

# An arbitrary high-order Discontinuous Galerkin method for elastic waves on unstructured meshes – V. Local time stepping and $p$ -adaptivity

Michael Dumbser,<sup>1,2</sup> Martin Käser<sup>1</sup> and Eleuterio F. Toro<sup>1</sup>

<sup>1</sup>Laboratory of Applied Mathematics, Department of Civil and Environmental Engineering, University of Trento, Trento, Italy.

E-mail: martin.kaeser@ing.unitn.it

<sup>2</sup>Institut für Aerodynamik und Gasdynamik, Universität Stuttgart, Stuttgart, Germany

Accepted 2007 March 5. Received 2007 January 30; in original form 2006 September 6

## SUMMARY

This article describes the extension of the arbitrary high-order Discontinuous Galerkin (ADER-DG) method to treat locally varying polynomial degree of the basis functions, so-called  $p$ -adaptivity, as well as locally varying time steps that may be different from one element to another. The  $p$ -adaptive version of the scheme is useful in complex 3-D models with small-scale features which have to be meshed with reasonably small elements to capture the necessary geometrical details of interest. Using a constant high polynomial degree of the basis functions in the whole computational domain can lead to an unreasonably high CPU effort since good spatial resolution at the surface may be already obtained by the fine mesh. Therefore, it can be more adequate in some cases to use a lower order method in the small elements to reduce the CPU effort without losing much accuracy. To further increase computational efficiency, we present a new local time stepping (LTS) algorithm. For usual explicit time stepping schemes the element with the smallest time step resulting from the stability criterion of the method will dictate its time step to all the other elements of the computational domain. In contrast, by using local time stepping, each element can use its optimal time step given by the local stability condition. Our proposed LTS algorithm for ADER-DG is very general and does not need any temporal synchronization between the elements. Due to the ADER approach, accurate time interpolation is automatically provided at the element interfaces such that the computational overhead is very small and such that the method maintains the uniform high order of accuracy in space and time as in the usual ADER-DG schemes with a globally constant time step. However, the LTS ADER-DG method is computationally much more efficient for problems with strongly varying element size or material parameters since it allows to reduce the total number of element updates considerably. This holds especially for unstructured tetrahedral meshes that contain strongly degenerate elements, so-called slivers. We show numerical convergence results and CPU times for LTS ADER-DG schemes up to sixth order in space and time on irregular tetrahedral meshes containing elements of very different size and also on tetrahedral meshes containing slivers. Further validation of the algorithm is provided by results obtained for the layer over half-space (LOH.1) benchmark problem proposed by the Pacific Earthquake Engineering Research Center. Finally, we present a realistic application on earthquake modelling and ground motion prediction for the alpine valley of Grenoble.

**Key words:** ADER approach, Discontinuous Galerkin schemes, elastic waves, local time stepping,  $p$ -adaptivity, tetrahedral meshes.

## 1 INTRODUCTION

Large-scale applications in numerical seismology including realistic material properties and complex geometries usually still require a tremendous effort in model building, mesh generation, computer storage and CPU time. In the past, advancements in the field of mesh generation have led to automated algorithms that produce unstructured tetrahedral meshes even for very complex geometries. However, the computational domain may exhibit zones where very small mesh elements are unavoidable in order to resolve the geometrical features of interest, for example, sedimentary basins, folded and faulted reservoirs or surface topography. Additionally, in other parts of the computational domain rather large elements might be admissible.

In general, explicit high-order methods face two main disadvantages for such strongly heterogeneous meshes, in particular, if the mesh contains mostly large elements and only a few very small elements in a limited zone of interest. First, using the high order ADER Discontinuous Galerkin (ADER-DG) approach in the entire computational domain is expensive, since the high order of accuracy may not be needed everywhere. In fact, the small elements might already provide sufficient accuracy due to their fine spatial resolution of the problem. In this case, the high order ADER Finite Volume (FV) approach (Dumbser *et al.* 2006b) might be advantageous since the computational cost on a given mesh is less than for ADER-DG. Secondly, the few smallest elements in the mesh reduce the allowed time step for *all* elements of the mesh. Usually, such time step limits apply to all explicit numerical methods because of stability issues.

Furthermore, automatic unstructured mesh generation can lead to extremely small and degenerated tetrahedrons, so-called slivers (Bern & Eppstein 1992; Joe 1995; Fleischmann *et al.* 1999). A sliver is characterized by four vertices, that lie almost in a plane and, therefore, lead to a very flat tetrahedral element of a volume that is almost zero. This phenomenon is well-known in the literature on unstructured 3-D mesh generation. It often can be avoided, but not always (Cheng *et al.* 2000; Edelsbrunner & Guoy 2002). However, in the presence of such degenerate elements the quality of the numerical solution in their vicinity might be reduced and, more importantly, the time step is restricted severely for explicit numerical methods. In extreme cases, it may even become practically impossible to run a numerical scheme with a reasonable time step on a mesh containing a sliver element.

In this paper, we propose a new  $p$ -adaptive ADER-DG scheme with local time stepping (LTS) to overcome the two above mentioned problems arising for realistic models of complex geometries. The work represents a crucial extension and improvement of the ADER-DG scheme for seismic wave propagation problems as introduced in Käser & Dumbser (2006), Dumbser & Käser (2006) and Käser *et al.* (2007). Similar to the approach of Houston & Süli (2001), the proposed ADER-DG scheme allows for locally different polynomial degrees of the numerical approximation inside each element and, therefore, obtains a better balance of accuracy and computational cost.

Furthermore, the new ADER-DG method applies a LTS technique such that in each element the maximum time step can be used according to the local stability criterion. Due to the ADER time integration approach (Toro *et al.* 2001; Titarev & Toro 2002; Toro & Titarev 2002) for high order accurate space–time flux computation, the method remains a *time-accurate* one-step scheme despite the LTS. LTS in combination with DG schemes on unstructured 3-D meshes was already proposed by Flaherty *et al.* (1997). However, they only presented first order results. LTS for FV methods on unstructured tetrahedral meshes has been developed by Fumeaux *et al.* (2004). However, their approach does not go beyond second order in space and time. We remark, that extending these previous approaches to higher order in time, for example, by using a Runge-Kutta (RK) time stepping method, would be very expensive in terms of memory and CPU time due to the required storage of the intermediate RK stages. In addition, a time interpolation step is required to synchronize the RK stages between different elements. Furthermore, the efficiency of RK time discretization schemes drastically decreases, if the order of accuracy becomes greater than four, due to the so-called Butcher barriers (Butcher 1987) and the number of intermediate RK stages becomes even larger than the formal order of accuracy.

In contrast, the use of the ADER approach offers two advantages: First, it is a one-step scheme and thus no intermediate RK stages have to be stored. Secondly, the time interpolation at the element interfaces is provided naturally by the Cauchy-Kovalewski procedure and, therefore, no additional interpolation is needed. Very recently, a similar DG scheme using the Cauchy-Kovalewski procedure for space–time expansion and LTS was introduced (Loercher *et al.* 2007) for nonlinear systems in one space dimension.

The structure of this article is as follows. First, we give a short introduction to the equations governing seismic wave propagation in Section 2. Then, the  $p$ -adaptive ADER-DG scheme with LTS is described for this type of equation in Section 3. A detailed discussion of the LTS algorithm follows in Section 3.3. To demonstrate the numerical convergence properties of the LTS scheme, results on several types of meshes are shown in Section 4. Various applications of the proposed scheme are presented in Section 5. The first application deals with the standard layer over half-space (LOH.1) test case, for which analytical and numerical reference solutions are available. The second test case is a 3-D extension of the stiff inclusion test case proposed in two space dimensions by LeVeque (2002). Finally, a realistic example of earthquake modelling and ground motion prediction in the alpine valley of Grenoble is presented, where all features of the  $p$ -adaptive ADER-DG scheme with LTS on unstructured tetrahedral meshes are exploited.

## 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) which leads to the following hyperbolic system in velocity–stress formulation:

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

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

Here,  $\lambda$  and  $\mu$  are the Lamé constants and  $\rho$  is the mass density of the material. The normal stress components are given by  $\sigma_{xx}$ ,  $\sigma_{yy}$  and  $\sigma_{zz}$ , and the shear stresses are  $\sigma_{xy}$ ,  $\sigma_{yz}$  and  $\sigma_{xz}$ . 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 (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  $\vec{x} = (x, y, z)^T \in \mathbb{R}^3$ . The physical properties of the material are functions of space but are constant in time, that is,  $\lambda = \lambda(\vec{x})$ ,  $\mu = \mu(\vec{x})$  and  $\rho = \rho(\vec{x})$ , in order to describe heterogeneous material.

In the following, we write system (2) in 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  $Q_p = (\sigma_{xx}, \sigma_{yy}, \sigma_{zz}, \sigma_{xy}, \sigma_{yz}, \sigma_{xz}, u, v, w)^T$  is the vector of the unknown variables of the system. Note, that classical tensor notation is used, which implies summation over each index that appears twice. The matrices  $A_{pq} = A_{pq}(\vec{x})$ ,  $B_{pq} = B_{pq}(\vec{x})$  and  $C_{pq} = C_{pq}(\vec{x})$  are the space dependent Jacobian matrices of size  $p \times q$ , with  $p, q = 1, \dots, 9$ , and are given, for example, in Dumbser & Käser (2006). For the anelastic case including viscoelastic attenuation (see Käser *et al.* 2007), details on ADER-DG schemes for anisotropic material can be found in de la Puente *et al.* (2006).

### 3 THE ADER-DG SCHEME WITH $p$ -ADAPTIVITY AND LOCAL TIME STEPPING

For the construction of the numerical scheme, we consider the general linear hyperbolic system of equations with variable coefficients as given in (2). The computational domain  $\Omega \in \mathbb{R}^3$  is divided into conforming tetrahedral elements  $T^{(m)}$  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  $T^{(m)}$ .

#### 3.1 Semi-discrete $p$ -adaptive scheme

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

$$Q_p^{(m)}(\xi, \eta, \zeta, t) = \hat{Q}_{pl}^{(m)}(t) \Phi_l(\xi, \eta, \zeta), \tag{3}$$

where  $\xi, \eta$  and  $\zeta$  are the coordinates in a reference coordinate system. The relation between the physical coordinates  $x$ - $y$ - $z$  and the reference coordinate system  $\xi$  -  $\eta$  -  $\zeta$  is a simple linear mapping (see e.g. Dumbser & Käser 2006). This mapping transforms the physical tetrahedron  $T^{(m)}$  to a canonical reference element  $T_E$  with the four vertices  $(0, 0, 0)$ ,  $(1, 0, 0)$ ,  $(0, 1, 0)$  and  $(0, 0, 1)$ . In eq. (3) the index  $p$  stands for the number of unknowns in the vector  $Q$  and  $l$  indicates the  $l$ th basis function. We use the hierarchical orthogonal basis functions  $\Phi_k$  as given in Cockburn *et al.* (2000). Hierarchical means, that the degrees of freedom are *ordered* with respect to the polynomial degree, starting with the lowest degree (zeroth degree polynomials) up to the highest one. In three space dimensions, for an element using basis functions of maximum degree  $N$ , the index  $l$  ranges from 1 to the number of degrees of freedom  $N_d = N_d(N) = \frac{1}{6}(N+1)(N+2)(N+3)$ . For the  $p$ -adaptive version of the scheme, the local degree of the basis polynomials may vary from element to element and thus  $N$  becomes a function of the element number  $(m)$ . We therefore, write  $N = N^{(m)} = N(T^{(m)})$ . We note in particular, that due to the orthogonality of the basis functions, a lower order polynomial approximation can be obtained from a higher order polynomial approximation by simply setting the degrees of freedom corresponding to the higher polynomial degrees to zero. This will become important later when computing the fluxes between elements of different local polynomial degree.

As usual for DG schemes, the governing eq. (2) is multiplied by a test function  $\Phi_k$  and integrated over each tetrahedral element. After integration by parts, inserting the DG approximation (3) and after inserting a numerical flux into the appearing boundary integral, the

semi-discrete formulation of the scheme (see Dumbser & Käser 2006, for more details) in the reference element  $T_E$  reads as

$$\begin{aligned}
 |J| M_{kl} \cdot \frac{\partial}{\partial t} \hat{Q}_{pl}^{(m)} + \sum_{j=1}^4 T_{pq}^j \frac{1}{2} (A_{qr}^{(m)} + |A_{qr}^{(m)}|) (T_{rs}^j)^{-1} |S_j| F_{kl}^{-,j} \cdot \hat{Q}_{sl}^{(m)} \\
 + \sum_{j=1}^4 T_{pq}^j \frac{1}{2} (A_{qr}^{(m)} - |A_{qr}^{(m)}|) (T_{rs}^j)^{-1} |S_j| F_{kn}^{+,j,i,h} \cdot \hat{Q}_{sn}^{(k_j)} \\
 - A_{pq}^* |J| K_{kl}^\xi \cdot \hat{Q}_{ql}^{(m)} - B_{pq}^* |J| K_{kl}^\eta \cdot \hat{Q}_{ql}^{(m)} - C_{pq}^* |J| K_{kl}^\zeta \cdot \hat{Q}_{ql}^{(m)} = 0,
 \end{aligned} \tag{4}$$

where  $|S_j|$  denotes the area of face  $j$ . Furthermore, we have

$$A_{pq}^* = A_{pq} \frac{\partial \xi}{\partial x} + B_{pq} \frac{\partial \xi}{\partial y} + C_{pq} \frac{\partial \xi}{\partial z}, \quad B_{pq}^* = A_{pq} \frac{\partial \eta}{\partial x} + B_{pq} \frac{\partial \eta}{\partial y} + C_{pq} \frac{\partial \eta}{\partial z}, \quad C_{pq}^* = A_{pq} \frac{\partial \zeta}{\partial x} + B_{pq} \frac{\partial \zeta}{\partial y} + C_{pq} \frac{\partial \zeta}{\partial z}. \tag{5}$$

The mass matrix

$$M_{kl} = \int_{T_E} \Phi_k \Phi_l d\xi d\eta d\zeta, \tag{6}$$

the stiffness matrices

$$K_{kl}^\xi = \int_{T_E} \frac{\partial \Phi_k}{\partial \xi} \Phi_l d\xi d\eta d\zeta, \quad K_{kl}^\eta = \int_{T_E} \frac{\partial \Phi_k}{\partial \eta} \Phi_l d\xi d\eta d\zeta, \quad K_{kl}^\zeta = \int_{T_E} \frac{\partial \Phi_k}{\partial \zeta} \Phi_l d\xi d\eta d\zeta, \tag{7}$$

and the flux matrices

$$F_{kl}^{-,j} \int_{\partial(T_E)_j} \Phi_k(\vec{\xi}^{(j)}(\chi, \tau)) \Phi_l(\vec{\xi}^{(j)}(\chi, \tau)) d\chi d\tau, \quad F_{kn}^{+,j,i,h} = \int_{\partial(T_E)_j} \Phi_k(\vec{\xi}^{(j)}(\chi, \tau)) \Phi_n(\vec{\xi}^{(i)}(\vec{\chi}^{(h)}(\chi, \tau), \vec{\tau}^{(h)}(\chi, \tau))) d\chi d\tau \tag{8}$$

in (4) can be calculated analytically on the reference element once for the largest polynomial degree arising in the computational domain and then be stored (see Käser & Dumbser 2006; Dumbser & Käser 2006) for details.

