

# An efficient ADER-DG local time stepping scheme for 3D HPC simulation of seismic waves in poroelastic media

Sebastian Wolf<sup>a,\*</sup>, Martin Galis<sup>b,c</sup>, Carsten Uphoff<sup>d</sup>, Alice-Agnes Gabriel<sup>d</sup>, Peter Moczo<sup>b,c</sup>, David Gregor<sup>b</sup>, Michael Bader<sup>a</sup>

<sup>a</sup> Department of Informatics, Technical University of Munich, Boltzmannstr. 3, 85748, Garching, Germany

<sup>b</sup> Faculty of Mathematics, Physics and Informatics, Comenius University in Bratislava, Mlynska dolina F1, 842 48, Bratislava, Slovakia

<sup>c</sup> Earth Science Institute, Slovak Academy of Sciences, Dubravská cesta 9, 840 05, Bratislava, Slovakia

<sup>d</sup> Department of Earth and Environmental Sciences, Ludwig-Maximilians-Universität München, Theresienstr. 41, 80333, Munich, Germany

## ARTICLE INFO

### Article history:

Received 23 August 2021

Received in revised form 1 December 2021

Accepted 2 December 2021

Available online 7 December 2021

Dataset link: <https://github.com/SeisSol/SeisSol>

Dataset link: <https://doi.org/10.5281/zenodo.5236133>

### Keywords:

Poroelasticity

Discontinuous Galerkin

Wave propagation

High Performance Computing (HPC)

Computational seismology

ADER-DG

## ABSTRACT

Many applications from the fields of seismology and geoenvironment require simulations of seismic waves in porous media. Biot's theory of poroelasticity describes the coupling between solid and fluid phases and introduces a stiff reactive source term (Darcy's Law) into the elastodynamic wave equations, thereby increasing computational cost of respective numerical solvers and motivating efficient methods utilising High-Performance Computing. We present a novel realisation of the discontinuous Galerkin scheme with Arbitrary High-Order DERivative time stepping (ADER-DG) that copes with stiff source terms. To integrate this source term with a reasonable time step size, we utilise an element-local space-time predictor, which needs to solve medium-sized linear systems – each with 1,000 to 10,000 unknowns – in each element update (i.e., billions of times). We present a novel block-wise back-substitution algorithm for solving these systems efficiently, thus enabling large-scale 3D simulations. In comparison to LU decomposition, we reduce the number of floating-point operations by a factor of up to 25, when using polynomials of degree 6. The block-wise back-substitution is mapped to a sequence of small matrix-matrix multiplications, for which code generators are available to generate highly optimised code.

We verify the new solver thoroughly against analytical and semi-analytical reference solutions in problems of increasing complexity. We demonstrate high-order convergence of the scheme for 3D problems. We verify the correct treatment of point sources and boundary conditions, including homogeneous and heterogeneous full space problems as well as problems with traction-free boundary conditions. In addition, we compare against a finite difference solution for a newly defined 3D layer over half-space problem containing an internal material interface and free surface. We find that extremely high accuracy is required to accurately resolve the slow, diffusive P-wave at a or near a free surface, while we also demonstrate that solid particle velocities are not affected by coarser resolutions. We demonstrate that by using a clustered local time stepping scheme, time to solution is reduced by a factor of 6 to 10 compared to global time stepping. We conclude our study with a scaling and performance analysis on the SuperMUC-NG supercomputer, demonstrating our implementation's high computational efficiency and its potential for extreme-scale simulations.

© 2021 Elsevier Inc. All rights reserved.

\* Corresponding author.

E-mail addresses: [wolf.sebastian@in.tum.de](mailto:wolf.sebastian@in.tum.de) (S. Wolf), [martin.galis@uniba.sk](mailto:martin.galis@uniba.sk) (M. Galis), [uphoff@geophysik.uni-muenchen.de](mailto:uphoff@geophysik.uni-muenchen.de) (C. Uphoff).

## 1. Introduction

Elastodynamic wave propagation in fluid-saturated porous rocks is a relevant and challenging topic in computational seismology. Specifically in the contexts of seismic exploration, monitoring of geological reservoirs and human-induced earthquakes, it is important to study the interaction between waves, fluids and solids in the subsurface. Applications from exploration geophysics to earthquake engineering require high-resolution 3D forward simulations of seismic wave propagation in porous media. To better understand which information seismic waves carry about the porosity, permeability and fluid-saturation of rocks, forward simulations of seismic waves propagating in poroelastic materials are required (e.g. [1]).

Poroelastic materials consist of a solid matrix with pores that are completely fluid-filled. Biot's theory of poroelasticity [2–5] describes the interaction between the fluid and the solid phase and is widely accepted and validated [6–8]. The resulting system of partial differential equations (PDEs) describes seismic wave propagation in porous media, extending the elastic model often used in computational seismology by additional quantities (e.g., fluid velocities) and, in particular, by a stiff reactive source term that is required to model viscosity effects of the fluid–solid interaction. For the numerical solution of the governing equations, this stiff source term is typically a key computational challenge (see Section 2).

In this study, we focus on the Discontinuous Galerkin method with Arbitrary DERivative time stepping (ADER-DG) for poroelastic materials, as introduced by de la Puente et al. [9]. The DG method combines advantages from finite volume and finite element methods [10–13]. DG schemes by design lead to strongly local data access patterns to advance one element in time – only the information from this particular element and its neighbours is needed. Therefore the DG method can be easily parallelised to be used on modern supercomputers [14–17]. Combined with ADER time stepping, we achieve the same high-order convergence in time as in space and can exploit local time stepping [18]. We extend SeisSol,<sup>1</sup> an open-source software for modelling seismic wave propagation and earthquake source dynamics, which relies on the ADER-DG method. SeisSol supports elastic, viscoelastic and anisotropic materials and regularised, non-associated Drucker–Prager plastic deformation [13,18–21]. It makes use of unstructured tetrahedral meshes to easily incorporate topography and complex material discontinuities. SeisSol also allows modelling nonlinear rupture dynamics of earthquake sources [22–25]. SeisSol is optimised for the latest CPU [16,17] and GPU [26] based supercomputers.

Here, we demonstrate that by using the space-time predictor variant of the ADER time stepping scheme [27], which makes the ADER-DG scheme locally implicit, the stiff source term can be integrated without a strict time step restriction. The solution procedure stays overall explicit, which means that no global system has to be assembled and solved. As a key part of the solution procedure, a medium-sized linear system with a few thousand unknowns has to be solved for each element and time step. As extreme-scale simulations may calculate more than  $10^{13}$  element updates [17], a highly efficient solver for these systems is needed. We show that with a standard approach, such as the LU decomposition [28], it is not feasible to tackle large-scale poroelastic problems due to the high demands on computational power and memory. We exploit the structure of the system to derive an efficient back-substitution algorithm. When we compare our algorithm to a standard LU-decomposition, we see a reduction in the number of floating-point operations by a factor of up to 25, when using polynomials of degree 6 and we expect that the speed-up factor further increases with higher degrees. In addition, the back-substitution procedure can be expressed as a chain of small matrix-matrix multiplications (GEMMs, [29]). For these operations, efficient implementations exist (e.g. [30]), thus high computational efficiency can be readily achieved.

In the following, we review existing approaches to solve the governing equations of poroelasticity in Section 2. Then, in Section 3, we summarise the governing equations of poroelasticity. Subsequently, we outline the spatial and temporal discretisation using the space-time variant of ADER-DG in Section 4. In Section 5, we present our novel solution algorithm. The new scheme is then compared to reference solutions in a series of verification exercises, in Section 6. We demonstrate the high-order convergence of our method with canonical 3D models of planar wave propagation. We verify the accurate treatment of wave propagation in a homogeneous full-space excited by an explosive point source, free surfaces and internal material interfaces in comparison to analytical and semi-analytical reference solutions. Finally, we present a new 3D poroelastic layer over half-space scenario, in which we verify all implementation aspects conjunctively in comparison to a finite difference method. In Section 7, we examine the performance and scalability of our implementation. We conclude with a discussion of our results in Section 8.

## 2. Related work

The PDEs describing wave propagation in poroelastic media are interesting from a mathematical and computational point of view because they contain a stiff source term. Several different approaches to solve these equations have been proposed. When simulating wave propagation in poroelastic media, the main challenge comes from the viscous coupling between the solid and the fluid, which introduces a stiff reactive source term to the equation. Here, we shortly summarise various semi-analytical and numerical methods and approaches for the simulation of seismic wave propagation in a poroelastic medium.

For simple models, Green's function approaches are applicable to solve the equations of motion combined with Biot's constitutive law for poroelastic media. A Green's function approach can be considered semi-analytical: The solution is

<sup>1</sup> <https://www.seissol.org>.

expressed analytically as a convolution of a Green's function with a source time function. To actually compute the solution at a given point, numerical quadrature is typically used. Diaz and Ezziani [31] solve the PDEs for the solid particle velocities of a poroelastic material filled with an inviscid fluid. They consider a homogeneous full-space, a contact of two half-spaces with distinct material or a half-space with a free surface. Karpfinger et al. [32] consider general moment-tensor sources (monopole, dipole and double-couple), and solve for the solid particle velocities as well as the relative fluid velocities in homogeneous full-spaces. They take inviscid and viscous fluids in the pore space into account. While both approaches give good results for geometrically simple test cases, they are not applicable for more complicated setups. We are not aware of analytical Green's functions that can readily account for non-planar topography or 3D heterogeneous poroelastic materials.

Carcione et al. [1] summarise the numerical methods used for wave propagation in poroelastic media. In contrast to the Green's function approaches, the PDEs are discretised and the system of discretised equations is then solved. Carcione and Quiroga-Goode [33] introduce a splitting method in time in combination with a pseudospectral element discretisation in space to overcome stability problems. Morency and Tromp [34] apply the spectral element method to solve the equations of poroelasticity for 2D applications. There, the governing equations are written down in second-order form, with the solid displacements and the relative fluid displacements as principal quantities. After the spatial discretisation, the time stepping is done using a Newmark scheme.

The finite-difference (FD) method has been applied to wave propagation in poroelastic media since the early 70s. One of the first studies was published by Garg et al. [35]. Since then, many authors have applied a variety of FD schemes to model seismic waves and diffusion in poroelastic media. A detailed overview is given by Moczo et al. [36] and Gregor et al. [37,38]. The three papers introduced the staggered-grid velocity–stress–pressure FD scheme with a sub-cell resolution in poroelastic media with zero, nonzero constant and frequency-dependent resistive friction.

Also, the DG framework has been used to solve the governing equations of poroelasticity [9,39–41]. de la Puente et al. [9] combine ADER time stepping with the DG method using modal basis functions (using a deprecated version of SeisSol). They compare a splitting approach with a space-time predictor to integrate the stiff source term. The DG method can also be combined with implicit-explicit (IMEX) Runge-Kutta time stepping – Dudley Ward et al. [42] focus mostly on the so-called high-frequency case. Shukla et al. [40] use operator splitting with nodal basis functions and Runge-Kutta time stepping. Zhang et al. [39] use ADER time stepping similar to SeisSol and focus on the coupling between wave propagation in elastic and poroelastic materials. Zhan et al. [41] also combine the DG method with Runge Kutta time stepping. However, they omit the stiff source term, by only considering inviscid fluids. Most of the approaches are presented for 2D scenarios; only a few [9,41] are able to solve more realistic 3D problems.

### 3. Governing equations

In the following, we outline the governing equations for wave propagation in poroelastic materials. The final PDE system combines 13 unknowns – six stress components, three solid particle velocities, the pore pressure and three relative fluid velocities – in the vector of unknowns:

$$q = (\sigma_{xx}, \sigma_{yy}, \sigma_{zz}, \sigma_{xy}, \sigma_{yz}, \sigma_{xz}, u, v, w, p, u_f, v_f, w_f)^T.$$

Following [9,43,8], the governing equation for wave propagation in a poroelastic medium can be written in the matrix-vector form as:

$$\frac{\partial q}{\partial t} + A \frac{\partial q}{\partial x} + B \frac{\partial q}{\partial y} + C \frac{\partial q}{\partial z} = Eq, \quad (1)$$

with matrices  $A, B, C, E \in \mathbb{R}^{13 \times 13}$ . The term  $Eq$  on the right-hand side of Equation (1) is denoted as reactive source term or viscous dissipation term. It accounts for the dissipation of energy due to the motion of the viscous fluid relative to the solid. The governing equations of poroelasticity form a linear hyperbolic partial differential equation with a stiff source term.

#### 3.1. Constitutive behaviour

To better understand the mechanics of this PDE system and to introduce all involved quantities and parameters, we summarise its underlying physical concepts and derivation. Poroelastic materials are inherently heterogeneous combining a solid and a fluid phase. They consist of a solid matrix (also called solid frame) with pores. The pore space is then completely filled by a fluid. The material of the solid phase is characterised by the bulk modulus  $K_S$  and the density  $\rho_S$ . The porosity  $\phi$  describes the volume fraction occupied by the pores. The solid matrix, including empty pores, behaves like an elastic body, i.e. its rheology can be characterised by the two Lamé parameters  $\lambda_M$  and  $\mu_M$ . The fluid phase is described by its bulk modulus  $K_F$ , density  $\rho_F$  and the viscosity  $\nu$ . We also need two parameters, which describe how the solid matrix and the fluid interact: A fluid particle moving from one point in the solid matrix to another cannot follow the direct path, but has to follow the path dictated by the pores. The tortuosity  $T$  describes how much longer this path is compared to the direct connection. The permeability  $\kappa$  is a measure of how well fluids can be transported through the pores. All poroelastic material parameters are summarised in Table 1.



**Table 1**  
Material parameters used to characterise poroelastic materials.

Parameter	Symbol	Unit
Solid Bulk modulus	$K_S$	Pa
Solid density	$\rho_S$	$\text{kg m}^{-3}$
Matrix 1st Lamé parameter	$\lambda_M$	Pa
Matrix 2nd Lamé parameter	$\mu_M$	Pa
Matrix permeability	$\kappa$	$\text{m}^2$
Matrix porosity	$\phi$	
Matrix tortuosity	$T$	
Fluid bulk modulus	$K_F$	Pa
Fluid density	$\rho_F$	$\text{kg m}^{-3}$
Fluid viscosity	$\nu$	Pa s

It is now possible to define the displacement of the solid matrix  $U_i^M$  and corresponding strain tensor  $\epsilon_{ij}^M = \frac{1}{2}(\partial_i U_j^M + \partial_j U_i^M)$ . In addition, we take the displacement of the fluid  $U^F$  and the pore pressure  $p$  into account and consider the variation of the fluid content as  $\zeta := -\nabla \cdot (\phi(U^F - U^M))$ . To derive a constitutive law of the poroelastic material, we consider a homogenised material, i.e. we neglect spatial scales smaller than a pore's diameter. For example, the effective density of the poroelastic material can be computed as  $\rho = \phi \cdot \rho_F + (1 - \phi) \cdot \rho_S$ . The total stress  $\sigma_{ij} = \sigma_{ij}^M - \phi p \delta_{ij}$  can be observed. We compute the bulk modulus of the solid matrix  $K_M = \lambda_M + \frac{2}{3}\mu_M$ . Finally, we consider the solid-fluid coupling modulus

$$M = \frac{K_S}{(1 - K_M/K_S) - \phi(1 - K_S/K_F)},$$

and the effective stress component

$$\alpha = 1 - K_M/K_S.$$

With all quantities defined, we can write down the constitutive law, relating  $\sigma$  and  $p$  to  $\epsilon^M$  and  $\zeta$  [43, sec. 2].

$$\begin{pmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \sigma_{yz} \\ \sigma_{xz} \\ \sigma_{xy} \\ -p \end{pmatrix} = \begin{pmatrix} \lambda_M + 2\mu_M + M\alpha^2 & \lambda + M\alpha^2 & \lambda + M\alpha^2 & 0 & 0 & 0 & M\alpha \\ \lambda + M\alpha^2 & \lambda_M + 2\mu_M + M\alpha^2 & \lambda + M\alpha^2 & 0 & 0 & 0 & M\alpha \\ \lambda + M\alpha^2 & \lambda + M\alpha^2 & \lambda_M + 2\mu_M + M\alpha^2 & 0 & 0 & 0 & M\alpha \\ 0 & 0 & 0 & \mu & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu & 0 \\ M\alpha & M\alpha & M\alpha & 0 & 0 & 0 & M \end{pmatrix} \begin{pmatrix} \epsilon_{xx}^M \\ \epsilon_{yy}^M \\ \epsilon_{zz}^M \\ \epsilon_{yz}^M \\ \epsilon_{xz}^M \\ \epsilon_{xy}^M \\ -\zeta \end{pmatrix}.$$

Note that by our sign convention, pressure is positive in compression while normal stresses are negative in compression.

### 3.2. Equations of motion

We now consider the time-dependent problem, in terms of the solid particle velocities  $v_i = \partial_t U_i^M$  and the relative fluid velocities  $v_i^F = \phi \partial_t (U_i^F - U_i^M)$ . The equations of motion can be combined with Darcy's law to obtain a system of PDEs [43, sec. 3]:

$$\begin{aligned} \sum_{j=1}^3 \frac{\partial \sigma_{ij}}{\partial x_j} &= \rho \frac{\partial v_i}{\partial t} + \rho_F \frac{\partial v_i^F}{\partial t} \\ -\frac{\partial p}{\partial x_i} &= \rho \frac{\partial v_i}{\partial t} + \frac{\rho^F T}{\phi} \frac{\partial v_i^F}{\partial t} + \frac{\nu}{\kappa} v_i^F. \end{aligned} \quad (2)$$

We note that Equation (2) is only valid for frequencies lower than Biot's characteristic frequency  $f_c = \frac{1}{2\pi} \frac{\nu \phi}{T \kappa \rho_F}$ . In this so-called low-frequency regime, the flow of the fluid in pore space can be approximated as a laminar flow. For higher frequencies, the fluid flow becomes turbulent and a more complex form of Darcy's law has to be considered. Biot's frequency depends mainly on the porosity  $\phi$  and the permeability  $\kappa$ . Within geo-reservoirs, permeability typically ranges from  $2 \cdot 10^{-4}$  mD to  $1 \cdot 10^2$  mD and porosity varies between 0% and 20% (e.g. [44, Fig. 14.1]), leading to Biot's frequency ranging from

<sup>2</sup> The unit Darcy (1 D =  $9.869 \cdot 10^{-13} \text{ m}^2$ ) is a unit for the permeability of a porous medium.



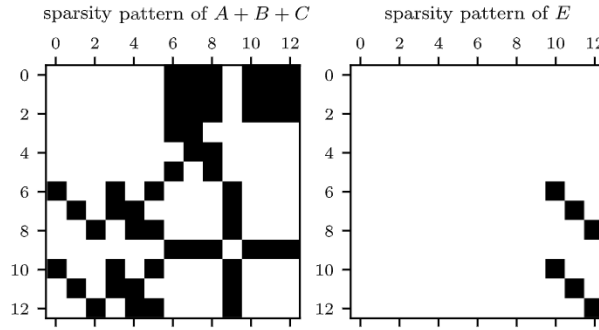


Fig. 1. Sparsity patterns of the matrices  $A + B + C$  and  $E$  of Equation (1).

tens of Hz to hundreds of kHz. We aim for simulations of seismic waves up to 10 Hz, which means that considering the low-frequency regime is sufficient (see also, e.g. [45, Fig 2.1]). If all pores are filled with an inviscid fluid, Biot's frequency is formally zero, implying the high-frequency regime. However, for the inviscid case, the equations for low- and high-frequency regimes are identical [9].

