TOWARDS A POROELASTODYNAMICS FRAMEWORK FOR INDUCED EARTHQUAKES

Saumik Dana[1]

[1]*University of Southern California, Los Angeles, CA 90089*

Earthquakes can be triggered after pore pressure perturbations activate critically stressed seismogenic faults, where the perturbations can originate from natural causes like earth tides, rainfall, snowfall or anthropogenic causes like wastewater disposal, CO_2 injection, oil production, or groundwater extraction. As the faults slip under the action of the induced stress field, seismic waves are spawned from the hypocenter location. The waves propagate through the domain with a velocity that evolves with the evolving pressure and stress fields. The effect of these waves on the surrounding rock and the seismic velocity recorded on the seismograph can be modeled accurately only by incorporating elastodynamics in the deformation model coupled with flow-induced pressure perturbations. Hitherto, most of the literature in the realm has been limited to elastostatics coupled with flow within a prescribed/kinematic or quasi-dynamic fault slip framework. In this work, we provide a framework for coupling of wave propagation with pore pressure perturbations using one-way coupled poroelastodynamics in the presence of faults in which the pore pressure is specified apriori as a spatiotemporal function. We present results from analysis of displacement and velocity fields in the domain and tractions and slip evolution on the fault. The rendition of two-way coupled poroelastodynamics in which the flow problem is also solved is proposed as future work.

KEY WORDS: *Poroelastodynamics, Earthquakes*

1. INTRODUCTION

The earthquake cycle, from slow deformation associated with interseismic behavior to rapid deformation associated with earthquake rupture, spans spatial scales ranging from fractions of a meter associated with the size of contact asperities on faults and individual grains to hundreds of kilometers associated with plate boundaries [1]. Similarly, temporal scales range from fractions of a second associated with slip at a point during earthquake rupture to hundreds of years of strain accumulation between earthquakes. In many cases, earthquakes are triggered after pore pressure perturbations activate critically stressed seismogenic faults [1], not just due to natural causes like earth tides [2], rainfall [3], snowfall [4], typhoons [5], but also due to human activity [6]. As faults slip, the induced stress field spawns seismic waves, which travel through and around the earth (see Fig. 1 and are recorded as ground displacement, velocity and/or acceleration time-series data at seismic stations. Understanding the causality between the events leading to fault slip and the recorded seismic data is important for seismic design and monitoring of underground structures [7] and reinforced concrete buildings [8] as well as climate mitigation projects like carbon sequestration [9] and energy technologies like enhanced geothermal systems [10] or oilfield wastewater disposal [11]. Earthquakes lead to losses to the tune of billions of dollars annually in the United States, with the FEMA and USGS pegging the number at $6.1 billion for the year 2016 (see Fig. 2). The number only includes direct economic losses to buildings, and does not cover damage and losses to critical facilities, transportation and utility lifelines or indirect economic losses.

In quasi-static simulations of fault slip and induced seismicity [12–26], wave propagation is ignored and the solution is a time series of solutions to static problems with potentially time-varying physical properties and boundary conditions. Such quasi-static simulations cannot accurately model dynamic stress changes resulting from an earthquake, whether they are natural [27] or induced [28, 29]. This often means poor accuracy in the predicted event

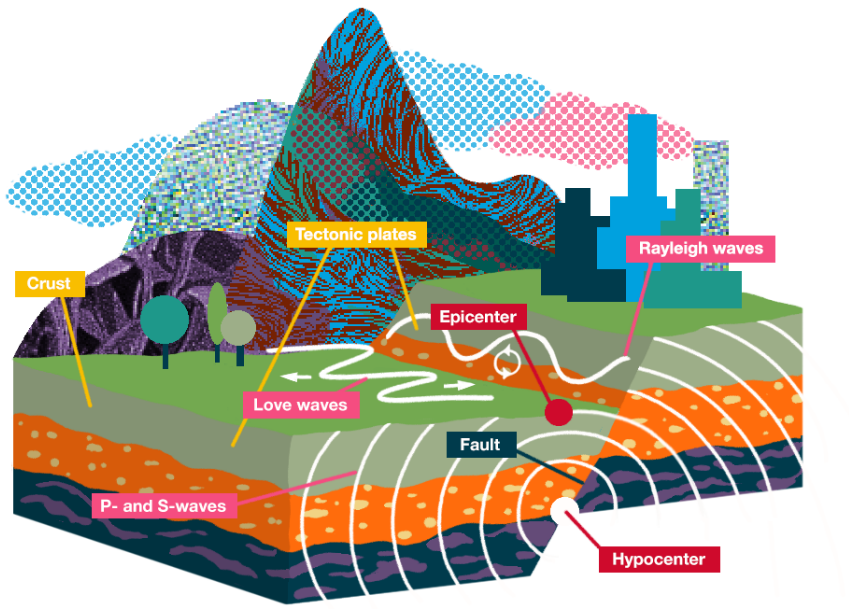


FIG. 1: Source: Caltech Science Exchange

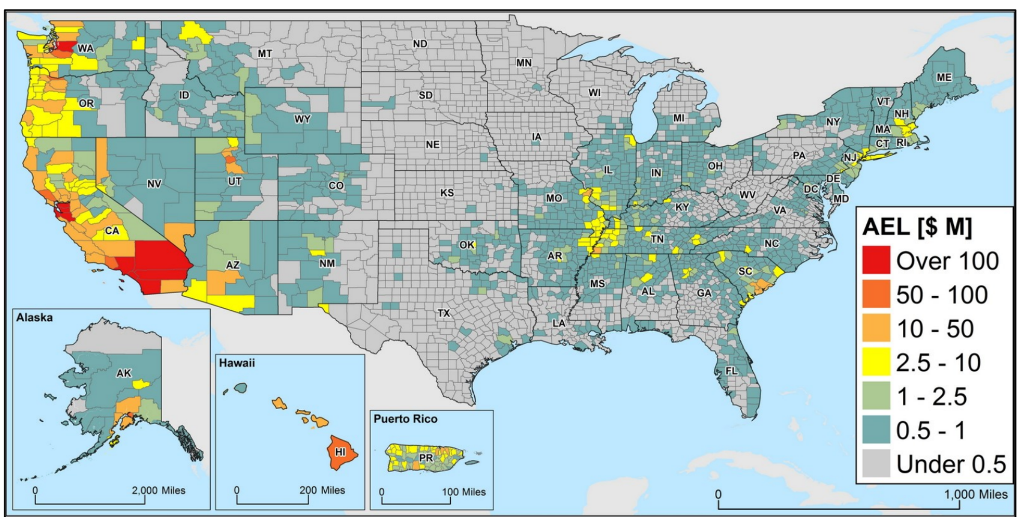


FIG. 2: Annualized Earthquake Losses (AEL: estimated long-term value of earthquake losses to the general building stock) by County. Sources: (1) https://www.usgs.gov/news/usgs-collaborates-fema-national-earthquake-loss-estimate and (2) https://www.fema.gov/hazus. The analysis yielded an estimate of the national AEL of $6.1 billion per year. Loss estimates based on the best science and engineering available (during 2016-2017).

