Application of code generation to high-order seismic modelling with the discontinuous Galerkin finite element method

Christian T. Jacobs, Michael Lange, Fabio Luporini, Gerard J. Gorman

Abstract

Modern seismological models feature high-order numerical schemes that are able to more accurately capture the P-wave and S-waves. Such schemes include finite difference methods (FDMs) which have been the norm for many years. However, implementing higher-order stencils for FDMs is difficult, often incurring a considerable development cost and yielding non-optimal code. Furthermore, the code is not future proof; running on more exotic hardware such as GPUs in the future may require large sections of the code to be re-written, thus adding to the overall development cost. Recent efforts in seismological modelling have also considered the discontinuous Galerkin finite element method (DG-FEM) which is more suitable for meshes that confine to complex domain geometries. However, existing implementations of DG-FEM suffer from the same drawbacks. This paper presents a new approach to high-order DG-FEM seismological modelling through the application of an automated code generation framework, called Firedrake, to produce an optimised seismological model from an abstract problem specification written in a domain-specific language.

1 Introduction

Seismological modelling has traditionally involved the use of finite difference methods (FDMs) to solve the equations governing the propagation of acoustic and elastic waves (Kelly et al., 1976; Virieux, 1986; Graves, 1996). Such methods are relatively easy to implement, and direct addressing and regular memory access patterns means that good performance can be readily achieved (Liu et al., 2014). Therefore, a lot of effort has focused on optimising these models for use in seismic imaging on a wide range of hardware architectures.

One way to increase the accuracy and computational efficiency is to use higher-order methods. As the order increases, the grid size (and therefore the timestep) can also increase. This is particularly important for elastic wave modelling because the shear waves (S-waves) generated have a smaller wavelength than the pressure wave (P-wave) and therefore require higher resolution. However, as the order of the FDM stencils increases, so does the cost of creating optimised implementations of these methods on modern many-core and multi-core platforms.

Interest in stencil languages and compilers such as Liszt (DeVito et al., 2011) and Pochoir (Tang et al., 2011) has rapidly grown in recent years. Using a stencil language in effect allows for a seperation of concerns between those developing the numerical algorithms, and compiler specialists who optimise these parallel patterns onto different computer architectures. FDMs are readily expressed in stencil languages and this provides a programmatic route to generating implementations of high-order FDM stencils. However, FDM methods themselves have their own problems; the inherent structured nature of the underlying computational mesh is unable to follow realistic topography (Liu et al., 2014), and unsuitable for adapting to features in the subsurface model (e.g. one ideally would like a coarser mesh in salt domes where the wave velocity is much higher than the surrounding medium).

Promising results have recently been published on high-order discontinuous Galerkin (DG) finite element methods (FEMs) (Dumbser and Käser, 2006; Käser and Dumbser, 2006; de la Puente et al., 2008, 2009; Delcourte et al., 2009; Hermann et al., 2011; Wenk et al., 2013; Tago et al., 2014; Mercerat and Glinsky, 2015). FEMs are more suitable for simulations requiring unstructured meshes that conform well to arbitrary geometries. Much like high-order finite difference stencils, it would be beneficial if higher-order DG FEM methods could be generated with ease.

Code generation techniques provided by the FEniCS framework (Logg et al., 2012; Logg and Wells, 2010) and, more recently, the Firedrake framework (Rathgeber et al., Submitted), have provided a way to produce optimised finite element-based models from a high-level problem description. Model developers provide the weak form of the equations they wish to solve using a language called the Unified Form Language (UFL) (Alnæs et al., 2014), which then gets optimised and compiled into optimised low-level C code (Luporini et al., 2015). A finite element tabulator called FIAT (Kirby, 2004) is used to construct arbitrary order polynomial spaces which is ideal for higher-order DG methods. Much like the FDM stencil compilers, these code generation frameworks provide a separation of concerns between application specialists and parallel computing experts. Furthermore, code generation techniques can potentially yield significant performance benefits over traditional 'static'/hard-coded finite element models which may be difficult or costly to optimise by hand.

In addition to many-core CPUs, the field of seismic modelling has already begun to benefit from using more exotic hardware architectures such as GPUs (Weiss and Shragge, 2013). In the case of the Firedrake framework, the code generation workflow allows the model code to be targeted towards a particular hardware architecture using the PyOP2 library (Rathgeber et al., 2012; Markall et al., 2013), thereby future-proofing the model in the face of future high-performance computer architectures such as Intel® Xeon Phi™ coprocessors.

