Modular and flexible spectral-element waveform modelling in two and three dimensions

Michael Afanasiev [1], Christian Boehm,[1] Martin van Driel,[1] Lion Krischer [1],[1] Max Rietmann,[1] Dave A. May,[2] Matthew G. Knepley[3] and Andreas Fichtner[1]

[1]*Department of Earth Sciences, ETH Zürich, Zürich, Switzerland. E-mail: michael.afanasiev@erdw.ethz.ch*
[2]*Department of Earth Sciences, University of Oxford, Oxford, United Kingdom*
[3]*Department of Computer Science and Engineering, University at Buffalo, Buffalo, NY, USA*

SUMMARY
In this paper, we present a series of mathematical abstractions for seismologically relevant wave equations discretized using finite-element methods, and demonstrate how these abstractions can be implemented efficiently in computer code. Our motivation is to mitigate the combinatorial complexity present when considering geophysical waveform modelling and inversion, where a variety of spatial discretizations, material models, and boundary conditions must be considered simultaneously. We accomplish this goal by first considering three distinct classes of abstract mathematical models: (1) those representing the physics of an underlying wave equation, (2) those describing the discretization of the chosen equation onto a finite-dimensional basis and (3) those describing any spatial transforms. A full representation of the discrete wave equation can then be constructed using a hierarchical nesting of models from each class. Additionally, each class is functionally orthogonal to the others, and with certain restrictions models within one class can be interchanged independently from changes in another. We then show how this recasting of the relevant equations can be implemented concisely in computer software using an abstract object-oriented design, and discuss how recent developments in the numerical and computational sciences can be naturally incorporated. This builds to a set of results where we demonstrate how the developments presented can lead to an implementation capable of multiphysics waveform simulations in completely unstructured domains, on both hypercubical and simplical spectral-element meshes, in both two and three dimensions, while remaining concise, efficient and maintainable.

Key words: Numerical modelling; Computational seismology; Wave propagation.

1 INTRODUCTION

As mechanical waves pass through a medium, they acquire information about that medium. By investigating the character of the waves at a series of detectors, this information can be used to reconstruct a 3-D model of that medium's interior. This reconstruction is the goal of full-waveform inversion. First introduced theoretically in the 1980s (Lailly 1983; Tarantola 1984; Gauthier *et al.* 1986), increasing computational power has allowed for the technique's application to ever larger and more realistic domains. The method is now a workhorse in the seismic exploration industry (Pratt *et al.* 1998; Brenders & Pratt 2007; Virieux & Operto 2009; Prieux *et al.* 2013), where it is used to probe domains many kilometres in size for natural resources. In academic settings, the method is widely applied at the regional (Tape *et al.* 2009; Colli *et al.* 2013; Fichtner

et al. 2013; Simutė *et al.* 2016) and the global (French & Romanowicz 2014; Afanasiev *et al.* 2016; Bozdag *et al.* 2016; Fichtner *et al.* 2018) scales. Recent applications have also moved beyond the field of seismology, into medical imaging (Pratt *et al.* 2007; Korta Martiartu *et al.* 2017) and nondestructive material testing (Rao *et al.* 2016; Seidl & Rank 2017).

An essential ingredient in full-waveform inversion is a method to compute synthetic waveforms. As closed-form solutions exist for only the simplest of media, numerical solutions to the wave equation are essential to the method's success. The first, and likely still most common, method used for the numerical computation of waveforms is the finite-difference method (see Moczo *et al.* (2014) for a comprehensive review). First applied to wave propagation problems on regular grids (Alterman & Karal 1968; Boore 1970; Kelly *et al.* 1976), more sophisticated staggered-grid approaches

brought an increase in solution accuracy relative to computational cost (Madariaga 1976; Virieux 1984, 1986). Physical sophistication was further developed by the proper treatment of boundary conditions and material discontinuities (Robertsson 1996; Ohminato & Chouet 1997; Moczo *et al.* 2002), the incorporation of general anisotropy (Igel *et al.* 1995), and formulations in terms of spherical coordinates for axisymmetric (Igel *et al.* 1995; Chaljub & Tarantola 1997) and fully 3-D domains (Igel & Weber 1995). Other work focused on increasing the combined spatial and temporal accuracy using 'optimal' operators (Geller & Takeuchi 1998), and applications via natural neighbours to more general, unstructured domains (Braun & Sambridge 1995; Käser *et al.* 2001; Käser & Igel 2001).

While finite-difference methods are highly efficient for simulating wave propagation on rectilinear meshes, some factors limit their usefulness for domains with complex geometries, a complicated material structure, or which require more advanced boundary conditions. For studies in areas with strong topography, along with applications in medical imaging and nondestructive testing, this limitation can become especially severe. Recent work shows promising extensions of finite-difference methods to more complex domains via curvilinear coordinate transforms (Fornberg 1988; Hestholm 1999; de la Puente *et al.* 2014; Petersson & Sjögreen 2015; Shragge 2016), but these extensions still require a continuous and smooth surface, and are associated with a higher computational cost. Furthermore, the explicit implementation of boundary conditions becomes increasingly difficult as the geometric complexity of the domain evolves. Finally, increasing the spatial accuracy of the finite-difference operators is associated with a growth in stencil size which can negatively affect the parallel scaling characteristics in larger applications demanding distributed memory parallelism.

A common response to these limitations is the use of finite-element and finite-volume methods, which elegantly include the effects of boundary topography, and do not suffer from severe communication overheads as the spatial approximation order is increased. This is fortunate, as low-order variants of either method are too inefficient to be useful (Marfut 1984; Käser & Igel 2001). Recent work on arbitrarily high-order finite-volume methods has shown promising results (Dumbser *et al.* 2007), but the computational cost for a given solution accuracy seems to favour discontinuous Galerkin methods over finite-volume (de la Puente *et al.* 2007; Käser *et al.* 2007; Wilcox *et al.* 2010; Tago *et al.* 2012). A particular advantage of both these schemes is their efficient and arbitrarily high-order discretization on unstructured simplicial meshes (tetrahedra, triangles). While discontinuous Galerkin methods are more flexible than their continuous counterparts across elemental boundaries, they are computationally more expensive, with additional work needed to compute the flux terms while also requiring a more restrictive CFL condition as a recent study by Ferroni *et al.* (2017) shows.

Thus, in applications with complex bounded domains which can be meshed with hypercubic elements (quadrilaterals, hexahedra), the spectral-element method has emerged as the most efficient numerical technique for solving wave propagation problems (Seriani & Priolo 1994; Faccioli *et al.* 1996, 1997; Komatitsch & Tromp 1999; Capdeville *et al.* 2003; Chaljub *et al.* 2003; Fichtner & Igel 2008; Peter *et al.* 2011; Nissen-Meyer *et al.* 2014). A high-order variant of classical finite-element methods, the spectral-element method combines high-order interpolating polynomials on each element with a specific quadrature rule which results in a globally diagonal mass matrix. This feature, coupled with the fact that the element-wise operations can be formulated as dense matrix–matrix products, makes the method especially suitable for the current generation of SIMD (single instruction, multiple data) computing architectures. This is in contrast to classical finite-element approaches, in which the global mass matrix is *not* diagonal, and the time evolution of the wavefield may involve the solution of a sparse linear system at each time step.

Improvements in the quality and quantity of seismological data, as well as increasing computational power, make it desirable and possible to model wave propagation physics in more detail. This trend, coupled with the variety of appropriate spatial discretizations and the ever-expanding network of applications where waveform modelling is relevant, risks to complexify wave propagation codes to a level where they become unmaintainable. In this paper we describe a series of theoretical and practical developments useful for designing software for full-waveform modelling and inversion based on the spectral-element method, with a focus on time-domain approaches. The results are inspired by popular finite-element packages such as FEniCS (Logg *et al.* 2012), deal.II (Bangerth *et al.* 2007), GRINS (Bauman & Stogner 2016) and DUNE (Dedner *et al.* 2010) in that many details of the underlying numerical methods are abstracted away. The developments allow us to make additional generalizations regarding the implementation of the wave equation for a wide variety of rheologies. We discuss the use of distributed data structures provided by PETSc (Balay *et al.* 1997, 2017a,b) and DMPLEX (Knepley & Karpeev 2009; Lange *et al.* 2016), and demonstrate how such structures allow for scaling to problems involving more than hundreds of millions of unknowns, and discretized onto general, unstructured meshes. Finally, we present several benchmark results on both 2-D and 3-D simplicial and hypercubic meshes, along with several other proof-of-concept examples, demonstrating the variety of scales and domains to which our SEM implementation can be applied. While several high-quality spectral-element implementations focused on FWI currently exist (Komatitsch & Tromp 2002a; Cupillard *et al.* 2012; Gokhberg & Fichtner 2016), none encapsulates the flexibility offered by the abstraction presented herein.

Section 2 reviews the strong and weak forms of the elastic and acoustic wave equations, and illustrates how they can both be written in a similar functional form. Piece by piece, this form is then related to a general continuous Galerkin finite-element discretization. Following this, we see how the functional notation can also lend itself to a concise representation of wave propagation through more complex rheologies, such as viscoelasticity, and coupled acoustic/elastic media. Section 3 relates these abstractions in a one-to-one manner to concepts from software engineering, showing how they lead naturally to 'function decorators'. We then show some simple examples of how such decorators can be implemented with C++ template mixins. Following this is a demonstration of how the individual decorators can be used to automatically construct functionals in complex unstructured domains, using the unstructured mesh abstractions in DMPLEX. Finally, we comment on some preliminary performance and scaling statistics, and follow with a short discussion on the benefits of flexibility.

2 AN ABSTRACT FORMULATION OF THE SEISMIC WAVE EQUATION

In this section, we explain the motivation behind selecting the finite-element method as a spatial discretization. We then show how the mathematical details of the method can be abstracted and implemented efficiently on a computer. Finally, we show how the details of a particular wave equation can themselves be generalized, leading to efficient implementations of coupled domains and viscoelasticity.

2.1 The weak form

Consider the wave equation for a simple elastic solid in an n-dimensional domain G ($n = 1$, 2 or 3) and time interval $(0, T]$ with $T > 0$ (Aki & Richards 2002):

$$\rho \partial_t^2 \boldsymbol{u} = \nabla \cdot \boldsymbol{\sigma} + \boldsymbol{f} \quad \text{in } G \times (0, T]. \tag{1}$$

Here ρ is density, $\boldsymbol{u}$ is displacement, $\boldsymbol{\sigma}$ is stress and $\boldsymbol{f}$ represents an external forcing term. The time and space dependency of these last three terms is taken as implicit to simplify notation. A simple way to relate stress to displacement in a purely elastic medium is via Hooke's law

$$\boldsymbol{\sigma} = \boldsymbol{C} : \nabla \boldsymbol{u}, \tag{2}$$

where $\boldsymbol{C}$ is a fourth-order tensor characterizing the stiffness of the medium, and the symbol : denotes a contraction over adjacent indices. Taken together, eqs (1) and (2) define the strong form of the linear elastic wave equation. Boundary conditions (such as the free-surface conditions), along with quiescent initial conditions, must be included explicitly in order to obtain a unique solution. Boundary and initial conditions take the form

$$\boldsymbol{\sigma} \cdot \hat{\boldsymbol{n}}|_{\partial G} = \bar{\boldsymbol{\tau}}(\boldsymbol{x}), \ \boldsymbol{u}|_{t=0} = \bar{\boldsymbol{u}}(\boldsymbol{x}), \ \boldsymbol{v}|_{t=0} = \bar{\boldsymbol{v}}(\boldsymbol{x}), \tag{3}$$

respectively, where $\bar{\boldsymbol{\tau}}(\boldsymbol{x})$, $\bar{\boldsymbol{u}}(\boldsymbol{x})$ and $\bar{\boldsymbol{v}}(\boldsymbol{x})$ represent spatially varying distributions of traction, displacement and velocity, respectively. $\hat{\boldsymbol{n}}$ is the outward pointing normal vector from ∂G (the boundary of the domain).

Replacing the spatial and temporal derivatives by discrete pointwise approximations forms the backbone of finite-difference methods. In contrast, finite-element methods work with the weak, or variational, formulation of the governing equations (Brenner & Scott 2007; Quarteroni *et al.* 2010). Here a major advantage arises from approximating the wavefield by suitably chosen basis functions, which leads to a natural representation of the solution and its derivatives on the whole domain, that is not only on a discrete set of points, and it enables us to compute integrals over volumes and surfaces.

To obtain the weak form of eq. (1) we first multiply with a smooth function $\boldsymbol{w}$ (referred to as a test function), which gives for any $t \in (0, T]$:

$$\int_G \rho \boldsymbol{w} \cdot \partial_t^2 \boldsymbol{u} \, d^n\boldsymbol{x} = \int_G \boldsymbol{w} \cdot (\nabla \cdot \boldsymbol{\sigma}) \, d^n\boldsymbol{x} + \int_G \boldsymbol{w} \cdot \boldsymbol{f} \, d^n\boldsymbol{x}, \tag{4}$$

where the test function $\boldsymbol{w}$ does not depend on time t. Applying the divergence theorem to the first term on the right in eq. (4) gives

$$\int_G \rho \boldsymbol{w} \cdot \partial_t^2 \boldsymbol{u} \, d^n\boldsymbol{x} = \int_{\partial G} \boldsymbol{w} \cdot (\boldsymbol{\sigma} \cdot \hat{\boldsymbol{n}}) \, d^{n-1}\boldsymbol{x}$$
$$- \int_G \nabla \boldsymbol{w} : \boldsymbol{\sigma} \, d^n\boldsymbol{x} + \int_G \boldsymbol{w} \cdot \boldsymbol{f} \, d^n\boldsymbol{x}. \tag{5}$$

Note the explicit appearance of the boundary integral in eq. (5). This term vanishes if the free-surface boundary ($\boldsymbol{\sigma} \cdot \hat{\boldsymbol{n}}|_{\partial G} = \boldsymbol{0}$) condition

from (3) is inserted, and eq. (5) reduces to

$$\int_G \rho \boldsymbol{w} \cdot \partial_t^2 \boldsymbol{u} \, d^n\boldsymbol{x} = -\int_G \nabla \boldsymbol{w} : \boldsymbol{\sigma} \, d^n\boldsymbol{x} + \int_G \boldsymbol{w} \cdot \boldsymbol{f} \, d^n\boldsymbol{x}, \tag{6}$$

The free-surface condition is now implicitly included without any special effort. This inclusion is a major advantage over finite-difference methods, especially when modelling wave propagation through regions with strong topography, considering sea bottom bathymetry or dealing with applications in engineering where the domains may exhibit an arbitrary level of geometric complexity. Of course, we may also specify any other boundary condition we like, with the caveat that the integral over ∂G may become non-zero. This is the case, for example, along the interface between coupled fluid and solid domains, or any internal discontinuity where the normal stresses are continuous.