We follow [9] in combining the constitutive behaviour and the equations of motions to derive the governing equations as a hyperbolic system in first-order form in the form of Equation (1). We show the sparsity patterns of the matrices  $A + B + C$  and  $E$  in Fig. 1. A detailed description of these matrices can be found in [46, pp. 114–115], where also the anisotropic case is discussed. Here we restrict ourselves to the isotropic case. At this point, we want to remark that  $E$  is an upper triangular matrix i.e.  $E_{ij} = 0 \quad \forall i > j$ , which will be important in Section 5.

We can combine Equation (2) with initial conditions. Typically, in the initial state, the system is at rest, and seismic waves are excited by an external source term, which additionally enters Equation (2) on the right-hand side. See for example [32], for an overview of different source types and their respective radiation patterns in a homogeneous full-space.

#### 4. ADER-DG discretisation

In this section we summarise the numerical method we use to discretise Equation (1). We focus on the Discontinuous Galerkin (DG) method, which has been increasingly attractive for simulation of elastic wave propagation [47–56]. We combine the DG method with Arbitrary high-order DERivative (ADER) time stepping, leading to high-order accuracy in time within a single step [57]. In essence, ADER-DG is a predictor-corrector scheme: First, in each element we predict a solution solely based on the information within the element itself. In the second step, the predicted solution is corrected using numerical fluxes across element boundaries. The scheme is explicit in time, which is attractive from a computational perspective, since no global system of equations has to be assembled and solved [27]. ADER-DG has been successfully used for a broad range of problems, for example, shallow water equations, relativistic magnetohydrodynamics or the Euler equations [58]. In particular, ADER-DG is the basis of many seismological applications [13,19,16,24,25,58,59,17].

##### 4.1. Spatial discretisation

We use a DG approach on unstructured tetrahedral elements. First, we partition the computational domain  $\Omega \subset \mathbb{R}^3$  in a set of conforming tetrahedrons  $\mathcal{T} = \{T_i\}_{i=1}^n$ . Within each tetrahedron, we expand the solution  $q$  in space:  $q_p(x, t) = \hat{Q}_{pl}^n(t) \psi_l^n(x)$ , for  $x \in T_n$  using a set of basis functions  $\psi_l^n$ . Here we use Einstein sum convention, i.e. a sum over repeated indices is implied. The superscript  $n$  emphasises that  $\hat{Q}$  and  $\psi_l$  are specific to the element  $T_n$ . For better readability, however, we will omit that superscript, where it is clear from the context. We introduce the reference element  $\mathcal{E} = \{(x, y, z) \in \mathbb{R}^3 : x, y, z > 0 \wedge x + y + z < 1\}$ . Now, we can construct an affine linear mapping  $\Xi^n : T_n \rightarrow \mathcal{E}$ , which maps global coordinates to reference coordinates. Finally, we define the basis functions  $\psi_l^n(x) = \phi_l(\Xi^n(x))$ , using a set of polynomials  $\phi_l$  defined on the reference element. As  $\phi_l$ , we choose Dubiner polynomials, which are an orthogonal set of polynomials on tetrahedrons and are based on Jacobi polynomials [60]. Any polynomial  $p(x, y, z)$  in three spatial variables can be written as  $\sum_{i,j,k} \alpha_{ijk} x^i y^j z^k$ . We define the degree of a polynomial as  $\deg(p) = \max_{\alpha_{ijk} \neq 0} i + j + k$ . For example, the polynomial 1 has degree 0, the polynomial  $x^2 y z^2$  has degree 5 and so on. In three space dimensions the space of polynomials, which are exactly of degree  $N$ , can be represented by  $\binom{N+2}{2}$  basis functions and the space of polynomials of degree equal or less than  $N$  is spanned by  $\binom{N+3}{3}$  basis functions. The Dubiner polynomials  $\phi_i$  are ordered such that the degree of  $\phi_i$  is less or equal to the degree of  $\phi_j$  for  $i \leq j$ .

We multiply Equation (1) with a test function  $\psi_k^n$  and integrate in space to get a weak formulation of our equation:

$$\int_{T_n} \frac{\partial q}{\partial t} \psi_k^n dV + \int_{T_n} \left( A \frac{\partial q}{\partial x} + B \frac{\partial q}{\partial y} + C \frac{\partial q}{\partial z} \right) \psi_k^n dV = \int_{T_n} E q \psi_k^n dV \quad (3)$$

We transform the integrals to the reference element and perform integration by parts (cf. [19]). The Jacobians on the reference element can be obtained as  $A^* = A\partial_x\xi + B\partial_y\xi + C\partial_z\xi$ ,  $B^* = A\partial_x\eta + B\partial_y\eta + C\partial_z\eta$  and  $C^* = A\partial_x\zeta + B\partial_y\zeta + C\partial_z\zeta$ . Here,  $\xi, \eta, \zeta$  are the coordinate components of the transformation  $\Xi^n$ . To ease notation, we set  $\mathcal{A}^1 = A^*$ ,  $\mathcal{A}^2 = B^*$ ,  $\mathcal{A}^3 = C^*$  and express derivatives in  $\xi$  direction as  $\partial_1$ , derivatives in  $\eta$  direction as  $\partial_2$ , and derivatives in  $\zeta$  direction as  $\partial_3$ . Then we get

$$\begin{aligned} & \frac{\partial \hat{Q}_{pl}^n}{\partial t} |J| \int_{\mathcal{E}} \phi_l \phi_k dV + \sum_{j=1}^4 F_{pk}^j(\hat{Q}^n, \hat{Q}^{n_j}) \\ & - \sum_{j=1}^3 \mathcal{A}_{pq}^j \hat{Q}_{ql}^n |J| \int_{\mathcal{E}} \partial_j \phi_k \phi_l dV = E_{pq} \hat{Q}_{ql}^n |J| \int_{\mathcal{E}} \phi_k \phi_l dV. \end{aligned} \quad (4)$$

We assume that the Jacobian matrices  $\mathcal{A}$  are constant on each element. This has the advantage that the integrals can be precomputed and no quadrature is needed. Here, we introduced the numerical flux  $F_{pk}^j(\hat{Q}^n, \hat{Q}^{n_j})$ , which evaluates the exchange of quantities across the  $j^{\text{th}}$  face of the tetrahedron. As a numerical flux we choose Godunov's flux method based on the solution of exact Riemann problems at the inter-cell boundaries [61,62].

#### 4.2. Temporal discretisation with ADER

We add ADER time stepping [57]. We replace continuous time with a series of time steps  $0 = t_0, t_1, \dots$ , for simplicity we assume a regular grid in time:  $t_i = i \cdot \Delta t$ . The idea of ADER time stepping is the following: Given the local solution  $q_p(t_i, x) = \hat{Q}_{pl}^n(t_i) \psi_l(x)$ , we predict a solution for upcoming times  $t_i + \delta t < t_{i+1}$ . This is classically done using a Taylor series approach,

$$q_p(t_i + \delta t) = \sum_{j=1}^N \frac{\partial^j q_p}{\partial t_j}(t_i) \frac{\delta t^j}{j!}. \quad (5)$$

The temporal derivatives at  $t_i$  are computed from the spatial derivatives using the Cauchy-Kovalevski procedure [63], where temporal derivatives can be replaced by spatial derivatives. This Taylor series predicts the time evolution of  $q_p$  based on the local information. If we integrate Equation (4) in time over  $[t_i, t_{i+1}]$ , we get an expression to compute  $\hat{Q}_{pl}^n(t_{i+1})$  from  $\hat{Q}_{pl}^n(t_i)$ . The integration in time is done using Gaussian quadrature. This is where the predicted solutions for arbitrary times  $t \in [t_i, t_{i+1}]$  is needed. This corrector step takes the flux information from the neighbouring elements into account.

This scheme has several advantages: It is a one-step scheme, i.e. no intermediate stages (such as for example, with Runge-Kutta schemes) have to be stored. The scheme is explicit, in particular, no global system of equations has to be solved. As a direct consequence, the scheme can easily be parallelised, using mesh partitioning, where only ghost cells at partition boundaries have to be exchanged. If we choose polynomials up to degree  $N$ , we achieve a convergence rate  $N + 1$  in space and time. To ensure stability, a CFL condition in the form

$$\Delta t \leq C(N) \frac{d}{v}$$

has to be fulfilled for each mesh element, where  $d$  is the diameter of the element's insphere and  $v$  is the maximal wave speed [18]. The constant  $C$  is set to  $C(N) = \frac{c}{2N+1}$  with  $c = \frac{1}{2}$ . If we use global time stepping,  $\Delta t$  is set to the smallest fraction  $\min_{T_i \in \mathcal{T}} C(N) d(T_i) / v(T_i)$ , taking into account all elements. Material parameters, and thus wave speeds, change over the computational domain. In addition, meshes are typically refined locally. So with global time stepping we might impose time steps smaller than actually necessary. Local time stepping can be added to the ADER-DG scheme, to save computational load where possible [18]. As modern supercomputers work best on structured data, clustered local time stepping is used [64].

#### 4.3. Space-time predictor

The right hand side in Equation (1),  $Eq$ , poses a stiff source term. As a consequence, the Cauchy-Kovalevski procedure to predict a solution becomes unstable. In [9], new time stepping schemes that are stable with a stiff term are analysed – operator splitting and space-time predictor. Since the splitting scheme does not achieve high-order convergence, we focus on the space-time predictor. For a more general review of this algorithm, we refer to [27, sec. 3.3]. The idea is to express the solution in time using a polynomial expansion:  $q_p(x, t) = \tilde{Q}_{pls}^{ni} \psi_l^n(x) \theta_s^i(t)$  for  $x \in T_n, t \in [t_i, t_{i+1}]$ . Note that  $\tilde{Q}^{ni}$  is not time dependent anymore, but remains constant on the space-time element  $\mathcal{E}_n \times [t_i, t_{i+1}]$ . We define  $[0, 1]$  as a reference element in time. On this element, we choose Jacobi polynomials  $\chi_s$  as a basis and obtain  $\theta^i$  via a transformation onto the

reference element. We chose the same degree for the spatial and temporal basis functions. Now we multiply Equation (1) with spatial and temporal basis functions to derive the following system of equations for  $\tilde{Q}^{ni}$  [9]:

$$\begin{aligned} \int_{t_i}^{t_{i+1}} \int_{T_n} \tilde{Q}_{pls}^{ni} \psi_l \frac{\partial \theta_s}{\partial t} \psi_k \theta_r \, dV \, dt \\ + \int_{t_i}^{t_{i+1}} \int_{T_n} \left( A_{pq} \tilde{Q}_{qls}^{ni} \frac{\partial \psi_l}{\partial x} \theta_s + B_{pq} \tilde{Q}_{qls}^{ni} \frac{\partial \psi_l}{\partial y} \theta_s + C_{pq} \tilde{Q}_{qls}^{ni} \frac{\partial \psi_l}{\partial z} \theta_s \right) \psi_k \theta_r \, dV \, dt \\ = \int_{t_i}^{t_{i+1}} \int_{T_n} E_{pq} \tilde{Q}_{qls}^{ni} \psi_l \theta_s \psi_k \theta_r \, dV \, dt. \end{aligned}$$

We apply integration by parts in time to the first integral and map onto the reference element:

$$\begin{aligned} \delta_{pq} \langle \chi_r(1) \phi_k, \chi_s(1) \phi_l \rangle \tilde{Q}_{qls}^{ni} - \langle \chi_r(0) \phi_k, \phi_l \rangle \tilde{Q}_{pl}^{n,0} - \delta_{pq} \left[ \frac{\partial \chi_r}{\partial \tau} \phi_k, \chi_s \phi_l \right] \tilde{Q}_{qls}^{ni} \\ + \sum_{j=1}^3 \mathcal{A}_{pq}^{j*} [\chi_r \phi_k, \chi_s \partial_j \phi_l] \tilde{Q}_{qls}^{ni} = E_{pq}^* [\chi_r \phi_k, \chi_s \phi_l] \tilde{Q}_{qls}^{ni}. \quad (6) \end{aligned}$$

Here,  $\langle \cdot, \cdot \rangle$  is a scalar product in space, whereas  $[\cdot, \cdot]$  is a scalar product in space and time:

$$\begin{aligned} \langle f, g \rangle &:= \int_{\mathcal{E}} f(\xi, \eta, \zeta) g(\xi, \eta, \zeta) \, dV(\xi, \eta, \zeta) \\ [f, g] &:= \int_0^1 \int_{\mathcal{E}} f(\tau, \xi, \eta, \zeta) g(\tau, \xi, \eta, \zeta) \, dV(\xi, \eta, \zeta) \, d\tau. \end{aligned}$$

Note that we use  $Q_{pl}^{n,0}$ , which collects the spatial degrees only from the previous time step. We introduce the source matrix  $E^* = \Delta t E$  transformed to the reference element, and the Jacobian matrices  $\mathcal{A}^{j*} = \Delta t \mathcal{A}^j$ , now also scaled with the time step  $\Delta t$ . Notice, that we can decompose most of the inner products, e.g.

$$[\chi_r \phi_k, \chi_s \phi_l] = \int_0^1 \int_{\mathcal{E}} \chi_r \phi_k \chi_s \phi_l \, dV \, d\tau = \int_0^1 \chi_r \chi_s \left( \int_{\mathcal{E}} \phi_k \phi_l \, dV \right) d\tau = \int_0^1 \chi_r \chi_s \, d\tau \int_{\mathcal{E}} \phi_k \phi_l \, dV.$$

We define mass and stiffness matrices in time,

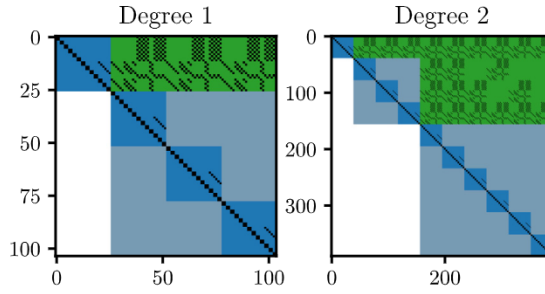
$$\begin{aligned} W_{rs} &= \chi_r(1) \chi_s(1) \\ w_r &= \chi_r(0) \\ S_{rs} &= \int_0^1 \chi_r \chi_s \, d\tau \\ K_{rs}^\tau &= \int_0^1 \frac{\partial \chi_r}{\partial \tau} \chi_s \, d\tau, \end{aligned}$$

and space,

$$\begin{aligned} M_{kl} &= \int_{\mathcal{E}} \phi_k \phi_l \, dV \\ K_{kl}^\alpha &= \int_{\mathcal{E}} \phi_k \frac{\partial \phi_l}{\partial \alpha} \, dV, \end{aligned}$$

where  $\alpha \in \{\xi, \eta, \zeta\}$  or 1, 2, 3. We start by inserting our matrix definitions into Equation (6) and drop the superscript  $ni$ :





**Fig. 2.** Sparsity patterns of the system matrices of Equation (8) with basis functions of maximum degree 1 and 2, respectively. The white blocks contain only zeros. We identify  $(N + 1)$  blue blocks of increasing size at the diagonal. We further distinguish these blocks in a light and a dark blue. The light-blue blocks also contain only zeros. The dark-blue blocks are all upper block-triangular with blocks of size  $(N + 1) \times (N + 1)$  along the diagonal. In total, there are  $B$  dark-blue blocks, all of size  $\mathcal{Q}(N + 1) \times \mathcal{Q}(N + 1)$ . Finally, we recognise a re-occurring sparsity pattern in the green blocks. See the text for a detailed explanation on the origin of the blue and green block structure. (For interpretation of the colours in the figure(s), the reader is referred to the web version of this article.)

$$\delta_{pq} W_{rs} M_{kl} \tilde{Q}_{qls} - \delta_{pq} w_r M_{kl} \tilde{Q}_{ql}^0 - \delta_{pq} K_{rs}^\tau M_{kl} \tilde{Q}_{qls} + \sum_{j=1}^3 \mathcal{A}_{pq}^{j*} S_{rs} K_{kl}^j \tilde{Q}_{qls} = E_{pq}^* S_{rs} M_{kl} \tilde{Q}_{qls}.$$

We now collect terms involving  $\tilde{Q}_{qlk}$  on the left and the other part on the right:

$$\left( \delta_{pq} W_{rs} M_{kl} - \delta_{pq} K_{rs}^\tau M_{kl} - E_{pq}^* S_{rs} M_{kl} + \sum_{j=1}^3 \mathcal{A}_{pq}^{j*} S_{rs} K_{kl}^j \right) \tilde{Q}_{qls} = \delta_{pq} w_r M_{kl} \tilde{Q}_{ql}^0. \quad (7)$$

The system of equations can now be stated in the form

$$Y_{pkrqls} \tilde{Q}_{qls} = r_{pkr}. \quad (8)$$

This system can be transformed into a matrix-vector form if we map the multi-indices to linear indices. By solving this system, we get a predicted solution for  $q_p(t_i + \delta t)$ , which replaces Equation (5). Again we can combine this with Equation (4), integrate from  $t_i$  to  $t_{i+1}$  and obtain the solution at time  $t_{i+1}$ .

## 5. A new efficient inversion of the system matrix

In Section 4.3 we introduce the ADER-DG method with a space-time predictor to effectively treat the stiff source term inherent to poroelastic wave propagation. What remains is to solve a local system as given in Equation (8). We can write this linear system of equations in standard matrix form,  $Ax = b$ , by unrolling the multi-indices  $pkr \rightarrow i$  and  $qls \rightarrow j$ . In the approach by de la Puente et al. [9], the inverse of this matrix was precomputed for every element. During the simulation phase, these systems were simply solved one after the other, which has two main disadvantages: (1) The operator  $Y$  contains information on the material parameters and the shape of the elements, thus, the operator differs for every element. (2) Since, for example, 4,368 unknowns are associated with polynomial degree 5, each LU decomposition needs 145.6 MB of storage per element. Even on large clusters, this easily poses a severe limitation. Additionally, precomputing the decomposition demands substantial computational resources.

Here, we present a new, optimised solution approach that relies on a modified back-substitution. It does not require explicit unrolling of multi-indices, but makes use of the tensor structure of  $\tilde{Q}$ . Thus, no decomposition has to be computed or stored. In terms of floating-point operations and memory requirement needed for one back-substitution, we outperform an LU decomposition by far. In addition, our scheme can be implemented using small matrix-matrix multiplications (GEMMs). For these kinds of tensor operations, the code generator YATeTo [65] can generate architecture-specific code to achieve high performance.