The work presented in this paper demonstrates the first known application of code generation techniques to high-order, finite element-based seismological modelling. The governing equations and associated initial and boundary conditions are first introduced in Section 2, followed by a description of the spatial and temporal discretisations chosen. The code generation technique used to produce the optimised finite element assembly code and automate the solution process is briefly described in Section 3. Test cases are then provided in Section 4 to verify and validate the correctness of the model implementation. Some concluding remarks are provided in Section 6.

2 The seismic model

2.1 Governing equations

This paper considers the elastic wave equation, a second-order linear PDE which may be written as

$$\rho \frac{\partial^2 \mathbf{r}}{\partial t^2} = \nabla \cdot \mathbb{T} \tag{1}$$

where t, ρ, $\mathbf{r}$, $\mathbb{T}$ are the time, density, displacement vector, and stress tensor, respectively (Virieux, 1986). The stress tensor is related, via Hooke's law, to the gradient of displacement such that

$$\mathbb{T} = \lambda \left(\nabla \cdot \mathbf{r}\right) \mathbb{I} + \mu \left(\nabla \mathbf{r} + \nabla \mathbf{r}^{\mathrm{T}}\right). \tag{2}$$

The coefficients λ and μ are known as the Lamé parameters, and are assumed to be constant (in both space and time) in this paper such that the elastic waves travel through isotropic media. The tensor $\mathbb{I}$ is the identity tensor. By defining the velocity $\mathbf{u}$ as

$$\mathbf{u} = \frac{\partial \mathbf{r}}{\partial t} \tag{3}$$

equation (1) can be re-written as two first-order linear PDEs

$$\rho \frac{\partial \mathbf{u}}{\partial t} = \nabla \cdot \mathbb{T} \tag{4}$$

$$\frac{\partial \mathbb{T}}{\partial t} = \lambda \left(\nabla \cdot \mathbf{u}\right) \mathbb{I} + \mu \left(\nabla \mathbf{u} + \nabla \mathbf{u}^{\mathrm{T}}\right) \tag{5}$$

resulting in the so-called velocity-stress formulation (Virieux, 1986) which will be solved for in the model presented here.

2.2 Boundary conditions

Integrating (4) by parts permits the weak imposition of the free-surface (also known as stress-free) boundary condition

$$\mathbb{T} \cdot \mathbf{n} = \mathbf{0}, \tag{6}$$

where $\mathbf{n}$ is the normal vector aligned with the boundary, and $\mathbf{0}$ is the zero vector.

Absorbing boundary conditions can be imposed to prevent waves from being reflected, thereby simulating an 'infinite' domain (see e.g. the upwind technique used by Käser and Dumbser (2006) and the perfectly matched layer technique adopted by Komatitsch and Tromp (2003)). However, for the simulations presented here we have opted to use a simple sponge/absorption region in the domain; this was achieved by adding an absorption term $-\sigma \mathbf{u}$, where σ is the problem-dependent absorption coefficient, to the right-hand side of (4). The absorption coefficient σ is non-zero only in the sponge region.

2.3 Spatial discretisation

2.3.1 Weak formulation

The discretisation of the governing equations in space was performed using the Galerkin finite element method. As a first step, the weak form of (4) and (5) is derived by multiplying through by a test function $\mathbf{w} \in H^1(\Omega)^d$, where $H^1(\Omega)^d$ is the first Hilbertian Sobolev space of dimension d (Elman et al., 2005), and integrating over the domain Ω by parts on the RHS. For (4), this yields

$$\int_\Omega \mathbf{w} \cdot \rho \frac{\partial \mathbf{u}}{\partial t} \, \mathrm{dV} = -\int_\Omega \mathbb{T} \cdot \nabla \mathbf{w} \, \mathrm{dV} + \int_{\partial \Omega_e} \left(\mathbf{w} \cdot \mathbb{T}|_{\partial \Omega_e}\right) \cdot \mathbf{n}_e \, \mathrm{dS} + \int_{\partial \Omega} \left(\mathbf{w} \cdot \mathbb{T}\right) \cdot \mathbf{n} \, \mathrm{ds} \tag{7}$$

where $\mathbf{n}$ is the vector normal to the domain boundary, $\partial\Omega$, and $\mathbf{n}$ is the vector normal to the facet of element e, $\partial\Omega_e$. Given this weak form, a solution $\mathbf{u} \in H^1(\Omega)^d$ is sought for all test functions $\mathbf{w} \in H^1(\Omega)^d$.