In seismology, it is often desired to solve the acoustic (scalar) version of the wave equation in addition to the elastic (vector) version. This is useful in modelling wave propagation through fluid or coupled fluid–solid systems (Komatitsch *et al.* 2000; Komatitsch & Tromp 2002b; Nissen-Meyer *et al.* 2008; Korta Martiartu *et al.* 2017), and also as a computationally efficient approximation to the elastic wave equation (Pratt *et al.* 1998; Prieux *et al.* 2013; Cance & Capdeville 2015). We can derive the weak form of the acoustic wave equation in a similar way to that demonstrated above. Starting from the strong form

$$\rho^{-1}c^{-2}\partial_t^2 u = \nabla \cdot (\rho^{-1}\nabla u) + f \quad \text{in } G \times [0, T], \tag{7}$$

where c is the speed of sound and u is a scalar displacement potential (Chaljub & Valette 2004), we multiply with the (scalar) test functions w, integrate, and apply the divergence theorem to obtain:

$$\int_G \rho^{-1}c^{-2}w \, \partial_t^2 u \, d^n\boldsymbol{x} = \int_{\partial G} \rho^{-1}w \, \nabla u \, d^{n-1}\boldsymbol{x}$$
$$- \int_G \rho^{-1}\nabla w \cdot \nabla u \, d^n\boldsymbol{x} + \int_G w \, f \, d^n\boldsymbol{x}. \tag{8}$$

At this stage, it is instructive to note the similarity between eqs (4) and (8). Each includes four distinct terms: a term involving a volume integral of density and acceleration (the *mass* term), a term involving a surface integral of the normal component of stresses (the *boundary* term), a term involving the volume integral of internal forces (the *stiffness* term), and a term involving a volume integral over external forces (the *forcing* term). All terms also involve the dot product with either the test functions $\boldsymbol{w}$ or their gradients $\nabla\boldsymbol{w}$, and act on either the wavefield $\boldsymbol{u}$ or its gradient $\nabla\boldsymbol{u}$. It is understood that in the case of the scalar wave equation, the vector quantities simply reduce to their scalar counterparts.

With these similarities in mind and using the notation $(\boldsymbol{a}, \boldsymbol{b})_G = \int_G \boldsymbol{a} \cdot \boldsymbol{b} \, d^n\boldsymbol{x}$ and $\langle \boldsymbol{a}, \boldsymbol{b} \rangle_{\partial G} = \int_{\partial G} \boldsymbol{a} \cdot \boldsymbol{b} \, d^{n-1}\boldsymbol{x}$, we can write eqs (5) and (8) in a more compact form. The weak forms now read: for all $t \in (0, T]$ and all $\boldsymbol{w} \in V^n$ (resp. $w \in V$), find $\boldsymbol{u}$ (resp. u) such that

$$\left(\rho \partial_t^2 \boldsymbol{u}, \boldsymbol{w}\right)_G - \left\langle (\boldsymbol{C} : \nabla\boldsymbol{u}) \cdot \hat{\boldsymbol{n}}, \boldsymbol{w}\right\rangle_{\partial G} + (\boldsymbol{C} : \nabla\boldsymbol{u}, \nabla\boldsymbol{w})_G$$
$$= (\boldsymbol{f}, \boldsymbol{w})_G, \tag{9}$$

or

$$\left(\rho^{-1}c^{-2}\partial_t^2 u, w\right)_G - \left\langle (\rho^{-1}\nabla u) \cdot \hat{\boldsymbol{n}}, w\right\rangle_{\partial G} + \left(\rho^{-1}\nabla u, \nabla w\right)_G$$
$$= (f, w)_G, \tag{10}$$

where we have used the elastic constitutive relationship given in eq. (2) and where V denotes a suitably chosen Galerkin space that will be defined in the next section. In this form, the similarities between the two equations are quite apparent. The only significant

difference comes in the way material coefficients are multiplied into each term, and the dimensions of the dynamic fields and test functions.

We can use these similarities to add one additional layer of abstraction. Here we will use the concept of a 'mathematical model' to describe a set of equations. Each model encompasses a specific term in eqs (9) and (10), and includes a description of the relevant material parameters, the choice of the test functions space, and the associated mathematical operations. We denote such a model by the symbol $\mathcal{I}$ and define it as

$$\mathcal{I}_I(\mathcal{P}), \text{ where } I \in \{A, E\}, \mathcal{P} \in \{M, B, S, F\}, \tag{11}$$

where A and E refer to the acoustic and elastic wave equation, and M, B, S, F refer to the mass, boundary, stiffness and forcing terms, respectively. In this notation, we can represent the acoustic or elastic wave equations with the simple expression

$$\mathcal{I}_I(M) - \mathcal{I}_I(B) + \mathcal{I}_I(S) = \mathcal{I}_I(F), \tag{12}$$

or even more compactly as

$$\sum_{\mathcal{P}\in\{M,B,S,F\}} \mathcal{I}_I(\mathcal{P}) = 0. \tag{13}$$

In essence, we have compressed all the information that does not change through time into $\mathcal{I}$.

Such a model is passed the time-varying field $\boldsymbol{u}$ as an argument to $\mathcal{P} = \mathcal{P}(\boldsymbol{u})$, on which it acts in much the same way as a function. As before, we allow $\boldsymbol{u}$ to reduce to its scalar equivalent in the case of the acoustic, or any other scalar, wave equation.

2.2 Galerkin approximation

In order to solve eq. (13) numerically we need to choose a proper finite-dimensional approximation $V_h \approx V$. For simplicity and efficiency we consider the Ritz–Galerkin approach, which is also commonly known as Galerkin method (Zienkiewicz *et al.* 2005). Here, the test functions $\boldsymbol{w}$ and the solution $\boldsymbol{u}$ are approximated by a set of functions $\boldsymbol{\phi}$ chosen from the same space. Let $\mathcal{B}$ be a basis of V_h and let N denote its size. We can form a discrete approximation of any continuous function g, which could represent a component of the displacement $\boldsymbol{u}$, or the stress $\boldsymbol{\sigma}$, by expanding this basis with coefficients F_j:

$$g(\boldsymbol{x}) \approx \sum_{j=1}^{N} F_j \boldsymbol{\phi}_j(\boldsymbol{x}). \tag{14}$$

The derivatives of g can then be approximated by the derivatives of $\boldsymbol{\phi}_j$:

$$\nabla g(\boldsymbol{x}) \approx \sum_{j=1}^{N} F_j \nabla \boldsymbol{\phi}_j(\boldsymbol{x}) \tag{15}$$

The accuracy of this approximation will be limited by how well V_h approximates the original function space V.

In addition to the accurate evaluation of these fields and their derivatives, we must also evaluate integrals like $(\boldsymbol{u}, \boldsymbol{w})_G$ and $\langle \nabla \boldsymbol{u} \cdot \hat{\boldsymbol{n}}, \boldsymbol{w} \rangle_{\partial G}$. To do this, we turn to numerical quadrature. Here, we combine a set of M integration weights W_i, with $g(\boldsymbol{x})$ and $\nabla g(\boldsymbol{x})$ evaluated on a set of integration points $\boldsymbol{x}_i$, resulting in

$$\int_G g(\boldsymbol{x}) \, d^n \boldsymbol{x} \approx \sum_{i=1}^{M} W_i^G \sum_{j=1}^{N} F_j \boldsymbol{\phi}_j(\boldsymbol{x}_i) \tag{16}$$

for volume integrals, and

$$\int_{\partial G} g(\boldsymbol{x}) \, d^{n-1} \boldsymbol{x} \approx \sum_{i=1}^{M} W_i^{\partial G} \sum_{j=1}^{N} F_j \boldsymbol{\phi}_j(\boldsymbol{x}_i) \tag{17}$$

for surface integrals. Note that the only difference between these two terms is the superscript on W. This allows us to write both n- and $(n-1)$-dimensional integrals in the same form, however in reality the weights $W^{\partial G}$ are only defined on ∂G.

To work the model $\mathcal{B}$ into the wave equation, we can expand the definition of $\mathcal{I}$ and insert any choice of the basis $\mathcal{B}$ as an additional argument into eq. (13), which encodes all operations introduced in this section

$$\sum_{\mathcal{P}\in\{M,B,S,F\}} \mathcal{I}_I(\mathcal{P}, \mathcal{B}) = \boldsymbol{0}. \tag{18}$$

Consider the powerful generalizations obtained when writing the wave equation in this manner. In addition to generalizing over the acoustic and elastic wave equation, we have not yet specified exactly *which* function space $\mathcal{B}$ describes. In fact, we are able to use *any* function space, as long as eqs (14)–(17) are well defined. Additionally, we have not yet specified the spatial dimension. Indeed, the same form holds for 1, 2 and 3 dimensions. All that is required is that the integrals in eqs (16) and (17) exist for a given dimension n. For example, in 2-D these equations refer to surface and line integrals, while in 3-D they refer to volume and surface integrals, respectively. Most importantly, we can separate all physical quantities (encoded in $\mathcal{P}$) from the finite-element discretization (parametrized by $\mathcal{B}$).

We can now analyse the steps needed to compute $\mathcal{I}_I(S, \mathcal{B})$ (the stiffness term). First, we use $\mathcal{B}$ to approximate the displacement field $\boldsymbol{u}$ with some finite-dimensional basis, for example polynomials, via eq. (14), resulting in a discretized approximation of $\boldsymbol{u}$. We then use additional functionality internal to $\mathcal{B}$ to compute the gradient of the discretized displacement via eq. (15). This gradient, equivalent to the elastic or acoustic strain, is now passed on to $\mathcal{I}_I$. For $I = A$ (the acoustic wave equation), this action involves multiplication by the inverse density ρ^{-1}, while for $I = E$ (the elastic wave equation) this involves a contraction with the elastic tensor $\boldsymbol{C}$. If we consider the strain and stress as being represented in Voigt notation [see Babuška & Cara (1991), for example], then the above operations simplify to a scalar-vector or matrix–vector product, respectively. The result of either operation is equivalent to the acoustic or elastic stresses. We then rely on $\mathcal{B}$ again to first compute the gradient of the test functions $\nabla \boldsymbol{w}$, and finally perform the numerical integration via the quadrature rule from eq. (16). This completes the computation of $\mathcal{I}_I(S, \mathcal{B})$.

Eq. (18) describes the semi-discrete system of the weak form of the wave equation and does not yet include the discretization in time. Again, this can be seen as another layer independent to the physics and the spatial discretization. Commonly used explicit time marching schemes for are, for instance, the second-order Newmark central-differences scheme (Hughes 2000; Peter *et al.* 2011), second-order leap-frog (Ferroni *et al.* 2017), fourth-order symplectic schemes (Nissen-Meyer *et al.* 2008) or high-order Runge–Kutta methods (Wilcox *et al.* 2010).

2.3 Finite-elements

In seismology, we are often concerned with accurately representing different discontinuities within the Earth. These can include several phase-transition boundaries within the mantle, the core–mantle

boundary, or, on a much smaller scale, the ocean bottom. Across these discontinuities the model parameters and strain field are not differentiable, and for this reason a smooth polynomial basis over the entire domain G is usually not appropriate. To mitigate this problem we turn to the finite-element method (FEM), where the domain G is subdivided into a set of non-overlapping elements, each enclosing the volume G_e. The basis $\mathcal{B}$ then consists of continuous functions on G which are restricted to piecewise polynomials defined on the individual elements. Within each element, we approximate functions and their derivatives as in eqs (14) and (15).