### 5.1. Structure of the system matrix

In Equations (7) and (8), the indices  $p$  and  $q$  range from 1 to  $\mathcal{Q}$  (number of quantities),  $r$  and  $s$  range from 1 to  $N + 1$  (number of temporal basis functions), and  $k$  and  $l$  range from 1 to  $B = B(N)$  (number of spatial basis functions with degrees up to  $N$ ). There are several ways to unroll the multi-indices  $pkr \rightarrow i$  and  $qls \rightarrow j$ . If we choose  $k/l$  as the slowest and  $r/s$  as the fastest-running indices, we observe a sparsity pattern and block-structure of the system matrix  $A_{ij}$ , as shown in Fig. 2. Overall, the system (8) is in upper block-triangular form. The blocks on the diagonal hinder us from using standard row-wise back-substitution (such as the triangular solver from Level 2 BLAS [29]). Nonetheless, we can make use of this property to derive a block-wise back-substitution algorithm in Section 5.2.

The block-triangular structure of the matrix stems from our choice of basis functions, which are orthogonal polynomials.

**Lemma 1.** If the basis functions  $\phi_l$  are orthogonal and ordered such that  $\deg(\phi_k) \leq \deg(\phi_l)$  for  $k \leq l$ , the stiffness matrix  $K^\alpha$  is upper triangular. In particular, we see larger blocks of zeros: Let  $B_n = \binom{n+3}{3}$ . Then for our choice of basis functions, we have:

$$\forall n \in [1, N+1] : \forall i \in (B_{n-1}, B_n] : \forall j \in [1, B_n] : K_{ij}^\alpha = 0$$

**Proof.** The basis functions are numbered with increasing degree, where each degree adds  $\binom{n+2}{2}$  basis functions. That is, the first basis function is of degree 0, the next 3 basis functions have degree 1, the next 6 basis functions have degree 2, and so forth. Let  $i \in (B_{n-1}, B_n]$  and  $j \in [1, B_n]$ . Denote the degree of  $\phi_i$  with  $k_i$  and the degree of  $\phi_j$  with  $k_j$ . Then  $k_j \leq k_i$ . We obtain the stiffness matrices by an inner product  $K_{ij}^\alpha = \langle \phi_i, \partial_\alpha \phi_j \rangle$ . Taking the derivative of a basis function of degree  $k_j$  yields a polynomial of degree  $k_j - 1$ , which we can write as a linear combination of the basis functions up to degree  $k_j - 1$ . Since all basis functions of degree  $k_i$  are orthogonal to the polynomials of degree up to  $k_j - 1$ , it follows that  $K_{ij}^\alpha = 0$ .  $\square$

Now, we can see how the blue and green blocks in Fig. 2 arise from Equation (7): The overall shape of the green blocks resembles the sparsity pattern of the stiffness matrices. Therefore, we identify the green block with the pattern  $\sum_{j=1}^3 \mathcal{A}_{pq}^{j*} S_{rs} K_{kl}^j$ . Within each block, we see a replication of the sparsity pattern of the Jacobian matrices ( $\mathcal{A}_{pq}^{j*}$ ), where each entry on the diagonal is replaced by the  $(N+1) \times (N+1)$  diagonal matrix  $S_{rs}$ . The blue blocks on the diagonal resemble the other part of Equation (7):  $(\delta_{pq} W_{rs} - \delta_{pq} K_{rs}^\tau - E_{pq}^* S_{rs}) M_{kl}$ . The mass matrix  $M_{kl}$  is diagonal, which explains the location of the dark-blue blocks and the existence of the light-blue blocks. Within each dark-blue block, we see smaller blocks of size  $(N+1) \times (N+1)$  on the diagonal  $(\delta_{pq} W_{rs} - \delta_{pq} K_{rs}^\tau)$ . Above the diagonal (in particular on a side diagonal), we see the entries  $E_{pq} S_{rs}$ . We can group the dark-blue blocks on the diagonal to  $(N+1)$  blocks of size  $(N+1) \mathcal{Q}(\frac{n+2}{2}) \times (N+1) \mathcal{Q}(\frac{n+2}{2})$ . Each of these blocks corresponds to basis functions of a certain polynomial degree.

## 5.2. Block-wise back-substitution

We continue with Equation (7) and derive the block-wise back-substitution procedure. First, we multiply with  $M_{mk}^{-1} S_{ur}^{-1}$ :

$$\left( \delta_{pq} S_{ur}^{-1} W_{rs} \delta_{ml} - \delta_{pq} S_{ur}^{-1} K_{rs}^\tau \delta_{ml} - E_{pq}^* \delta_{us} \delta_{ml} + \sum_{j=1}^3 \mathcal{A}_{pq}^{j*} \delta_{us} M_{mk}^{-1} K_{kl}^j \right) Q_{qls} = \delta_{pq} S_{ur}^{-1} w_r \delta_{ml} Q_{ql}^0 = S_{ur}^{-1} w_r Q_{pm}^0.$$

Now, we move the parts containing the spatial stiffness matrices onto the right-hand side, since we know this part vanishes for some index combinations:

$$(\delta_{pq} S_{ur}^{-1} W_{rs} \delta_{ml} - \delta_{pq} S_{ur}^{-1} K_{rs}^\tau \delta_{ml} - E_{pq}^* \delta_{us} \delta_{ml}) Q_{qls} = S_{ur}^{-1} w_r Q_{pm}^0 - \left( \sum_{j=1}^3 \mathcal{A}_{pq}^{j*} \delta_{us} M_{mk}^{-1} K_{kl}^j \right) Q_{qls}.$$

Next, we can factor out some of the  $\delta$  functions. Furthermore, we introduce  $\hat{K}_{ml}^\alpha = M_{mr}^{-1} K_{rl}^\alpha$ , which renders the equations simpler:

$$(\delta_{pq} S_{ur}^{-1} W_{rs} - \delta_{pq} S_{ur}^{-1} K_{rs}^\tau - E_{pq}^* \delta_{us}) Q_{qms} = S_{ur}^{-1} w_r Q_{pm}^0 - \left( \sum_{j=1}^m \mathcal{A}_{pq}^{j*} \hat{K}_{ml}^j \right) Q_{qlu}.$$

Since the mass matrix is diagonal due to the choice of orthogonal basis functions,  $\hat{K}^\alpha$  has the same sparsity pattern as  $K^\alpha$ . We can write the sum over  $l$  on the right-hand side explicitly now and neglect all parts which are 0. Then:

$$b_{pmu} := S_{ur}^{-1} w_r Q_{pm}^0 - \sum_{l=m+1}^B \left( A_{pq}^* \hat{K}_{ml}^\xi + B_{pq}^* \hat{K}_{ml}^\eta + C_{pq}^* \hat{K}_{ml}^\zeta \right) Q_{qlu}. \quad (9)$$

Thus,  $b_{pmu}$  only depends on  $Q_{:l}$  with  $l > m$ . In particular, for  $m = B$  we obtain

$$b_{pBu} := S_{ur}^{-1} w_r Q_{pB}^0.$$

Recall that we want to solve the system for every  $m$ :

$$(\delta_{pq} S_{ur}^{-1} W_{rs} - \delta_{pq} S_{ur}^{-1} K_{rs}^\tau - E_{pq}^* \delta_{us}) Q_{qms} = b_{pmu}. \quad (10)$$

For  $m = B$ , the right-hand side  $b$  does not depend on  $Q$ . For  $m < B$ , the right-hand side  $b$  depends on  $Q_{:l}$  with  $l > m$ . Hence we can solve the system of equations for  $Q_{:m}$  backwards in the order  $m = B, \dots, 1$  and update the right-hand side with the already computed values of  $Q$ , as shown in Algorithm 1. Here we still need to solve a system of size  $\mathcal{Q}(N+1) \times \mathcal{Q}(N+1)$  for  $B$  iterations.

```

1  $b_{pBu} \leftarrow S_{ur}^{-1} w_r Q_{pB}^0$ ;
2 for  $m \leftarrow B$  downto 1 do
3   // dark-blue blocks
4   Solve  $(\delta_{pq} S_{ur}^{-1} W_{rs} - \delta_{pq} S_{ur}^{-1} K_{rs}^\tau - E_{pq}^* \delta_{us}) Q_{qms} = b_{pmu}$ ;
5   // green blocks
6   Update  $b_{p(m-1)u}$  using Equation (9);
7 end

```

**Algorithm 1:** First simple block-wise back-substitution algorithm.

We can further optimise the algorithm by using the sparsity pattern of  $E^*$ , which is upper triangular and repeat what we have done earlier. We collect the matrices on the left-hand side of Equation (10) further:

$$\left( \delta_{pq} \underbrace{S_{ur}^{-1} (W_{rs} - K_{rs}^\tau)}_{=: Z_{us}} - E_{pq}^* \delta_{us} \right) Q_{qms} = b_{pmu}.$$

Then we split  $E^*$  in a diagonal and in a strictly upper triangular part, i.e.  $E^* = F + G$ , where  $F = \text{diag}(E^*)$ , and  $G = E^* - F$ . We put  $G$  on the right-hand side and explicitly write the sum over  $q$ :

$$(\delta_{pq} Z_{us} - F_{pq} \delta_{us}) Q_{qms} = b_{pmu} + \sum_{o=p+1}^Q G_{po} Q_{omu} =: \hat{b}_{pmu}.$$

Again, we see that the right-hand side does not depend on  $Q$  for  $p = Q$ :

$$\hat{b}_{Qmu} = b_{Qmu}.$$

$F$  is diagonal, i.e.  $F_{pq} = E_{pp}^* \delta_{pq}$ . Note that the upper-case  $P$  has the same value as the lower-case  $p$  but no summation is implied. We can further simplify the equations by pulling out  $\delta_{pq}$  and obtain the final system that needs to be solved for all  $p$  and  $m$ :

$$(Z_{us} - E_{pp}^* \delta_{us}) Q_{pms} = \hat{b}_{pmu}.$$

In summary, we obtain Algorithm 2, where  $\cdot$  stands for a matrix multiplication and  $\circ$  for a tensor product. In comparison to Algorithm 1, we only have to solve  $BQ$  systems of size  $(N+1) \times (N+1)$ .

```

1  $b \leftarrow Q^0 \circ (S^{-1} w)$ ;
2 for  $m \leftarrow B$  downto 1 do
3   // use sparsity pattern of dark-blue blocks
4   for  $p \leftarrow Q$  downto 1 do
5      $Q_{pm} \leftarrow (Z - E_{pp}^*)^{-1} \cdot b_{pm}$ ;
6     for  $o \leftarrow 1$  to  $p-1$  do
7        $b_{om} \leftarrow b_{om} + G_{op} \cdot Q_{pm}$ ;
8     end
9   end
10  // green blocks
11  for  $n \leftarrow 1$  to  $m-1$  do
12     $b_{:n} \leftarrow b_{:n} - \hat{K}_{nm}^\xi A^* \cdot Q_{:m} - \hat{K}_{nm}^\eta B^* \cdot Q_{:m} - \hat{K}_{nm}^\zeta C^* \cdot Q_{:m}$ ;
13  end
14 end

```

**Algorithm 2:** Back-substitution algorithm unrolled over  $m$  and  $p$ .

We can even further optimise the algorithm by using the internal structure of the blue and green blocks. Until now, we have only used the block structure given by the dark-blue blocks, but we also see that the system matrix contains larger blocks of zeros (light-blue in Fig. 2). Thus, for  $m \leq B_n$  the right-hand side  $b_{pmu}$  depends on  $Q_{:l}$  for  $l \in (B_n, B]$ . We can use this information to construct Algorithm 3, in which we fuse iterations within the loop over  $m$ .

Next, we compare the number of floating-point operations needed to solve one system of equations using an LU decomposition and our Algorithm 3. As material parameters do not change over time, we can reuse the same decomposition in each time step. The LU decomposition can be computed in advance for each element and stored. During the simulation, we have to perform back-substitution twice, thus, the number of floating-point operations is  $2s^2$ , where  $s$  is the number of unknowns. Now, consider the newly proposed block-wise back-substitution approach: Let  $d_n = \binom{n+2}{2}$ . The number of



```

1  $b \leftarrow Q^0 \circ (S^{-1}w);$ 
2 for  $n \leftarrow N+1$  downto 1 do
3   // dark and light-blue blocks together
4    $m \leftarrow (B_{n-1}, B_n);$ 
5   for  $p \leftarrow Q$  downto 1 do
6      $Q_{pm} \leftarrow b_{pm} \cdot (Z - E_{pp}^* I)^{-T};$  //  $2d_n(N+1)^2$  flop
7     for  $o \leftarrow 1$  to  $p-1$  do
8        $b_{om} \leftarrow b_{om} + G_{op} \cdot Q_{pm};$  //  $2d_n(N+1)$  flop
9     end
10  end
11  // green blocks
12  if  $n > 1$  then
13     $b \leftarrow b - A^* \times_1 Q_{:,m} \times_2 \hat{K}_{:,m}^\xi;$ 
14     $b \leftarrow b - B^* \times_1 Q_{:,m} \times_2 \hat{K}_{:,m}^\eta;$ 
15     $b \leftarrow b - C^* \times_1 Q_{:,m} \times_2 \hat{K}_{:,m}^\zeta;$ 
16    // each:  $QB(N+1) + 2(N+1)Q^2d_n + 2(N+1)Qd_nB$  flop
17  end
18 end

```

**Algorithm 3:** Back-substitution algorithm with iterations fused over  $m$  to better match the sparsity pattern. The number of floating-point operations for each tensor contraction is provided as comments. The product  $\times_n$  denotes the  $n$ -mode product, cf. [66].

**Table 2**

Comparison of the computational effort (absolute count of floating-point operations) and memory requirements for the standard LU decomposition and our newly proposed approach (STP, Algorithm 3).

$N$	2	3	4	5	6
#unknowns	$3.90 \cdot 10^2$	$1.04 \cdot 10^3$	$2.28 \cdot 10^3$	$4.37 \cdot 10^3$	$7.64 \cdot 10^3$
#FLOP LU	$3.04 \cdot 10^5$	$2.16 \cdot 10^6$	$1.04 \cdot 10^7$	$3.82 \cdot 10^7$	$1.17 \cdot 10^8$
#FLOP STP	$5.99 \cdot 10^4$	$2.27 \cdot 10^5$	$7.13 \cdot 10^5$	$1.94 \cdot 10^6$	$4.72 \cdot 10^6$
reduction	5.08	9.52	$1.45 \cdot 10^1$	$1.97 \cdot 10^1$	$2.48 \cdot 10^1$
storage LU [MB]	1.16	8.26	$3.95 \cdot 10^1$	$1.46 \cdot 10^2$	$4.46 \cdot 10^2$
storage STP [MB]	$4.81 \cdot 10^{-3}$	$5.50 \cdot 10^{-3}$	$6.39 \cdot 10^{-3}$	$7.48 \cdot 10^{-3}$	$8.77 \cdot 10^{-3}$
reduction	$2.42 \cdot 10^2$	$1.50 \cdot 10^3$	$6.18 \cdot 10^3$	$1.95 \cdot 10^4$	$5.08 \cdot 10^4$

floating-point operations for each tensor contraction is provided in Algorithm 3. Now we only have to sum over the loops. Let us note here that  $G$  has 3 non-zero entries, thus, we execute line 8 only three times. The number of floating-point operations for the  $n^{\text{th}}$  execution of the outermost loop is bounded by

$$\begin{aligned}
& \underbrace{Q \cdot 2d_n(N+1)^2}_{\text{line 6}} + \underbrace{3 \cdot 2d_n(N+1)}_{\text{line 8}} + \underbrace{3 \left( QB(N+1) + 2(N+1)Q^2d_n + 2(N+1)Qd_nB \right)}_{\text{lines 13-16}} \\
& = 2d_n \left( Q(N+1)^2 + 3(N+1) + 3(N+1)Q^2 + 3(N+1)QB \right) + 3QB(N+1).
\end{aligned}$$

As  $\sum_{n=1}^N d_n = B$  and the other terms are independent of  $n$ , we obtain

$$2B \left( Q(N+1)^2 + 3(N+1) + 3(N+1)Q^2 + 3(N+1)QB \right) + 3QB(N+1)^2$$

as an upper bound for Algorithm 3. Comparison of the number of floating-point operations in Table 2 shows significant speed-up comparing our Algorithm 3 to the LU decomposition for all considered orders of accuracy. The speed-up factor increases approximately linearly with the polynomial degree. Specifically, we see a reduction of computational effort by a factor of  $\approx 25$  for polynomial degree 6. Our Algorithm 3 outperforms LU decomposition also in terms of memory requirements. For the LU decomposition, two triangular matrices of size  $BQ(N+1) \times BQ(N+1)$  have to be stored. For our back-substitution algorithm, we only need to store the matrices  $(Z - E_{pp}^* I)^{-1}$  (13 matrices of size  $(N+1) \times (N+1)$ ) and the matrices  $A^*, B^*, C^*$  (each of size  $Q \times Q$ ) and  $E^*$  (6 non-zero entries).

## 6. Verification

To verify the implementation of the block-wise back-substitution approach (Algorithm 3 in Section 4.3) in SeisSol, we perform a series of numerical verification tests with canonical models of different complexity:

**Table 3**  
Material parameters used for the planar wave convergence analysis.

Parameter		Value	
Solid Bulk modulus	$K_S$	$4.00 \cdot 10^{10}$	Pa
Solid density	$\rho_S$	$2.50 \cdot 10^3$	$\text{kg m}^{-3}$
Matrix 1 <sup>st</sup> Lamé parameter	$\lambda_M$	$1.20 \cdot 10^{10}$	Pa
Matrix 2 <sup>nd</sup> Lamé parameter	$\mu_M$	$1.00 \cdot 10^{10}$	Pa
Matrix porosity	$\phi$	0.20	
Matrix permeability	$\kappa$	$6.00 \cdot 10^{-13}$	$\text{m}^2$
Matrix tortuosity	$T$	3.00	
Fluid bulk modulus	$K_F$	$2.5 \cdot 10^9$	Pa
Fluid density	$\rho_F$	$1.04 \cdot 10^3$	$\text{kg m}^{-3}$
Fluid viscosity	$\nu$	$1.00 \cdot 10^{-3}$	Pa s

1. We impose a planar wave as initial condition and let it evolve over time. On a cascade of finer and finer meshes, we verify convergence of all 13 unknowns (stresses, solid particle velocities, pore pressure, relative fluid velocities) against an analytical reference solution.
2. We consider an explosive point source in a homogeneous full space with either a viscous or an inviscid fluid. We compare the time histories of the solid particle velocities and relative fluid velocities at selected receiver positions using the semi-analytical solution by Karpfinger et al. [32].
3. We consider a contact of two half-spaces with an inviscid fluid to assess how well SeisSol resolves reflection and transmission of waves at internal interfaces. We compare the solid particle velocities to the semi-analytical solution by Diaz and Ezziani [31].
4. We verify the reflection of waves at a free surface of a half-space filled with an inviscid fluid. Again, the reference for the solid particle velocities is given by [31].
5. As final verification and demonstration example, we define a new layer over half-space problem. For isotropic elastic materials, the layer over half-space community benchmark is widely accepted (cf. [67]). Here, we combine verification setups 2, 3 and 4 to simulate wave propagation with a free surface and an internal interface in a poroelastic medium filled with a viscous fluid. To the best of our knowledge, there is no analytical or semi-analytical solution available for such model configurations. Therefore, we compare our solution against an independent numerical simulation using the FD method [36,37].