magnitude or the post-earthquake stress state of the fault and the surrounding rock, which in turn means poor prediction of the timing and location of subsequent events. An important fallout of the quasi-static assumption in the context of induced seismicity modeling is that we do not fully understand the impact of pressure field variations on the seismic velocity and stress changes. Some of the prior studies [28, 29] attempted to address this by combining a quasi-static formulation for the fault loading period with an elastodynamic formulation for the coseismic period. However, this required an ad-hoc definition of a transition time between the two formulations and sharp changes in the numerical solution of the problem, which can create numerical oscillations and affect the velocity of stress waves. In this work, we explore coupling of the wave propagation phenomenon with pore pressure perturbations using a one-way coupled poroelastodynamics framework in which the pore pressure is specified apriori as a spatiotemporal function without solving the flow problem. The pressure function is selected to represent instantaneous (as opposed to continuous) injection into a fault zone to induce seismicity. The objective is to study the impact of pressure perturbations on the dynamic response of the system with a frictional rheology model specified for fault slip. The rendition of two-way coupled poroelastodynamics is proposed as future work.

2. MODEL EQUATIONS AND FINITE ELEMENT FORMULATION

The governing PDE for displacement $\mathbf{u} : \Omega \times [0,T] \rightarrow \mathbb{R}^3$ is the linear momentum balance given by

$$\nabla \cdot \boldsymbol{\sigma} + \rho_b \mathbf{g} = \rho_b \ddot{\mathbf{u}} \tag{1}$$

with the constitutive laws relating poroelastic total stress tensor $\boldsymbol{\sigma}$, effective stress tensor $\boldsymbol{\sigma}'$, strain tensor $\boldsymbol{\epsilon}$, volumetric strain $\epsilon = tr(\boldsymbol{\epsilon}$, and pore pressure p along with boundary and initial conditions given by

$$\begin{gathered}
\boldsymbol{\sigma} = \boldsymbol{\sigma}' - bp\mathbf{I} \\
\boldsymbol{\sigma}' = \lambda\epsilon\mathbf{I} + 2G\boldsymbol{\epsilon} \\
\boldsymbol{\epsilon}(\mathbf{u}) = \tfrac{1}{2}(\nabla\mathbf{u} + \nabla^T\mathbf{u}) \\
\mathbf{u} = \overline{\mathbf{u}} \text{ on } \Gamma_D,\ \dot{\mathbf{u}} = \overline{\mathbf{u}} \text{ on } \Gamma_D,\ \boldsymbol{\sigma}^T\mathbf{n} = \overline{\mathbf{t}} \text{ on } \Gamma_N \\
\mathbf{u}(\mathbf{x},0) = \mathbf{u}_0(\mathbf{x}),\ \dot{\mathbf{u}}(\mathbf{x},0) = \dot{\mathbf{u}}_0(\mathbf{x})
\end{gathered} \tag{2}$$

where Γ_D and Γ_N are Dirichlet and Neumann boundaries respectively, ρ_b is the bulk density, λ is a Lame's first constant, G is the shear modulus, $b = 1 - \frac{K_b}{K_s}$ is the Biot-Willis parameter, $K_b \equiv \lambda + \frac{1}{3}2G$ is the drained bulk modulus, K_s is the bulk modulus of the solid grains, $\mathbf{t}$ is the traction boundary condition, and $\mathbf{I}$ is the second order identity tensor.

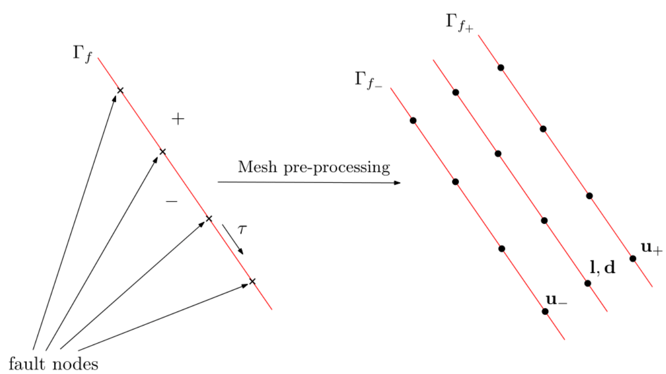


FIG. 3: A fault surface Γ_f (shown as a line in a 2D domain) is processed during mesh processing to create three surfaces: the positive side surface Γ_{f+} containing $\boldsymbol{u}_+$, the negative side surface Γ_{f-} containing $\boldsymbol{u}_-$, and the middle slip surface containing the fault effective traction vector $\boldsymbol{l}$ and the slip vector $\boldsymbol{d}$. The fault surface need not be planar i.e. it can be curved with the normal vector $\boldsymbol{n}$ varying in space on the fault surface.

Using a domain decomposition approach, we consider the fault surface as an interior boundary between two domains. As shown in Fig. 3, we assign a fault normal direction to this interior boundary and "positive" and "negative" labels to the two sides of the fault, such that the fault normal is the vector from the negative side of the fault to the positive side of the fault. Slip on the fault is the displacement of the positive side relative to the negative side:

$$(\boldsymbol{u}_+ - \boldsymbol{u}_-) - \boldsymbol{d} = 0 \text{ on } \Gamma_f, \tag{3}$$

Recognizing that the tractions on the fault surface are analogous to the boundary tractions, we add in the contributions from integrating the Lagrange multipliers $\boldsymbol{l} \equiv \boldsymbol{\sigma}'\boldsymbol{n}$ over the fault surface to get the problem statement as: find $(\boldsymbol{u}, \boldsymbol{l}$ belonging to appropriate functional spaces satisfying the essential boundary conditions ($\boldsymbol{u} = \overline{\boldsymbol{u}}$ on Γ_D) such that

$$\int_\Omega \nabla\boldsymbol{\eta} : (\boldsymbol{\sigma}' - bp\mathbf{I}\; d\Omega - \int_\Omega \boldsymbol{\eta}\cdot\rho_b\boldsymbol{g}\; d\Omega - \int_{\Gamma_N} \boldsymbol{\eta}\cdot\bar{\boldsymbol{t}}\; d\Gamma + \int_\Omega \boldsymbol{\eta}\cdot\rho_b\ddot{\mathbf{u}}\; d\Omega$$

$$+ \int_{\Gamma_{f_+}} \boldsymbol{\eta}\cdot(\boldsymbol{l} - bp_+\boldsymbol{n}\; d\Gamma - \int_{\Gamma_{f_-}} \boldsymbol{\eta}\cdot(\boldsymbol{l} - bp_-\boldsymbol{n}\; d\Gamma = 0, \tag{4}$$

$$\int_{\Gamma_{f_+}} \boldsymbol{\eta}\cdot\boldsymbol{u}_+\; d\Gamma - \int_{\Gamma_{f_-}} \boldsymbol{\eta}\cdot\boldsymbol{u}_-\; d\Gamma - \int_{\Gamma_f} \boldsymbol{\eta}\cdot\boldsymbol{d}\; d\Gamma = 0 \tag{5}$$

for all test functions $\boldsymbol{\eta}$ ($n_{\text{dim}} \times 1$ vector) belonging to the appropriate functional space satisfying $\boldsymbol{\eta} = 0$ on Γ_u where p_+ and p_- are the pore pressures on the 'positive' and the 'negative' side of the fault.

