PHILOSOPHICAL TRANSACTIONS OF THE ROYAL SOCIETY A

Phil. Trans. R. Soc. A (2009) **367**, 4613–4631
doi:10.1098/rsta.2009.0158

Space–time adaptive numerical methods for geophysical applications

BY C. E. CASTRO[1,*], M. KÄSER[1] AND E. F. TORO[2]

[1]*Department für Geo- und Umweltwissenschaften, Geophysik, Ludwig-Maximilians-Universität München, Germany*
[2]*Laboratory of Applied Mathematics, Department of Civil and Environmental Engineering, University of Trento, Italy*

In this paper we present high-order formulations of the finite volume and discontinuous Galerkin finite-element methods for wave propagation problems with a space–time adaptation technique using unstructured meshes in order to reduce computational cost without reducing accuracy. Both methods can be derived in a similar mathematical framework and are identical in their first-order version. In their extension to higher order accuracy in space and time, both methods use spatial polynomials of higher degree inside each element, a high-order solution of the generalized Riemann problem and a high-order time integration method based on the Taylor series expansion. The static adaptation strategy uses locally refined high-resolution meshes in areas with low wave speeds to improve the approximation quality. Furthermore, the time step length is chosen locally adaptive such that the solution is evolved explicitly in time by an optimal time step determined by a local stability criterion. After validating the numerical approach, both schemes are applied to geophysical wave propagation problems such as tsunami waves and seismic waves comparing the new approach with the classical global time-stepping technique. The problem of mesh partitioning for large-scale applications on multi-processor architectures is discussed and a new mesh partition approach is proposed and tested to further reduce computational cost.

Keywords: finite volumes; discontinuous Galerkin finite elements; ADER; local time stepping; seismic wave propagation; tsunami wave propagation

1. Introduction

Within the last decade many disciplines in natural and engineering sciences have extensively made use of scientific computations in addition to purely theoretical, classical experimental or observational research. Today, supercomputers are invaluable virtual laboratories, where experiments can be carried out to test theoretical mathematical formulations of physical problems, to investigate large parameter spaces, and to design and optimize models under investigation. However, the ever increasing computational resources are not enough to achieve sufficient resolution in the current Earth science problems if a simple uniform

---

*Author for correspondence (castro@geophysik.uni-muenchen.de).

One contribution of 10 to a Theme Issue 'Mesh generation and mesh adaptation for large-scale Earth-system modelling'.

increase of the discrete model resolution is employed. This is particularly true in cases where large computational domains contain local small-scale features that strongly affect the behaviour of the global solution. Therefore, different adaptive strategies have been developed to maximize the resolution to cost ratio when numerically solving partial differential equations. Adaptive mesh refinement (AMR) has been developed in different scientific communities and is used in a static or dynamic way to locally achieve high spatial resolution (e.g. Babuska *et al.* 1995; Plewa *et al.* 2000; Käser & Iske 2005; Shi *et al.* 2006). However, for explicit numerical time-integration schemes, a fine spatial resolution of the mesh limits the time-step length due to the stability condition. Therefore, local time-stepping approaches have been developed, where the size of a time step can vary locally such that small time steps only need to be used in zones of the computational domain where the spatial resolution is high or waves are fast. Such schemes in combination with first-order discontinuous Galerkin (DG) finite-element schemes on unstructured three-dimensional meshes have been proposed by Flaherty *et al.* (1997) and local time-stepping for finite volume (FV) methods on unstructured tetrahedral meshes have been developed by Fumeaux *et al.* (2004). A high-order DG scheme using the Cauchy–Kowalewski procedure for space–time expansion and local time-stepping has been introduced for linear seismic wave equations by Dumbser *et al.* (2007) and later extended to nonlinear hyperbolic–parabolic systems (e.g. Lörcher *et al.* 2007; Gassner *et al.* 2008). Despite the overall reduction of computational cost using adaptive numerical methods, their application to large-scale problems on multi-processor supercomputers still requires further developments. Owing to the adaptive character of such numerical schemes, the distribution of the computational load becomes a non-trivial task. To efficiently use the computational performance of large systems, the numerical algorithms have to scale to several thousands of processors, which requires a homogeneous load balance that can become a problematic issue, especially for complex heterogeneous numerical models.

The aim of this paper is to present high-order FV and DG schemes on unstructured meshes statically adapted to particular physical problems using a local time-step approach. Furthermore, a new partitioning strategy is introduced and its effect on the computational performance is compared for large-scale applications in tsunami and earthquake wave-propagation scenarios.

2. Numerical method

The numerical scheme is constructed considering a conforming discretization of the computational domain $\Omega = \cup \mathcal{T}_m$, with elements $\mathcal{T}_m$ identified using a unique index m. In two-dimensional study we use triangular elements, while in three-dimensional study we use tetrahedral elements. The numerical method is used to approximate the solution of the following hyperbolic balance law:

$$\partial_t \boldsymbol{U} + \partial_x \boldsymbol{F}(\boldsymbol{U}) + \partial_y \boldsymbol{G}(\boldsymbol{U}) + \partial_z \boldsymbol{H}(\boldsymbol{U}) = \boldsymbol{S}(\boldsymbol{U}), \tag{2.1}$$

with $\boldsymbol{U} = [u_1, \ldots, u_p]$ the unknown vector of conservative variables, $\boldsymbol{F}(\boldsymbol{U})$, $\boldsymbol{G}(\boldsymbol{U})$ and $\boldsymbol{H}(\boldsymbol{U})$ the flux vectors in each spatial coordinate $\vec{x} = (x, y, z)$ and $\boldsymbol{S}(\boldsymbol{U})$

a source term. The quasi-linear form of equation (2.1) reads

$$\partial_t \boldsymbol{U} + \boldsymbol{A}(\boldsymbol{U})\partial_x \boldsymbol{U} + \boldsymbol{B}(\boldsymbol{U})\partial_y \boldsymbol{U} + \boldsymbol{C}(\boldsymbol{U})\partial_z \boldsymbol{U} = \boldsymbol{S}(\boldsymbol{U}), \tag{2.2}$$

with the Jacobian matrices $\boldsymbol{A} = \partial \boldsymbol{F}/\partial \boldsymbol{U}$, $\boldsymbol{B} = \partial \boldsymbol{G}/\partial \boldsymbol{U}$ and $\boldsymbol{C} = \partial \boldsymbol{H}/\partial \boldsymbol{U}$. Equation (2.2) is said to be hyperbolic if for any direction of a normalized vector $\vec{n} = (n_x, n_y, n_z)$ the matrix $\boldsymbol{A}n_x + \boldsymbol{B}n_y + \boldsymbol{C}n_z$ has real eigenvalues and a complete set of linear-independent eigenvectors. Considering one component $u(\vec{x}, t)$ of the unknown vector $\boldsymbol{U}(\vec{x}, t)$ for simplicity, we can approximate it by $u_h(\vec{x}, t)$

$$u(\vec{x}, t) \approx u_h(\vec{x}, t) \equiv \sum_{l=0}^{N-1} \hat{u}_l(t)\phi_l(\vec{x}), \tag{2.3}$$

with $\hat{u}_l(t)$ the time-dependent scalar degrees of freedom, $\phi_l(\vec{x})$ the space-dependent polynomial basis functions and N the number of degrees of freedom. We use orthogonal basis functions from Jacobi polynomials defined in a reference element $\mathcal{T}_{\mathrm{E}}$, which are mapped into each element $\mathcal{T}_m$ of the discretized computational domain Ω. Approximation (2.3) introduces high-order polynomial representation in space of the unknown vector. The number of degrees of freedom and therefore the number of basis functions is defined by the order of the numerical approximation $\mathcal{O}$. For two-dimensional problems we use $N = (\mathcal{O}+1)\mathcal{O}/2$ and for three-dimensional problems we use $N = (\mathcal{O}+2)(\mathcal{O}+1)\mathcal{O}/6$ basis function hierarchically ordered. We write equation (2.1) in divergence form as

$$\partial_t u + \nabla \cdot \mathcal{F}(u) - s(u) = 0, \tag{2.4}$$

with $\nabla = (\partial_x, \partial_y, \partial_z)$ and $\mathcal{F}(u) = (f(u), g(u), h(u))$. Multiplying equation (2.4) by the basis function $\phi_k(\vec{x})$, integrating in the control volume $\mathcal{T}_m$ and using the divergence theorem we get

$$\int_{\mathcal{T}_m} \phi_k \partial_t u \, \mathrm{d}\boldsymbol{x} + \int_{\partial \mathcal{T}_m} \phi_k \mathcal{F}(u) \cdot \vec{n} \, \mathrm{d}\boldsymbol{x} - \int_{\mathcal{T}_m} \nabla \phi_k \cdot \mathcal{F}(u) \, \mathrm{d}\boldsymbol{x} = \int_{\mathcal{T}_m} \phi_k s(u) \, \mathrm{d}\boldsymbol{x}. \tag{2.5}$$

