Geophysical Prospecting

EAGE EUROPEAN ASSOCIATION OF GEOSCIENTISTS & ENGINEERS

Geophysical Prospecting, 2014, **62**, 1111–1125 doi: 10.1111/1365-2478.12138

A comparison of continuous mass-lumped finite elements with finite differences for 3-D wave propagation

Elena Zhebel[1]*, Sara Minisini[1], Alexey Kononov[2] and Wim A. Mulder[1,3]

[1]*Shell Global Solutions International, Kessler Park 1, 2288 GS Rijswijk, The Netherlands,* [2]*Source Contracting, Triosingel 37, 4101 XC Culemborg, The Netherlands, and* [3]*Department of Geotechnology, Faculty of Civil Engineering and Geosciences, Delft University of Technology, PO Box 5048, 2600 GA Delft, The Netherlands*

Received February 2013, revision accepted October 2013

ABSTRACT

The finite-difference method on rectangular meshes is widely used for time-domain modelling of the wave equation. It is relatively easy to implement high-order spatial discretization schemes and parallelization. Also, the method is computationally efficient. However, the use of finite elements on tetrahedral unstructured meshes is more accurate in complex geometries near sharp interfaces. We compared the standard eighth-order finite-difference method to fourth-order continuous mass-lumped finite elements in terms of accuracy and computational cost. The results show that, for simple models like a cube with constant density and velocity, the finite-difference method outperforms the finite-element method by at least an order of magnitude. Outside the application area of rectangular meshes, i.e., for a model with interior complexity and topography well described by tetrahedra, however, finite-element methods are about two orders of magnitude faster than finite-difference methods, for a given accuracy.

Key words: 3D, continuous finite elements, finite differences, seismic modelling.

INTRODUCTION

The finite-difference method has become the workhorse for time-domain modelling of the wave equation, with applications in acquisition optimization, development and testing of seismic processing algorithms, reverse-time migration and full waveform inversion. Its advantages are the relative ease of coding and parallelization, the use of high-order spatial discretization schemes and explicit time stepping, and its computational speed. Common opinion and experience is that finite-difference methods perform well in rectangular domains with smooth velocity variations. However, in the case of an irregular free surface or sharp contrast in the properties of the medium, finite-difference methods lose accuracy when using a Cartesian coordinate system. If the interface does not follow the grid, the staircasing effect generates local first-order errors. Because the solution is continuous but not differentiable across an impedance contrast, a local second-order error will be incurred as well. The standard finite-difference method can be modified to avoid the staircase approximation of the free surface in topography (Hestholm and Ruud 1994; Lombard *et al.* 2008; Tarrass *et al.* 2011) and to avoid loss of accuracy across sharply dipped subsurface interfaces (Preston *et al.* 2008; Sofronov *et al.* 2012). However, more work on high-order finite-difference schemes for boundaries in these approaches is still required.

Finite-element methods have some advantages over finite-difference methods because they can easily handle geometric or property discontinuities by using unstructured meshes and spatial local refinement. Finite elements that follow sharp interfaces do not suffer from loss of accuracy. Moreover, they offer flexibility in mixing discretization orders and element geometries and deploying hybrid discretizations. The choice of a suitable time discretization scheme enables explicit time stepping. If the standard finite-element approach is used, a large sparse linear system of equations has to be solved at each time step, and this has a negative impact on performance.

*E-mail: Elena.Zhebel@shell.com

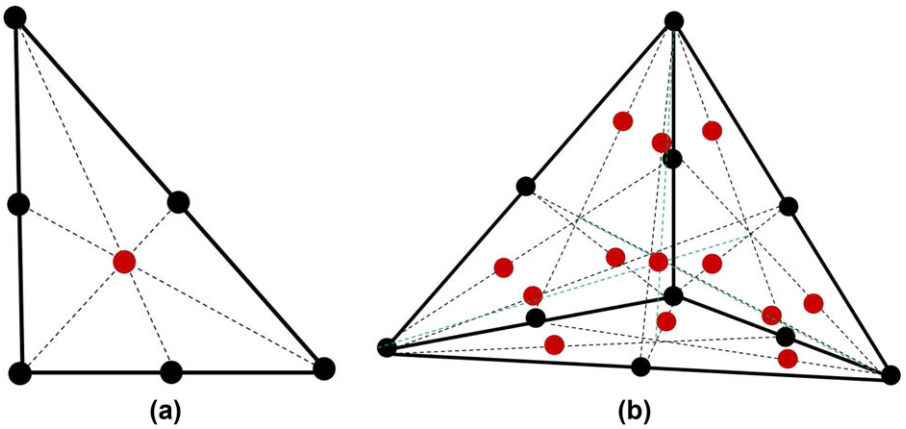


Figure 1 Continuous mass-lumped finite element of degree 2 in (a) 2D and (b) 3D. Black nodes mark the discretization nodes of the standard finite element of second degree and red nodes mark the augmented nodes required for mass lumping without loss of accuracy.

There are several finite-element techniques for avoiding the inversion of the large sparse matrix, for example, spectral elements, continuous mass-lumped finite elements and discontinuous Galerkin finite elements.

The spectral finite-element method employs Gauss–Lobatto nodes for discretization on hexahedra, leading to a diagonal mass matrix (Orszag 1980; Patera 1984; Rønquist and Patera 1987; Maday and Rønquist 1990; Komatitsch and Vilotte 1998; Komatitsch and Tromp 1999; Komatitsch *et al.* 1999; Mulder 1999; Tromp *et al.* 2008). However, hexahedral grids are less flexible than tetrahedral meshes, for instance, near pinch-outs. To overcome this problem, Taniguchi *et al.* (1996), Owen (1998) and Charara *et al.* (2011) suggested dividing each tetrahedron into four hexahedra, increasing the number of nodes per tetrahedron. According to Owen (1998), the resulting hexahedral meshes tend to be of poor quality.

A spatial discretization using any of the various discontinuous Galerkin methods leads to a block-diagonal system of equations that can be solved explicitly. There are two main approaches among several discretization schemes (Arnold *et al.* 2002). The first reformulates the wave equation into a first-order system of equations, using the velocity–stress formulation. Schemes using this approach include the central fluxes used by Etienne *et al.* (2010), the upwind fluxes based on a Riemann solver used by LeVeque (1990) and the ADER-DG scheme, which maintains the same accuracy in space and time (Dumbser and Käser 2006). The second formulation is an interior penalty method based on the second-order formulation of the wave equation, with pressure in the acoustic case and displacement for the elastic case. This class of methods includes symmetric interior penalty, non-symmetric and incomplete interior penalty methods (Rivière 2008).

Continuous mass-lumped finite elements allow for explicit time stepping at the expense of requiring additional degrees of freedom to maintain spatial accuracy. In this case, the linear system of equations has a diagonal form that is trivial to invert. This technique is straightforward for piecewise linear elements but requires particular quadrature rules and additional discretization nodes for higher-order schemes, in order to preserve accuracy (Fried and Malkus 1975; Cohen *et al.* 1995; Tordjman 1995; Mulder 1996; Chin-Joe-Kong *et al.* 1999).

Various authors have made comparisons between the different methods (Fornberg 1987; Pasquetti and Rapetti 2004; Chaljub *et al.* 2007; De Basabe and Sen 2007; Wang *et al.* 2010; Moczo *et al.* 2011). Minisini *et al.* (2012) compared continuous mass-lumped and discontinuous Galerkin finite elements in terms of accuracy, stability and computational cost. Experiments on a three-dimensional problem showed that both methods are similar in terms of stability, accuracy and computational time required to obtain a result with a given accuracy.

Here, we compare the continuous mass-lumped finite-element method on tetrahedral unstructured meshes with the finite-difference method on uniform rectangular meshes. In 2D, Mulder (1996) showed that higher-order mass-lumped finite elements outperform the finite-difference method in terms of the computational time required to obtain a solution with a given numerical accuracy as soon as the velocity model has some non-trivial structure. In spite of the additional complexity of the finite-element method and the more restrictive stability limit for the time step, the use of larger triangles where the velocity is higher allows it to be more efficient than the finite-difference method. In 3D, it is less obvious whether

finite elements are superior in efficiency, since the 3D mass-lumped tetrahedral elements have a relatively large number of nodes compared with the standard element (Mulder 1996; Chin-Joe-Kong *et al.* 1999). To answer the question of efficiency, we consider several examples, from a simple homogeneous velocity model in a cube suitable for both methods to a fairly complex model with rough topography, which is outside the application area of high-order finite differences on rectangular meshes. This paper focuses on comparisons between the two methods in terms of their performance for a given accuracy. The stability analysis is covered by Zhebel *et al.* (2012).

The paper is organized as follows. The first two sections describe the continuous mass-lumped finite-element and standard finite-difference methods, respectively. Then, we define how the methods are compared in terms of accuracy. A set of numerical examples for problems of increased complexity is considered next and the two methods are compared in terms of accuracy and performance. Then, the scalability of the methods is considered. Finally, the conclusions are summarized.

FINITE ELEMENTS

We consider the constant-density acoustic wave equation,

$$\frac{1}{c^2}\frac{\partial^2 u}{\partial t^2} - \Delta u = f, \tag{1}$$

on a bounded domain $\Omega \subset \mathbb{R}^3$ with velocity $c(\mathbf{x})$, pressure $u(t, \mathbf{x})$ as a function of time t and position $\mathbf{x} = (x, y, z) \in \Omega$ and source $f(t, \mathbf{x})$. The Laplace operator Δ consists of the sum of second derivatives in each space direction. If the domain Ω has topography, then the free surface can have reflecting (zero pressure) or natural (zero normal derivative) boundary conditions. Absorbing boundary conditions are imposed elsewhere. The source term $f(t, \mathbf{x}) = w(t)s(\mathbf{x} - \mathbf{x}_s)$, with wavelet $w(t)$ and spatial source function s, is typically a delta function that peaks at the source position $\mathbf{x}_s$.