The first three verification setups follow [9], the two last test cases are inspired by similar 2D examples in [37].

### 6.1. Planar wave convergence analysis

We consider a cube  $[-1, 1]^3$  with periodic boundary conditions. The cube is refined into  $4^3$ ,  $8^3$ ,  $16^3$ ,  $32^3$  and  $64^3$  equally sized sub-cubes and every cube is partitioned into 5 tetrahedrons. We express the reference solution in the form of a planar wave:

$$q(x, t) = \sum_{n=1}^{13} \alpha_n r_n e^{i(\omega_n t - k \cdot x)}.$$

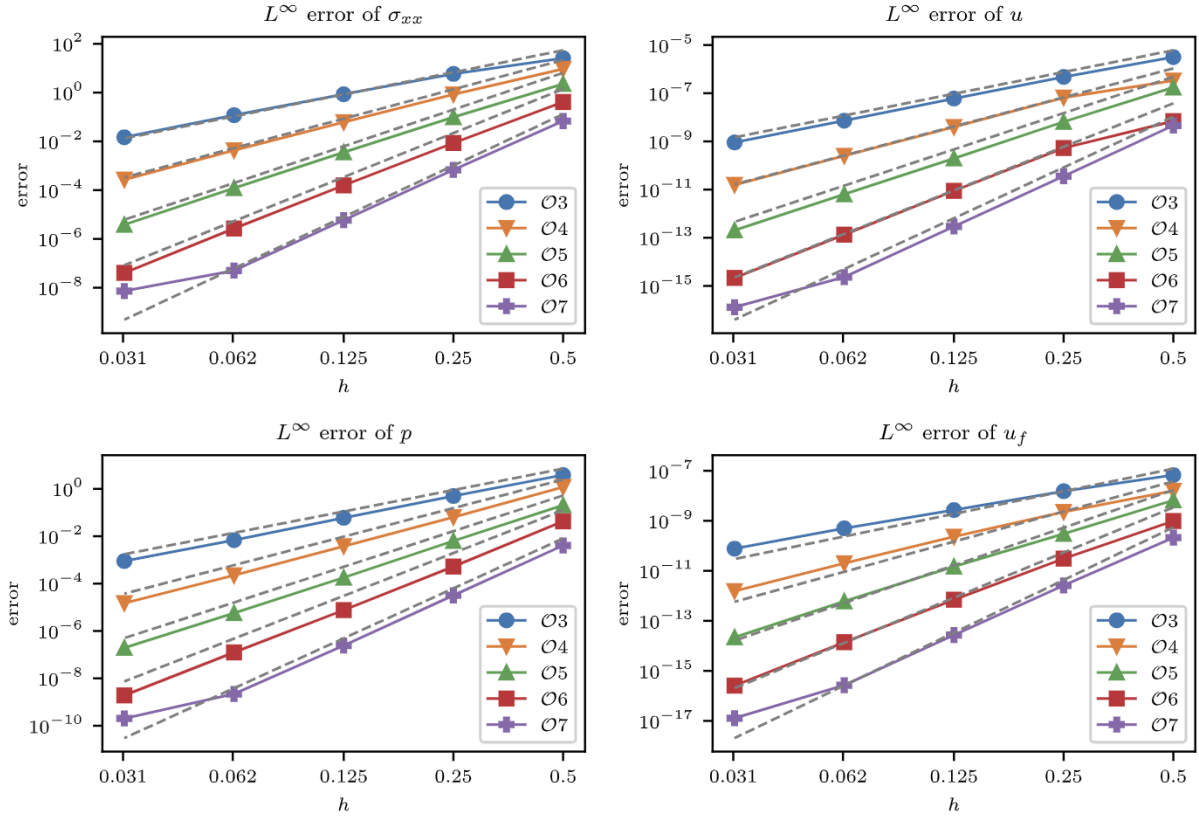
Here  $\omega_n, r_n$  are eigenpairs of the matrix  $k_x A + k_y B + k_z C - iE$  and  $i$  denotes the imaginary unit. It is easy to verify that the real part of  $\hat{Q}$  is a solution of Equation (1). Each eigenpair corresponds to a wave mode. The vector  $k = (\pi, \pi, \pi)$  describes the direction in which the waves propagate. The scaling factors  $\alpha_n$  define the relative amplitudes of these wave modes. We set:  $\alpha_1, \alpha_2, \alpha_3, \alpha_4 = 100.0$  and all others to 0. The first mode corresponds to a fast P-wave, the second and third modes correspond to different polarisations of the S-wave, and the fourth mode is the slow P-wave. The material parameters are given in Table 3. As an initial condition we set  $\hat{Q}(x, 0)$ . The final time of the simulation is  $1 \cdot 10^{-4}$  s. The fast P-wave has a velocity of  $2,715.6 \text{ m s}^{-1}$ , thus, the wave travels 0.27 m during the simulation time. We compare the simulation result at the final time step of our simulation with the analytic reference solution using the  $L^1$ ,  $L^2$  and  $L^\infty$  norms within the cube  $[-1, 1]^3$ .

In Fig. 3 we show the dependence of the  $L^\infty$  error between simulated results and analytic solution on the characteristic edge length  $h$  for stress  $\sigma_{xx}$ , solid particle velocity  $u$ , pore pressure  $p$  and relative fluid velocity  $u_f$ . Table 4 summarises the characteristic element edge length  $h$  and the number of elements for the considered meshes. When we use polynomial basis functions up to degree  $N$ , we expect convergence of the order  $\mathcal{O} = N + 1$  [27]. In Fig. 3, we clearly see the expected convergence behaviour for the different choices of basis functions. For convergence order 7 and the finest mesh, the convergence slows down, since we reach machine precision for the solid particle velocities and the relative fluid velocities. In other norms and for other quantities we find the same expected convergence behaviour, see Appendix A.

**Table 4**

Characteristic length in meter and number of elements for the different meshes used in the planar wave convergence analysis.

$N$	4	8	16	32	64
$h$	0.5	0.25	0.125	0.0625	0.03125
#elem	320	2560	20480	163840	1310720



**Fig. 3.** Convergence plots for selected unknowns of the planar wave convergence setup in the  $L^\infty$  norm. The expected convergence order is plotted in grey dashed lines. For  $\mathcal{O}7$  we are close to machine precision on the finest mesh.

**Table 5**

Receiver positions (in meter) for the homogeneous full space problem.

	x1	x2	x3	x4	d1	d2	d3	d4
$x$	-1,000	-600	600	1,000	-575	-345	345	575
$y$	0	0	0	0	-575	-345	345	575
$z$	0	0	0	0	-575	-345	345	575

## 6.2. Homogeneous full-space

Next, we verify the wave propagation excited by a point source by modifying a similar problem setup in [9]. We consider a homogeneous full-space with an explosive point source at the origin. The time history is a Ricker wavelet with dominant frequency  $f_0 = 16$  Hz and time delay  $t_0 = 0.07$  s:

$$s(t) = \left(1 - 2(\pi f_0(t - t_0))^2\right) \cdot \exp\left(-(\pi f_0(t - t_0))^2\right).$$

The source acts on all three diagonal parts of the stress tensor,  $\sigma_{xx}$ ,  $\sigma_{yy}$ ,  $\sigma_{zz}$ , and the pore pressure  $p$ . A set of four receivers is placed along the x-axis, while another set of four receivers is placed along the body diagonal. The coordinates of all receivers are stated in Table 5. The material parameters are stated in Table 6. We consider the viscous case ( $\nu = 0.001$  Pa s) and the inviscid case ( $\nu = 0.0$  Pa s).



**Table 6**  
Material parameters for the homogeneous full-space problem.

Parameter	Value	
$K_S$	$2.00 \cdot 10^{10}$	Pa
$\rho_S$	$2.08 \cdot 10^3$	$\text{kg m}^{-3}$
$\lambda_M$	$5.28 \cdot 10^9$	Pa
$\mu_M$	$6.40 \cdot 10^9$	Pa
$\phi$	0.40	
$\kappa$	$6.00 \cdot 10^{-13}$	$\text{m}^2$
$T$	2.00	
$K_F$	$2.50 \cdot 10^9$	Pa
$\rho_F$	$1.04 \cdot 10^3$	$\text{kg m}^{-3}$
$\nu$	$1.00 \cdot 10^{-3}$	Pa s

The computational domain is a box  $[-3000, 3000]^3$ . To simulate the full space, we impose absorbing boundary conditions at the outer faces. SeisSol supports absorbing boundary conditions as detailed in [13]. These boundary conditions absorb waves with normal incidence very well, but in corners or in the case of grazing incidence spurious reflections can occur. We typically enlarge the computational domain, such that these artifacts are not observable in the region of interest. In the cube  $[-1500, 1500]^3$  the mesh has a resolution of 30 m. The mesh is further refined around the origin with element edge size down to 3 m to be able to resolve the explosive source accurately. Outside of the refined cube, the mesh is coarsened towards the mesh borders. In total, the mesh consists of  $6.27 \cdot 10^6$  elements. In the 1.5 s simulation window the generated waves pass through all receivers. The simulation is carried out with convergence order 6.

We compare the simulation results obtained from SeisSol with semi-analytical solutions calculated using a Green's function approach [32]. Throughout this section we use time-frequency misfits [68,69] to quantify differences between SeisSol and the reference solution. A detailed comparison for the x-component of the solid particle velocity ( $u$ ) at receiver d4 in Fig. 4 illustrates the excellent agreement between both solutions. In the inviscid case (Fig. 4a), we observe both the fast P-wave (the first pulse) and the slow P-wave (the second pulse). In the viscous case (Fig. 4b), we only observe the fast P-wave since the slow P-wave is strongly diffusive and attenuates very quickly with distance from the source.

In Fig. 5, we compare single-valued envelope misfits (EM) and phase misfits (PM) for all components of the solid particle velocity and relative fluid velocity for all considered receivers. This summary comparison reveals consistently excellent agreement between SeisSol and semi-analytic solutions with PM below 0.2% and EM below  $\approx 1\%$ , which is slightly exceeded only for  $u_f$  at receiver x4. We note that small variations at individual receivers are expected due to the unstructured mesh.

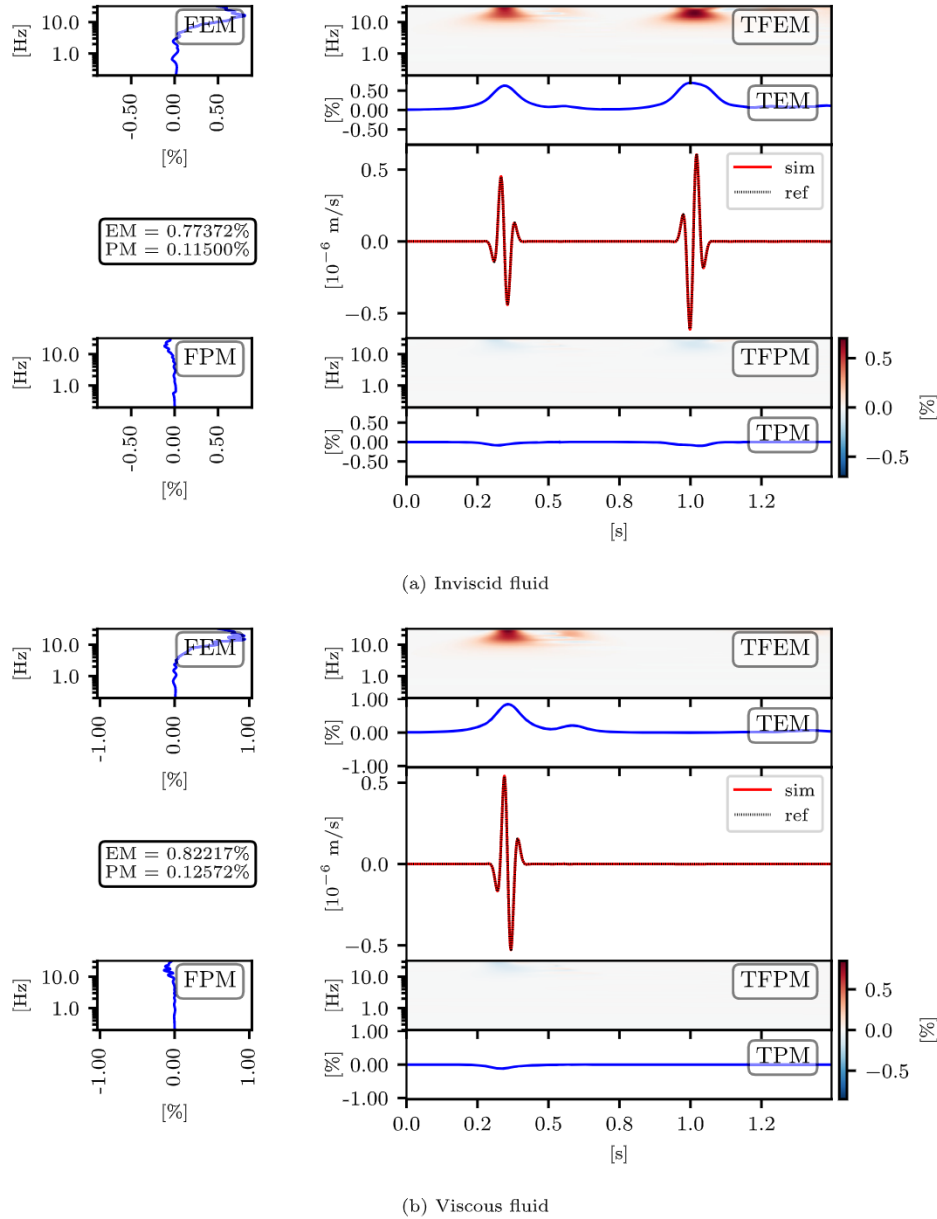
### 6.3. Contact of two half-spaces

To verify whether SeisSol correctly simulates waves reflected from and transmitted through an internal interface, we consider a contact of two half-spaces benchmark problem. We modify a 2D version proposed in [9] to a 3D volume with an adapted source-receiver configuration. Material properties of the half-spaces are given in Table 7. We place an explosive point source at (0 m, 0 m, 500 m). As in Section 6.2 the time history is a Ricker wavelet with  $f_0 = 16$  Hz and a time delay of 0.07 s. We consider two receivers: Receiver r (500 m, 400 m, 500 m) at the same side of the interface as the source, to see the reflected waves, and Receiver t (500 m, 400 m, -500 m) across the material interface to record the transmitted waves.

The computational domain is  $[-5000, 5000]^3$  with absorbing boundary conditions. We refine in a cube with edge length 2 km around the origin up to a characteristic length of 40 m with further refinement around the source and coarsening towards the boundary. The mesh has  $2.08 \cdot 10^6$  elements. The method by [32], used to calculate the reference solution in the previous setup, is applicable only in full spaces. Therefore, we use the semi-analytical code Gar6More3D [31] that allows to include planar interfaces between two half-spaces. However, Gar6More3D only supports inviscid fluids ( $\nu = 0$  Pa s) and provides solutions for solid particle velocities. Fig. 6 shows a detailed comparison for the x-component of the solid particle velocity between simulated results and the reference solution for both receivers. Table 8 summarises all EM and PM values for all components of the solid particle velocity. Both comparisons illustrate excellent agreement between the SeisSol and the reference solutions. However, we note that the agreement at receiver t is slightly worse. The speed of the fast P-wave is  $2,480.7 \text{ m s}^{-1}$  for  $z < 0$  and  $4,246.9 \text{ m s}^{-1}$  for  $z > 0$ . With a lower wave speed the wavelength decreases, hence a finer mesh is needed for the same accuracy. We used the same characteristic edge length on both sides of the interface, which explains the slightly increased misfit at receiver t.

### 6.4. Free surface

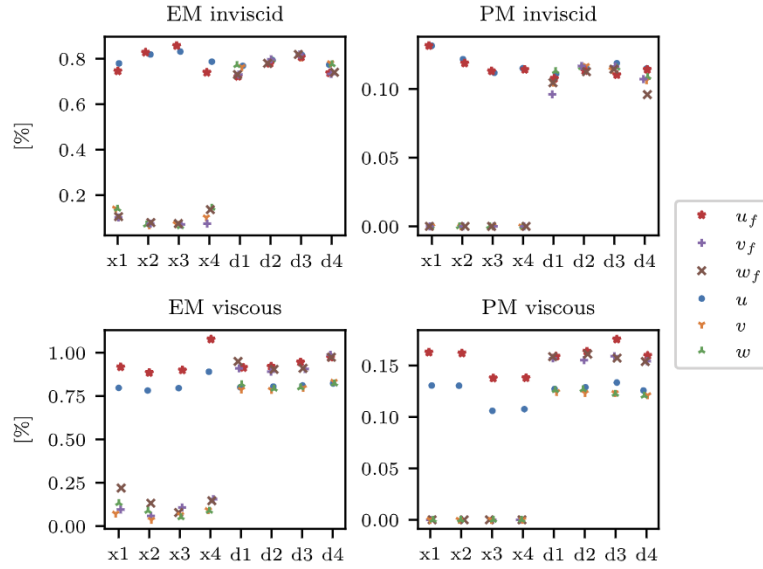
Including a free surface is of special interest in seismic simulations, since in most seismological configurations the traction-free condition is a sufficient approximation of the Earth's surface. Furthermore, seismic motion is commonly



**Fig. 4.** Detailed misfit plots for the solid particle velocity component  $u$  at receiver  $d_4$  for the homogeneous full-space problem. The wave is excited by an explosive point source in a homogeneous full space. We plot Frequency Envelope Misfit (FEM), Time Frequency Envelope Misfit (TFEM), Time Envelope Misfit (TEM), single-valued Envelope Misfit (EM), single-valued Phase Misfit (PM), the simulated and reference signal, Frequency Phase Misfit (FPM), Time Frequency Phase Misfit (TFPM) and Time Phase Misfit (TPM).

recorded at seismic stations located at Earth's surface. A benchmark setup with a homogeneous half-space is reported in [37] for a 2D geometry. We here modify this to 3D, use different material parameters and another source-receiver configuration. We consider a half-space with a free surface boundary condition ( $\sigma \cdot n = 0$ ,  $p = 0$ ) at  $z = 0$ . The half-space is homogeneous with material parameters in Table 7 for  $z > 0$ . We place an explosive point source (Ricker wavelet,  $f_0 = 5$  Hz,  $t_0 = 0.25$  s) at (0 m, 0 m, 500 m). We consider a receiver at a shallow depth at (500 m, 400 m, 0.5 m) due to slow convergence of the reference solution using Gar6More3D [31] directly at the free surface. Additionally, we consider a receiver at depth (500 m, 400 m, 500 m) to record the reflected waves. The computational domain is  $[-5000, 5000] \times [-5000, 5000] \times [0, 5000]$ . The mesh has a characteristic length of 30 m in the cuboid  $[-1000, 1000] \times [-1000, 1000] \times [0, 1000]$ . As before, the mesh is coarsened towards the boundary and refined towards the source. It consists of  $2.17 \cdot 10^6$  elements.