While using the FEM, for reasons of efficiency, we can take advantage of pre-tabulated integration weights and basis function derivatives defined on a reference element, which are often defined in an element-specific reference coordinate system $\boldsymbol{\xi}$. To transform these integrals and derivatives to the physical coordinates $\boldsymbol{x}$, we require an element-specific Jacobian $\partial\boldsymbol{x}/\partial\boldsymbol{\xi}$, which defines the mapping from reference to physical coordinates. With this transform, eqs (14)–(17) become

$$g(\boldsymbol{x}) \approx \sum_{j=1}^{N} F_j \boldsymbol{\phi}_j(\boldsymbol{\xi}(\boldsymbol{x})), \tag{19}$$

$$\nabla g(\boldsymbol{x}) \approx \sum_{j=1}^{N} F_j \frac{\partial}{\partial\boldsymbol{\xi}} \boldsymbol{\phi}_j(\boldsymbol{\xi}(\boldsymbol{x})) \frac{\partial\boldsymbol{\xi}}{\partial\boldsymbol{x}} \tag{20}$$

$$\int_{G_e} g(\boldsymbol{x}) \, d^n\boldsymbol{x} \approx \sum_{i=1}^{M} W_i^{G_e} \sum_{j=1}^{N} F_j \boldsymbol{\phi}_j(\boldsymbol{\xi}(\boldsymbol{x}_i)) \left|\frac{\partial\boldsymbol{x}}{\partial\boldsymbol{\xi}}\right|_{\boldsymbol{x}=\boldsymbol{x}_i}^{n}, \tag{21}$$

and

$$\int_{\partial G_e} g(\boldsymbol{x}) \, d^{n-1}\boldsymbol{x} \approx \sum_{i=1}^{M} W_i^{\partial G_e} \sum_{j=1}^{N} F_j \boldsymbol{\phi}_j(\boldsymbol{\xi}(\boldsymbol{x}_i)) \left|\frac{\partial\boldsymbol{x}}{\partial\boldsymbol{\xi}}\right|_{\boldsymbol{x}=\boldsymbol{x}_i}^{n-1}, \tag{22}$$

respectively. To assemble the integrals in eqs (21) and (22) over the whole domain G, one simply needs to sum contributions from each element. With this in mind, the only difference to eqs (14)–(17) is the dependence on the reference coordinate system $\boldsymbol{\xi}$, along with the determinant of the Jacobian matrix and its inverse, which describes the local mapping from reference to physical coordinates in an n-dimensional manifold. To calculate these additional quantities, all that is required is the definition of the mapping from the reference element to the physical domain. Let us now define $\mathcal{E}$ as a model containing this data, along with instructions on how to compute the Jacobian for a given element type. Then, in the finite-element discretization of the weak form, we can rewrite $\mathcal{B}$ as $\mathcal{B}(\mathcal{E})$, and thus eq. (13) becomes

$$\sum_{\mathcal{P}\in\{M,B,S,F\}} \mathcal{I}_I(\mathcal{P}, \mathcal{B}(\mathcal{E})) = \boldsymbol{0}. \tag{23}$$

The additional layer introduced by $\mathcal{E}$ allows us to specify the reference element type independently of the basis $\mathcal{B}$. For instance, we can use linear or quadratic functions to define the geometry of the reference element. Of course, there are some restrictions implied by the dimensionality of the problem and the details of $\mathcal{B}$ but with these restrictions eq. (23) is valid whether the domain is partitioned into either simplex- or hypercube-type meshes, or something else more exotic. The choice of these functionals depends on the specific problem at hand.

For most of the results presented in the next sections, we follow the classical spectral-element discretization commonly used in seismology, and choose the tensorized GLL basis as our discretization

model (Komatitsch & Tromp 2002b; Fichtner *et al.* 2009; Cupillard *et al.* 2012). This also restricts our choice of finite-elements to those of a hypercubical character. It is important to note that this choice of basis prevents us from integrating the discrete mass term with full accuracy, as the GLL quadrature rule is only exact for polynomials up to order $2n - 1$. As we use the same polynomials of order n to represent both the dynamic field variables and the test functions, the mass term includes polynomials of degree $2n$. Given a field with a fixed spatial complexity (e.g. a band-limited wavefield) this inaccuracy can always be mitigated for a given domain and model through both mesh refinement and/or increasing the order of the polynomial approximation. In Section 3.3, we show examples validating the accuracy of the spectral-element method for select media with constant material parameters through benchmark results comparing numerically generated and semi-analytic solutions. Once material parameters are allowed to vary within each element, either linearly or otherwise, the order of the corresponding polynomial approximations increase in turn. While this suggests that the accuracy of the simulated scalar displacement potential given in eq. (10) is degraded to a higher degree than the elastic wave eq. (9) due to the product of the inverses of material coefficients, this can be remedied with a change of parameterization. For instance, by parameterizing eq. (10) with compressibility ($\rho^{-1} \cdot c^{-2}$) and inverse density, the accuracy of the discrete solutions to (10) and (9) are identical. A thorough study of the convergence properties of spectral-element methods is beyond the scope of this paper. A recent and comprehensive study can be found in Ferroni *et al.* (2017).

2.4 Complex rheologies

While the abstractions introduced so far allow us to construct a general finite-element discretization of the seismic wave equation, their utility extends to the mathematical models describing wave propagation physics as well. Consider a move beyond linear elasticity into viscoelastic and weakly non-linear regimes. Many of these effects these rheologies have on the wavefield can be expressed as modifications to the elastic or acoustic stresses, which are contained within the term $\mathcal{I}_I(S, \mathcal{B}(\mathcal{E}))$. Consider the viscoelastic wave equation. In this case, the stiffness term in eq. (9) depends on the entire strain history

$$(\boldsymbol{C} : \nabla\boldsymbol{u}, \nabla\boldsymbol{w}) \rightarrow \left(\int_{-\infty}^{t} \partial_t \boldsymbol{C}(t - \tau) : \nabla\boldsymbol{u}(\tau) \, d\tau, \nabla\boldsymbol{w}\right). \tag{24}$$

It is common to approximate this time-dependence by a discrete Laplace transform (or discrete relaxation spectrum) that can be equivalently represented as generalized Zener or Maxwell bodies (Emmerich & Korn 1987; Robertsson *et al.* 1994; Moczo & Kristek 2005; van Driel & Nissen-Meyer 2014). Using this discrete formulation, we can rewrite the stiffness term in (24) as

$$\left(\boldsymbol{C}_U : \nabla\boldsymbol{u}(t) - \sum_{j=1}^{n_m} \boldsymbol{M}^j(t), \nabla\boldsymbol{w}\right), \tag{25}$$

where the $\boldsymbol{M}^j(t)$ are memory variables which encode the strain history of the Maxwell bodies. $\boldsymbol{C}_U$ refers to the 'unrelaxed elastic tensor', the coefficients of which represent the instantaneous elastic response to a strain impulse.

Note the following similarities between eq. (25) and the purely elastic stiffness as defined in eq. (9), and written in model notation as $\mathcal{I}_E(S, \mathcal{B}(\mathcal{E}))$. The operations provided by $\mathcal{B}(\mathcal{E})$ are identical. The gradient of the displacement $\boldsymbol{u}$ and test functions $\boldsymbol{w}$ maintain the same form. On the element level $\mathcal{E}$, we require the same mapping

to physical coordinates which have already been provided. Further-more, we retain the contraction of a fourth-order tensor with the displacement gradient $\boldsymbol{u}$, although the elastic tensor is modified to take on an un-relaxed state through the subtraction of the memory variables. We can encode these additional instructions in a new model $\mathcal{A}(\cdot)$:

$$\sum_{\mathcal{P} \in M, B, S, F} \mathcal{I}_E(\mathcal{A}(\mathcal{P}), \mathcal{B}(\mathcal{E})) = \boldsymbol{0}. \tag{26}$$

Assuming a free-surface condition on the boundary of G, the application of $\mathcal{A}$ only modifies the term $\mathcal{I}_E(S, \mathcal{B}(\mathcal{E}))$. Thus we can consider it as an identity operator over the other models and eq. (26) simplifies to

$$\sum_{\mathcal{P} \in M, F} \mathcal{I}_E(\mathcal{P}, \mathcal{B}(\mathcal{E})) + \mathcal{I}_E(\mathcal{A}(S), \mathcal{B}(\mathcal{E})) = \boldsymbol{0}. \tag{27}$$

As we did with $\mathcal{B}$, we now explore the steps required to compute $\mathcal{I}_E(\mathcal{A}(S), \mathcal{B}(\mathcal{E}))$. First, we use the conventional stress term S to compute $\boldsymbol{C}_U : \nabla \boldsymbol{u}(t)$, the first operation in eq. (25). The polynomial approximation provided by $\mathcal{B}$ is required, which in turn requires the Jacobian matrix provided by $\mathcal{E}$. Once this term is computed, we use the instructions on the $\mathcal{A}$ layer to subtract the sum of the j memory variables $\boldsymbol{M}^j(t)$. Once this is complete, we return to the conventional stress computation S and its arguments to compute the contraction with $\nabla \boldsymbol{w}$ and the resulting integral. Finally, we use the instructions on $\mathcal{A}$ to update the differential equation controlling the time evolution of the memory variables.

Note that we have yet to specify any anisotropic symmetries (handled by the functional $\mathcal{P}$), the specifics of the polynomial interpolation (handled by the functional $\mathcal{B}$), or the element type (handled by the functional $\mathcal{E}$). Nevertheless, we have a compact representation of the resulting weak form equation which can be used with any valid combination of $\mathcal{P}$, $\mathcal{B}$, and $\mathcal{E}$. This form is valid for *any* modification to elastic or acoustic stresses. Such is also the case when applying gravitational corrections in the Cowling approximation (Cowling 1941; Chaljub *et al.* 2007), or simulating wave propagation in weakly non-linear/hyperelastic materials (Rivière *et al.* 2013). We simply require some modifier $\mathcal{F}$ which takes the physics defined in the wave equation as an argument, and implements modifications for all those terms $\mathcal{P}$ for which $\mathcal{F}(\mathcal{P})$ is not an identity transform.

Another illustrative example involves the coupling between the acoustic elastic rheologies, relevant when simulating wave propagation in combined fluid–solid domains. In this case, all elements along the coupling boundaries must compute additional quantities to ensure the continuity of traction between the two domains (Komatitsch & Tromp 2002a; Chaljub & Valette 2004; Nissen-Meyer *et al.* 2008). We can bring the (time-independent) coupling effects into the model notation by defining $\mathcal{C}_{FS}$ (fluid to solid) and $\mathcal{C}_{SF}$ (solid to fluid) and applying them to the coupled wave equation in a manner similar to eq. (26). Since these only act on the boundary terms, the application of these functionals to the wave equation results in identity transforms for all other terms. Inserting these identity transforms where appropriate, we obtain the simplified result

$$\sum_{\substack{I \in \{A, E\} \\ \mathcal{P} \in \{M, S, F\}}} \mathcal{I}_I(\mathcal{P}, \mathcal{B}(\mathcal{E})) + \mathcal{I}_E(\mathcal{C}_{FS}(B), \mathcal{B}(\mathcal{E})) + \mathcal{I}_A(\mathcal{C}_{SF}(B), \mathcal{B}(\mathcal{E}))$$

$$= \boldsymbol{0}. \tag{28}$$

We can include attenuation in the solid regions by including $\mathcal{A}$ in eq. (28):

$$\sum_{\substack{I \in \{A, E\} \\ \mathcal{P} \in \{M, F\}}} \mathcal{I}_I(\mathcal{P}, \mathcal{B}(\mathcal{E})) + \mathcal{I}_A(S, \mathcal{B}(\mathcal{E})) + \mathcal{I}_E(\mathcal{A}(S), \mathcal{B}(\mathcal{E}))$$

$$+ \mathcal{I}_E(\mathcal{C}_{FS}(\mathcal{A}(B)), \mathcal{B}(\mathcal{E})) + \mathcal{I}_A(\mathcal{C}_{SF}(B), \mathcal{B}(\mathcal{E}))$$

$$= \boldsymbol{0}. \tag{29}$$

and follow a similar process for including corrections for self-gravitation as well.

A complete numerical model of wave propagation can be built up from a set of functionally orthogonal models. In an abstract sense, the utility of these models seems be limited to generating concise representations of the discretized wave equation, which is generalized over rheology. In the next section we will explore their practical utility as tools for writing maintainable and efficient numerical modelling software.

3 IMPLEMENTATION IN MODELLING SOFTWARE

The above abstractions have allowed us to split the physics, spatial discretization and finite-element shape mappings into three classes of logically orthogonal components. The components within each of these classes can, with some restrictions (e.g. those induced by the spatial dimension), be interchanged independently from changes in any other class. To be of practical use, these abstractions must be implemented in computer code, and it must be ensured that their use does not negatively affect performance. To achieve this goal, we turn to the C++ programming language, and make extensive use of the 'template mixin' paradigm.

3.1 Models as template mixins

As a simple introduction, consider a concrete instantiation of the model $\mathcal{B}(\mathcal{E})$. We can write a one-to-one correspondence between abstract model notation and C++ template mixin notation as

$$\mathcal{B}(\mathcal{E}) \mapsto \texttt{TensorGll}\langle\texttt{QuadP1}\rangle \tag{30}$$

Here, following the classical spectral-element discretization commonly used in seismology, we have chosen the tensorized GLL basis as our discretization model $\mathcal{B}$, and restricted the support of this discretization to a 2-D 4-node quadrilateral element with a bilinear (P1) shape mapping, which takes the place of $\mathcal{E}$. In practise, each component of the model (i.e. `TensorGll`) is a separate C++ class. Alternatively, we could use the same GLL basis as $\mathcal{B}$, but change the support to a 3-D 8-node hexahedral element, and write this in mixin notation as

$$\mathcal{B}(\mathcal{E}) \mapsto \texttt{TensorGll}\langle\texttt{HexP1}\rangle. \tag{31}$$

Here we get a sense for the origin of the term 'mixin'. Given a collection of possible discretization schemes, and a collection of elemental restrictions, we 'mix-in' the two collections together to create a concrete instance of the abstract model $\mathcal{B}(\mathcal{E})$. The concrete instantiations of all the abstract models can be referred to in this way. As was the case in Section 2, the efficiency of this notation grows and becomes more obvious as the rheologies and discretizations become more complex.

Consider a toy implementation of eq. (20), which computes the gradient of some discrete function f, given in Listing 1.