Introducing the polynomial approximation (2.3) and integrating in the time interval $[t^n, t^{n+1}]$ we obtain

$$\begin{aligned} \hat{u}_k^{n+1} = \hat{u}_k^n - \frac{1}{|\mathcal{T}_m| m_k} \Bigg[& \int_{t^n}^{t^{n+1}} \int_{\partial \mathcal{T}_m} \phi_k \mathcal{F}(u) \cdot \vec{n} \, \mathrm{d}\boldsymbol{x} \, \mathrm{d}t \\ & - \int_{t^n}^{t^{n+1}} \int_{\mathcal{T}_m} \nabla \phi_k \cdot \mathcal{F}(u) \, \mathrm{d}\boldsymbol{x} \, \mathrm{d}t - \int_{t^n}^{t^{n+1}} \int_{\mathcal{T}_m} \phi_k s(u) \, \mathrm{d}\boldsymbol{x} \, \mathrm{d}t \Bigg], \end{aligned} \tag{2.6}$$

with $|\mathcal{T}_m|$ the Jacobian matrix of the mapping between the reference element $\mathcal{T}_{\mathrm{E}}$ and the physical element $\mathcal{T}_m$, and m_k the diagonal mass matrix computed in the reference element. We refer to the work of Käser & Dumbser (2006) and Dumbser & Käser (2006) for details on the numerical method when applied to the linear seismic wave equation and its efficient implementation and to the work of Castro (2007) for an application to nonlinear hyperbolic equations.

Equation (2.6) is a one-step scheme for the numerical solution from time $t = t^n$ to $t = t^{n+1}$ providing the approximation to the flux, volume and source space–time integrals. Note that we update each single degrees of freedom of the polynomial approximation (2.3). This is a standard DG finite-element discretization. If we consider the FV approach, we use only first-order approximation in the polynomial representation (2.3), the term $\nabla\phi_k$ cancels out and the scheme (2.6) updates the cell average of the numerical solution. The choice of the scheme, i.e. DG or FV, is problem-dependent. In general, DG numerical methods produce smaller errors and are compact, meaning that each element needs only its direct neighbours. However, they can become expensive in computational time when applied to nonlinear equations. On the other hand, FV schemes need a high-order non-oscillatory reconstruction step that involves several large stencils containing more elements than just the direct neighbours, and therefore the schemes are less compact. Equation (2.6) is a general representation of the numerical scheme and applies to one-, two-, and three-dimensional problems integrating over the corresponding element $\mathcal{T}_m$.

The flux integral along the boundaries of the element $\mathcal{T}_m$ is computed by solving the generalized Riemann problem using the arbitrary high-order derivative (ADER) approach, originally presented by Toro *et al.* (2001) and Titarev & Toro (2002). We refer to the work of Castro & Toro (2008) for a detailed description of the solution and comparison of three different strategies. A fourth possible solution was presented by Dumbser *et al.* (2008).

The generalized Riemann problem introduces time accuracy of the same order as the spatial accuracy and is expressed as the following initial value problem:

$$\left.\begin{array}{ll}\text{PDEs:} & \partial_t \boldsymbol{U} + \partial_\xi \boldsymbol{F}(\boldsymbol{U}) = \boldsymbol{S}(\boldsymbol{U}), \quad \xi \in (-\infty, \infty), \quad \tau > 0, \\ \text{IC:} & \boldsymbol{U}(\xi, 0) = \begin{cases} \boldsymbol{U}_{\mathrm{L}}(\xi) & \text{if } \xi < 0, \\ \boldsymbol{U}_{\mathrm{R}}(\xi) & \text{if } \xi > 0, \end{cases}\end{array}\right\} \tag{2.7}$$

where ξ is a local coordinate aligned with the normal vector $\vec{n}_j$ to the boundary face $\partial\mathcal{T}_{mj}$, with the origin $\xi = 0$ right at the interface. The index j refers to the boundaries of the control volume with $\partial\mathcal{T}_m = \sum_j \partial\mathcal{T}_{mj}$. The Riemann problem (2.7) is generalized in two ways, the initial conditions are high-order polynomials in space and the equations include source terms. The solution is a time-dependent solution that allows us to evaluate the numerical flux $\mathcal{F}(u) \cdot \vec{n}$ and therefore the calculation of the flux integral in equation (2.6).

The initial conditions $\boldsymbol{U}_{\mathrm{L}}(\xi)$ and $\boldsymbol{U}_{\mathrm{R}}(\xi)$ are obtained from the polynomial representation of the numerical solution at time $t = t^n$ for DG schemes. In the case of FV schemes, we use a weighted essentially non-oscillatory (WENO) reconstruction procedure of order $\mathcal{O}$, for example the one presented by Dumbser & Käser (2007), to recover WENO polynomial representation of the unknown vector $\boldsymbol{U}$ in each element at each time step. In both cases the data have to be rotated in order to align them to the normal direction $\vec{n}_j$.

The volume and source integrals in equation (2.6) are approximated using a space–time high-order evaluation of the integrals, where we use the Cauchy–Kowalewski method to express high-order spatial derivatives into high-order time derivatives and then use a Taylor time-series expansion in the time interval $[t^n, t^{n+1}]$. In practice, we do an L2 projection of each time derivatives $\partial_t^{(r)} u_h(\vec{x}, t^n)$ inside element $\mathcal{T}_m$ to obtain a space–time evolution equation expressed as follows:

$$u(\vec{x}, \tau) = \hat{u}_{lr} \phi_l(\vec{x}) \varphi_r(\tau), \tag{2.8}$$

where $\varphi_r(\tau)$ are the coefficients of the Taylor series with $r=0,\ldots,\mathcal{O}-1$ and $\tau\in[0,\Delta t]$. The time step $\Delta t=t^{n+1}-t^n$ is defined by the stability condition

$$\Delta t\leq\frac{C_{\mathrm{cfl}}}{2(\mathcal{O}-1)+1}\frac{h}{v},\tag{2.9}$$

where C_{cfl} is the Courant–Friedrichs–Lewy number proposed by Courant *et al.* (1928), $\mathcal{O}$ the order of the numerical approximation used in equation (2.3), h a measure of the size of the element and v the fastest wave velocity in the particular element. As mentioned above, FV methods use $\mathcal{O}=1$ in approximation (2.3) and therefore equation (2.9) reduces to the classical FV stability condition. This condition is applied to each element in the computational domain and then the smallest time step is used to update the numerical solution for all elements. We refer to this approach as the global time step (GTS) method, which is still the most common approach for solving balance laws like equation (2.1).

If we consider the same C_{cfl} number for the full computational domain, equation (2.9) shows that the maximal time step Δt in each element depends on the element size h, the order $\mathcal{O}$ of the approximation used in equation (2.3) and the maximum wave velocity v. Some of these parameters can be adjusted to recover a homogeneous time step throughout the computational domain, while others depend on the particular problem that we are trying to solve. In both cases an adaptive local time step (LTS) is a powerful tool to improve computational efficiency on medium- and large-scale numerical simulations, in particular for geophysical problems. In the next section we present an LTS scheme that can be applied to ADER-DG and ADER-FV numerical methods.

3. Space–time adaptive approach

When solving realistic large-scale geophysical problems we normally encounter complex geometries that define the computational domain. On the other hand, the maximum and minimum wave speeds involved in the numerical simulation can present highly heterogeneous values and therefore forcing us to adapt the mesh spacing to provide enough spatial resolution. Take for example a seismic wave propagation problem, where the velocity model consists of a large-scale homogeneous material but with a confined small basin of very low velocity. Because of the small-scale geometry of the basin and the slow velocity of the waves inside, we are forced to prescribe a fine mesh in order to resolve the wave structure inside this basin. Another example is the wave propagation of a tsunami wave where long period waves propagating in the open ocean approach and interact with a complex small-scale coast line. In these two examples, an LTS scheme would improve the computational effort to simulate the problem.