To discretize with finite elements, the initial wave equation (1) has to be modified. By multiplying (1) with a suitable test function v and integrating by parts (or using the divergence theorem), the weak formulation for the continuous finite elements is obtained as

$$\int_\Omega c^{-2} v \frac{\partial^2 u}{\partial t^2}\,\mathrm{d}\Omega + \int_\Omega \nabla u \cdot \nabla v\,\mathrm{d}\Omega - \int_\Gamma (\mathbf{n}\cdot\nabla u) v\,\mathrm{d}\Gamma = \int_\Omega v f\,\mathrm{d}\Omega, \tag{2}$$

for all test functions $v(\mathbf{x})$ in the Sobolev space $H^1(\Omega)$, $t \in [0, T]$. Note that $\boldsymbol{n}$ denotes the outward-directed normal on the outer boundary $\Gamma = \partial\Omega$ of the domain Ω.

The domain Ω can be partitioned into tetrahedral elements τ_m, $m \in \mathbb{N}$. Advantages of tetrahedral elements are their flexibility to follow geological interfaces and topography accurately and their ability to provide meshes that scale with velocity. This latter advantage is similar to the requirement of having a certain number of points per wavelength in finite-difference modelling. With tetrahedra, we require the diameter of the inscribed sphere to scale with the dominant wavelength, and hence with the local velocity. A meshing approach with this property has been described by Kononov *et al.* (2012).

Here, we discretize (2) into continuous mass-lumped finite elements (Chin-Joe-Kong *et al.* 1999). This method, combined with an explicit time-stepping scheme, leads to a fully explicit method, since it produces a linear system of equations in diagonal form that is trivial to invert. The procedure of obtaining a diagonal matrix is called mass lumping. To avoid loss of accuracy after lumping, these elements have to be enriched with additional discretization nodes in the interior, which support polynomials of higher degree that vanish at the edges (Fried and Malkus 1975). At present, 2D elements on triangles are known for degrees 1 to 6 (Fried and Malkus 1975; Tordjman 1995; Cohen *et al.* 1995, 2001; Mulder 1996; Chin-Joe-Kong *et al.* 1999; Mulder 2013). The 3D extension to the tetrahedron (Mulder 1996) required higher-degree polynomials in the interior of the faces and of the tetrahedron and resulted in an element of degree 2 at the edges and degree 4 in the interior of the faces and elements. Chin-Joe-Kong *et al.* (1999) found two types of element with degree 3 at the edges. Figure 1 depicts a continuous mass-lumped finite element of second degree in 2D and 3D. The black nodes denote the same discretization nodes as for standard finite elements. The red nodes denote the augmented nodes needed to preserve accuracy during mass lumping. Lesage *et al.* (2010) and Zhebel *et al.* (2011) presented 3D applications of these elements.

Table 1 lists the number of discretization nodes per element. The first column presents the polynomial degree of an element. The second column lists the number of nodes for a single mass-lumped element of given degree. Since nodes on edges and faces are shared, the effective number of nodes that needs to be stored per element is smaller than the number of nodes required during the computations, assuming that the stiffness matrices are assembled on the fly at each time step to save storage. This effective number of nodes is listed in the third column. It is obtained by dividing the total number

Table 1 Degrees of freedom for mass-lumped finite elements. The second column represents the number of nodes per element required for calculations. The third column shows storage requirements per element obtained by dividing the observed number of degrees of freedom by the number of elements. The last column shows the number of nodes per element for standard finite elements.

	Mass-lumped		
Degree p	Compute	Store	Standard
1	4	0.17	4
2	23	8.4	10
3	50	25	20

of degrees of freedom by the number of elements for some of the finer meshes used in the computations. For reference, the last column lists the number of nodes for a single standard tetrahedral element. Table 1 clearly shows a trade-off between memory storage and computational time.

Let $\mathbf{x}_i$, $i = 1, \ldots, N$, denote the discretization nodes of a mass-lumped element of degree p, where N is given in the second column of Table 1. The solution u and a test function v can be expressed as a linear combination of basis functions ϕ_j,

$$v(\mathbf{x}) = \phi_i(\mathbf{x}) \ (i = 1, \ldots, N), \quad u(t, \mathbf{x}) = \sum_{j=1}^{N} u_j(t)\phi_j(\mathbf{x}). \tag{3}$$

The basis functions are the same for each element and are defined as Lagrangian polynomials with the property that $\phi_j(\mathbf{x}_i) = 1$ if $i = j$ and zero otherwise. The polynomials are chosen in such a way that mass lumping can be applied. To avoid loss of spatial accuracy compared with the original scheme, polynomials of higher degree that vanish at the edges have to be included in 2D (Fried and Malkus 1975), except for the lowest degree, $p = 1$. The corresponding basis functions for the faces are polynomials of degree p_f that vanish at the edges. In 3D, we need additional polynomials of degree p_i in the interior of the tetrahedron that vanish on the faces (Mulder 1996).

More precisely, the following requirements should be met. The elements should be conforming and have nodes that obey the symmetry of the tetrahedron. They must not lose accuracy due to lumping, and they must be unisolvent and have positive weights after lumping. Conforming means that the shape functions are continuous across elements. The spatial accuracy is preserved if polynomials up to degree $q = p + \max(p, p_\mathrm{f}, p_\mathrm{i}) - 2$ are integrated exactly by numerical quadrature, for shape functions of a maximum degree p and additional degrees p_f and p_i. Unisolvency refers to the requirement that the Lagrange interpolating polynomials can be uniquely determined without any degeneracies. Positive integration weights are necessary for the stability of the explicit time-stepping scheme. The construction of the basis functions for mass-lumped tetrahedral elements has been described in detail by Chin-Joe-Kong *et al.* (1999).

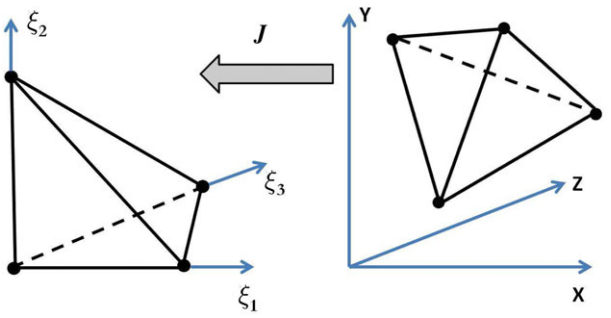


Figure 2 Mapping from an element to the reference element.

By inserting the representation (3) in (2), we obtain a system of second-order ordinary differential equations

$$\mathcal{M}\mathbf{u}_{tt} + \mathcal{K}\mathbf{u} = \mathbf{f}, \tag{4}$$

where $\mathcal{M}$ and $\mathcal{K}$ are the global mass and stiffness matrices, respectively. Both matrices are symmetrical due to the construction. The spatial operator is then given by

$$\mathcal{L} = \mathcal{M}^{-1}\mathcal{K}. \tag{5}$$

Lumping of the mass matrix produces a diagonal matrix that is easily inverted, without the need to solve a linear system of equations.

Consider the reference tetrahedron $\tilde{\tau}$ that is located at the origin of a Cartesian coordinate system (Figure 2). Let $\xi_i, i = 0, \ldots, 3$, with $\xi_0 = 1 - \xi_1 - \xi_2 - \xi_3$, be the barycentric coordinates on the reference tetrahedron. An arbitrary element τ_m in the mesh can easily be transformed to the reference element using the Jacobian of the coordinate transformation. Then, the contribution of each element τ_m to the mass matrix $\mathcal{M}$ is c_m^{-2} $\det(J_\mathrm{m})A$, where c_m is the local velocity in τ_m, J_m is the Jacobian of the coordinate transformation as shown in Figure 2 and

$$A_{k,l} = \int_{\tilde{\tau}} \phi_k \phi_l \, \mathrm{d}\xi_1 \mathrm{d}\xi_2 \mathrm{d}\xi_3 \tag{6}$$

is evaluated on the reference element $\tilde{\tau}$ with basis functions ϕ_k, $k = 1, \ldots, N$. If mass lumping is used, the matrix A is replaced by a diagonal matrix obtained from row sums

$\bar{A}_{k,k} = \sum_{l=1}^{N} A_{k,l}$. These values match the numerical integration weights.

The contribution to the global stiffness matrix $\mathcal{K}$ for each tetrahedron τ_m is $\sum_{k,l} C^\mathrm{m}_{k,l} B^{k,l}$, where

$$B^{k,l}_{i,j} = \int_{\bar{\tau}} \frac{\partial \phi_i}{\partial \xi_k} \cdot \frac{\partial \phi_j}{\partial \xi_l} \mathrm{d}\xi_1 \mathrm{d}\xi_2 \mathrm{d}\xi_3 \tag{7}$$

and $C^\mathrm{m}_{k,l}$ are the entries of $C^\mathrm{m} = \det(J_\mathrm{m}) J_\mathrm{m}^{-1} J_\mathrm{m}^{-\mathrm{T}}$. The matrices $B^{k,l}$ on the reference tetrahedron remain fixed and are hard-coded into the programme. Note that $B^{k,l}_{i,j} = B^{l,k}_{j,i}$, which together with the symmetry of C^m can be used to reduce the number of operations. The contribution then becomes

$$\bar{A}^{-1}_{i,i} \sum_{k=1}^{3} \left[C^\mathrm{m}_{k,k} \sum_j B^{k,k}_{i,j} u_j + \sum_{l=k+1}^{3} C^\mathrm{m}_{k,l} \sum_j \left(B^{k,l}_{i,j} + B^{l,k}_{i,j} \right) u_j \right]. \tag{8}$$

If the stiffness matrix is computed on the fly to save storage, the cost of assembly per element will depend on the number of nodes per element, as given in column 2.