We emphasize that due to the hierarchical orthogonal basis functions, eq. (4) already contains the fully  $p$ -adaptive case. One only has to pay attention for the range of the indices when computing the flux across the element faces. Indices  $k$  and  $l$  always range from 1 to the local number of degrees of freedom inside the element ( $m$ ), that is,  $1 \leq k \leq N_d[N^{(m)}]$  and  $1 \leq l \leq N_d[N^{(m)}]$ . For the flux contributions of the neighbour element ( $k_j$ ) across face  $j$ , however, the *maximum* of the polynomial degrees in element ( $m$ ) and neighbour ( $k_j$ ) has to be taken. The explanation for this is very simple: in the ADER-DG scheme, the numerical flux is based on the solution of Generalized Riemann Problems (GRP) at the element interfaces which consist in piecewise polynomial initial conditions separated by a discontinuity at the element interface (Titarev & Toro 2002). The degree of the GRP is given by the maximum polynomial degree arising on any of the two sides of the interface since lower degree polynomials are only special cases of higher degree polynomials. Therefore, the index  $n$  is inside the range  $1 \leq n \leq N_d\{\max[N^{(m)}, N^{(k_j)}]\}$ . The missing degrees of freedom in the lower order element have to be filled with zeros. This means that the flux matrices and the stiffness matrices have to be computed up to the maximum polynomial degree arising in the entire computational domain. Then, for each element, only the corresponding submatrices up to the necessary degree have to be considered. For the flux contributions of the neighbour elements, these submatrices may be rectangular when fluxes between two elements of different polynomial degree  $N$  have to be computed. For convenience, we store for each element all degrees of freedom up to the maximum polynomial degree arising in the entire computational domain, setting those degrees of freedom to zero whose degree exceed the local polynomial degree inside the element. This increases computer storage but simplifies flux computation across elements with different polynomial degree.

Finally, we would like to remark that further details on the construction of the ADER-DG scheme for the elastic wave equations in two and three space dimensions can be found in Käser & Dumbser (2006), Dumbser & Käser (2006), Käser *et al.* (2007) and de la Puente *et al.* (2006), where also details on the treatment of typical boundary conditions such as free surface and open boundary conditions can be found.

### 3.2 Local time stepping using the ADER-DG approach

The efficiency of RK time discretization schemes drastically decreases if the order of accuracy becomes greater than four, due to the so-called Butcher barriers (Butcher 1987), and the number of intermediate RK stages becomes larger than the formal order of accuracy. Therefore, we apply the ADER approach to the semi-discrete form of the DG scheme (4) in order to achieve the same accuracy of the time discretization as for the space discretization. Furthermore, time accurate LTS of high order of accuracy would become quite complicated with RK schemes since inside an LTS approach the intermediate RK time levels of two neighbouring elements do not match in the general case. This makes accurate temporal interpolation necessary between the elements. Of course, also the LTS version of the proposed ADER-DG approach needs accurate time interpolation. However, this comes out naturally of the construction of the method thanks to the use of the Cauchy-Kovalewski procedure which provides an accurate prediction of the evolution of the degrees of freedom in each element during one time step.

The main ingredient of the ADER approach is the solution of the Generalized Riemann Problems (GRP) (Toro *et al.* 2001), which takes the form of a Taylor series expansion in time. The Cauchy-Kovalewski procedure is then used to replace the time derivatives in the Taylor series by space derivatives. Since the state and the spatial derivatives are in general discontinuous at the element interfaces, their value at the interface is defined solving Riemann problems for the state and the spatial derivatives (see also Toro & Titarev 2002). Formally, the fully discrete scheme using LTS is the same as the ADER-DG schemes with global time stepping presented in Dumbser & Munz (2005); Käser &

Dumbser (2006); Dumbser & Käser (2006), however the time integration tensor  $I_{plqm}$  will have to be computed taking into account the local time stepping algorithm.

To realize the LTS ADER-DG scheme, each tetrahedron  $T^{(m)}$  stores its own local time  $t^{(m)}$  and its associated local time step  $\Delta t^{(m)}$ . The stability criterion for the local time step is written as

$$\Delta t^{(m)} < \frac{1}{2N + 1} \cdot \frac{l_{\min}^{(m)}}{a_{\max}^{(m)}}, \tag{9}$$

where  $l_{\min}^{(m)}$  is the diameter of the insphere of tetrahedron  $T^{(m)}$  and  $a_{\max}^{(m)}$  is the maximum signal speed arising in the tetrahedron.  $a_{\max}^{(m)}$  is the maximum eigenvalue of the Jacobians  $A_{pq}$ ,  $B_{pq}$  and  $C_{pq}$ , which means for the isotropic elastic wave equations  $a_{\max}^{(m)} = c_p$ , the local seismic  $P$ -wave velocity in tetrahedron  $T^{(m)}$ .

As usual for ADER-DG schemes, we first write the governing eq. (2) in the reference coordinate system as

$$\frac{\partial Q_p}{\partial t} + A_{pq}^* \frac{\partial Q_q}{\partial \xi} + B_{pq}^* \frac{\partial Q_q}{\partial \eta} + C_{pq}^* \frac{\partial Q_q}{\partial \zeta} = 0. \tag{10}$$

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 (10) and is given by

$$\frac{\partial^k Q_p}{\partial t^k} = (-1)^k \left( A_{pq}^* \frac{\partial}{\partial \xi} + B_{pq}^* \frac{\partial}{\partial \eta} + C_{pq}^* \frac{\partial}{\partial \zeta} \right)^k Q_q. \tag{11}$$

We then develop the solution of (10) in a Taylor series in time up to local order  $N = N^{(m)}$  with respect to the current local time level  $t^{(m)}$ ,

$$Q_p(\xi, \eta, \zeta, t) = \sum_{k=0}^{N^{(m)}} \frac{(t - t^{(m)})^k}{k!} \frac{\partial^k}{\partial t^k} Q_p(\xi, \eta, \zeta, t^{(m)}), \tag{12}$$

and replace time derivatives by space derivatives, using (11). We then introduce the DG approximation (3) and get

$$Q_p(\xi, \eta, \zeta, t) = \sum_{k=0}^{N^{(m)}} \frac{(t - t^{(m)})^k}{k!} (-1)^k \left( A_{pq}^* \frac{\partial}{\partial \xi} + B_{pq}^* \frac{\partial}{\partial \eta} + C_{pq}^* \frac{\partial}{\partial \zeta} \right)^k \Phi_l(\xi, \eta, \zeta) \hat{Q}_{ql}(t^{(m)}). \tag{13}$$

This approximation can now be projected onto each basis function in order to get an approximation of the evolution of the degrees of freedom during one local time step, that is, in the time interval  $t \in [t^{(m)}; t^{(m)} + \Delta t^{(m)}]$ . We obtain

$$\hat{Q}_{pl}(t) = \frac{\left\langle \Phi_n, \sum_{k=0}^{N^{(m)}} \frac{(t - t^{(m)})^k}{k!} (-1)^k \left( A_{pq}^* \frac{\partial}{\partial \xi} + B_{pq}^* \frac{\partial}{\partial \eta} + C_{pq}^* \frac{\partial}{\partial \zeta} \right)^k \Phi_m(\xi, \eta, \zeta) \right\rangle}{\langle \Phi_n, \Phi_l \rangle} \hat{Q}_{qm}(t^{(m)}), \tag{14}$$

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  $\Phi_l$ .

The most important equations for the LTS algorithm are the update criterion for the elements and the time interval for flux computation. An element may be updated if and only if the update criterion

$$t^{(m)} + \Delta t^{(m)} \leq \min(t^{(k_j)} + \Delta t^{(k_j)}) \quad \forall k_j \tag{15}$$

is fulfilled with respect to all the direct neighbouring tetrahedrons  $T^{(k_j)}$ . When an element is updated, fulfilling (15), the numerical fluxes between two elements  $T^{(m)}$  and  $T^{(k_j)}$  have to be computed in the time interval

$$[t_1; t_2] = [\max(t^{(m)}, t^{(k_j)}); \min(t^{(m)} + \Delta t^{(m)}, t^{(k_j)} + \Delta t^{(k_j)})]. \tag{16}$$

Similar to the global time stepping case presented, for example, in Käser & Dumbser (2006) and Dumbser & Käser (2006), we introduce the definition

$$\int_{t_1}^{t_2} \hat{Q}_{pl}(t) dt = I_{plqm}(t_0, t_1, t_2) \hat{Q}_{qm}(t_0), \tag{17}$$

with

$$I_{plqm}(t_0, t_1, t_2) = \frac{\left\langle \Phi_n, \sum_{k=0}^{N^{(m)}} \frac{(t_2 - t_0)^{k+1} - (t_1 - t_0)^{k+1}}{(k+1)!} (-1)^k \left( A_{pq}^* \frac{\partial}{\partial \xi} + B_{pq}^* \frac{\partial}{\partial \eta} + C_{pq}^* \frac{\partial}{\partial \zeta} \right)^k \Phi_m(\xi, \eta, \zeta) \right\rangle}{\langle \Phi_n, \Phi_l \rangle}, \tag{18}$$



where  $I_{plqm}(t_0, t_1, t_2)$  formally contains the Cauchy-Kovalewski procedure for the time interval  $t \in [t_1; t_2]$  with the Taylor series expanded about time level  $t_0$ . With (18), the fully discrete  $p$ -adaptive LTS ADER-DG scheme is given under condition (15) by:

$$\begin{aligned} & [\hat{Q}_{pl}^{(m)}(t^{(m)} + \Delta t^{(m)}) - \hat{Q}_{pl}^{(m)}(t^{(m)})] |J| M_{kl} \\ & + \frac{1}{2} \sum_{j=1}^4 T_{pq}^j (A_{qr}^{(m)} + |A_{qr}^{(m)}|) (T_{rs}^j)^{-1} |S_j| F_{kl}^{-,j} \cdot I_{slmo}(t^{(m)}, t_1, t_2) \cdot \hat{Q}_{mo}^{(m)}(t^{(m)}) \\ & + \frac{1}{2} \sum_{j=1}^4 T_{pq}^j (A_{qr}^{(m)} - |A_{qr}^{(m)}|) (T_{rs}^j)^{-1} |S_j| F_{kn}^{+,j,i,h} \cdot I_{snmo}(t^{(k_j)}, t_1, t_2) \cdot \hat{Q}_{mo}^{(k_j)}(t^{(k_j)}) \\ & - A_{pq}^* |J| K_{kl}^\xi \cdot I_{qlmo}(t^{(m)}, t^{(m)}, t^{(m)} + \Delta t^{(m)}) \cdot \hat{Q}_{mo}^{(m)}(t^{(m)}) \\ & - B_{pq}^* |J| K_{kl}^\eta \cdot I_{qlmo}(t^{(m)}, t^{(m)}, t^{(m)} + \Delta t^{(m)}) \cdot \hat{Q}_{mo}^{(m)}(t^{(m)}) \\ & - C_{pq}^* |J| K_{kl}^\zeta \cdot I_{qlmo}(t^{(m)}, t^{(m)}, t^{(m)} + \Delta t^{(m)}) \cdot \hat{Q}_{mo}^{(m)}(t^{(m)}) = \mathcal{M}\mathcal{V}_{pk}^{(m)}. \end{aligned} \quad (19)$$

Here, the indices  $l$  as well as  $n$  and  $o$  obey the same range limits as given in detail for the semi-discrete  $p$ -adaptive scheme (4). Furthermore, the times  $t_1$  and  $t_2$  denote the limits of the flux time interval and are given according to (16). The accurate interpolation of element time levels needed by a consistent flux computation are done automatically by the Cauchy-Kovalewski procedure using (17) with (18). We note that the contributions of the stiffness matrices are always computed for the full local time step  $\Delta t_m$  since the volume integrals depend only on element-local information.

However, since the fluxes computed for element  $T^{(m)}$  in the interval (16) in general do not cover the entire local time step, all other flux contributions must be computed by the neighbouring elements when they fulfil the update criterion (15). In other words, the flux integral for the entire time interval  $[t^{(m)}; t^{(m)} + \Delta t^{(m)}]$  is split into the element's own contribution in the subinterval  $[t_1; t_2]$  and the contributions of the corresponding neighbour element in the rest of the interval. These neighbour contributions may come from multiple local time steps of the neighbour. On this behalf, the neighbours add their respective flux contributions to a so-called flux memory variable  $\mathcal{M}\mathcal{V}_{pl}^{(m)}$  for tetrahedron  $T^{(m)}$ . We stress that the flux memory variables  $\mathcal{M}\mathcal{V}_{pl}^{(m)}$  are used to store the flux contributions of the neighbour elements and that they must not be confounded with the usual memory variables used in order to model anelastic attenuation. At the initial time  $t = 0$  all flux memory variables are initialized with zero.

After a local element time step has been performed according to (19), the flux memory variable  $\mathcal{M}\mathcal{V}_{pk}^{(m)}$  of this element itself is reset to zero and the increments  $\Delta \mathcal{M}\mathcal{V}_{pk}^{(k_j)}$  that have to be added to the flux memory variables of all its neighbours  $T^{(k_j)}$  are computed by:

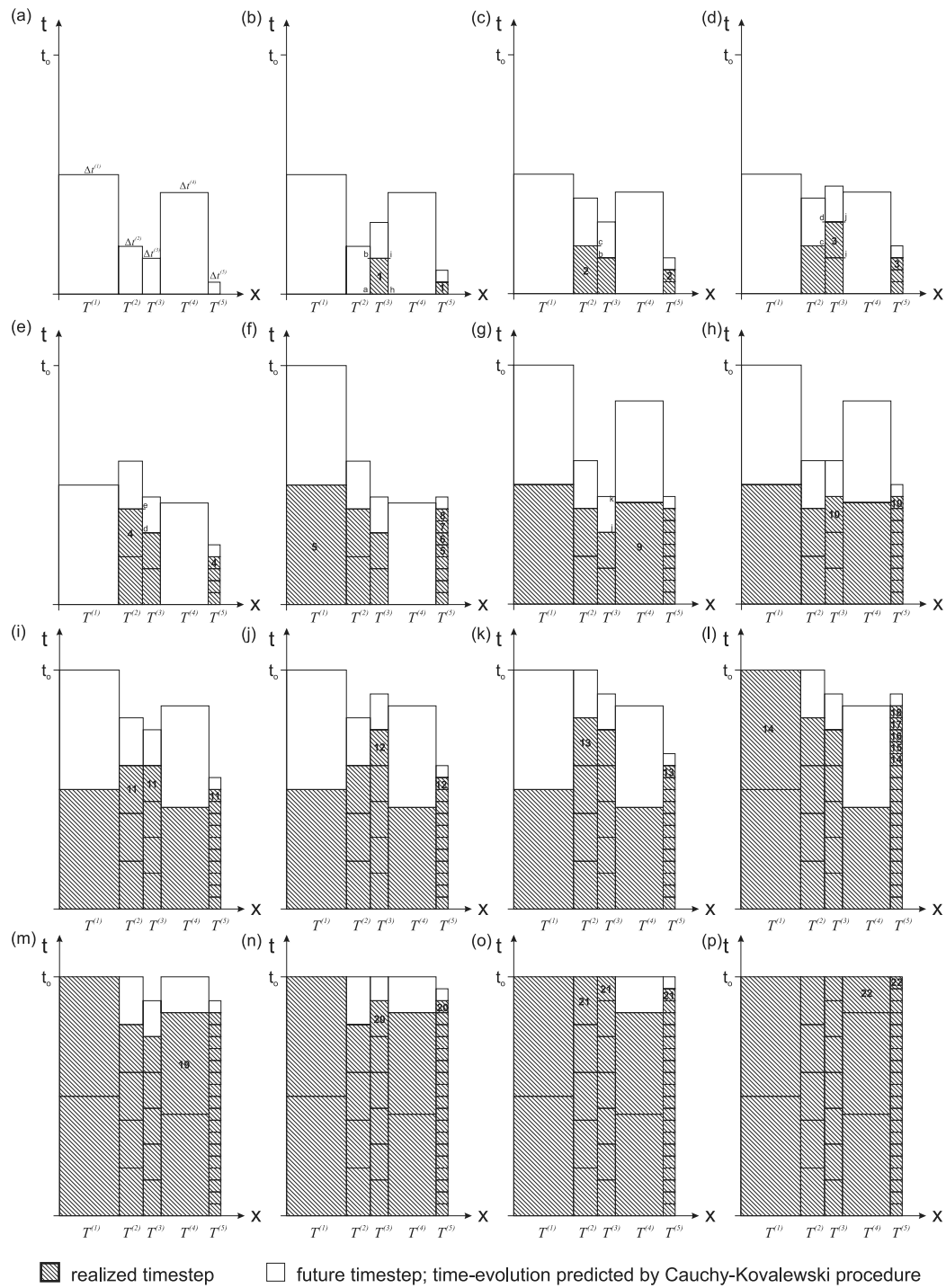
$$\begin{aligned} & -\frac{1}{2} T_{pq}^j (A_{qr}^{(k_j)} + |A_{qr}^{(k_j)}|) (T_{rs}^j)^{-1} |S_j| F_{kl}^{-,i} \cdot I_{qlmo}(t^{(k_j)}, t_1, t_2) \cdot \hat{Q}_{mo}^{(k_j)}(t^{(k_j)}) \\ & -\frac{1}{2} T_{pq}^j (A_{qr}^{(k_j)} - |A_{qr}^{(k_j)}|) (T_{rs}^j)^{-1} |S_j| F_{kn}^{+,i,j,h} \cdot I_{qnmo}(t^{(m)}, t_1, t_2) \cdot \hat{Q}_{mo}^{(m)}(t^{(m)}) = \Delta \mathcal{M}\mathcal{V}_{pk}^{(k_j)}, \quad \forall k_j. \end{aligned} \quad (20)$$