An exploration of this Listing gives insight into how these abstract models are used in practice. First, we note that this function is a component of the 'Basis' model $\mathcal{B}$. In Listing 1 this is specified by prefixing the name of the model the function belongs to (Basis) to the function name itself (ComputeGradient). At this stage, the element $\mathcal{E}$ is still abstract and waiting to be 'mixed-in', which is evidenced by the template <typename Element> preceding the function definition. Internal to the function, we first allocate a 2-D array to hold the gradient of F, which is of the size (NumDof, NumDim), or the number of degrees of freedom on this element, and the number of dimensions of the problem, respectively. Here we already take advantage of the abstract nature of the model notation: NumDof is provided by the current discretization model, but NumDim is currently unknown, and only provided once the complete model is specified. Now, we loop through each dimension, and fill up Gradient by following eq. (20) more or less verbatim. We multiply and sum F at the current integration point by the gradient in reference coordinates of the test functions at all the integration points. To transform this into physical coordinates, we rely on $\mathcal{E}$ to provide us with the shape mapping DepsDx. Although the true implementation of such functions can be made more sophisticated than the example in Listing 1, the concept here is what is important: through the C++ template mixin paradigm we are able to recognize the benefits of the abstract and compact model notation introduced in Section 2.

The remaining models can be implemented in the same way. Consider the case of the acoustic, or isotropic/anisotropic elastic stiffness term $\mathcal{I}_I(S, \mathcal{B}(\mathcal{E}))$, both implemented in Listing 2. Here we are mixing *two* models: the discretization (Basis), and a particular model of computing stresses (Physics).

If we unpack the information compressed in these models, we see that Listing 2 is written in a similar fashion as the continuous weak form given in eqs (9) and (10). Indeed, such a function remains unchanged across dimensions, discretization models, and element types (of course, as long as each of these are consistent with each other). An example of some concrete instantiations of $\mathcal{I}_I(S, \mathcal{B}(\mathcal{E}))$ are

$\mathcal{I}_I(S, \mathcal{B}(\mathcal{E})) \mapsto$ Acoustic⟨TensorGll⟨QuadP1⟩⟩,

$\mathcal{I}_I(S, \mathcal{B}(\mathcal{E})) \mapsto$ Elastic⟨TensorGll⟨HexP1⟩⟩, (32)

$\mathcal{I}_I(S, \mathcal{B}(\mathcal{E})) \mapsto$ Elastic⟨EnrichedLagrange⟨TetP1⟩⟩,

among many others, where EnrichedLagrange refers to a discretization model based on mixed-order Lagrange polynomials suitable for high-order mass lumping on simplicial elements (Chin-Joe-Kong *et al.* 1999; Cohen *et al.* 2001; Zhebel *et al.* 2014; Mulder & Shamasundar 2016).

The implementation of more complex rheologies follows in a similar manner. Taking an implementation of viscoelasticity using memory variables, as described in Section 2.4, we can write a simple implementation of $\mathcal{I}_E(\mathcal{A}(S), \mathcal{B}(\mathcal{E}))$, or eq. (27), as expressed in Listing 3. The syntax we are using here can again be related to the notation introduced in section 2. The template parameter specifies the discretized wave equation $\mathcal{I}_E(S, \mathcal{B}(\mathcal{E}))$, and the resulting model takes the wavefield $\boldsymbol{u}$ as an argument. The Attenuation class performs the job of $\mathcal{A}$, specifically time-stepping the state of the memory variables, and modifying the constitutive relationship accordingly. As in Listing 2, Listing 3 is valid for all consistent dimensions, discretization models and element types, and is also valid regardless of the exact form

of the elastic stiffness computation (i.e. isotropic or anisotropic). XXXX

Here we note that all these functions could be made even simpler by explicitly building the finite-element stiffness matrix, and just performing one matrix-vector product to compute the internal forces. Our reasons for not taking this approach are two-fold. First, there are significant efficiency gains in this 'on-the-fly' computation of the stiffness matrix for all wave equation types, although these gains are restricted to hypercube-type elements with a tensorized basis (Deville *et al.* 2002). Secondly, and more generally, this approach allows us to save costs in the computation of complex constitutive relationships which modify the stress and / or strain. Using the explicit stiffness matrix would require the recomputation of these terms.

In general, the implementation of the discretized wave equation with C++ template mixins allows an almost one-to-one correspondence with the hierarchical model notation introduced in section 2. This can be expressed with an analogy to building blocks, as illustrated in Fig. 1. Here we use the term 'Decorator' to represent any modification to the linear elastic rheology that can be written in a form similar to eq. (26).

3.2 Constructing mixins in complex domains

The above approach lends itself well to complex coupled domains, as each individual element in the domain can just be constructed from an arbitrary model/mixin. Consider, for example, the 4 element domain illustrated in Fig. 2. Each element is alternatively solid or fluid, the top domain boundary is of type Dirichlet, and the other three domain boundaries are meant to absorb energy. In mixin notation, we can write each element as follows:

Element 0 ↦ AbsorbingBoundary⟨DirichletBoundary
⟨Acoustic⟨TensorGll⟨Quadrilateral⟨QuadP1⟩⟩⟩⟩⟩

Element 1 ↦ AbsorbingBoundary⟨Elastic⟨TensorGll
⟨Quadrilateral⟨QuadP1⟩⟩⟩⟩

Element 2 ↦ AbsorbingBoundary⟨DirichletBoundary
⟨Elastic⟨TensorGll⟨Quadrilateral⟨QuadP1⟩⟩⟩⟩⟩ (33)

Element 3 ↦ AbsorbingBoundary⟨Acoustic⟨TensorGll
⟨Quadrilateral⟨QuadP1⟩⟩⟩⟩

With the equivalence of the mixin notation and the model notation, this explicit characterization of each element ensures that the proper equations will be solved on each element. The question remains how to construct the mixins automatically for domains of arbitrary complexity. Domains with simple fluid-solid boundaries are found in many seismological applications. Coupling between the mantle and outer core, and oceanic exploration surveys, are some common examples. In these cases it is relatively simple to mark specific facets (a generalization of edges in 2-D, and faces in 3-D) in the mesh as coupling boundaries. However, this is disadvantageous as such boundaries need to be re-computed if the mesh discretization ever changes. More difficulties are encountered in examples where each individual element may vary between a fluid and solid. This may be the case, for example, in a saturated porous medium. The domain illustrated in Fig. 2(a) gives a simple example of such a pathological case. To handle such complex domains we build on the functionality provided by DMPLEX, which is a library for handling unstructured meshes contained within PETSC (Knepley & Karpeev 2009; Lange *et al.* 2016). DMPLEX represents a computational mesh as a directed acrylic graph (DAG), and allows

```
template <typename Element>
Vector Basis::ComputeGradient(Vector f)
{
  const int dim = Element::NumDim();
  const int numDof = Basis::NumDof(Element::NumDim());
  Matrix grad(numDof, dim);
  for (int d = 0; d < dim; ++d) {
    for (int i = 0; i < numDof; ++i) {
      grad(i, d) += f(i) * Basis::RefGrad(i, d) *
        Element::DepsDx(Basis::IntCrd(i), d);
    }
  }
  return grad;
}
```

Listing 1: A possible implementation of a function belonging to $\mathcal{B}(\mathcal{E})$

```
template <typename Basis, typename Physics>
Vector WaveEquation<Basis, Physics>::ComputeStiff(Vector u)
{
  Vector strain = Basis::ComputeSymmetricGradient(u);
  Vector stress = Physics::ComputeStress(strain);
  return Basis::ApplyGradTestAndIntegrate(stress);
}
```

Listing 2: Abstract implementation of the acoustic and elastic stiffness terms $\mathcal{I}_I(S, \mathcal{B}(\mathcal{E}))$

```
template <typename WaveEquation>
Vector Attenuation<WaveEquation>::ComputeStiff(Vector u)
{
  Vector strain = WaveEquation::ComputeSymmetricStrain(u);
  Vector stress = WaveEquation::ComputeStress(strain) -
    Attenuation::MemoryVariables;
  return WaveEquation::ApplyGradTestAndIntegrate(stress);
}
```

Listing 3: Simple implementation of $\mathcal{I}_E(\mathcal{P}(S), \mathcal{B}(\mathcal{E}))$

us to trivially construct complete mixins for elements in a domain of arbitrary geometric and physical complexity.

The mesh outlined in Fig. 2(a) contains five distinct physical components. On the elemental level, we consider elements that are responsible for simulating either the acoustic (blue) or elastic (brown) wave equation. In the fluid elements we consider a formulation of the acoustic wave equation based on a scalar displacement potential (eq. 8) in order to maintain an explicit time stepping scheme (Chaljub & Valette 2004). Along the left, bottom, and right edges we consider absorbing style boundary conditions (green) (Clayton & Engquist 1977; Komatitsch *et al.* 2000). Finally, along the top boundary we consider a homogeneous Dirichlet-type rigid boundary (red). The right side of Fig. 2(a) also demonstrates another way to look at the mesh, here focusing on the topological entities. Following the same order as in the previous paragraph, we proceed from entities of dimension two (faces F_N), to entities of

dimension one (edges E_N), to those of dimension zero (vertices V_N). As in Fig. 2(a), the mesh entities are coloured by the corresponding physics they introduce. Here we can consider the acoustic and elastic terms as forming the *element* physics, with the boundary terms serving to *mix-in* with the element physics. And additional complication introduced by the Dirichlet boundaries. Since these operate in a point-wise fashion, they must also be associated with the vertices of each edge they decorate, in contrast to the absorbing boundary terms which only consider surface integrals over an element facet.

Of course, there are several mixins which are absent from Fig. 2(a). These are the coupling mixins which we expect along $E3$, $E5$, $E6$ and $E8$. As well, since each model in eq. (33) is explicitly constructed once per element, it is essential that all mixins are known at the element (F_N) level. Fig. 2(b) recasts the topology outlined in Fig. 2(a) as a Hasse diagram, functionality provided by
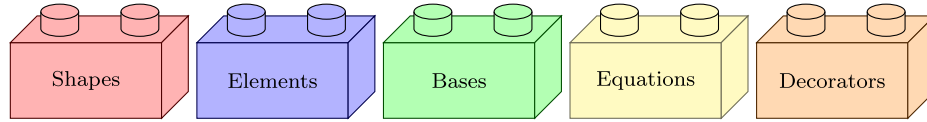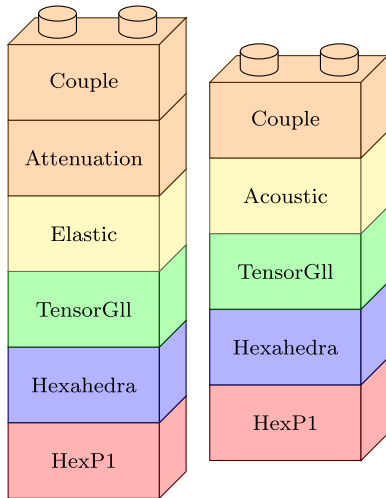
(a) Fundamental mathematical model / mixin categories
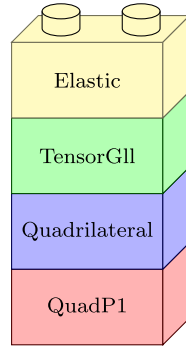
$$\sum_{\mathcal{P} \in \{M,B,S,F\}} \mathcal{I}_E(\mathcal{P}, \mathcal{B}(\mathcal{E})) = \mathbf{0}.$$

Elastic⟨TensorGll⟨Quadrilateral⟨QuadP1⟩⟩⟩

(b) Equivalent ways to express the 2D elastic wave equation (graphically via building blocks, in the model notation introduced in Section 2, and in the template mixin notation introduced in Section 3) The colors correspond to the fundamental categories introduced in (a).

$$\sum_{\substack{I \in \{A,E\} \\ j \in \{M,F\}}} \mathcal{I}_I(\mathcal{P}, \mathcal{B}(\mathcal{E})) + \mathcal{I}_A(S, \mathcal{B}(\mathcal{E})) + \mathcal{I}_E(\mathcal{A}(S), \mathcal{B}(\mathcal{E})) +$$
$$\mathcal{I}_E(\mathcal{C}_{FS}(\mathcal{A}(B)), \mathcal{B}(\mathcal{E})) + \mathcal{I}_A(\mathcal{C}_{SF}(B), \mathcal{B}(\mathcal{E})) = \mathbf{0}.$$

ToAcoustic⟨Visco⟨Elastic⟨TensorGll⟨Hexahedra⟨HexP1⟩⟩⟩⟩⟩
ToElastic⟨Acoustic⟨TensorGll⟨Hexahedra⟨HexP1⟩⟩⟩⟩

(c) Equivalent ways to express the 3-D coupled elastic / acoustic wave equation, with attenuation in the elastic regions (graphically via building blocks, in the model notation introduced in Section 2, and in the template mixin notation introduced in Section 3)

Figure 1. Visualization of the abstractions as building blocks. Constructing a complete mathematical model or mixing involves at least choosing a shape mapping, an element type, a set of basis functions, and a fundamental wave equation (a). These individual components can be mixed together in many ways. Some constructions equivalent in both model and mixin notation are given and visualized in (b) and (c).

DMPLEX. Here, each level, or 'stratum' in the diagram represents a mesh entity of increasing dimension, with the labels on each entity retained. The goal now becomes the efficient communication of all relevant mixins to the base of the Hasse diagram.

We can perform this communication in two stages. First, we traverse the upward and downward closure for each edge (Fig. 2c), and investigate the labels residing on each connected entity. As an example, for $E0$ the upward closure visits $V0$ and $V1$ while the downward closure visits $F0$. Any labels on these entities are then added to the set of labels on each edge. Once these sets are constructed, we can simply traverse one level in the upward closure of each element (F_N), and add all labels associated with the connected edges to the set of elemental physics (Fig. 2d). At this stage we have

all the information necessary to construct the complete model for each element.