In this section we generalize the approach presented by Dumbser *et al.* (2007) for the ADER-DG LTS in order to be used also in the context of ADER-FV schemes. The main difficulty to directly apply the algorithm of ADER-DG LTS to high-order FV methods is the reconstruction process. In this process we make use of cell averages of neighbour elements to construct a polynomial representation of the unknown vector inside each element at each time step, typically performing a WENO type of reconstruction. The number of cell averages needed is defined by the number of elements in the reconstruction stencil. The cell average values in

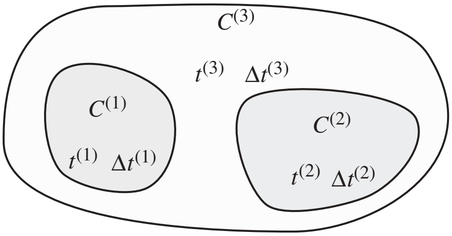
$C^{(3)}$
$t^{(3)}$ $\Delta t^{(3)}$
$C^{(1)}$
$t^{(1)}$ $\Delta t^{(1)}$
$C^{(2)}$
$t^{(2)}$ $\Delta t^{(2)}$

Figure 1. Cluster organization of the computational domain $\Omega = C^{(1)} \cup C^{(2)} \cup C^{(3)}$. In this figure, the full computational domain has been divided into three clusters, each one running with different time steps.

the corresponding stencil need to be at the same time level as the element that is reconstructed. This is typically not possible when each element runs its own time step. In the work of Fumeaux *et al.* (2004) in the context of FV, they enforce some level of synchronization by allowing the elements to advance by a multiple of 2 of a fundamental time step. A more general approach was presented by Tessmer (2000) for the finite difference scheme, allowing for any positive multiple of a fundamental time step. In the work of Lörcher *et al.* (2007) and Gassner *et al.* (2008) they extend the ADER-DG LTS to nonlinear Euler and Navier–Stokes equations considering DG methods.

The LTS approach presented here introduces two modifications to the one presented by Dumbser *et al.* (2007). First, we use LTS applied to clusters of elements instead to a single element; and second, we propose a reconstruction step based on evolved cell averages when the neighbour elements in the reconstruction stencil belong to different clusters and therefore to different time levels. Thereby, the reconstruction stencils have to be fully contained in direct neighbour clusters. In figure 1 we show the computational domain subdivided into three clusters. Each cluster defines the maximum time step by taking the smallest step from the stability criterion (2.9) applied to the elements inside the cluster.

The main ingredient of the LTS approach is the update criterion which when applied to a cluster of elements reads: a cluster of elements, identified with superscript c, is updated from time level $t^{(c)}$ to the time level $t^{(c)} + \Delta t^{(c)}$ if and only if the update criterion

$$t^{(c)} + \Delta t^{(c)} \leq \min(t^{(c_j)} + \Delta t^{(c_j)}) \quad \forall c_j \tag{3.1}$$

is satisfied with respect to all direct cluster neighbours c_j. Once the criterion has been fulfilled for the cluster c, the elements in the cluster with neighbours inside the same cluster can compute the flux in the full interval $\Delta t^{(c)}$, while the ones along the boundary of the cluster compute the flux integrals in the following time interval:

$$[t1; t2] = [\max(t^{(c)}, t^{(c_j)}); \min(t^{(c)} + \Delta t^{(c)}, t^{(c_j)} + \Delta t^{(c_j)})]. \tag{3.2}$$

Note that $[t1; t2]$ depends on the corresponding cluster neighbour c_j. Equations (3.1) and (3.2) are identical to the one presented by Dumbser *et al.* (2007), but now applied to clusters of elements.

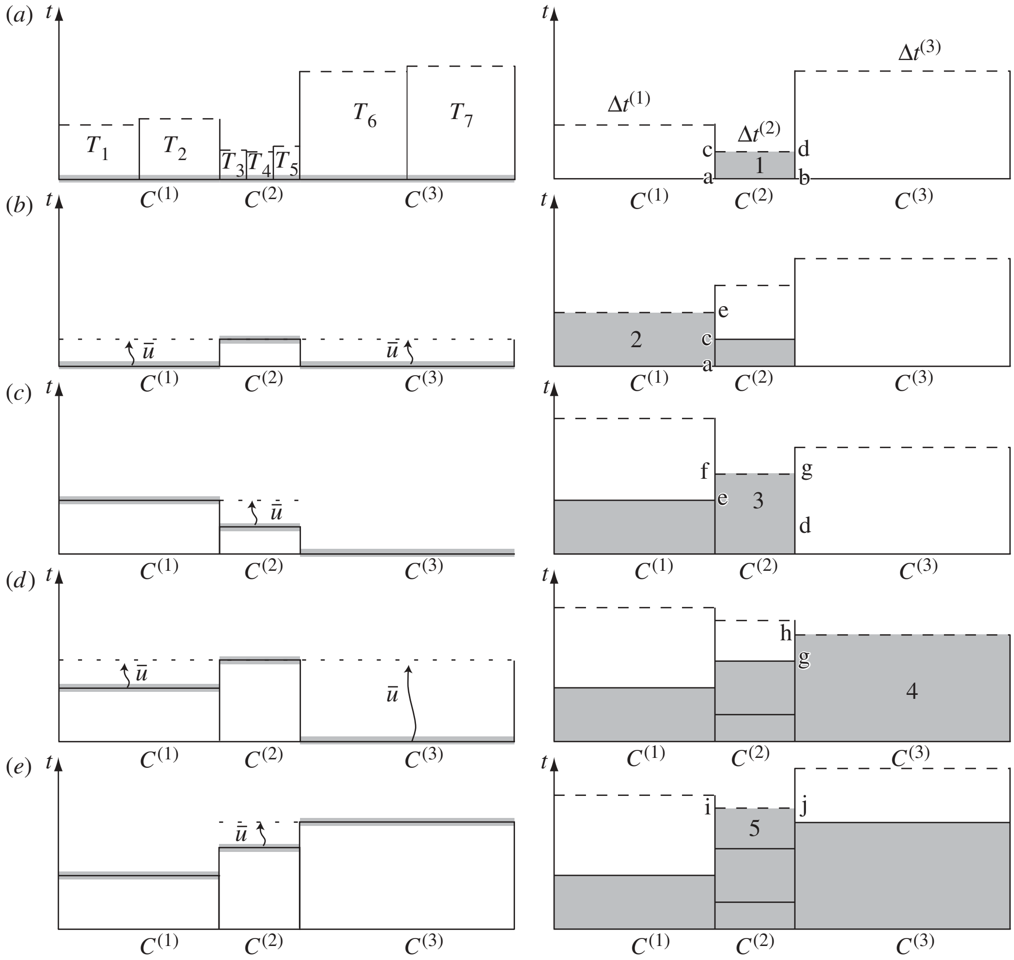

Figure 2. Reconstruction and update for local time step schemes. On the left the reconstruction step. On the right the update step. Horizontal solid black lines are actual time level of clusters. Horizontal grey lines are reconstructed values. Horizontal dashed lines are maximum time step for the cluster. Horizontal dotted lines (reconstruction column) are extrapolated cell average time with $\bar{u}$ the cell averages.

The novelty of this approach is the reconstruction step which we explain next, via a one-dimensional example, for simplicity.

