

An arbitrary high-order discontinuous Galerkin method for elastic waves on unstructured meshes – II. The three-dimensional isotropic case

Michael Dumbser^{1,2} and Martin Käser¹

¹Department of Civil and Environmental Engineering, University of Trento, Trento, Italy

²Institut für Aerodynamik und Gasdynamik, Universität Stuttgart, Stuttgart, Germany.

E-mails: michael.dumbser@iag.uni-stuttgart.de, martin.kaeser@ing.uniitn.it

Accepted 2006 June 12. Received 2006 June 12; in original form 2006 March 30

SUMMARY

We present a new numerical method to solve the heterogeneous elastic wave equations formulated as a linear hyperbolic system using first-order derivatives with arbitrary high-order accuracy in space and time on 3-D unstructured tetrahedral meshes. The method combines the Discontinuous Galerkin (DG) Finite Element (FE) method with the ADER approach using Arbitrary high-order DERivatives for flux calculation. In the DG framework, in contrast to classical FE methods, the numerical solution is approximated by piecewise polynomials which allow for discontinuities at element interfaces. Therefore, the well-established theory of numerical fluxes across element interfaces obtained by the solution of Riemann-Problems can be applied as in the finite volume framework. To define a suitable flux over the element surfaces, we solve so-called Generalized Riemann-Problems (GRP) at the element interfaces. The GRP solution provides simultaneously a numerical flux function as well as a time-integration method. The main idea is a Taylor expansion in time in which all time-derivatives are replaced by space derivatives using the so-called Cauchy–Kovalevski or Lax–Wendroff procedure which makes extensive use of the governing PDE. The numerical solution can thus be advanced for one time step without intermediate stages as typical, for example, for classical Runge–Kutta time stepping schemes. Due to the ADER time-integration technique, the same approximation order in space and time is achieved automatically. Furthermore, the projection of the tetrahedral elements in physical space on to a canonical reference tetrahedron allows for an efficient implementation, as many computations of 3-D integrals can be carried out analytically beforehand. Based on a numerical convergence analysis, we demonstrate that the new schemes provide very high order accuracy even on unstructured tetrahedral meshes and computational cost and storage space for a desired accuracy can be reduced by higher-order schemes. Moreover, due to the choice of the basis functions for the piecewise polynomial approximation, the new ADER–DG method shows spectral convergence on tetrahedral meshes. An application of the new method to a well-acknowledged test case and comparisons with analytical and reference solutions, obtained by different well-established methods, confirm the performance of the proposed method. Therefore, the development of the highly accurate ADER–DG approach for tetrahedral meshes provides a numerical technique to approach 3-D wave propagation problems in complex geometry with unforeseen accuracy.

Key words: Discontinuous Galerkin, elastic waves, high-order accuracy, tetrahedral meshes.

1 INTRODUCTION

The solution of the elastic wave equation with very high order of accuracy in space and time is still a challenging task, especially if geometrically complicated 3-D computational domains are involved. The flexibility of unstructured tetrahedral meshes proved to be advantageous when

difficult model geometries have to be respected. However, the numerical schemes handling such tetrahedral meshes can lead to less-accurate results in comparison to structured, regular grids. As the waveforms, that is, amplitude and phase, contain important information about the interior structure and consistency of a material, it is necessary to develop highly accurate numerical methods, in particular for unstructured meshes in order to handle computational domains with geometrically complex external and internal features. Furthermore, such highly accurate numerical schemes serve to eliminate, as far as possible, the errors introduced purely by the numerical approximation of the underlying physical or mathematical model. Therefore, the misfit of simulated and measured data should only be due to errors in the mathematical description of the physical problem or due to incorrect assumptions of the physical parameters of the investigated model and *not* to errors introduced by the numerical scheme itself.

In the past, a number of different numerical schemes have been developed in order to solve the seismic wave equations. Finite difference (FD) schemes for the simulation of *SH* and *P-SV* waves on regular, staggered grids were introduced by Madariaga (1976) and Virieux (1984, 1986). Later these schemes were extended to higher orders (Levander 1988), three space dimensions (Mora 1989; Moczo *et al.* 2002) and anisotropic material (Igel *et al.* 1995; Tessmer 1995), just to mention a few. FD-like methods on triangular and unstructured meshes were developed in Magnier *et al.* (1994) and Käser & Igel (2001a,b). Alternatively, pseudo-spectral (PS) methods (Fornberg 1996) have been developed in Carcione (1994), Tessmer & Kosloff (1994) and Igel (1999), where the space-dependent variables are expanded in a set of orthogonal basis functions which are known exactly at discrete locations and which also allow for the accurate computation of derivatives. However, quite regular meshes have to be used and due to the global character of the derivative operators parallelization is cumbersome. Furthermore, the spectral element method (SEM), originally introduced by Patera (Patera 1984) in the field of computational fluid mechanics, has successfully been applied to the problem of elastic wave propagation in two and three space dimensions in Priolo *et al.* (1994), Komatitsch & Vilotte (1998) and Seriani (1998) and demonstrated advantages, in particular, for the problem of global wave propagation (Komatitsch & Tromp 1999, 2002). The fundamental idea is to combine the advantages of a spectral method with those of a Finite Element (FE) method.

However, we mention that all these schemes usually use rather low-order schemes for the time-integration, for example, second-order Newmark-type schemes (Hughes 1987) or at most fourth-order Runge–Kutta schemes (Igel 1999). Furthermore, we note that for wave propagation problems time-accuracy is crucial, in particular when waves are propagating over many wavelengths, as shown in Dumbser *et al.* (2006). Therefore, we want to point out that the consideration of time-accuracy is an important issue, especially for the seismic wave equation. Recently, a new approach, combining the Discontinuous Galerkin (DG) Method with a time-integration method using Arbitrary high-order DERivatives (ADER), was introduced by Dumbser (2003) and Dumbser & Munz (2005a,b) for linear hyperbolic systems with constant coefficients or for linear systems with variable coefficients in conservative form. The extension to non-conservative systems with variable coefficients and source terms and its application to the simulation of elastic wave equations on unstructured triangular meshes in two space dimensions was then presented by Käser & Dumbser (2005).

In this paper, we present an extension of this new ADER–DG approach in order to solve the elastic wave equations with very high accuracy in both space *and* time on 3-D tetrahedral meshes. As shown in Käser & Dumbser (2005), the proposed numerical method relies on the combination of the ADER approach, originally introduced by Toro (2001) and further developed in Schwartzkopff *et al.* (2004), Titarev & Toro (2002) and Schwartzkopff *et al.* (2004) in the finite volume (FV) framework, and the DG FE method. In our approach, we approximate the unknown solution inside each tetrahedral element by a polynomial, whose coefficients—the degrees of freedom—are advanced in time. Hereby, the solution can be *discontinuous* across the element interfaces, which allows to incorporate the well-established ideas of numerical flux functions from the FV framework. We especially point out that allowing jumps in the numerical solution may be useful with respect to dynamic rupture simulations or to the treatment of large jumps in material properties as shown in Käser & Dumbser (2005).

This paper is structured as follows. In Section 2, we introduce the system of the 3-D elastic wave equations in the non-conservative velocity–stress formulation and its eigenstructure. The proposed DG method is presented in Section 3 together with the ADER approach and an efficient formulation of the required Cauchy–Kovalewski procedure. In Section 4, we briefly address the conditions for absorbing and free-surface boundaries. Results of the numerical convergence rates of the proposed ADER–DG scheme on tetrahedral meshes are shown in Section 5. Finally, in Section 6, we present a comparison of our results with those published after an acknowledged 3-D benchmark test of the Pacific Earthquake Engineering Research Center providing an analytic and a number of reference solutions obtained by well-established codes of other research institutions.

2 ELASTIC WAVE EQUATIONS

The propagation of waves in an elastic medium is based on the theory of linear elasticity (Bedford & Drumheller 1994; Aki & Richards 2002). Combining the definition of strain caused by deformations (Hooke’s law) and the equations of the dynamic relationship between acceleration and stress, the elastic wave equations can be derived as shown in LeVeque (2002). Considering the 3-D elastic wave equation for an isotropic medium in velocity–stress formulation leads to a linear hyperbolic system of the form

$$\begin{aligned}
 \frac{\partial}{\partial t} \sigma_{xx} - (\lambda + 2\mu) \frac{\partial}{\partial x} u - \lambda \frac{\partial}{\partial y} v - \lambda \frac{\partial}{\partial z} w &= 0, \\
 \frac{\partial}{\partial t} \sigma_{yy} - \lambda \frac{\partial}{\partial x} u - (\lambda + 2\mu) \frac{\partial}{\partial y} v - \lambda \frac{\partial}{\partial z} w &= 0, \\
 \frac{\partial}{\partial t} \sigma_{zz} - \lambda \frac{\partial}{\partial x} u - \lambda \frac{\partial}{\partial y} v - (\lambda + 2\mu) \frac{\partial}{\partial z} w &= 0, \\
 \frac{\partial}{\partial t} \sigma_{xy} - \mu \left(\frac{\partial}{\partial x} v + \frac{\partial}{\partial y} u \right) &= 0, \\
 \frac{\partial}{\partial t} \sigma_{yz} - \mu \left(\frac{\partial}{\partial z} v + \frac{\partial}{\partial y} w \right) &= 0, \\
 \frac{\partial}{\partial t} \sigma_{xz} - \mu \left(\frac{\partial}{\partial z} u + \frac{\partial}{\partial x} w \right) &= 0, \\
 \rho \frac{\partial}{\partial t} u - \frac{\partial}{\partial x} \sigma_{xx} - \frac{\partial}{\partial y} \sigma_{xy} - \frac{\partial}{\partial z} \sigma_{xz} &= 0, \\
 \rho \frac{\partial}{\partial t} v - \frac{\partial}{\partial x} \sigma_{xy} - \frac{\partial}{\partial y} \sigma_{yy} - \frac{\partial}{\partial z} \sigma_{yz} &= 0, \\
 \rho \frac{\partial}{\partial t} w - \frac{\partial}{\partial x} \sigma_{xz} - \frac{\partial}{\partial y} \sigma_{yz} - \frac{\partial}{\partial z} \sigma_{zz} &= 0,
 \end{aligned} \tag{1}$$

where λ and μ are the Lamé constants and ρ is the mass density of the material. The normal stress components are given by σ_{xx} , σ_{yy} , and σ_{zz} , and the shear stresses are σ_{xy} , σ_{yz} , and σ_{xz} . The components of the particle velocities in x -, y -, and z -direction are denoted by u , v and w , respectively. This system is a natural extension of the 2-D system already used in the seminal papers of Virieux (Virieux 1984, 1986).

We remark that for notational simplicity we skip the time and space dependencies of the variables, that is, for the rest of this paper, the stresses and velocities are always assumed to be functions of time $t \in \mathbb{R}$ and space $\mathbf{x} = (x, y, z)^T \in \mathbb{R}^3$. The physical properties of the material are functions of space but are constant in time, that is, $\lambda = \lambda(\mathbf{x})$, $\mu = \mu(\mathbf{x})$, and $\rho = \rho(\mathbf{x})$, in order to describe heterogeneous material.