2.1 Fully discrete system of equations

The displacement, the Lagrange multiplier, and the slip fields are approximated as follows:

$$\boldsymbol{u} \approx \boldsymbol{u}_h = \sum_{b=1}^{n_{\text{node}}} \eta_b \boldsymbol{U}_b, \qquad \boldsymbol{l} \approx \boldsymbol{l}_h = \sum_{b=1}^{n_{f,\text{node}}} \eta_b \boldsymbol{L}_b, \qquad \boldsymbol{d} \approx \boldsymbol{d}_h = \sum_{b=1}^{n_{f,\text{node}}} \eta_b \boldsymbol{D}_b$$

where n_{node} is the total number of nodes and $n_{f,\text{node}}$ is the number of Lagrange nodes. We employ explicit time stepping via Newmark's method [30] with a central difference scheme where the acceleration and velocity are given by

$$\ddot{\mathbf{u}}_h^n = \frac{\mathbf{u}_h^{n+1} - 2\mathbf{u}_h^n + \mathbf{u}_h^{n-1}}{(\Delta t)^2} \equiv \frac{d\mathbf{u}_h^n - \mathbf{u}_h^n + \mathbf{u}_h^{n-1}}{(\Delta t)^2} \tag{6}$$

$$\dot{\mathbf{u}}_h^n = \frac{\mathbf{u}_h^{n+1} - \mathbf{u}_h^{n-1}}{2\Delta t} \equiv \frac{d\mathbf{u}_h^n + \mathbf{u}_h^n - \mathbf{u}_h^{n-1}}{2\Delta t} \tag{7}$$

where Δt is the time step and $d\mathbf{u}_h^n = \mathbf{u}_h^{n+1} - \mathbf{u}_h^n$. After substitution of the finite element and temporal approximations into the weak form of the problem (Eqs. (4)–(5)), we obtain the fully-discrete equations in residual form for all nodes a and lagrange nodes $\bar{a}$:

$$\boldsymbol{R}_{u,a} = \int_\Omega \boldsymbol{B}_a^T(\boldsymbol{\sigma}'^n_h - bp_h^n\boldsymbol{I})d\Omega - \int_\Omega \boldsymbol{\eta}_a^T\rho_{b,h}^n\boldsymbol{g}d\Omega - \int_{\Gamma_N}\boldsymbol{\eta}_a^T\bar{\boldsymbol{t}}d\Gamma + \int_\Omega \boldsymbol{\eta}_a^T\rho_{b,h}^n\left(\frac{d\mathbf{u}_h^n - \mathbf{u}_h^n + \mathbf{u}_h^{n-1}}{(\Delta t)^2}\right)d\Omega$$

$$+ \int_{\Gamma_{f_+}} \boldsymbol{\eta}_a^T(\boldsymbol{l}_h^n - bp_{+,h}^n\boldsymbol{n})d\Gamma - \int_{\Gamma_{f_-}} \boldsymbol{\eta}_a^T(\boldsymbol{l}_h^n - bp_{-,h}^n\boldsymbol{n})d\Gamma = 0 \tag{8}$$

$$\boldsymbol{R}_{l,\bar{a}} = \int_{\Gamma_{f_+}} \boldsymbol{\eta}_{\bar{a}}^T\boldsymbol{u}_{h_+}^{n+1}d\Gamma - \int_{\Gamma_{f_-}} \boldsymbol{\eta}_{\bar{a}}^T\boldsymbol{u}_{h_-}^{n+1}\; d\Gamma - \int_{\Gamma_f} \boldsymbol{\eta}_{\bar{a}}^T\boldsymbol{d}_h^{n+1}d\Gamma = 0 \tag{9}$$

where $\boldsymbol{\eta}$ and $\boldsymbol{B}$ are nodal matrices for the shape function and its gradient respectively. Eq. (9) can also be written as

$$\boldsymbol{R}_{l,\bar{a}} = \int_{\Gamma_{f_+}} \boldsymbol{\eta}_{\bar{a}}^T(d\boldsymbol{u}_{h_+}^n + \boldsymbol{u}_{h_+}^n d\Gamma - \int_{\Gamma_{f_-}} \boldsymbol{\eta}_{\bar{a}}^T(d\boldsymbol{u}_{h_-}^n + \boldsymbol{u}_{h_-}^n\; d\Gamma - \int_{\Gamma_f} \boldsymbol{\eta}_{\bar{a}}^T\boldsymbol{d}_h^{n+1}d\Gamma = 0 \tag{10}$$

3. COMPUTATIONAL FRAMEWORK

It is evident from Eqs. (8) and (10) that we are solving for $d\boldsymbol{u}_{h_+}^n$ and $\boldsymbol{l}_h^n$ at time step n. In lieu of that, we find the system Jacobian matrix by isolating the term for the displacement field temporal increment and Lagrange multipliers at time step n. The system of linear equations is:

$$\begin{bmatrix} \boldsymbol{K} & \boldsymbol{C}^T \\ \boldsymbol{C} & 0 \end{bmatrix} \begin{bmatrix} d\boldsymbol{U} \\ \boldsymbol{L} \end{bmatrix} = - \begin{bmatrix} \boldsymbol{R}_{u,a} \\ \boldsymbol{R}_{l,\bar{a}} \end{bmatrix} \tag{11}$$

where the block matrices are obtained via element-by-element assembly of the individual nodal contributions to the Jacobian:

$$\begin{aligned} \boldsymbol{K} &= \int_\Omega \frac{1}{(\Delta t)^2} \boldsymbol{\eta}_a^T \rho_b \boldsymbol{\eta}_b \, d\Omega \qquad ''\text{mass matrix}'' \\ \boldsymbol{C} &= \int_{\Gamma_f} \boldsymbol{\eta}_a^T \boldsymbol{\eta}_{\bar{b}} \, d\Gamma \qquad ''\text{rotation matrix}'' \end{aligned} \tag{12}$$

- ✓ We expect the displacements to be generally on the order of millimeter to centimeter, whereas the fault tractions are expected to be on the order of Mega Pascal (MPa). Thus, if we use dimensioned quantities in SI units, then we would expect the solution to include terms that differ by up to 9 orders of magnitude. This results in a rather ill-conditioned system. This ill-conditioning is avoided by *nondimensionalizing* all of the quantities involved in the problem based upon user-specified scales [31, 32], facilitating the formation of well-conditioned systems of equations for deformation problems across a wide range of spatial and temporal scales.
- ✓ Numerical solution of the wave propagation problem on a discrete mesh often requires *numerical damping* methods to dampen non-physical or numerical waves generated during explicit time-stepping. These are usually high frequency waves/oscillations with wavelengths of the order of the mesh element size near the source of wave generation e.g. a fault or a boundary. The numerical damping method we use is based on defining an adjusted displacement field using an artificial viscosity term η^* as follows [31, 32],

$$\boldsymbol{u}^{adj} = \boldsymbol{u} + \eta^* \Delta t \, \dot{\boldsymbol{u}}$$

- ✓ We employ *absorbing boundary conditions* [31, 32] to prevent seismic waves reflecting off of a boundary by placing simple dashpots on the boundary. Normally incident dilatational and shear waves are perfectly absorbed. Waves incident at other angles are only partially absorbed. This boundary condition is simpler than a perfectly matched layer (PML) [33] boundary condition but does not perform quite as well, especially for surface waves. If the waves arriving at the absorbing boundary are relatively small in amplitude compared to the amplitudes of primary interest, this boundary condition gives reasonable results.