The agreement of the SeisSol and reference solution is excellent as documented by all misfit values below 1% (Table 9). Fig. 7 shows a detailed comparison between the SeisSol and reference solutions for the  $x$  component of the solid particle velocity at both receivers. We note that the reference solution contains, in the plots barely visible, a wave at approximately



**Fig. 5.** Plot of the envelope and phase misfits of all receivers for the homogeneous full-space problem. We distinguish between an inviscid and a viscous fluid. For both cases, we achieve excellent agreement between simulation and reference.

**Table 7**

Material parameters we used for the contact of two half-spaces problem.

Parameter	$z > 0$	$z < 0$	
$K_S$	$4.00 \cdot 10^{10}$	$7.60 \cdot 10^9$	Pa
$\rho_S$	$2.50 \cdot 10^3$	$2.21 \cdot 10^3$	$\text{kg m}^{-3}$
$\lambda_M$	$1.20 \cdot 10^{10}$	$3.96 \cdot 10^9$	Pa
$\mu_M$	$1.20 \cdot 10^{10}$	$3.96 \cdot 10^9$	Pa
$\phi$	0.20	0.16	
$\kappa$	$6.00 \cdot 10^{-13}$	$1.00 \cdot 10^{-13}$	$\text{m}^2$
$T$	2.00	2.00	
$K_F$	$2.50 \cdot 10^9$	$2.50 \cdot 10^9$	Pa
$\rho_F$	$1.04 \cdot 10^3$	$1.04 \cdot 10^3$	$\text{kg m}^{-3}$
$\nu$	0.00	0.00	Pa s

**Table 8**

Misfits (in %) for the contact of two half-spaces problem.

Receiver	EM $u$	PM $u$	EM $v$	PM $v$	EM $w$	PM $w$
r	0.26	0.05	0.22	0.04	0.17	0.02
t	1.41	0.20	1.12	0.16	1.54	0.28

**Table 9**

Misfits (in %) for the free surface problem.

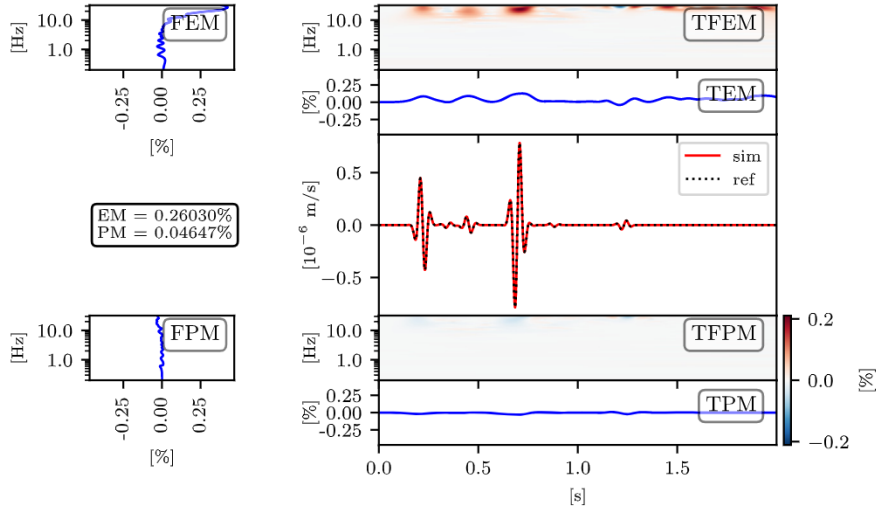
Receiver	EM $u$	PM $u$	EM $v$	PM $v$	EM $w$	PM $w$
0.5 m	0.27	0.80	0.22	0.64	0.40	1.22
500 m	0.35	0.12	0.28	0.10	0.30	0.08

1.5 s. The amplitude of this wave slowly decreases with a finer resolution of the numerical quadrature scheme used in Gar6more3D. In agreement with one of the Gar6more3D authors (Diaz, 2021, personal communication), we conclude that this wave is unphysical. By shifting the receiver to a shallow depth, we were able to obtain a reliable and sufficiently accurate reference solution.

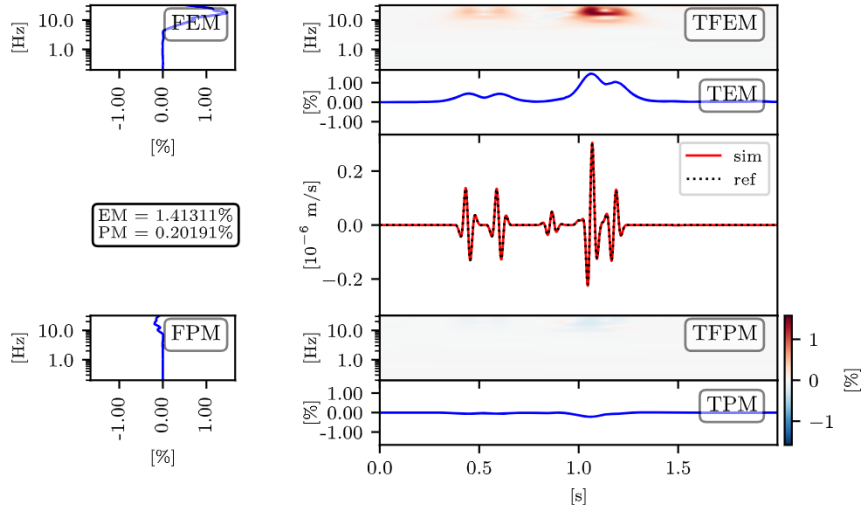
### 6.5. Layer over half-space

Our final verification benchmark is inspired by the SISMOWINE LOH1 configuration [67]. To emphasise that we consider a poroelastic layer over half-space scenario, we will call this setup LOHp. In [37], a similar setup has been included using poroelastic materials for 2D geometries. This model configuration is more complex compared to the previous ones





(a) Primary and reflected waves at receiver r.



(b) Transmitted waves at receiver t.

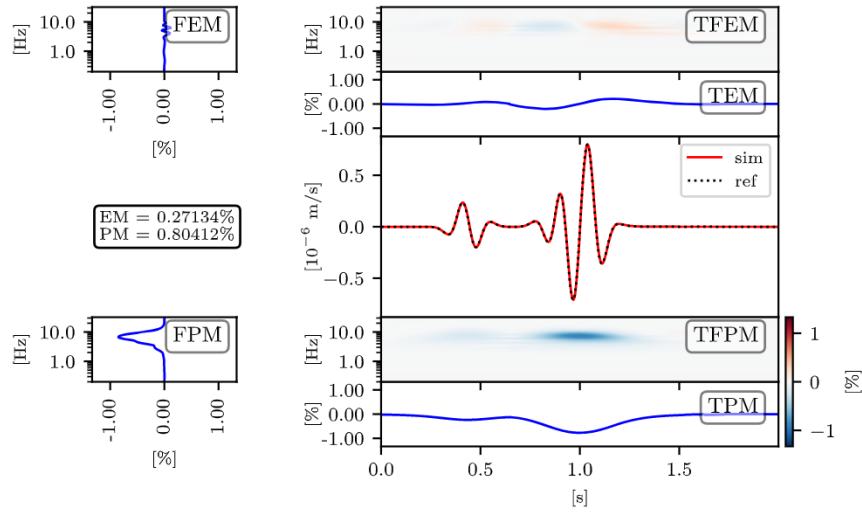
**Fig. 6.** Detailed misfit plots for the solid particle velocity component  $u$  at the receivers  $r$  and  $t$  for the contact of two half-spaces problem. For an explanation of the abbreviations, see Fig. 4. The materials for each half-space have different wave speeds, but the unstructured mesh has the same resolution everywhere, which explains the slightly increased misfit at receiver  $t$ .

since it contains both the internal material interface and the free surface at the same time. Since there is no analytical or semi-analytical solution for this model, we compare our SeisSol solution with a reference solution computed by the FD method [37].

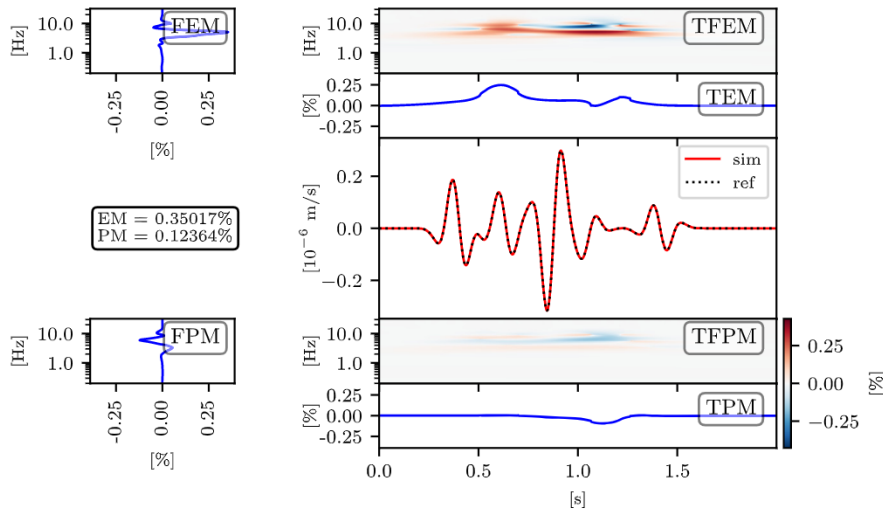
Fig. 8 shows the geometry and the source–receiver configuration. We consider a 500 m-thick layer atop a homogeneous half-space. We slightly modify the material parameters from Table 7 ( $z > 0$  for the half-space and  $z < 0$  for the layer). Since we are interested in a realistic scenario, we consider a fluid with *non-zero viscosity*:  $\nu = 0.001$  Pa.s. Additionally, we set the permeabilities to  $6 \cdot 10^{-12}$  m<sup>2</sup> and  $1 \cdot 10^{-12}$  m<sup>2</sup> in the half-space and the layer, respectively. We place an explosive source at (0 m, 0 m, 1010 m). We use a Gabor-type source time function:

$$s(t) = \cos(\omega \cdot T + \psi) \cdot \exp(-(\omega \cdot T / \gamma)^2),$$

where  $T = t - t_0$ ,  $\omega = 2 \cdot \pi \cdot f_0$  with parameters:  $f_0 = 0.5$ ,  $\gamma = 0.25$ ,  $\psi = 0$ ,  $t_0 = 0.25$ . With this choice of parameters, we excite waves with a flat amplitude spectrum up to  $\approx 10$  Hz. To be able to compare to a 2D solution, we approximate a line source in SeisSol by 401 point sources at  $(0, k \cdot 50, 1010)$ ,  $k \in \{-200, \dots, 200\}$ . The simulated time window is 5 s long to allow waves to pass through all receivers.

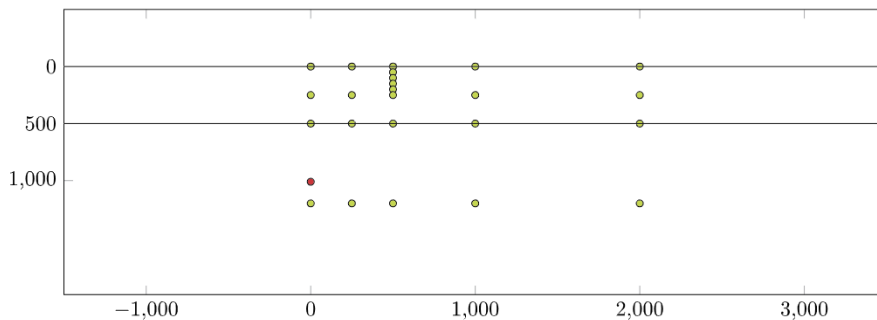


(a) Waveform at the free surface (receiver 0.5m).

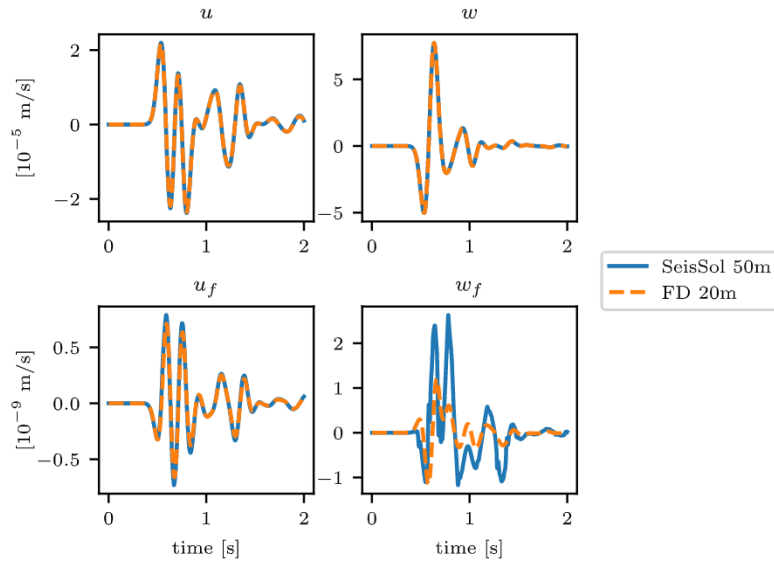


(b) Waveform at depth (receiver 500m).

**Fig. 7.** Detailed misfit plots for the solid particle velocity  $u$  at the receiver at the surface and at depth for the free surface problem. For an explanation of the abbreviations, see Fig. 4.



**Fig. 8.** Geometry of the LOHp problem: Source in red and receivers in green. We place receivers at the free surface, in the middle of the layer, 10 m above and below the interface and 200 m below the source. In the horizontal direction, receivers are 0 m, 250 m, 500 m, 1,000 m and 2,000 m away from the source. Instead of two receivers at the interface, only one is plotted. Under the receiver at  $x = 500$ , we place a line of receivers from the free surface to the middle of the layer at 5 m distance to see how the solution varies from the free surface into the layer. For this line, most receivers are omitted for readability.



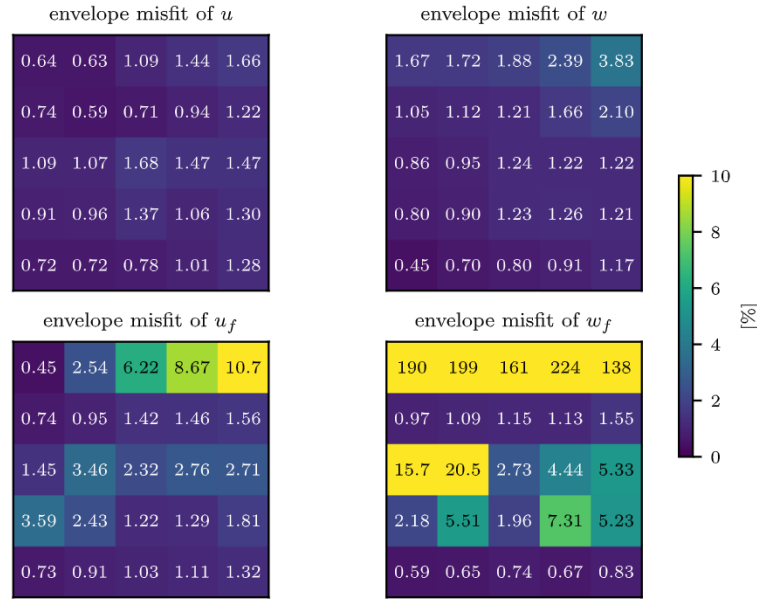
**Fig. 9.** Horizontal and vertical component of the solid and relative fluid velocities at (500 m, 0 m, 0 m) for the LOHp problem calculated by SeisSol with characteristic edge length 50 m method and FD method with grid spacing 20 m.

First, we use a standard resolution, i.e. a mesh with characteristic edge length determined according to Käser et al. [70], who thoroughly analysed the accuracy of ADER-DG for elastic wave propagation. Meshes built according to these rules provided reliable results in the previous benchmarks (Sections 6.2 to 6.4). For the LOHp model, the standard resolution yields excellent agreement between the SeisSol and the FD solutions only for the solid particle velocities. However, we find discrepancies with respect to the relative fluid velocities near the free surface and near the interface (cf. Fig. 9). Note that in the previous tests with an internal material interface or a free surface (Sections 6.3 and 6.4), we could not evaluate the accuracy of the relative fluid velocities due to limitations in the reference solution. Therefore, now we use finer resolutions to verify that both methods converge to the same solution even for the relative fluid velocity at and near the free surface (cf. Fig. 11). We first discuss the results with standard resolution and then with the fine resolution.

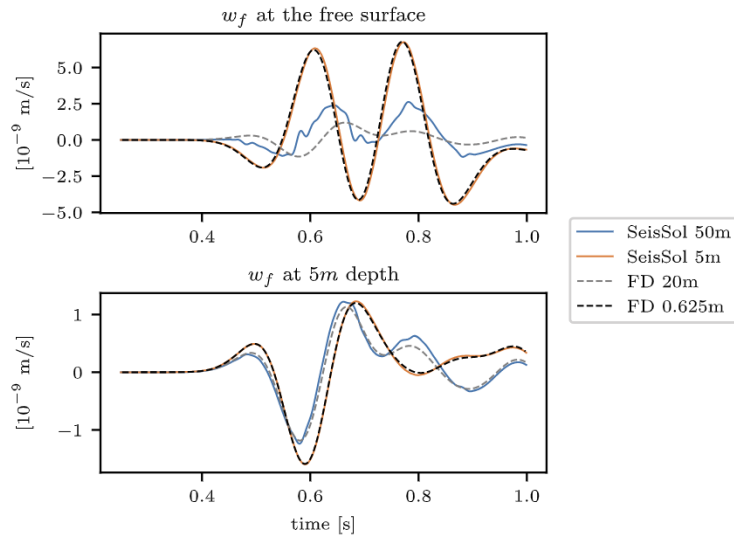
First, we consider SeisSol solutions for a characteristic edge length of 50 m in the cuboid  $[-500, 2500] \times [-1500, 1500] \times [0, 1500]$ . Furthermore, the mesh is refined towards the source and coarsened towards the boundary. As the reference solution, we consider FD solution for grid spacing of 20 m. Because we consider a 2D problem,  $y$ -components of the solid and relative fluid velocities ( $v, v_f$ ) are zero. In Fig. 9 we compare the waveforms obtained with SeisSol and the FD code for a selected receiver at the free surface. A visual comparison reveals good agreement for  $u, w$  and  $u_f$ , however, it also reveals substantial differences for  $w_f$ .

Next, we present a quantitative error analysis. Since we want to compare a large number of receivers, we concentrate on the single-valued envelope misfit (EM) only, while considering the FD solution as a reference. Fig. 10 summarises the EM values for the vertical and horizontal components of the solid and relative fluid velocities at the grid of receivers, shown in Fig. 8. In the case of solid particle velocities, we see excellent agreement (EM below 4%) between the SeisSol and FD solutions. We note that a slightly larger error is expected for receivers further away from the source due to the cumulative character of numerical errors. We also note that the receivers directly at the surface and close to the material interface have a slightly worse match than the ones in the middle of the layer or below the source independent of their distance to the source. However, in the case of the relative fluid velocities, we observe agreement comparable to that for solid particle velocities only in the middle of the layer and below the source. The misfits for the vertical component near the material interface and the free surface are unacceptably large. The mismatch at the free surface is very high, such that the waveforms do qualitatively differ (cf. Fig. 9).