For the investigation of the eigenstructure of the system (1), we use the more compact form

$$\frac{\partial Q_p}{\partial t} + A_{pq} \frac{\partial Q_q}{\partial x} + B_{pq} \frac{\partial Q_q}{\partial y} + C_{pq} \frac{\partial Q_q}{\partial z} = 0, \tag{2}$$

where Q is the vector of the p unknown variables, that is, $Q = (\sigma_{xx}, \sigma_{yy}, \sigma_{zz}, \sigma_{xy}, \sigma_{yz}, \sigma_{xz}, u, v, w)^T$. Note that classical tensor notation is used, which implies summation over each index that appears twice. The matrices $A_{pq} = A_{pq}(\mathbf{x})$, $B_{pq} = B_{pq}(\mathbf{x})$, and $C_{pq} = C_{pq}(\mathbf{x})$ are the space-dependent Jacobian matrices of size $p \times q$, with $p, q = 1, \dots, 9$, and are given by

$$A_{pq} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & -(\lambda + 2\mu) & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -\lambda & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -\lambda & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\mu & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\mu \\ -\frac{1}{\rho} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{1}{\rho} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -\frac{1}{\rho} & 0 & 0 & 0 \end{pmatrix}, \tag{3}$$

$$B_{pq} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\lambda & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -(\lambda + 2\mu) & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\lambda & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -\mu & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\mu \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{1}{\rho} & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{1}{\rho} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\frac{1}{\rho} & 0 & 0 & 0 & 0 \end{pmatrix}, \tag{4}$$

$$C_{pq} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\lambda \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\lambda \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -(\lambda + 2\mu) \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\mu & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -\mu & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -\frac{1}{\rho} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\frac{1}{\rho} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{\rho} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \tag{5}$$

The propagation velocities of the elastic waves are determined by the eigenvalues of the Jacobian matrices A_{pq} , B_{pq} and C_{pq} and are given by

$$s_1 = -c_p, \quad s_2 = -c_s, \quad s_3 = -c_s, \quad s_4 = 0, \quad s_5 = 0, \quad s_6 = 0, \quad s_7 = c_s, \quad s_8 = c_s, \quad s_9 = c_p, \tag{6}$$

where

$$c_p = \sqrt{\frac{\lambda + 2\mu}{\rho}}, \quad \text{and} \quad c_s = \sqrt{\frac{\mu}{\rho}} \tag{7}$$

are the P wave and S wave velocities, respectively. We remark that for the isotropic case the wave speeds in (6) and (7) do not depend on the propagation direction. The eigenvalues show, that the two P waves propagate with speeds s_1 and s_9 and the four S waves propagate with speeds s_2, s_3, s_7 and s_8 , however, with different polarizations. The vectors $R_{p1}^A, \dots, R_{p9}^A$ denote the nine right eigenvectors of the Jacobian matrix in eq. (3) and are given by the columns of the matrix of right eigenvectors

$$R_{pq}^A = \begin{pmatrix} \lambda + 2\mu & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda + 2\mu \\ \lambda & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \lambda \\ \lambda & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \lambda \\ 0 & \mu & 0 & 0 & 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \mu & 0 & 0 & 0 & \mu & 0 & 0 \\ c_p & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -c_p \\ 0 & c_s & 0 & 0 & 0 & 0 & 0 & -c_s & 0 \\ 0 & 0 & c_s & 0 & 0 & 0 & -c_s & 0 & 0 \end{pmatrix}. \tag{8}$$

The derived eigenstructure of the elastic wave equations in three space dimensions provides the necessary information to construct the ADER–DG scheme as described in Section 3.

3 THE NUMERICAL SCHEME

For the construction of the numerical scheme, we consider the general linear hyperbolic system of equations with variable coefficients as given in eq. (2). The computational domain $\Omega \in \mathbb{R}^3$ is divided into conforming tetrahedral elements $\mathcal{T}^{(m)}$ being addressed by a unique index (m) . Furthermore, we suppose the matrices A_{pq} , B_{pq} and C_{pq} to be piecewise constant inside an element $\mathcal{T}^{(m)}$.

3.1 Semi-discrete form of the scheme

The numerical solution Q_h of eq. (2) is approximated inside each tetrahedron $\mathcal{T}^{(m)}$ by a linear combination of space-dependent but time-independent polynomial basis functions $\Phi_l(\xi, \eta, \zeta)$ of degree N with support $\mathcal{T}^{(m)}$ and with only time-dependent degrees of freedom $\hat{Q}_{pl}^{(m)}(t)$:

$$[Q_h^{(m)}]_p(\xi, \eta, \zeta, t) = \hat{Q}_{pl}^{(m)}(t)\Phi_l(\xi, \eta, \zeta), \tag{9}$$

where ξ, η and ζ are the coordinates in a reference element \mathcal{T}_E and are defined by eq. (A1) in Appendix A. The index h denotes the numerical solution, the index p stands for the number of unknowns in the vector Q and l indicates the l th basis function. We use the orthogonal basis functions Φ_k as given in Cockburn *et al.* (2000). Multiplying eq. (2) by a test function Φ_k and integrating over a tetrahedral element $\mathcal{T}^{(m)}$ gives

$$\int_{\mathcal{T}^{(m)}} \Phi_k \frac{\partial Q_p}{\partial t} dV + \int_{\mathcal{T}^{(m)}} \Phi_k \left(A_{pq} \frac{\partial Q_q}{\partial x} + B_{pq} \frac{\partial Q_q}{\partial y} + C_{pq} \frac{\partial Q_q}{\partial z} \right) dV = 0. \tag{10}$$

Table 1. Face definition on tetrahedrons.

Face	Points		
1	1	3	2
2	1	2	4
3	1	4	3
4	2	3	4

Integration of eq. (10) by parts then yields

$$\int_{\mathcal{T}^{(m)}} \Phi_k \frac{\partial Q_p}{\partial t} dV + \int_{\partial \mathcal{T}^{(m)}} \Phi_k F_p^h dS - \int_{\mathcal{T}^{(m)}} \left(\frac{\partial \Phi_k}{\partial x} A_{pq} Q_q + \frac{\partial \Phi_k}{\partial y} B_{pq} Q_q + \frac{\partial \Phi_k}{\partial z} C_{pq} Q_q \right) dV = 0, \tag{11}$$

where a numerical flux F_p^h has been introduced in the surface integral since Q_h may be discontinuous at an element boundary. As we suppose rotational invariance of the system in (2) for the isotropic case, the flux can be derived for a coordinate system, which is aligned with the outward pointing unit normal vector \mathbf{n} of an element boundary, that is, a triangular face of a tetrahedron. The required transformation (rotation) of the unknowns in vector Q_p from the global Cartesian system to the vector Q_q^n in a local normal, that is, *face-aligned*, coordinate system is given by

$$Q_p = T_{pq} Q_q^n. \tag{12}$$

For the 3-D elastic wave equations, the transformation (rotation) matrix T_{pq} in eq. (12) reads as

$$T_{pq} = \begin{pmatrix} n_x^2 & s_x^2 & t_x^2 & 2n_x s_x & 2s_x t_x & 2n_x t_x & 0 & 0 & 0 \\ n_y^2 & s_y^2 & t_y^2 & 2n_y s_y & 2s_y t_y & 2n_y t_y & 0 & 0 & 0 \\ n_z^2 & s_z^2 & t_z^2 & 2n_z s_z & 2s_z t_z & 2n_z t_z & 0 & 0 & 0 \\ n_y n_x & s_y s_x & t_y t_x & n_y s_x + n_x s_y & s_y t_x + s_x t_y & n_y t_x + n_x t_y & 0 & 0 & 0 \\ n_z n_y & s_z s_y & t_z t_y & n_z s_y + n_y s_z & s_z t_y + s_y t_z & n_z t_y + n_y t_z & 0 & 0 & 0 \\ n_z n_x & s_z s_x & t_z t_x & n_z s_x + n_x s_z & s_z t_x + s_x t_z & n_z t_x + n_x t_z & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & n_x & s_x & t_x \\ 0 & 0 & 0 & 0 & 0 & 0 & n_y & s_y & t_y \\ 0 & 0 & 0 & 0 & 0 & 0 & n_z & s_z & t_z \end{pmatrix}, \tag{13}$$

with the components of the normal vector $\mathbf{n} = (n_x, n_y, n_z)^T$ and the two tangential vectors $\mathbf{s} = (s_x, s_y, s_z)^T$ and $\mathbf{t} = (t_x, t_y, t_z)^T$, which lie in the plane determined by the boundary face of the tetrahedron and are orthogonal to each other and the normal vector \mathbf{n} . Usually, we define vector \mathbf{s} so that it points from the local face node 1 to the local face node 2, see Table 1.

For the linear system in eq. (2), we use an exact Riemann solver, as given in Toro (1999), to compute the state at the cell interfaces by upwinding. The flux is then computed with the Jacobian matrix in eq. (3), that is, with the material properties given in the considered element. In the global, Cartesian xyz -system, the numerical flux F_p^h in eq. (11) for tetrahedron $\mathcal{T}^{(m)}$ across the interface with one of the neighbour tetrahedrons $\mathcal{T}^{(m_j)}$, $j = 1, 2, 3, 4$, is then given by

$$F_p^h = \frac{1}{2} T_{pq} (A_{qr}^{(m)} + |A_{qr}^{(m)}|) (T_{rs})^{-1} \hat{Q}_{sl}^{(m)} \Phi_l^{(m)} + \frac{1}{2} T_{pq} (A_{qr}^{(m)} - |A_{qr}^{(m)}|) (T_{rs})^{-1} \hat{Q}_{sl}^{(m_j)} \Phi_l^{(m_j)}, \tag{14}$$

where $\hat{Q}_{sl}^{(m)} \Phi_l^{(m)}$ and $\hat{Q}_{sl}^{(m_j)} \Phi_l^{(m_j)}$ are the boundary extrapolated values of the numerical solution from element $\mathcal{T}^{(m)}$ and the j th-side neighbour $\mathcal{T}^{(m_j)}$, respectively. The matrix $(T_{rs})^{-1}$ represents the back-transformation into the global xyz -system. We point out that the boundary values of both elements adjacent to an interface contribute to the numerical flux. Furthermore, due to the rotation it is sufficient to evaluate the Jacobian matrix $A_{qr}^{(m)}$ of eq. (3) in x -direction with the material properties given in tetrahedron $\mathcal{T}^{(m)}$. This way, the numerical flux respects the non-conservative form of the governing eq. (2). The notation $|A_{qr}^{(m)}|$ of the absolute value of the Jacobian matrix has the meaning of applying the absolute value operator to the eigenvalues given in eq. (6), that is,

$$|A_{qr}^{(m)}| = R_{qp}^A |\Lambda_{ps}| (R_{sr}^A)^{-1}, \quad \text{with} \quad |\Lambda_{ps}| = \text{diag}(|s_1|, |s_2|, \dots). \tag{15}$$

Now, inserting eqs (9) and (14) into eq. (11) and splitting the boundary integral into the contributions of each face j of the tetrahedron $\mathcal{T}^{(m)}$, we obtain

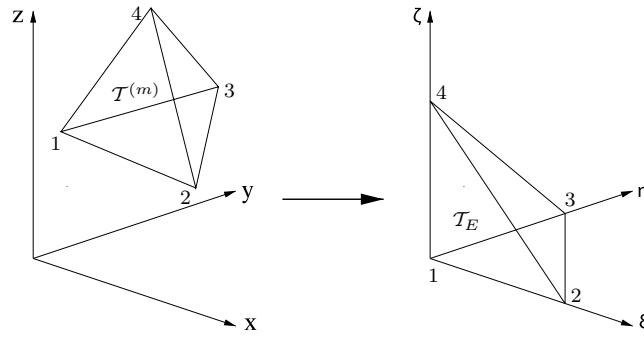


Figure 1. Transformation from the physical tetrahedron $\mathcal{T}^{(m)}$ to the canonical reference tetrahedron \mathcal{T}_E with nodes $(0, 0, 0)$, $(1, 0, 0)$, $(0, 1, 0)$, and $(0, 0, 1)$.

$$\begin{aligned}
 & \frac{\partial}{\partial t} \hat{Q}_{pl}^{(m)} \int_{\mathcal{T}^{(m)}} \Phi_k \Phi_l dV \\
 & + \sum_{j=1}^4 \frac{1}{2} T_{pq}^j (A_{qr}^{(m)} + |A_{qr}^{(m)}|) (T_{rs}^j)^{-1} \hat{Q}_{sl}^{(m)} \int_{[\partial \mathcal{T}^{(m)}]_j} \Phi_k^{(m)} \Phi_l^{(m)} dS \\
 & + \sum_{j=1}^4 \frac{1}{2} T_{pq}^j (A_{qr}^{(m)} - |A_{qr}^{(m)}|) (T_{rs}^j)^{-1} \hat{Q}_{sl}^{(m_j)} \int_{[\partial \mathcal{T}^{(m)}]_j} \Phi_k^{(m)} \Phi_l^{(m_j)} dS \\
 & - A_{pq} \hat{Q}_{ql}^{(m)} \int_{\mathcal{T}^{(m)}} \frac{\partial \Phi_k}{\partial x} \Phi_l dV - B_{pq} \hat{Q}_{ql}^{(m)} \int_{\mathcal{T}^{(m)}} \frac{\partial \Phi_k}{\partial y} \Phi_l dV - C_{pq} \hat{Q}_{ql}^{(m)} \int_{\mathcal{T}^{(m)}} \frac{\partial \Phi_k}{\partial z} \Phi_l dV = 0.
 \end{aligned} \tag{16}$$

Eq. (16) is written in the global xyz -system. If we transform each physical tetrahedron $\mathcal{T}^{(m)}$ to a canonical reference tetrahedron \mathcal{T}_E in a $\xi\eta z$ -reference system as shown in Fig. 1, the method can be implemented much more efficiently, since many integrals can be pre-computed beforehand in this reference system.