3.1 Time marching algorithm

To visualize the fault contribution to the linear system, we can write Eq. (11) as

$$\left[\begin{array}{ccc:c} \boldsymbol{K}_{rr} & \boldsymbol{K}_{r+} & \boldsymbol{K}_{r-} & 0 \\ \boldsymbol{K}_{+r} & \boldsymbol{K}_{++} & 0 & \boldsymbol{C}_+^T \\ \boldsymbol{K}_{-r} & 0 & \boldsymbol{K}_{--} & -\boldsymbol{C}_-^T \\ \hdashline 0 & \boldsymbol{C}_+ & -\boldsymbol{C}_- & 0 \end{array}\right] \left[\begin{array}{c} d\boldsymbol{U}_r \\ d\boldsymbol{U}_+ \\ d\boldsymbol{U}_- \\ \hline \boldsymbol{L} \end{array}\right] = - \left[\begin{array}{c} \boldsymbol{R}_{u,r} \\ \boldsymbol{R}_{u,+} \\ \boldsymbol{R}_{u,-} \\ \hline \boldsymbol{R}_l \end{array}\right] \tag{13}$$

where the top row corresponds to displacement nodes excluding the fault positive and negative side nodes. The displacement update in Eq. (8) is evidently explicit in nature, and the CFL condition [34] controls the maximum allowable time step. As shown in Algorithm 1, the time marching requires inversion of both the mass matrix and rotation matrix. Given that the maximum allowable time step is limited by the CFL condition, the marching would require substantially more time steps compared to the scenario without any time step size constraint. The desire to reduce the floating point operations for inversion of matrices leads us to employ the lumping strategy [31, 32] to diagonalize the matrices. The lumping is performed on each submatrix of the matrix given in Eq. (13). At the virgin state, the displacement fields are initialized based on the initial conditions for the problem. At each time step, the residuals (Eqs. (8) and (9)) are evaluated by assuming the temporal increment in displacement is zero. These residuals are then used to compute increments in displacement and lagrange multipliers.

Algorithm 1: Time-marching in one-way coupled poroelastodynamics

$n \leftarrow 0$; $t \leftarrow 0$;

$\boldsymbol{u}_h^{n=0} \leftarrow \boldsymbol{u}_{0_h}$; $\boldsymbol{u}_h^{n=-1} \leftarrow 0$; // Initial condition

$\boldsymbol{K} \leftarrow \boldsymbol{K}_d$; $\boldsymbol{C} \leftarrow \boldsymbol{C}_d$; // Diagonalize mass and rotation matrix

while $t < T$ **do**

/* time marching till final time */

$\boldsymbol{R}^*_{u,a} \leftarrow \boldsymbol{R}_{u,a}|_{d\boldsymbol{u}_h^n=0}$;$\boldsymbol{R}^*_{l,\bar{a}} \leftarrow \boldsymbol{R}_{l,\bar{a}}|_{d\boldsymbol{u}_h^n=0}$; // Get residuals corresponding to zero increment in displacement

$d\boldsymbol{U}^* \leftarrow \boldsymbol{K}^{-1}\boldsymbol{R}^*_{u,a}$; // Initial guess

$d\boldsymbol{L} \leftarrow \left(\boldsymbol{C}(\boldsymbol{K}^{-1}_{++} + \boldsymbol{K}^{-1}_{--})\boldsymbol{C}^T\right)^{-1}\left(-\boldsymbol{R}^*_{l,\bar{a}} + \boldsymbol{C}(d\boldsymbol{U}^*_+ + d\boldsymbol{U}^*_-)\right)$;

$\boldsymbol{L} \leftarrow \boldsymbol{L} + d\boldsymbol{L}$; // Update lagrange multipliers

$d\boldsymbol{U} \leftarrow d\boldsymbol{U}^* - \boldsymbol{K}^{-1}\boldsymbol{C}^T d\boldsymbol{L}$; // Corrected increment

$\boldsymbol{U} \leftarrow \boldsymbol{U} + d\boldsymbol{U}$; // Update displacements

$\boldsymbol{d}_h^{n+1} \leftarrow \boldsymbol{u}_{h_+}^{n+1} - \boldsymbol{u}_{h_-}^{n+1}$; // Update fault slip

for *Loop over lagrange nodes* **do**

$\tau \leftarrow |\boldsymbol{l}_h^{n+1} - (\boldsymbol{l}_h^{n+1} \cdot \boldsymbol{n})\boldsymbol{n}|$; // Obtain shear stress on fault

$\tau_f \leftarrow \tau_f|_{\boldsymbol{d}_h^{n+1}}$; // Compute fault friction based on the slip value at previous time step

while $\tau > \tau_f$ **do**

/* Satisfy fault constitutive law. If the Lagrange multipliers do not exceed the fault tractions allowed by the fault constitutive model, the increment in fault slip remains zero, and no adjustments to the solution are necessary */

$d\boldsymbol{L} \leftarrow \frac{\tau - \tau_f}{\tau}\boldsymbol{L}$;

$\boldsymbol{L} \leftarrow \boldsymbol{L} + d\boldsymbol{L}$; // Update lagrange multipliers

$\boldsymbol{U}_+ \leftarrow \boldsymbol{U}_+ - \boldsymbol{K}^{-1}_{++}\boldsymbol{C}^T d\boldsymbol{L}$;

$\boldsymbol{U}_- \leftarrow \boldsymbol{U}_- + \boldsymbol{K}^{-1}_{--}\boldsymbol{C}^T d\boldsymbol{L}$;

$\boldsymbol{d}_h^{n+1} \leftarrow \boldsymbol{u}_{h_+}^{n+1} - \boldsymbol{u}_{h_-}^{n+1}$; // Update fault slip

$\tau_f \leftarrow \tau_f|_{\boldsymbol{d}_h^{n+1}}$; // Update fault friction

end

end

$n \leftarrow n + 1$; // time step number

$t \leftarrow t + \Delta t$; // current time

end

3.1.1 Fault constitutive model in the for-loop

The magnitude of the effective normal traction on the fault is

$$\sigma'_n = \boldsymbol{l} \cdot \boldsymbol{n} \tag{14}$$

where $\boldsymbol{n}$ is the outward unit normal vector on the fault surface. The shear traction vector is, by definition, tangent to the fault surface and its magnitude is

$$\tau = |\boldsymbol{l} - \sigma'_n \boldsymbol{n}| \equiv |\boldsymbol{l} - (\boldsymbol{l} \cdot \boldsymbol{n})\,\boldsymbol{n}| \tag{15}$$