Discrete representations for both $\mathbf{w}$ and $\mathbf{u}$, given by a linear combination of basis functions $\{\phi_i\}_{i=1}^{N_e}$, which are discontinuous across element facets, are then used to replace them in (11):

$$\mathbf{w} = \sum_{i=1}^{N_e} \phi_i \mathbf{w}_i, \tag{8}$$

$$\mathbf{u} = \sum_{j=1}^{N_e} \phi_j \mathbf{u}_j, \tag{9}$$

where N is the total number of solution nodes. The stress field $\mathbb{T}$ is also represened by its own set of basis functions $\{\psi_i\}_{i=1}^{N_e}$

$$\mathbb{T} = \sum_{k=1}^{N_e} \psi_k \mathbb{T}_k. \tag{10}$$

By substituting (8), (9) and (10) into (11), and applying the fact that $\mathbf{w}_i$ are arbitrary, this yields

$$\sum_{j=1}^{N_e} \int_{\Omega_e} \phi_i \cdot \rho \phi_j \, \mathrm{dV} \, \frac{\partial \mathbf{u}_j}{\partial t} = -\sum_{k=1}^{N_e} \int_{\Omega_e} \psi_k \cdot \nabla \phi_i \, \mathrm{dV} \, \mathbb{T}_k + \sum_{k=1}^{N_e} \int_{\partial\Omega_e} (\phi_i \cdot \mathbb{T}|_{\partial\Omega_e}) \cdot \mathbf{n}_e \, \mathrm{dS} + \sum_{k=1}^{N_e} \int_{\partial\Omega} (\phi_i \cdot \mathbb{T}) \cdot \mathbf{n} \, \mathrm{ds} \tag{11}$$

for all ϕ_i in element e. The volume integrals and interior facet integrals have been restricted to an individual element e here because the discontinuous formulation results in each element becoming its own independent problem.

2.3.2 First-order upwinding

As a result of the discontinuous nature of the solution fields and test functions at element boundaries, the values of $\mathbb{T}|_{\partial\Omega_e}$ and $\mathbf{u}|_{\partial\Omega_e}$ in the facet integral terms must be treated carefully. For this work we use a first-order upwinding scheme such that the average values of $\mathbb{T}$ and $\mathbf{u}$ across the facets are used.

2.3.3 The discrete system

The end result is a discrete system of size $N \times N$ for both the velocity and stress equations:

$$\mathbf{M}\frac{\partial \mathbf{u}}{\partial t} = \mathbf{D}\mathbb{T}, \tag{12}$$

where

$$\mathbf{M}_{ij} = \int_{\Omega_e} \phi_i \rho \phi_j \, \mathrm{dV}, \tag{13}$$

$$\mathbf{D}_{ij} = -\int_{\Omega_e} \psi_j \cdot \nabla\phi_i \ \mathrm{dV} + \int_{\partial\Omega_e} (\phi_i \cdot \mathbb{T}|_{\partial\Omega_e}) \cdot \mathbf{n}_e \ \mathrm{dS} + \int_{\partial\Omega} (\phi_i \cdot \mathbb{T}) \cdot \mathbf{n} \ \mathrm{ds} \tag{14}$$

The coefficients of the discrete representation of $\mathbf{u}$ must be solved for using a numerical solution method. A similar procedure can be performed on the stress equation (5).

2.4 Temporal discretisation

Following the work by Glinsky et al. (2010), a fourth-order explicit leap-frog scheme is used to treat the time derivatives in (4) and (5). This is based on a truncated Taylor series expansion of the velocity and stress fields whilst staggering their solutions by half a time unit. Hence, the velocity is first solved to obtain a solution at time level $n+1$ using information about the stress at time $n+\frac{1}{2}$. This new velocity solution is then in turn used to solve for the stress field at time $n+\frac{3}{2}$. Mathematically, this is written as

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t \frac{\partial \mathbf{u}^{n+\frac{1}{2}}}{\partial t} + \frac{\Delta t^3}{24} \frac{\partial^3 \mathbf{u}^{n+\frac{1}{2}}}{\partial t^3} \tag{15}$$

$$\mathbb{T}^{n+\frac{3}{2}} = \mathbb{T}^{n+\frac{1}{2}} + \Delta t \frac{\partial \mathbb{T}^{n+1}}{\partial t} + \frac{\Delta t^3}{24} \frac{\partial^3 \mathbb{T}^{n+1}}{\partial t^3} \tag{16}$$

where the higher-order terms ($O(\Delta t^5)$) have been neglected. The remaining time derivatives can be evaluated using auxiliary fields which need to be solved for at each time-step:

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t \mathbf{u}_{\bigstar}^{n+\frac{1}{2}} + \frac{\Delta t^3}{24} \mathbf{u}_{\bigstar\bigstar}^{n+\frac{1}{2}} \tag{17}$$

$$\mathbb{T}^{n+\frac{3}{2}} = \mathbb{T}^{n+\frac{1}{2}} + \Delta t \mathbb{T}_{\bigstar}^{n+1} + \frac{\Delta t^3}{24} \mathbb{T}_{\bigstar\bigstar}^{n+1} \tag{18}$$

These auxiliary fields are defined as

$$\mathbf{u}_{\bigstar}^{n+\frac{1}{2}} = f(\mathbb{T}^{n+\frac{1}{2}}) \tag{19}$$

$$\mathbb{T}_{\bullet}^{n+\frac{1}{2}} = g(\mathbf{u}_{\bigstar}^{n+\frac{1}{2}}) \tag{20}$$

$$\mathbf{u}_{\bigstar\bigstar}^{n+\frac{1}{2}} = f(\mathbb{T}_{\bullet}^{n+\frac{1}{2}}) \tag{21}$$

and

$$\mathbb{T}_{\bigstar}^{n+1} = g(\mathbf{u}^{n+1}) \tag{22}$$

$$\mathbf{u}_{\bullet}^{n+1} = f(\mathbb{T}_{\bigstar}^{n+1}) \tag{23}$$

$$\mathbb{T}_{\bigstar\bigstar}^{n+1} = g(\mathbf{u}_{\bullet}^{n+1}) \tag{24}$$

where f and g are the right-hand sides of (4) and (5), respectively. When coupled with the fluxes across discontinuous mesh elements described in Section 2.3, such a discretisation scheme can be proven to have the desirable property of being non-dissipative (Delcourte et al., 2009).

3 Automated code generation

* Introduce UFL, the general Firedrake workflow (User writes UFL -> COFFEE compiler + optimisations -> AST -> FFC -> C kernels -> PyOP2 parallel execution) * Talk about future-proofing for the user code.

The treatment of solution discontinuities is accomplished using the `jump` and `avg` UFL operators, the latter of which is particularly important when implementing the upwinding method to take the average value of the solution field across interior element facets.

4 Verification and validation

In order to check that the model is solving the governing equations correctly, and to evaluate the model's ability to represent the physics behind seismological problems well, three 2-D test cases described by Glinsky et al. (2010) were considered.

4.1 Eigenmode convergence analysis

A convergence analysis was used to verify the correctness of the model's implementation, and to demonstrate that high-order convergence can be attained with the solution method. A time and space-dependent solution involving the propagation of an eigenmode was considered. Four different two-dimensional, structured, unit meshes of increasing resolution were used to resolve the solution ($\Delta x = 0.25, 0.125, 0.0625, 0.03125$). For each mesh, piecewise-discontinuous polynomial basis functions of degree 1, 2, 3 and 4 (also known as P1, P2, P3 and P4 basis functions) were employed, yielding a total of 16 simulations.

The initial conditions for velocity and stress were chosen to agree with the known exact solution at time $t = 0$ for velocity and $t = \Delta t/2$ for stress

$$\mathbf{u}_x = a\cos(\pi x)\sin(\pi y)\cos(at) \tag{25}$$
$$\mathbf{u}_y = -a\sin(\pi x)\cos(\pi y)\cos(at) \tag{26}$$
$$\mathbb{T}_{xx} = -b\sin(\pi x)\sin(\pi y)\sin(at) \tag{27}$$
$$\mathbb{T}_{xy} = \mathbb{T}_{yx} = 0 \tag{28}$$
$$\mathbb{T}_{yy} = b\sin(\pi x)\sin(\pi y)\sin(at) \tag{29}$$

where $a = \sqrt{2}\pi V_S$ and $b = 2\pi\mu$ (Glinsky et al., 2010). The physical properties were set to $\rho = 1.0$, $\mu = 0.25$ and $\lambda = 0.5$, and a free-surface boundary condition was weakly applied along all boundaries of the domain. Note that these quantities are arbitrary and have no units in this particular setup.