We assume that the above-mentioned disagreement between the FD and SeisSol solutions is due to the fact that the slow P-wave is poorly resolved in the close vicinity of the free surface and material interface in one or both methods. In the low-frequency regime, the slow P-wave behaves as a diffusive wave, having very small wavelength compared to the fast P-wave and S-wave, and attenuates very quickly with distance from its origin. Dutta and Odé [71] show that when a fast P- or S-wave impinges on a material interface or free surface, as part of the partitioning of energy, mode conversion to a slow P-wave occurs, and its generation draws energy from the propagating wave process. However, the relative fluid velocities are  $\approx 5$  orders of magnitude smaller than the solid particle velocities and thus this inaccuracy does not affect the accuracy of the solid particle velocities. Moreover, because of the diffusive character of the slow P-wave, the slow P-wave remains in the vicinity of the interface during time scales for seismic wave propagation. Therefore, we do not observe differences at the receivers at depths much greater than the characteristic diffusion length [72], which is  $\approx 0.7$  m for our configuration.



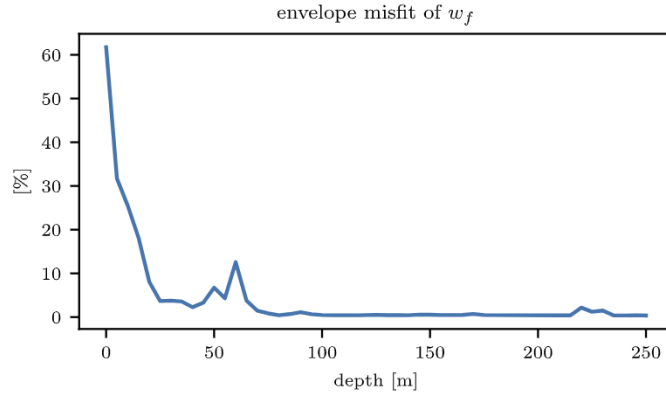
**Fig. 10.** Comparison of the EM between the SeisSol and FD solutions at the grid of receivers for the LOHp problem. The first row in each matrix represents the receivers at the free surface. The second row has the receivers in the middle of the layer. In the next two rows are the receivers a little above and below the interface. The last row gathers all receivers below the source.



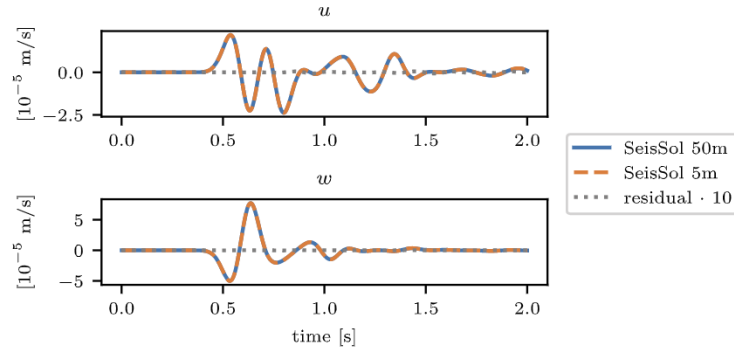
**Fig. 11.** Comparison of the waveform of  $w_f$  for the LOHp problem directly at the free surface and 5 m below it. We plot SeisSol solutions with characteristic edge lengths of 5 m and 50 m and FD solutions with grid spacings of 20 m and 0.625 m. SeisSol and FD solutions for the coarse resolutions do not match, but the ones obtained on refined meshes match.

To overcome the observed discrepancies, we refined both the SeisSol and the FD resolutions to see, if we can also resolve the slow P-wave accurately. For FD, we use a grid spacing of 0.625 m, based on a series of simulations with gradually finer grid resolution to verify that the solution converged (see Fig. B.18 in Appendix B). For SeisSol we refined the mesh at the free surface to a characteristic edge length of only 5 m. For computational reasons, we restricted the computation time to 2 s. Fig. 11 reveals that the qualitative behaviour of the waveform changes considerably with grid spacing/mesh refinement. If we compare the refined solutions, we conclude that both numerical solvers converge to the same solution. Finally, we would like to remark that the qualitative behaviour changes rapidly between the receiver at the free surface and 5 m below the free surface, which indicates that the diffusive P-wave plays a significant role here. To further assess how the slow P-wave affects the quality of the solution near the free surface, we compare the SeisSol solutions for characteristic element lengths 50 m and 5 m for the vertical component of the relative fluid velocity ( $w_f$ ) at various depths (Fig. 12). We observe large EM values only above 70 m from the free surface. This means that the large differences





**Fig. 12.** Envelope misfit between the SeisSol solutions with 50 m resolution and SeisSol solutions with 5 m refinement towards the free surface for the LOHp problem.



**Fig. 13.** Comparison of the solid particle velocities at (500 m, 0 m, 0 m) for the LOHp problem on meshes with different characteristic edge lengths with SeisSol.

are, in fact, concentrated in a layer with a thickness comparable with characteristic element length (50 m). This justifies our assumption that the slow P-wave was the source of mismatch at the free surface.

To conclude, we remark that the slow P-wave at the free surface can be simulated accurately with both, the here presented SeisSol extension and the 2D FD approach, but both methods need an extremely fine grid to accurately resolve the relative fluid velocity at the free surface. For most practical scenarios, the simulation at the coarse level will suffice, because:

- If we compare the solid particle velocities at the free surface for different mesh resolutions (cf. Fig. 13), we do not see a difference. We conclude that at the free surface, the solid particle velocities are not affected by the slow P-wave. This means for a seismic simulation, where one is interested in ground motion, the direct effect from the slow P-wave at the free surface is negligible.
- The slow P-wave only affects the solution at small spatial scales. Within the volume, all reflected waves are correctly simulated, even with coarse resolutions. If the relative fluid velocities are needed at a high resolution, another approach, e.g. by solving a diffusion equation, might be better suited.

## 7. Performance

SeisSol is optimised for large-scale simulations on supercomputers. Such simulations can require meshes consisting of several hundred millions of elements to resolve all phenomena accurately. Hence the number of unknowns can reach  $10^{11}$  or more (e.g. [15–17]). Therefore, we present a performance and scalability analysis of our extension of SeisSol towards poroelastic wave propagation. All experiments are carried out on SuperMUC-NG ( $2 \times$  Intel Xeon Platinum 8174 with 48 cores @ 2.5 GHz, 96 GB RAM per node [73]), which is installed at the Leibniz Supercomputing Centre, Garching, Germany.

### 7.1. Implementation using YATeTo

The space-time predictor algorithm as presented in Section 5 has proven to produce accurate simulation results, see Section 6. Its implementation relies heavily on the code generator YATeTo [65]. Algorithm 3 is already formulated as a

```

for n in range(N, -1, -1):
    for o in range(numberOfQuantities-1, -1, -1):
        kernels.append(stp['kpt'] <= stp['kpt'] + selectModes(n) ['kl']
            * selectQuantity(o) ['pq'] * stpRhs['lqu'] * Zinv(o) ['ut'])

```

Listing 1: Example of a tensor contraction, which appears in the space-time predictor solver, implemented using YATeTo. This example shows lines 2 to 6 of Algorithm 3.

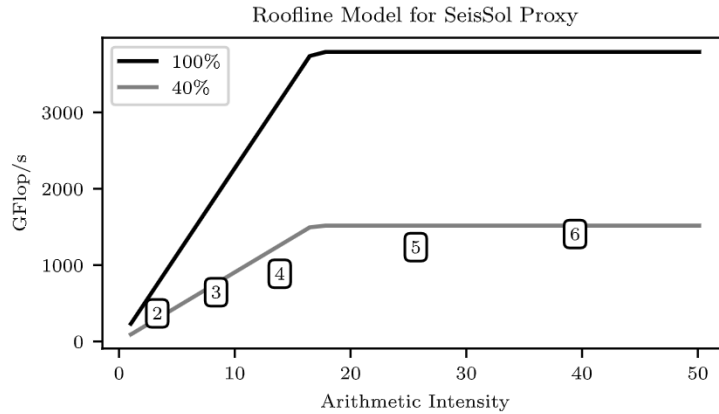


Fig. 14. Roofline model for the SeisSol performance proxy for different polynomial degrees with maximal attainable performance and 40% roofline.

sequence of tensor operations. YATeTo provides a domain-specific language, embedded into Python, to express these tensor operations in Einstein sum convention. Listing 1 shows an example of how parts of Algorithm 3 are implemented in YATeTo. Here `selectModes(n)` is a matrix, which extracts the basis functions in  $(B_{n-1}, B_n]$ , similarly, `selectQuantity(o)` selects the  $o$ th quantity and `Zinv(o)` is the matrix  $(Z - E_{00}^* I)^{-1}$ . YATeTo then builds an abstract syntax tree for these tensor operations and maps the tensor contractions to matrix-matrix multiplications. Specialised code-generators are available for these matrix-matrix multiplications. For processors of the Intel Skylake generation, we use a combination of `libxsmm` [30] for dense-dense multiplications and `PSpMM`<sup>3</sup> for multiplications with sparse matrices. These backends are used to generate the operational code, which is then used as compute kernel during the simulation phase.

## 7.2. Roofline model

First, we examine the single-node performance of the new back-substitution algorithm for the space-time predictor (i.e., Algorithm 3). The roofline model in Fig. 14 provides an overview of how well our implementation of the space-time predictor utilises the available computer resources [74]. It provides insight, whether a computation is memory or compute-bound and thus also gives hints where to further optimise.

We utilise a SeisSol performance proxy application, which executes the compute kernels on random data and omits other aspects such as I/O or communication [65,16]. Node-level performance is subject to variation across nodes. Therefore, we first use the `likwid` suite [75] to measure the theoretical node performance. Running the test on 10 nodes individually, we obtain a mean floating-point performance of  $3,792.9 \text{ GFLOPs}^{-1}$  and a memory bandwidth of  $226.8 \text{ GBs}^{-1}$ . Then we use the SeisSol performance proxy on 10 nodes individually with  $1 \cdot 10^6$  cells for 10 time steps. The results in Fig. 14 show that the performance of the SeisSol proxy roughly follows the 40% roofline. The maximum performance of  $1,406.7 \text{ GFLOPs}^{-1}$  is achieved for polynomial degree 6. With a high arithmetic intensity, our implementation is compute-bound for polynomial degrees 5 and 6. Although our approach attains a reasonable performance on SuperMUC-NG, it does not achieve performance similar to the kernels for elastic wave propagation ( $2,241 \text{ GFLOPs}^{-1}$  for polynomial degree 6 [65]). Architecture-specific backends for general matrix-matrix multiplications ( $C = \alpha A \cdot B$ ) are the key part of our compute kernels. The GEMM generator from `libxsmm`, does only support  $\alpha = 1$ , whereas in our case we need  $\alpha = \Delta t \neq 1$ . Hence, a fallback to a standard `for` loop is needed for the scalar multiplication.

Still, for polynomial degree 6, our approach is about a factor of 10 faster compared to an ideally performing LU solver. From Table 2, we observe that our approach requires only 4% of the floating-point operations needed for a back-substitution with precomputed LU decomposition. If that solution was perfectly implemented to achieve full performance, time to solution with our approach ( $T_{STP}$ ) would still be only 11% of time to solution with an LU decomposition ( $T_{LU}$ ):

<sup>3</sup> <https://github.com/peterwauligmann/PSpMM>.

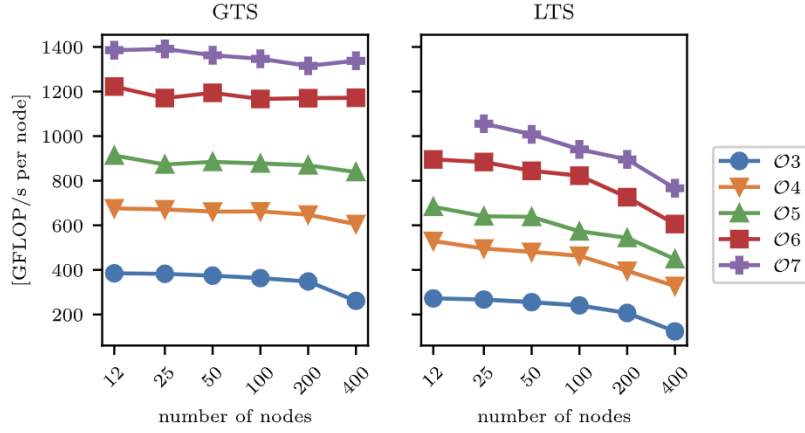


Fig. 15. Parallel efficiency for global and local time stepping using the LOHp model with  $7.33 \cdot 10^6$  elements on SuperMUC-NG.

$$T_{STP} = \frac{0.04 \cdot \#OP_{LU}}{0.37 \cdot \text{performance}_{LU}} \approx 0.108 \cdot \frac{\#OP_{LU}}{\text{performance}_{LU}} = 0.108 \cdot T_{LU}.$$

### 7.3. Scaling

SeisSol adopts a hybrid MPI+OpenMP parallelisation strategy. Among MPI ranks, we parallelise using graph-based mesh partitioning. In the initialisation phase, the elements of the mesh are distributed to the available MPI ranks, such that the load per rank is equally distributed. We distinguish between local (LTS) and global (GTS) time stepping. For GTS, the workload per element is homogeneous. For LTS, elements that need a smaller time step are updated more often than others, consequently, these elements generate a higher workload. Hence, respective element weights are provided for mesh partitioning. Within each rank, we use OpenMP to assign the available elements to compute cores. A dedicated thread is reserved for asynchronous I/O and communication between ranks [16,17].

We use the LOHp benchmark (cf. Section 6.5) for a strong scaling test. Since we do not need to compare with a reference 2D solution, we consider only a single point source. We set the final time to 0.1 s to test LTS and to 0.01 s to test GTS. We scale from 12 to 400 nodes of SuperMUC-NG using a mesh with 7,334,942 elements. The results are plotted in Fig. 15.

In the GTS results, we observe nearly constant node performance, which implies we efficiently use the available parallel resources. With 400 nodes, each node computes less than 20,000 elements, which is remarkably little compared to non-poroelastic SeisSol applications. For example, Krenz et al. use more than 150,000 elements per node [17] for their largest mesh in a SeisSol simulation with an elastic-acoustic material model. We attribute this to the higher workload per element, which is due to the more complex space-time predictor. With a peak performance of  $1,385 \text{ GFLOP s}^{-1}$ , we achieve roughly the same performance as measured with the proxy in Section 7.2.

For the LTS results, we observe that the absolute speed is slower than for GTS, which is expected due to the more complicated LTS scheme. We also observe that the scaling is not as good as with GTS and decays with increasing order. Still, for polynomial degree 6, we obtain  $1,056 \text{ GFLOP s}^{-1}$  on 25 nodes and reach  $766 \text{ GFLOP s}^{-1}$  on 400 nodes, which resembles a parallel efficiency of  $\approx 72.5\%$ . More importantly, by using LTS, time to solution is reduced by a factor of 6 to 10 compared to GTS, depending on the number of nodes and convergence order.

## 8. Discussion

### 8.1. Comparison to elastic wave propagation kernels

The poroelastic material model is inherently computationally more expensive than the elastic model. First of all, we increase the number of quantities ( $Q$ ) from 9 to 13, thus, the number of total DOFs in a simulation increases. Secondly, the space-time predictor, to compute the predicted element-local solution with a stiff source term, is substantially more complicated than the Cauchy-Kovalewski procedure used in the elastic case. For the predictor step, the elastic kernel requires 0.477 million floating-point operations with polynomials of degree 6. For the poroelastic model, the predictor kernel requires 1.087 million floating-point operations. This increase in computational workload can be attributed to the increased number of quantities, but also to the source term, which is absent in the elastic case. Furthermore, the poroelasticity kernel does not achieve the same performance as the elasticity kernel (cf. Section 7.2). In conclusion, we estimate that the per-element cost of a simulation using poroelastic materials is about 3.6 times higher than a simulation with elastic materials. Naturally, this does not include differences in time step size (e.g., in case of different P-wave speeds for poroelastic and elastic materials), mesh refinement requirements (consider, e.g., the refinement necessary for resolving a slow P-wave, as in the LOHp scenario) or in parallel scalability (where the higher per-element costs may be beneficial).

## 8.2. Limitations

With applications using seismic wavefield synthetics up to  $\approx 10$  Hz in mind, we focus on the low-frequency case. If an application requires the high-frequency regime, i.e. simulation of waves with frequencies comparable or larger than Biot's frequency (tens of Hz to hundreds of kHz for geo-reservoirs), the frequency-dependent permeability and resistive friction ( $b = \kappa/\nu$ ) have to be taken into account by Darcy's law in the equations of motion. Gregor et al. [37] show how to incorporate the high-frequency case in the 2D FD framework.

An additional limitation is, that, in the derivation of our scheme, we assumed constant material parameters per element. We could achieve subcell resolution if we computed the second integral in Equation (3) by quadrature on each element as detailed in [76]. However, in this case, the stiffness matrices are not matrices anymore but 3D tensors. The scheme would become more complex and would require careful performance evaluation and optimisation.

## 8.3. Future GPU and multi-physics implementation

SeisSol is currently available as a CPU and a GPU version, which share a large portion of the codebase. All compute kernels of SeisSol are expressed in a domain-specific language (cf. Section 7.1), which is then translated to machine code, for either CPUs or GPUs [26]. Therefore, it does not pose a major challenge to run also simulations with the poroelastic model on a GPU cluster, but a careful performance study has to be done again.

Up to now only point sources have been considered. For physics-based earthquake simulations and to study fault-fluid interaction we have to consider more complex sources. These will include moment-tensor based double-couple point source implementations, kinematic finite earthquake source models [77] and non-linear earthquake rupture dynamics taking the interaction of frictional shear fracture and propagating waves into account [78,22,23]. To do so, a fault will be embedded as an internal boundary in the mesh. At this interface, we do not just exchange information by numerical fluxes, but instead, we solve a nonlinear friction problem. Dynamic rupture simulations in (visco-)elastic media can already be simulated with SeisSol. To combine this source mechanism with poroelastic materials, the coupling between poroelastic parameters and parameters of friction laws have to be investigated. To fully capture the interaction of fluids, fault slip and seismic waves, additional multi-physics interactions can be accounted for that describe the thermal pressurisation of pore fluids [79–81] during earthquake rupture. The thermal pressurisation model was recently implemented in SeisSol [82]. In the context of geo-reservoirs, pressure increase can drive fluid flow and in turn govern earthquake dynamics e.g. [83], however, a holistic method allowing to couple poroelastic effects on wave propagation and on rupture dynamics at the same time is currently not available.

## 8.4. Other applications

The solution approach presented in Section 5 is general and can be applied to a broader class of problems described by linear hyperbolic PDEs with a stiff reactive source term. In our derivations, we made two assumptions on the sparsity pattern of the stiffness matrices  $K^\alpha$  (cf. Section 5.1) and the source matrix  $E$ . The matrix  $K^\alpha$  is problem independent, but the matrix  $E$  depends on the PDE which we consider. For example, in the case of viscoelastic attenuation, it takes an upper triangular form [19], just as in our poroelastic case.