In figure 2 we show two columns. The left one shows the reconstruction step while the right one shows the update step. At the beginning of the simulation, figure 2*a*, we cluster the elements with similar time step: $C^{(1)}=[\mathcal{T}_1,\mathcal{T}_2]$, $C^{(2)}=[\mathcal{T}_3,\mathcal{T}_4,\mathcal{T}_5]$ and $C^{(3)}=[\mathcal{T}_6,\mathcal{T}_7]$. Because all the elements are at the same time level $t=0$ we can reconstruct the initial data. After each cluster defines its maximum time step, see dashed line in the update column, we use the update criterion (3.1). In figure 2*a*, cluster $C^{(2)}$ fulfills the criterion and is updated to the time level $t=\Delta t^{(2)}$. In updating $C^{(2)}$ we compute fluxes in the time interval $\bar{ac}$ with respect to $C^{(1)}$ and $\bar{bd}$ with respect to $C^{(3)}$. These fluxes are stored in the corresponding neighbour elements on the neighbouring clusters. In this case the flux associated to the interface $\bar{ac}$ is stored by element $\mathcal{T}_2$ in $C^{(1)}$ and the flux associated to the interface $\bar{bd}$ is stored by element $\mathcal{T}_6$ in $C^{(3)}$. At the next time cycle represented in figure 2*b*, we need to reconstruct the elements in $C^{(2)}$ at the new time level. The problem is that the neighbouring clusters remain at time $t=0$. Here, we use an evolved cell average value when the elements in the reconstruction stencil belong to a different time level. In this reconstruction step, elements of $C^{(2)}$ will use evolved cell averages from $C^{(1)}$ and $C^{(3)}$ when needed as represented with $\bar{u}$ in figure 2*b*. After $C^{(2)}$ is reconstructed and the new maximum time step is computed, the update criterion is used. In figure 2*b* $C^{(1)}$ is updated to the time level $t=\Delta t^{(1)}$. All elements on the boundary between $C^{(1)}$ and $C^{(2)}$ will compute flux only in the time interval $\bar{ce}$ as defined by equation (3.2). The flux in the segment $\bar{ac}$ was already computed and stored in the previous cycle. In figure 2*c* we reconstruct $C^{(1)}$ using information from $C^{(2)}$ and compute the maximum time step. The update criterion tells us to update $C^{(2)}$ to the time level $\bar{fg}$. Boundary elements between $C^{(1)}$ and $C^{(2)}$ compute the flux in the interval $\bar{ef}$ and then is stored by element $\mathcal{T}_2$ in $C^{(1)}$. Elements in the boundary between $C^{(2)}$ and $C^{(3)}$ compute the flux for the full update time interval $\bar{dg}$, which is stored by element $\mathcal{T}_6$ in $C^{(3)}$. In figure 2*d* we reconstruct $C^{(2)}$ using evolved cell averages from $C^{(1)}$ and $C^{(3)}$ and then $C^{(3)}$ is updated. Elements at the interface between $C^{(2)}$ and $C^{(3)}$ need to compute the flux integral only in the time segment $\bar{gh}$. In figure 2*e* we reconstruct $C^{(3)}$ using evolved cell averages from $C^{(2)}$ and then update $C^{(2)}$. This procedure continues until a synchronization time level is imposed or the final time simulation is reached.

Because we consider reconstruction stencils to be fully contained in direct neighbour clusters we can ensure that the evolved cell averages are within the valid time interval defined by the time step of each cluster. The cell average evolution within the time interval $t=[t^{(c)}, t^{(c)}+\Delta t^{(c)}]$ is obtained from equation (2.8) considering only the first basis function $\phi_0(\vec{x})=1$ as follows:

$$\bar{u}(\vec{x},\tau)=\hat{u}_{0r}\varphi_r(\tau), \tag{3.3}$$

and then evaluated at the desired time level.

With this new method we can use local time steps associated with each cluster of elements to reduce computational time in large-scale simulations. Moreover, the optimized time step with respect to the stability criterion reduces the numerical diffusion of the scheme. The proposed scheme reproduces the one presented by Dumbser *et al.* (2007) for ADER-DG, and therefore without reconstruction step, if we consider each single element in the computational domain as a cluster.

With the LTS modification the update scheme reads as follows:

$$\begin{aligned}\hat{u}_k^{t_c^{n+1}} = \hat{u}_k^{t_c^n} - \frac{1}{|\mathcal{T}_m| m_k}\Bigg[&\int_{t1}^{t2}\int_{\partial\mathcal{T}_m}\phi_k\mathcal{F}(u)\cdot\vec{n}\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t+\mathcal{H}^{(c)} \\ &-\int_{t_c^n}^{t_c^{n+1}}\int_{\mathcal{T}_m}\nabla\phi_k\cdot\mathcal{F}(u)\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t-\int_{t_c^n}^{t_c^{n+1}}\int_{\mathcal{T}_m}\phi_k s(u)\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}t\Bigg],\end{aligned} \tag{3.4}$$

with $t_c^n = t^{(c)}$ and $t_c^{n+1} = t^{(c)} + \Delta t^{(c)}$. If the element that is updated has direct neighbouring elements inside the same cluster $\mathcal{H}^{(c)} = 0$ and $[t1; t2] = [t_c^n; t_c^{n+1}]$. On the other hand, if the element to update has a neighbouring element in a different cluster $\mathcal{H}^{(c)} \neq 0$ and contains the precomputed fluxes from previous neighbouring cluster updates. In the same manner $[t1; t2]$ are computed from equation (3.2). After the element is updated, $\mathcal{H}^{(c)}$ is set to zero.

The fact that we restrict this approach to consider reconstruction stencils fully contained in direct neighbouring clusters does not present practical restrictions. The LTS approach only pays off if the computational domain has a small number of short time-step elements, which then one can cluster together. Moreover, if we only consider two *clusters* of elements, which are not necessarily formed by neighbouring elements, this approach still applies.

In the next section we validate the proposed ADER-FV LTS scheme through a convergence test for the two-dimensional nonlinear shallow water equations to illustrate that the new scheme achieves the expected order of convergence. In Dumbser *et al.* (2007), the accuracy of the ADER-DG LTS was already presented for linear seismic wave equations in three dimensions.

4. Validation

In order to verify the expected order of accuracy we perform a convergence test where the numerical solution is compared with an exact solution on a sequence of refined meshes for the nonlinear shallow water equations with non-horizontal bathymetry; see the book of Toro (2001) for further details of the equations. Measuring the error resulting from different meshes we can compute the empirical order of convergence of the ADER-FV LTS numerical scheme. Here, we consider an exact solution $\mathcal{U}(\vec{x}, t)$ given by

$$\mathcal{U}(\vec{x}, t) = \begin{bmatrix} h(\vec{x}, t) \\ h(\vec{x}, t)u(\vec{x}, t) \\ h(\vec{x}, t)v(\vec{x}, t) \end{bmatrix}, \quad \text{with} \quad \left. \begin{aligned} b(\vec{x}) &= 0.2\mathrm{e}^{-8(x^2+y^2)}, \\ h(\vec{x}, t) &= \mathrm{e}^{0.1t} - b(\vec{x}), \\ u(\vec{x}, t) &= 0.2 + 0.1\sin(x\pi), \\ v(\vec{x}, t) &= 0.2 + 0.1\sin(y\pi). \end{aligned} \right\} \tag{4.1}$$

Introducing this exact solution into the two-dimensional version of the balance law (2.1) we obtain the new source term, $\mathcal{S}(\mathcal{U})(\vec{x}, t) = \partial_t \mathcal{U} + \partial_x \boldsymbol{F}(\mathcal{U}) + \partial_y \boldsymbol{G}(\mathcal{U}) - \boldsymbol{S}(\mathcal{U})$. The convergence test problem is then defined as follows:

$$\left. \begin{aligned} &\text{PDEs:} \quad \partial_t \boldsymbol{U} + \partial_x \boldsymbol{F}(\boldsymbol{U}) + \partial_y \boldsymbol{G}(\boldsymbol{U}) = \boldsymbol{S}(\boldsymbol{U}) + \mathcal{S}(\boldsymbol{U}), \\ &\text{IC:} \quad\;\;\;\; \boldsymbol{U}(\vec{x}, 0) = \mathcal{U}(\vec{x}, 0). \end{aligned} \right\} \tag{4.2}$$

The computational domain $[-1, 1] \times [-1, 1]$ is discretized with structured triangular meshes and two clusters are defined with different characteristic element sizes determining the local time steps (figure 3). Periodic boundary conditions are applied. The final time in the computation is set to $t = 1$. In table 1 we observe that the expected orders are reached for second to fifth order for this test problem using the ADER-FV LTS numerical method. The fine cluster $C^{(2)}$ computes 32, 35 and 39 per cent more time steps than the coarse cluster $C^{(1)}$ for meshes 4, 8 and 16.