The time-step Δt was chosen such that a constant upper bound on the Courant number of 0.5 was enforced. Glinsky et al. (2010) found that the maximum Courant number required for stability was dependent on the degree of the polynomial approximation; each time the degree d was increased by unity, the bound on the Courant number was therefore halved such that

$$\Delta t = \frac{0.5\Delta x}{V_P 2^{d-1}} \tag{30}$$

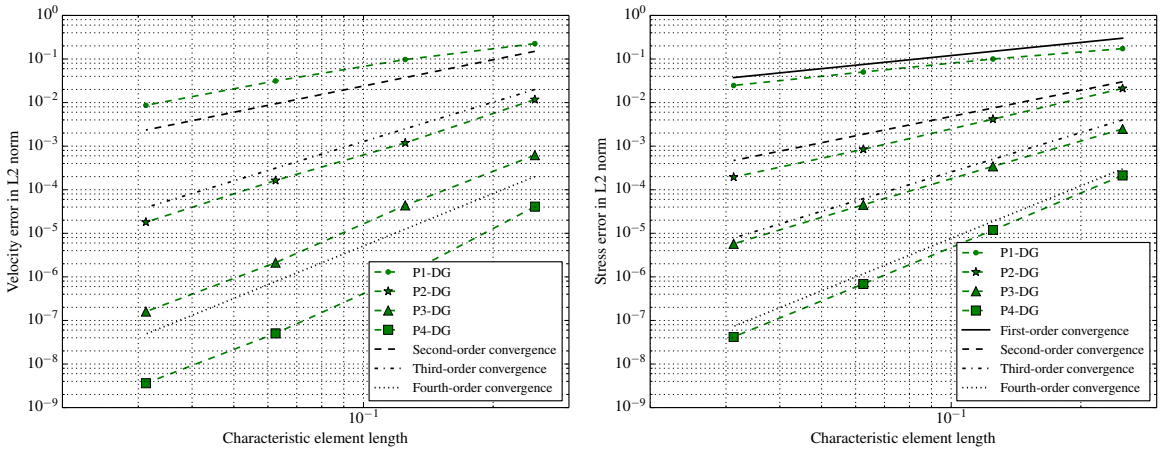

Figure 1: The L2 error of the velocity (left) and stress (right) fields.

where V_P is the velocity of the P-wave. All simulations were run until $T = 5.0$, after which the error between the numerical and exact solution was calculated in the L2 (i.e. Euclidean) norm.

The results in Figure 1 show that for piecewise-discontinuous polynomial approximations of degree d for the velocity field, $O(d+1)$ convergence is attained, except for the case when $d > 3$ where the convergence order is limited to fourth-order by the choice of time-stepping scheme. In the case of the stress field, only $O(d)$ convergence is attained because of the presence of the velocity gradient. The convergence of the fields at the expected order provides confidence in the correctness of the model.

4.2 Pulse propagation

This test case considered the propagation of a smooth wave across a 2D rectangular domain, defined by $0 \leq x \leq 4$ m and $0 \leq y \leq 1$ m. Not only does this allow one to check that the wave is propagating at the expected velocity, it also demonstrates that the wave does not dissipate over time as a result of the non-dissipative nature of the numerical scheme.

The initial condition for the x-component of velocity and xx-component of stress are given by

$$\mathbf{u}_x = e^{-50(x-1)^2} \tag{31}$$

$$\mathbb{T}_{xx} = -e^{-50(x-1)^2} \tag{32}$$

An absorption region was applied at both ends of the domain ($0 \leq x \leq 0.5$ m and $3.5 \leq x \leq 4.0$ m), with an absorption coefficient of $\sigma = 100$, to model an infinite domain and to prevent any spurious reflections off the end boundaries. Note that the solution does not depend on the y direction, and as such, $\mathbf{u}_y$, $\mathbb{T}_{xy}$ or $\mathbb{T}_{yy}$ were not solved for here.

The physical parameters were set to $\mu = 0.5$ Pa, $\lambda = 0.25$ Pa and $\rho = 1.0$ kgm^{-3}. For discretisation purposes, a structured mesh comprising triangular elements of characteristic length $\Delta x = 0.01$ m was used. The time-step Δt of 0.0025 s was set, corresponding to a maximum Courant number of 0.25, and the solutions were advanced forward in time until $T = 2.0$ s. Both the velocity and stress fields were represented by piecewise-discontinuous linear (also known as P1-DG) polynomials, since this simulation did not require such a high degree of accuracy to demonstrate the robustness of the numerical scheme.