Shear tractions on the fault are limited by fault friction or fault strength. A fault constitutive model is used to compute the frictional stress τ_f on the fault as

$$\tau_f = \begin{cases} \tau_c - \mu_f \boldsymbol{l} \cdot \mathbf{n}, & \boldsymbol{l} \cdot \mathbf{n} < 0 \quad \text{i.e., under compression} \\ \tau_c, & \boldsymbol{l} \cdot \mathbf{n} \geq 0 \quad \text{i.e., under tension} \end{cases} \tag{16}$$

where τ_c is the cohesive strength of the fault, and μ_f is the coefficient of friction, which is modeled as a function of slip evolution on the fault. We use the slip-weakening model, where μ_f is a function of the slip magnitude $|\boldsymbol{d}|$, and drops linearly from static friction μ_s to dynamic friction μ_d over a critical slip distance d_c,

$$\mu_f = \begin{cases} \mu_s - (\mu_s - \mu_d)\frac{|\boldsymbol{d}|}{d_c}, & |\boldsymbol{d}| \leq d_c, \\ \mu_d, & |\boldsymbol{d}| > d_c \end{cases} \tag{17}$$

We use the Mohr-Coulomb theory [35] to define the stability criterion for the fault.

- ✓ When the shear traction on the fault is below the friction stress, $\tau \leq \tau_f$, the fault does not slip
- ✓ When the shear traction is larger than the friction stress, $\tau > \tau_f$, the contact problem is solved to determine the Lagrange multipliers and slip on the fault, such that the Lagrange multipliers are compatible with the frictional stress. Instead of assessing fault stability on both sides of the fault separately, we define a fault pressure p_f that is a function of the pressures on the two sides as $p_f = \frac{p_- + p_+}{2}$. By univocally defining the pressure at the fault, we also univocally define the effective traction at the fault (the Lagrange multiplier $\boldsymbol{l}$)

4. COMPUTER IMPLEMENTATION

The simulator is a mixed programming language hybrid MPI and OpenMP framework in which the open source geodynamics code [31, 32] written in Python and C++ is MPI-based and the in-house multiphase/multicomponent flow code written in C++ is OpenMP-based [36–38]. There are three layers in which the code framework operates out of separate directories

- ✓ The high-level Python code in the geodynamics module makes use of Pyre [39] to link the sub-modules together at runtime
- ✓ The low-level C++ code performs operations on matrices and vectors in parallel using the Standard Template Library [40]
- ✓ The SWIG [41] code with all the .i files, and SWIG-generated python code and MODULENAME_wrap.cxx which wraps the Python code around the C++ code to create python modules. Changes made to the C++ code entail a makefile related compilation of the code base. If there are changes to the C++ header files that need to be exposed to the input file, then those changes have to be copied to the corresponding module's .i file. This means SWIG has to be run again on that module to update MODULENAME_wrap.cxx and Python files of the module before building the executable using its Makefile.

The numpy library [42] allows direct manipulation in Python of multi-dimensional arrays, using memory allocated in C++. Some of the code snippets are provided in APPENDIX A.

5. REPRESENTATIVE NUMERICAL SIMULATIONS

As shown in Fig. 4, we focus on the dynamics of a shear wave propagating down an 8.0 km long bar with a 400 m wide cross-section. Motion is limited to shear deformation by fixing the longitudinal degree of freedom, u_x. The domain is discretized with a Cartesian structured mesh with uniform discretization size of 200 m. We use static coefficient of friction $\mu_s = 0.6$, dynamic coefficient of friction $\mu_d = 0.5$, critical slip distance $d_c = 0.2\,m$, and no cohesion $\tau_c = 0\,MPa$. For nondimensionalizing the equations, we use a shear wave speed of $1\,km/s$, a mass density of $3000\,kg/m^3$ and a wave period of $1.0\,s$. We simulate $12.0\,s$ of motion with a time step of $1/20\,s$ and the artificial viscosity is set to $\eta^* = 0.1$. The simulations were performed using the Amazon Web Services instance t2.xlarge.

We study the effect of pressure perturbation on wave propagation in the bar. We generate the pressure perturbation as the solution of a simple 1-D diffusion equation and use that to define the function p_h in Eq. (8). The spatiotemporal pore pressure function is

$$p(x,t) = \begin{cases} 0, & t = 0 \\ \frac{A}{\sqrt{4\pi D t'}} e^{-\frac{(x' - x_0')^2}{4Dt'}}, & t > 0 \end{cases} \tag{18}$$

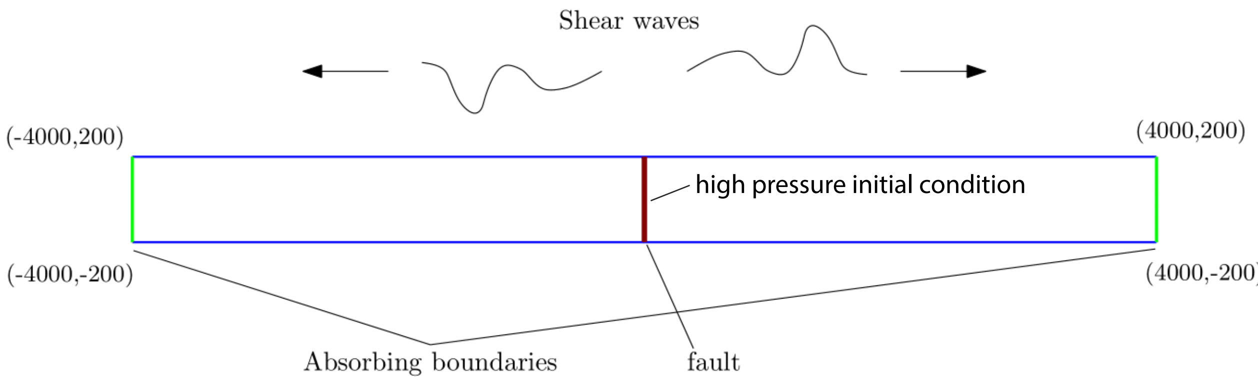


FIG. 4: Domain for shear wave propagation in a $8\,km \times 400\,m$ (length $\times$ height) bar. We generate a shear wave via tractions imposed on a fault located in the middle of the bar. To study the effect of pressure perturbation on wave propagation, we initialize the pressure field with an instantaneous high pressure within the fault. Photo courtesy of [31]

which is the solution to the 1-D diffusion equation

$$\frac{\partial p}{\partial t'} = D\frac{\partial^2 p}{\partial x'^2} \tag{19}$$

