivers 1 and 2 for Lamb’s problem.
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 \( f_r = 2.5 \text{Hz} \). For the LOH.1 test case \( \lambda \) and \( \mu \) are the unrelaxed material parameters and \( Q_p \) and \( Q_s \) are set to infinity.

<table>
<thead>
<tr>
<th>Medium</th>
<th>( c_p(f_r) ) [m/s]</th>
<th>( c_s(f_r) ) [m/s]</th>
<th>( \rho ) [kg/m(^3)]</th>
<th>( \lambda(f_r) ) [GPa]</th>
<th>( \mu(f_r) ) [GPa]</th>
<th>( Q_p )</th>
<th>( Q_s )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Medium 1</td>
<td>4000</td>
<td>2000</td>
<td>2600</td>
<td>20.8</td>
<td>10.4</td>
<td>120</td>
<td>40</td>
</tr>
<tr>
<td>Medium 2</td>
<td>6000</td>
<td>3464</td>
<td>2700</td>
<td>32.4</td>
<td>32.4</td>
<td>155.9</td>
<td>69.3</td>
</tr>
</tbody>
</table>

Figure 7. (a) One of four symmetric quarters is shown for the LOH test cases, where a layer of 1000m (Medium 1) is lying on top of another layer (Medium 2). The source is a point dislocation at 2000m depth represented by a moment tensor with the only non-zero components \( M_{xy} = M_{yx} \). (b) Cut into the discretisation of the LOH.1 and LOH.3 model to visualize the problem-adapted tetrahedral mesh, which is refined under the receiver line.

In the following, we present the comparison of our results obtained with a fourth order ADER Finite Volume 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 2006a) and (Käser, Dumbser, de la Puente & Igel 2006) with a fourth order ADER Discontinuous Galerkin method are shown. Analogous to the LOH.1 and LOH.3 test cases in the LIFELINES PROGRAM TASK 1.4.02 the visual comparisons in Figures 8 and 9 show the radial, transversal and vertical components of the seismic velocity field recorded at receiver 10 at \((x_{10}, y_{10}, z_{10}) = (6000m, 8000m, 0m)\). Additionally, each plot gives the relative seismogram misfit

\[
E = \sum_{j=1}^{n_t} (s_j - s_{ja})^2 / \sum_{j=1}^{n_t} (s_{ja})^2 ,
\]

(79)

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 \( s_{ja} \) 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, Bielak, Dreger, Graves, Larsen, Olsen & Pitarka 2003).

Whereas all solutions agree quite well for the LOH.1 test case, see Figure 8, the four reference solutions shown in Figure 9(a)-(d) remarkably differ from each other due to the different ways of incorporating viscoelastic attenuation. Amplitude errors (e.g. for CMUN) and phase errors (e.g. for UCSB) are quite noticeable. In addition, the results of UCBL, WCC2 and CMUN produce strong, unphysical oscillations in the transverse component.

The results with the fourth-order ADER-FV scheme in Figures 8(f) and 9(f) using 3 attenuation mechanisms for the LOH.3 case clearly match the analytic solution much better than the Finite Difference schemes UCBL, UCSB and WCC using the same maximal mesh resolution of 100m 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 Finite Volume approach is more appropriate to discretise 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...
Figure 8. 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 O4 and (f) ADER-FV O4. The relative seismogram misfit $E$ from equation (79) is given for each trace.
Figure 9. 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-DGO4 with 3 attenuation mechanisms and (f) ADER-FVO4 with 3 attenuation mechanisms. The relative seismogram misfit $E$ from equation (79) is given for each trace.
(LOH.1 and LOH.3) comparable to the ADER-DG scheme shown in Figures 8(c) and 9(c). We note, however, that the mesh used for the ADER-FV computation presented in 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 2006a) and (Käser, Dumbser, de la Puente & Igel 2006).

9 CONCLUSION

We have presented a new class of Finite Volume 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 Finite Volume 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 that in both the LOH.1 and the LOH.3 test case the mesh spacing used was similar to the one used by the Finite Difference 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 Finite Volume 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 discretised 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 three-dimensional tetrahedral meshes. Due to the lower CPU times w.r.t. 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.

ACKNOWLEDGMENT

The presented work was supported by the Deutsche Forschungsgemeinschaft through the Emmy Noether-Programm (KA 2281/1-1), the DFG Forschungsstipendium (DU 1107/1-1), 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.