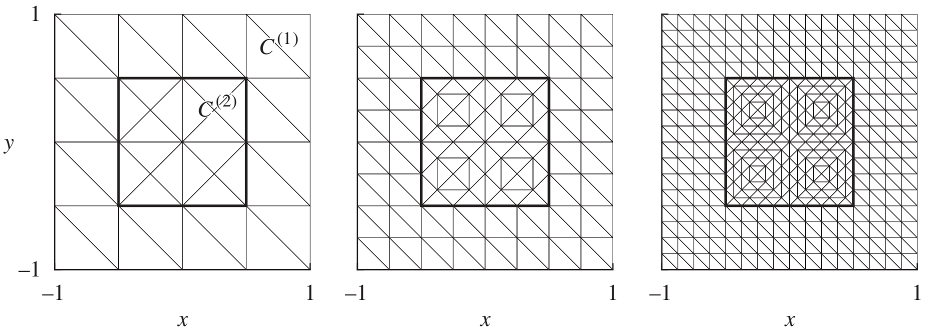

Figure 3. Set of mesh refinements for the convergence test. We refer to them as 4, 8 and 16. The bold line defines the interface between the two clusters with different characteristic time step lengths.

Table 1. Errors for the numerical convergence test for component u_2 from second (P1) to fifth (P4) order. In the last column the ratio of time steps between $C^{(2)}$ and $C^{(1)}$ is shown.

mesh	L^1	$\mathcal{O}_{L^1}$	L^2	$\mathcal{O}_{L^2}$	L^∞	$\mathcal{O}_{L^\infty}$	$C^{(2)}/C^{(1)}$
4	7.95×10^{-2}	—	5.21×10^{-2}	—	1.05×10^{-1}	—	1.32
8	2.13×10^{-2}	1.90	1.44×10^{-2}	1.86	5.57×10^{-2}	0.91	1.35
16	3.91×10^{-3}	2.45	2.70×10^{-3}	2.41	1.53×10^{-2}	1.86	1.39
4	4.04×10^{-2}	—	2.65×10^{-2}	—	6.60×10^{-2}	—	1.32
8	9.07×10^{-3}	2.16	6.20×10^{-3}	2.09	2.41×10^{-2}	1.45	1.35
16	9.03×10^{-4}	3.33	6.26×10^{-4}	3.31	3.74×10^{-3}	2.69	1.39
4	4.07×10^{-2}	—	2.82×10^{-2}	—	9.33×10^{-2}	—	1.32
8	5.77×10^{-3}	2.82	4.10×10^{-3}	2.79	1.98×10^{-2}	2.24	1.35
16	3.38×10^{-4}	4.09	2.61×10^{-4}	3.97	1.98×10^{-3}	3.32	1.39
4	2.84×10^{-2}	—	2.09×10^{-2}	—	8.25×10^{-2}	—	1.32
8	4.57×10^{-3}	2.63	3.77×10^{-3}	2.47	2.99×10^{-2}	1.46	1.35
16	1.52×10^{-4}	4.91	8.95×10^{-5}	5.39	2.63×10^{-4}	6.83	1.39

5. Geophysical applications

(a) *Tsunami wave propagation*

Here we show an application of the ADER-FV LTS method where LTS adaptation helps to reduce computational effort to solve a real-scale problem. The case considers a tsunami wave propagation problem modelled with the two-dimensional nonlinear shallow water equations. We use the well-balanced ADER-FV scheme presented by Castro *et al.* (submitted) with and without the LTS modification. The computational domain represents an area offshore the west coast of Chile close to San Antonio with the coordinates $(x, y) = (0, 0)$ centred at longitude 78°W and latitude 38°S (figure 4*a*). The bathymetry of the area (Amante & Eakins 2008) was used to extract the coast line. In figure 4*a* we

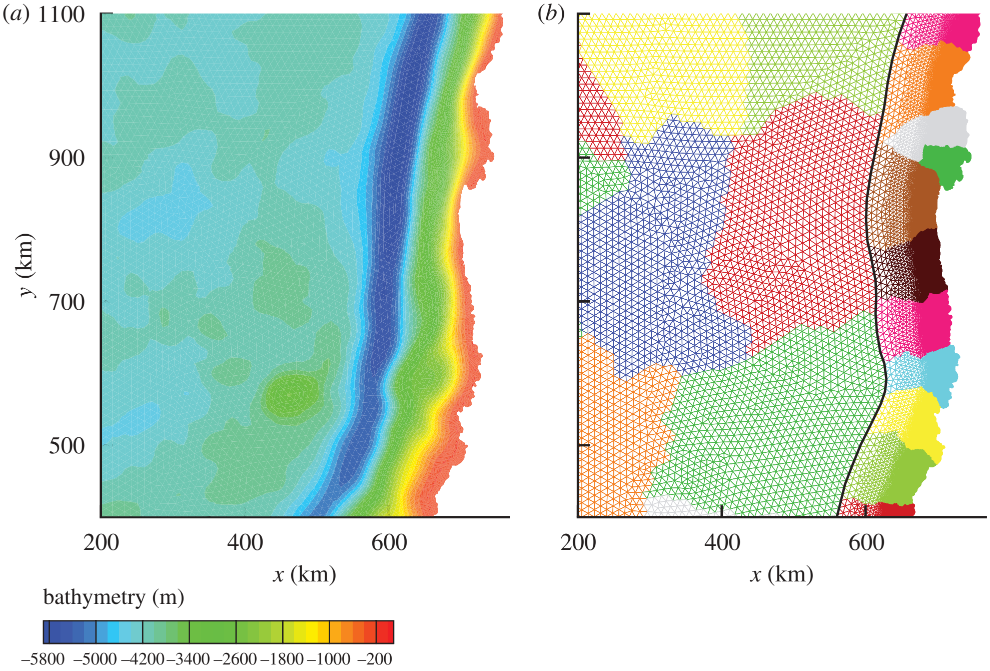

Figure 4. In this figure we show part of the computational domain. The top, left and bottom of the domain are open boundaries while the right boundary represents the coast line represented with reflective boundaries. (*a*) Bathymetry of the area of interest. (*b*) Discretization of the computational domain with statically adapted triangular mesh. The black line identifies the interface between the two clusters $C^{(1)}$ on the right and $C^{(2)}$ on the left.

plot the bathymetry of the area of interest. Observe the strong contrast on the bathymetry depth with variation from -5800 m in the Atacama Trench to 0 m on the coast. In figure 4*b* we plot the mesh used to discretize the physical domain. The mesh was constructed to accurately describe the coast line with very small elements (see right part of figure 4*b*). Because in the open ocean the velocity of the tsunami wave is much higher, the wave length is larger. Furthermore, there are no geometrical restrictions and we can use larger elements to discretize that part of the domain (see left part of figure 4*b*). This unbalance of mesh size produces very small time steps near the coast and large time steps away from the coast. Therefore, we subdivide the mesh into two clusters when using the LTS scheme identified in figure 4*b* with a bold black line parallel to the coast. The right cluster is denoted as $C^{(1)}$ and the left one as $C^{(2)}$. Coloured parts of the mesh identify different message-passing interface (MPI) domains.

For this numerical simulation we use third order of approximation in space and time and subdivide the computational domain into 16 MPI subdomains. We run the code on the TETHYS cluster of the Institute of Geophysics at the Ludwig-Maximilians-Universität München (Oeser *et al.* 2006). In figures 5 and 6 we show the numerical results of this simulation. Figure 5 shows snapshots at $t=0, 5, 10, 15, 20, 25$ min and the location of three stations where we register the free surface displacement. The time histories of the three stations are plotted in