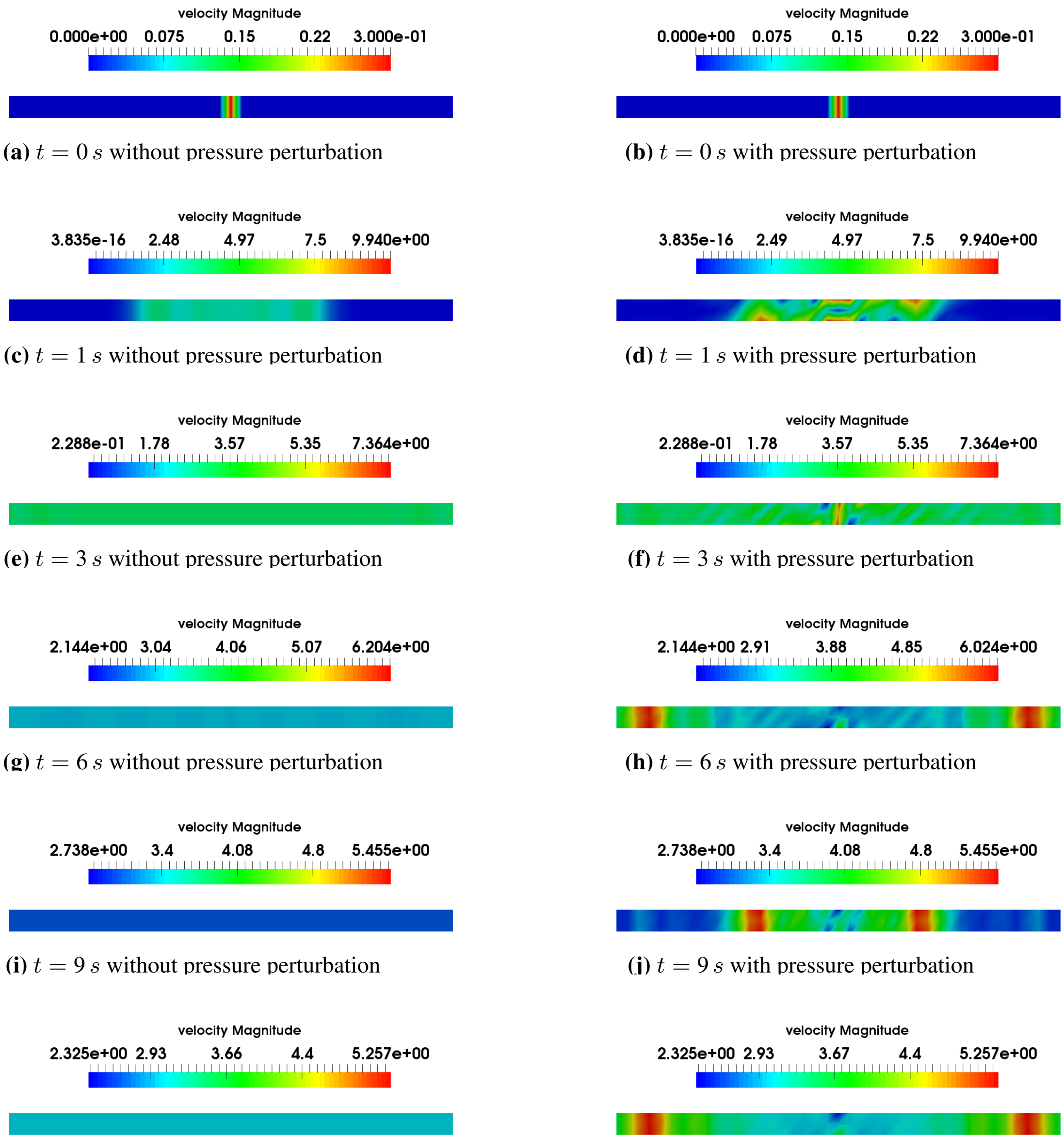
with concentrated overpressure at $x' = x'_0$. The dimensionless distance is defined as $x' = x/8000$ with x in meter. D is the hydraulic diffusivity [43, 44], which is a function of porosity ϕ, permeability k, fluid viscosity μ, and poroelastic moduli of the porous medium, e.g. in consolidation problems $D = (kM/\mu)(K_b + (G/3))/(K_u + (G/3))$, where M is the Biot modulus and $K_u = K_b + b^2M$ is the undrained bulk modulus. The dimensionless time is $t' = t/12$ with t in second. The pressure solution in Eq. (18) satisfies the diffusion PDE Eq. (19), which we can see from the derivatives of Eq. (18):

$$\begin{aligned} \frac{\partial^2 p}{\partial x'^2} &= -\frac{2p(x,t)}{4Dt'} + \frac{4p(x,t)(x'-x'_0)^2}{(4Dt')^2} \\ \frac{\partial p}{\partial t'} &= -\frac{p(x,t)}{2t'} + \frac{1}{t'}\frac{(x'-x'_0)^2}{4Dt'}p(x,t) \end{aligned} \tag{20}$$

In our simulations, $x_0 = 0$ and $A = 50$ MPa. A real-world scenario giving rise to such a pressure function is that of fluid injection into a fault zone up to the time of slip nucleation at which point injection is stopped to let pressure diffuse away from the fault. This pressure behavior is also similar to the pressure solution used in a prior study on injection-induced microseismicity generation [45]. Here, we neglected the volumetric strain rate (equivalently, volumetric stress rate or porosity evolution) term in the pressure equation Eq. (19) [38] because we are assuming that the effect of wave propagation on pressure is negligible, which allows us to use this one-way coupled formulation with coupling from fluid to solid but not from solid to fluid. This assumption may or may not be satisfied depending on the values of poroelastic properties which determine the time scales of wave propagation and pressure diffusion phenomena. Since the focus of this paper is to develop a numerical framework for wave propagation under poro-elastodynamics and study its sensitivity to pressure perturbations, we make this one-way coupled assumption for easier interpretation of the results. In a future extension of this work, we will consider the two-way coupling between flow and geomechanics, which will effectively render diffusivity D as time-dependent because porosity, compressibility and permeability can evolve with time.

We perform four simulations and use the result to conduct three comparisons:

- ✓ Comparison of solutions with shear wave generated via fault tractions without and with the pressure perturbation with diffusivity $D = 1\,m^2/s$. We generate a shear wave using tractions imposed to the fault, which is located at the center of the bar. The tractions are 75 MPa of right-lateral shear and 120 MPa of normal compression. See Fig. 5.
- ✓ Comparison of solutions with the pressure perturbation with diffusivity $D = 1\,m^2/s$, with and without the shear wave generated via fault tractions. See Fig. 6.



(a) $t = 0\,s$ without pressure perturbation

(b) $t = 0\,s$ with pressure perturbation

(c) $t = 1\,s$ without pressure perturbation

(d) $t = 1\,s$ with pressure perturbation

(e) $t = 3\,s$ without pressure perturbation

(f) $t = 3\,s$ with pressure perturbation

(g) $t = 6\,s$ without pressure perturbation

(h) $t = 6\,s$ with pressure perturbation

(i) $t = 9\,s$ without pressure perturbation

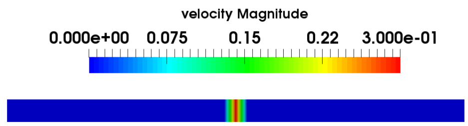
(j) $t = 9\,s$ with pressure perturbation

(k) $t = 12\,s$ without pressure perturbation

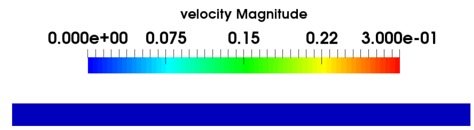
(l) $t = 12\,s$ with pressure perturbation

FIG. 5: Comparison of velocity solutions without (left column) and with (right column) pressure perturbation generated using diffusivity $D = 1\,m^2/s$. A shear wave is generated via fault tractions imposed on the fault: 75 MPa right-lateral shear and 120 MPa compression. The fault is located at the center of the bar. Without pressure perturbation, the velocity field is homogeneous along the bar.