With respect to the coordinate transformation into the $\xi\eta z$ -reference system, we have furthermore

$$dx dy dz = |J| d\xi d\eta, d\zeta \tag{17}$$

and the transformed gradients of the form

$$\begin{pmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} \end{pmatrix} = \begin{pmatrix} \frac{\partial \xi}{\partial x} & \frac{\partial \eta}{\partial x} & \frac{\partial \zeta}{\partial x} \\ \frac{\partial \xi}{\partial y} & \frac{\partial \eta}{\partial y} & \frac{\partial \zeta}{\partial y} \\ \frac{\partial \xi}{\partial z} & \frac{\partial \eta}{\partial z} & \frac{\partial \zeta}{\partial z} \end{pmatrix} \begin{pmatrix} \frac{\partial}{\partial \xi} \\ \frac{\partial}{\partial \eta} \\ \frac{\partial}{\partial \zeta} \end{pmatrix}. \tag{18}$$

Integrating in the reference system using eqs (17) and (18), the semi-discrete DG formulation of eq. (16) in the reference element \mathcal{T}_E reads as

$$\begin{aligned}
 & \frac{\partial}{\partial t} \hat{Q}_{pl}^{(m)} |J| \int_{\mathcal{T}_E} \Phi_k \Phi_l d\xi d\eta d\zeta \\
 & + \sum_{j=1}^4 T_{pq}^j \frac{1}{2} (A_{qr}^{(m)} + |A_{qr}^{(m)}|) (T_{rs}^j)^{-1} \hat{Q}_{sl}^{(m)} |S_j| F_{kl}^{-,j} \\
 & + \sum_{j=1}^4 T_{pq}^j \frac{1}{2} (A_{qr}^{(m)} - |A_{qr}^{(m)}|) (T_{rs}^j)^{-1} \hat{Q}_{sl}^{(m_j)} |S_j| F_{kl}^{+,j,i,h} \\
 & - A_{pq}^* \hat{Q}_{ql}^{(m)} |J| \int_{\mathcal{T}_E} \frac{\partial \Phi_k}{\partial \xi} \Phi_l d\xi d\eta d\zeta - B_{pq}^* \hat{Q}_{ql}^{(m)} |J| \int_{\mathcal{T}_E} \frac{\partial \Phi_k}{\partial \eta} \Phi_l d\xi d\eta d\zeta - C_{pq}^* \hat{Q}_{ql}^{(m)} |J| \int_{\mathcal{T}_E} \frac{\partial \Phi_k}{\partial \zeta} \Phi_l d\xi d\eta d\zeta = 0,
 \end{aligned} \tag{19}$$

where $|S_j|$ denotes the area of face j and the special linear combination of the Jacobians

$$A_{pq}^* = A_{pq} \frac{\partial \xi}{\partial x} + B_{pq} \frac{\partial \xi}{\partial y} + C_{pq} \frac{\partial \xi}{\partial z}, \tag{20}$$

$$B_{pq}^* = A_{pq} \frac{\partial \eta}{\partial x} + B_{pq} \frac{\partial \eta}{\partial y} + C_{pq} \frac{\partial \eta}{\partial z}, \tag{21}$$

$$C_{pq}^* = A_{pq} \frac{\partial \zeta}{\partial x} + B_{pq} \frac{\partial \zeta}{\partial y} + C_{pq} \frac{\partial \zeta}{\partial z}. \tag{22}$$

The integrals

$$M_{kl} = \int_{\mathcal{T}_E} \Phi_k \Phi_l d\xi d\eta d\zeta, \tag{23}$$

Table 2. (a) 3-D volume coordinates $\xi^{(j)}$ as a function of the face parameters χ and τ . (b) Transformation of the face parameters χ and τ of the tetrahedron's face to the face parameters $\tilde{\chi}$ and $\tilde{\tau}$ in the neighbour tetrahedron according to the three possible orientations h of the neighbour face.

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

$$K_{kl}^\xi = \int_{\mathcal{T}_E} \frac{\partial \Phi_k}{\partial \xi} \Phi_l d\xi d\eta d\zeta, \tag{24}$$

$$K_{kl}^\eta = \int_{\mathcal{T}_E} \frac{\partial \Phi_k}{\partial \eta} \Phi_l d\xi d\eta d\zeta, \tag{25}$$

$$K_{kl}^\zeta = \int_{\mathcal{T}_E} \frac{\partial \Phi_k}{\partial \zeta} \Phi_l d\xi d\eta d\zeta, \tag{26}$$

over the reference element \mathcal{T}_E can be easily calculated beforehand by a computer algebra system (e.g. Maple). Furthermore, the flux matrices F_{kl}^{-j} and $F_{kl}^{+,j,i,h}$ in eq. (19) can be calculated analytically once on the reference element via eq. (27) and then can be stored.

In the following, we give the details of calculating those flux matrices on tetrahedrons in 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 two space dimensions as well as in three space dimensions (see Fig. 1). The vector of volume coordinates ξ is then given on the faces via mapping functions from the face parameters χ and τ , as shown in Table 2(a). Last but not the 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 χ and τ inside the element to the corresponding face parameters $\tilde{\chi}$ and $\tilde{\tau}$ in the neighbour face. Whereas in two space dimensions this transformation is always $\tilde{\chi} = 1 - \chi$, in three space dimensions the transformation depends on the orientation of the neighbour's face with respect to the local face of the considered element $\mathcal{T}^{(m)}$, since via rotation of the triangular faces there may be three possible orientations. The corresponding mappings are given in Table 2(b).

In three space dimensions, all possible flux matrices are given by

$$F_{kl}^{-j} = \int_{\partial(\mathcal{T}_E)_j} \Phi_k(\xi^{(j)}(\chi, \tau)) \Phi_l(\xi^{(j)}(\chi, \tau)) d\chi d\tau, \quad \forall 1 \leq j \leq 4, \tag{27}$$

$$F_{kl}^{+,j,i,h} = \int_{\partial(\mathcal{T}_E)_j} \Phi_k(\xi^{(j)}(\chi, \tau)) \Phi_l(\xi^{(i)}(\tilde{\chi}^{(h)}(\chi, \tau), \tilde{\tau}^{(h)}(\chi, \tau))) d\chi d\tau, \quad \forall 1 \leq i \leq 4, \quad \forall 1 \leq h \leq 3.$$

The left-hand state flux matrix (superscript ‘-’) F_{kl}^{-j} accounts for the contribution of the element (m) itself to the fluxes over face j and the right-hand state flux matrix (superscript ‘+’) $F_{kl}^{+,j,i,h}$ accounts for the contribution of the element's direct side neighbour (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 $1 \leq h \leq 3$ denotes the number of the local node in the neighbour's face which lies on the local vertex 1 of face j in tetrahedron number (m). Index h also depends on the mesh generator. On a given tetrahedral mesh, where indices i and h are known, only four of the 48 possible matrices $F_{kl}^{+,j,i,h}$ are used per element.

If the semi-discrete eq. (19) is integrated in time with a Runge–Kutta method, we obtain a quadrature-free Runge–Kutta DG scheme, see Atkins & Shu (1998), the order of which would be essentially limited to 4 if we do not want to use inefficient Runge–Kutta schemes. In the following, we present the ADER time-integration technique making extensive use of the governing eq. (2).

3.2 The ADER time-discretization

The efficiency of Runge–Kutta time-discretization schemes drastically decreases if the order of accuracy becomes greater than 4, due to the so-called Butcher barriers (Butcher 1987), and the number of intermediate Runge–Kutta stages becomes larger than the formal order of accuracy. Therefore, we apply the ADER approach to the semi-discrete form of the DG scheme (19) in order to achieve the same accuracy of the time-discretization as for the space discretization.

The main ingredients of the ADER approach are a Taylor expansion in time, the solution of Derivative Riemann-Problems (DRP) (Toro & Titarev 2002) to approximate the space derivatives at the interface and the Cauchy–Kovalewski procedure for replacing the time-derivatives in the Taylor series by space derivatives. In this section, we show how the ADER–DG approach, presented in Dumbser (2005) and Dumbser & Munz (2005a), can be extended to three space dimensions to achieve high-order time-integration of the DG method for linear hyperbolic systems.

For the development of linear ADER–DG schemes, we first need a general formula for the Cauchy–Kovalewski procedure in order to replace the time-derivatives in the Taylor expansion in time by pure space derivatives. Since the basis functions Φ_l in (9) are given in the $\xi\eta\zeta$ -system, we need a Cauchy–Kovalewski procedure which makes use of the spatial derivatives with respect to ξ , η and ζ . Therefore, we rewrite our original PDE (2) with the use of eq. (18) as

$$\frac{\partial Q_p}{\partial t} + A_{pq} \left(\frac{\partial \xi}{\partial x} \frac{\partial Q_q}{\partial \xi} + \frac{\partial \eta}{\partial x} \frac{\partial Q_q}{\partial \eta} + \frac{\partial \zeta}{\partial x} \frac{\partial Q_q}{\partial \zeta} \right) + B_{pq} \left(\frac{\partial \xi}{\partial y} \frac{\partial Q_q}{\partial \xi} + \frac{\partial \eta}{\partial y} \frac{\partial Q_q}{\partial \eta} + \frac{\partial \zeta}{\partial y} \frac{\partial Q_q}{\partial \zeta} \right) + C_{pq} \left(\frac{\partial \xi}{\partial z} \frac{\partial Q_q}{\partial \xi} + \frac{\partial \eta}{\partial z} \frac{\partial Q_q}{\partial \eta} + \frac{\partial \zeta}{\partial z} \frac{\partial Q_q}{\partial \zeta} \right) = 0.$$

Rearranging this equation, we get

$$\frac{\partial Q_p}{\partial t} + \left(A_{pq} \frac{\partial \xi}{\partial x} + B_{pq} \frac{\partial \xi}{\partial y} + C_{pq} \frac{\partial \xi}{\partial z} \right) \frac{\partial Q_q}{\partial \xi} + \left(A_{pq} \frac{\partial \eta}{\partial x} + B_{pq} \frac{\partial \eta}{\partial y} + C_{pq} \frac{\partial \eta}{\partial z} \right) \frac{\partial Q_q}{\partial \eta} + \left(A_{pq} \frac{\partial \zeta}{\partial x} + B_{pq} \frac{\partial \zeta}{\partial y} + C_{pq} \frac{\partial \zeta}{\partial z} \right) \frac{\partial Q_q}{\partial \zeta} = 0,$$

and finally, by using the definitions in eqs (20), (21) and (22), we obtain

$$\frac{\partial Q_p}{\partial t} + A_{pq}^* \frac{\partial Q_q}{\partial \xi} + B_{pq}^* \frac{\partial Q_q}{\partial \eta} + C_{pq}^* \frac{\partial Q_q}{\partial \zeta} = 0. \tag{28}$$

The k th time-derivative as a function of pure space derivatives in the $\xi\eta\zeta$ -reference system is the result of the Cauchy–Kovalewski procedure applied to eq. (28) and is given by

$$\frac{\partial^k Q_p}{\partial t^k} = (-1)^k \left(A_{pq}^* \frac{\partial}{\partial \xi} + B_{pq}^* \frac{\partial}{\partial \eta} + C_{pq}^* \frac{\partial}{\partial \zeta} \right)^k Q_q. \tag{29}$$

We point out that it is a key issue to use the Cauchy–Kovalewski procedure in the form (29) since it allows us to pre-calculate many matrices beforehand, as we will show in the following.