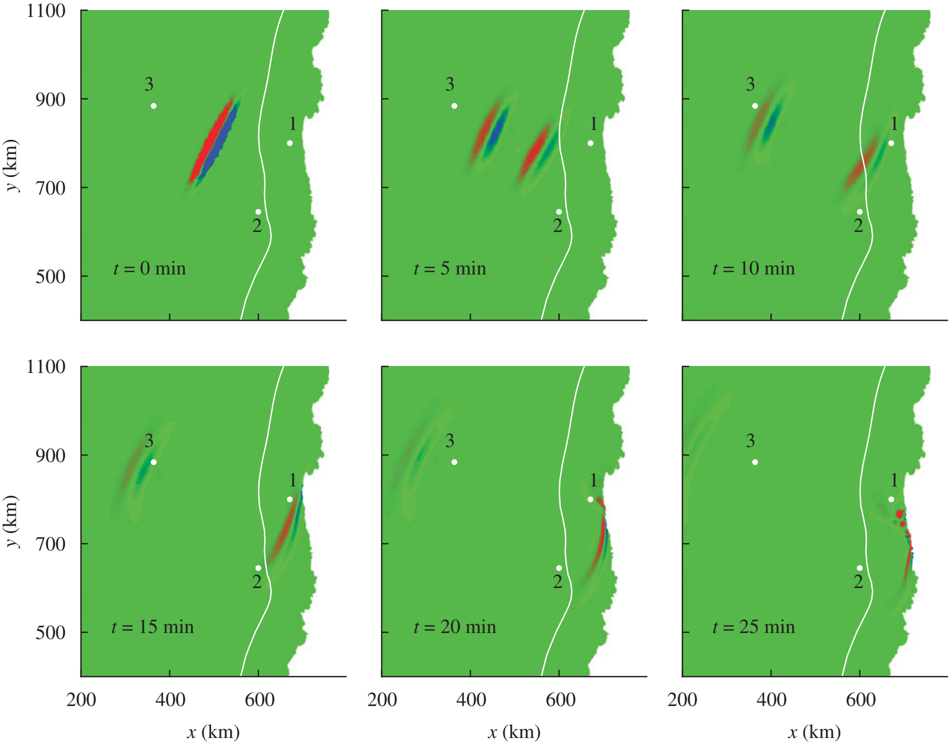

Figure 5. Snapshots of the vertical water displacement for the tsunami wave propagation scenario approaching the coast. The white line represents the interface between the two clusters of elements running at different time steps.

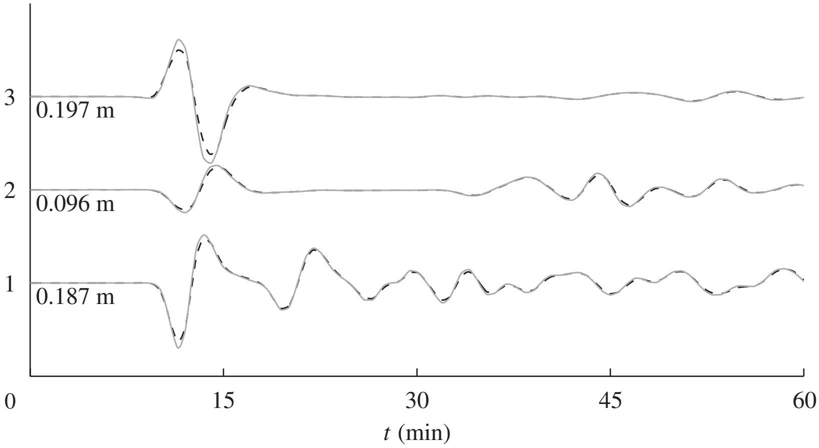

Figure 6. Signals of the tsunami wave heights registered at the three locations indicated in figure 5. On the vertical axis the numbers indicate the receivers from 1 to 3. Immediately inside we place the maximum wave signal for each receiver. Broken line, global time step; continuous line, local time step.

figure 6. We compare the numerical solution using the GTS and the LTS scheme. The GTS scheme runs for 12 154 s on 16 AMD Opteron 250 2.4 GHz cores, while the LTS scheme uses 9810 s, an improvement of 20 per cent. In the LTS the smaller time step cluster $C^{(1)}$ uses 2333 updates, while the cluster $C^{(2)}$ only requires 528 wich is, on average, a difference of 4.4 number of updates. In figure 5 at time $t=0$ we observe the initial condition with a wave amplitude of the order of 1 m. A negative wave travels to the coast crossing the cluster boundary without generating any spurious numerical artefacts. See time $t=5,\ldots,20$ min. At $t=15$ min this wave arrives to the coast and reflects back. On the other hand, a positive wave travels through deep ocean heading northwest. This wave is sampled by the register number 3.

In figure 6 we plot the signals for both the methods showing the same behaviour during the full simulation time, as expected. We observe a slightly higher amplitude in the first wave arrivals obtained by the LTS scheme, particularly visible in signal 3. We explain this as an effect of the reduced numerical diffusion due to the optimal use of the time step. Signal 1 shows a much richer wave structure due to the vicinity of the coast and the corresponding wave reflections from there. Signal 2 is very interesting as it is recorded just to the left of the cluster interface. After the first wave passage there are about 15 min without vertical water displacement. Later, after roughly 30 min of simulation time, we register again a vertical displacement due to reflections from the coast. We consider these results as a clear indication that the LTS scheme works properly.

(*b*) *Seismic wave propagation*

Here we present a large-scale application of the proposed ADER-DG LTS scheme to a ground motion simulation problem in the continental area of San Antonio on the west coast of Chile. This is done by solving the set of linear hyperbolic equations describing seismic wave propagation in three space dimensions as presented by Dumbser & Käser (2006). We want to remark that this application represents a simplified case of the real geophysical situation. However, the test case clearly demonstrates the flexibility and feasibility of the proposed methodology and could also be applied for more realistic scenarios where detailed knowledge of the earthquake source and the geological properties of the subsurface are included. Furthermore, we recall that the accuracy and correctness of the adaptive DG scheme for simple seismic wave propagation problems were already validated in previous the work of Dumbser *et al.* (2007), and therefore we concentrate our attention on the comparison of the efficiency of the method. To this end, we first show a simulation using the classical ADER-DG GTS scheme using a GTS defined by the smallest time step in the entire computational domain following equation (2.9). Then, we show the results of the same simulation using the LTS scheme with the same standard mesh partition. Finally, we improve the efficiency of that scheme by modifying the partitioning strategy to get a better load balance for multi-processor calculations.

The computational domain with the coordinate origin $(x,y,z)=(0,0,0)$ centred at longitude 72°W, latitude 34°S and elevation 0 m.a.s.l. is shown in figure 7*a*, where five different geological units with distinct material properties are shown: (i) a 200 m thick sedimentary basin at the surface embedded into

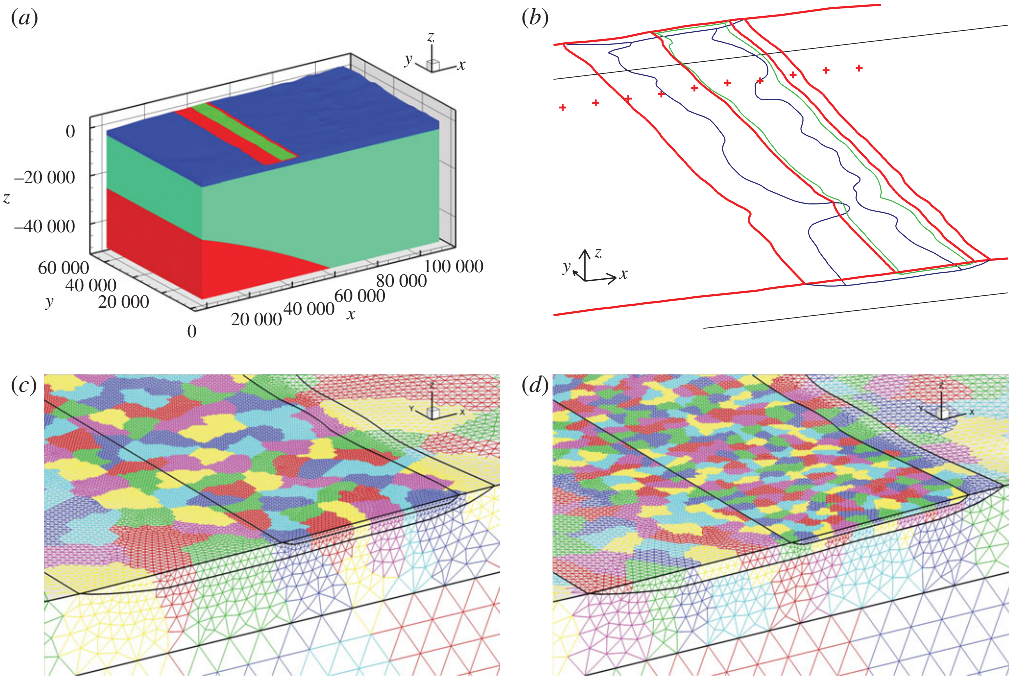

Figure 7. (*a*) Computational model representing the San Antonio area with its topography and the basin and subduction geometry. (*b*) Zoomed view of the geometry of the two thin basins embedded in a near-surface layer. Crosses show the surface locations of the 10 seismic receivers. (*c*) Colour-coded MPI domains obtained by a standard partition of Metis. (*d*) Improved partitioning strategy by subdividing each geological unit (zone) separately in the desired number of MPI domains.