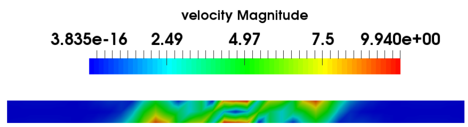
✓ Comparison of solutions with no shear wave generated via fault tractions with different spatiotemporal pressure perturbations due to different diffusivities. In one case, the diffusivity is $D = 1\,m^2/s$ and in the other case, diffusivity is $D = 10\,m^2/s$. See Fig. 7.



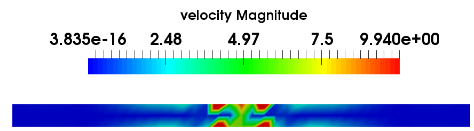
(a) $t = 0\,s$ with shear wave



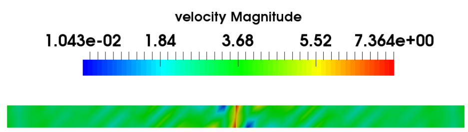
(b) $t = 0\,s$ without shear wave



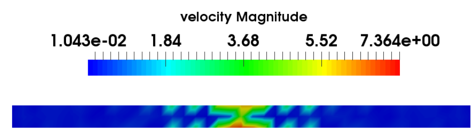
(c) $t = 1\,s$ with shear wave



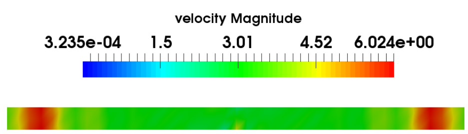
(d) $t = 1\,s$ without shear wave



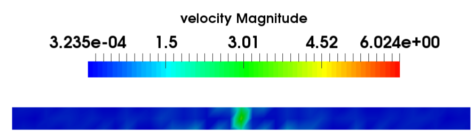
(e) $t = 3\,s$ with shear wave



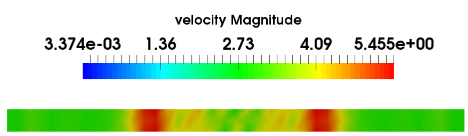
(f) $t = 3\,s$ without shear wave



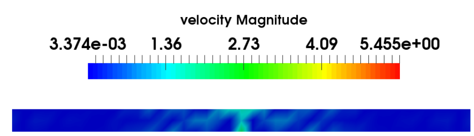
(g) $t = 6\,s$ with shear wave



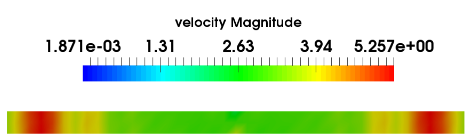
(h) $t = 6\,s$ without shear wave



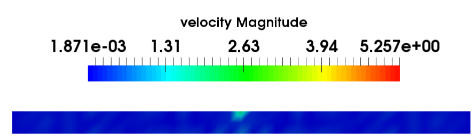
(i) $t = 9\,s$ with shear wave



(j) $t = 9\,s$ without shear wave



(k) $t = 12\,s$ with shear wave

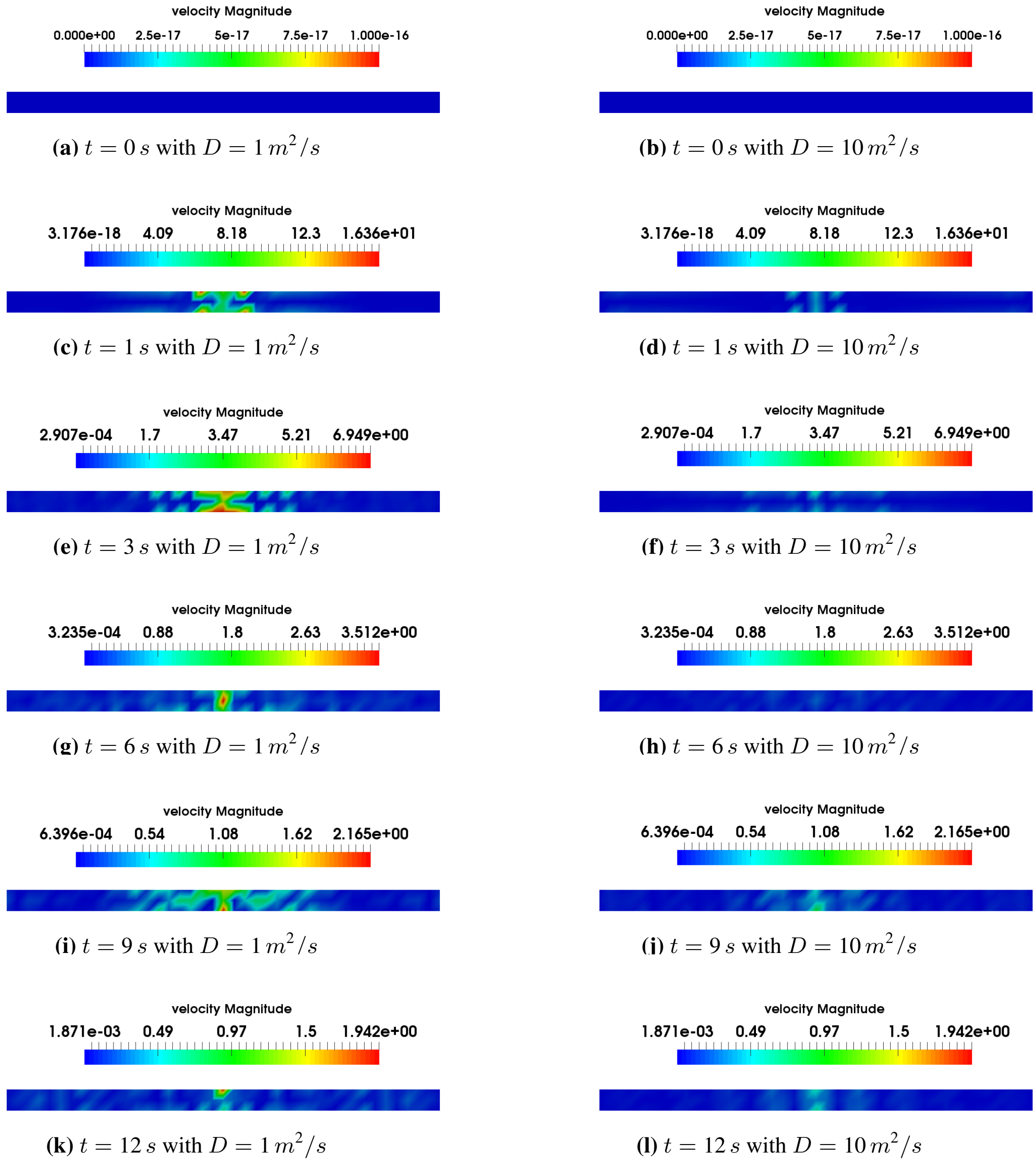


(l) $t = 12\,s$ without shear wave

FIG. 6: Comparison of solutions with the pressure perturbation, generated using diffusivity $D = 1\,m^2/s$, with and without the shear wave generated via fault tractions. In absence of imposed fault tractions, the effect of pressure perturbation decays with time as shown by a decaying velocity field in the right column figures.