An additional applicable example are the damped Maxwell equations, which model the interaction of electric ( $\mathcal{E}$ ) and magnetic ( $\mathcal{H}$ ) fields [84]:

$$\epsilon \frac{\partial \mathcal{E}}{\partial t} = \nabla \times \mathcal{H} - \sigma \mathcal{E} + j \quad \mu \frac{\partial \mathcal{H}}{\partial t} = \nabla \times \mathcal{E}$$

Here,  $\epsilon$  is the relative electric permeability,  $\mu$  the relative magnetic permeability and  $\sigma$  the conductivity. The current density  $j$  is a source term comparable to seismic sources in the context of poroelasticity. With a non-zero conductivity, the Maxwell equations contain a possibly stiff source term. In order to apply our proposed scheme, we expand the rotation operator and can write down the equation in a similar fashion as Equation (1). The source matrix is then upper triangular again, such that we can apply the space-time DG method and Algorithm 3 to solve the resulting linear system of equations.

## 9. Conclusion

We have derived a new efficient algorithm for solving the system of equations, which arises from the discretisation of the governing equations of wave propagation in poroelastic media using the space-time variant of ADER-DG. The algorithm relies on a block-wise back-substitution procedure, which can be efficiently implemented by chains of matrix-matrix products. The implementation is thoroughly validated against reference solutions. Detailed numerical tests revealed that for viscous problems with a free surface or internal material interface, a standard mesh resolution (based on an accuracy analysis of the elastic problem) is sufficient for the solid particle velocities. However, if the relative fluid velocities have to be calculated at or very close to the boundary, much finer spatial resolution is necessary. Performance and scalability experiments show that our method is suited to be used on recent supercomputers. With our focus on high-performance and scalability, we enable large-scale seismic simulations in poroelastic materials. Since the implementation is open-source software, we envision broad applicability of our work within the geophysics and engineering communities.



## CRedit authorship contribution statement

**Sebastian Wolf:** Investigation, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. **Martin Galis:** Investigation, Methodology, Software, Validation, Writing – review & editing. **Carsten Uphoff:** Methodology, Writing – review & editing. **Alice-Agnes Gabriel:** Conceptualization, Funding acquisition, Resources, Supervision, Writing – review & editing. **Peter Moczo:** Funding acquisition, Resources, Software, Supervision, Validation, Writing – review & editing. **David Gregor:** Investigation, Software, Validation, Writing – review & editing. **Michael Bader:** Conceptualization, Funding acquisition, Resources, Supervision, Writing – review & editing.

## Declaration of competing interest

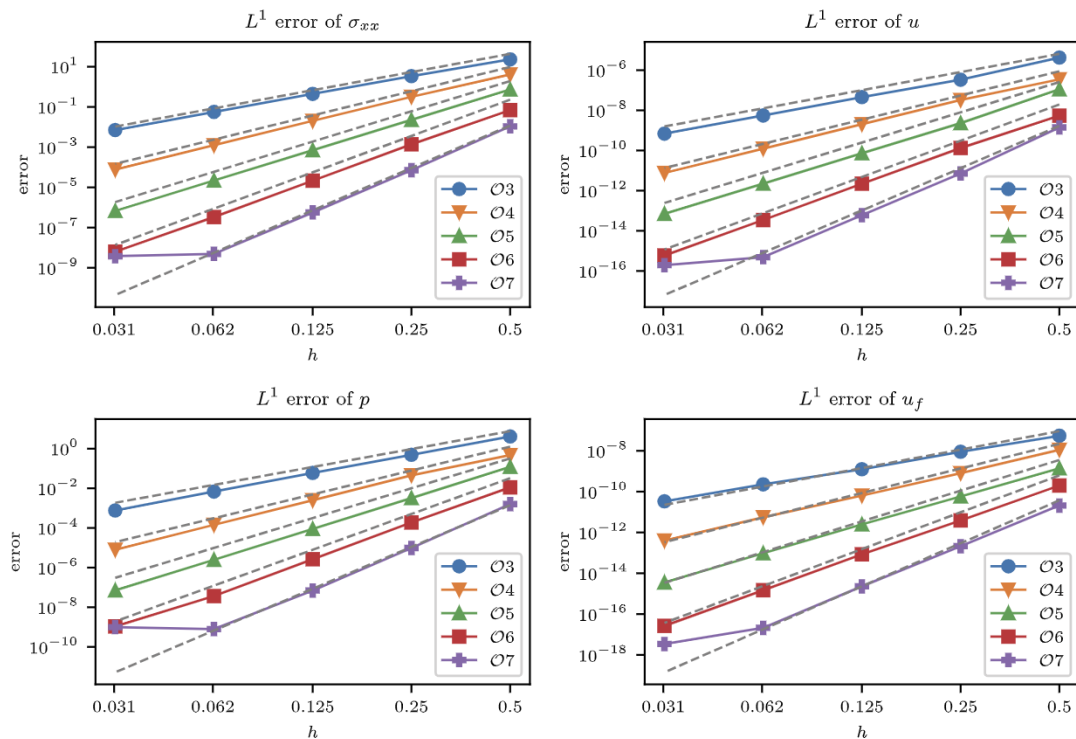
The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Data availability

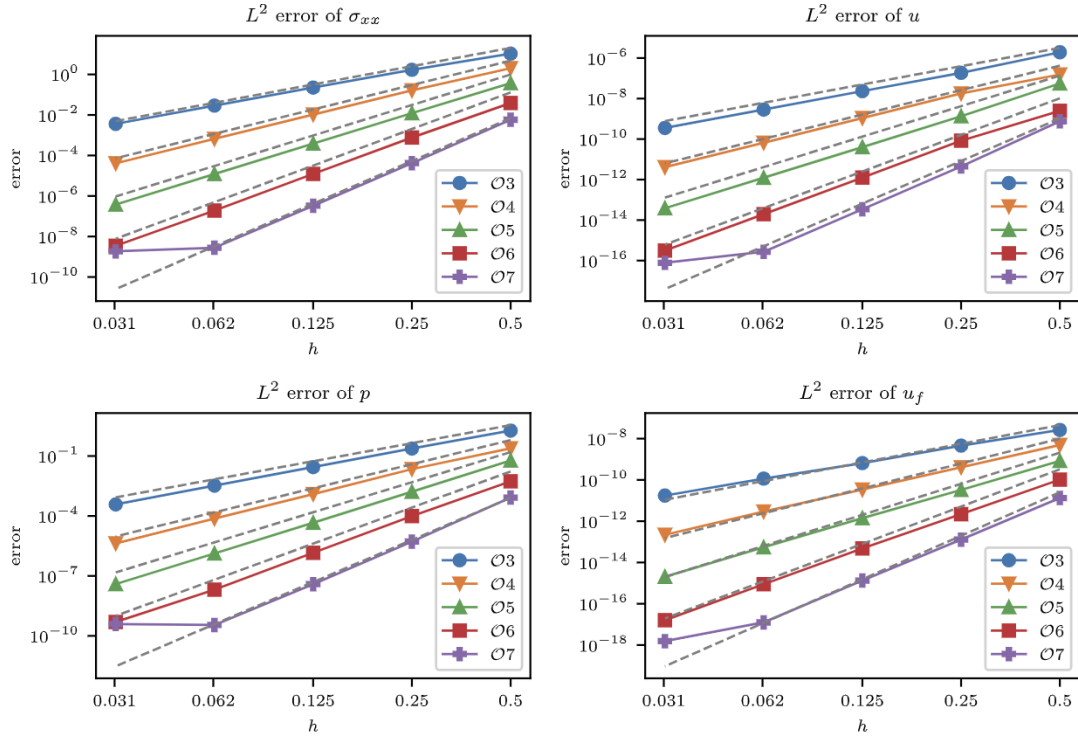
The source code of SeisSol is available as open-source software under <https://github.com/SeisSol/SeisSol>. The model descriptions, simulation outputs and reference data can be found under <https://doi.org/10.5281/zenodo.5236133>.

## Acknowledgements

The presented work has been funded by the European Union's Horizon 2020 Research and Innovation program (EN-ERXICO, grant agreement No. 828947). Computing resources were provided by the Leibniz Supercomputing Centre (project no. pr83no on SuperMUC-NG) and KAUST Supercomputing Laboratory (project no. k1343 on Shaheen II). C.U. and A.-A.G. acknowledge support by the European Union's Horizon 2020 Research and Innovation Programme under ERC StG TEAR, no. 852992 and the German Research Foundation (DFG) (grants no. GA 2465/2-1, GA 2465/3-1). M.G. and A.-A.G. acknowledge support by KAUST-CRG (grant no. ORS-2017-CRG6 3389.02). S.W. and M.B. acknowledge support by KONWIHR (project "Optimisation of SeisSol for Large Scale Simulations of Induced Earthquakes"). M.G., P.M. and D.G. acknowledge support by the Slovak Research and Development Agency under the contract APVV-15-0560 (project ID-EFFECTS).



**Fig. A.16.** Convergence plots for selected quantities of the planar wave convergence test in the  $L^1$  norm. The expected convergence order is plotted in grey dashed lines. For O7 we are close to machine precision on the finest mesh.



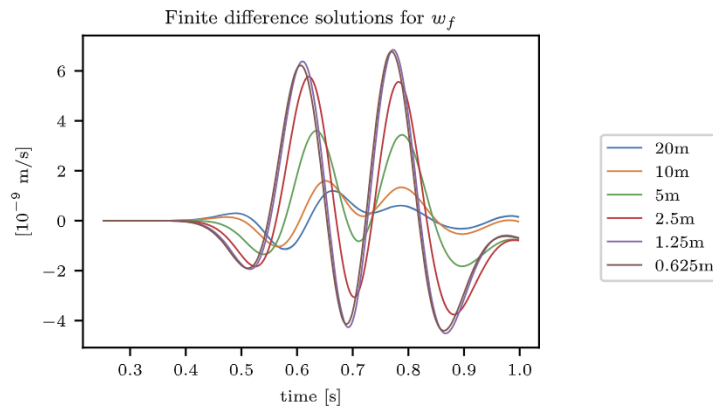
**Fig. A.17.** Convergence plots for selected quantities of the planar wave convergence test in the  $L^2$  norm. The expected convergence order is plotted in grey dashed lines. For  $O7$  we are close to machine precision on the finest mesh.

## Appendix A. Convergence results in the $L^1$ and $L^2$ norm

In Section 6.1, we show the high-order convergence of our scheme with a planar wave scenario. Figs. A.16 and A.17 show the convergence results for the same setup in the  $L^1$  and  $L^2$  norm. Also in these norms, we observe the same convergence behaviour as for the  $L^\infty$  norm.

## Appendix B. Convergence of the FD solutions

For the LOHp model (Section 6.5), we use a 2D FD code as a reference. Because the solutions obtained with a coarse resolution did not resolve the slow P-wave on the vertical component of relative fluid velocity ( $w_f$ ) accurately enough, we conducted a convergence study for the FD solutions. We used grid spacings of 20 m, 10 m, 5 m, 2.5 m, 1.25 m and 0.625 m. Fig. B.18 shows only a small difference between the solutions for grid spacings 1.25 m and 0.625 m, indicating that the solution converged. We note that we choose the solution for grid spacing 0.625 m as the reference solution in Section 6.5.



**Fig. B.18.** Convergence of the vertical component of the relative fluid velocity ( $w_f$ ) obtained with different grid spacings using the FD method.

## References

- [1] J.M. Carcione, C. Morency, J.E. Santos, Computational poroelasticity — a review, *Geophysics* (2010), <https://doi.org/10.1190/1.3474602>, <https://library.seg.org/doi/abs/10.1190/1.3474602>.
- [2] M.A. Biot, Theory of deformation of a porous viscoelastic anisotropic solid, *J. Appl. Phys.* 27 (1956) 459–467, <https://doi.org/10.1063/1.1722402>, <https://aip.scitation.org/doi/10.1063/1.1722402>.
- [3] M.A. Biot, Theory of propagation of elastic waves in a fluid-saturated porous solid. I. Low-frequency range, *J. Acoust. Soc. Am.* 28 (1956) 168–178, <https://doi.org/10.1121/1.1908239>, <https://asa.scitation.org/doi/10.1121/1.1908239>.
- [4] M.A. Biot, Theory of propagation of elastic waves in a fluid-saturated porous solid. II. Higher frequency range, *J. Acoust. Soc. Am.* 28 (1956) 179–191, <https://doi.org/10.1121/1.1908241>, <https://asa.scitation.org/doi/10.1121/1.1908241>.
- [5] M.A. Biot, Mechanics of deformation and acoustic propagation in porous media, *J. Appl. Phys.* 33 (1962) 1482–1498, <https://doi.org/10.1063/1.1728759>, <https://aip.scitation.org/doi/10.1063/1.1728759>.
- [6] T.J. Plona, Observation of a second bulk compressional wave in a porous medium at ultrasonic frequencies, *Appl. Phys. Lett.* 36 (1980) 259–261, <https://doi.org/10.1063/1.91445>, <https://aip.scitation.org/doi/10.1063/1.91445>.
- [7] J.G. Berryman, Confirmation of Biot's theory, *Appl. Phys. Lett.* 37 (1980) 382–384, <https://doi.org/10.1063/1.91951>, <https://aip.scitation.org/doi/abs/10.1063/1.91951>.
- [8] J.M. Carcione, *Wave Fields in Real Media*, third edition, Elsevier, Oxford, 2015, <http://www.sciencedirect.com/science/article/pii/B9780080999999999921>.
- [9] J. de la Puente, M. Dumbser, M. Käser, H. Igel, Discontinuous Galerkin methods for wave propagation in poroelastic media, *Geophysics* 73 (2008) T77–T97, <https://doi.org/10.1190/1.2965027>, <https://library.seg.org/doi/10.1190/1.2965027>.
- [10] W.H. Reed, T.R. Hill, Triangular mesh methods for the neutron transport equation, Technical Report LA-UR-73-479; CONF-730414-2, Los Alamos Scientific Lab, N.Mex. (USA), 1973, <https://www.osti.gov/biblio/4491151-triangular-mesh-methods-neutron-transport-equation>.
- [11] B. Cockburn, C.-W. Shu, TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws II: general framework, *Math. Comput.* 52 (1989) 411–435, <https://doi.org/10.2307/2008474>, <https://www.jstor.org/stable/2008474>.
- [12] J.S. Hesthaven, T. Warburton, Nodal high-order methods on unstructured grids: I. Time-domain solution of Maxwell's equations, *J. Comput. Phys.* 181 (2002) 186–221, <https://doi.org/10.1006/jcph.2002.7118>, <https://www.sciencedirect.com/science/article/pii/S0021999102971184>.
- [13] M. Dumbser, M. Käser, An arbitrary high-order discontinuous Galerkin method for elastic waves on unstructured meshes - II. The three-dimensional isotropic case, *Geophys. J. Int.* 167 (2006) 319–336, <https://doi.org/10.1111/j.1365-246X.2006.03120.x>, <https://academic.oup.com/gji/article-lookup/doi/10.1111/j.1365-246X.2006.03120.x>.
- [14] C. Burstedde, O. Ghattas, M. Gurnis, T. Isaac, G. Stadler, T. Warburton, L. Wilcox, Extreme-scale AMR, in: *SC '10: Proceedings of the 2010 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis*, 2010, pp. 1–12, ISSN: 2167-4337.
- [15] A. Heinecke, A. Breuer, S. Rettenberger, M. Bader, A.-A. Gabriel, C. Pelties, A. Bode, W. Barth, X.-K. Liao, K. Vaidyanathan, M. Smelyanskiy, P. Dubey, Petascale high order dynamic rupture earthquake simulations on heterogeneous supercomputers, in: *SC '14: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 2014, pp. 3–14, ISSN: 2167-4337.
- [16] C. Uphoff, S. Rettenberger, M. Bader, E.H. Madden, T. Ulrich, S. Wollherr, A.-A. Gabriel, Extreme scale multi-physics simulations of the Tsunamigenic 2004 Sumatra Megathrust Earthquake, in: *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '17, ACM, New York, NY, USA, 2017, pp. 21:1–21:16, <https://doi.acm.org/10.1145/3126908.3126948>.
- [17] L. Krenz, C. Uphoff, T. Ulrich, A.-A. Gabriel, L.S. Abrahams, E.M. Dunham, M. Bader, 3D acoustic-elastic coupling with gravity: the dynamics of the 2018 Palu, Sulawesi earthquake and tsunami, in: *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '21, Association for Computing Machinery, New York, NY, USA, 2021, pp. 1–14.
- [18] M. Dumbser, M. Käser, E.F. Toro, An arbitrary high-order Discontinuous Galerkin method for elastic waves on unstructured meshes - V. Local time stepping and p-adaptivity, *Geophys. J. Int.* 171 (2007) 695–717, <https://doi.org/10.1111/j.1365-246X.2007.03427.x>, <https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1365-246X.2007.03427.x>.
- [19] M. Käser, M. Dumbser, J. de la Puente, H. Igel, An arbitrary high-order Discontinuous Galerkin method for elastic waves on unstructured meshes - III. Viscoelastic attenuation, *Geophys. J. Int.* 168 (2007) 224–242, <https://doi.org/10.1111/j.1365-246X.2006.03193.x>, <https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1365-246X.2006.03193.x>.
- [20] S. Wollherr, A.-A. Gabriel, C. Uphoff, Off-fault plasticity in three-dimensional dynamic rupture simulations using a modal Discontinuous Galerkin method on unstructured meshes: implementation, verification and application, *Geophys. J. Int.* 214 (2018) 1556–1584, <https://doi.org/10.1093/gji/ggy213>, <https://academic.oup.com/gji/article/214/3/1556/5017447>.
- [21] S. Wolf, A.-A. Gabriel, M. Bader, Optimization and local time stepping of an ADER-DG scheme for fully anisotropic wave propagation in complex geometries, in: V.V. Krzhizhanovskaya, G. Závodszy, M.H. Lees, J.J. Dongarra, P.M.A. Sloot, S. Brissos, J. Teixeira (Eds.), *Computational Science – ICCS 2020*, in: *Lecture Notes in Computer Science*, Springer International Publishing, Cham, 2020, pp. 32–45.
- [22] C. Pelties, J. de la Puente, J.-P. Ampuero, G.B. Brietzke, M. Käser, Three-dimensional dynamic rupture simulation with a high-order discontinuous Galerkin method on unstructured tetrahedral meshes, *J. Geophys. Res., Solid Earth* 117 (2012), <https://doi.org/10.1029/2011JB008857>, <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2011JB008857>.
- [23] C. Pelties, A.-A. Gabriel, J.-P. Ampuero, Verification of an ADER-DG method for complex dynamic rupture problems, *Geosci. Model Dev.* 7 (2014) 847–866, <https://doi.org/10.5194/gmd-7-847-2014>, <https://www.geosci-model-dev.net/7/847/2014/>.
- [24] T. Ulrich, A.-A. Gabriel, J.-P. Ampuero, W. Xu, Dynamic viability of the 2016 Mw 7.8 Kaikōura earthquake cascade on weak crustal faults, *Nat. Commun.* 10 (2019) 1–16, <https://doi.org/10.1038/s41467-019-09125-w>, <https://www.nature.com/articles/s41467-019-09125-w>.
- [25] K.H. Palgunadi, A.-A. Gabriel, T. Ulrich, J.A. López-Comino, P.M. Mai, Dynamic fault interaction during a fluid-injection-induced earthquake: the 2017 Mw 5.5 Pohang event, *Bull. Seismol. Soc. Am.* 110 (2020) 2328–2349, <https://doi.org/10.1785/0120200106>, <https://pubs.geoscienceworld.org/ssa/bssa/article-abstract/110/5/2328/588530/Dynamic-Fault-Interaction-during-a-Fluid-Injection>.
- [26] R. Dorozhinskiy, M. Bader, SeisSol on distributed Multi-GPU systems: CUDA code generation for the modal discontinuous Galerkin method, in: *The International Conference on High Performance Computing in Asia-Pacific Region, HPC Asia 2021*, Association for Computing Machinery, New York, NY, USA, 2021, pp. 69–82.
- [27] G. Gassner, M. Dumbser, F. Hindenlang, C. Munz, Explicit one-step time discretizations for discontinuous Galerkin and finite volume schemes based on local predictors, *J. Comput. Phys.* (2011), <https://doi.org/10.1016/j.jcp.2010.10.024>.
- [28] G.H. Golub, C.F. Van Loan, *Matrix Computations*, fourth edition, Johns Hopkins Studies in the Mathematical Sciences, The Johns Hopkins University Press, Baltimore, 2013, OCLC, ocn824733531.
- [29] L.S. Blackford, J. Demmel, J. Dongarra, I. Duff, S. Hammarling, G. Henry, M. Heroux, L. Kaufman, A. Lumsdaine, A. Petitet, R. Pozo, K. Remington, R.C. Whaley, An updated set of Basic Linear Algebra Subprograms (BLAS), *ACM Trans. Math. Softw.* 28 (2001) 135–151.
- [30] A. Heinecke, G. Henry, M. Hutchinson, H. Pabst, LIBSMM: accelerating small matrix multiplications by runtime code generation, in: *SC16: International Conference for High Performance Computing, Networking, Storage and Analysis*, IEEE, Salt Lake City, UT, 2016, pp. 981–991, <https://ieeexplore.ieee.org/document/7877162/>.