Table 2. Material values for the five units of the San Antonio model for seismic ground motion simulation.

unit	ρ (kg m^{-3})	v_p (m s^{-1})	v_s (m s^{-1})
1	2100	3000	400
2	2100	3600	700
3	2300	4300	2500
4	2600	4500	2600
5	2600	5400	3120

(ii) a wider and 500 m deep basin structure, (iii) a near-surface layer of 2000 m thickness over (iv) a large block of continental crustal material, and (v) an oceanic wedge indicating the subducting slab along a dipping curved interface. The material properties used are given in table 2.

The topography of the model is taken from a digital elevation model (Amante & Eakins 2008) and the source is placed close to the subduction interface at $(x_\mathrm{S}, y_\mathrm{S}, z_\mathrm{S}) = (16\,650, 51\,006, -30\,000)$ m, where the hypocentre of the

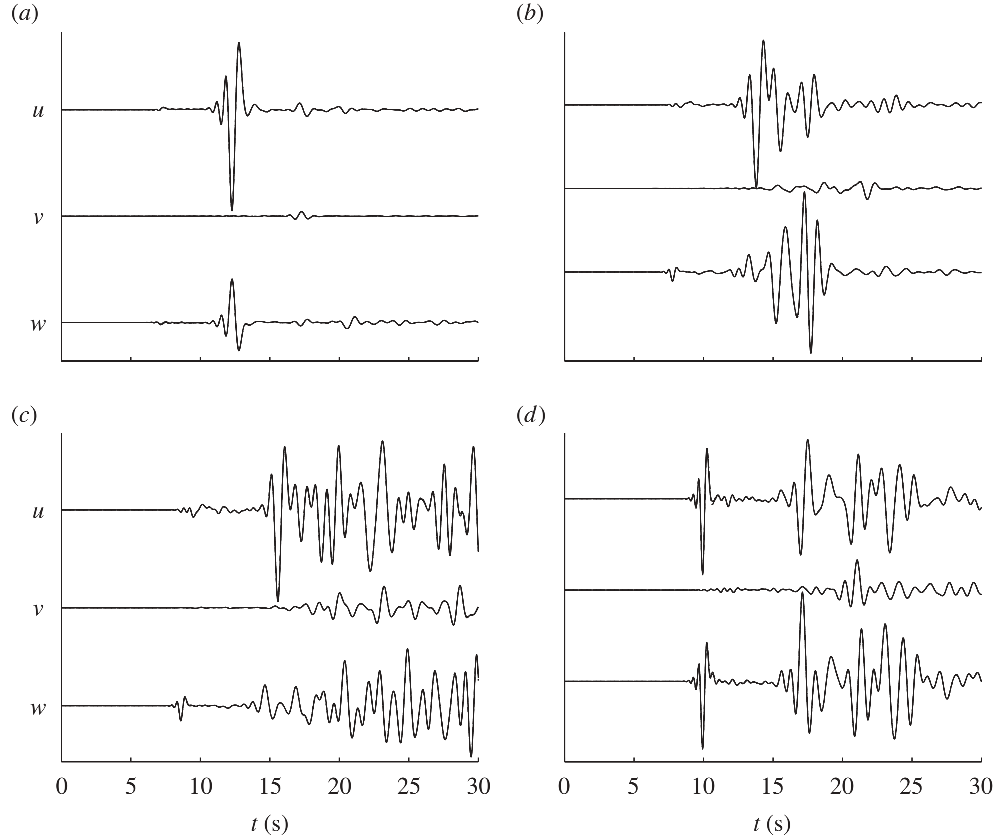

Figure 8. Seismograms of the ground motion velocities u, v and w at four selected receivers. The seismograms of the three different simulations are superimposed and coincide visually in one single line, which confirms that there is no difference in the obtained results. The seismogram scaling is the same within each plot, but might differ from one plot to the other to improve visualization. (*a*) Receiver 2 is positioned west of the basins; (*b*) receiver 4 is inside the outer basin; (*c*) receiver 6 is inside the inner basin; and (*d*) receiver 9 is positioned east of the basins.

1985 Santiago earthquake has been located. Synthetic seismograms are registered at 10 receivers with a regular distance of 3 km in increasing x-direction across the basins at the model surface as shown in figure 7*b*. The seismograms are used to check the consistency of the results obtained by the three different simulations.

The first simulation using an ADER-DG GTS scheme of order 4 and a standard Metis (Karypis & Kumar 1998) partition into 1020 subdomains of equal numbers of elements requires 43 256 s on 1020 Intel Itanium2 Montecito Dual Core 1.6 GHz cores. Leaving the simulation setup unchanged, but exchanging the GTS with an LTS scheme, the computational time significantly reduces to 29 444 s. However, the used standard mesh partition as shown in figure 7*c* is not adequate any more. Whereas for GTS schemes it is important that each subdomain consists of the same number of elements in order to ensure well-adjusted load balance, for

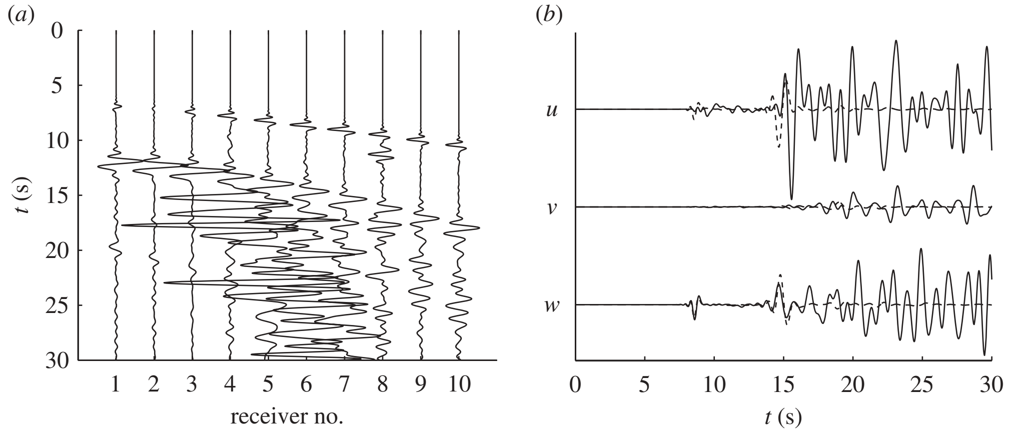

Figure 9. (a) Equally scaled seismograms of the vertical ground motion velocity plotted as a cross section from west to east (left to right) show a clear amplification and prolongation of the signals within the basin structure. In particular, it is obvious that the strong ground motion is caused by the arrival of S-waves. (b) Comparison of the velocity seismograms at receiver 6 inside the inner basin considering or neglecting the basin. It is obvious how the basin causes strong wave amplification and prolongation of the shaking due to wave trapping. Solid line, with basin; dashed line, no basin.

LTS this strategy leads to problems. In fact, each element evolves its solution by an LTS and therefore the number of update cycles for each element to reach a certain time level is different. Furthermore, after each update cycle neighbouring information has to be exchanged via MPI communication between the different processors. Therefore, for LTS schemes it is important that the number of element updates determined by equation (3.1) in each subdomain is equal. This is a highly non-trivial task, as the number of element updates per cycle in each subdomain also changes with time. Therefore, a dynamic load-balancing technique should provide the best load balance. However, an efficient implementation of a dynamic load-balance algorithm for unstructured meshes using LTS is subject to future work and might result in significant MPI overhead. Nevertheless, our preliminary and effective solution is the partition of the mesh respecting the different geological units (zones). This way, each zone is partitioned into the desired number of subdomains as shown in figure 7d. This is similar to the case of the tsunami wave simulation in §5a where each cluster is partitioned separately. Finally, each processor obtains one subdomain from each geological zone that provides an improved load balance due to the fact that computationally expensive (small) and cheap (large) elements are distributed more evenly. Using the LTS scheme with the modified partition of figure 7d in a third simulation reduces the computational time to 26 976 s.