For the simplest element, $p = 1$, which only has the four vertices as nodes, it is more efficient to evaluate the contribution of the stiffness matrix directly for each element than to use precomputed matrices $B^{k,l}$. In that case, we end up with a contribution of the form $(6 \det(J_\mathrm{m}))^{-1} G_\mathrm{m} G_\mathrm{m}^\mathrm{T} \mathbf{u}$ per element τ_m, where G_m^T is a matrix containing outer products as columns:

$$G_\mathrm{m}^\mathrm{T} = \begin{pmatrix} (\mathbf{x}_3 - \mathbf{x}_1) \times (\mathbf{x}_2 - \mathbf{x}_1) & (\mathbf{x}_2 - \mathbf{x}_0) \times (\mathbf{x}_3 - \mathbf{x}_0) \\ (\mathbf{x}_3 - \mathbf{x}_0) \times (\mathbf{x}_1 - \mathbf{x}_0) & (\mathbf{x}_1 - \mathbf{x}_0) \times (\mathbf{x}_2 - \mathbf{x}_0) \end{pmatrix} \tag{9}$$

and $\mathbf{x}_i$, $i = 0, \ldots, 3$, denote the vertices of the element τ_m. By choosing a symmetrical time-marching scheme, for example 'leapfrog', we obtain a fully algebraic system:

$$\mathbf{u}^{n+1} = 2\mathbf{u}^n - \mathbf{u}^{n-1} - \Delta t^2 (\mathcal{L}\mathbf{u}^n - \mathcal{M}^{-1}\mathbf{f}^n). \tag{10}$$

The only unknown is the vector $\mathbf{u}^{n+1}$. The values of the solution at the previous time steps n and $(n-1)$ are known. The storage space of $\mathbf{u}^{n-1}$ can be reused for $\mathbf{u}^{n+1}$. For computational efficiency, the inverse mass matrix is computed before starting the time steps. It requires the same amount of storage as the solution at one time instance, of which we need 2. To save storage, the stiffness matrices can be recomputed in each time step. Higher-order time-stepping methods can be readily implemented (Dablain 1986).

The time step is chosen according to the stability condition

$$\Delta t \leq \mathtt{CFL} \min_{\tau_\mathrm{m}} \left(\frac{d_\mathrm{m}}{c_\mathrm{m}} \right) \leq \frac{2}{\sqrt{\rho(\mathcal{L})}}, \tag{11}$$

where d_m and c_m are the diameter and velocity in the element τ_m, CFL denotes the Courant–Friedrichs–Lewy number (Courant *et al.* 1928). The spectral radius of the spatial operator, which is its largest absolute eigenvalue, is denoted $\rho(\mathcal{L})$.

Table 2 Estimated values of CFL, determining the time-stepping stability limit, for mass-lumped elements of degree p. Note that there are two types of third-degree element. Three different measures of the element diameter are considered: the diameter of the inscribed sphere (d_i), a measure based on the sum over the eigenvalues (d_b) and a measure based on the spectral radius of the spatial operator $\mathcal{L}$ (d_e) for the element of degree $p = 1$ on a single tetrahedron with homogeneous Neumann boundary conditions and unit velocity.

		CFL		
p	Type	d_i	d_b	d_e
1		0.72	1	0.77
2		0.062	0.079	0.063
3	1	0.036	0.048	0.034
3	2	0.066	0.088	0.065

Table 3 Values of CFL for 3D finite differences with order of discretization M and second-order time stepping.

M	2	4	6	8	10	12	14	16
CFL	0.57	0.50	0.47	0.45	0.44	0.43	0.43	0.42

Zhebel *et al.* (2012) analyzed the time-stepping stability for the continuous mass-lumped finite-element method. There estimates of the CFL values were based on the spectral radius $\rho(\mathcal{L})$ of the spatial operator for a particular case. They considered two cases: a single reference element with homogeneous Neumann boundary conditions for mass-lumped elements of arbitrary shape and a distorted unit cube packed with six tetrahedral elements with periodic boundary conditions. Note that, to avoid too strong distortions, the ratio of the diameter of the circumscribed sphere to the diameter of the inscribed sphere was chosen to be smaller than 100 in that paper. We assume that a mesh generator is able to produce such meshes. Moreover, three different measures of the element diameter were analyzed: the diameter of the inscribed sphere (d_i), a measure based on the sum over the eigenvalues (d_b) and a measure based on the spectral radius of the spatial operator $\mathcal{L}$ (d_e) for the element of degree $p = 1$ on a single tetrahedron with homogeneous Neumann boundary conditions and unit velocity. The resulting CFL for a given diameter measure is then the minimum over all distortions of the reference element and the periodic cube. Mass-lumped elements of degree $p = 1$, 2 and 3 were considered. Table 2 summarizes the results of that study. Among the three diameters, d_b produced

Table 4 With time-stepping of order M_t, the values of CFL from Table 3 must be multiplied by σ_t. For large M_t, these estimates approach either $\pi/2$ or π in an alternating fashion.

M_t	2	4	6	8	10	12	14	16
σ_t	1	1.73	1.38	2.32	1.54	2.77	1.57	3.04

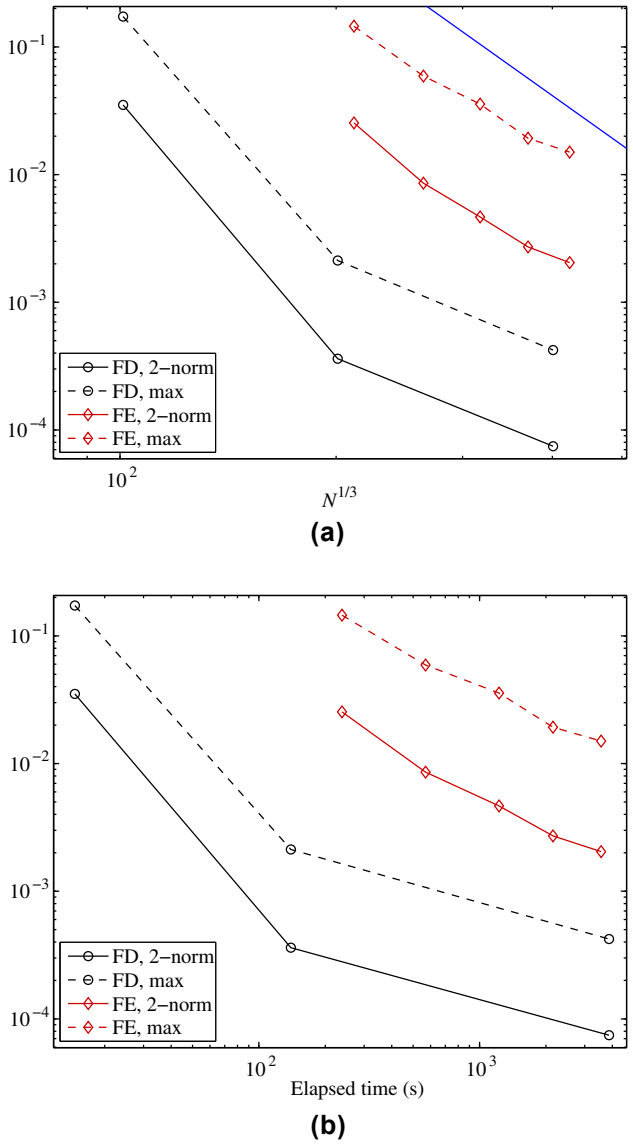


Figure 3 (a) Errors for problem **P1** as a function of $1/h = N^{1/3}$, where N is the number of grid points for the finite-difference method (FD) or the number of degrees of freedom for the finite-element method (FE). Dashed lines represent the maximum norm; solid lines, the 2-norm. The blue line marks the theoretical spatial convergence rate for the finite elements. Second-order time stepping was used. (b) Errors as a function of computational time on 12 cores.

the smallest spread in stability limit as a function of distortion and is therefore the most attractive measure. Later numerical experiments, however, still used a time step based on d_i, the diameter of the inscribed sphere.

FINITE DIFFERENCES

The standard finite-difference method is given on a regular Cartesian grid and uses central finite differences. In this case, the domain Ω is given by

$$\Omega = \{(x_i, y_j, z_k) \mid x_i = i\Delta x,\ y_j = j\Delta y,$$

$$z_k = k\Delta z,\ i, j, k \in \mathbb{N}\},$$

where Δx, Δy, Δz are grid sizes in the x, y and z directions, respectively. A second derivative of M-th order in the x direction at the point x_i is equal to

$$\left.\frac{\partial^2 u}{\partial x^2}\right|_i \simeq -\tfrac{1}{\Delta x^2}\left(w_0 u_i + \sum_{k=1}^{M/2} w_k \left(u_{i+k} + u_{i-k}\right)\right) \tag{12}$$

$$=: (D_{xx}\mathbf{u})_i,$$

where

$$w_0 = \sum_{j=1}^{M/2} \frac{2}{j^2}, \qquad w_k = (-1)^k \sum_{j=k}^{M/2} \frac{2}{j^2} \frac{(j!)^2}{(j-k)!(j+k)!},$$

for $k = 1, \ldots, M/2$. We refer to, e.g., Fornberg (1988) for details of this scheme and to Holberg (1987), Kindelan *et al.* (1990), and Tam and Webb (1993) for alternative ways of calculating the finite-difference coefficients. Derivatives in the other coordinate directions follow the same pattern. By discretizing the second-order derivative in time with the explicit leapfrog scheme (second-order central differencing) and in space with (12), equation (1) becomes

$$u^{n+1}_{i,j,k} = 2u^n_{i,j,k} - u^{n-1}_{i,j,k} + \Delta t^2 c^2_{i,j,k}\Big((D_{xx}\mathbf{u}^n)_{i,j,k} + (D_{yy}\mathbf{u}^n)_{i,j,k} + (D_{zz}\mathbf{u}^n)_{i,j,k} + f^n_{i,j,k}\Big), \tag{13}$$

or in vector form,

$$\mathbf{u}^{n+1} = 2\mathbf{u}^n - \mathbf{u}^{n-1} - \Delta t^2 C^2 (\mathcal{L}\mathbf{u}^n - \mathbf{f}^n), \tag{14}$$

where $\mathcal{L}$ denotes the spatial finite-difference operator and C is a diagonal matrix containing velocity. The solutions at different time steps are denoted by the superscript n for time $t^n = T_{\min} + n\Delta t$, with $n = 0, \ldots, N_T$ and $\Delta t = (T_{\max} - T_{\min})/N_T$. Sponge absorbing boundary conditions (Cerjan *et al.* 1985) allow for the simulation of non-reflecting boundaries. The