We develop the solution of eq. (2) in a Taylor series in time up to order N :

$$Q_p(\xi, \eta, \zeta, t) = \sum_{k=0}^N \frac{t^k}{k!} \frac{\partial^k}{\partial t^k} Q_p(\xi, \eta, \zeta, 0), \tag{30}$$

and replace time-derivatives by space derivatives, using eq. (29):

$$Q_p(\xi, \eta, \zeta, t) = \sum_{k=0}^N \frac{t^k}{k!} (-1)^k \left(A_{pq}^* \frac{\partial}{\partial \xi} + B_{pq}^* \frac{\partial}{\partial \eta} + C_{pq}^* \frac{\partial}{\partial \zeta} \right)^k Q_q(\xi, \eta, \zeta, 0). \tag{31}$$

We then introduce the DG approximation (9) and obtain

$$Q_p(\xi, \eta, \zeta, t) = \sum_{k=0}^N \frac{t^k}{k!} (-1)^k \left(A_{pq}^* \frac{\partial}{\partial \xi} + B_{pq}^* \frac{\partial}{\partial \eta} + C_{pq}^* \frac{\partial}{\partial \zeta} \right)^k \Phi_l(\xi, \eta, \zeta) \hat{Q}_{ql}(0). \tag{32}$$

This approximation can now be projected on to each basis function in order to get an approximation of the evolution of the degrees of freedom during one time step from time-level n to time-level $n + 1$. We obtain

$$\hat{Q}_{pl}(t) = \frac{\left\langle \Phi_n, \sum_{k=0}^N \frac{t^k}{k!} (-1)^k \left(A_{pq}^* \frac{\partial}{\partial \xi} + B_{pq}^* \frac{\partial}{\partial \eta} + C_{pq}^* \frac{\partial}{\partial \zeta} \right)^k \Phi_m(\xi, \eta, \zeta) \right\rangle}{\langle \Phi_n, \Phi_l \rangle} \hat{Q}_{qm}(0) \tag{33}$$

where $\langle a, b \rangle = \int_{T_E} a \cdot b \, dV$ denotes the inner product over the reference tetrahedron T_E and the division by $\langle \Phi_n, \Phi_l \rangle$ 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 Φ_l . Eq. (33) can be integrated analytically in time and we obtain

$$\int_0^{\Delta t} \hat{Q}_{pl}(t) \, dt = \frac{\left\langle \Phi_n, \sum_{k=0}^N \frac{\Delta t^{(k+1)}}{(k+1)!} (-1)^k \left(A_{pq}^* \frac{\partial}{\partial \xi} + B_{pq}^* \frac{\partial}{\partial \eta} + C_{pq}^* \frac{\partial}{\partial \zeta} \right)^k \Phi_m(\xi, \eta, \zeta) \right\rangle}{\langle \Phi_n, \Phi_l \rangle} \hat{Q}_{qm}(0). \tag{34}$$

Introducing the definition

$$I_{plqm}(\Delta t) = \frac{\left\langle \Phi_n, \sum_{k=0}^N \frac{\Delta t^{(k+1)}}{(k+1)!} (-1)^k \left(A_{pq}^* \frac{\partial}{\partial \xi} + B_{pq}^* \frac{\partial}{\partial \eta} + C_{pq}^* \frac{\partial}{\partial \zeta} \right)^k \Phi_m(\xi, \eta, \zeta) \right\rangle}{\langle \Phi_n, \Phi_l \rangle}, \tag{35}$$

the time-integrated degrees of freedom in eq. (34) can be expressed as

$$\int_0^{\Delta t} \hat{Q}_{pl}(t) \, dt = I_{plqm}(\Delta t) \hat{Q}_{qm}(0), \tag{36}$$

where $I_{plqm}(\Delta t)$ represents a 4-D tensor including the Cauchy–Kovalewski procedure and $\hat{Q}_{qm}(0)$ denotes the $q \times m$ matrix of the degrees of freedom at time-level n . Finally, the fully discrete ADER–DG scheme by integration of eq. (19) in time is given by

$$\begin{aligned} & \left[\left(\hat{Q}_{pl}^{(m)} \right)^{n+1} - \left(\hat{Q}_{pl}^{(m)} \right)^n \right] |J| M_{kl} \\ & + \frac{1}{2} \sum_{j=1}^4 T_{pq}^j (A_{qr}^{(m)} + |A_{qr}^{(m)}|) (T_{rs}^j)^{-1} |S_j| F_{kl}^{-,j} \cdot I_{slmn}(\Delta t) \left(\hat{Q}_{mn}^{(m)} \right)^n \\ & + \frac{1}{2} \sum_{j=1}^4 T_{pq}^j (A_{qr}^{(m)} - |A_{qr}^{(m)}|) (T_{rs}^j)^{-1} |S_j| F_{kl}^{+,j,i,h} \cdot I_{slmn}(\Delta t) \left(\hat{Q}_{mn}^{(m)} \right)^n \\ & - A_{pq}^* |J| K_{kl}^\xi \cdot I_{qlmn}(\Delta t) \left(\hat{Q}_{mn}^{(m)} \right)^n - B_{pq}^* |J| K_{kl}^\eta \cdot I_{qlmn}(\Delta t) \left(\hat{Q}_{mn}^{(m)} \right)^n - C_{pq}^* |J| K_{kl}^\zeta \cdot I_{qlmn}(\Delta t) \left(\hat{Q}_{mn}^{(m)} \right)^n = 0. \end{aligned} \quad (37)$$

The scheme is quadrature-free and performs high-order time-integration from time-level 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. Note that for any DG scheme the initial condition at t^0 must be projected on to the degrees of freedom via L^2 projection.

3.3 An efficient algorithm for the Cauchy–Kovalewski procedure of linear systems

On unstructured meshes, we must evaluate eq. (35) in each tetrahedron in each time step, if we do not want to store all the tensor coefficients $I_{plqm}(\Delta t)$ for each tetrahedron. Therefore, in the following we show a very efficient algorithm for its evaluation on the reference element, which is a straightforward extension of the algorithm presented in Käser & Dumbser (2005). Eq. (35) contains a binomial formula for matrices whose products unfortunately do not commute in the general case. If expanded explicitly, eq. (35) becomes

$$I_{plqm}(\Delta t) = \langle \Phi_n, \Phi_l \rangle^{-1} \cdot \left\langle \Phi_n, \sum_{k=0}^N \frac{\Delta t^{(k+1)}}{(k+1)!} (-1)^k \cdot \sum_{n_B=0}^k \sum_{n_C=0}^{k-n_B} S_{pq} (A_{pq}^*, B_{pq}^*, C_{pq}^*, k - n_B - n_C, n_B, n_C) \frac{\partial^k}{\partial \xi^{(k-n_B-n_C)} \partial \eta^{n_B} \partial \zeta^{n_C}} \Phi_m \right\rangle. \quad (38)$$

Hence, in the evaluation of eq. (35), we have to compute the sums S_{pq} of all products of A_{pq}^* , B_{pq}^* , and C_{pq}^* in which matrix A_{pq}^* appears $n_A = k - n_B - n_C$ times, matrix B_{pq}^* appears n_B times and matrix C_{pq}^* appears n_C times, taking into account all possible matrix permutations. Without proof, we give a recursive algorithm defining the S_{pq} in Step 1 of Algorithm 1. It recycles all previously computed data if implemented in an unrolled manner, storing intermediate results. In Algorithm 1, δ_{pq} denotes the usual Kronecker symbol.

Algorithm 1: (efficient Cauchy–Kovalewski procedure)

Step 1: Compute and store temporarily the sums of all matrix products of A_{pq}^* , B_{pq}^* , and C_{pq}^* with all possible permutations:

$$\begin{aligned} S_{pq}(A_{pq}^*, B_{pq}^*, C_{pq}^*, n_A, n_B, n_C) & := 0, \quad \text{if } n_A < 0 \quad \vee \quad n_B < 0 \quad \vee \quad n_C < 0, \\ S_{pq}(A_{pq}^*, B_{pq}^*, C_{pq}^*, 0, 0, 0) & := \delta_{pq}, \\ S_{pq}(A_{pq}^*, B_{pq}^*, C_{pq}^*, n_A, n_B, n_C) & = S_{pr}(A_{pr}^*, B_{pr}^*, C_{pr}^*, n_A - 1, n_B, n_C) \cdot A_{rq}^* + \\ & \quad S_{pr}(A_{pr}^*, B_{pr}^*, C_{pr}^*, n_A, n_B - 1, n_C) \cdot B_{rq}^* + \\ & \quad S_{pr}(A_{pr}^*, B_{pr}^*, C_{pr}^*, n_A, n_B, n_C - 1) \cdot C_{rq}^*, \quad \forall \quad 0 \leq n_A + n_B + n_C \leq N. \end{aligned}$$

Step 2: Exchange summation and integration in eq. (38) and compute $I_{plqm}(\Delta t)$ as

$$\begin{aligned} I_{plqm}(\Delta t) & = \sum_{k=0}^N \frac{\Delta t^{(k+1)}}{(k+1)!} (-1)^k \sum_{n_B=0}^k \sum_{n_C=0}^{k-n_B} S_{pq}(A_{pq}^*, B_{pq}^*, C_{pq}^*, k - n_B - n_C, n_B, n_C) \cdot \\ & \quad \langle \Phi_n, \Phi_l \rangle^{-1} \left\langle \Phi_n, \frac{\partial^k}{\partial \xi^{(k-n_B-n_C)} \partial \eta^{n_B} \partial \zeta^{n_C}} \Phi_m \right\rangle. \end{aligned} \quad (39)$$

This algorithm is used in each tetrahedron $\mathcal{T}^{(m)}$ in each time step. However, all projections of the space derivatives of the basis functions on to all the test functions, as they appear in Step 2 in Algorithm 1, can be pre-computed once and then stored. They are defined on the canonical reference tetrahedron and depend neither on the mesh nor on the time step.

4 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–DG method in the following.

4.1 Absorbing boundaries

At absorbing boundaries, no waves are supposed to enter the computational domain and the waves travelling outwards should pass the boundary without reflections. There is a whole scientific community dealing with absorbing or non-reflective boundary conditions; however, in this

section, we present a simple approach, that so far yielded satisfactory results, at least for our purposes. The numerical flux (14) is based on the solution of the Riemann-Problem given by the jump across the element interface. It is a strict upwind method, that is, outgoing waves at an element interface are only influenced by the state in the inside element itself. In contrast, the flux contribution of incoming waves is purely due to the state in the neighbour element. Thus, a simple implementation of absorbing boundary conditions is to use the following numerical flux in eq. (37) at all those tetrahedral faces that coincide with an absorbing boundary:

$$F_p^{\text{AbsorbBC}} = \frac{1}{2} T_{pq} (A_{qr}^{(m)} + |A_{qr}^{(m)}|) (T_{rs})^{-1} \hat{Q}_{sl}^{(m)} \Phi_l^{(m)}. \quad (40)$$

The flux function (40) allows only for outgoing waves, which are merely defined by the state in the element due to upwinding. Since incoming waves are not allowed, the respective flux contribution must vanish, that is, it is set to zero in the implementation of the method.

We are aware of that these absorbing boundary conditions have some problems at corners or for *grazing* incidence of waves. Therefore, in future work, approaches like the Perfectly Matched Layer (PML) technique, as introduced in Bérenger (1994) and applied in Collino & Tsogka (2001) and Komatitsch & Tromp (2003), should be incorporated to improve the performance of the proposed scheme for such boundaries.