The resulting ground motion components u, v and w in x-, y- and z-direction, respectively, are shown as seismograms in figure 8 at four selected receivers obtained by the three different simulations. In fact, the seismograms appear as one line indicating that all the three simulations visually produce the same results. Receiver 2 located west outside the basin shows clear P- and S-wave arrivals, whereas receiver 4 inside the outer basin and receiver 6 inside the inner basin show a much richer seismogram structure due to trapped waves inside the basins. Trapped waves are a well-known basin effect, if the impedance contrast between a hard basement rock and a soft sedimentary basin rock allows only a small fraction of the wave energy inside the basin to escape from it. Therefore, subsequent internal reflections from the basin boundaries cause wave interference that can significantly amplify and prolongate the ground motion. Receiver 9 on the east side of the basin again shows the direct wave arrivals more clearly; however, followed by a surface wave coda mainly caused by the basin.

Plotting the seismograms of all stations as a cross section from west to east in figure 9*a*, the ground motion amplification and longer duration due to the basins are clearly visible. Furthermore, it is obvious that the S-wave arrival is causing the large amplitudes, while the direct P-waves do not show a significant coda. To emphasize the importance of considering the low-velocity basins even if extremely thin in their spatial extent, we also show a comparison with the results obtained by neglecting the basins. To this end, we leave the geometry of the problem unchanged but fill the basins (units 1 and 2) with the same material as the surrounding surface material of unit 3. Figure 9*b* displays the comparison with and without the basin at receiver 6. Without the basin there is only weak ground motion after about 16 s, i.e. after the direct S-wave has passed the receiver. Therefore, considering such fine, low-velocity structures is currently a key issue in realistic ground motion modelling and seismic hazard analysis, but remains a challenge for many non-adaptive numerical schemes.

6. Conclusions

We presented a space–time adaptive algorithm to be used in the context of fully discrete, one step high-order FV and DG finite-element methods on unstructured meshes. We empirically demonstrated the expected order of convergence up to fifth order. In addition to statically adapted meshes we apply the LTS approach to two real-scale geophysical applications in the FV and DG finite-element framework. In both cases the computational effort is reduced while keeping or even slightly improving the numerical accuracy. For large-scale problems where multi-processor computations are necessary, the LTS approach requires a new mesh partition strategy as the computational work may not be distributed equally. Therefore, we apply a partition technique subdividing clusters of elements or geological zones with equivalent elements separately into the desired number of MPI domains. However, further studies especially with respect to dynamic load-balancing algorithms need to be done for each method and different physical problems. Scalability is becoming an increasingly important issue when enlarging the number of processors to several thousands, which seems to be the future trend in high-performance computing.

The first two authors thank the DFG (Deutsche Forschungsgemeinschaft), as the work was supported through the Emmy Noether-Program (KA 2281/2-1). Furthermore, we thank the Leibniz-Rechenzentrum in München for use of their computational resources within the DEISA Consortium (www.deisa.eu), co-funded through EU FP6 projects RI-508830 and RI-031513, and for their support within the DEISA Extreme Computing Initiative.

References

Amante, C. & Eakins, B. W. 2008 Etopo1 1 arc-minute global relief model: procedures, data sources and analysis. National Geophysical Data Center, NESDIS, NOAA, US Department of Commerce, Boulder, CO.

Babuska, I., Henshaw, W., Oliger, J., Flaherty, J., Hopcroft, J. & Tezduyar, T. (eds) 1995 *Modeling, mesh generation, and adaptive numerical methods for partial differential equations.* IMA Volumes in Mathematics and its Applications, vol. 75, pp. 417–430. Berlin, Germany: Springer.

Castro, C. E. 2007 *High order ADER FV/DG numerical methods for hyperbolic equations.* Monographs of the School of Doctoral Studies in Environmental Engineering. Trento, Italy: University of Trento.

Castro, C. E. & Toro, E. F. 2008 Solvers for the high-order Riemann problem for hyperbolic balance laws. *J. Comput. Phys.* **227**, 2481–2513. (doi:10.1016/j.jcp.2007.11.013)

Castro, C. E., Toro, E. F. & Käser, M. Submitted. ADER schemes on unstructured meshes for shallow water: simulation of tsunami waves.

Courant, R., Friedrichs, K. O. & Lewy, H. 1928 Über die partiellen Differenzialgleichungen der mathematischen Physik. *Mathemat. Annal.* **100**, 32–74. (doi:10.1007/BF01448839)

Dumbser, M. & Käser, M. 2006 An arbitrary high order discontinuous Galerkin method for elastic waves on unstructured meshes: II. The three-dimensional isotropic case. *Geophys. J. Int.* **167**, 319–336. (doi:10.1111/j.1365-246X.2006.03120.x)

Dumbser, M. & Käser, M. 2007 Arbitrary high order non-oscillatory finite volume schemes on unstructured meshes for linear hyperbolic systems. *J. Comput. Phys.* **221**, 693–723. (doi:10.1016/j.jcp.2006.06.043)

Dumbser, M., Käser, M. & Toro, E. F. 2007 An arbitrary high order discontinuous Galerkin method for elastic waves on unstructured meshes: V. Local time stepping and p-adaptivity. *Geophys. J. Int.* **171**, 695–717. (doi:10.1111/j.1365-246X.2007.03427.x)

Dumbser, M., Enaux, C. & Toro, E. F. 2008 Finite volume schemes of very high order of accuracy for stiff hyperbolic balance laws. *J. Comput. Phys.* **227**, 3971–4001. (doi:10.1016/j.jcp.2007.12.005)

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. (doi:10.1006/jpdc.1997.1412)

Fumeaux, C., Baumann, D., Leuchtmann, P. & Vahldieck, R. 2004 A generalized local time-step scheme for efficient FVTD simulations in strongly inhomogeneous meshes. *IEEE Trans. Microwave Theory Tech.* **52**, 1067–1076. (doi:10.1109/TMTT.2004.823595)

Gassner, G., Lörcher, F. & Munz, C. D. 2008 A discontinuous Galerkin scheme based on a space-time expansion II. Viscous flow equations in multi dimensions. *J. Sci. Comput.* **34**, 260–286. (doi:10.1007/s10915-007-9169-1)

Karypis, G. & Kumar, V. 1998 Multilevel k-way partitioning scheme for irregular graphs. *J. Parallel Distrib. Comput.* **48**, 96–129. (doi:10.1006/jpdc.1997.1404)

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. (doi:10.1111/j.1365-246X.2006.03051.x)

Käser, M. & Iske, A. 2005 ADER schemes on adaptive triangular meshes for scalar conservation laws. *J. Comput. Phys.* **205**, 486–508. (doi:10.1016/j.jcp.2004.11.015)

Lörcher, 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.* **32**, 175–199. (doi:10.1007/s10915-007-9128-x)

Oeser, J., Bunge, H.-P. & Mohr, M. 2006 Cluster design in the Earth sciences: TETHYS. In *High performance computing and communications* (eds M. Gerndt & D. Kranzlmüller). Springer Lecture Notes in Computer Science, vol. 4208, pp. 31–40. Berlin, Germany: Springer.

Plewa, T., Linde, T. & Weirs, V. G. (eds) 2000 *Adaptive mesh refinement: theory and applications.* Springer Lecture notes in Computational Science and Engineering, vol. 41. Berlin, Germany: Springer.

Shi, Z.-C., Chen, Z., Tang, T. & Yu, E. (eds) 2006 *Recent advances in adaptive computation.* Contemporary Mathematics, vol. 383. Oxford, UK: Oxford University Press.

Tessmer, E. 2000 Seismic finite-difference modeling with spatially varying time steps. *Geophysics* **65**, 1290–1293. (doi:10.1190/1.1444820)

Titarev, V. A. & Toro, E. F. 2002 ADER: arbitrary high order Godunov approach. *J. Sci. Comput.* **17**, 609–618. (doi:10.1023/A:1015126814947)

Toro, E. F. 2001 *Shock-capturing methods for free-surface shallow flows.* Chichester, UK: John Wiley & Sons Ltd.

Toro, E. F., Millington, R. C. & Nejad, L. A. M. 2001 Towards very high order Godunov schemes. In *Godunov methods: theory and applications* (ed. E. F. Toro), pp. 907–940. New York, NY: Kluwer Academic/Plenum Publishers.