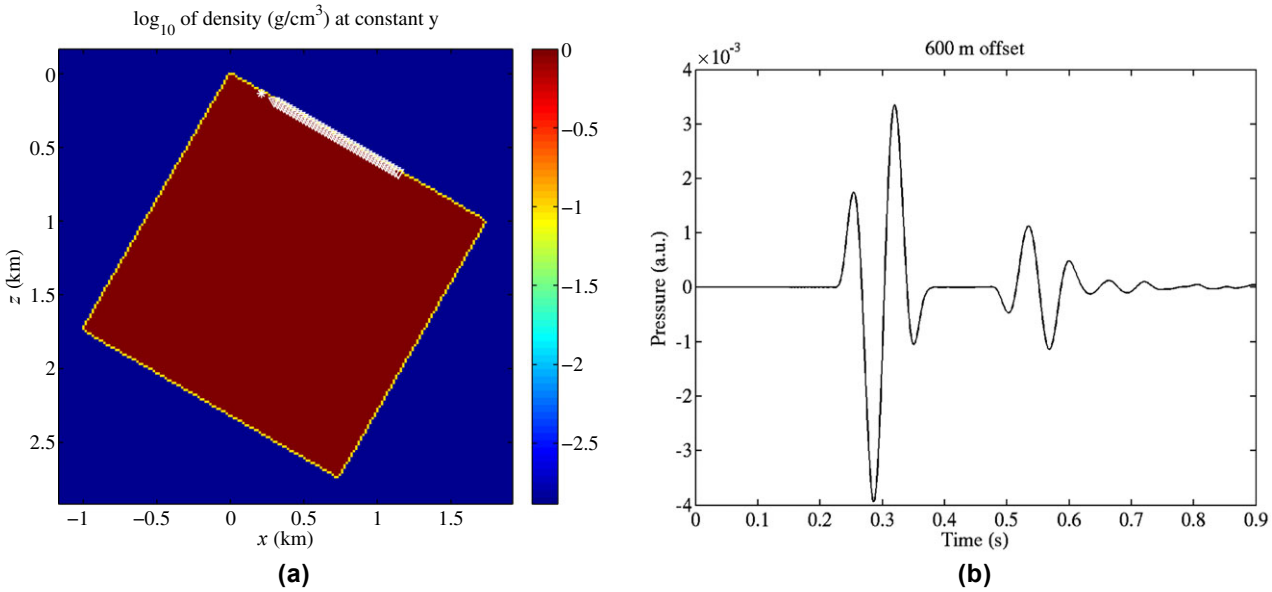


Figure 4 (a) Velocity model for problem **P2**. The source is denoted by a white star and receivers by white triangles. (b) The signal recorded at one of the receivers.

stability condition for time stepping, similar to (11), is expressed by

$$\Delta t \leq \texttt{CFL}\frac{d}{c_{\max}} \leq \frac{2}{\sqrt{\rho(\mathcal{L})}}, \tag{15}$$

where CFL denotes the Courant–Friedrichs–Lewy number and $c_{\max}$ is the maximum velocity over the whole computational domain. The diameter d is taken as

$$d = \left(1/\Delta x^2 + 1/\Delta y^2 + 1/\Delta z^2\right)^{-2}. \tag{16}$$

Standard Fourier analysis provides the spectral radius of the spatial operator on a periodic grid:

$$\rho(\mathcal{L}) \leq c_{\max}^2 \left(\frac{\rho(-D_{xx})}{\Delta x^2} + \frac{\rho(-D_{yy})}{\Delta y^2} + \frac{\rho(-D_{zz})}{\Delta z^2}\right). \tag{17}$$

If the grid spacings in all three coordinate directions are the same, we can take $\texttt{CFL} = 2/(\sqrt{3\rho(-D_{xx})})$, where

$$\rho(-D_{xx}) = \sum_{k=1}^{M/2} \frac{4^k}{k^2 \binom{2k-1}{k-1}}. \tag{18}$$

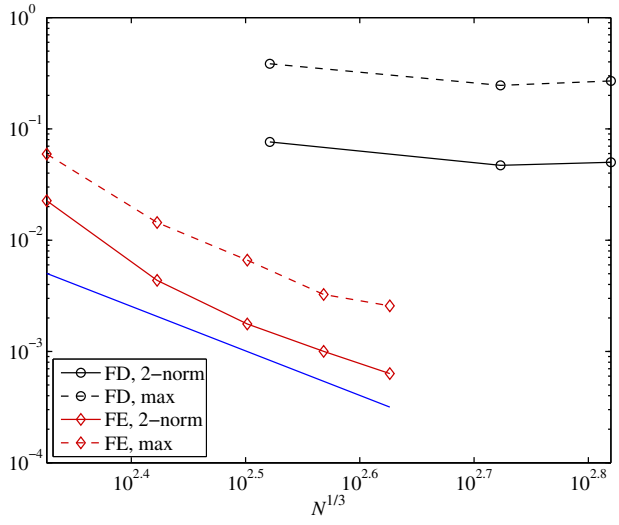
This is the truncated series of $4\arcsin^2(\sqrt{\xi})$ around $\xi = 0$, evaluated at $\xi = 1$. The result approaches π^2 with increasing order M. The resulting estimates for CFL for different orders of the discretization in three dimensions are given in Table 3.

Higher-order time stepping can be obtained by the Cauchy–Kovalewski or Lax–Wendroff procedure (Lax and Wendroff 1960), which replaces higher time-derivatives with spatial derivatives using the partial differential equation. This approach is also known as the modified equation approach (Shubin and Bell 1987) or Dablain's scheme (Dablain 1986). Higher-order time stepping enlarges CFL by a factor σ_t, which depends on the time-stepping order M_t, which should be an even number. Estimates for σ_t are considered elsewhere (Chen 2009; De Basabe and Sen 2010; Mulder *et al.* 2014). Table 4 lists the results. For the second-order time discretization, the factor σ_t equals one, meaning that the value of CFL is taken from Table 3. CFL for arbitrary order M in space and M_t in time obeys

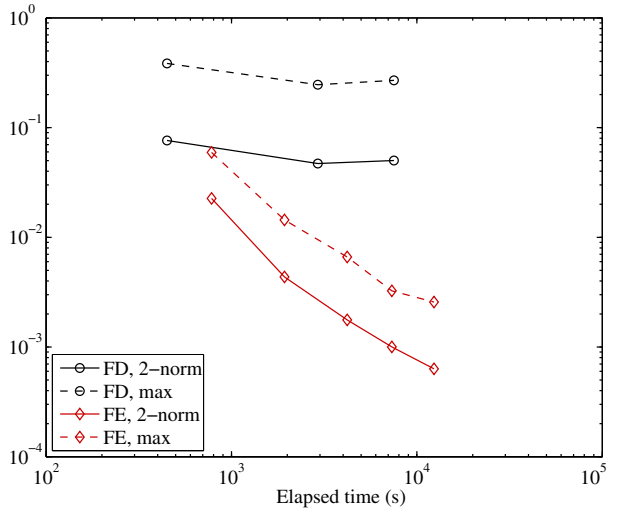
$$\texttt{CFL}(M_t, M) = \sigma_t \texttt{CFL}(2, M).$$

In the numerical examples, we have only presented results for second-order time stepping, as the time-stepping errors only started to show up at very small spatial errors.

Having the stability estimates in Table 2 and 3, let us have a look at the time step for finite elements and finite differences. On the one hand, by comparing values of CFL for finite differences with those of finite elements, we can see that the latter is at least 5 times smaller than the CFL for finite differences. On the other hand, the ratio of the diameter of the element to the velocity is larger in the case of finite elements, since the mesh scales with the velocity. With the acoustic wave equation, the time step for finite differences is comparable with that for finite elements, assuming the variation of the velocity in the model is more than a factor 5. However, with the elastic wave equation, where the shear velocity can be



(a)



(b)

Figure 5 (a) Errors for problem **P2** as a function of $1/h = N^{1/3}$, where N is the number of grid points for the finite-difference method (FD) or the number of degrees of freedom for the finite-element method (FE). Dashed lines represent the maximum norm; solid lines, the 2-norm. The blue line marks the theoretical spatial convergence rate for the finite elements. Second-order time stepping was used. (b) Errors as a function of computational time on 12 cores.

much smaller than the P-wave velocity, the situation can turn in favour of finite elements.

COMPARISONS

As has been shown by Zhebel *et al.* (2011), the higher-degree elements are more efficient for a given accuracy than those of lower degree. Therefore, for our comparisons, we consider the mass-lumped element of degree 3, type 2 (Chin-Joe-Kong *et al.* 1999), which has a more favourable time-stepping stability limit than type 1, as can be seen from Table 2. The element is enriched with polynomials of degree 5 in the interior of the faces and of degree 6 in the interior of the tetrahedron, leading to a total of 50 nodes for each element. It will result in a fourth-order spatial error.