REFERENCES


APPENDIX A: RECONSTRUCTION BASIS FUNCTIONS

We use orthogonal hierarchical basis functions as given in (Cockburn, Karniadakis & Shu 2000) and (Karniadakis & Sherwin 1999), which extend the original ideas of Dubiner (1991). The basis functions are given in terms of the Jacobi polynomials \( P_{\alpha,\beta}^n(x) \), which are solutions of the Jacobi differential equation:

\[
(1-x^2)y' + [\beta - \alpha - (\alpha + \beta + 2)x]y' + n(n + \alpha + \beta + 1)y = 0.
\]  

They are given on the interval \([-1; 1]\) by

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

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

\[
\Theta^p_i(x) = P_{i}^{0,0}(x), \quad \Theta^q_j(x) = \left(\frac{1-x}{2}\right)^j P_{j}^{2i+1,0}(x), \quad \Theta^r_{ijk}(x) = \left(\frac{1-x}{2}\right)^{i+j} P_{k}^{2i+2j+2,0}(x).
\]  

The sets of reconstruction basis functions \(\Psi_k\) 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

\[
T_E = \{(\xi, \eta) \in \mathbb{R}^2 | 0 \leq \xi \leq 1 \land 0 \leq \eta \leq 1 - \xi\}.
\]  

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

\[
\Psi_k(p,q)(\xi, \eta) = \Theta^p_i(\delta) \cdot \Theta^q_j(2\eta - 1)
\]  

with

\[
\delta = \frac{2\xi}{1-\eta} - 1.
\]  

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

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

A2 Three Space Dimensions

For tetrahedrons the reference element \(T_E\) is defined as

\[
T_E = \{(\xi, \eta, \zeta) \in \mathbb{R}^3 | 0 \leq \xi \leq 1 \land 0 \leq \eta \leq 1 - \xi \land 0 \leq \zeta \leq 1 - \xi - \eta\}.
\]  

Furthermore, the edge parameters \(\chi\) and \(\tau\) that parametrize the four triangular faces are defined in the reference triangle (A4). The basis functions \(\Psi_k(\xi, \eta, \zeta)\) are defined on this reference element as the following product of the primal functions:

\[
\Psi_k(p,q,r)(\xi, \eta, \zeta) = \Theta^p_i(\alpha) \cdot \Theta^q_j(\beta) \cdot \Theta^r_{ijk}(\gamma)
\]  

with

\[
\begin{align*}
\alpha &= \frac{\eta - 1 + \zeta + 2\xi}{1 - \eta - \zeta}, \\
\beta &= \frac{2\eta - 1 + \zeta}{1 - \zeta}, \\
\gamma &= -1 + 2\zeta.
\end{align*}
\]
The mono-index \( k = k(p, q, r) \) is again a function of the index triple \((p, q, r)\). The three-dimensional reconstruction basis functions up to degree two (for a third order FV scheme) are:

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

\(A11\)

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

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

\[
\begin{align*}
F_{i}^{-1} &= \begin{pmatrix} 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & -1 \end{pmatrix} \\
F_{i}^{-2} &= \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \\
F_{i}^{-3} &= \begin{pmatrix} 1 & -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \end{pmatrix}
\end{align*}
\]

\(B1\)

\[
\begin{align*}
F_{i}^{+1,h} &= \begin{pmatrix} 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & -1 \end{pmatrix} \\
F_{i}^{+2,h} &= \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \\
F_{i}^{+3,h} &= \begin{pmatrix} 1 & -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \end{pmatrix}
\end{align*}
\]

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

\[
\begin{align*}
F_{i}^{-1} &= \begin{pmatrix} 1 & 0 & 0 & -\frac{1}{2} & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} \end{pmatrix} \\
F_{i}^{-2} &= \begin{pmatrix} 1 & -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \end{pmatrix} \\
F_{i}^{-3} &= \begin{pmatrix} 1 & -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \end{pmatrix} \\
F_{i}^{-4} &= \begin{pmatrix} 1 & -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \end{pmatrix}
\end{align*}
\]

\(B2\)

\[
\begin{align*}
F_{i}^{+1,1} &= \begin{pmatrix} 1 & 0 & 0 & -\frac{1}{2} & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} \end{pmatrix} \\
F_{i}^{+1,2} &= \begin{pmatrix} 1 & -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \end{pmatrix} \\
F_{i}^{+1,3} &= \begin{pmatrix} 1 & -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \end{pmatrix} \\
F_{i}^{+1,4} &= \begin{pmatrix} 1 & -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \end{pmatrix}
\end{align*}
\]

\(B2\)
LIST OF FIGURES

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 $(0,0,0)$, $(1,0,0)$, $(0,1,0)$, $(0,0,1)$.

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

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

4 Sequence of discretisations of the computational domain $\Omega$ via regularly refined tetrahedral meshes used for the numerical convergence analysis.

5 Absolute value of the velocity vector ($\sqrt{u^2 + v^2}$) at $t = 0.48s$ for Lamb’s problem obtained on an unstructured triangular mesh using an ADER-FV $O_6$ scheme.

6 Seismograms of the normal and tangential velocity components w.r.t. the surface at the two receivers 1 and 2 for Lamb’s problem.

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

8 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 $O_4$ and (f) ADER-FV $O_4$. The relative seismogram misfit $E$ from equation (79) is given for each trace.

9 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 $O_4$ with 3 attenuation mechanisms and (f) ADER-FV $O_4$ with 3 attenuation mechanisms. The relative seismogram misfit $E$ from equation (79) is given for each trace.