3.3 Examples

Using the Hasse Diagrams is an attractive way of determining which decorators to add to the element physics as it is local to each element, but also agnostic to both element type and dimension. For simplicial elements, the number of edges and vertices associated with each element changes, but the Hasse diagram can be drawn in a similar way. As well, in three dimensions, another stratum is added to the bottom of the diagram indicating element volumes, but the discussions about closure traversal hold. Taken together,
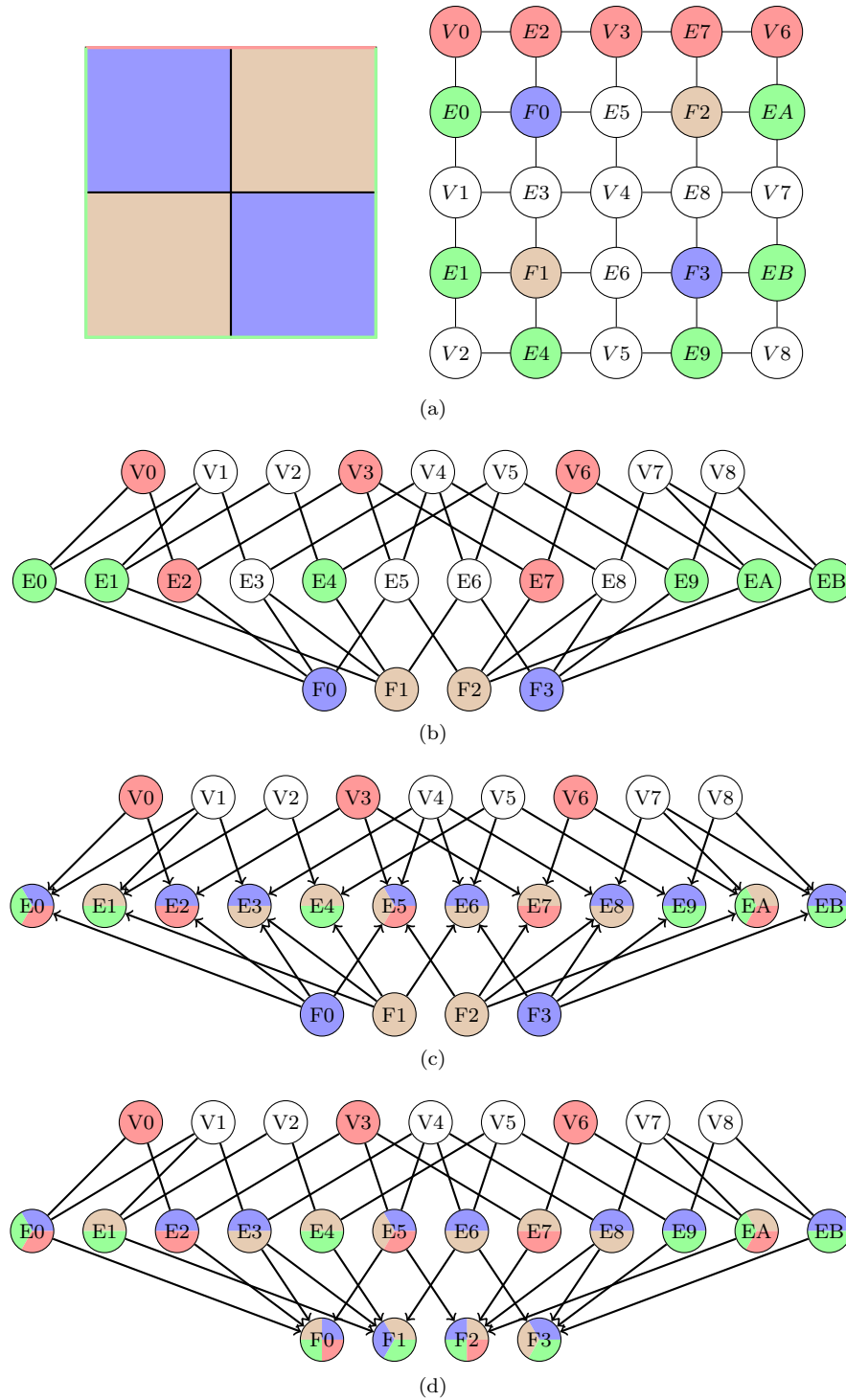
Figure 2. Illustration of how decorator chains are constructed in complex domains. (a, left) Chequerboard mesh with boundaries, represented as a mesh. The elements solve either the acoustic (blue) or elastic (brown) wave equation. A Dirichlet type boundary is placed on the top edge (red), with absorbing boundaries on the other 3 edges (green) The Dirichlet boundaries act pointwise, and must therefore be associated with both edges and vertices. The absorbing boundaries act on the wavefield through a surface integral, and are therefore only defined for edges. (a, right) The same mesh represented by means of topological entities including faces *F* (2-D elements), edges *E*, and vertices *V*, with each entity colored by its associated label. (b) A recasting of the mesh as a Hasse diagram, retaining the connectivity and labels outlined in (a). (c) Description of cycle 1 of the label communication: the upwards and downwards closure of each edge is queried, and any additional labels are added to the set of labels of each edge. (d) Description of the second cycle, where the set of labels on each edge is added to the set on each element. We now have the information necessary to combine the relevant mixins, and construct the full wave propagation model on each element.
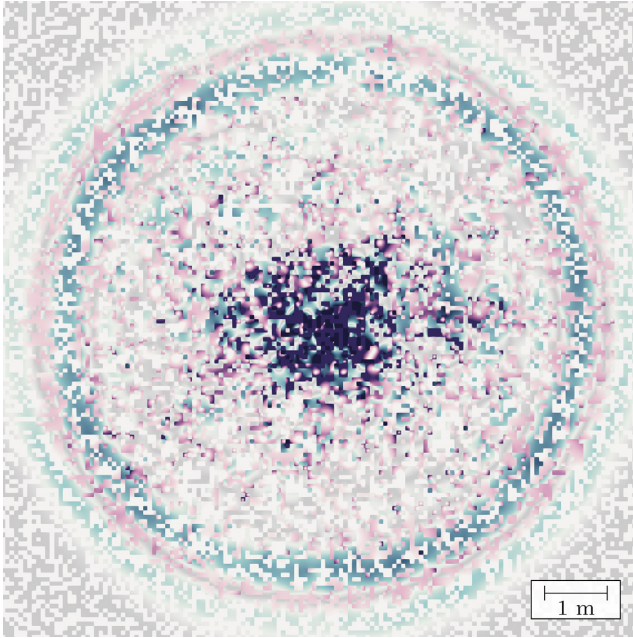
Figure 3. Snapshot of wave propagation in a complex coupled domain where each element is randomly marked as either fluid ($V_p = 1500$ m s^{-1}, $\rho = 1200$ kg m^{-3}) or solid ($V_p = 1500$ m s^{-1}, $V_s = 1000$ m s^{-1}, $\rho = 1000$ kg m^{-3}). The source was a Ricker wavelet with a centre frequency of 1 kHz. Such a domain may be used to represent a solid medium saturated with fluid. The normalized amplitude of the acoustic waves are visualized with blue hues, and the normalized amplitude of the elastic waves are visualized with purple hues. The individual solid or fluid elements can be differentiated by their grey or white colour, respectively. Internal coupling boundaries are automatically detected at runtime via the methods outlined in section 3.2.

these factors allow us to flexibly and efficiently discretize complex domains.

An example of this flexibility can be seen when considering a medium in which each element randomly simulates either the acoustic or elastic wave equation, with the separate solutions coupled along element edges. This may, for example, represent a type of fluid-filled porous medium, but it is more appropriate to consider this example as a complex end-member in the class of coupled simulations through unstructured domains. A snapshot of wave propagation through one realization of such a domain can be seen in Fig. 3. Note that here two distinct equations are being solved: the isotropic elastic wave equation, and an acoustic wave equation based on the scalar displacement potential, the weak forms of which are given in eqs (4) and (8) respectively. On the model generation side, all that was required was the labelling of individual elements as either acoustic or elastic. The coupling boundaries were found automatically at runtime by traversing the Hasse Diagram provided by DMPLEX as outlined in Section 3.2.

The benchmarks outlined in Fig. 4 present more examples of the benefits of both the model/mixin design and the generality of the Hasse diagrams, in this case through the discretization of the domain with four distinct element types. Simulated is coupled acoustic/elastic wave propagation in 2-D and 3-D on quadrilateral, triangular, hexahedral, and tetrahedral meshes. The 2-D or 3-D domain is identical in all cases, as are all models/mixins above the level specifying the polynomial basis. The functions computing the propagation in the acoustic and elastic subdomains, as well as the functions computing the coupling terms, are identical. The Hasse

diagram provided by DMPLEX was used to automatically determine the coupling edges. The only differences are at the bottom of the mixin chain with, for example, ⟨`TensorGll` ⟨`QuadP1`/`HexP1`⟩⟩ computing the spatial operations for the hypercubic elements, and ⟨`EnrichedLagrange` ⟨`Tri1`/`TetP1`⟩⟩ computing the spatial operations for the simplical elements. A detailed exploration of the associated convergence properties is beyond the scope of this paper.

4 DISCUSSION

The concepts outlined in Sections 2 and 3 have been implemented within the SALVUS software package. The next paragraphs will discuss several examples from this implementation and cover issues such as performance and scaling.

4.1 Performance and scaling

Abstractions in software often negatively affect performance. This point is especially relevant when considering full-waveform inversion, which has its place amongst the largest PDE-constrained optimization problems (Afanasiev *et al.* 2016; Bozdag *et al.* 2016). While the decomposition of the wave equation into models/mixins allows for a very concise representation of increasingly complex terms, in reality this approach is practically useless unless the resulting software is performance competitive. Considering that a realistic problem may include millions of elements, tens of thousands of time-steps, thousands of sources, and hundreds of iterations, it is likely that some of the functions described in the above Listings will be called 10^{19} times over the course of an inversion.

Rather than being detrimental, we believe that the software abstractions outlined above in fact allow for improved performance. One of the reasons is technical, and has to do with the implementation via C++ templates. During the code compilation stage, for each complete and meaningful wave propagation model a separate copy of the entire mixin chain is generated, meaning that there is *no* runtime cost associated with determining the physics being simulated on each individual element. On the contrary, since the compiler is able to see all code paths during the compilation stage, most of the function calls are actually 'inlined' – this is these instances, the overhead associated with the function call itself in completely eliminated. The inlining of all the individual components in Listing 3 allows the viscoelastic stiffness to be calculated without deferring to any other function. This can be contrasted with more classical object-oriented approaches involving virtual function lookups, where each function call would trigger a Virtual Method Table lookup, stalling the process until the appropriate delegate was found. This process would need to be repeated for each layer of the mixin chain, which can grow to be quite complex. Another reason for the efficiency of this design stems from the power of the abstractions themselves. Since each individual model is simple, care can be taken to ensure that each component is fully optimized. For example, the functions computing the wavefield gradient for each element type in eqs (9) and (10) are completely independent from the rest of the code. Since these functions are called from any combination of mixins, speedups in these functions have a wide-ranging impact.