For the finite-difference method, we chose the eighth-order spatial discretization. Formally, it has an eighth-order spatial error. However, in the case of large contrasts, the finite differences are expected to have a reduced accuracy of first- or second-order (Brown 1984; Symes and Vdovina 2009). Large finite-difference stencils allow us to choose fewer points per wavelength, leading to smaller computational grids (Alford *et al.* 1974).

To simulate topography in the finite-difference code, we took the approach of Bartel *et al.* (2000). We added an extra layer on top of the computational domain to make it rectangular, taking the velocity of the top layer of the solid model, but a low density of 0.0013 g/cm^3. In the solid part of the model, the density was set to 1 g/cm^3. To avoid instabilities, an extra point with a density of 0.1 g/cm^3 was added just above the free surface, as recommended by Bartel *et al.* (2000). We preferred to simulate the topography with a density contrast rather than with a velocity contrast, to avoid air wave artefacts in the seismic data. Another advantage of this approach is the reduced size of the computational grid, since the spacing in each direction depends on the minimum velocity in the model. The disadvantage of the large density contrast is a reduction of the CFL stability bound below the usual von Neumann stability estimate listed in Table 3. The source was represented as a tapered sinc function, a variant of the one proposed by Hicks (2002).

We need an estimate of the numerical errors produced by the codes as a function of the elapsed time. If the exact solution is known, the L_2-error is computed for one receiver by

$$\|e\|_{L_2} = \frac{\left((N_T+1)^{-1} \sum_{n=0}^{N_T} \left[u^n - u^n_{\text{exact}} \right]^2 \right)^{1/2}}{\max_n |u^n_{\text{exact}}|}, \tag{19}$$

where n denotes the time and N_T is the maximum number of time steps. For the maximum error, we take

$$\|e\|_{\max} = \frac{\max_n |u^n - u^n_{\text{exact}}|}{\max_n |u^n_{\text{exact}}|}. \tag{20}$$

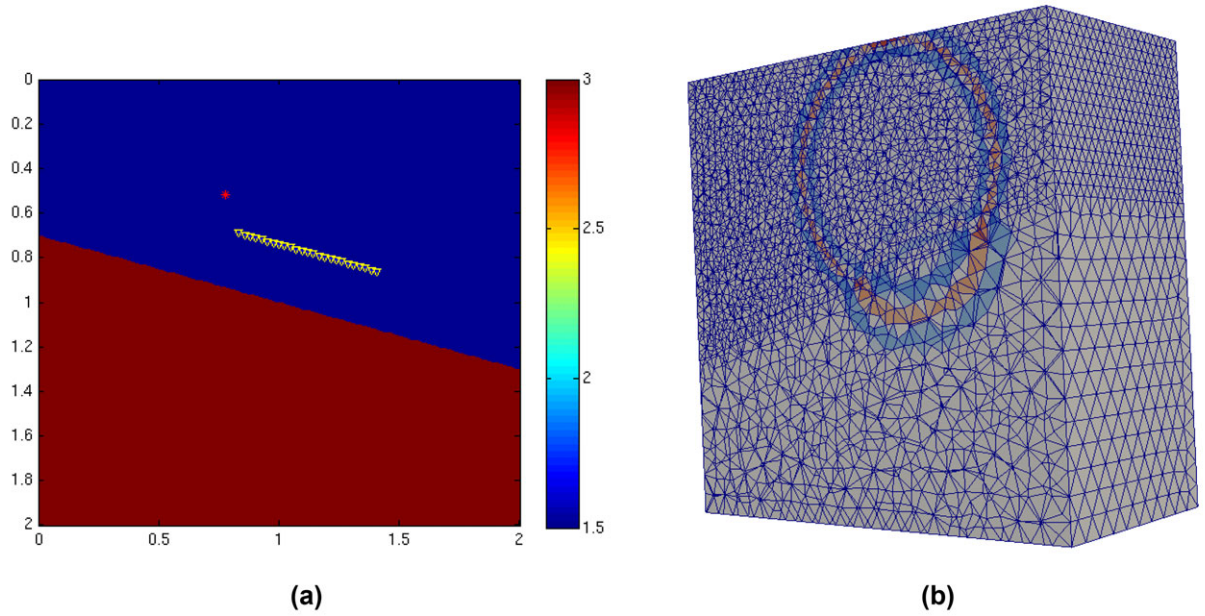


Figure 6 (a) Velocity model for problem **P3**. The source is denoted by a red star and receivers by yellow triangles. (b) Snapshot of the wave field.

If the exact solution is unknown, the absolute error can be estimated from the relative errors, obtained by comparing the differences between solutions on a sequence of grids of various mesh sizes. Wang *et al.* (2010) presented an example of this approach in a geophysical context. Let h be a measure of the grid spacing. Instead of the spacing, we take $h = N^{-1/3}$, where N is the number of grid points of the finite-difference mesh or the number of degrees of freedom of the finite-element discretization. We expect the spatial error to behave as some power q of h:

$$u^h - u^h_{\text{exact}} \simeq a h^q, \tag{21}$$

where a is a constant depending on position and time, u^h is the numerical solution on a given mesh and u^h_{exact} is the projection of the exact solution to the same mesh. The relative error on two meshes with h_1 and h_2 will then behave as

$$\left\| u^{h_1} - u^{h_2} \right\| \simeq \|a\| \, |h_1^q - h_2^q|. \tag{22}$$

For the norm, we will consider the maximum norm and the standard L_2 norm. Given runs on four different grids, we can estimate $\|a\|$ and q with a nonlinear fit, assuming that the spatial error dominates over the time-stepping error. Since the error behaviour is only an asymptotic estimate, additional runs may be useful to assess the quality of the result.

RESULTS

In this section, we compare finite differences and finite elements on several problems starting with simple ones and later including models with topography and complex geometries. In all cases, we discretize the subsurface either using tetrahedral meshes that scale with the velocity for the finite-element method or using a uniform Cartesian grid for the standard finite-difference method.

The first model, **P1**, represents a homogeneous cube of size $[0, 2]\,\text{km} \times [0, 2]\,\text{km} \times [0, 2]\,\text{km}$ with a constant velocity $c = 1.5$ km/s and with reflecting boundary conditions. The source is taken as a 12-Hz Ricker wavelet and is located at (1, 1, 1.5) km. A single receiver is placed at (1, 1, 0.5) km. The maximum time of the experiment is 1.2 s. We discretized this problem with continuous mass-lumped finite elements of degree $p = 1$, 2 and 3 on a sequence of unstructured tetrahedral meshes with decreasing diameters of the tetrahedra. A sequence of finite-difference grids had spacings of 20, 10 and 5 m and grids of sizes 101^3, 201^3 and 401^3, and the time steps were 2, 4/3 and 2/3 ms, respectively. The finite-element meshes had values for $N^{1/3}$ of 211.8, 264.6, 317.3, 370.1 and 422.9, with N the total number of degrees of freedom. The time steps were 1.25, 1, 0.8, 0.7 and 0.625 ms, respectively. Figure 3 displays the results. Note that for finer meshes, the second-order time-stepping error starts to appear. It is obvious that for this cube with constant velocity and density,

finite differences beat finite elements by at least one order of magnitude in speed.

Next, we take the homogeneous cube, rotate it by 30°, drape an air layer around it and project it on a grid in the original coordinates. The resulting model is denoted ***P*2** and is illustrated in Figure 4. This problem is simple enough to allow for an exact solution of the wave field inside the rotated cube, but at the same time it is complex enough to simulate the effect of the topography. The air is simulated by a density contrast, as described in the previous section, with the values 0.0013, 0.1, 1 g/cm^3, the latter being the density inside the cube. The density contrast has only been used for the finite differences, since the topography is naturally embedded in the tetrahedral finite-element meshes. Source and receivers are close to the boundary of the rotated cube and are shown as the white star and white triangles in Figure 4(a). The signal recorded at one of the receiver positions is presented in Figure 4(b). The wavelet is affected by the surface ghost and the later arrival is the reflection from the left boundary of the rotated cube.

Figure 5 shows the errors for data recorded at the central receiver as a function of the degrees of freedom N and the computational time for the model problem ***P*2**, using either the $p = 3$ type 2 continuous mass-lumped finite-element method or the finite-difference scheme with an eighth-order approximation of the second derivatives in each coordinate direction. The finite-difference code was run with a time step of 0.5 ms on a 301 × 201 × 301 grid, 0.4 ms on a 382 × 251 × 382, and 0.18 ms on a 758 × 501 × 758 grid. The finite-element code ran with time steps of 0.9, 0.72, 0.6, 0.5, 0.45 ms for problems with 384 000, 750 000, 1 296 000, 2 058 000 and 3 072 000 elements. The convergence for the mass-lumped finite elements is close to the theoretical order 4, whereas the finite-difference method has problems in converging in this setting.

The third model, ***P*3**, is the dipping interface shown in Figure 6(a). We consider a 3D domain of size [0, 2] km × [0, 2] km × [0, 2] km with two half spaces having velocities of 1.5 and 3.0 km/s. The interface runs at a depth from 0.7 to 1.3 km for x of 0 and 2 km, so the dip angle is 16.7°. A shot is located at (779.7, 1000, 516.3) m, 350 m above the interface in the shallow low-velocity part of the model. The receivers are located 250 m above the interface and have offsets from 100 to 700 m with a 25-m interval, parallel to the interface in the down-dip direction of the source.