- [31] J. Diaz, A. Ezziani, Analytical solution for wave propagation in stratified poroelastic medium. Part II: The 3D case, arXiv:0807.4067 [physics], 2008, <http://arxiv.org/abs/0807.4067>.
- [32] F. Karpfinger, T.M. Müller, B. Gurevich, Green's functions and radiation patterns in poroelastic solids revisited, *Geophys. J. Int.* 178 (2009) 327–337, <https://doi.org/10.1111/j.1365-246X.2009.04116.x>, <https://academic.oup.com/gji/article/178/1/327/644784>.
- [33] J. Carcione, G. Quiroga-Goode, Some aspects of the physics and numerical modeling of Biot compressional waves, *J. Comput. Acoust.* 3 (1995) 261–280, <https://doi.org/10.1142/S0218396X95000136>.
- [34] C. Morency, J. Tromp, Spectral-element simulations of wave propagation in porous media, *Geophys. J. Int.* 175 (2008) 301–345, <https://doi.org/10.1111/j.1365-246X.2008.03907.x>, <https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1365-246X.2008.03907.x>.
- [35] S.K. Garg, A.H. Nayfeh, A.J. Good, Compressional waves in fluid-saturated elastic porous media, *J. Appl. Phys.* 45 (1974) 1968–1974, <https://doi.org/10.1063/1.1663532>, <https://aip.scitation.org/doi/10.1063/1.1663532>.
- [36] P. Moczo, D. Gregor, J. Kristek, J. de la Puente, A discrete representation of material heterogeneity for the finite-difference modelling of seismic wave propagation in a poroelastic medium, *Geophys. J. Int.* 216 (2019) 1072–1099, <https://doi.org/10.1093/gji/ggy412>.
- [37] D. Gregor, P. Moczo, J. Kristek, A. Mesgouez, G. Lefeuvre-Mesgouez, M. Kristekova, Subcell-resolution finite-difference modelling of seismic waves in Biot and JKD poroelastic media, *Geophys. J. Int.* 224 (2021) 760–794, <https://doi.org/10.1093/gji/ggaa454>.
- [38] D. Gregor, P. Moczo, J. Kristek, A. Mesgouez, G. Lefeuvre-Mesgouez, C. Morency, J. Diaz, M. Kristekova, Seismic waves in medium with poroelastic/elastic interfaces: a two-dimensional P-SV finite-difference modelling, *Geophys. J. Int.* 228 (2021) 551–588, <https://doi.org/10.1093/gji/ggab357>.
- [39] Y. Zhang, J. Gao, W. Han, Y. He, A discontinuous Galerkin method for seismic wave propagation in coupled elastic and poroelastic media, *Geophys. Prospect.* 67 (2019) 1392–1403, <https://doi.org/10.1111/1365-2478.12781>, <https://onlinelibrary.wiley.com/doi/abs/10.1111/1365-2478.12781>.
- [40] K. Shukla, J.S. Hesthaven, J.M. Carcione, R. Ye, J. de la Puente, P. Jaiswal, A nodal discontinuous Galerkin finite element method for the poroelastic wave equation, *Comput. Geosci.* 23 (2019) 595–615, <https://doi.org/10.1007/s10596-019-9809-1>.
- [41] Q. Zhan, M. Zhuang, Y. Fang, Y. Hu, Y. Mao, W.-F. Huang, R. Zhang, D. Wang, Q.H. Liu, Full-anisotropic poroelastic wave modeling: a discontinuous Galerkin algorithm with a generalized wave impedance, *Comput. Methods Appl. Mech. Eng.* 346 (2019) 288–311, <https://doi.org/10.1016/j.cma.2018.12.003>, <http://www.sciencedirect.com/science/article/pii/S0045782518306017>.
- [42] N.F. Dudley Ward, T. Lähivaara, S. Eveson, A discontinuous Galerkin method for poroelastic wave propagation: the two-dimensional case, *J. Comput. Phys.* 350 (2017) 690–727, <https://doi.org/10.1016/j.jcp.2017.08.070>, <https://www.sciencedirect.com/science/article/pii/S0021999117306587>.
- [43] J.M. Carcione, Energy balance and fundamental relations in dynamic anisotropic poro-viscoelasticity, *Proc. R. Soc. Lond., Ser. A, Math. Phys. Eng. Sci.* (2001), <https://doi.org/10.1098/rspa.2000.0669>, <https://royalsocietypublishing.org/doi/abs/10.1098/rspa.2000.0669>.
- [44] J.R. Franchi, *Petroleum Engineering Handbook: Volume I - General Engineering*, Society of Petroleum Engineers, 2006.
- [45] V. Saxena, M. Krief, L. Adam, *Handbook of Borehole Acoustics and Rock Physics for Reservoir Characterization*, Elsevier, Washington, WA, 2018.
- [46] J. de la Puente, Seismic wave simulation for complex rheologies on unstructured meshes, PhD, Ludwig-Maximilians-Universität München, Munich, 2008, <http://nbn-resolving.de/urn:nbn:de:bvb:19-80745>.
- [47] E.T. Chung, B. Engquist, Optimal Discontinuous Galerkin methods for wave propagation, *SIAM J. Numer. Anal.* 44 (2006) 2131–2158, <https://doi.org/10.1137/050641193>, <https://epubs.siam.org/doi/abs/10.1137/050641193>.
- [48] B. Rivière, S. Shaw, J.R. Whiteman, Discontinuous Galerkin finite element methods for dynamic linear solid viscoelasticity problems, *Numer. Methods Partial Differ. Equ.* 23 (2007) 1149–1166, <https://doi.org/10.1002/num.20215>, <https://onlinelibrary.wiley.com/doi/abs/10.1002/num.20215>.
- [49] J.D. De Basabe, M.K. Sen, M.F. Wheeler, The interior penalty discontinuous Galerkin method for elastic wave propagation: grid dispersion, *Geophys. J. Int.* 175 (2008) 83–93, <https://doi.org/10.1111/j.1365-246X.2008.03915.x>.
- [50] J. Diaz, M.J. Grote, Energy conserving explicit local time stepping for second-order wave equations, *SIAM J. Sci. Comput.* 31 (2009) 1985–2014, <https://doi.org/10.1137/070709414>, <https://epubs.siam.org/doi/abs/10.1137/070709414>.
- [51] V. Etienne, E. Chaljub, J. Virieux, N. Glinsky, An hp-adaptive discontinuous Galerkin finite-element method for 3-D elastic wave modelling, *Geophys. J. Int.* 183 (2010) 941–962, <https://doi.org/10.1111/j.1365-246X.2010.04764.x>.
- [52] L.C. Wilcox, G. Stadler, C. Burstedde, O. Ghattas, A high-order discontinuous Galerkin method for wave propagation through coupled elastic-acoustic media, *J. Comput. Phys.* 229 (2010) 9373–9396, <https://doi.org/10.1016/j.jcp.2010.09.008>, <https://www.sciencedirect.com/science/article/pii/S0021999110005024>.
- [53] P.F. Antonietti, I. Mazzieri, A. Quarteroni, F. Rapetti, Non-conforming high order approximations of the elastodynamics equation, *Comput. Methods Appl. Mech. Eng.* 209–212 (2012) 212–238, <https://doi.org/10.1016/j.cma.2011.11.004>, <https://www.sciencedirect.com/science/article/pii/S0045782511003410>.
- [54] I. Mazzieri, M. Stupazzini, R. Guidotti, C. Smerzini, SPEED: spectral elements in elastodynamics with Discontinuous Galerkin: a non-conforming approach for 3D multi-scale problems, *Int. J. Numer. Methods Eng.* 95 (2013) 991–1010, <https://doi.org/10.1002/nme.4532>, <https://www.onlinelibrary.wiley.com/doi/abs/10.1002/nme.4532>.
- [55] F. Peyrusse, N. Glinsky, C. Célis, S. Lanteri, A high-order Discontinuous Galerkin method for viscoelastic wave propagation, in: M. Azaiez, H. El Fekih, J.S. Hesthaven (Eds.), *Spectral and High Order Methods for Partial Differential Equations - ICOSAHOM 2012*, in: *Lecture Notes in Computational Science and Engineering*, Springer International Publishing, Cham, 2014, pp. 361–371.
- [56] E.D. Mercierat, N. Glinsky, A nodal high-order Discontinuous Galerkin method for elastic wave propagation in arbitrary heterogeneous media, *Geophys. J. Int.* 201 (2015) 1101–1118, <https://doi.org/10.1093/gji/ggv029>.
- [57] V.A. Titarev, E.F. Toro, ADER: arbitrary high order Godunov approach, *J. Sci. Comput.* 17 (2002) 609–618, <https://doi.org/10.1023/A:1015126814947>.
- [58] A. Reinarz, D.E. Charrier, M. Bader, L. Bovard, M. Dumbser, K. Duru, F. Fambri, A.-A. Gabriel, J.-M. Gallard, S. Köppel, L. Krenz, L. Rannabauer, L. Rezzolla, P. Samfass, M. Tavelli, T. Weinzierl, ExaHyPE: an engine for parallel dynamically adaptive simulations of wave problems, *Comput. Phys. Commun.* 254 (2020) 107251, <https://doi.org/10.1016/j.cpc.2020.107251>, <http://www.sciencedirect.com/science/article/pii/S001046552030076X>.
- [59] K. Duru, L. Rannabauer, A.-A. Gabriel, H. Igel, A new Discontinuous Galerkin method for elastic waves with physically motivated numerical fluxes, *J. Sci. Comput.* 88 (2021) 51, <https://doi.org/10.1007/s10915-021-01565-1>.
- [60] B. Cockburn, G.E. Karniadakis, C.-W. Shu, M. Griebel, D.E. Keyes, R.M. Nieminen, D. Roose, T. Schlick (Eds.), *Discontinuous Galerkin Methods: Theory, Computation and Applications*, *Lecture Notes in Computational Science and Engineering*, vol. 11, Springer, Berlin, Heidelberg, 2000, <http://link.springer.com/10.1007/978-3-642-59721-3>.
- [61] R.J. LeVeque, *Finite Volume Methods for Hyperbolic Problems*, *Cambridge Texts in Applied Mathematics*, Cambridge University Press, Cambridge, 2002, <https://ebookcentral-proquest-com.eaccess.ub.tum.de/lib/Munchentech/detail.action?docID=202080>.
- [62] J.S. Hesthaven, T. Warburton, *Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications*, *Texts in Applied Mathematics*, Springer-Verlag, New York, 2008, <https://www.springer.com/de/book/9780387720654>.
- [63] E.F. Toro, *Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction*, 3rd ed., Springer-Verlag, Berlin, Heidelberg, 2009, <https://ebookcentral-proquest-com.eaccess.ub.tum.de/lib/Munchentech/detail.action?docID=438075>.
- [64] A. Breuer, A. Heinecke, M. Bader, Petascale local time stepping for the ADER-DG finite element method, in: *2016 IEEE International Parallel and Distributed Processing Symposium, IPDPS, 2016*, pp. 854–863.
- [65] C. Uphoff, M. Bader, Yet another tensor toolbox for Discontinuous Galerkin methods and other applications, *ACM Trans. Math. Softw.* 46 (2020) 34:1–34:40, <https://doi.org/10.1145/3406835>.
- [66] T. Kolda, B.W. Bader, Tensor decompositions and applications, *SIAM Rev.* (2009), <https://doi.org/10.1137/0707011X>.



- [67] P. Moczo, J.P. Ampuero, J. Kristek, S.M. Day, M. Kristekova, P. Pazak, M. Galis, H. Igel, Comparison of numerical methods for seismic wave propagation and source dynamics - the SPICE code validation, in: Third International Symposium on the Effects of Surface Geology on Seismic Motion, Laboratoire Central des Ponts et Chaussées, Paris, France, 2006, pp. 1–10, <http://resolver.caltech.edu/CaltechAUTHORS:20120829-132714257>.
- [68] M. Kristekova, J. Kristek, P. Moczo, S.M. Day, Misfit criteria for quantitative comparison of seismograms, *Bull. Seismol. Soc. Am.* 96 (2006) 1836–1850, <https://doi.org/10.1785/0120060012>, <https://pubs.geoscienceworld.org/bssa/article/96/5/1836-1850/146655>.
- [69] M. Kristeková, J. Kristek, P. Moczo, Time-frequency misfit and goodness-of-fit criteria for quantitative comparison of time signals, *Geophys. J. Int.* 178 (2009) 813–825, <https://doi.org/10.1111/j.1365-246X.2009.04177.x>, <https://academic.oup.com/gji/article/178/2/813/625606>.
- [70] M. Käser, V. Hermann, J.d.I. Puente, Quantitative accuracy analysis of the discontinuous Galerkin method for seismic wave propagation, *Geophys. J. Int.* 173 (2008) 990–999, <https://doi.org/10.1111/j.1365-246X.2008.03781.x>.
- [71] N.C. Dutta, H. Odé, Seismic reflections from a gas-water contact, *Geophysics* 48 (1983) 148–162, <https://doi.org/10.1190/1.1441454>, <https://library.seg.org/doi/10.1190/1.1441454>.
- [72] F. Wenzlau, T.M. Müller, Finite-difference modeling of wave propagation and diffusion in poroelastic media, *Geophysics* (2009), <https://doi.org/10.1190/1.3122928>, <https://library.seg.org/doi/abs/10.1190/1.3122928>.
- [73] Leibniz-Rechenzentrum (LRZ), Hardware of SuperMUC-NG, <https://doku.lrz.de/display/PUBLIC/Hardware+of+SuperMUC-NG>, 2021. (Accessed 16 August 2021).
- [74] S. Williams, A. Waterman, D. Patterson, Roofline: an insightful visual performance model for multicore architectures, *Commun. ACM* 52 (2009) 65–76, <https://doi.org/10.1145/1498765.1498785>.
- [75] J. Treibig, G. Hager, G. Wellein, LIKWID: a lightweight performance-oriented tool suite for x86 multicore environments, in: 2010 39th International Conference on Parallel Processing Workshops, IEEE Computer Society, 2010, pp. 207–216, ISSN: 1530-2016, <https://www.computer.org/csdl/proceedings-article/icppw/2010/4157a207/120mNrkJVjb>.
- [76] C. Castro, M. Käser, G. Brietzke, Seismic waves in heterogeneous material: subcell resolution of the Discontinuous Galerkin method, *Geophys. J. Int.* 182 (2010) 250–264, <https://doi.org/10.1111/j.1365-246X.2010.04605.x>.
- [77] P.M. Mai, K.K.S. Thingbaijam, SRCMOD: an online database of finite-fault rupture models, *Seismol. Res. Lett.* 85 (2014) 1348–1357, <https://doi.org/10.1785/0220140077>.
- [78] J. de la Puente, J.-P. Ampuero, M. Käser, Dynamic rupture modeling on unstructured meshes using a Discontinuous Galerkin method, *J. Geophys. Res., Solid Earth* 114 (2009), <https://doi.org/10.1029/2008JB006271>, <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2008JB006271>.
- [79] R.H. Sibson, Interactions between temperature and pore-fluid pressure during earthquake faulting and a mechanism for partial or total stress relief, *Nat. Phys. Sci.* 243 (1973) 66–68, <https://doi.org/10.1038/physci243066a0>, <https://www.nature.com/articles/physci243066a0>.
- [80] H. Noda, E.M. Dunham, J.R. Rice, Earthquake ruptures with thermal weakening and the operation of major faults at low overall stress levels, *J. Geophys. Res., Solid Earth* 114 (2009), <https://doi.org/10.1029/2008JB006143>, <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2008JB006143>.
- [81] R.C. Viesca, D.I. Garagash, Ubiquitous weakening of faults due to thermal pressurization, *Nat. Geosci.* 8 (2015) 875–879, <https://doi.org/10.1038/ngeo2554>, <https://www.nature.com/articles/ngeo2554>.
- [82] A.-A. Gabriel, J.C. Vyas, T. Ulrich, J.-P. Ampuero, M.P. Mai, 3D dynamic rupture modeling with thermal pressurization, in: Poster Presentation at 2020 SCEC Annual Meeting, 2020, p. 158, <https://www.scec.org/meetings/2020/am/poster/158>.
- [83] M. Galis, J.P. Ampuero, P.M. Mai, F. Cappa, Induced seismicity provides insight into why earthquake ruptures stop, *Sci. Adv.* 3 (2017) eaap7528, <https://doi.org/10.1126/sciadv.aap7528>, <https://advances.sciencemag.org/content/3/12/eaap7528>.
- [84] M.J. Grote, T. Mitkova, Explicit local time-stepping methods for Maxwell's equations, *J. Comput. Appl. Math.* 234 (2010) 3283–3302, <https://doi.org/10.1016/j.cam.2010.04.028>, <https://www.sciencedirect.com/science/article/pii/S0377042710002360>.