4.2 Free-surface boundaries

On the free surface of an elastic medium, the normal stress and the shear stresses with respect to the boundary are determined by physical constraints. Outside the elastic medium, there are no external forces that retract the particles into their original position. Therefore, the normal stress and the shear stress values at the free surface have to be zero. In contrast to classical continuous FE methods or SEM, we have no direct control on the values at the boundaries within the DG framework. However, the boundary conditions can be imposed via the numerical flux, as in the FV framework. 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 Q_p , that we want to be zero at the domain boundary, we prescribe a virtual boundary extrapolated component outside 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 eq. (37) can then be formulated as follows:

$$F_p^{\text{FreeBC}} = \frac{1}{2} T_{pq} (A_{qr}^{(m)} + |A_{qr}^{(m)}|) (T_{rs})^{-1} \hat{Q}_{sl}^{(m)} \Phi_l^{(m)} + \frac{1}{2} T_{pq} (A_{qr}^{(m)} - |A_{qr}^{(m)}|) \Gamma_{rs} (T_{st})^{-1} \hat{Q}_{tl}^{(m)} \Phi_l^{(m)}, \quad (41)$$

where the matrix $\Gamma_{rs} = \text{diag}(-1, 1, 1, -1, 1, -1, 1, 1, 1)$ accounts for the mirroring of normal and shear stresses with respect to the face-normal direction. We remark that the solution of the inverse Riemann-Problem is *not* equivalent to the FD approach of adding fictitious *ghost* points, but is similar to the FV framework and provides the exact values of the normal and shear stresses as required by the free-surface boundary condition. Numerical tests, such as Lamb's problem in two space dimensions shown in Käser & Dumbser (2005) confirm the performance and accuracy of this approach. Results of the numerical experiment in Section 6 make it clear that this approach directly extends into three space dimensions.

5 CONVERGENCE ANALYSIS

Here, we present the results of the numerical convergence analysis in order to confirm the very high accuracy of the proposed ADER-DG method on unstructured tetrahedral meshes. Convergence orders of the ADER-DG schemes are shown from second to seventh order in space and time and are denoted by ADER-DG $\mathcal{O}2$ to ADER-DG $\mathcal{O}7$, respectively. Furthermore, we show the spectral convergence of the ADER-DG method on tetrahedral meshes due to the choice of the orthogonal basis functions Φ_l in eq. (9).

To determine the convergence orders, we solve the 3-D elastic wave equations in eq. (1) in a cube-shaped computational domain $\Omega = [-50, 50] \times [-50, 50] \times [-50, 50] \in \mathbb{R}^3$ with periodic boundary conditions. The initial condition is given by

$$Q_p^0 = Q_p(\mathbf{x}, 0) = R_{p1}^{A_n} \sin(\mathbf{k} \cdot \mathbf{x}) + R_{p8}^{A_n} \sin(\mathbf{k} \cdot \mathbf{x}), \quad (42)$$

with the wave number

$$\mathbf{k} = (k_x, k_y, k_z)^T = \frac{2\pi}{100} (1, 1, 1)^T. \quad (43)$$

The vectors $R_{p1}^{A_n}$ and $R_{p8}^{A_n}$ are two of the eigenvectors of eq. (8) rotated in normal direction. Therefore, the initial condition (42) creates a plane sinusoidal *P* wave travelling along the diagonal direction $\mathbf{n} = (-1, -1, -1)^T$ of the cube, as well as a plane sinusoidal *S* wave travelling in the opposite direction.

The homogeneous material parameters are set to

$$\lambda = 2, \quad \mu = 1, \quad \rho = 1, \quad (44)$$

throughout the computational domain Ω leading to the constant wave propagation velocities

$$c_p = 2, \quad c_s = 1, \quad (45)$$

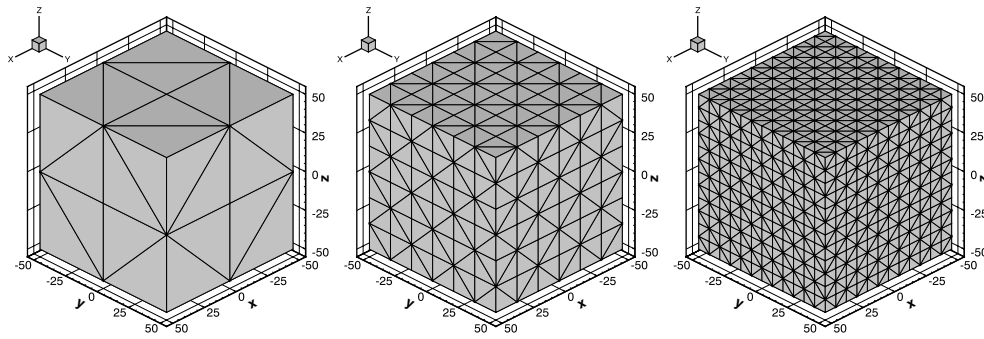


Figure 2. Sequence of discretizations of the computational domain Ω via regularly refined tetrahedral meshes, which are used for the numerical convergence analysis.

for the P and S wave, respectively. The total simulation time T is set to $T = 100\sqrt{3}$, such that the P and S waves at simulation time $t = T$ coincide with the initial condition at time $t = 0$. This way the exact reference solution is given by the initial condition in eq. (42), that is,

$$Q_p(\mathbf{x}, T) = Q_p(\mathbf{x}, 0). \tag{46}$$

The stability of our explicit time stepping scheme is controlled by the CFL number, introduced by Courant (Courant *et al.* 1928). The CFL number is set in all computations to 50 per cent of the stability limit $\frac{1}{2N+1}$ of Runge–Kutta DG schemes. For a thorough investigation of the linear stability properties of the ADER–DG schemes via a von Neumann analysis, see Dumbser (2005).

The numerical analysis to determine the convergence orders is performed on a sequence of tetrahedral meshes as shown in Fig. 2. The mesh sequence is obtained by dividing the computational domain Ω into a number of subcubes, which are then subdivided into five tetrahedrons as shown in Fig. 2. We remark that this subdivision leads to four equal tetrahedrons with one-sixth of the cube’s volume and one regular central tetrahedron of one-third of the cube’s volume. This way, the refinement is controlled by changing the number of subcubes in each space dimension.

We arbitrarily pick one of the variables of the system of the elastic wave eq. (1) to numerically determine the convergence order of the used ADER–DG schemes. In Table 3, we show the errors for the shear stress component σ_{yz} . The errors of the numerical solution Q_h with respect to the exact solution Q_e is measured in the L^∞ -norm and the continuous L^2 -norm

$$E_{L^\nu}^s = \|Q_{h^s} - Q_e\|_{L^\nu(\Omega)} = \left(\int_{\Omega} |Q_{h^s} - Q_e|^\nu dV \right)^{1/\nu}, \quad \text{with } \nu = 2, \infty, \tag{47}$$

Table 3. Convergence rates of σ_{yz} of the ADER–DG $\mathcal{O}2$ up to ADER–DG $\mathcal{O}7$ schemes on tetrahedral meshes.

h	E_{L^∞}	\mathcal{O}_{L^∞}	E_{L^2}	\mathcal{O}_{L^2}	N_d	I	CPU (s)
5,41	5.0420×10^{-2}	–	$2.8919 \times 10^{+1}$	–	81 920	790	1451
3,61	2.1217×10^{-2}	2.1	$1.2415 \times 10^{+1}$	2.1	276 480	1190	7327
2,71	1.1549×10^{-2}	2.1	6.8847×10^0	2.0	655 360	1580	23 171
2,17	7.2733×10^{-3}	2.1	4.3761×10^0	2.0	1280 000	1970	55 973
10,83	2.4353×10^{-2}	–	$1.0173 \times 10^{+1}$	–	25 600	660	277
5,41	3.4247×10^{-3}	2.8	1.0776×10^0	3.2	204 800	1320	4405
3,61	1.0139×10^{-3}	3.0	3.0678×10^{-1}	3.1	691 200	1970	22 175
2,71	4.4319×10^{-4}	2.9	1.2769×10^{-1}	3.0	1638 400	2630	69 971
10,83	3.8223×10^{-3}	–	6.0726×10^{-1}	–	51200	920	1014
7,22	7.0762×10^{-4}	4.1	1.2019×10^{-1}	4.0	172 800	1380	5137
5,41	2.5817×10^{-4}	3.5	3.8192×10^{-2}	4.0	409 600	1840	16 057
3,61	5.2253×10^{-5}	4.0	7.5876×10^{-3}	4.0	1382400	2760	81583
21,65	1.0794×10^{-2}	–	1.4660×10^0	–	11 200	600	233
10,83	4.7161×10^{-4}	4.5	4.0304×10^{-2}	5.2	89 600	1190	3669
7,22	6.5100×10^{-5}	4.9	5.3699×10^{-3}	5.0	302 400	1780	18 708
5,41	1.5694×10^{-5}	4.9	1.2783×10^{-3}	5.0	716800	2370	58 051
43,30	8.0163×10^{-2}	–	$1.2829 \times 10^{+1}$	–	2240	370	51
21,65	8.2725×10^{-4}	6.6	1.2160×10^{-1}	6.7	17 920	730	784
10,83	4.1418×10^{-5}	4.3	2.4132×10^{-3}	5.7	143 360	1450	12 331
7,22	3.4413×10^{-6}	6.1	2.1408×10^{-4}	6.0	483 840	2170	61 103
43,30	3.4083×10^{-2}	–	2.6723×10^0	–	3360	430	160
21,65	3.3679×10^{-4}	6.7	2.0206×10^{-2}	7.0	26 880	860	2578
14,43	2.1823×10^{-5}	6.7	9.3183×10^{-4}	7.6	90 720	1280	12 797
10,83	3.1332×10^{-6}	6.7	1.2627×10^{-4}	6.9	215 040	1710	39 648