Figure 7 shows the errors as a function of the degrees of freedom N and the computational time, using either the $p = 3$ type 2 continuous mass-lumped finite-element method or the finite-difference scheme with an eighth-order approximation of the second derivatives in each coordinate direction. The finite-difference code was run at 83% of the stability limit, with a time step of 1 ms on a 251^3 grid, 0.5 ms on a 501^3 grid and 0.25 ms on a 1001^3 grid. The finite-element code ran with time steps of 0.5, 0.35 and 0.2 ms at about 0.93 to 0.97% of the stability limit on the meshes with 294 508, 567 071

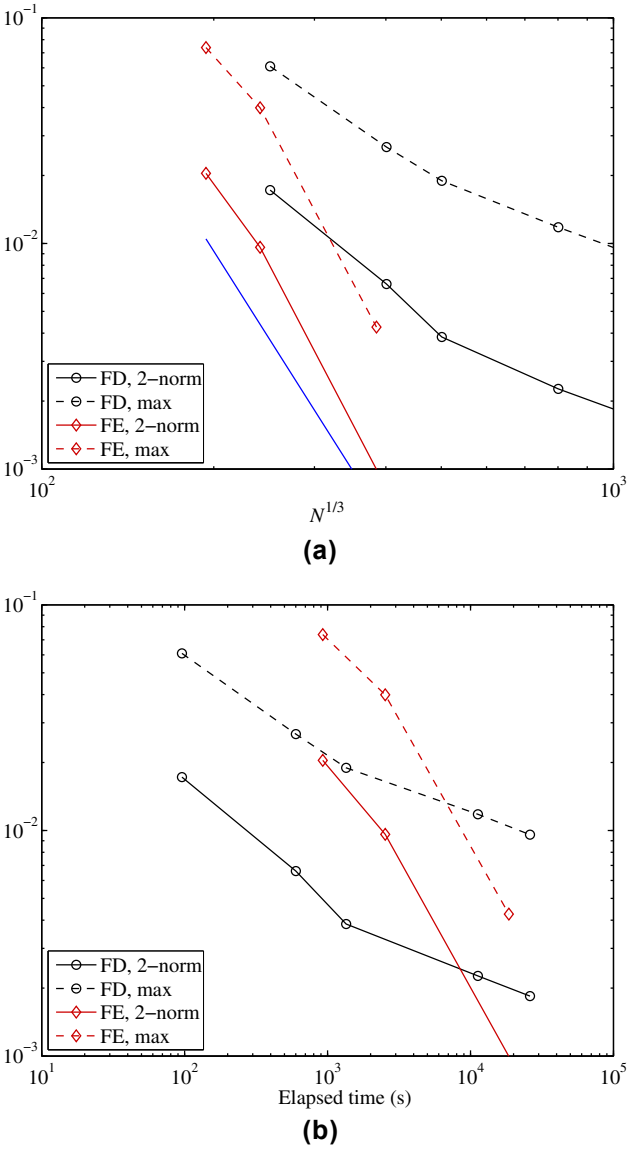


Figure 7 (a) Errors for problem **P3** as a function of $1/h = N^{1/3}$, where N is the number of grid points for the finite-difference method (FD) or the number of degrees of freedom for the finite-element method (FE). The dashed lines represent the maximum norm; the solid lines, the 2-norm. The blue line corresponds to a spatial fourth-order error. Second-order time stepping was used. (b) Errors as a function of computational time on 12 cores.

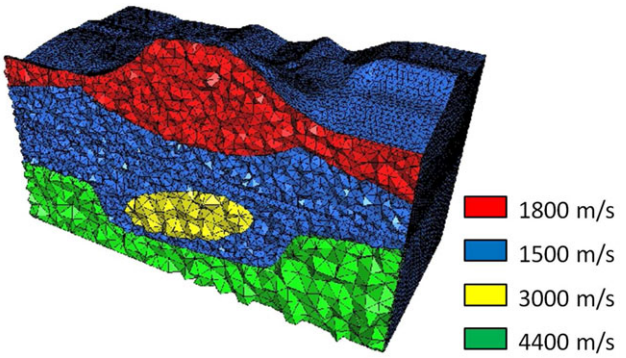


Figure 8 Vertical cross section at $y = 2$ km through the tetrahedral mesh for the model problem **P4**.

and 2 320 289 elements, respectively. For both finite differences and finite elements, the direct wave was modelled on the same mesh but with a constant velocity of 1.5 km/s and then subtracted to leave only the reflected or refracted event. The exact solution was computed in the frequency–wavenumber domain and then transformed to time and space. It is obvious that, asymptotically, the continuous mass-lumped finite elements are more accurate than the finite differences because the mesh follows the interface. However, finite differences show better performance when the desired accuracy is not very high.

Figure 8 depicts a section through a tetrahedral mesh used in the finite-element computations for our fourth model problem, **P4**, with $x_{\min} = 0$ m, $x_{\max} = 2000$ m, $y_{\min} = 0$ m, $y_{\max} = 2000$ m, $z_{\min} = -819.398$ m and $z_{\max} = 1800$ m. Again, the finite-element mesh was generated in such a way that the element sizes would scale with the local velocity. Figure 8 shows a vertical cross-section of the somewhat unrealistic velocity model. A shot was placed in the centre of the model, at x_s =2000 m, y_s =2000 m and 10 m below the surface. Receivers were placed from x_r =1012.5 m to 2987.5 m at 25-m intervals, y_r =2000 m and a depth of 10 m below the surface in the vertical direction. The continuous mass-lumped finite-element method of degree 3, type 2 was run on the problems with 641 746, 795 500 and 918 626 elements. In all cases, the time step was equal to 0.31 ms because the ratio of the diameter of an element to the velocity in this element was kept the same. The runs for finite differences were performed on grid sizes of $201 \times 201 \times 136$, $401 \times 401 \times 271$ and $801 \times 801 \times 541$, with time steps of 0.5, 0.25, 0.125 ms, respectively.

Figure 9(a) shows estimates of the absolute error, $||a||h^q$, in the trace data for finite differences and finite elements as a function of the degrees of freedom N to the power 1/3, using three runs for each, on the model problem **P4**. Although the finite-difference method formally has an eighth-order spatial error, the estimated error lies somewhere between first- and second-order, as expected, because the position of the interface is not known within the distance of a grid spacing, yielding a local first-order error, and because the solution has a discontinuous derivative across the interface, resulting in a local second-order error.

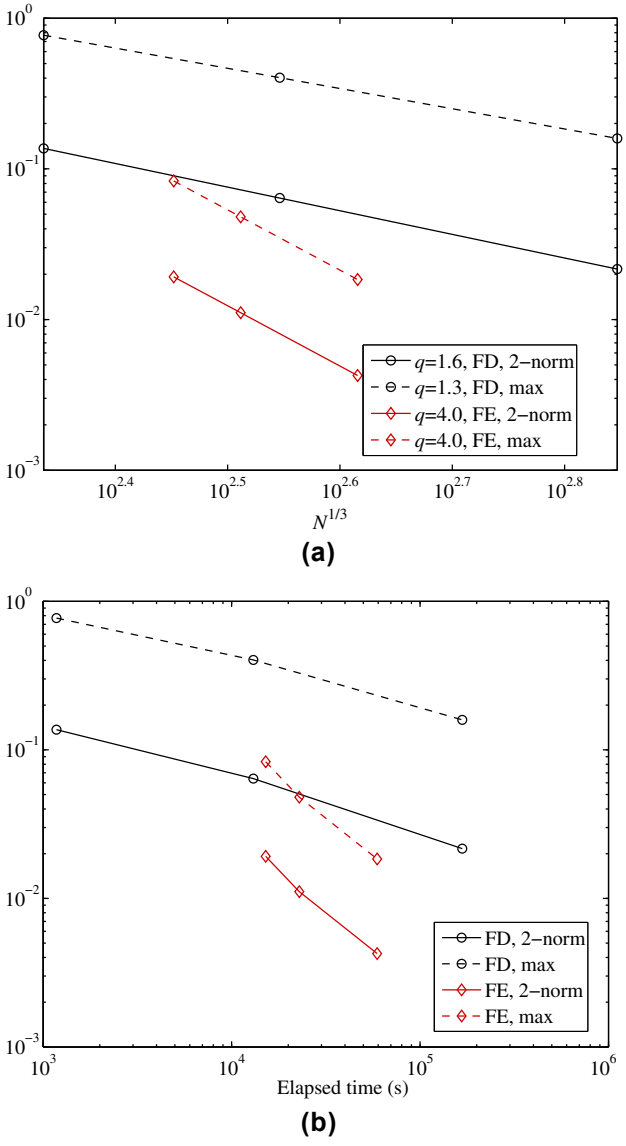


Figure 9 (a) Estimated errors for problem **P4** as a function of $1/h = N^{1/3}$, where N is the number of grid points for the finite-difference method (FD) or the number of degrees of freedom in the finite-element method (FE). Dashed lines represent the maximum norm; solid lines, the 2-norm. (b) Estimated errors as a function of computational time on 12 cores.

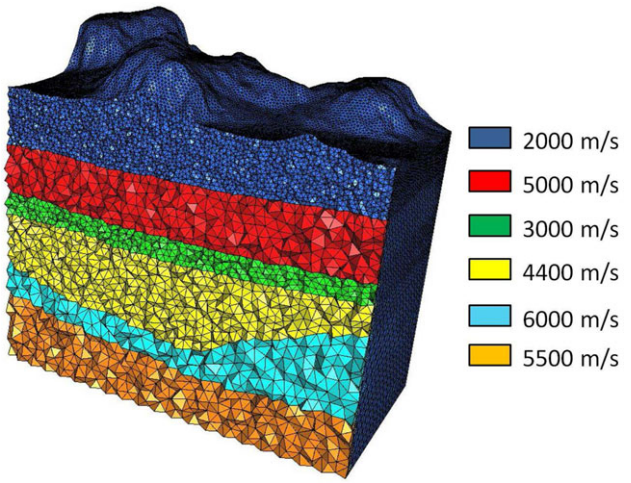


Figure 10 Section through the tetrahedral mesh for the model problem **P5**.