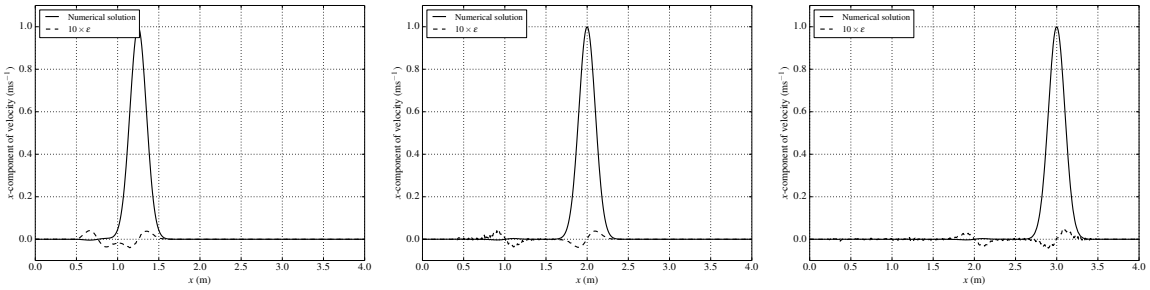

Figure 2: The x-component of the velocity field at $t = 0.25$, 1.0 and 2.0 (from left to right). The pulse wave travels at the correct velocity of 1 ms^{-1}. The difference between the analytical and numerical solution, ϵ, is scaled by a factor of 10 to emphasise the insignificant dissipation over time.

The results in Figure 2 show how the pulse propagates from left-to-right at a speed of 1 ms^{-1}, whilst displaying excellent agreement with the analytical solution derived by Delcourte et al. (2009).

4.3 Explosive source

The final test case considers the propagation of waves in a 2D rectangular domain following an explosion at a point source, based on the work by Garvin (1956). The dimensions of the domain are $0 \leq x \leq 300$ m and $0 \leq y \leq 150$ m. The mesh is unstructured and comprises triangular elements with a characteristic element length of 2.5 m. Fourth-order piecewise-discontinuous (P4) polynomial basis functions are used to approximate the velocity and stress fields.

A zero initial condition is used for both velocity and stress. A free-surface boundary condition is applied along the top of the 2D domain, and a 5 m-thick sponge region adjacent to the other boundaries is used to absorb any waves, thereby simulating an infinite domain.

A time-dependent source term is used to produce a Ricker wavelet (Ricker, 1953) to simulate an explosion, defined by

$$s(t) = \left(-1 + 2a(t - 0.3)^2\right) e^{\left(-a(t-0.3)^2\right)} \tag{33}$$

where $a = 159.42$ here. This is applied to the xx and yy-components of the stress tensor on the right-hand side of (5).

The simulation was run with a time-step of 0.001 s until $T = 2.5$ s. The physical parameters were set to $\rho = 1$ kgm^{-3}, $\mu = 3{,}600$ Pa, and $\lambda = 3{,}599.3664$ Pa.

Three sensors (labelled C1, C2 and C3) were placed in the domain to measure the vertical velocity throughout time. These were placed 1 m away from the top boundary of the domain, and at $x = 45$, 90 and 140 m, respectively. The results from the sensors presented in Figure 3 are in strong agreement with the reference solution provided by Glinsky et al. (2010), and represent a firm first-step towards the validation of the numerical model.

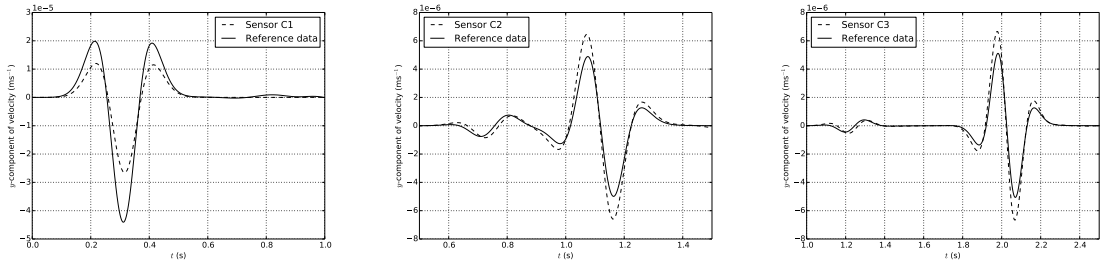

Figure 3: The y-component of the velocity field at sensors C1, C2 and C3 (from left to right). The reference solution is from the work by Glinsky et al. (2010).