Instead of using the tensor given in (18) one can alternatively use the faster algorithm given in the appendix of Käser *et al.* (2007) and store the resulting temporal Taylor series for each degree of freedom in each element. This increases computational speed at the expense of higher memory requirements.

A particular feature of the proposed  $p$ -adaptive ADER-DG scheme is its  $p$ -adaptivity not only in space, but also in *time*. This comes in automatically via the Cauchy-Kovalewski procedure, which automatically matches spatial and temporal accuracy. This would be quite difficult to obtain with a RK DG scheme since the use of RK schemes with locally different order of accuracy would require additional interpolation at the element interfaces since the time levels of the intermediate RK stages do not necessarily match. Therefore, we decide to call our ADER-DG scheme  $p_t$ -adaptive, indicating that it adapts the local polynomial degree not only in space but also in time.

### 3.3 Discussion of the local time stepping algorithm

In this section, we would like to illustrate the algorithm given by (19) and (20) under conditions (15) and (16) in detail on a schematic 1-D example. Consider five non-equidistantly spaced elements  $T^{(1)}$  to  $T^{(5)}$  as shown in Fig. 1 with their associated different local time steps  $\Delta t^{(1)}$  to  $\Delta t^{(5)}$  as plotted in Fig. 1(a). Assume that all elements start at the same common time level  $t = 0$  and that all of them have to reach the same final output time  $t = t_o$ . Since for a LTS algorithm the time advancement of the elements will not be synchronous, in the following we will talk of ‘cycles’, which we define as follows: during one cycle, only the subset of all elements  $T^{(m)}$  which fulfills (15) is updated. Turning back to our example shown in Fig. 1 we see that in the first cycle only elements  $T^{(3)}$  and  $T^{(5)}$  fulfill the update criterion (15). Thus, in cycle 1 they are allowed to perform an update. In order to compute the fluxes, the values for the neighbouring elements are interpolated via the Cauchy-Kovalewski procedure thanks to (18) in the intervals given by (16). This is discussed in more detail on the example of element  $T^{(3)}$  and its neighbours in the following. The time intervals in which flux computation is performed according to (16) are  $[a; b]$  on the interface with neighbour  $T^{(2)}$  and  $[h; i]$  on the interface with neighbour  $T^{(4)}$ . The flux memory variables  $\mathcal{M}\mathcal{V}_{pk}^{(2)}$  and  $\mathcal{M}\mathcal{V}_{pk}^{(4)}$  of both neighbours are updated according to (20). In cycle 2, only  $T^{(2)}$  and  $T^{(5)}$  are allowed to perform an update, see also Fig. 1(c). According to (16), the flux between  $T^{(2)}$  and  $T^{(3)}$  has to be computed only in the time interval  $[b; c]$  since the flux in the interval  $[a; b]$  was already computed in cycle 1 by the neighbour element  $T^{(3)}$ . Since this contribution was stored in the flux memory variable of  $T^{(2)}$ , it is now correctly taken into account



Downloaded from https://academic.oup.com/gji/article/171/2/695/657128 by U.S. Department of Justice user on 16 August 2022

**Figure 1.** Illustration of the local time stepping algorithm in one space dimension showing the space–time elements.

during the update step for  $T^{(2)}$ . The flux contribution of element  $T^{(2)}$  to its neighbour  $T^{(3)}$  is likewise added to the flux memory variable of neighbour  $T^{(3)}$ . Remember that at the end of cycle 2  $\mathcal{M}\mathcal{V}_{pk}^{(2)}$  is reset to zero. In a similar way,  $T^{(5)}$  will add its flux contribution to the flux memory variables of element  $T^{(4)}$ . In cycle 3 besides  $T^{(5)}$  also element  $T^{(3)}$  fulfills again condition (15), see Fig. 1(d), and computes the fluxes in the time interval  $[c;d]$  at the interface with neighbour  $T^{(2)}$  since the contribution of interval  $[b;c]$  was already computed in cycle 2 by  $T^{(2)}$  and stored in  $\mathcal{M}\mathcal{V}_{pk}^{(3)}$ . With respect to neighbour  $T^{(4)}$  the flux is computed according to (16) in the interval  $[i;j]$ , contributing obviously to the update of element  $T^{(3)}$  but also to a corresponding update of the flux memory variable  $\mathcal{M}\mathcal{V}_{pk}^{(4)}$ . In the following cycle 4, the elements  $T^{(2)}$  and  $T^{(5)}$  perform a time update, see Fig. 1(e). The flux time interval at the interfaces between  $T^{(2)}$  and  $T^{(3)}$  is now  $[d;e]$ .  $T^{(1)}$  fulfills the update criterion for the first time in cycle 5, see Fig. 1(f). In cycles 6–8, only the smallest element  $T^{(5)}$  is updated since none of the other elements fulfills the update criterion. This LTS procedure continues in this manner until cycle 22, when all elements have reached the final

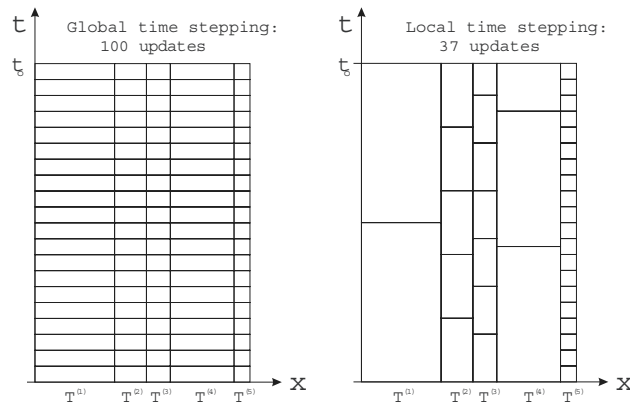


Figure 2. Comparison of the resulting space–time meshes using global time stepping (left) and local time stepping (right).

output time  $t_o$ , see Figs 1(f)–(p). A cycle of particular interest is cycle 11, where the adjacent elements  $T^{(2)}$  and  $T^{(3)}$  fulfil the update criterion simultaneously because they reach a common local time level, say  $t_c$ . We want to use this example to illustrate how the local time stepping works in this special case. At the beginning of the cycle, both elements are at the different local time levels  $t_c - \Delta t^{(2)}$  and  $t_c - \Delta t^{(3)}$ . Suppose that within cycle 11 the algorithm first considers element  $T^{(2)}$  for update. This means, the interface flux will be computed by  $T^{(2)}$ , updating the flux memory variable  $\mathcal{M}_{pk}^{(3)}$ . Element  $T^{(2)}$  subsequently reaches time  $t_c$ . When it is the turn of  $T^{(3)}$  to be updated within the same cycle, the flux at the common interface has already been computed by its neighbour. This is correctly respected by the flux time interval (16) since  $t_1 = \max [t^{(2)}, t^{(3)}] = t_c$  and  $t_2 = \min [t^{(2)} + \Delta t^{(2)}, t^{(3)} + \Delta t^{(3)}] = t_c$ . The time interval  $[t_1 ; t_2] = [t_c ; t_c]$  becomes singular and, therefore, no additional flux calculation results from  $T^{(3)}$ . The same reasoning is true for a different ordering of both elements, that is, if *first* element  $T^{(3)}$  is scheduled for updating and then element  $T^{(2)}$ . Note that except for the final time steps and intermediate output times, where all local element time steps are adjusted in order to reach exactly  $t_o$ , no synchronization is used.

Finally, we would like to illustrate the benefits of the LTS algorithm discussing Fig. 2. On the left we can see the usual element update procedure for a global time stepping scheme. The time step of all elements is restricted to the smallest one arising in the whole computational domain. In our example, this is  $\Delta t^{(5)}$  and leads to a total number of 100 element updates. For time-dependent problems, the solution depends on both space and time and, therefore, when the solution is advanced in time, the final mesh is not only a spatial mesh, but a space–time mesh. This is illustrated in Fig. 2, where the *spatial* mesh in  $x$ -direction and the *temporal* mesh in  $t$ -direction are shown. Since for a global time stepping scheme there are no hanging nodes in time, the mesh is called conforming. Instead of 100 element updates, we can also say that the method uses a space–time mesh with 100 conforming space–time elements. For the LTS scheme the mesh size in time is determined in an optimal manner since each element can run at its maximum time step allowed by the stability criterion (9). This, however, leads to a non-conforming mesh with hanging nodes in time. To deal with these hanging nodes, the concept of flux memory variables was introduced since the flux integral in time is linearly additive. The resulting space–time mesh for the LTS scheme contains only 37 space–time elements. It is more complicated in structure than the conforming mesh of the global time stepping scheme with 100 space–time elements but leads to less computational effort.

In contrast to the space–time DG approach (van der Vegt & van der Ven 2002), which is implicit in time and whose basis functions depend on both space and time, our algorithm remains explicit in time with basis functions depending only on space. The space–time character of the ADER-DG schemes comes in directly via the Cauchy-Kovalevski procedure and the Taylor series in time in eq. (12). We emphasize that compared to the space–time DG scheme of van der Vegt & van der Ven (2002) our ADER-DG algorithm needs less basis functions at the same polynomial degree since on 3-D meshes their approach leads to a 4-D polynomial basis.

We note that the general overhead associated with the LTS algorithm is very low compared to an ADER-DG scheme using global time stepping. Computations with the serial code on 3-D tetrahedral meshes have shown an overhead of only about 15 per cent. Please note, however, that the efficient *parallel* execution of a LTS scheme can become a very challenging task. Since each element performs its update only when (15) is fulfilled locally, the element updates are in general completely asynchronous which induces a very irregular load balancing on the different processors. Therefore, for the parallel implementation of our code, we first group elements of zones with similar material properties and element size together and partition each of these zones separately into a number of partitions that is equal to the total number of processors. For this purpose we are using the free METIS software package introduced by Karypis & Kumar (1998). Then, the resulting partitions of the different zones are merged together and thus provide the final domain decomposition. Note that the final partition in general contains non-connected subdomains. However, since the ADER-DG approach provides a one-step scheme, the total communication overhead induced by these non-contiguous partitions is quite small compared to the achieved benefits from the improved load balancing. Note that just using METIS on the whole domain with large weights on elements with small time steps can not resolve the problem since the processor load may remain ill-balanced due to the asynchronous element updates. As a final remark on parallelization we would like to note that in our implementation the MPI communication is always performed after each cycle.



At the end of this section, we would like to investigate theoretically the maximum speed-up that can be reached ideally by a LTS scheme. We consider a spatial computational domain  $\Omega$  with a total number of  $E$  spatial elements. We assume that  $\Omega$  contains only two different time steps, the small time step  $\Delta t_1$  and the large time step  $\Delta t_2$ , with  $\Delta t_2 = \tau \Delta t_1$  and the time step ratio  $\tau > 1$ . We assume furthermore that the percentage of elements with the small time step  $\Delta t_1$  is  $\alpha < 1$  and the percentage of the elements with the larger time step  $\Delta t_2$  is  $(1 - \alpha)$ . The total number of space–time elements  $N_{STE}^{LTS}$  in the space–time domain  $\Omega \times T$  is for the LTS scheme

$$N_{STE}^{LTS} = \alpha E \frac{T}{\Delta t_1} + (1 - \alpha) E \frac{T}{\Delta t_2}. \quad (21)$$

For a global time stepping method, all elements have to run the small time step  $\Delta t_1$ , so the corresponding number of space–time elements is

$$N_{STE}^{GTS} = \alpha E \frac{T}{\Delta t_1} + (1 - \alpha) E \frac{T}{\Delta t_1}. \quad (22)$$

The theoretical speed-up  $\sigma$  is now given as the ratio of the number of space–time elements in (22) and (21) and can finally be written as

$$\sigma = \frac{N_{STE}^{GTS}}{N_{STE}^{LTS}} = \frac{\tau}{1 - \alpha + \alpha\tau}. \quad (23)$$

To understand this formula in a more concrete sense, we consider the two extreme cases  $\tau \rightarrow \infty$ , which is the case for an extreme disparity in the two time steps, and the case  $\alpha \rightarrow 0$ , which is the case when only a negligibly small number of mesh elements has to use the small time step  $\Delta t_1$ . In the first case, we obtain

$$\lim_{\tau \rightarrow \infty} \sigma = \frac{1}{\alpha}, \quad (24)$$

and in the second case

$$\lim_{\alpha \rightarrow 0} \sigma = \tau. \quad (25)$$

Especially the first result (24) is important for practical applications. In the limit case of an extremely large time step ratio  $\tau$ , it is *not* the time step ratio that determines the maximum theoretical speed-up, but it is the *percentage*  $\alpha$  of elements using the small time step  $\Delta t_1$ . This means that, for example, in a computational domain where 20 per cent of the elements have to use the small time step  $\Delta t_1$  the maximum speed-up can not go beyond 5. When already 50 per cent of the elements must use the small time step, one can not expect the speed-up to be better than two. One should always be aware of this important result when deciding whether to use a LTS algorithm or not, especially considering the difficulties with load-balancing of LTS schemes on massively parallel supercomputers.