The experimentally estimated power for the asymptotic spatial error in the finite-element method is better than 5, but was set to its theoretical value of 4, giving less optimistic and more conservative estimates. However, Figure 9(b) demonstrates that the efficiency of the finite-element method is superior. To achieve, for instance, an accuracy of 1%, the finite-element method is about two orders of magnitude faster than the finite-difference method. Results were obtained on 12 cores on a single board, using MPI for the finite-difference code and OpenMP with multi-threading for the finite-element code.

Figure 10 shows a vertical section of the fifth model problem, **P5**, with $x_{\min} = 0\,\text{m}$, $x_{\max} = 4460\,\text{m}$, $y_{\min} = 0\,\text{m}$, $y_{\max} = 4440\,\text{m}$, $z_{\min} = -919.642\,\text{m}$ and $z_{\max} = 3300\,\text{m}$. In addition to the mountainous topography, we inserted some high-velocity geological structures: a chalk layer marked in red and a salt body, in cyan. A shot was placed at the top middle of the domain with coordinates $x_s = 2089\,\text{m}$, $y_s = 2150\,\text{m}$, at a depth of 100 m below the surface in the vertical direction.

The accuracy and performance estimates for problem **P5** are given in Figure 11. The results are similar to problem **P4**. To achieve 1% accuracy, the continuous mass-lumped finite-element method is two orders of magnitude faster than the standard finite-difference method.

Of course, the timings depend on implementation details, code optimization, compilers and hardware. It should be mentioned that the finite-element method uses local assembly of the stiffness matrices on the fly at each time step to save storage. We have observed that global assembly of the matrix $\mathcal{M}^{-1}\mathcal{K}$ will speed up the code by a factor of about 3, far less than can be expected by considering the number of floating point operations. This is due to memory access and cache misses. The element of degree 1 was an exception. It allowed for more than one order of magnitude speed-up. We want to emphasize that these results should be taken only as crude indications of performance.

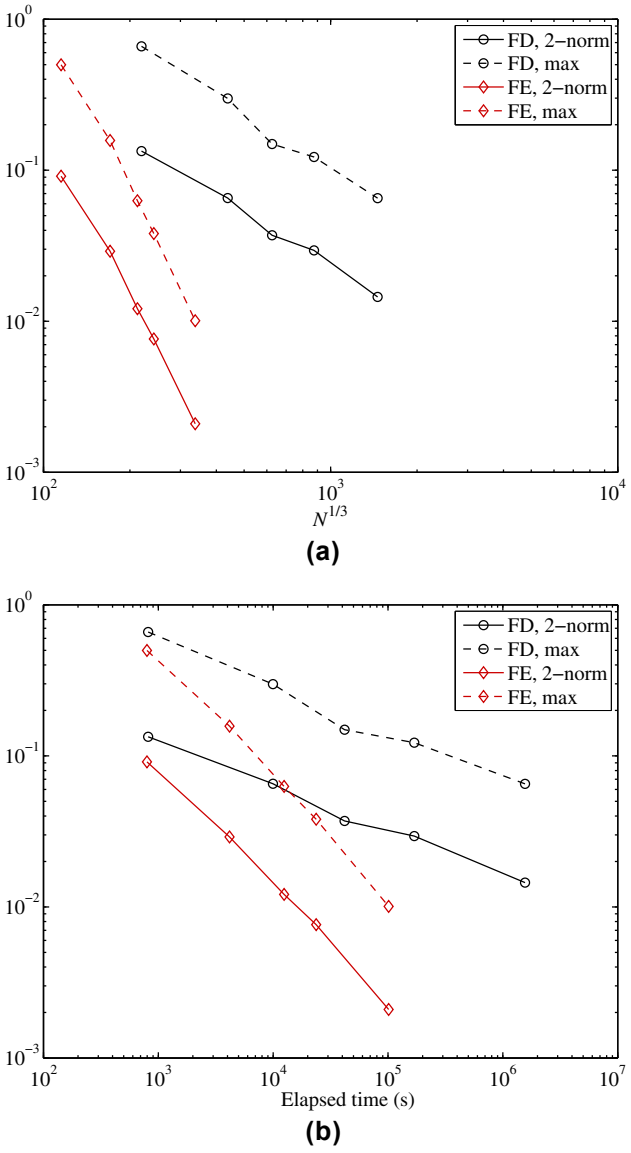


Figure 11 (a) Estimated errors for problem **P5** as a function of $1/h = N^{1/3}$, where N is the number of grid points for the finite-difference method (FD) or the number of degrees of freedom for the finite-element method (FE). Dashed lines represent the maximum norm; solid lines, the 2-norm. (b) Estimated errors as a function of computational time on 12 cores.

SCALABILITY

With constantly developing computational architectures, especially with multi-core processors, there is a need to adapt existing algorithms. Some algorithms have a better scalability than others, indicating how efficient an algorithm is when using an increasing numbers of cores. There are several ways to measure parallel performance; see Sun and Ni (1993) for an overview. For our experiments, we chose the strong scalability or fixed-size measure, where the problem size stays fixed but the number of cores is increased. Since the problems we are trying to solve are very large, so that they barely fit in the memory, the question arises: 'What is the performance of an algorithm applied to a given data set if more processors are used?' By performance, we mean the elapsed time of the time stepping without the preparation phase that involves reading the velocity model and constructing the discretization operators.

The finite-difference method is often considered easy to parallelize, because each point in space at a given time step can be treated independently of its neighbours. However, the computations require information about a number of neighbouring nodes, which is equal to the length of the finite-difference order M in (12). Basically, to calculate a value at one point in space, we need to perform $3(M+1)+4$ reads from memory and $3(M+4)+6$ arithmetic operations.

As mentioned before, the finite elements are also easily parallelized: the computation of values for one element at a given time step is independent of that for its neighbours. It only requires information about the discretization nodes of an element; information about the neighbours is not needed during the assembly on the fly. Assuming a mass-lumped element of degree 3, type 2, we need to treat 50 nodes per element (see Table 1), which requires six matrix–vector multiplications. The size of the matrices is 50×50. Therefore, finite elements are computationally more intensive than finite differences. Also, there are many more arithmetic operations than memory fetches.

Here, we compare the finite-difference method and continuous mass-lumped finite-element method in terms of efficiency. We define efficiency as the ratio of measured elapsed time on a single core to N times the elapsed time on N cores. Ideally, the efficiency equals one, meaning that running on N cores provides speed-up by a factor of N.

Table 5 lists the observed efficiency for finite differences and finite elements for problem **P3**. It is clear that the finite-element method scales better than the finite-difference method. This is because it is more computationally intensive and less memory-bound. However, the more cores used, the more memory-bound the code will be.

Table 5 Efficiency of finite differences (FD) and finite elements (FE) on multi-core architectures.

N cores	FD	FE
4	0.86	0.96
6	0.79	0.95
8	0.71	0.92
12	0.58	0.88

CONCLUSIONS

We have compared a fourth-order mass-lumped continuous finite-element method on tetrahedral unstructured meshes with a standard eighth-order finite-difference method on uniform rectangular meshes in terms of performance and accuracy, by which we mean the elapsed computational time it takes to obtain a numerical solution at fixed accuracy. For a model with constant velocity, the finite-element method is inefficient compared with the finite-difference method. At the other extreme, a model with sharp interfaces and topography, this particular type of finite-element method easily wins. We have also considered both methods in terms of scalability on the multi-core architectures. The finite-element method shows better scalability with the number of cores than the finite-difference method.

REFERENCES

Alford R.M., Kelly K.R., and Boore D.M., 1974. Accuracy of finite-difference modelling of the acoustic wave equation, *Geophysics*, **39**(6), 834–842.

Arnold D.N., Brezzi F., Cockburn B., and Marini L.D., 2002. Unified analysis of discontinuous Galerkin methods for elliptic problems, *SIAM Journal on Numerical Analysis*, **39**(5), 1749–1779.

Bartel L.C., Symons N.P., and Aldridge D.F., 2000. Graded boundary simulation of air/Earth interfaces in finite-difference elastic wave modeling, *SEG Technical Program Expanded Abstracts*, **19**(1), 2444–2447.

Brown D.L., 1984. A note on the numerical solution of the wave equation with piecewise smooth coefficients, *Mathematics of Computation*, **42**, 369–391.

Cerjan C., Kosloff D., Kosloff R., and Reshef M., 1985. A nonreflecting boundary condition for discrete acoustic and elastic wave equations, *Geophysics*, **50**(4), 705–708.

Chaljub E., Komatitsch D., Vilotte J., Capdeville Y., Valette B., and Festa G., 2007. Spectral element analysis in seismology, *in* R.S. Wu

and V. Maupin, *eds.*, *Advances in Wave Propagation in Heterogeneous Media, Advances in Geophysics*, pp. 365–419.

Charara M., Vershinin A., Sabitov D., and Pekar G., 2011. SEM wave propagation in complex media with tetrahedral to hexahedral mesh, in *73rd EAGE Conference & Exhibition, Vienna, Austria, Extended Abstracts*.

Chen J.-B., 2009. Lax-Wendroff and Nyström methods for seismic modelling, *Geophysical Prospecting*, **57**(6), 931–941.

Chin-Joe-Kong M.J.S., Mulder W.A., and van Veldhuizen M., 1999. Higher-order triangular and tetrahedral finite elements with mass lumping for solving the wave equation, *Journal of Engineering Mathematics*, **35**(4), 405–426.

Cohen G., Joly P., and Tordjman N., 1995. Higher order triangular finite elements with mass lumping for the wave equation, in *Proceedings of the Third International Conference on Mathematical and Numerical Aspects of Wave Propagation*, edited by G. Cohen, E. Bécache, P. Joly, and J. Roberts, pp. 270–279, SIAM, Philadelphia.

Cohen G., Joly P., Roberts J.E., and Tordjman N., 2001. Higher order triangular finite elements with mass lumping for the wave equation, *SIAM Journal on Numerical Analysis*, **38**(6), 2047–2078.