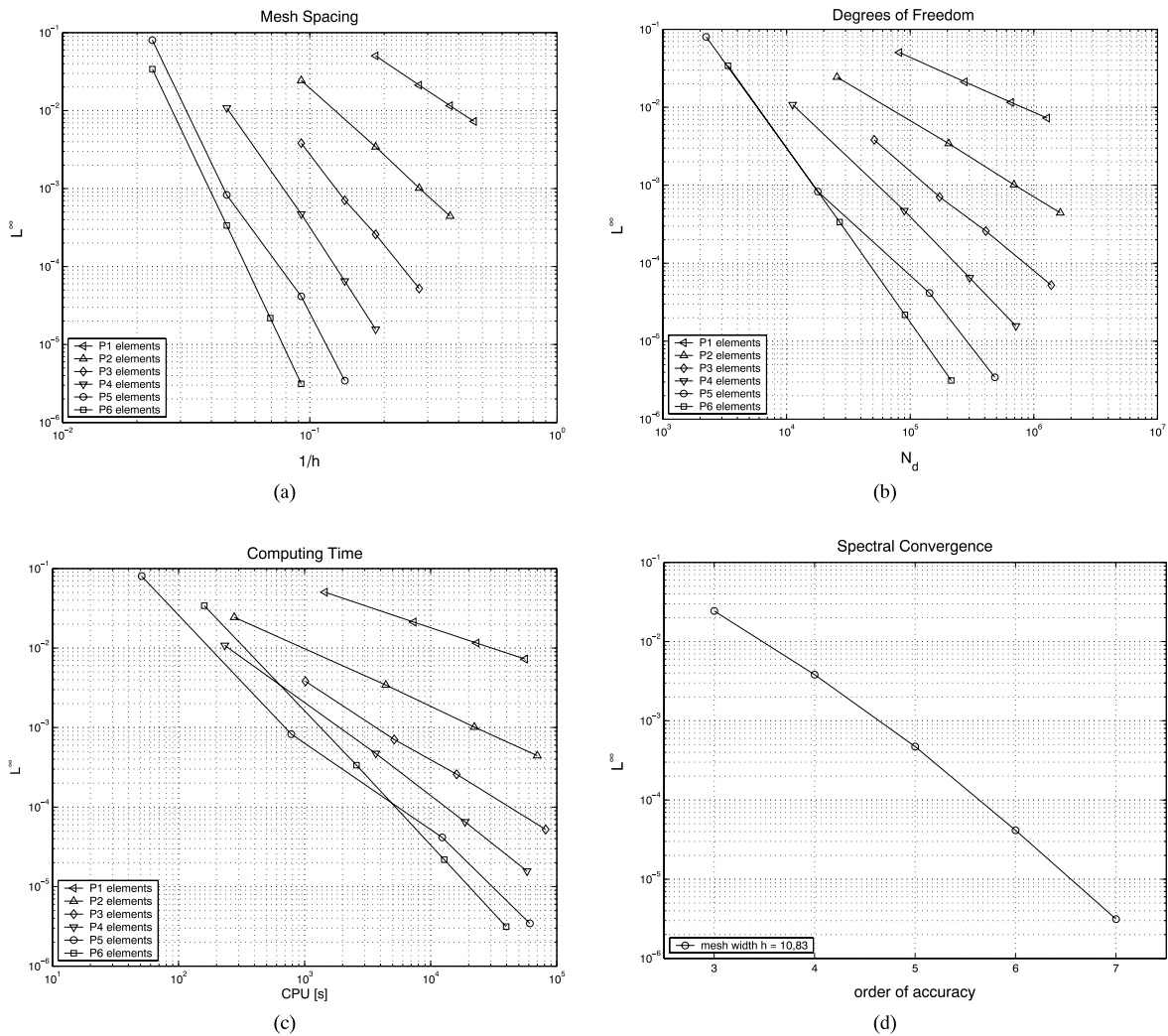


Figure 3. Convergence results of σ_{yz} of Table 3. The L^∞ error is plotted versus (a) the mesh spacing h , (b) the number of degrees of freedom N_d and (c) the CPU time. The symbols $P1$ to $P6$ denote the degree of the approximation polynomials, for example, $P1$ polynomials lead to a scheme of order 2, etc. (d) Plotting the L^∞ error versus the order of accuracy of the ADER–DG schemes for a fixed mesh clearly shows its spectral convergence.

where the integration is approximated by Gaussian integration using twice the order of accuracy as in the numerical scheme. The L^∞ -norm is approximated by the maximum error arising at any of these Gaussian integration points. The convergence orders are then computed through

$$\mathcal{O}_{L^\nu} = \frac{\log(E_{L^\nu}^s/E_{L^\nu}^{s-1})}{\log(h^s/h^{s-1})}, \quad \text{with } \nu = 2, \infty, \tag{48}$$

where h^s indicates the mesh spacing h of mesh number s in the sequence of meshes. The first column in Table 3 shows the mesh spacing h , represented by the maximum diameter of the circumscribed spheres of the tetrahedrons. The following four columns show the L^∞ and L^2 errors with the corresponding convergence orders \mathcal{O}_{L^∞} and \mathcal{O}_{L^2} determined by successively refined meshes. Furthermore, we present the total number N_d of degrees of freedom, which is a measure of required storage space during runtime and is given by the product of the number of total mesh elements and the number N_e of degrees of freedom per element. N_e depends on the order of the scheme, that is, the degree N of the polynomial basis functions in (9) via $N_e(N) = \frac{1}{6}(N+1)(N+2)(N+3)$. In the last two columns, we give the number I of iterations and the CPU times in seconds needed to reach the simulation time $T = 100\sqrt{3}$ on a Pentium Xeon processor with 3.6 GHz and 4 GB of RAM.

In Fig. 3, we visualize the convergence results of Table 3 to demonstrate the dependence of the L^∞ error with respect to (a) mesh width h , (b) number of degrees of freedom N_d and (c) CPU. In these three plots, we show clearly that higher-order methods in space and time, that is, methods with elements of higher approximation order, pay off if very high accuracy on tetrahedral meshes is desired. If the order of the schemes is increased, the errors decrease much faster with mesh refinement, see Fig. 3(a). Similarly, for a desired error the total number of degrees of freedom in Fig. 3(b), which is directly proportional to the required computer storage space, decreases for higher-order schemes. In other words, for a given storage space higher accuracy can be reached with higher-order schemes. In Fig. 3(c), we demonstrate that the CPU-time needed to reach a very high accuracy also decreases when using higher-order schemes. Finally, in Fig. 3(d), we show the *spectral* convergence of the ADER–DG schemes on tetrahedral meshes due to the choice of the basis functions Φ_I in eq. (9). Based on the results of

the convergence analysis we claim that ADER–DG schemes represent a new class of numerical methods, which are well suited for solving the elastic wave equations with very high accuracy. We also note that these results extend to any other linear hyperbolic systems, such as the linearized Euler equations, as shown in Dumbser (2005) and Dumbser & Munz (2005b).

Furthermore, the new highly accurate method can handle 3-D, unstructured tetrahedral meshes and therefore provides an enormous potential in applications with very complex 3-D geometry. In Section 6, we demonstrate the performance of the proposed schemes applied to a well-acknowledged test case in three space dimensions.

Remarks on CPU time. In the following, we would like to comment on CPU requirements of the proposed ADER–DG method in a more detailed manner. We compare our fourth-order ADER–DG scheme with a standard fourth order non-staggered FD scheme solving eq. (1) and using classical fourth-order Runge–Kutta time stepping. The FD scheme is implemented in such a way that the data structure is quite similar to the one of the ADER–DG code. We now solve the same test problem as described in this section on convergence analysis using a Cartesian mesh for the FD scheme and tetrahedral meshes according to Fig. 2 for the ADER–DG scheme. We obtain the following result: fourth-order ADER–DG needs about 2.3×10^{-4} s per element and time step, whereas fourth-order FD only needs about 7.0×10^{-6} s per point and time step on the same machine and for the same test case. Both codes were run in their serial versions on one Pentium Xeon processor with 3.6 GHz and 4 GB of RAM. This means that one time step per element using the FD method is about 30–40 times faster than a corresponding element update of ADER–DG. Please note, however, that the time step for DG schemes in general decreases with the degree of the basis polynomials N as $1/(2N + 1)$, which leads to a further disadvantage of ADER–DG with respect to CPU cost. However, using the same number of ADER–DG elements as FD points, the ADER–DG method is about 40–50 times more accurate. For a fair comparison, one should therefore compare the methods at the *same precision*. We recalculate the convergence test with the fourth-order FD scheme in order to reach an L^2 error of $E_{L^2} = 0.1$. This requires an FD grid that is about 4.5 times finer per space dimension than the corresponding ADER–DG mesh. The final result is that, for the same precision level, the fourth-order FD method is about *five times faster* than a fourth-order ADER–DG scheme. This relative speed advantage of FD further reduces for higher-order methods. For example, running the same benchmark with a sixth-order ADER–DG scheme against the fourth-order FD scheme results in a small CPU advantage of 8 per cent for the sixth-order ADER–DG method. We remark that for all these benchmarks the FD scheme was run on a purely Cartesian grid, whereas the ADER–DG method was run on unstructured tetrahedral meshes as shown in Fig. 2. Clearly, the use of an unstructured mesh is not optimal for such a simple benchmark test. Unstructured grids can only play their full advantage in highly complex geometries, where it quickly may even become impossible to generate an adequate structured grid for standard FD schemes. One should furthermore be aware that such CPU time-comparisons depend on many factors, such as implementation, compiler settings, hardware architecture and so on. We especially underline that the ADER–DG method is algorithmically quite complex which makes it more difficult to be optimized automatically by the compiler. However, as a final remark, we would like to emphasize that the ADER–DG scheme is a one-step method, which means that time-integration does not need any intermediate stages as, for example, the classical Runge–Kutta time stepping schemes. Therefore, there is no need of intermediate MPI communications for intermediate stages, which is a clear benefit for a massively parallel implementation of ADER–DG on modern supercomputers.

6 APPLICATION EXAMPLE

We apply the proposed ADER–DG method on a well-defined 3-D test problem, which was published in the final report of the *LIFELINES PROGRAM TASK 1A01* (Day 2001) of the Pacific Earthquake Engineering Research Center. The test case is part of a multi-institutional code validation project of a series of different numerical methods employed in numerical modelling of earthquake ground motion in 3-D earth models. Therefore, besides a quasi-analytic solution, simulation results from five different well-established codes exist and serve as additional reference solutions. The results of these five codes are denoted by four-character abbreviations indicating the respective institutions:

UCBL (Doug Dreger and Shawn Larsen, University of California, Berkeley/Lawrence Livermore National Laboratory),

UCSB (Kim Olsen, University of California, Santa Barbara),

WCC1 (Robert Graves, URS Corporation),

WCC2 (Arben Pitarka, URS Corporation), and

CMUN (Jacobo Bielak, Carnegie-Mellon University).

The first four codes use Finite Differences of uniform, structured grids with staggered locations of the velocity and stress components and fourth-order accuracy in space. However, we do not have detailed information about their differences as far as their particular formulation or implementation is concerned. The CMUN code uses piecewise linear interpolation on unstructured tetrahedral Finite Elements.

The quasi-analytic solution of the problem is computed by the reflectivity method and is compared to all numerical solutions to evaluate their accuracy. The setup of the test problem LOH.1 (Layer Over Halfspace) is shown in Fig. 4(a), where for clarity only one of four symmetrical quarters of the complete computational domain $\Omega = [-15000 \text{ m}, 15000 \text{ m}] \times [-15000 \text{ m}, 15000 \text{ m}] \times [0 \text{ m}, 17000 \text{ m}]$ is plotted. The material parameters of the layer (Medium 1) of the top 1000 m and the halfspace (Medium 2) are given in Table 4.

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

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

$$S^T(t) = \frac{t}{T^2} \exp\left(-\frac{t}{T}\right), \quad (49)$$

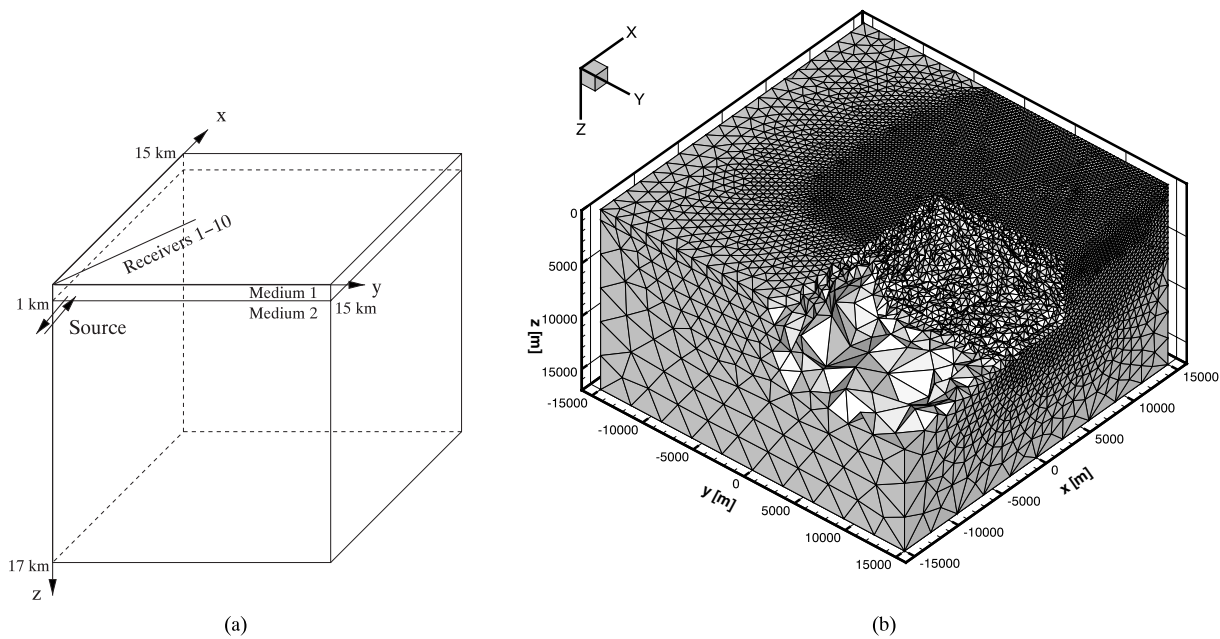


Figure 4. (a) One of four symmetric quarters is shown for the LOH.1 test case, where a layer of 1 km (Material 1) is lying on top of another layer (Material 2). The source is a point dislocation at 2000 km depth represented by a moment tensor with the only non-zero components $M_{xy} = M_{yx}$. (b) Cut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

Table 4. Material parameters for the LOH.1 test case.

	c_p (m s ⁻¹)	c_s (m s ⁻¹)	ρ (kg m ⁻³)	λ (GPa)	μ (GPa)
Medium 1	4000	2000	2600	20.8	10.4
Medium 2	6000	3464	2700	32.4	32.4

where the smoothness parameter T , controlling the frequency content and amplitude of the source time function, is set to $T = 0.1$ s. We remark that details of the discretization of external source terms in the framework of ADER–DG methods are outlined in previous work (Käser & Dumbser 2005).