The time step ratio  $\tau$  determines the speed-up only in the case when extremely few elements with the small time step  $\Delta t_1$  are present in the domain ( $\alpha \rightarrow 0$ ). This is the case, for example, when the tetrahedral mesh contains only one single sliver. An application of this extreme case is shown in the following Section 4.2, where we show the convergence results obtained with our LTS scheme on a regular tetrahedral mesh containing four slivers.

#### 4 NUMERICAL CONVERGENCE STUDIES OF THE LTS ADER-DG SCHEME

Here, we present the results of the numerical convergence analysis in order to confirm the uniformly high accuracy in space and time of the proposed ADER-DG method with time-accurate LTS. Convergence orders of the LTS ADER-DG schemes are shown from second to sixth order in space and time and are denoted by LTS ADER-DG  $\mathcal{O}2$  to ADER-DG  $\mathcal{O}6$ , respectively. We note that an ADER-DG scheme using basis polynomials  $\Phi_i$  of maximum degree  $N$  has the formal order of accuracy  $N + 1$  in space and time. This means that, for example, an ADER-DG  $\mathcal{O}6$  method uses basis polynomials of maximal degree  $N = 5$ .

To determine the convergence orders we solve the 3-D elastic wave equations in (2) in a cube shaped computational domain  $\Omega = [-1, 1] \times [-1, 1] \times [-1, 1] \in \mathbb{R}^3$  with periodic boundary conditions. The initial condition is given by

$$\mathcal{Q}_p^0 = \mathcal{Q}_p(\vec{x}, 0) = R_{p1}^{A_n} \sin(\vec{k} \cdot \vec{x}) + R_{p8}^{A_n} \sin(\vec{k} \cdot \vec{x}), \quad (26)$$

with the wave number

$$\vec{k} = (k_x, k_y, k_z)^T = (\pi, \pi, \pi)^T. \quad (27)$$

The vectors  $R_{p1}^{A_n}$  and  $R_{p8}^{A_n}$  are the two eigenvectors of matrix  $A_{pq}$  rotated in normal direction, associated with the eigenvalues  $-c_p$  and  $c_s$ . Therefore, the initial condition (26) creates a plane sinusoidal  $P$ -wave travelling along the diagonal direction  $\vec{n} = (-1, -1, -1)^T$  of the cube, as well as a plane sinusoidal  $S$ -wave travelling into the opposite direction. The homogeneous material parameters are set to

$$\lambda = 2, \quad \mu = 1, \quad \rho = 1, \quad (28)$$

throughout the computational domain  $\Omega$  leading to the constant wave propagation velocities

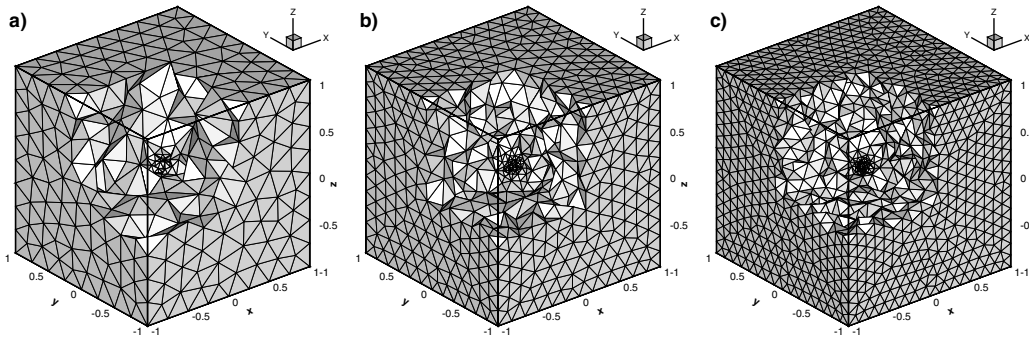
$$c_p = 2, \quad c_s = 1, \quad (29)$$

for the  $P$  and  $S$  wave, respectively. The exact solution of this problem is

$$\mathcal{Q}_e = \mathcal{Q}_e(\vec{x}, t) = R_{p1}^{A_n} \sin(\vec{k} \cdot \vec{x} + |\vec{k}|c_p \cdot t) + R_{p8}^{A_n} \sin(\vec{k} \cdot \vec{x} - |\vec{k}|c_s \cdot t). \quad (30)$$

**Table 1.** Numerical convergence rates for variable  $w$  using ADER-DG schemes with local time stepping from second to sixth order on the irregular tetrahedral meshes shown in Fig. 3.

$h$	$E_{L^\infty}$	$E_{L^1}$	$E_{L^2}$	$\mathcal{O}_{L^\infty}$	$\mathcal{O}_{L^1}$	$\mathcal{O}_{L^2}$	CPU (s)
1.65E-01	1.7362E-01	1.3511E-01	6.4464E-02	–	–	–	15
1.19E-01	8.7756E-02	6.2315E-02	3.0440E-02	2.1	2.4	2.3	71
9.05E-02	5.3290E-02	3.5194E-02	1.7267E-02	1.8	2.1	2.1	174
7.17E-02	3.3972E-02	2.1057E-02	1.0432E-02	1.9	2.2	2.2	438
1.65E-01	4.1876E-02	1.6274E-02	8.5126E-03	–	–	–	33
1.19E-01	1.7583E-02	5.1326E-03	2.7391E-03	2.6	3.5	3.5	152
9.05E-02	1.0291E-02	2.1468E-03	1.1720E-03	2.0	3.2	3.1	375
7.17E-02	4.0023E-03	9.9995E-04	5.5279E-04	4.1	3.3	3.2	936
1.65E-01	8.8193E-03	1.3238E-03	7.5536E-04	–	–	–	74
1.19E-01	3.1491E-03	3.0846E-04	1.8126E-04	3.1	4.4	4.3	336
9.05E-02	8.2239E-04	9.8214E-05	5.8509E-05	5.0	4.2	4.2	816
7.17E-02	4.3934E-04	3.5546E-05	2.1520E-05	2.7	4.4	4.3	2042
1.65E-01	1.4000E-03	9.2758E-05	5.9655E-05	–	–	–	169
1.19E-01	2.8336E-04	1.4341E-05	9.4223E-06	4.9	5.7	5.6	767
9.05E-02	9.1573E-05	3.5845E-06	2.4368E-06	4.2	5.1	5.0	1854
7.17E-02	2.2474E-05	1.0218E-06	7.0855E-07	6.0	5.4	5.3	4639
1.65E-01	1.1073E-04	5.3844E-06	3.8582E-06	–	–	–	392
1.19E-01	2.7071E-05	5.9898E-07	4.4779E-07	4.3	6.7	6.6	1799
9.05E-02	3.2553E-06	1.1377E-07	8.4015E-08	7.8	6.1	6.2	4319
7.17E-02	1.5113E-06	2.6350E-08	1.9615E-08	3.3	6.3	6.3	10944

**Figure 3.** Sequence of irregular tetrahedral meshes, used for the numerical convergence analysis of the local time stepping scheme.

The total simulation time  $T$  for the convergence test is set to  $T = 0.1$ . The stability of our explicit LTS scheme is controlled by condition (9).

We then pick arbitrarily one of the variables of the system of the elastic wave eq. (2) to numerically determine the convergence order of the used LTS ADER-DG schemes. In Tables 1 and 3, we show the errors for the velocity component  $w$ . The error of the numerical solution  $Q_h$  with respect to the exact solution  $Q_e$  is measured in the  $L^\infty$ -norm and the continuous  $L^1$  and  $L^2$  norms,

$$E_{L^v}^s = \|Q_{h^s} - Q_e\|_{L^v(\Omega)} = \left( \int_{\Omega} |Q_{h^s} - Q_e|^v dV \right)^{1/v}, \quad \text{with } v \in \{1, 2, \infty\}, \quad (31)$$

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

$$\mathcal{O}_{L^v} = \log \left( \frac{E_{L^v}^s}{E_{L^v}^{s-1}} \right) / \log \left( \frac{h^s}{h^{s-1}} \right), \quad \text{with } v \in \{1, 2, \infty\}, \quad (32)$$

where  $h^s$  indicates the mesh spacing  $h$  of mesh number  $s$  in the sequence of meshes. The first column in Tables 1 and 3 shows the mesh spacing  $h$ , represented by the maximum diameter of the circumscribed spheres of the tetrahedrons. The following six columns show the  $L^\infty$ ,  $L^1$  and  $L^2$  errors with the corresponding convergence orders  $\mathcal{O}_{L^\infty}$ ,  $\mathcal{O}_{L^1}$  and  $\mathcal{O}_{L^2}$  determined by successively refined meshes. In the last column we give the CPU time in seconds needed to reach the final simulation time  $T = 0.1$  with the serial version of the code on one Pentium Xeon processor with 3.6 GHz and 4GB of RAM.

**Table 2.** Comparison of ADER-DG schemes using global time stepping with pure P3 elements (top), local time stepping with pure P3 elements (middle) and local time stepping with P2 and P3 elements (bottom). The analysed variable is  $w$ , the meshes used for this test are shown in Fig. 3.

$h$	$E_{L^\infty}$	$E_{L^1}$	$E_{L^2}$	$\mathcal{O}_{L^\infty}$	$\mathcal{O}_{L^1}$	$\mathcal{O}_{L^2}$	CPU (s)	Speed-up
1.65E-01	1.0217E-02	1.3756E-03	7.9740E-04	–	–	–	219	–
1.19E-01	3.7863E-03	3.1806E-04	1.9050E-04	3.0	4.5	4.4	1050	–
9.05E-02	9.4231E-04	1.0197E-04	6.2392E-05	5.1	4.2	4.1	2992	–
7.17E-02	5.1042E-04	3.6568E-05	2.2793E-05	2.6	4.4	4.3	7410	–
1.65E-01	8.8193E-03	1.3238E-03	7.5536E-04	–	–	–	74	2.95
1.19E-01	3.1491E-03	3.0846E-04	1.8126E-04	3.1	4.4	4.3	336	3.13
9.05E-02	8.2239E-04	9.8214E-05	5.8509E-05	5.0	4.2	4.2	816	3.67
7.17E-02	4.3934E-04	3.5546E-05	2.1520E-05	2.7	4.4	4.3	2042	3.63
1.65E-01	8.8193E-03	1.3238E-03	7.5536E-04	–	–	–	72	3.02
1.19E-01	3.1491E-03	3.0846E-04	1.8126E-04	3.1	4.4	4.3	331	3.17
9.05E-02	8.2239E-04	9.8214E-05	5.8509E-05	5.0	4.2	4.2	788	3.80
7.17E-02	4.3934E-04	3.5546E-05	2.1520E-05	2.7	4.4	4.3	1971	3.76

#### 4.1 Example 1: Tetrahedral mesh with strongly varying element size

The convergence study to determine the numerical order of accuracy is first performed on a sequence of irregular tetrahedral meshes as shown in Fig. 3, where the ratio of the edge lengths of the largest to the smallest tetrahedron is chosen to be  $h_{\max}/h_{\min} = 5$ .

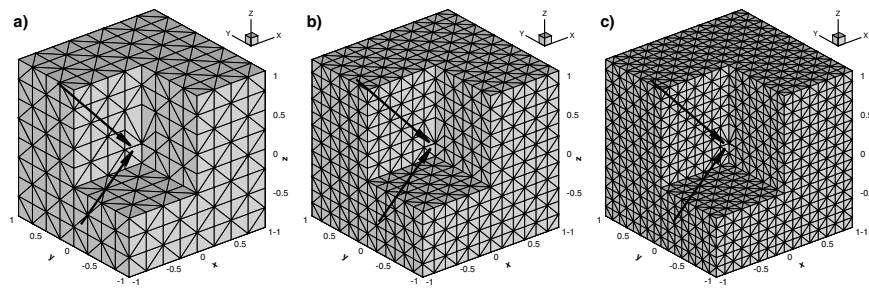
From the numerical convergence rates shown in Table 1 we deduce that our proposed LTS ADER-DG scheme maintains time accuracy perfectly even when using LTS. At the same time the scheme increases computational efficiency since in each element the optimal time step can be used. The main advantage of such an algorithm is the considerable increase of robustness with respect to the underlying mesh which for very complex realistic 3-D geometries may contain elements of strongly varying element size. For example, in applications with surface topography, close to the surface, one usually must use quite a fine mesh in order to resolve the features of the topography correctly, whereas in the deeper regions below the surface, less quickly changing geometrical features may allow larger elements. For a global time stepping scheme, the elements on the surface would restrict the time step for all the other elements, even for the large elements far below the surface.

To study the speed-up induced by LTS and furthermore to assess the accuracy of the  $p_\tau$ -adaptive version of the scheme, we perform the same convergence analysis again using the fourth order global time stepping ADER-DG scheme as presented in Käser & Dumbser (2006) and Dumbser & Käser (2006) as a reference. The results shown in Table 2 clearly show that the algorithm presented in this article using the LTS technique is much faster than the one using global time stepping. On the finest mesh, the CPU time needed by the LTS scheme is less than 28 per cent of the global time stepping method. Another interesting additional benefit of LTS is that the method is not only faster, but the results are even *better*. This is due to the uniformly high local CFL number which should be as large as possible to have the least errors due to numerical dissipation and dispersion. Using global time stepping, the local CFL number is quite low in the large tetrahedrons inducing an unnecessary amount of numerical dissipation and dispersion. The details on the correlation of the CFL number and the dispersion and dissipation errors of ADER FV schemes are given in Dumbser *et al.* (2006a).

Table 2, furthermore, shows that the  $p_\tau$ -adaptive version of the LTS ADER-DG scheme furthermore increases computational efficiency, without loss of accuracy with respect to the LTS ADER-DG scheme using pure P3 elements ( $N = 3$ ). However, we remark that on sufficiently refined meshes the global order of accuracy of the mixed P2-P3 scheme must decrease to  $\mathcal{O}3$ . In the testcase shown, the speed-up due to the  $p_\tau$ -adaptive scheme is quite small. This is due to the fact that only a small number of elements uses the reduced order. However, using a lower order in the small tetrahedrons in the interior of the mesh is beneficial from several points of view: first, the small tetrahedrons lead to a higher spatial resolution compared to the larger tetrahedrons at the exterior of the domain. Therefore, the use of a lower order method is justified in order to balance accuracy in the whole mesh. Second, the lower order ADER-DG method implies a more generous stability limit on the time step since for DG schemes the time step limit is not only a function of the maximum wave speed and the element size, but also of the degree  $N$  of the basis polynomials. Thus, the decrease of the order in the small tetrahedrons can compensate to some extent the more severe time step restriction due to the small size of the elements. Third, the lower order (P2) elements in the interior are computationally faster than the higher order (P3) elements in the rest of the domain. Since in the LTS algorithm the small elements have to do more time steps than the larger ones, this leads to a further benefit concerning CPU time.

#### 4.2 Example 2: Tetrahedral mesh with four extremely distorted sliver elements

This second test case is very similar compared to the previous one. However, we now consider a sequence of regular tetrahedral meshes into which a set of four degenerate tetrahedrons, so-called slivers, has been introduced, see Fig. 4. The four slivers have been generated based on a perfectly regular tetrahedral grid dividing the  $z$ -coordinate  $x_3$  by 1000 for the vertex  $V = (0/0/x_3)$  on the positive  $z$ -axis which is closest to the origin  $O = (0/0/0)$ . After this modification, the modified vertex  $V' = (0/0/\frac{x_3}{1000})$  and the origin  $O$  almost coincide. This leads to



**Figure 4.** Sequence of regular tetrahedral meshes including a set of degenerate tetrahedrons (slivers) indicated between the arrows.