To demonstrate that the abstractions minimally impact the runtime efficiency, we performed a simple single-thread benchmark. The test machine was powered by an Intel® Core™ i7-6850K (Broadwell) processor with a base operating frequency of 3.60 Hz.
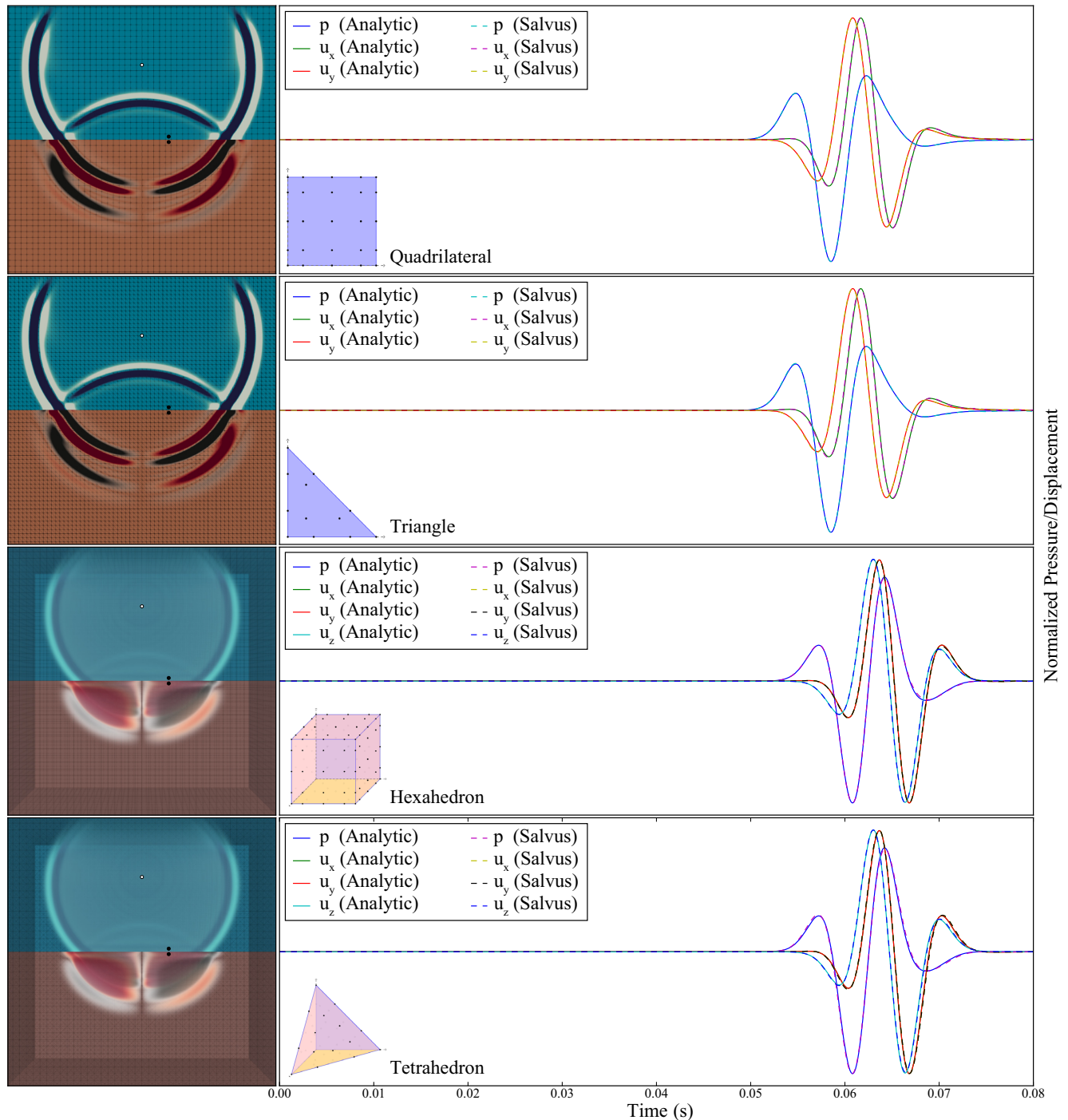
Figure 4. Benchmark simulations of coupled acoustic/elastic wave propagation on quadrilateral, triangular, hexahedral and tetrahedral meshes (top to bottom). The 2-D and 3-D reference elements are visualized. Simulation domain is 500 m in each dimension, with the coupling boundary placed at y or $z = 250$ m (depending on the dimension). A Ricker-wavelet source with a central frequency of 100 Hz was placed 125 m below the top boundary, and was centred with respect to the remaining boundaries (white dot). Receivers measuring pressure were placed 5 m above the interface, and those measuring displacement were placed 5 m below (black dots). V_p and V_s in the solid were 5800 and 4000 m s^{-1}, respectively, while V_p in the fluid was 4000 m s^{-1}. ρ was 2600 kg m^{-3}. Panels on the right show normalized comparisons between numerically generated solutions, computed using the methods described in this paper, and semi-analytical solutions, generated with Gar6more2D (Diaz & Ezziani 2010a) and Gar6more3D (Diaz & Ezziani 2010b). The scalar pressure (p) is plotted in the acoustic medium, along with the vector displacement (u_x, u_y, u_z) in the elastic medium. *n*-dimensional snapshots of the simulation volume are shown in panels on the left. In the fluid domain (blue), the color scale qualitatively shows the scalar displacement potential, while in the solid domain the *x*-component of displacement is visualized.
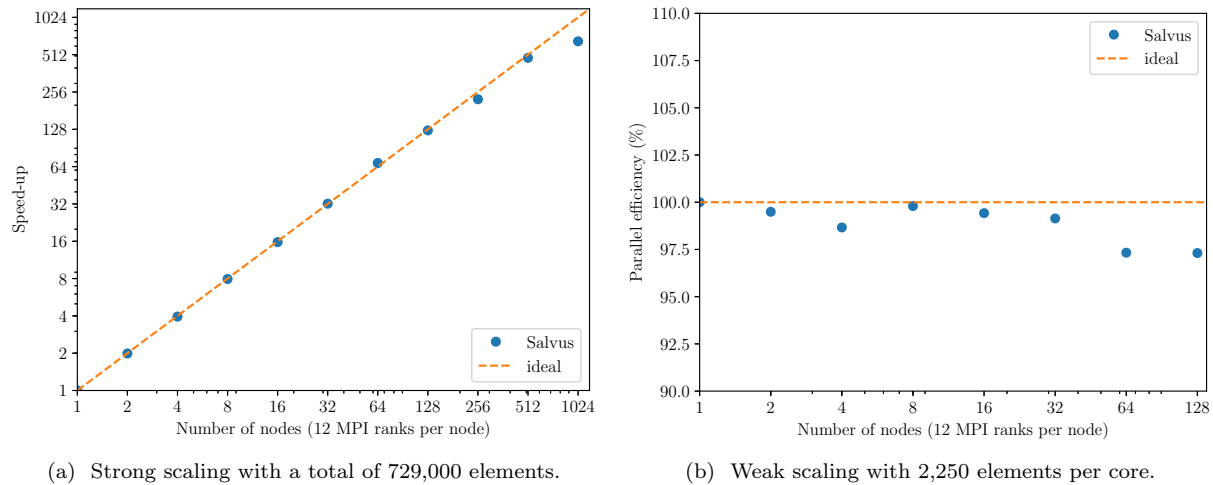
(a) Strong scaling with a total of 729,000 elements.

(b) Weak scaling with 2,250 elements per core.

Figure 5. Strong and weak scaling results for the time loop of a visco-elastic wave simulation on *Piz Daint* Cray XC50 the Swiss National Supercomputing Centre. Each node was equipped with 12 compute cores.

The peak theoretical double precision floating-point performance of the entire chip is reported as 345.6 GFLOPS and with six compute cores we expect a maximum per-core performance of 57.6 GFLOPS (Intel 2018). Our benchmark simulation consisted of a 3-D hexahedral mesh with 20736 4th-order isotropic elastic elements, and was run for 300 time steps. We measured the cumulative time spent in the computation of the elastic internal forces, as outlined in Listing 2. Carefully counting the number of floating-point operations within the strain, stress, and stiffness functions, we recorded a performance of 10.57 GFLOPS for double precision calculations. As this reaches an appreciable fraction of the maximum theoretical efficiency, and considering the memory-bound character of the computations, we can conclude that a significant amount of compute time is spent performing floating point operations, rather than resolving function calls within the mixin hierarchy. As further evidence of this, we can look at the total time spent within the internal forces calculation, and compare this time between single (18.37 s) and double (35.615 s) precision runs. The single precision run is almost exactly twice as fast as the double precision run. Due to memory bandwidth limitations, and fixed-width vector registers, this behaviour is expected. As the time taken to resolve the function calls are identical in both cases, we see that the expense of the floating-point operations are by far the dominant factor contributing to the wall time. If performance degradation from the mixin-based design was significant, we would expect to see the wall-time differential between single and double precision runs to narrow, but this is not the case.

In addition to the efficiency of the element-wise operations, parallel scaling performance is extremely important. To assist with parallelism, we can again leverage the expertise of PETSc, which provides a distributed vector API and hides the complexity of broadcasting the degrees of freedom on partition boundaries. The diagonal nature of the spectral-element mass matrix ensures that the communication halo never extends into neighbouring partitions, allowing for very efficient communication patterns which scale with the surface area of each parallel partition. Fig. 5 shows scaling statistics on the Piz Daint supercomputer, an XC50 system composed of 12-core Intel Haswell nodes located at the Swiss National Supercomputing Center. Strong scaling results, where a variable number of compute cores are used to solve a problem of a fixed size, are shown on the left. Here we consider a 3-D mesh with 729 000 elements and a polynomial degree of 4, resulting in 125 degrees of freedom per element. Visco-elastic wave propagation is simulated. The scaling is near optimal until the number of elements per core drops to approximately 1000, where we see slightly greater than 100 per cent parallel efficiency. We attribute this 'super-scaling' to processor cache effects; since less memory is used per core, it is more likely that data will remain in the processor cache throughout the simulation. Eventually, the parallel efficiency drops if there are too few elements per core and the communication overhead starts to dominate the cost of elemental operations. This can be seen from the last data point in Fig. 5(a), which uses 1024 nodes (12 288 cores) and roughly 60 elements per core. Fig. 5(b) presents weak scaling results. In this case, the number of elements per core remains fixed at 2250, while the total problem size is increased. Here we again see greater than 97 per cent parallel efficiency when increasing the total number of elements from 27 000 to 3 456 000.

To lower the communication costs associated with distributed memory parallelism, we overlap computation and communication during the global finite-element assembly phase. On each core the degrees of freedom shared between neighbouring elements are duplicated, and the stiffness term for each element is computed sequentially. Following this, we begin a broadcast of the ghost nodes. Asynchronously from this broadcast we sequentially assemble the global stiffness vector on each core, and once this is complete halt until the communication has finished. Finally, we finish the parallel global assembly by incorporating the communicated stiffness values on each core. Currently all communication is handled via the Message Passing Interface (MPI) (MPI Forum 2009), and no optimizations for shared memory are made. The explicit use of shared memory is a topic for future research, but as the communication overhead is already small we do not expect a significant increase in performance when considering CPU-only architectures.

4.2 Outlook

The abstractions introduced in this paper allow for a significant amount of application flexibility. The dimension independence promotes experimentation on small 2-D domains, the results of which can be immediately used and translated to large 3-D experiments. An example might be experimental design in exploration seismic surveys. Given a model of the subsurface, the optimal spacing of
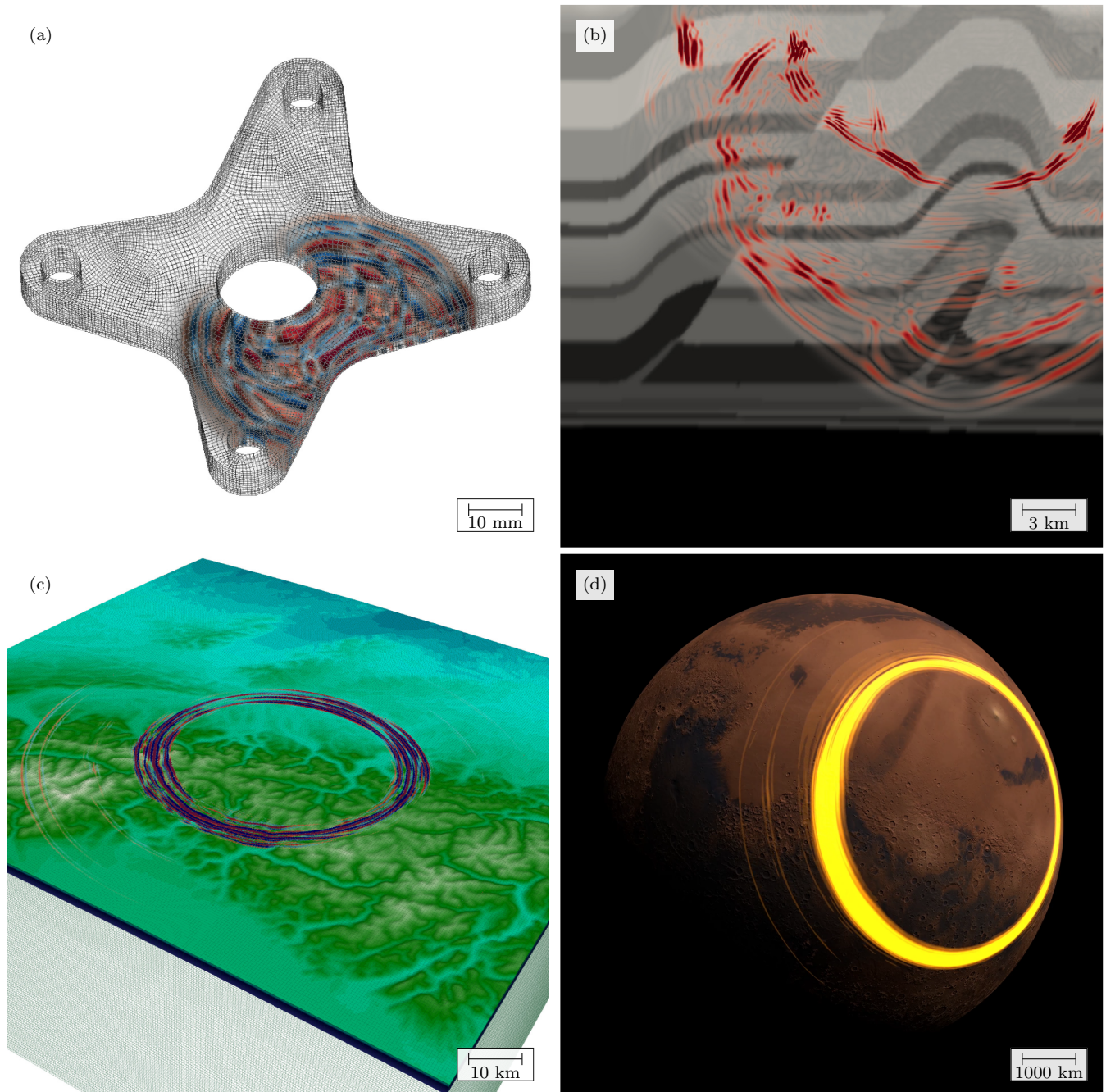
Figure 6. Qualitative illustrations of SALVUS running forward simulations at the laboratory, exploration, regional and planetary scale. (a) Elastic simulation of ultrasonic waves in a steel engine support clamp, using a setup similar to that described in Fig. 7(a) and discretized with approximately 10 000 elements; (b) Elastic simulation through a 2-D slice of the SEG Overthrust model (Aminsadeh 1996), also discretized with approximately 10 000 elements; (c) Elastic simulation of the 2017 Linthal Valley earthquake in Switzerland, using the PREM (Dziewoński & Anderson 1981) as a background velocity model, and discretized with 6.5 million elements; (d) 6.7 million element coupled acoustic/elastic wave propagation through a radially symmetric 1-D model of Mars (Khan *et al.* 2016), over which is imposed a crustal model with 3-D Moho and surface topography (Smith *et al.* 1999; Wieczorek & Zuber 2004). In all of these examples, hypercube-type elements and the GLL basis were used. In addition to the different spatial scales simulated, the computing power required for each simulation varied significantly: (a) was run on a desktop workstation with 8 processor cores, (b) was run on a single core on a laptop, (c) was run on 1200 processor cores on the Piz Daint supercomputer at the Swiss National Supercomputing Center (CSCS), (d) was run with 2400 cores on the same machine.

sources and receivers along a seismic line can be determined in 2-D, using computing resources no more powerful than a standard desktop computer. Once the design is set, the same interface can be used to scale the problem to 3-D on massively parallel supercomputers (Fig. 6).