The signals are recorded up to a simulation time of 9 s by 10 receivers on the free surface, as indicated in Fig. 4(a). The receiver locations $(x_i, y_i, z_i) = (i600 \text{ m}, i800 \text{ m}, 0 \text{ m})$, for $i = 1, \dots, 10$.

The computational domain Ω is discretized by an unstructured, tetrahedral mesh as shown in Fig. 4(b) using 249 338 elements. Furthermore, the mesh is generated in a problem-adapted manner. To this end, in the zone of interest the waves travelling from the source to the receivers pass through tetrahedral elements with an average edge length of 350 m, whereas in other zones the mesh is coarsened up to average edge lengths of 3000 m to reduce the number of total elements and therefore computational cost. We remark that neither the source location nor the receiver locations have to coincide with nodes of the tetrahedral mesh, as in the ADER–DG framework the numerical solution is represented by polynomials within each element and therefore can be evaluated at any position within an element (see (Käser & Dumbser 2005)). This greatly simplifies the process of mesh generation and does not restrict the desired flexibility provided by unstructured meshes. However, the mesh respects the material interface between Medium 1 and Medium 2 as the faces of the tetrahedral elements are aligned with the material interface as shown in Figs 4(a) and (b).

In the following, we present the comparison of our results obtained by a ADER–DG $\mathcal{O}4$ and ADER–DG $\mathcal{O}5$ scheme and the best four results of the reference codes (UCBL, UCSB, WCC2 and CMUN) against the analytical solution. Analogous to the LOH.1 test case in the *LIFELINES PROGRAM TASK 1A01*, the visual comparisons in Fig. 5 show the radial, transversal and vertical components of the seismic velocity field recorded at receiver 10 at $(x_{10}, y_{10}, z_{10}) = (6000 \text{ m}, 8000 \text{ m}, 0 \text{ m})$. Additionally, each plot gives the relative seismogram misfit

$$E = \sum_{j=1}^{n_t} (s_j - s_j^a)^2 / \sum_{j=1}^{n_t} (s_j^a)^2, \quad (50)$$

where n_t is number of time-samples of the seismogram, s_j is the numerical value of the particular seismogram at sample j and s_j^a is the corresponding analytical value. 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 (2001).

As shown in Figs 5(a)–(f), all numerical solutions (thin line) reproduce the analytical solution (thick line) with various discrepancies. However, the four reference solutions shown in Figs 5(a)–(d) produce incorrect oscillations, possibly due to dispersion errors, especially on the transverse component between 4.5 and 6 s. These errors are strongly reduced by the fourth-order ADER–DG $\mathcal{O}4$ scheme and even

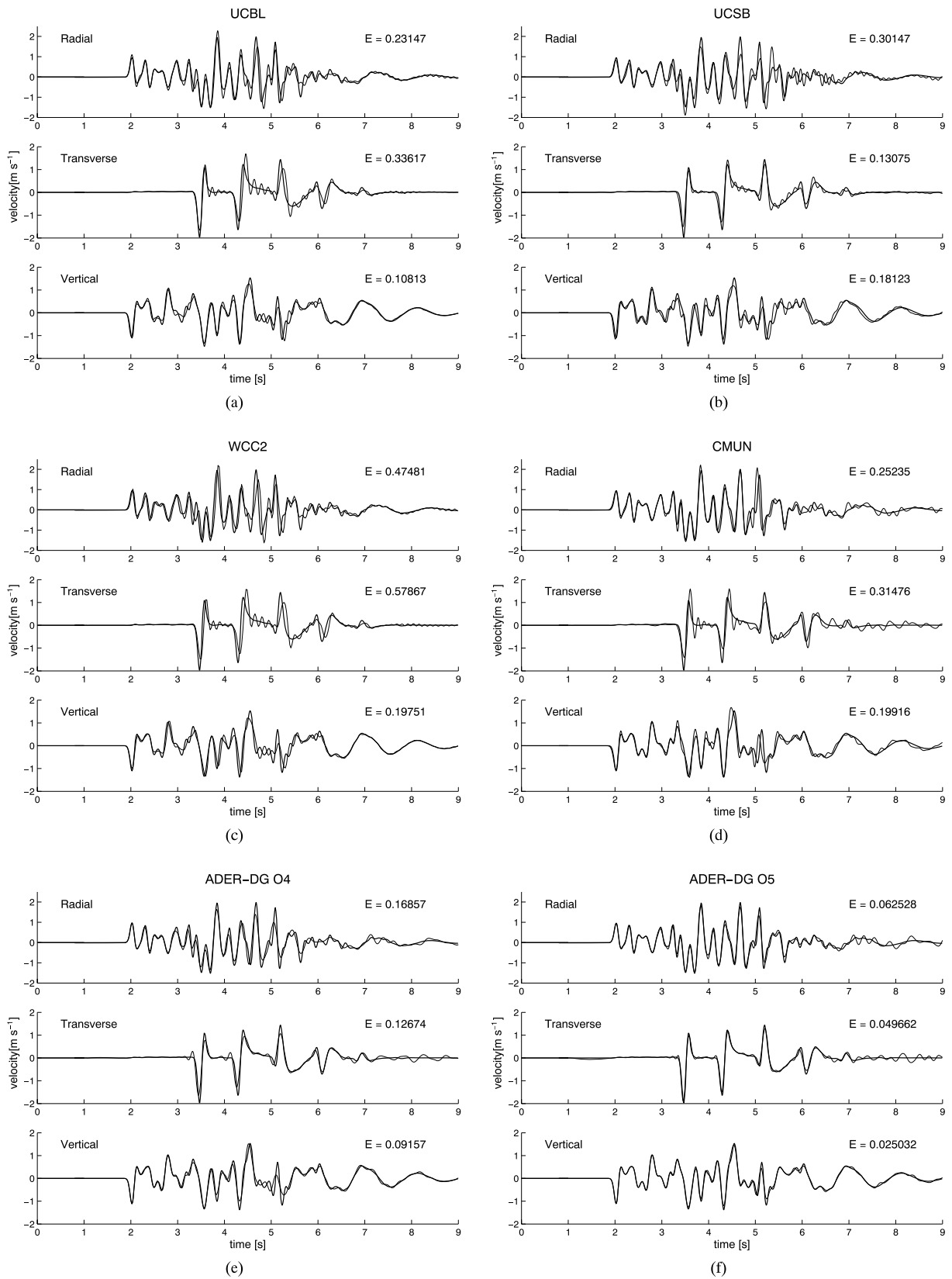


Figure 5. Comparison of the radial, transverse and vertical velocity components for the LOH.1 test case on receiver 10. The analytical solution (thick line) is plotted against the numerical one (thin line) obtained by (a) UCBL, (b) UCSB, (c) WCC2, (d) CMUN, (e) ADER-DG O4, and (f) ADER-DG O5. The relative seismogram misfit E from eq. (50) is given for each trace.

more by ADER–DG $\mathcal{O}5$. Furthermore, overshoots and phase errors obtained by all reference codes and mainly responsible for the relative seismogram misfit in eq. (50), are decreased by the ADER–DG methods as shown in Figs 5(e) and (f), which leads to more precise match of the numerical and analytical solutions. The oscillations that are visible in the seismograms after 6.5 s are due to boundary effects as receiver 10 is already receiving reflections from the model boundary. The high-order ADER–DG schemes are retaining these reflected signals and do not lose them, for example, due to numerical diffusion. We remark that we do not use any sophisticated absorbing boundaries, but simply the boundary conditions based on the flux given in eq. (40). The detailed investigation of the treatment of absorbing, non-reflecting boundaries is still a pending task.

Considering the computational storage requirements, we point out that the FD reference codes were using a regular grid of mesh width 100 m leading to 15.3×10^6 grid points to discretize the computational domain Ω . In our case, the unstructured tetrahedral mesh with 249 338 elements is much coarser. Using fourth-order accuracy, that is, the ADER–DG $\mathcal{O}4$ scheme, leads to 20 degrees of freedom per element and therefore results in slightly less than 5.0×10^6 degrees of freedom in total. With ADER–DG $\mathcal{O}5$, we have 35 degrees of freedom per element and 8.7×10^6 in total, but we still remain way below the number of grid points used in the regular FD meshes. However, as demonstrated by the seismograms in Figs 5(a)–(f) the results are superior even though we use such a coarse, unstructured tetrahedral mesh.

Considering the CPU-times a comparison turned out to be difficult, as all reference codes have been run on different machines with different levels of parallelization and no CPU-time data were available from the final report of the *LIFELINES PROGRAM TASK 1A01* (Day 2001) of the Pacific Earthquake Engineering Research Center. However, to give a rough indication of the CPU-time requirements for our ADER–DG $\mathcal{O}5$ simulation, the run time was approximately 31 hr on 64 Intel Xeon EM64T 3.2-GHz processors each with 0.5 GB of RAM.

7 CONCLUSION

In this paper, we presented the extension of the new ADER–DG approach to numerically solve the heterogeneous elastic wave equations on unstructured 3-D tetrahedral meshes. The mathematical formulation of the method is outlined in detail and results of a numerical convergence analysis are demonstrated. To the knowledge of the authors, the proposed ADER–DG method is the first method, that achieves arbitrary high-order accuracy in space and time on tetrahedral meshes. Therefore, it provides a unique numerical scheme to simulate elastic wave propagation phenomena with high approximation order even for problems with complex geometrical features, where the flexibility of tetrahedral meshes is highly beneficial. As the data quality and geometrical complexity of subsurface models are continuously increasing, it is necessary to simulate wave propagation in such media with highly accurate numerical schemes able to handle flexible unstructured meshes. Only this way, the numerically introduced errors can be decreased to a sufficient extent and the full information of synthetic seismograms can be interpreted safely and attributed to geometrical features or the geophysical material distribution in the model. In this respect, the new ADER–DG method on unstructured tetrahedral meshes can play an important role for understanding the details of wave propagation effects in geometrically complex, heterogeneous media. Due to the subcell resolution of high-order polynomials, extremely coarse meshes can be used without abandoning accuracy. In particular, the increase of the mesh spacing in three space dimensions leads to a drastic reduction of necessary mesh elements and therefore to less computer storage requirements. Furthermore, coarse meshes allow for larger time steps and enhance computational efficiency. In our opinion, the proposed method has a great potential for handling extremely complex geometries as element interfaces can nicely be aligned to free-surface topography or internal material discontinuities. However, this will mainly depend on advances in automatic mesh-generation algorithms, where tetrahedral meshes seem to be advantageous. Therefore, with the ongoing developments in tetrahedral mesh generation and the rapidly growing computer power, the new ADER–DG approach will automatically provide a method capable of reaching machine precision by increasing the order of the scheme, as it is not restricted by theoretical barriers.

ACKNOWLEDGMENTS

The authors thank the DFG (Deutsche Forschungsgemeinschaft), as this work was supported through the Emmy Noether-program (KA 2281/1-1) and the DFG-CNRS research group FOR 508, Noise Generation in Turbulent Flows. The support and comments by Steven Day to set up the LOH.1 test case and providing the analytical and reference solutions are highly acknowledged.