**Table 3.** Numerical convergence rates for variable  $w$  using ADER-DG schemes with local time stepping from second to sixth order on the regular tetrahedral meshes including four slivers as shown in Fig. 4.

$h$	$E_{L^\infty}$	$E_{L^1}$	$E_{L^2}$	$\mathcal{O}_{L^\infty}$	$\mathcal{O}_{L^1}$	$\mathcal{O}_{L^2}$	CPU (s)
1.98E-01	1.6058E-01	1.7690E-01	7.9682E-02	–	–	–	16
1.32E-01	8.0926E-02	7.0999E-02	3.3510E-02	1.7	2.3	2.1	89
9.92E-02	4.6125E-02	3.9241E-02	1.8634E-02	2.0	2.1	2.0	279
7.94E-02	3.0424E-02	2.4037E-02	1.1657E-02	1.9	2.2	2.1	670
1.98E-01	4.7417E-02	2.3606E-02	1.1258E-02	–	–	–	41
1.32E-01	7.7286E-03	6.7786E-03	3.2771E-03	4.5	3.1	3.0	178
9.92E-02	3.5890E-03	2.7267E-03	1.3269E-03	2.7	3.2	3.1	506
7.94E-02	1.5846E-03	1.2560E-03	6.4294E-04	3.7	3.5	3.2	1272
1.98E-01	9.4030E-03	2.3390E-03	1.1777E-03	–	–	–	86
1.32E-01	2.3362E-03	3.9467E-04	2.1107E-04	3.4	4.4	4.2	339
9.92E-02	7.6429E-04	1.3282E-04	6.7896E-05	3.9	3.8	3.9	844
7.94E-02	3.3179E-04	4.8942E-05	2.5815E-05	3.7	4.5	4.3	2070
1.98E-01	1.2266E-03	1.7541E-04	9.4201E-05	–	–	–	166
1.32E-01	8.2925E-05	1.9842E-05	1.0348E-05	6.6	5.4	5.4	578
9.92E-02	1.2253E-05	4.8083E-06	2.5230E-06	6.6	4.9	4.9	1422
7.94E-02	4.7892E-06	1.5016E-06	8.0887E-07	4.2	5.2	5.1	3391
1.98E-01	1.8494E-04	1.1606E-05	7.2153E-06	–	–	–	572
1.32E-01	2.0914E-05	8.1676E-07	5.2066E-07	5.4	6.5	6.5	1877
9.92E-02	3.9283E-06	1.5152E-07	8.7313E-08	5.8	5.9	6.2	4294
7.94E-02	1.0296E-06	3.9531E-08	2.2398E-08	6.0	6.0	6.1	9737

four extremely degenerate tetrahedrons connected to the modified vertex  $V'$  and the origin  $O$  so that it is already impossible to see them without heavy zooming. Their location is indicated by the arrows in Fig. 4. Due to the severe time step restriction induced by such elements, the presented meshes could not be practically used by classical explicit time stepping schemes. However, as shown in Table 3 our proposed local time stepping method obtains perfect convergence rates in reasonable CPU time even on a mesh containing such degenerate elements. Running the scheme on a perfectly regular tetrahedral grid without modification yields almost the same error norms.

Although such an extreme ratio of  $h_{\max}/h_{\min} = 1000$  may not seem to be very realistic, we added this very difficult example in order to underline the capability and the robustness of the LTS ADER-DG method even in such an extreme situation. Real applications, where this huge discrepancy in element size may indeed arise, could consist in earthquake simulations on the regional scale where small-scale features of interest, such as buildings or bridges, are embedded directly in the numerical model used for the earthquake simulation. Such an application may be of interest in civil engineering, where the impact of earthquakes on civil structures is to be studied in order to assess their security.

## 5 APPLICATION EXAMPLES

### 5.1 Layer over half-space

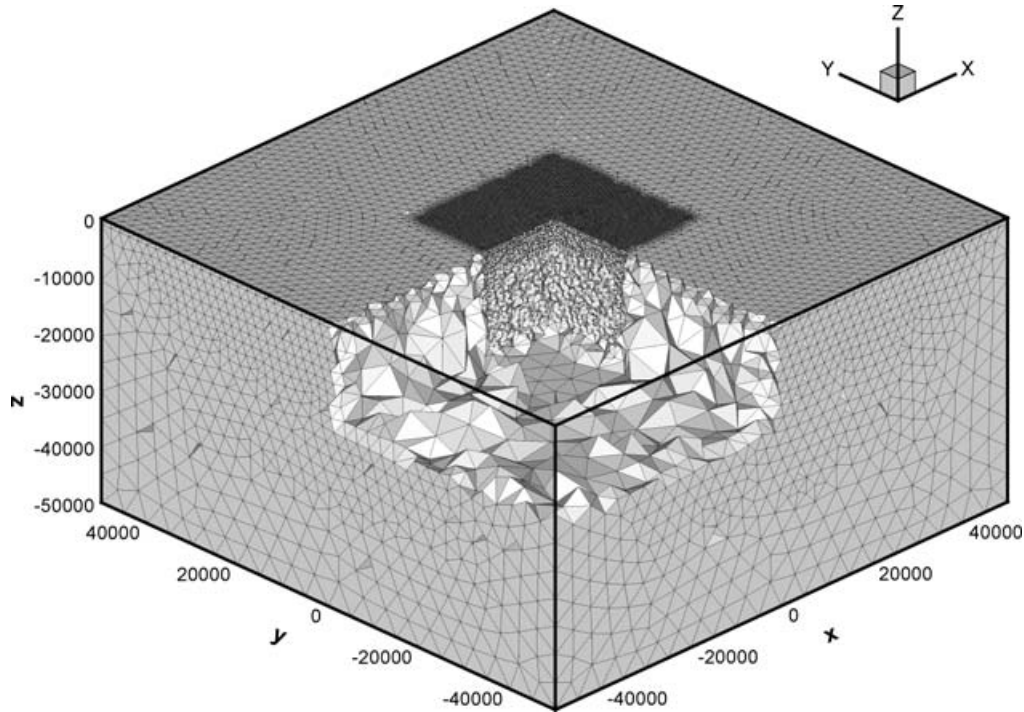
We apply the proposed ADER-DG method on a well-defined 3-D test problem, which was published in the final report of the *LIFELINES PROGRAM TASK 1A01* (Day 2001) of the Pacific Earthquake Engineering Research Center. The test case is part of a multi-institutional code validation project of a series of different numerical methods employed in numerical modelling of earthquake ground motion in 3-D earth models. 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.

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



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

	$c_p$ (m/s)	$c_s$ (m s <sup>-1</sup> )	$\rho$ (kg m <sup>-3</sup> )	$\lambda$ (GPa)	$\mu$ (GPa)
Medium 1	4000	2000	2600	20.8	10.4
Medium 2	6000	3464	2700	32.4	32.4


**Figure 5.** Cut into the discretization of the LOH.1 model to visualize the tetrahedral mesh, which is extended with large elements in order to provide better absorbing boundary conditions.

- (ii) UCSB (Kim Olsen, University of California, Santa Barbara).
- (iii) WCC1 (Robert Graves, URS Corporation).
- (iv) WCC2 (Arben Pitarka, URS Corporation).
- (v) 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 is a frequency-wavenumber solution obtained by a modification of the method presented in Luco & Apsel (1983) and Apsel & Luco (1983) and is compared to all numerical solutions to evaluate their accuracy. The setup of the test problem LOH.1 is given in detail in Dumbser & Käser (2006). The material parameters of the layer (Medium 1) of the top 1000 m and the half-space (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  $M_{xy} = M_{yx} = M_0 = 10^{18}$  N m. The location of the point source is  $(x_s, y_s, z_s) = (0 \text{ m}, 0 \text{ m}, 2000 \text{ m})$ , that is, in the centre of the  $xy$  plane of the domain  $\Omega$  in 2000 m depth.

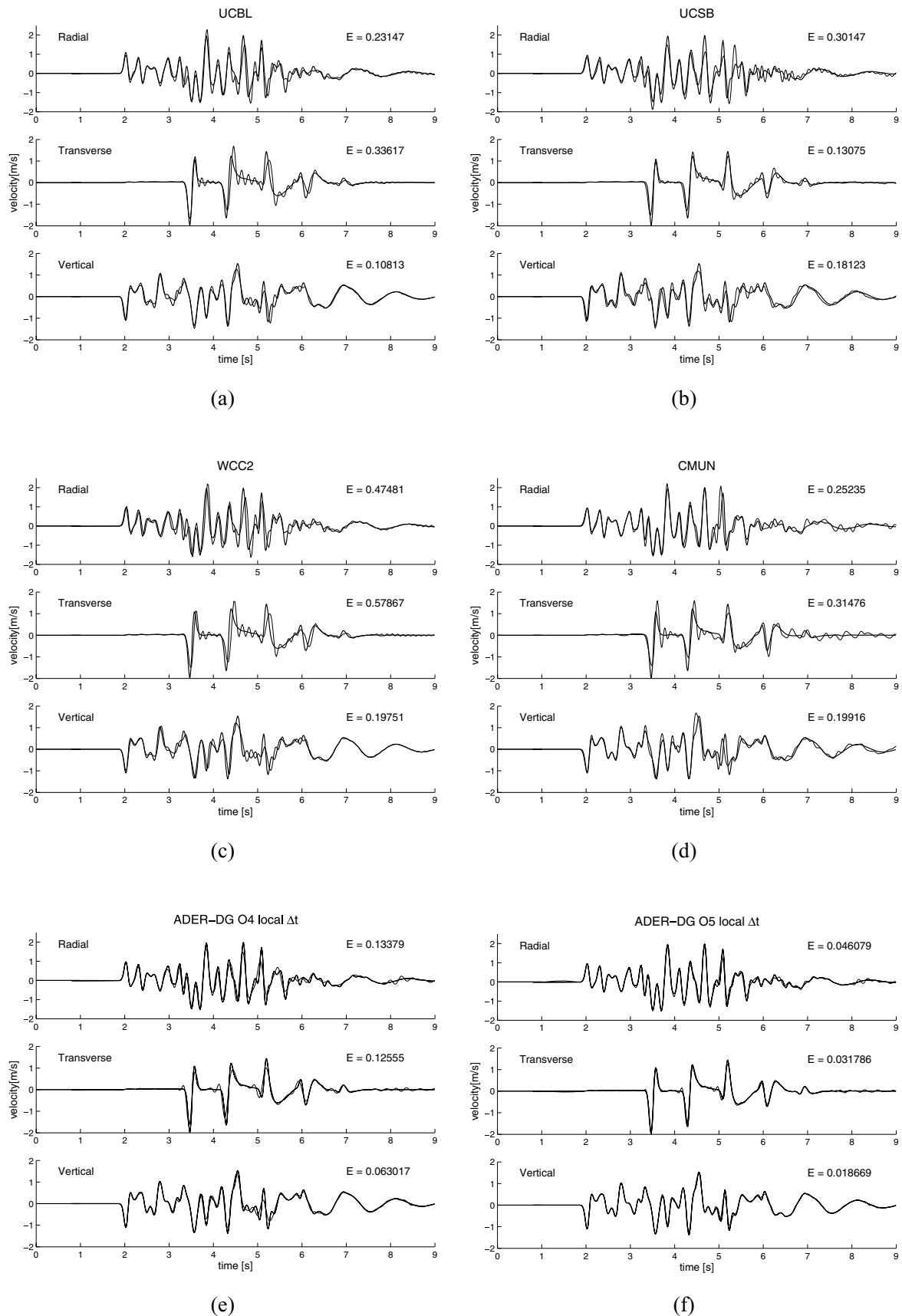
The moment-rate time history is given through the source time function

$$S^T(t) = \frac{t}{T^2} \exp\left(-\frac{t}{T}\right), \quad (33)$$

where the smoothness parameter  $T$ , controlling the frequency content and amplitude of the source time function, is set to  $T = 0.1$  s. We remark, that details of the discretization of external source terms in the framework of ADER-DG methods are outlined in previous work (Käser & Dumbser 2006). The signals are recorded up to a simulation time of 9 s by 10 receivers on the free surface. The receiver locations are  $(x_i, y_i, z_i) = (i \cdot 600 \text{ m}, i \cdot 800 \text{ m}, 0 \text{ m})$ , for  $i = 1, \dots, 10$ .

The computational domain  $\Omega$  is discretized by an unstructured tetrahedral mesh as shown in Fig. 5 using 776 523 elements. Furthermore, the mesh is generated in a special manner to reduce spurious reflections from the domain boundaries. To this end, the original computational domain  $\Omega = [-15\,000 \text{ m}, 15\,000 \text{ m}] \times [-15\,000 \text{ m}, 15\,000 \text{ m}] \times [0 \text{ m}, 17\,000 \text{ m}]$  is discretized with elements using an average edge length of





**Figure 6.** 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) LTS ADER-DG O4 and (f) LTS ADER-DG O5. The relative seismogram misfit  $E$  from eq. (34) is given for each trace.

350 m at the surface and 900 m at the bottom of the model. Then, in order to reduce spurious reflections from the boundary, the computational domain is embedded in a much larger extended domain  $\Omega_{\text{ext}} = [-50\,000\text{ m}, 50\,000\text{ m}] \times [-50\,000\text{ m}, 50\,000\text{ m}] \times [0\text{ m}, 25\,000\text{ m}]$  that is discretized using very coarse tetrahedrons of average edge length 2000 m. Since the large additional elements (about 23 per cent of the total number of elements) allow also large time steps, the induced additional computational effort is only quite small (about 16 per cent more CPU time) due to the use of local time stepping. We note that 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 Fig. 5.

In the following, we present the comparison of our results obtained with LTS ADER-DG  $\mathcal{O}4$  and LTS ADER-DG  $\mathcal{O}5$  and the best four results of the reference codes (UCBL, UCSB, WCC2 and CMUN) against the analytical solution. We stress that the maximal polynomial degree of the basis functions for ADER-DG  $\mathcal{O}4$  is  $N = 3$  so that the chosen mesh resolution for the smallest average tetrahedral edge length of 350 m corresponds to the requirements of the LOH.1 benchmark which asked for a characteristic mesh size of 100 m. Analogous to the LOH.1 test case in the *LIFELINES PROGRAM TASK 1A01* the visual comparisons in Fig. 6 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

$$E = \frac{\sum_{j=1}^{n_t} (s_j - s_j^a)^2}{\sum_{j=1}^{n_t} (s_j^a)^2}, \tag{34}$$

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 6(a)–(f), all numerical solutions (thin line) reproduce the analytical solution (thick line) with various degrees of discrepancy. However, the four reference solutions shown in Figs 6(a)–(d) produce unphysical 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 LTS ADER-DG  $\mathcal{O}4$  scheme and even more by LTS ADER-DG  $\mathcal{O}5$ . Furthermore, overshoots and phase errors obtained by all reference codes and mainly responsible for the relative seismogram misfit in (34), are decreased by the ADER-DG methods as shown in 6(e) and (f), which leads to a more precise match of the numerical and analytical solutions. We remark that with ADER-DG schemes using global time stepping as presented in Dumbser & Käser (2006), oscillations were visible in the seismograms after 6.5 s due to boundary effects. The LTS ADER-DG schemes presented in this article can handle the extended computational domain  $\Omega_{\text{ext}}$  shown in Fig. 5 in order to obtain better non-reflecting boundary conditions without much additional CPU effort since the additional domain contains only a relatively small number of very large elements that use large time steps and, therefore, add only a small computational overhead. The seismograms presented in this article indeed show a considerable decrease of these oscillations, see Figs 6(e) and (f).