Courant R., Friedrichs K., and Lewy H., 1928. Über die partiellen Differenzengleichungen der mathematischen Physik, *Mathematische Annalen*, **100**(1), 32–74.

Dablain M.A., 1986. The application of high-order differencing to the scalar wave equation, *Geophysics*, **51**(1), 54–66.

De Basabe J.D. and Sen M.K., 2007. Grid dispersion and stability criteria of some common finite-element methods for acoustic and elastic wave equations, *Geophysics*, **62**(6), T81–T95.

De Basabe J.D. and Sen M.K., 2010. Stability of the high-order finite elements for acoustic or elastic wave propagation with high-order time stepping, *Geophysical Journal International*, **181**(1), 577–590.

Dumbser M. and Käser M., 2006. An arbitrary high-order discontinuous Galerkin method for elastic waves on unstructured meshes – II. The three-dimensional isotropic case, *Geophysical Journal International*, **167**(1), 319–336.

Etienne V., Chaljub J., Virieux J., and Glinsky N., 2010. An hp-adaptive discontinuous Galerkin finite elements method for 3-D elastic wave modelling, *Geophysical Journal International*, **183**(2), 941–962.

Fornberg B., 1987. The pseudospectral method: comparisons with finite-differences for the elastic wave equation, *Geophysics*, **52**(4), 483–501.

Fornberg B., 1988. Generation of finite difference formulas on arbitrarily spaced grids, *Mathematics of Computation*, **51**(184), 699–706.

Fried I. and Malkus D.S., 1975. Finite element mass matrix lumping by numerical integration with no convergence rate loss, *International Journal of Solids and Structures*, **11**, 461–466.

Hestholm S. and Ruud B., 1994. 2D finite-difference elastic wave modelling including surface topography, *Geophysical Prospecting*, **42**(5), 371–390.

Hicks G.J., 2002. Arbitrary source and receiver positioning in finite-difference schemes using Kaiser windowed sinc functions, *Geophysics*, **67**(1), 156–165.

Holberg O., 1987. Computational aspects of the choice of operator and sampling interval for numerical differentiation in large-scale simulation of wave phenomena, *Geophysical Prospecting*, **35**(6), 629–655.

Kindelan M., Kamel A., and Sguazzero P., 1990. On the construction and efficiency of staggered numerical differentiators for the wave equation, *Geophysics*, **55**, 107–110.

Komatitsch D. and Tromp J., 1999. Introduction to the spectral-element method for 3-D seismic wave propagation, *Geophysical Journal International*, **139**(3), 806–822.

Komatitsch D. and Vilotte J.P., 1998. The spectral-element method: an efficient tool to simulate the seismic response of 2-D and 3-D geological structures, *Bulletin of the Seismological Society of America*, **88**(2), 368–392.

Komatitsch D., Vilotte J.P., Vai R., Castillo-Covarrubias J.M., and Sánchez-Sesma F.J., 1999. The spectral element method for elastic wave equations: application to 2-D and 3-D seismic problems, *International Journal for Numerical Methods in Engineering*, **45**(9), 1139–1164.

Kononov A., Minisini S., Zhebel E., and Mulder W.A., 2012. A 3-D tetrahedral mesh generator for seismic problems, in *74th EAGE Conference & Exhibition, Copenhagen, Denmark, Extended Abstracts*.

Lax P. and Wendroff B., 1960. Systems of conservation laws, *Communications on Pure and Applied Mathematics*, **31**(2), 217–237.

Lesage A.C., Aubry R., Houzeaux G., Polo M. Araya, and Cela J.M., 2010. 3D spectral element method combined with h-refinement, in *72nd EAGE Conference & Exhibition, Barcelona, Spain, Extended Abstracts*.

LeVeque R.J., 1990. *Numerical Methods for Conservation Laws*, Lectures in Mathematics, Birkhauser-Verlag, ETH-Zurich.

Lombard B., Piraux J., Gelis C., and Virieux J., 2008. Free and smooth boundaries in 2-D finite-difference schemes for transient elastic waves, *Geophysics Journal International*, **172**, 252–261.

Maday Y. and Rønquist E.M., 1990. Optimal error analysis of spectral methods with emphasis on non-constant coefficients and deformed geometries, *Computer Methods in Applied Mechanics and Engineering*, **80**(1–3), 91–115.

Minisini S., Zhebel E., Kononov A., and Mulder W.A., 2012. Efficiency comparisons for higher-order continuous mass-lumped and discontinuous Galerkin finite-element methods for the 3-D acoustic wave equation, 74th EAGE Conference & Exhibition, Extended Abstracts, A004, Copenhagen, Denmark.

Moczo P., Kristek J., Galis M., Chaljub E., and Etienne V., 2011. 3-D finite-difference, finite-element, discontinuous-Galerkin and spectral-element schemes analysed for their accuracy with respect to P-wave to S-wave speed ratio, *Geophysics Journal International*, **187**(3), 1645–1667.

Mulder W.A., 1996. A comparison between higher-order finite elements and finite differences for solving the wave equation, in *Proceedings of the Second ECCOMAS Conference on Numerical Methods in Engineering*, edited by J. Désidéri, P. LeTallec, E. O nate, J. Périaux, and E. Stein, pp. 344–350, John Wiley & Sons, Chichester.

Mulder W.A., 1999. Spurious modes in finite-element discretisations of the wave equation may not be all that bad, *Applied Numerical Mathematics*, **30**, 425–445.

Mulder W.A., 2013. New triangular mass-lumped finite elements of degree 6 for wave propagation, *Progress in Electromagnetic Research*, **141**, 671–692.

Mulder W.A., Zhebel E., and Minisini S., 2014. Time-stepping stability of continuous and discontinuous finite-element methods for 3-D wave propagation, *Geophysical Journal International*, **196**(2), 1123–1133.

Orszag S.A., 1980. Spectral methods for problems in complex geometries, *Journal of Computational Physics*, **37**(1), 70–92.

Owen S.J., 1998. A survey of unstructured mesh generation technology, in *Proceedings of the 7th International Meshing Roundtable*, pp. 26–28.

Pasquetti R. and Rapetti F., 2004. Spectral element methods on triangles and quadrilaterals: comparisons and applications, *Journal of Computational Physics*, **198**(1), 349–362.

Patera A.T., 1984. A spectral element method for fluid dynamics: laminar flow in a channel expansion, *Journal of Computational Physics*, **54**(3), 468–488.

Preston L.A., Aldridge D.F., and Symons N.P.B., 2008. Finite-difference modeling of 3D seismic wave propagation in high-contrast media, in *78th Annual Meeting and International Exposition, 9-14 November 2008, Las Vegas, Nevada, USA*, Society of Exploration Geophysicists.

Rivière B., 2008. *Discontinuous Galerkin methods for solving elliptic and parabolic equations: Theory and implementation*, vol. **35** of **Frontiers in Mathematics**, SIAM, Philadelphia.

Rønquist E.M. and Patera A.T., 1987. A Legendre spectral element method for the Stefan problem, *International Journal for Numerical Methods in Engineering*, **24**(12), 2273–2299.

Shubin G.R. and Bell J.B., 1987. A modified equation approach to constructing fourth order methods for acoustic wave propagation, *SIAM Journal on Scientific and Statistical Computing*, **8**(2), 135–151.

Sofronov I.L., Zaitsev N.A., Daryin A., Voskoboinikova O., and Dovgilovich L., 2012. Multi-block FD method for 3D geophysical simulation with explicit representation of sub-horizontal interfaces, in *74th EAGE Conference & Exhibition, Extended Abstracts, A004, Copenhagen, Denmark*.

Sun X. and Ni L.M., 1993. Scalable problem and memory-bounded speedup, *Journal of Parallel and Distrubuted Computing*, **19**, 27–37.

Symes W.W. and Vdovina T., 2009. Interface error analysis for numerical wave propagation, *Computers & Geosciences*, **12**(3), 363–371.

Tam C.K.W. and Webb J.C., 1993. Dispersion-relation-preserving finite difference schemes for computational acoustics, *Journal of Computational Physics*, **107**, 262–281.

Taniguchi T., Goda T., Kasper H., and Zielke W., 1996. Hexahedral mesh generation of complex composite domain, in *5th International Conference on Grid Generation in Computational Field Simulations*, pp. 699–707, Mississippi State University.

Tarrass I., Giraud L., and Thore P., 2011. New curvilinear scheme for elastic wave propagation in presence of curved topography, *Geophysical Prospecting*, **59**(5), 889–906.

Tordjman N., 1995. *Élements finis d'order élevé avec condensation de masse pour l'equation des ondes*, Ph.D. thesis, L'Université Paris IX Dauphine.

Tromp J., Komatitsch D., and Liu Q., 2008. Spectral-element and adjoint methods in seismology, *Communications in Computational Physics*, **3**(1), 1–32.

Wang X., Symes W.W., and Warburton T., 2010. Comparison of discontinuous Galerkin and finite difference methods for time domain acoustics, *SEG Technical Program Expanded Abstracts*, **29**(1), 3060–3065.

Zhebel E., Minisini S., Kononov A., and Mulder W.A., 2011. Solving the 3D acoustic wave equation with higher-order mass-lumped tetrahedral finite elements, in *73rd EAGE Conference & Exhibition, Vienna, Austria, Extended Abstracts*.

Zhebel E., Minisini S., Kononov A., and Mulder W.A., 2012. On the time-stepping stability of continuous mass-lumped and discontinuous Galerkin finite elements for the 3D acoustic wave equation, in *CD-ROM Proceedings of the 6-th European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS 2012), September 10-14, 2012, VIenna, Austria*, Eds.: Ebergardsteiner J.; Böhm H.J.; Rammesdorfer F.G., Publisher: Vienna University of Technology, Austria, ISBN: 978-3-9502481-9-7 (11 pages).