REFERENCES

- Aki, K. & Richards, P.G., 2002. *Quantitative Seismology*, University Science Books, Sausalito, California.
- Atkins, H. & Shu, C.W., 1998. Quadrature-free implementation of the Discontinuous Galerkin method for hyperbolic equations, *AIAA J.*, **36**, 775–782.
- Bedford, A. & Drumheller, D.S., 1994. *Elastic Wave Propagation*, John Wiley & Sons, Chichester.
- Bérenger, 1994. A Perfectly Matched Layer for the absorption of electromagnetic waves, *J. Comput. Phys.*, **114**, 185–200.
- Butcher, J.C., 1987. *The Numerical Analysis of Ordinary Differential Equations: Runge-Kutta and General Linear Methods*, John Wiley & Sons, New York.
- Carcione, J.M., 1994. The wave equation in generalised coordinates, *Geophysics*, **59**, 1911–1919.
- Cockburn, B., Karniadakis, G.E. & Shu, C.W., 2000. *Discontinuous Galerkin Methods, Theory, Computation and Applications*, LNCSE, **11**, Springer, Berlin, Germany.
- Collino, F. & Tsogka, C., 2001. Application of the PML absorbing layer model to the linear elastodynamic problem in anisotropic heterogeneous media, *Geophysics*, **66**, 294–307.

- Courant, R., Friedrichs, K.O. & Lewy, H., 1928. Über die partiellen Differenzialgleichungen der mathematischen Physik, *Mathematische Annalen*, **100**, 32–74.
- Day, S.M., 2001. Tests of 3D Elastodynamics Codes, LIFELINES PROGRAM TASK 1A01 *Final Report to Pacific Earthquake Engineering Research Center*, September 10, 2001. Coordinated by S.M. Day, San Diego State University.
- Dumbser, M., 2003. On the Improvement of Efficiency and Storage Requirements of the Discontinuous Galerkin Method for Aeroacoustics, *PAMM*, **3**, 426–427.
- Dumbser, M., 2005. *Arbitrary High Order Schemes for the Solution of Hyperbolic Conservation Laws in Complex Domains*, Shaker Verlag, Aachen.
- Dumbser, M. & Munz, C.D., 2005a. Arbitrary High Order Discontinuous Galerkin Schemes, in *Numerical Methods for Hyperbolic and Kinetic Problems*, eds Cordier, S., Goudon, T., Gutnic, M. & Sonnendrucker, E., IRMA series in mathematics and theoretical physics, EMS Publishing House, Zurich, Switzerland, 295–333.
- Dumbser, M. & Munz, C.D., 2005b. ADER Discontinuous Galerkin Schemes for Aeroacoustics, in *Comptes Rendus Mécanique*, **333**, 683–687.
- Dumbser, M., Schwartzkopff, T. & Munz, C.D., 2006. Arbitrary High Order Finite Volume Schemes for Linear Wave Propagation, in *Computational Science and High Performance Computing II*, eds Krause, E., Shokin, Y., Resch, M., Shokina, N., Notes on Numerical Fluid Mechanics and Multidisciplinary Design (NNFM), Springer Berlin/Heidelberg, 129–144.
- Fornberg, B., 1996. *A Practical Guide to Pseudospectral Methods*, Cambridge University Press, Cambridge, UK.
- Hughes, T.J.R., 1987. *The Finite Element Method—Linear Static and Dynamic Finite Element Analysis*, Prentice Hall, Englewood Cliffs, NJ.
- Igel, H., 1999. Wave propagation in three-dimensional spherical sections by the Chebyshev spectral method, *Geophys. J. Int.*, **136**, 559–566.
- Igel, H., Mora, P. & Rioulet, B., 1995. Anisotropic wave propagation through finite-difference grids, *Geophysics*, **60**, 1203–1216.
- Käser, M. & Dumbser, M., 2005. An Arbitrary High Order Discontinuous Galerkin Method for Elastic Waves on Unstructured Meshes I: The Two-Dimensional Isotropic Case with External Source Terms, *Geophys. J. Int.*
- Käser, M. & Igel, H., 2001. Numerical simulation of 2D wave propagation on unstructured grids using explicit differential operators, *Geophys. Prospect.*, **49**, 607–619.
- Käser, M. & Igel, H., 2001. A comparative study of explicit differential operators on arbitrary grids, *J. Comput. Acoustics*, **9**, 1111–1125.
- Komatitsch, D. & Tromp, J., 1999. Introduction to the spectral-element method for 3-D seismic wave propagation, *Geophys. J. Int.*, **139**, 806–822.
- Komatitsch, D. & Tromp, J., 2002. Spectral-element simulations of global seismic wave propagation – I. Validation, *Geophys. J. Int.*, **149**, 390–412.
- Komatitsch, D. & Tromp, J., 2003. A perfectly matched layer absorbing boundary condition for the second-order seismic wave equation, *Geophys. J. Int.*, **154**, 146–153.
- Komatitsch, D. & Vilotte, J.P., 1998. The spectral-element method: an efficient tool to simulate the seismic response of 2D and 3D geological structures, *Bull. seism. Soc. Am.*, **88**, 368–392.
- Levander, A.R., 1988. Fourth-order finite difference P-SV seismograms, *Geophysics*, **53**, 1425–1436.
- LeVeque, R.L., 2002. *Finite Volume Methods for Hyperbolic Problems*, Cambridge University Press, Cambridge, UK.
- Madariaga, R., 1976. Dynamics of an expanding circular fault, *Bull. Seism. Soc. Am.*, **65**, 163–182.
- Magnier, S.A., Mora, P. & Tarantola, A., 1994. Finite Differences on minimal Grids, *Geophysics*, **59**, 1435–1443.
- Moczo, P., Kristek, J., Vavrycuk, V., Archuleta, R.J. & Halada, L., 2002. 3D heterogeneous staggered-grid finite-difference modeling of seismic motion with volume harmonic and arithmetic averaging of elastic moduli and densities, *Bull. seism. Soc. Am.*, **92**, 3042–3066.
- Mora, P., 1989. Modeling anisotropic seismic waves in 3-D, *59th Ann. Int. Mtg Exploration Geophysicists, expanded abstracts*, 1039–1043.
- Patera, A.T., 1984. A spectral-element method for fluid dynamics: laminar flow in a channel expansion, *J. Comput. Phys.*, **144**, 45–58.
- Priolo, E., Carcione, J.M. & Seriani, G., 1994. Numerical simulation of interface waves by high-order spectral modeling techniques, *J. acoust. Soc. Am.*, **95**, 681–693.
- Schwartzkopff, T., Munz, C.D. & Toro, E.F., 2002. ADER: A High-Order Approach for Linear Hyperbolic Systems in 2D, *J. Sci. Comput.*, **17**, 231–240.
- Schwartzkopff, T., Dumbser, M. & Munz, C.D., 2004. Fast high order ADER schemes for linear hyperbolic equations, *J. Comput. Phys.*, **197**, 532–539.
- Seriani, G., 1998. 3-D large-scale wave propagation modeling by a spectral-element method on a Cray T3E multiprocessor, *Comput. Methods Appl. Mech. Eng.*, **164**, 235–247.
- Tessmer, E., 1995. 3-D Seismic modelling of general material anisotropy in the presence of the free surface by a Chebyshev spectral method, *Geophys. J. Int.*, **121**, 557–575.
- Tessmer, E. & Kosloff, D., 1994. 3-D Elastic modelling with surface topography by a Chebyshev spectral method, *Geophysics*, **59**, 464–473.
- Titarev, V.A. & Toro, E.F., 2002. ADER: Arbitrary high order Godunov approach, *J. Sci. Comput.*, **17**, 609–618.
- Toro, E.F., 1999. *Riemann Solvers and Numerical Methods for Fluid Dynamics*, Springer, Berlin.
- Toro, E.F. & Titarev, V.A., 2002. Solution of the generalized Riemann problem for advection-reaction equations, *Proc. Roy. Soc. London*, **458**, 271–281.
- Toro, E.F., Millington, A.C. & Nejad, L.A., 2001. Towards very high order Godunov schemes, in *Godunov methods; Theory and applications*, Kluwer Academic Plenum Publishers, Oxford, 907–940.
- Virieux, J., 1984. SH-wave propagation in heterogeneous media: Velocity-stress finite-difference method, *Geophysics*, **49**, 1933–1942.
- Virieux, J., 1986. P-SV wave propagation in heterogeneous media: Velocity-stress finite-difference method, *Geophysics*, **51**, 889–901.

APPENDIX A: COORDINATE TRANSFORMATION OF TETRAHEDRONS

The coordinate transformation of a tetrahedron in the global, Cartesian xyz -coordinate system into the $\xi\eta\zeta$ -reference system as shown in Fig. 1 is defined by

$$\xi = \frac{1}{|\mathbf{J}|} \{x_1(y_4z_3 - y_3z_4) + x_3(y_1z_4 - y_4z_1) + x_4(y_3z_1 - y_1z_3) \\ + [y_1(z_3 - z_4) + y_3(z_4 - z_1) + y_4(z_1 - z_3)]x \\ + [x_1(z_4 - z_3) + x_3(z_1 - z_4) + x_4(z_3 - z_1)]y \\ + [x_1(y_3 - y_4) + x_3(y_4 - y_1) + x_4(y_1 - y_3)]z\},$$

$$\begin{aligned}
\eta &= \frac{1}{|\mathbf{J}|} \{y_1(x_4z_2 - x_2z_4) + y_2(x_1z_4 - x_4z_1) + y_4(x_2z_1 - x_1z_2) \\
&\quad + [y_1(z_4 - z_2) + y_2(z_1 - z_4) + y_4(z_2 - z_1)]x \\
&\quad + [x_1(z_2 - z_4) + x_2(z_4 - z_1) + x_4(z_1 - z_2)]y \\
&\quad + [x_1(y_4 - y_2) + x_2(y_1 - y_4) + x_4(y_2 - y_1)]z\}, \\
\zeta &= \frac{1}{|\mathbf{J}|} \{z_1(x_3y_2 - x_2y_3) + z_2(x_1y_3 - x_3y_1) + z_3(x_2y_1 - x_1y_2) \\
&\quad + (y_1(z_2 - z_3) + y_2(z_3 - z_1) + y_3(z_1 - z_2))x \\
&\quad + (x_1(z_3 - z_2) + x_2(z_1 - z_3) + x_3(z_2 - z_1))y \\
&\quad + (x_1(y_2 - y_3) + x_2(y_3 - y_1) + x_3(y_1 - y_2))z\}, \tag{A1}
\end{aligned}$$

where

$$\begin{aligned}
|\mathbf{J}| &= x_1[y_2(z_4 - z_3) + y_3(z_2 - z_4) + y_4(z_3 - z_2)] \\
&\quad + x_2[y_1(z_3 - z_4) + y_3(z_4 - z_1) + y_4(z_1 - z_3)] \\
&\quad + x_3[y_1(z_4 - z_2) + y_2(z_1 - z_4) + y_4(z_2 - z_1)] \\
&\quad + x_4[y_1(z_2 - z_3) + y_2(z_3 - z_1) + y_3(z_1 - z_2)] \tag{A2}
\end{aligned}$$

is the determinant of the Jacobian matrix \mathbf{J} of the transformation being equal to six times the volume of the tetrahedron.

The back-transformation is given by

$$\begin{aligned}
x &= x_1 + (x_2 - x_1)\xi + (x_3 - x_1)\eta + (x_4 - x_1)\zeta, \\
y &= y_1 + (y_2 - y_1)\xi + (y_3 - y_1)\eta + (y_4 - y_1)\zeta, \\
z &= z_1 + (z_2 - z_1)\xi + (z_3 - z_1)\eta + (z_4 - z_1)\zeta. \tag{A3}
\end{aligned}$$

In eqs (A1), (A2) and (A3), the x_i , y_i and z_i denote the physical vertex coordinates of the tetrahedron.