Considering the CPU-times a comparison turned out to be difficult, as no CPU-time data was available from the the final report of the *LIFELINES PROGRAM TASK 1A01* (Day 2001) of the Pacific Earthquake Engineering Research Center. Furthermore, all reference codes have been run on different machines and with different levels of parallelisation. However, to give a precise indication of the CPU-time requirements for our LTS ADER-DG $\mathcal{O}4$  simulation: the run was carried out on the HLRB2 supercomputer of the *Leibniz Rechenzentrum* in München, Germany, and took 5.9 h wall-clock time on 128 Intel Itanium2 Madison 9M processors, each with 1.6 GHz and 4.0 GB of RAM. In comparison, the same computation with ADER-DG $\mathcal{O}4$  using global time stepping was 12 h with the same hardware configuration.

### 5.2 Stiff 3-D inclusion

After having validated the LTS ADER-DG scheme with the convergence studies and the LOH.1 test case, in this section we consider a 3-D extension of the inclusion test case given in LeVeque (2002) and Käser & Dumbser (2006) to show the efficiency and the high flexibility provided by a  $p_\tau$ -adaptive LTS ADER-DG scheme on a geometrically and computationally more challenging example. The computational domain of this test consists in a cylinder with radius  $R = 1$  and length  $L = 2$ , into which is embedded a circular cone with length  $l = 1$ , bottom radius  $r_1 = 0.1$  and tip radius  $r_2 = 0.001$ . Both embedded bodies are aligned with the  $z$ -axis. The material properties are  $\lambda_i = 200$  and  $\mu_i = 100$  inside the embedded cone and  $\lambda_o = 2$  and  $\mu_o = 1$  in the rest of the cylinder. The density is constant  $\rho = 1$  everywhere. This choice of the material parameters leads to ten times higher  $P$ - and  $S$ -wave velocities inside the cone-shaped inclusion, inducing automatically a much more severe restriction for the time step than in the outer material. Furthermore, the mesh needs to be refined heavily towards the tip of the cone in order to resolve the geometry appropriately, see Fig. 7 where a cut through the whole mesh of the problem containing 172 836 elements is depicted. The mesh spacing outside the inclusion is  $h_{\text{max}} = 0.045$  and at the tip of the cone it is  $h_{\text{min}} = 0.00157$  in order to resolve the tip of the cone with four mesh edges. This leads to a mesh-size ratio between largest and smallest element of about  $h_{\text{max}}/h_{\text{min}} = 28.6$ .

This refinement leads to an additional decrease of the time step in the tip of the cone with respect to the other elements in the inclusion. Using a global time step and a constant polynomial degree throughout the whole computational domain would lead to a ratio of smallest time step, arising in the tip of the cone, to largest time step, occurring in the outer region, of about 1:300. It is obvious that such a high discrepancy in time steps leads to an unreasonably high computational effort, especially for a 3-D application. Since in the stiff cone-shaped inclusion the wave speeds are much higher than in the outer material, one expects much larger wavelengths in the inclusion than in the rest of the cylinder. This effect was already confirmed in the 2-D setting (see LeVeque 2002; Käser & Dumbser 2006). Therefore, it is adequate to use a lower order approximation in the inclusion since larger wavelengths need less spatial resolution. The benefits of this lower order approximation in

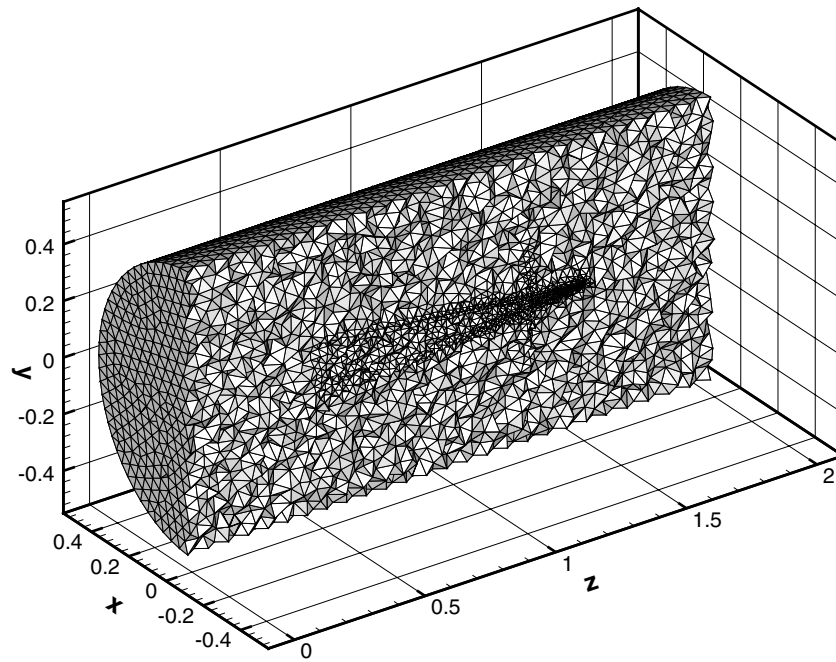


Figure 7. Tetrahedral mesh used for the 3-D inclusion test case.

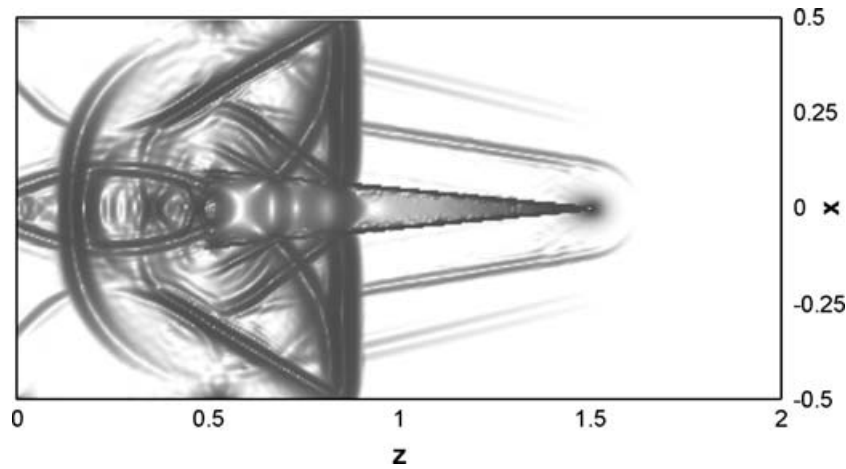


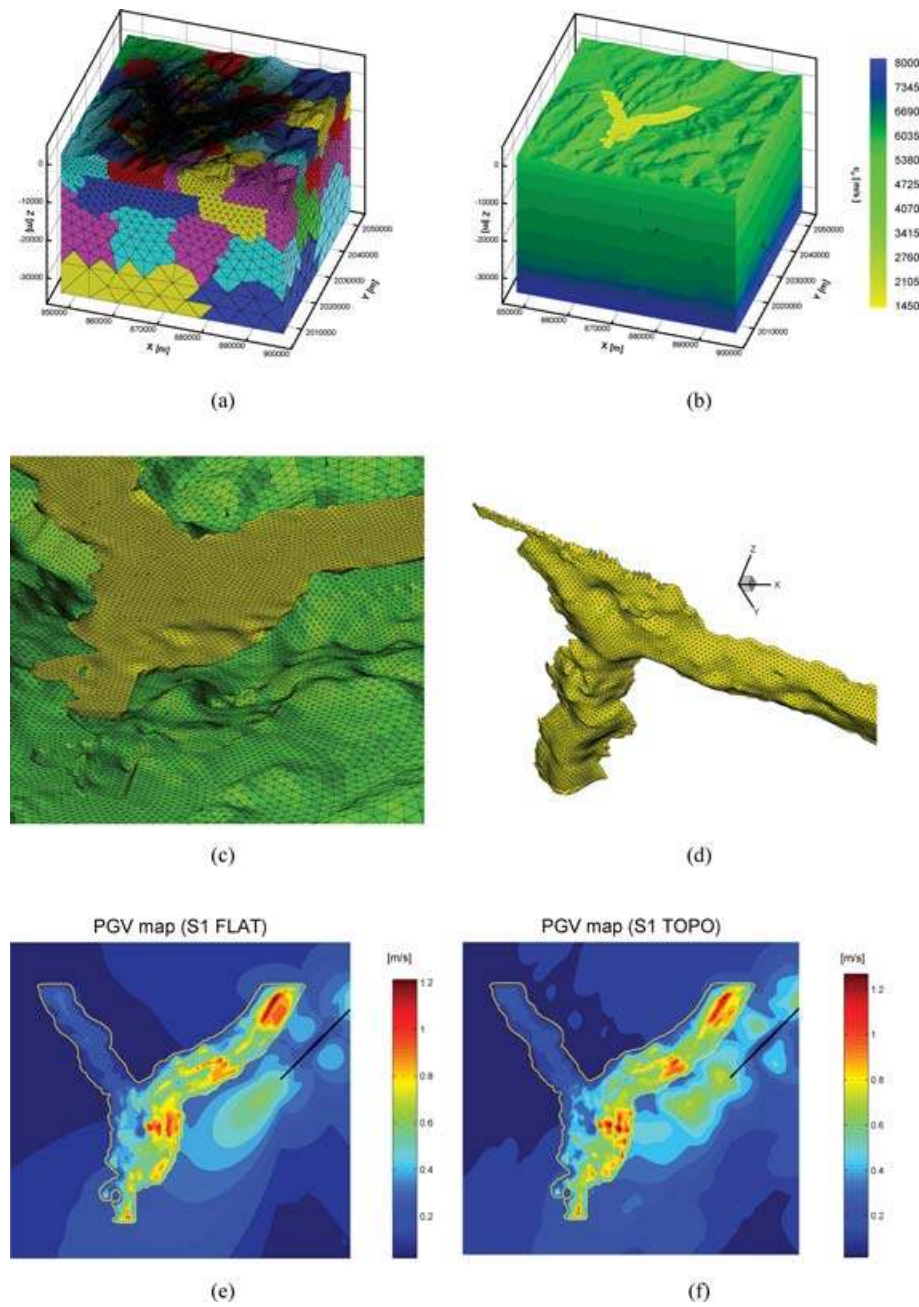
Figure 8. Stress component  $\sigma_{zz}$  at  $t = 0.3$  obtained with the  $p_\tau$ -adaptive LTS ADER-DG scheme using P2 and P5 elements.

the interior of the cone together with a LTS algorithm are twofold: first, the use of a lower order ADER-DG scheme in the inclusion leads to a larger time step since the time step of DG schemes decreases with the degree of the basis polynomials  $N$  as  $1/(2N + 1)$  according to eq. (9). Second, using the LTS ADER-DG scheme, the small tetrahedrons in the inclusion will do much more time steps than the tetrahedrons in the outer material, so it is convenient to use a cheaper method in this part of the domain where most of the cycles are expected. In our particular setting, we use a third order method ( $N = 2$ ) inside the inclusion and a sixth order method ( $N = 5$ ) in the rest of the domain. The initial condition is given by a planar Gaussian-shaped  $P$ -wave travelling into positive  $z$ -direction:

$$u_p(\vec{x}, 0) = R_{p9}^{A_n} \exp\left[-\frac{1}{2} \frac{(x - x_0)^2}{\sigma^2}\right], \quad (35)$$

Here, the normal vector is  $\vec{n} = (0, 0, 1)^T$ , the initial location of the centre of the Gaussian distribution is  $x_0 = 0.25$ , the halfwidth is chosen as  $\sigma = 0.03$  and  $R_{p9}^{A_n}$  is the eigenvector associated with the corresponding  $P$  wave. This leads to an average resolution of the incident plane wave of  $2\sigma/h_{\max} = 1.3$  elements. We compute the problem until  $t = 0.3$ . The wall clock time needed by the  $p_\tau$ -adaptive LTS ADER-DG scheme on the HLRB2 supercomputer of the LRZ in München, Germany, was 57 h using 128 Intel Itanium 2 Madison CPUs, each with 1.6 GHz and 4 GB of RAM. The total number of element updates necessary to reach this output time using different ADER-DG schemes are as follows.

- (i)  $71.665 \times 10^9$  using a purely sixth order ADER-DG scheme ( $N = 5$ ) with *global* time stepping.
- (ii)  $32.575 \times 10^9$  using a  $p_\tau$ -adaptive ADER-DG scheme ( $N = 2$  and 5) with *global* time stepping.



**Figure 9.** Visualization of the tetrahedral mesh and its partition into 64 subdomains for the Grenoble valley benchmark with topography (a). Topography and velocity structure of the model (b). Zoom of an exploded view of the sediment/bedrock interface and its tetrahedrization (c). View of the bottom topography of the sedimentary basin from below (d). Computed peak ground velocity in the Grenoble valley for the model *without* topography (e) and *with* topography (f).

- (iii)  $954.47 \times 10^6$  using a purely sixth order ADER-DG scheme ( $N = 5$ ) with *local* time stepping.
- (iv)  $573.84 \times 10^6$  using a  $p_\tau$ -adaptive ADER-DG scheme ( $N = 2$  and  $5$ ) with *local* time stepping.

This leads to a CPU time reduction of a factor greater than 125 for the  $p_\tau$ -adaptive LTS ADER-DG method compared to the use of a sixth order ADER-DG scheme with global time stepping everywhere, considering only the number of element updates. Further CPU savings for the  $p_\tau$ -adaptive scheme are due to the fact that third order elements are cheaper than sixth order elements.

In Fig. 8, we show a cut through the computational domain in the  $x - z$  plane at  $y = 0$  for time  $t = 0.3$ . The stress component  $\sigma_{zz}$  obtained with the  $p_\tau$ -adaptive LTS ADER-DG scheme is depicted. Similar to the 2-D results shown in LeVeque (2002) and Käser & Dumbser (2006) we can see the direct  $P$  wave, the Rayleigh waves at the outer cylinder surface and the shear waves emerging from the inclusion due to the reflection of the  $P$  wave inside the cone because of the large difference of the material properties.

We are convinced that this test case is a very challenging one for all explicit numerical methods that have to obey a global time step restriction. We emphasize furthermore that already the mesh generation for this test case using methods based on hexahedral grids is not a



**Table 5.** Material parameters of the bedrock of the Grenoble valley benchmark.

Depth of layer top (km)	$c_p$ (m s <sup>-1</sup> )	$c_s$ (m s <sup>-1</sup> )	$\rho$ (kg m <sup>-3</sup> )	$Q$ -factor
0	5600	3200	2720	$\infty$
3	5920	3430	2720	$\infty$
27	6600	3810	2920	$\infty$
35	8000	4450	3320	$\infty$

**Table 6.** Material parameters of the sedimentary basin of the Grenoble valley benchmark.

$c_p$	=	$1450.0 + 1.2 d$
$c_s$	=	$300.0 + 19.0 \sqrt{d}$
$\rho$	=	$2140.0 + 0.125 d$
$Q_p$	=	$(3c_p/4c_s^2) Q_s$
$Q_s$	=	50

trivial task. To match the inner grid inside the cone with the outer grid of the cylinder can become difficult with structured and unstructured hexahedral grids, even when using modern commercial mesh generators.