5 Performance results

Michael and Fabio's section(s) here.

6 Conclusion

This paper has introduced a new seismological model which features novel code generation technology. The weak formulation of the elastic wave equation was written in just a few lines of high-level UFL, demonstrating the level of abstraction that the Firedrake framework offers to model developers and domain specialists. Verification and validation of the model demonstrated successful solution convergence, albeit up to fourth-order because of the time-stepping scheme employed. Future work will focus on even higher order schemes such as ADER (Käser and Dumbser, 2006; Dumbser and Käser, 2006) and spectral element methods (**?**).

7 Acknowledgements

This work is part of the Open Performance portablE SeismiC Imaging (OPESCI) project, funded by the Intel® Parallel Computing Center at Imperial College London and SENAI CIMATEC. Special thanks go to Nathalie Glinsky for providing the reference data presented by Glinsky et al. (2010), and permission to use it in Figure 3.

References

M. S. Alnæs, A. Logg, K. B. Ølgaard, M. E. Rognes, and G. N. Wells. Unified Form Language: A domain-specific language for weak formulations of partial differential equations. *ACM Transactions on Mathematical Software*, 40(2), 2014.

J. de la Puente, M. Dumbser, M. Käser, and H. Igel. Discontinuous Galerkin methods for wave propagation in poroelastic media. *Geophysics*, 73(5):T77–T97, 2008. doi: 10.1190/1.2965027.

J. de la Puente, J.-P. Ampuero, and M. Käser. Dynamic rupture modeling on unstructured meshes using a discontinuous Galerkin method. *Journal of Geophysical Research*, 114(B10), 2009. doi: 10.1029/2008JB006271.

S. Delcourte, L. Fezoui, and N. Glinsky-Olivier. A high-order Discontinuous Galerkin method for the seismic wave propagation. In *ESAIM: Proceedings*, volume 27, pages 70–89, 2009.

Z. DeVito, N. Joubert, F. Palacios, S. Oakley, M. Medina, M. Barrientos, E. Elsen, F. Ham, A. Aiken, K. Duraisamy, E. Darve, J. Alonso, and P. Hanrahan. Liszt: A Domain Specific Language for Building Portable Mesh-based PDE Solvers. In *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '11, pages 9:1–9:12, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0771-0. doi: 10.1145/2063384.2063396. URL `http://doi.acm.org/10.1145/2063384.2063396`.

M. Dumbser and M. Käser. An arbitrary high-order discontinuous Galerkin method for elastic waves on unstructured meshes – II. The three-dimensional isotropic case. *Geophysical Journal International*, 167(1):319–336, 2006. doi: 10.1111/j.1365-246X.2006.03120.x.

H. C. Elman, D. J. Silvester, and A. J. Wathen. *Finite Elements and Fast Iterative Solvers: with applications in incompressible fluid dynamics*. Oxford University Press, 2005.

W. W. Garvin. Exact Transient Solution of the Buried Line Source Problem. *Proceedings of the Royal Society of London, Series A*, 234(1199), 1956. doi: 10.1098/rspa.1956.0055.

N. Glinsky, S. Moto Mpong, and S. Delcourte. A High-order Discontinuous Galerkin Scheme for Elastic Wave Propagation. Technical Report inria-00543664, INRIA, 2010.

Robert W. Graves. Simulating seismic wave propagation in 3D elastic media using staggered-grid finite differences. *Bulletin of the Seismological Society of America*, 86(4):1091–1106, 1996.

V. Hermann, M. Käser, and C. E. Castro. Non-conforming hybrid meshes for efficient 2-D wave propagation using the Discontinuous Galerkin Method. *Geophysical Journal International*, 184(2):746–758, 2011. doi: 10.1111/j.1365-246X.2010.04858.x.

M. Käser and M. Dumbser. An arbitrary high-order discontinuous Galerkin method for elastic waves on unstructured meshes – I. The two-dimensional isotropic case with external source terms. *Geophysical Journal International*, 166(2):855–877, 2006. doi: 10.1111/j.1365-246X.2006.03051.x.

K. R. Kelly, R. W. Ward, Sven Treitel, and R. M. Alford. Synthetic seismograms: a finite-difference approach. *Geophysics*, 41(1):2–27, 1976. doi: 10.1190/1.1440605.