The flexibility also encourages the application of FWI to novel domains. Recently, the method has been applied to problems in ultrasonic medical imaging (Pratt *et al.* 2007; Korta Martiartu *et al.* 2017) and non-destructive testing (Rao *et al.* 2016; Seidl & Rank
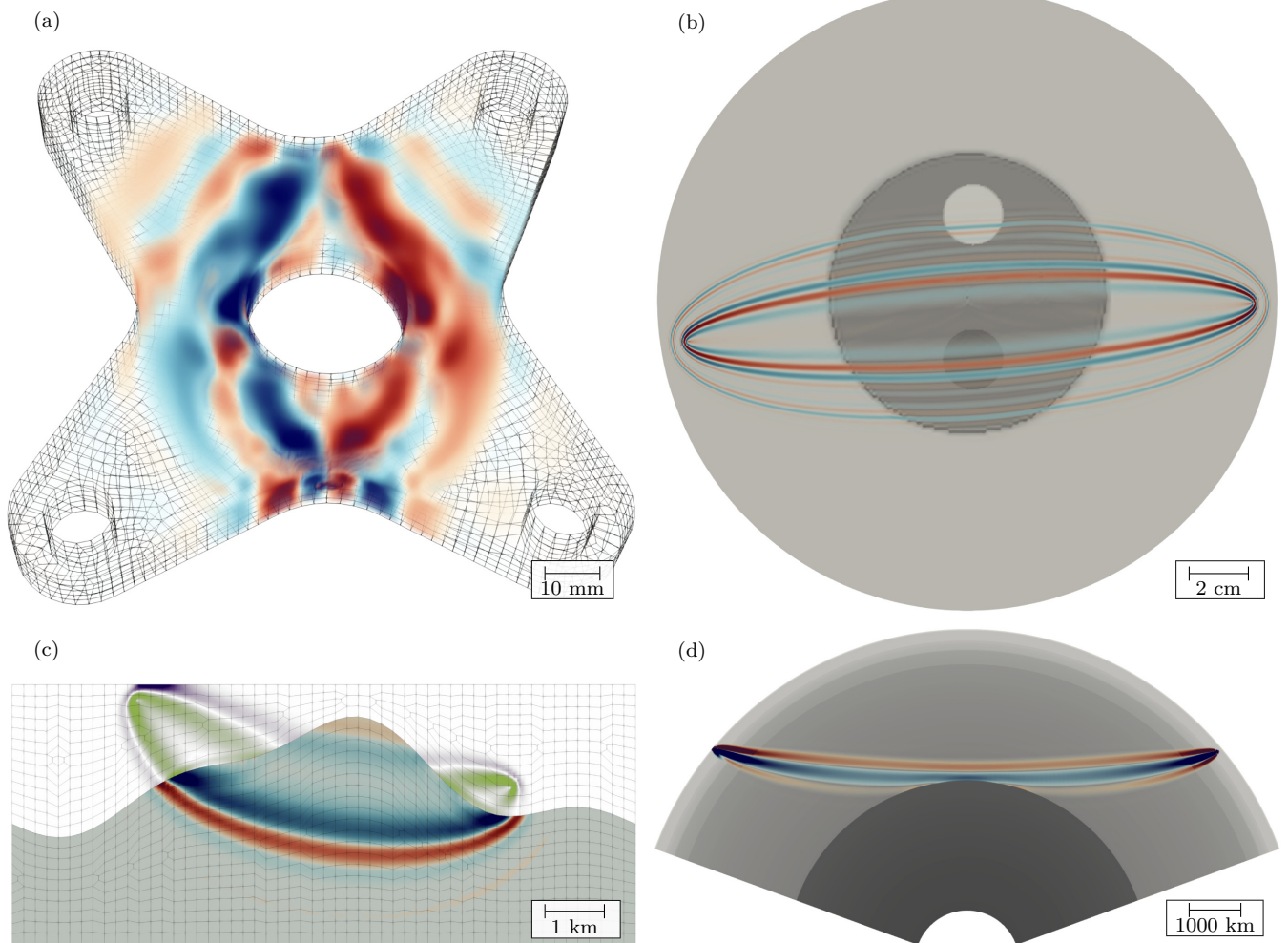
Figure 7. Gallery of normalized gradients for different wave propagation physics and scales ranging from millimetres to thousands of kilometres, meant to exhibit the range of problems accessible to a general implementation. All examples consider a single source–receiver pair. Blue hues represent regions where an increase in the value of the relevant parameter will decrease the associated misfit measure; red hues represent the opposite. (a) L^2 sensitivity to shear modulus μ on an engine support clamp, using a wavefield generated by a 200 kHz Ricker source; (b) L^2 sensitivity to the speed of sound in a synthetic breast phantom (grey-scale image) for ultrasound tomography, using a wavefield generated by a 500 kHz Ricker source. (c) L^2 sensitivity to fluid compressibility (purple/green hues) and shear modulus (red/blue hues) in a coupled acoustic/elastic domain, using a 3 Hz Ricker wavelet in the fluid region as a source; (d) Pdiff sensitivity to cross-correlation traveltime shifts on a 2-D slice through Earth, using PREM (Dziewoński & Anderson 1981) as a background model. The source used was a moment tensor with a dominant period of 10 s.

2017). These situations bring realistic domains of significant complexity. In medical imaging, this includes coupling between tissue, which is mostly modelled as a fluid, to elastic bones, and to the emitting devices themselves. In non-destructive testing applications, complex rheologies including hyperelastic and weakly non-linear wave propagation physics may be essential for accurate modelling. These situations can be investigated with the help of the model/mixin approaches outlined above, without any major changes in software design. A preliminary gallery of sensitivity kernels (Tarantola 1984; Tromp *et al.* 2005; Fichtner *et al.* 2006), a first step towards FWI, is given in Fig. 7. Additionally, the geometric complexities of these applications may warrant simulation on non-conforming meshes, motivating the implementation of discontinuous-Galerkin (DG) style basis functions (Käser *et al.* 2007; Wilcox *et al.* 2010). As with the complex rheologies, the implementation of these bases need only involve

the $\mathcal{B}$ model, without changes to the WaveEquation and Element mixins.

5 CONCLUSIONS

Waveform modelling within domains of a realistic physical complexity almost always requires fully numerical solutions to the wave equation. While traditionally undertaken within a geophysical context, advances in acquisition hardware, computational power, and method development has led to the application of these solutions to a growing variety of problems both within and external to seismology. This growth in the application space has driven an equivalent growth in the complexity of waveform modelling software which represents a threat to maintainability of current and future developments. As one of the primary analytical tools used to investigate the character of Earth, this is a serious issue.

In this paper, we have introduced a set of conceptual and practical developments which attempt to mitigate the effects of the combinatorial complexity that characterizes modern applications of full-waveform modelling and inversion. We first focussed on a modular interpretation of seismologically relevant wave equations, where the wave propagation physics and the spatial discretization were separated into distinct and functionally orthogonal mathematical models. We then showed how these abstract concepts could be made concrete in modelling software, and gave examples of the variety of applications made accessible by such an approach. It is our hope that this manuscript demonstrates that it is indeed possible *and* practical to develop maintainable and scalable modelling tools.

REFERENCES

Afanasiev, M., Peter, D., Sager, K., Simutė, S., Ermert , L., Krischer, L. & Fichtner, A., 2016. Foundations for a multiscale collaborative global Earth model, *Geophys. J. Int.,* **204,** 39–58.

Aki, K. & Richards, P., 2002. *Quantitative Seismology.,* University Science Books.

Alterman, Z. & Karal, F.C., 1968. Propagation of elastic waves in layered media by finite-difference methods, *Bull. seism. Soc. Am.,* **58,** 367–398.

Aminsadeh, F., 1996. 3-D salt and overthrust seismic models, in *AAPG Studies in Geology No. 42 and SEG Geophysical Developments Series No. 5,* pp. 247–256, eds., Weimer, P. & Davis, T. L., AAPG/SEG, Tulsa.

Babuška, V. & Cara, M., 1991. *Seismic Anisotropy in the Earth.,* Kluwer Academic Publishers.

Balay, S., Gropp, W.D., McInnes, L.C. & Smith, B.F., 1997. Efficient management of parallelism in object oriented numerical software libraries, in *Modern Software Tools in Scientific Computing,* pp. 163–202, Birkhäuser Press.

Balay, S. *et al.*, 2017a. *PETSc users manual, Tech. Rep. ANL-95/11 - Revision 3.8,* Argonne National Laboratory.

Balay, S. *et al.*, 2017b. *PETSc Web page,* http://www.mcs.anl.gov/petsc.

Bangerth, W., Hartmann, R. & Kanschat, G., 2007. deal.II – a general purpose object oriented finite element library, *ACM Trans. Math. Softw.,* **33**(4), 24/1–24/27.

Bauman, P.T. & Stogner, R.H., 2016. Grins: A multiphysics framework based on the libmesh finite element library, *SIAM J. Sci. Comp.,* **38**(5), S78–S100.

Boore, D.M., 1970. Love waves in nonuniform waveguides: finite difference calculations, *J. geophys. Res.,* **1970,** 1512–1527.

Bozdag, E., Peter, D., Lefebvre, M., Komatitsch, D., Tromp, J., Hill, J., Podhorszki, N. & Pugmire, D., 2016. Global adjoint tomography: first-generation model, *Geophys. J. Int.,* **207**(3), 1739.

Braun, J. & Sambridge, M., 1995. A numerical method for solving partial differential equations on highly irregular evolving grids, *Nature,* **376,** 655–660.

Brenders, A.J. & Pratt, R.G., 2007. Full waveform tomography for lithospheric imaging: results from a blind test in a realistic crustal model, *Geophys. J. Int.,* **168,** 133–151.

Brenner, S. & Scott, R., 2007. *The mathematical theory of finite element methods,* vol. **15,** Springer Science & Business Media.

Cance, P. & Capdeville, Y., 2015. Validity of the acoustic approximation for elastic waves in heterogeneous media, *Geophysics,* **80**(4), T161–T173.

Capdeville, Y., Chaljub, E., Vilotte, J.P. & Montagner, J.P., 2003. Coupling the spectral element method with a modal solution for elastic wave propagation in global earth models, *Geophys. J. Int.,* **152,** 34–66.

Chaljub, E. & Tarantola, A., 1997. Sensitivity of SS precursor to topography on the upper–mantle 660-km discontinuity, *Geophys. Res. Lett.,* **24,** 2613–2616.

Chaljub, E. & Valette, B., 2004. Spectral element modelling of three-dimensional wave propagation in a self-gravitating earth with an arbitrarily stratified outer core, *Geophys. J. Int.,* **158**(1), 131–141.

Chaljub, E., Capdeville, Y. & Vilotte, J., 2003. Solving elastodynamics in a solid heterogeneous 3-sphere: a spectral element approximation on geometrically non-conforming grids, *J. Comp. Physics,* **183,** 457–491.

Chaljub, E., Komatitsch, D., Vilotte, J.-P., Capdeville, Y., Valette, B. & Festa, G., 2007. Spectral-element analysis in seismology, *Adv. Geophys.,* **48,** 365–419.

Chin-Joe-Kong, M., Mulder, W.A. & Van Veldhuizen, M., 1999. Higher-order triangular and tetrahedral finite elements with mass lumping for solving the wave equation, *J. Eng. Math.,* **35**(4), 405–426.

Clayton, R. & Engquist, B., 1977. Absorbing boundary conditions for acoustic and elastic wave equations., *Bull. seism. Soc. Am.,* **67,** 1529–1540.

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

Colli, L., Fichtner, A. & Bunge, H.-P., 2013. Full waveform tomography of the upper mantle in the South Atlantic region: Imaging westward fluxing shallow asthenosphere?, *Tectonophysics,* **604,** 26–40.

Cowling, T.G., 1941. The non-radial oscillations of polytropic stars, *Mon. Not. R. Astron. Soc.,* **101,** 367.

Cupillard, P., Delavaud, E., Burgos, G., Festa, G., Vilotte, J.-P., Capdeville, Y. & Montagner, J.-P., 2012. RegSEM: a versatile code based on the spectral element method to compute seismic wave propagation at the regional scale, *Geophys. J. Int.,* **188,** 1203–1220.

de la Puente, J., Dumbser, M., Käser, M. & Igel, H., 2007. An arbitrary high-order discontinuous Galerkin method for elastic waves on unstructured methods - iv. anisotropy, *Geophys. J. Int.,* **169,** 1210–1228.

de la Puente, J., Ferrer, M., Hanzich, M., Castillo, J.E. & Cela, J.M., 2014. Mimetic seismic wave modelling including topography on deformed staggered grids, *Geophysics,* **79**(3), T125–T141.

Dedner, A., Klöfkorn, R., Nolte, M. & Ohlberger, M., 2010. A generic interface for parallel and adaptive discretization schemes: abstraction principles and the dune-fem module, *Computing,* **90**(3), 165–196.

Deville, M., Fischer, P. & Mund, E., 2002. *High-Order Methods for Incompressible Fluid Flow,* Cambridge University Press.

Diaz, J. & Ezziani, A., 2010a. Analytical solution for waves propagation in heterogeneous acoustic/porous media. Part I: the 2D case, *Commun. Comput. Phys.,* **7**(1), 171–194.

Diaz, J. & Ezziani, A., 2010b. Analytical solution for waves propagation in heterogeneous acoustic/porous media. Part II: the 3D case, *Commun. Comput. Phys.,* **7**(3), 445–472.