### 5.3 Earthquake modelling and ground motion prediction

In this section, we apply the proposed ADER-DG scheme with LTS to a benchmark test of earthquake modelling to confirm the performance and the functionality of this approach for real-world applications. Thus, we chose the ESG 2006 benchmark for ground motion simulation in the Grenoble valley, where details are given in Chaljub (2006). It is well known, that alpine valleys produce strong site effects due to the contact and large impedance contrasts between solid bedrock and less consolidated sediments as shown in Bard & Buchon (1980a,b). Furthermore, alpine valleys usually exhibit strong topographic variations of the free surface and of the internal boundary between the bedrock and the sedimentary basin.

The setup of this test case provides a velocity model of  $50.4 \times 47.4$  km horizontal extent and 35 km depth, as shown in Fig. 9(a) and (b). The detailed topography information is given by a digital elevation model of 50 m lateral resolution. To discretize the computational domain we use an unstructured tetrahedral mesh of 1 259 721 elements, see Fig. 9(a), that respects the free surface topography and the internal material boundaries, in particular, the geometrically complicated interface between the sedimentary basin and the bedrock material. Furthermore, the mesh is refined locally in the zones of interest and is coarsened with increasing depth and towards the model boundaries as illustrated in Fig. 9(a). The sedimentary basin under Grenoble appears as an Y-shaped structure on the surface as shown in Fig. 9(b). The colour code in Fig. 9(b) displays the  $P$ -wave velocity distribution. Details about the velocity model of the bedrock are given in Table 5. The material parameters inside a bedrock layer vary linearly with depth.

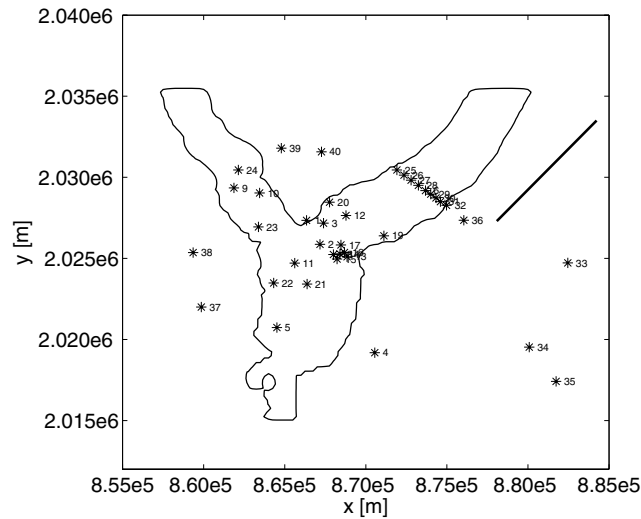
An enlarged illustration of the 3-D shape of the sedimentary basin and its discretization by an unstructured, tetrahedral mesh is displayed in Fig. 9(c). Note, that in this plot we show an exploded view, such that the sediment is detached from the bedrock in order to see the internal interface between the two geological zones. The depth-dependent distribution of the material parameters inside the sedimentary basin are given in Table 6, where the unit of depth  $d$  is (m), the one of the velocities  $c_p$  and  $c_s$  is (m s<sup>-1</sup>) and of the density  $\rho$  is (kg m<sup>-3</sup>).

Fig. 9(d) represents a view from below the sedimentary basin in order to display in more detail the topography of the sediment/bedrock interface together with a cut to better visualize the tetrahedral elements of the basin discretization. We note, that the tetrahedral elements discretizing the basin have an average tetrahedron edge length of 200 m. In comparison to the tetrahedral elements at 35 km depth with average edge lengths of up to 6200 m, see Fig. 9(a), the relation of the local mesh spacing yields a factor of 31. We remark, that a global time stepping scheme would have to update these large tetrahedrons many times according to the minimum time step that occurs in the entire computational domain. Our LTS scheme, however, uses the maximum time step allowed for these large tetrahedral elements. We point out, that this maximum time step also takes into account the local material parameters, that is, the local seismic wave velocities, and the local polynomial degree  $N$  of the approximation polynomial. Therefore, the computational effort can be reduced considerably.

**Table 7.** Comparison of different numerical methods and meshing strategies applied to the Grenoble valley test case with topography and the associated time requirements for the meshing process and for the wave propagation simulation.

Numerical method	Mesh topology	Domain size	Number of elements	Number of CPUs	Wallclock time (h)
SEM (Chaljub)	hexahedrons	31.0x 24.0x50.0 km <sup>3</sup>	332 160	32	7
ADER-DG (Dumbser <i>et al.</i> )	tetrahedrons	50.4x47.4x35.0 km <sup>3</sup>	1259 721	255	50





**Figure 10.** Spatial location and numbering of the receivers for the Grenoble valley benchmark problem. The projection of the lateral extent of the sedimentary basin (thin solid line) as well as of the rupture plane (thick solid line) are also shown.

As seismic attenuation has to be considered inside the basin structure, the attenuation properties are approximated by a viscoelastic material using three attenuation mechanisms in the frequency band from 0.1 to 10 Hz as introduced by Emmerich & Korn (1987) and described in detail within the framework of ADER-DG schemes in Käser *et al.* (2007).

In our simulation, we consider the strong motion case, where the parameters of an extended rupture source describe a right-lateral strike-slip event of moment magnitude  $M_w = 6.0$  on the eastern part of the Belledonne border fault as given in Chaljub (2006). The kinematics of the rupture is represented by a circular crack propagating inside a rectangular fault, that is, a Haskell crack with constant final displacement, with a rupture velocity of  $2.8 \text{ km s}^{-1}$ . The nucleation point is in 3 km depth and in the centre of a vertical fault plane of 9 km length and 4.5 km width with orientation: strike =  $45^\circ$ , dip =  $90^\circ$  and rake =  $180^\circ$ . The rupture plane is subdivided into  $15 \times 30 = 450$  subfaults of size  $300 \times 300 \text{ m}$ . The source is implemented via a seismic moment tensor describing a double couple source with the source time function of the displacement rate

$$S^T(t) = \frac{T}{\sqrt{\pi}} \exp\{-[T(t - t_D)]^2\}, \quad (36)$$

with smoothness factor  $T = 3.5842$  and delay time  $t_D = 2.232 \text{ s}$  as described in Chaljub (2006). A map of locations of receivers on the free surface of the model and the projection of the rupture plane onto the surface with respect to the shape and position of the sedimentary basin is shown in Fig. 10.

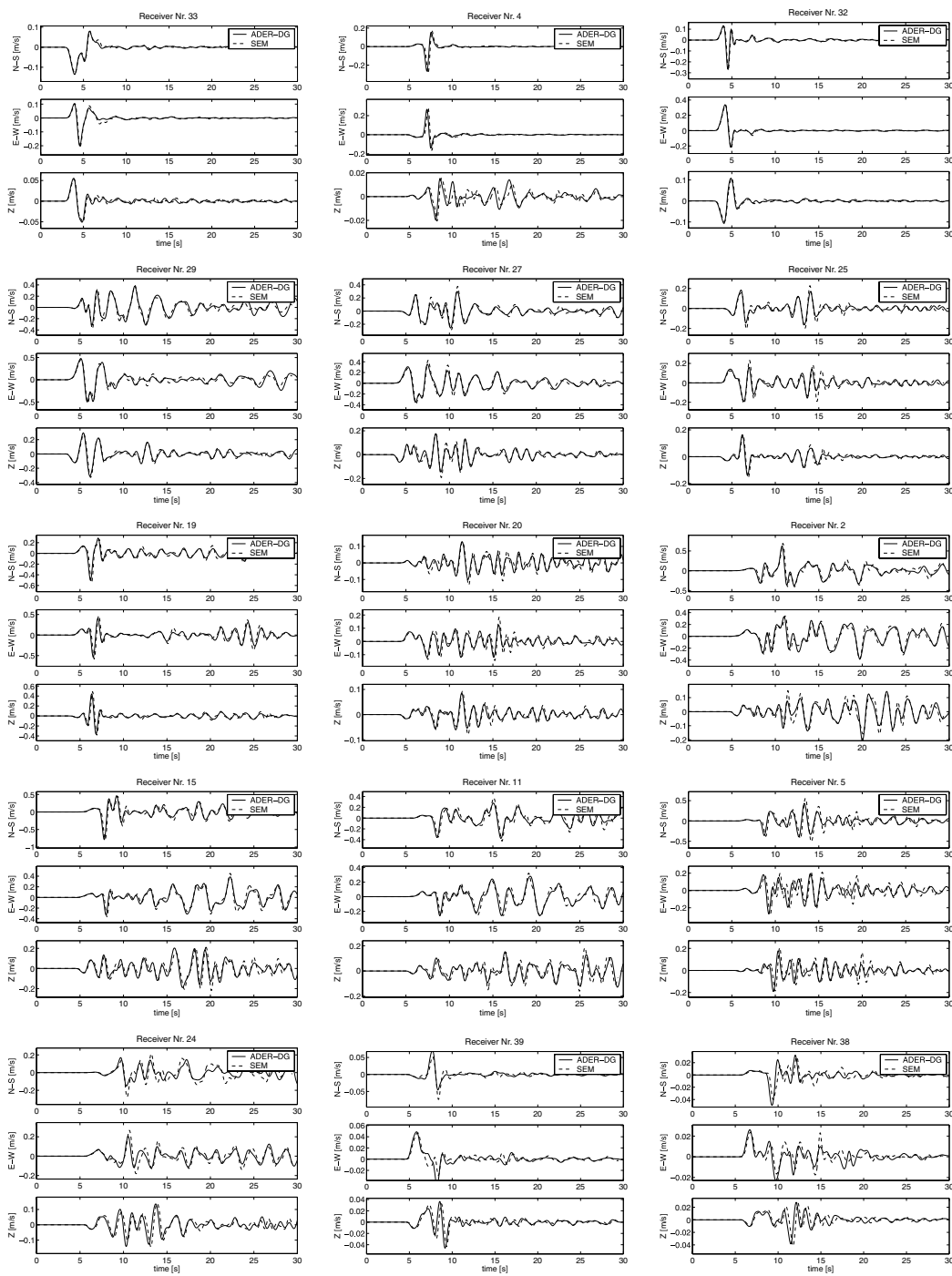
For the simulation, we decompose the model into four geometrical, geological zones: (1) the basin structure, (2) the surface layer above 3 km depth excluding the basin, (3) the layer between 3 and 27 km depth and (4) the layer between 27 and 35 km depth. We use the LTS ADER-DG scheme with  $p_\tau$ -adaptivity, where the following distribution of the polynomial degree  $N$  is applied: zone (1)  $N = 3$ , zone (2)  $3 \leq N \leq 4$ , zone (3)  $3 \leq N \leq 4$  and zone (4)  $N = 2$ . The mesh is partitioned into 255 subdomains for parallel computing as shown in Fig. 9(a). The total simulation time is set to 30 s.

According to the benchmark proposed by Chaljub (2006), the computation is done first without surface topography but including the basin structure. This model is in the following also denoted as the flat model. In the second computation, also the complex surface topography is included. These two computations allow an assessment of the effects of surface topography on the seismograms and the observed peak ground velocity.

In Figs 9(e) and (f) we present the maps of the peak ground velocity recorded by an array of  $120 \times 120$  receivers with a spatial interval of 250 m on the surface across the area of the Grenoble valley for the flat test case and the model with surface topography. The strongest ground velocity is predicted for the southeastern part of the Grenoble valley at the contact between the soft sediments and the solid bedrock, which agrees with observed amplification effects of sedimentary basins in Chaljub *et al.* (2005). The well-known focusing effect of waves along the strike direction of a rupture fault can be seen clearly, as the area of the sediment/bedrock contact closer but perpendicular to the rupture plane exhibits ground velocity of much smaller amplitude. Furthermore, the sedimentary basin acts as a shield leading to rather weak ground motion on the opposite (western) side of the valley.

Comparing Figs 9(e) and (f) it can be seen clearly that for the considered seismic event, surface topography has no large impact on the peak ground motion since the observed maximal amplitudes are more or less the same in both cases.

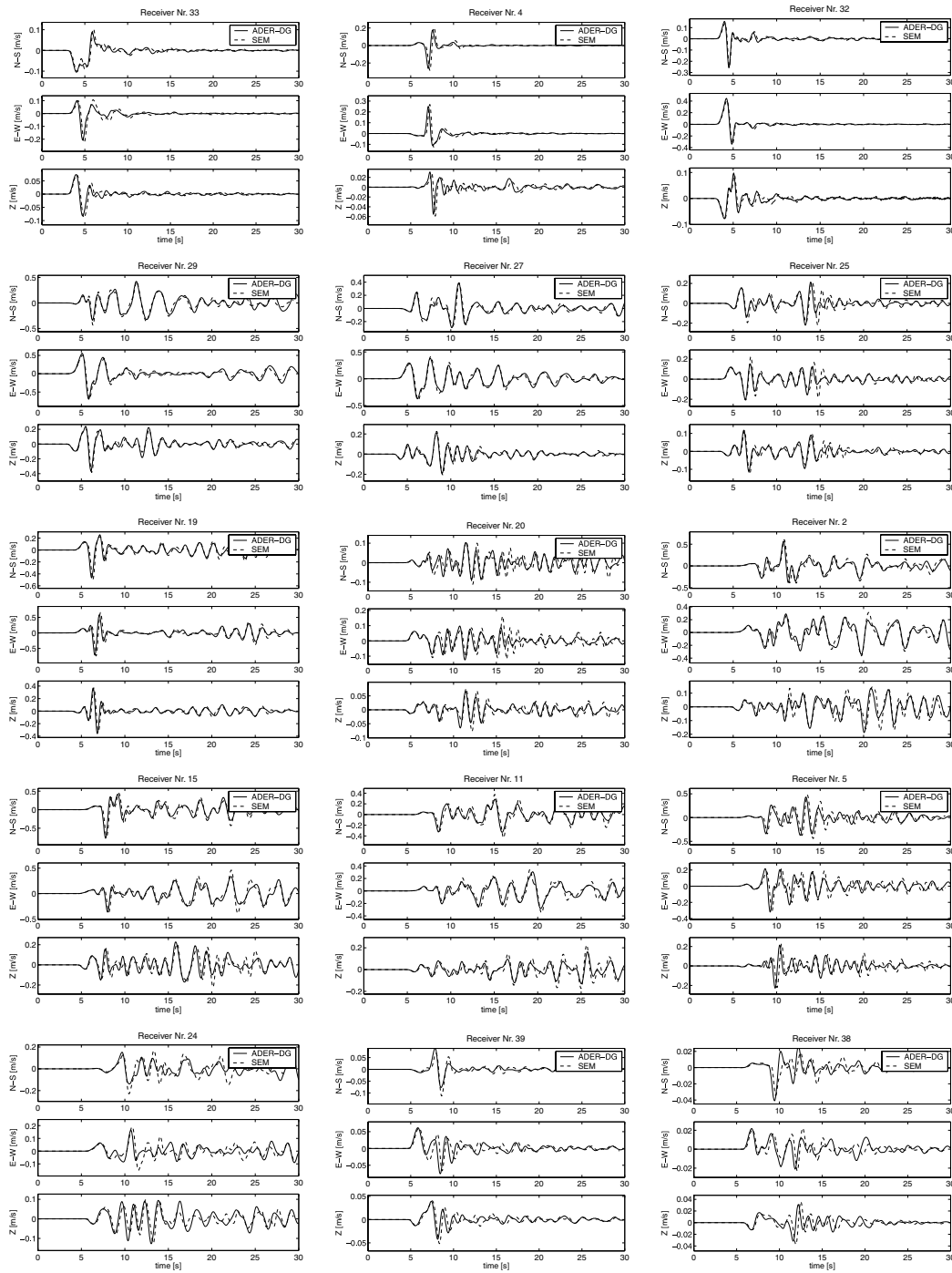
In Figs 11 and 12 we display the *unscaled* and *unfiltered* seismograms for the three velocity components in  $x$ ,  $y$  and  $z$ -directions recorded on 15 of the receiver locations shown in 10 for the flat model and the model with surface topography. We note that in both cases receivers 4, 32, 33, 38 and 39 show much smaller velocity amplitudes than the other receivers as they are located on bedrock outside the basin. Except the amplitude amplification the signals recorded in the basin show longer duration of oscillations due to the multiple reflections of the waves



**Figure 11.** Seismograms obtained with the  $p_\tau$ -adaptive LTS ADER-DG method compared with the SEM reference solution of Chaljub *et al.* showing the three velocity components on 15 selected receivers of the *flat model* for the Grenoble valley benchmark problem.

trapped inside the basin. These effects are in good correlation with other simulations and observations in this area as shown in Chaljub *et al.* (2005). In agreement with the peak ground velocity maps discussed previously we observe by comparing the seismogram data for the flat model and the surface topography model at the selected receiver locations that for this test case there are only small changes in the seismograms due to surface topography effects.