R. C. Kirby. Algorithm 839: FIAT, a New Paradigm for Computing Finite Element Basis Functions. *ACM Transactions on Mathematical Software*, 30(4):502–516, 2004. doi: 10.1145/1039813.1039820.

D. Komatitsch and J. Tromp. A perfectly matched layer absorbing boundary condition for the second-order seismic wave equation. *Geophysical Journal International*, 154(1):146–153, 2003. doi: 10.1046/j.1365-246X.2003.01950.x.

Y. Liu, J. Teng, H. Lan, X. Si, and X. Ma. A comparative study of finite element and spectral element methods in seismic wavefield modeling. *Geophysics*, 79(2):T91–T104, 2014. doi: 10.1190/geo2013-0018.1.

A. Logg and G. N. Wells. DOLFIN: Automated finite element computing. *ACM Transactions on Mathematical Software*, 37(2), 2010. doi: 10.1145/1731022.1731030.

A. Logg, K.-A. Mardal, G. N. Wells, et al. *Automated Solution of Differential Equations by the Finite Element Method*. Springer, 2012. doi: 10.1007/978-3-642-23099-8.

F. Luporini, A. L. Varbanescu, F. Rathgeber, G.-T. Bercea, J. Ramanujam, D. A. Ham, and P. H. J. Kelly. Cross-Loop Optimization of Arithmetic Intensity for Finite Element Local Assembly. *ACM Transactions on Architecture and Code Optimization*, 11(4), 2015. doi: 10.1145/2687415.

Graham R. Markall, Florian Rathgeber, Lawrence Mitchell, Nicolas Loriant, Carlo Bertolli, David A. Ham, and Paul H.J. Kelly. Performance-Portable Finite Element Assembly Using PyOP2 and FEniCS. In *28th International Supercomputing Conference, ISC, Proceedings*, volume 7905 of *Lecture Notes in Computer Science*, pages 279–289. Springer, 2013.

E. D. Mercerat and N. Glinsky. A nodal high-order discontinuous Galerkin method for elastic wave propagation in arbitrary heterogeneous media. *Geophysical Journal International*, 201 (2):1101–1118, 2015. doi: 10.1093/gji/ggv029.

F. Rathgeber, D. A. Ham, L. Mitchell, M. Lange, F. Luporini, A. T. T. McRae, G.-T. Bercea, G. R. Markall, and P. H. J. Kelly. Firedrake: automating the finite element method by composing abstractions. *ACM Transactions on Mathematical Software*, Submitted. URL `http://arxiv.org/abs/1501.01809`.

Florian Rathgeber, Graham R. Markall, Lawrence Mitchell, Nicolas Loriant, David A. Ham, Carlo Bertolli, and Paul H.J. Kelly. PyOP2: A High-Level Framework for Performance-Portable Simulations on Unstructured Meshes. In *High Performance Computing, Networking Storage and Analysis, SC Companion*, pages 1116–1123. IEEE Computer Society, 2012.

N. Ricker. The form and laws of propagation of seismic wavelets. *Geophysics*, 18(1):10–40, 1953. doi: 10.1190/1.1437843.

J. Tago, L. Métivier, and J. Virieux. SMART layers: a simple and robust alternative to PML approaches for elastodynamics. *Geophysical Journal International*, 199(2):700–706, 2014. doi: 10.1093/gji/ggu298.

Y. Tang, R. A. Chowdhury, B. C. Kuszmaul, C.-K. Luk, and C. E. Leiserson. The Pochoir Stencil Compiler. In *Proceedings of the Twenty-third Annual ACM Symposium on Parallelism in Algorithms and Architectures*, SPAA '11, pages 117–128, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0743-7. doi: 10.1145/1989493.1989508.

J. Virieux. P-SV wave propagation in heterogeneous media: Velocity-stress finite-difference method. *Geophysics*, 51(4):889–901, 1986. doi: 10.1190/1.1442147.

R. M. Weiss and J. Shragge. Solving 3D Anisotropic Elastic Wave Equations on Parallel GPU Devices. *Geophysics*, 78(2):F7–F15, 2013. doi: 10.1190/geo2012-0063.1.

S. Wenk, C. Pelties, H. Igel, and M. Käser. Regional wave propagation using the discontinuous Galerkin method. *Solid Earth*, 4(1):43–57, 2013. doi: 10.5194/se-4-43-2013.