Dumbser, M., Käser, M. & de la Puente, J., 2007. Arbitrary high-order finite-volume schemes for seismic wave propagation on unstructured meshes in 2D and 3D, *Geophys. J. Int.,* **171,** 665–694.

Dziewoński, A.M. & Anderson, D.L., 1981. Preliminary reference Earth model, *Phys. Earth planet. Inter.,* **25,** 297–356.

Emmerich, H. & Korn, M., 1987. Incorporation of attenuation into time-domain computations of seismic wave fields, *Geophysics,* **52,** 1252–1264.

Faccioli, E., Maggio, F., Quarteroni, A. & Tagliani, A., 1996. Spectral-domain decomposition methods for the solution of acoustic and elastic wave equations, *Geophysics,* 61:4, 1160–1174.

Faccioli, E., Maggio, F., Paolucci, R. & Quarteroni, A., 1997. 2D and 3D elastic wave propagation by a pseudospectral domain decomposition method, *J. Seismol.,* **1,** 237–251.

Ferroni, A., Antonietti, P., Mazzieri, I. & Quarteroni, A., 2017. Dispersion-dissipation analysis of 3D continuous and discontinuous spectral element methods for the elastodynamics equation, *Geophys. J. Int.,* p. ggx384.

Fichtner, A. & Igel, H., 2008. Efficient numerical surface wave propagation through the optimization of discrete crustal models - a technique based on non-linear dispersion curve matching (DCM), *Geophys. J. Int.,* **173,** 519–533.

Fichtner, A., Bunge, H.-P. & Igel, H., 2006. The adjoint method in seismology - I. Theory, *Phys. Earth planet. Inter.,* **157,** 86–104.

Fichtner, A., Kennett, B.L.N., Igel, H. & Bunge, H.-P., 2009. Spectral-element simulation and inversion of seismic waves in a spherical section of the Earth., *J. Numer. Anal. Ind. Appl. Math.,* **4,** 11–22.

Fichtner, A., Trampert, J., Cupillard, P., Saygin, E., Taymaz, T., Capdeville, Y. & Villasenor, A., 2013. Multi-scale full waveform inversion, *Geophys. J. Int.,* **194,** 534–556.

Fichtner, A. *et al.*, 2018. The collaborative seismic earth model: generation 1, *Geophys. Res. Lett.,* **45.**

Fornberg, B., 1988. The pseudospectral method: accurate representation of interfaces in elastic wave calculations, *Geophysics,* **53**(5), 625–637.

French, S.W. & Romanowicz, B.A., 2014. Whole-mantle radially anisotropic shear velocity structure from spectral-element waveform tomography, *Geophys. J. Int.,* **199**(3), 1303–1327.

Gauthier, O., Virieux, J. & Tarantola, A., 1986. Two-dimensional nonlinear inversion of seismic waveforms: numerical results., *Geophysics,* **51,** 1387–1403.

Geller, R.J. & Takeuchi, N., 1998. Optimally accurate second-order time-domain finite difference scheme for the elastic equation of motion: one-dimensional case, *Geophys. J. Int.,* **135**(1), 48–62.

Gokhberg, A. & Fichtner, A., 2016. Full-waveform inversion on heterogeneous HPC systems, *Comp. Geosci.,* **89**(Suppl. C), in review.

Hestholm, S., 1999. Three-dimensional finite difference viscoelastic wave modelling including surface topography, *Geophys. J. Int.,* **139**(3), 852–878.

Hughes, T., 2000. *The Finite Element Method,* Dover Publications.

Igel, H. & Weber, M., 1995. SH-wave propagation in the whole mantle using high-order finite differences, *Geophys. Res. Lett.,* **22**(6), 731–734.

Igel, H., Mora, P. & Riollet, B., 1995. Anisotropic wave propagation through FD grids, *Geophysics,* **60,** 1203–1216.

Intel, 2018. *Export compliance metrics for intel microprocessors,* https://www.intel.com/content/dam/support/us/en/documents/processors/APP-for-Intel-Core-Processors.pdf.

Käser, M. & Igel, H., 2001. Numerical simulation of 2D wave propagation on unstructured grids using explicit differential operators, *Geophys. Prospect.,* **49**(5), 607–619.

Käser, M., Igel, H., Sambridge, M. & Braun, J., 2001. A comparative study of explicit differential operators on arbitrary grids, *J. Comput. Acoust.,* **09**(03), 1111–1125.

Käser, M., Mai, P.M. & Dumbser, M., 2007. Accurate calculation of fault-rupture models using the high-order discontinuous galerkin method on tetrahedral meshes, *Bull. seism. Soc. Am,* 97(5), 1570–1586.

Kelly, K., Ward, R., Treitel, S. & Alford, R., 1976. Synthetic seismograms: a finite difference approach, *Geophysics,* **41,** 2–27.

Khan, A. *et al.*, 2016. Single-station and single-event marsquake location and inversion for structure using synthetic martian waveforms, *Phys. Earth planet. Inter.,* **258**(Supplement C), 28–42.

Knepley, M.G. & Karpeev, D.A., 2009. Mesh algorithms for PDE with Sieve I: mesh distribution, *Sci Program,* **17**(3), 215–230.

Komatitsch, D. & Tromp, J., 1999. Introduction to the spectral element method for three-dimensional seismic wave propagation, *Geophys. J. Int.,* **139,** 806–822.

Komatitsch, D. & Tromp, J., 2002a. Spectral-element simulations of global seismic wave propagation, part II: 3-D models, oceans, rotation, and gravity, *Geophys. J. Int.,* **150,** 303–318.

Komatitsch, D. & Tromp, J., 2002b. Spectral-element simulations of global seismic wave propagation, part I: validation, *Geophys. J. Int.,* **149,** 390–412.

Komatitsch, D., Barnes, C. & Tromp, J., 2000. Wave propagation near a fluid-solid interface: a spectral element approach, *Geophysics,* **65**(2), 623–631.

Korta Martiartu, N., Boehm, C., Vinard, N., Jovanovi Balic, I. & Fichtner, A., 2017. Optimal experimental design to position transducers in ultrasound breast imaging, *Proc. SPIE,* **10139,** 101390M–101390M–15.

Lailly, P., 1983. The seismic inverse problem as a sequence of before stack migrations, in *Proceedings of the Conference on Inverse Scattering: Theory and Application,* Soc. Industr. appl. Math., Philadelphia, PA.

Lange, M., Mitchell, L., Knepley, M.G. & Gorman, G.J., 2016. Efficient mesh management in Firedrake using PETSc-DMPlex, *SIAM J. Sci. Comput.,* **38**(5), S143–S155.

Logg, A., Mardal, Kent-Andre & Wells, Garth, 2012. *Automated Solution of Differential Equations by the Finite Element Method,* Springer.

Madariaga, R., 1976. Dynamics of an expanding circular fault, *Bull. seism. Soc. Am.,* **65,** 163–182.

Marfut, K.J., 1984. Accuracy of finite–diference and finite–element modelling of the scalar wave equation, *Geophysics,* **49,** 533–549.

Moczo, P. & Kristek, J., 2005. On the rheological models for time-domain methods of seismic wave propagation, *Geophys. Res. Lett.,* **32** .

Moczo, P., Kristek, J., Vavrycuk, V., Archuleta, R. & Halada, L., 2002. 3D heterogeneous staggered-grid finite-difference modelling of seismic motion with volume harmonic and arithmetic averaging of elastic moduli, *Bull. seism. Soc. Am.,* **92,** 3042–3066.

Moczo, P., Kristek, J. & Galis, M., 2014. *The Finite-Difference Modelling of Earthquake Motions: Waves and Ruptures,* Cambridge University Press.

MPI Forum, 2009. Message Passing Interface (MPI) Forum Home Page, http://www.mpi-forum.org/.

Mulder, W. & Shamasundar, R., 2016. Performance of continuous mass-lumped tetrahedral elements for elastic wave propagation with and without global assembly, *Geophys. J. Int.,* **207**(1), 414.

Nissen-Meyer, T., Fournier, A. & Dahlen, F.A., 2008. A two-dimensional spectral-element method for computing spherical-earth seismograms - II. waves in solid-fluid media., *Geophys. J. Int.,* **174,** 873–888.

Nissen-Meyer, T., van Driel, M., Stähler, S., Hosseini, K., Hempel, S., Auer, L. & Fournier, A., 2014. AxiSEM: broadband 3-D seismic wavefields in axisymmetric media, *Solid Earth,* **5,** 425–445.

Ohminato, T. & Chouet, B.A., 1997. A free-surface boundary condition for including 3D topography in the finite-difference method, *Bull. seism. Soc. Am.,* **87**(2), 494–515.

Peter, D. *et al.*, 2011. Forward and adjoint simulations of seismic wave propagation on fully unstructured hexahedral meshes, *Geophys. J. Int.,* **186,** 721–739.

Petersson, N.A. & Sjögreen, B., 2015. Wave propagation in anisotropic elastic materials and curvilinear coordinates using a summation-by-parts finite difference method, *J. Comp. Phys.,* **299,** 820–841.

Pratt, R., Shin, C. & Hicks, G., 1998. Gauss-Newton and full Newton methods in frequency domain seismic waveform inversion, *Geophys. J. Int.,* **133,** 341–362.

Pratt, R.G., Huang, L., Duric, N. & Littrup, P., 2007. Sound-speed and attenuation imaging of breast tissue using waveform tomography of transmission ultrasound data, *Proc. SPIE,* **6510,** 65104S–65104S–12.

Prieux, V., Brossier, R., Operto, S. & Virieux, J., 2013. Multiparameter full waveform inversion of multicomponent ocean-bottom-cable data from the Valhall field. Part 1: Imaging compressional wave speed, density and attenuation, *Geophys. J. Int.,* **194,** 1640–1664.

Quarteroni, A., Sacco, R. & Saleri, F., 2010. *Numerical Mathematics,* Vol. **37,** Springer Science & Business Media.

Rao, J., Ratassepp, M. & Fan, Z., 2016. Guided wave tomography based on full waveform inversion, *IEEE Trans. Ultrason., Ferroelect., Freq. Control,* **63**(5), 737–745.

Rivière, J., Renaud, G., Guyer, R.A. & Johnson, P.A., 2013. Pump and probe waves in dynamic acousto-elasticity: comprehensive description and comparison with nonlinear elastic theories, *J. Appl. Phys.,* **114**(5), 054905.

Robertsson, J.O.A., 1996. A numerical free-surface condition for elastic/viscoelastic finite-difference modelling in the presence of topography, *Geophysics,* **61**(6), 1921–1934.

Robertsson, J.O.A., Blanch, J.O. & Symes, W.W., 1994. Viscoelastic finite-difference modelling, *Geophysics,* **59**, 1444–1456.

Seidl, R. & Rank, E., 2017. Full waveform inversion for ultrasonic flaw identification, *AIP Conf. Proc.,* **1806**(1), 090013.

Seriani, G. & Priolo, E., 1994. Spectral element method for acoustic wave simulation in heterogeneous media, *Finite Elem. Anal. Des.,* **16**, 337–348.

Shragge, J., 2016. Acoustic wave propagation in tilted transversely isotropic media: incorporating topography, *Geophysics,* **81**(5), C265–C278.

Simutė, S., Steptoe, H., Cobden, L., Gokhberg, A. & Fichtner, A., 2016. Full-waveform inversion of the Japanese islands region, *J. geophys. Res.,* **121**(5), 3722–3741.

Smith, D.E. *et al.*, 1999. The global topography of mars and implications for surface evolution, *Science,* **284**(5419), 1495–1503.

Tago, J., Cruz-Atienza, V.M., Virieux, J., Etienne, V. & Sánchez-Sesma, F.J., 2012. A 3D hp-adaptive discontinuous galerkin method for modelling earthquake dynamics, *J. geophys. Res,* **117**(B9), B09312.

Tape, C., Liu, Q., Maggi, A. & Tromp, J., 2009. Adjoint tomography of the southern California crust., *Science,* **325**, 988–992.

Tarantola, A., 1984. Inversion of seismic reflection data in the acoustic approximation, *Geophysics,* **49**, 1259–1266.

Tromp, J., Tape, C. & Liu, Q., 2005. Seismic tomography, adjoint methods, time reversal and banana-doughnut kernels, *Geophys. J. Int.,* **160**, 195–216.

van Driel, M. & Nissen-Meyer, T., 2014. Optimized viscoelastic wave propagation for weakly dissipative media, *Geophys. J. Int.,* **199**(2), 1078.

Virieux, J., 1984. SH wave propagation in heterogeneous media: velocity-stress finite difference method, *Geophysics,* **49**, 1933–1942.

Virieux, J., 1986. P-SV wave propagation in heterogeneous media: velocity-stress finite difference method, *Geophysics,* **51**, 889–901.

Virieux, J. & Operto, S., 2009. An overview of full waveform inversion in exploration geophysics, *Geophysics,* **74**, WCC127–WCC152.

Wieczorek, M.A. & Zuber, M.T., 2004. Thickness of the martian crust: Improved constraints from geoid-to-topography ratios, *J. geophys. Res.,* **109**(E1), E01009.

Wilcox, L.C., Stadler, G., Burstedde, C. & Ghattas, O., 2010. A high-order discontinuous galerkin method for wave propagation through coupled elastic-acoustic media, *J. Comp. Phys.,* **229**(24), 9373–9396.

Zhebel, E., Minisini, S., Kononov, A. & Mulder, W.A., 2014. A comparison of continuous mass-lumped finite elements with finite differences for 3D wave propagation, *Geoph. Prospect.,* **62**(5), 1111–1125.

Zienkiewicz, O.C., Taylor, R.L. & Z, Z.J., 2005. *The Finite Element Method: Its Basis and Fundamentals,* 6th edn, Elsevier.