As a reference, the *unscaled* and *unfiltered* seismogram data obtained by E. Chaljub using a SEM method for both models are also plotted for all considered receiver locations. In both cases we note in general a very good agreement between the SEM reference seismograms and our LTS ADER-DG simulations. Not only with respect to amplitudes and phases, but even the individual waveforms are very similar at all receiver locations. Considering the geometrical complexity of this test case with a non-trivial source, such a high level of agreement between two completely different numerical methods running on completely different mesh topologies and model setups, as given in Table 7, is very



**Figure 12.** Seismograms obtained with the  $p_\tau$ -adaptive LTS ADER-DG method compared with the SEM reference solution of Chaljub *et al.* showing the three velocity components on 15 selected receivers of the *model with surface topography* for the Grenoble valley benchmark problem.

satisfactory for both methods and underlines the reliability of modern numerical modelling tools in complex geophysical application cases of practical relevance.

## 6 CONCLUDING REMARKS

We have presented an arbitrary high order DG scheme on unstructured tetrahedral meshes that may adapt the local polynomial degree as well as the local time step in a problem-dependent manner. The apparent main advantage of the proposed LTS ADER-DG scheme is its extreme flexibility considering its ability to deal with complex geometry because it is able to run on tetrahedral meshes as well as the adaptivity of the underlying numerical algorithm.

Since our numerical method is based on a completely different mesh topology compared to other methods such as FD or SEM, a direct and fair computational cost comparison is very difficult. We especially point out that in order to mesh two equal computational domains with a similar mesh spacing, that is, with the same edge length of elements, one needs about six times more tetrahedral elements compared to a hexahedral mesh. At the same time, tetrahedrons require a more severe time step restriction (between two and three times smaller time steps) compared to a regular hexahedral grid. Hence, the pure computational cost of our method is in general *high* because of the underlying tetrahedral mesh topology. Even the LTS version of our ADER-DG scheme is still slower compared to standard FD or SEM methods.

However, we emphasize that tetrahedral mesh generation is almost completely automatic, even for very complex geometries, which definitely induces the advantage of *flexibility* for our method. All test cases presented in this article were computed with the same code, where the model can be almost completely set up already in the tetrahedral mesh generator. We use the GAMBIT software package, available at the HLRS supercomputing centre in Stuttgart, which allows to define so-called volume zones. For each zone, the local material properties ( $\lambda$ ,  $\mu$ ,  $\rho$ , as well as  $Q_p$  and  $Q_s$  for viscoelastic material) are defined, as well as the desired local polynomial degree  $N$  of the basis functions. Also the boundary conditions (free surface, periodic or open boundaries) can be directly specified in the mesh generator. The entire generation of the tetrahedral mesh for the inclusion testcase shown in Fig. 7 for the LTS ADER-DG scheme can be done *completely automatic* in about 15 s on a standard Intel Xeon Workstation with 3.6 GHz and 4GB of RAM. To generate the entire 3-D model and the tetrahedral mesh for the Grenoble valley test case including surface topography as shown in Fig. 9(a), the GAMBIT mesh generator needed only 165 s without any manual interactions. In contrast to this, the generation of hexahedral grids as shown in Stupazzini (2006) for the same testcase needs many manual interactions and thus the mesh generation process for such hexahedral meshes can easily take several hours, even for experienced users. Taking into account also the time needed for mesh generation, the LTS ADER-DG scheme seems to be a very flexible and practical alternative to the current approaches, such as FD and SE schemes, which require more time-consuming hexahedral mesh generation. If the computational model contains a very complex geometry that is difficult to mesh with hexahedral grids and if the computational model leads to very different time steps, such as the presented Grenoble valley benchmark, the LTS ADER-DG scheme may offer advantages due to its flexibility. In other cases, such as earthquake simulation on the global scale, where the underlying geometry is basically a sphere, the SEM seems to be preferable.

Furthermore, the proposed LTS ADER-DG scheme is very robust concerning mesh quality. As we can see from the convergence studies presented in Section 4, the error norms and CPU times are comparable for both the irregular mesh containing tetrahedrons of very different size as well as for the regular mesh containing the four slivers. This robustness is required for applications with complex geometries in which the annoying problem of slivers (Joe 1995; Bern & Eppstein 1992; Fleischmann *et al.* 1999) may occur during mesh generation. Such slivers, if they cannot be avoided, not only reduce the solution quality but also considerably decrease the time step of global time stepping schemes, which adds an unnecessary amount of CPU time.

In our opinion, a future main application of the LTS algorithm proposed in this article could be the embedding of small local structures of interest for civil engineering such as bridges, towers or other buildings into an earthquake scenario on the regional scale, where the computational model may be considerably enriched within the region of interest, for example, taking into account subsurface sediment structures and surface topography. Since the small refined local region of interest does not degenerate the time step used on the global scale, the global and the local wave fields could be computed simultaneously and in a *fully coupled* manner. A second major application could be the computation of elastic waves in complex layered sediment structures where the sediments have very different material properties and thus inducing very different time steps. The third future application may concern dynamic rupturing processes where the dynamic computation of the rupture fault may be done on a locally strongly refined mesh, requiring very small local time steps. This dynamic rupture computation may then be coupled directly with the wave propagation computation into the far field on a much coarser mesh using the full benefits of the admissible large time steps for the far field computation.

## ACKNOWLEDGMENTS

The authors thank the *Deutsche Forschungsgemeinschaft* (DFG), as the work was financed by the *DFG Forschungsstipendium* (DU 1107/1-1), the DFG-CNRS research group FOR 508 and the *Emmy Noether Programm* (KA 2281/1-1).

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. The authors are especially grateful to E. Chaljub for carefully setting up the Grenoble valley benchmark problem and for providing all the necessary simulation parameters as well as the numerical reference solution.

Many thanks also to the Leibniz Rechenzentrum in München, Germany and the HLRS supercomputing centre in Stuttgart for providing the GAMBIT mesh generator and the necessary hardware to run the demanding 3-D test cases.

Finally we would like to thank the two referees of this paper whose constructive comments and suggestions definitely helped to improve the clarity and the quality of this article.

## REFERENCES

- Aki, K. & Richards, P.G., 2002. *Quantitative Seismology*, University Science Books, Sausalito, California, USA.
- Apsel, R.J. & Luco, J.E., 1983. On the Green's functions for a layered half-space, Part II, *Bull. seism. Soc. Am.*, **73**, 931–951.
- Bard, P.-Y. & Bouchon, M., 1980a. The seismic response of sediment-filled valleys, Part 1: The case of incident SH waves, *Bull. seism. Soc. Am.*, **70**, 1263–1286.
- Bard, P.-Y. & Bouchon, M., 1980b. The seismic response of sediment-filled valleys, Part 2: The case of incident P and SV waves, *Bull. seism. Soc. Am.*, **70**, 1921–1941.

- Bedford, A. & Drumheller, D.S., 1994. *Elastic Wave Propagation*, John Wiley & Sons, Chichester, UK.
- Bern, M. and Eppstein, D, 1992. Mesh Generation and Optimal Triangulation, in *Computing in Euclidean Geometry*, pp. 201–204, World Scientific, Singapore.
- Butcher, J.C., 1987. *The Numerical Analysis of Ordinary Differential Equations: Runge-Kutta and General Linear Methods*, John Wiley & Sons, New York.
- Chaljub, E., Cornou, C., Guéguen, P., Causse, M. & Komatitsch, D., 2005. Spectral-element modeling of 3D wave propagation in the alpine valley of Grenoble, France, in *Geophysical Research Abstracts*, Vol. 7, pp. 05 225, EGU 2nd General Assembly, Wien, Austria.
- Chaljub, E., 2006. Numerical Benchmark of 3D Ground Motion Simulation In the Valley of Grenoble, French Alps, <http://esg2006.obs.ujf-grenoble.fr/BENCH2/benchmark.html>.
- Cheng, S.-W., Dey, T.K., Edelsbrunner, H., Facello, M.A. & Teng, S.-H., 2000. Sliver exudation, *J. ACM* 47, 5, 883–904.
- Cockburn, B., Karniadakis, G.E. & Shu, C.W., 2000. *Discontinuous Galerkin Methods, Theory, Computation and Applications*, LNCSE, Vol. 11, Springer, Berlin, Germany.
- Day, S.M., 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.
- de la Puente, J., Käser, M., Dumbser, M. & Igel, H., 2006. An Arbitrary High Order Discontinuous Galerkin Method for Elastic Waves on Unstructured Meshes—IV. Anisotropy. *Geophys. J. Int.*, in press.
- 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., 2005. Arbitrary High Order Discontinuous Galerkin Schemes, in *Numerical Methods for Hyperbolic and Kinetic Problems*, pp. 295–333, eds Cordier, S., Goudon, T., Gutnic, M. & Sonnendrucker, E., IRMA series in mathematics and theoretical physics, EMS Publishing House.
- Dumbser, M. & Käser, M., 2006. An Arbitrary High Order Discontinuous Galerkin Method for Elastic Waves on Unstructured Meshes—II. The Three-Dimensional Isotropic Case. *Geophys. J. Int.*, 167, 316–336.
- Dumbser, M., Schwartzkopff, T. & Munz, C.D., 2006a. Arbitrary High Order Finite Volume Schemes for Linear Wave Propagation, in *Computational Science and High Performance Computing II*, pp. 129–144, eds Krause, E., Shokin, Y., Resch, M. & Shokina, N., Notes on Numerical Fluid Mechanics and Multidisciplinary Design (NNFM), Springer, Berlin/Heidelberg.
- Dumbser, M., Käser, M. & de la Puente, J., 2006b. *Arbitrary High Order Finite Volume Schemes for Seismic Wave Propagation on Unstructured Meshes in 2D and 3D*, in press.
- Edelsbrunner, H. & Guoy, D., 2002. An Experimental Study of Sliver Exudation, *Eng. Comput.*, 18, 229–240.
- Emmerich, H. & Korn, M., 1987. Incorporation of attenuation into time-domain computations of seismic wave fields, *Geophysics*, 52, 1252–1264.
- Fumeaux, C., Baumann, D., Leuchtman, P. & Vahldieck, R., 2004. A generalized local time-step scheme for efficient FVTD simulations in strongly inhomogeneous meshes, *IEEE Trans. Microwave Theory Tech.*, 52(3), 1067–1076.
- Flaherty, J., Loy, R., Shephard, M., Szymanski, B., Teresco, J. & Ziantz, L., 1997. Adaptive local refinement with octree load-balancing for the parallel solution of three-dimensional conservation laws, *J. Parallel Distrib. Comput.*, 47, 139–152.
- Fleischmann, P., Pyka, W. & Selberherr, S., 1999. Mesh Generation for Application in Technology CAD, *IEICE Trans. Electron.*, E82-C(6), 937–947.
- Joe, B., 1995. Construction of three-dimensional improved-quality triangulations using local transformations, *SIAM: J. Sci. Comput.*, 16(6), 1292–1307.
- Karypis, G. & Kumar, V., 1998. Multilevel  $k$ -way partitioning scheme for irregular graphs, *J. Parallel Distrib. Comput.*, 48(1), 96–129.
- Käser, M. & Dumbser, M., 2006. An Arbitrary high order discontinuous galerkin method for elastic waves on unstructured meshes - I. The two-dimensional isotropic case with external source terms. *Geophys. J. Int.*, 166, 855–877.
- Käser, M., Dumbser, M., de la Puente, J. & Igel, H., 2007. An Arbitrary High Order Discontinuous Galerkin Method for Elastic Waves on Unstructured Meshes—III. Viscoelastic Attenuation. *Geophys. J. Int.*, 168, 224–242.
- Komatitsch, D. & Tromp, J., 1999. Introduction to the spectral-element method for 3-D seismic wave propagation, *Geophys. J. Int.*, 139, 806–822.
- Komatitsch, D. & Tromp, J., 2002. Spectral-element simulations of global seismic wave propagation—I. Validation, *Geophys. J. Int.*, 149, 390–412.
- Komatitsch, D. & Tromp, J., 2003. A perfectly matched layer absorbing boundary condition for the second-order seismic wave equation, *Geophys. J. Int.*, 154, 146–153.
- Komatitsch, D. & Vilotte, J.P., 1998. The spectral-element method: an efficient tool to simulate the seismic response of 2D and 3D geological structures, *Bull. seism. Soc. Am.*, 88, 368–392.
- LeVeque, R.L., 2002. *Finite Volume Methods for Hyperbolic Problems*, Cambridge University Press, Cambridge.
- Loercher, F., Gassner, G. & Munz, C.D., 2007. A discontinuous Galerkin scheme based on a space–time expansion. I. Inviscid compressible flow in one space dimension. *J. Sci. Comput.*, in press.
- Luco, J.E. & Apsel, R.J., 1983. On the Green’s functions for a layered half-space, Part I, *Bull. seism. Soc. Am.*, 73, 909–929.
- Stupazzini, M., 2006. 3D Ground Motion Simulation of the Grenoble valley by GeoELSE, in proceedings of the *3rd International Symposium on the Effects of Surface Geology on Seismic Motion (ESG)*, Grenoble, France.
- Houston, P. & Süli, E., 2001. hp-Adaptive discontinuous Galerkin finite element methods for first-order hyperbolic problems, *SIAM J. Sci. Comput.*, 23(4), 1225–1251.
- Titarev, V.A. & Toro, E.F., 2002. ADER: Arbitrary high order Godunov approach, *J. Sci. Comput.*, 17, 609–618.
- Toro, E.F., Millington, A.C. & Nejad, L.A., 2001. Towards very high order Godunov schemes, in *Godunov Methods; Theory and Applications*, pp. 907–940, Kluwer Academic Plenum Publishers, Oxford.
- Toro, E.F. & Titarev, V.A., 2002. Solution of the generalized Riemann problem for advection-reaction equations, *Proc. Roy. Soc. Lond.*, pp. 271–281.
- van der Vegt, J.J.W. & van der Ven, H., 2002. Space–time discontinuous Galerkin finite element method with dynamic grid motion for inviscid compressible flows: I. General formulation, *J. Comput. Phys.*, 182(2), 546–585.
- 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.