6. DISCUSSION

We developed and implemented a computational framework to study the effect of fluid flow on wave propagation in the subsurface in the presence of faults. We focused on how poro-elastodynamic fault rupture, driven by tractions imposed on the fault, responds to changes in the pore pressure field which we prescribe as a spatiotemporal function representing instantaneous injection into the fault. We analyzed displacement and velocity fields mostly via visual



(a) $t = 0\,s$ with $D = 1\,m^2/s$ **(b)** $t = 0\,s$ with $D = 10\,m^2/s$

(c) $t = 1\,s$ with $D = 1\,m^2/s$ **(d)** $t = 1\,s$ with $D = 10\,m^2/s$

(e) $t = 3\,s$ with $D = 1\,m^2/s$ **(f)** $t = 3\,s$ with $D = 10\,m^2/s$

(g) $t = 6\,s$ with $D = 1\,m^2/s$ **(h)** $t = 6\,s$ with $D = 10\,m^2/s$

(i) $t = 9\,s$ with $D = 1\,m^2/s$ **(j)** $t = 9\,s$ with $D = 10\,m^2/s$

(k) $t = 12\,s$ with $D = 1\,m^2/s$ **(l)** $t = 12\,s$ with $D = 10\,m^2/s$

FIG. 7: Comparison of solutions with no shear wave generated via fault tractions with different spatiotemporal pressure perturbations arising from different diffusivity values. In one case, the diffusivity is $D = 1\,m^2/s$ and in the other case, diffusivity is $D = 10\,m^2/s$. As expected, higher diffusivity leads to lower velocities and a faster decay of the velocity field.

comparison. A more in-depth analysis of the pressure effect on the stress field can inform predictive models of induced seismicity. From the model development point of view, the next course of action would be to couple geodynamics with flow simulation to study the effect of elastodynamics on the flow process, in particular on the pore pressure evolution. Here, the challenge would be to develop numerical schemes that are stable and accurate for systems showing strong flow-deformation coupling and nonlinear constitutive behaviour, which are often present at induced seismicity sites with oil production, CO_2 sequestration, groundwater extraction, or geothermal extraction. Including the physics of elastodynamics in flow-induced fault activation can help in interpreting real-time seismic/microseismic data collected at these sites.

APPENDIX A. CODE SNIPPETS

```
# TimeDependent class
class TimeDependent(Problem):
  ...
  def run(self, app): #Solve time dependent problem
    ...
    stepCount = 0
    ...
    # Normal time loop
    t = self.formulation.getStartTime()
    while t < self.formulation.getTotalTime():
      ...
      self.formulation.step(t, dt, self.normalizer, stepCount) # time march
      self.formulation.poststep(t, dt)
      t += dt
      stepCount += 1
    return
```

Listing 1: Python driver code for time stepping

```
# Explicit class
class Explicit(Formulation, ModuleExplicit):
  ...
  def step(self, t, dt, normalizer, stepCount):
    ...
    if self.flowsim == None: # one-way coupled
      self.stepWithoutFlow(t, dt, normalizer) # time march
      return
    ...
    # Driver code for two-way coupled--work in progress
    ...
    return

  def stepWithoutFlow(self, t, dt, normalizer):
    ...
    self.prestep(t, dt) # form jacobian
    ...
    self._reformResidual(t, dt) # form residual, lower level C++ code
    ...
    self.solver.solve(dispIncr, self.jacobian, residual) # solve
    disp += dispIncr
    Formulation.poststep(self, t, dt)
    return
```

Listing 2: Python driver code for explicit time stepping

```
void pylith::problems::Formulation::reformJacobianLumped(void)
{ // Set jacobian to zero.
  _jacobianLumped->zero();
  // Add in contributions other than fault
  int numIntegrators = _meshIntegrators.size();
  for (int i=0; i < numIntegrators; ++i) {
    if (0 == _flowsim) _meshIntegrators[i]->integrateJacobian(_jacobianLumped
   , _t, _fields);
    else _meshIntegrators[i]->integrateJacobian(_jacobianLumped, _t, _fields,
    _flowsim);
  }
  // Add in contributions from fault
  numIntegrators = _submeshIntegrators.size();
  for (int i=0; i < numIntegrators; ++i) {
    if (0 == _flowsim) _submeshIntegrators[i]->integrateJacobian(
   _jacobianLumped, _t, _fields);
    else _submeshIntegrators[i]->integrateJacobian(_jacobianLumped, _t,
   _fields, _flowsim);
  }
  // Assemble jacobian.
  _jacobianLumped->complete();
}
```

Listing 3: Lower level C++ code snippet for jacobian construction

```
void pylith::problems::Formulation::reformResidual(const PetscVec*
    tmpResidualVec, const PetscVec* tmpSolutionVec)
{ ...
  // Update rate fields (must be consistent with current solution).
  calcRateFields();
  // Set residual to zero.
  residual.zero();
  // Add in contributions other than fault
  int numIntegrators = _meshIntegrators.size();
  assert(numIntegrators > 0); // must have at least 1 bulk integrator
  for (int i=0; i < numIntegrators; ++i) {
    _meshIntegrators[i]->timeStep(_dt);
    if (0 == _flowsim)  _meshIntegrators[i]->integrateResidual(residual, _t,
   _fields);
    else _meshIntegrators[i]->integrateResidual(residual, _t, _fields,
   _flowsim);
  } // for
  // Add in fault contributions
  numIntegrators = _submeshIntegrators.size();
  for (int i=0; i < numIntegrators; ++i) {
    _submeshIntegrators[i]->timeStep(_dt);
    if (0 == _flowsim)  _submeshIntegrators[i]->integrateResidual(residual,
   _t, _fields);
    else  _submeshIntegrators[i]->integrateResidual(residual, _t, _fields,
   _flowsim);
  } // for
```

```
  // Assemble residual.
  residual.complete();
}
```

Listing 4: Lower level C++ code snippet for residual construction

References

[1] Hiroo Kanamori and Emily E Brodsky. The physics of earthquakes. *Reports on Progress in Physics*, 67(8):1429, 2004.

[2] Christopher H Scholz, Yen Joe Tan, and Fabien Albino. The mechanism of tidal triggering of earthquakes at mid-ocean ridges. *Nature communications*, 10(1):1–7, 2019.

[3] Sebastian Hainzl, Toni Kraft, Joachim Wassermann, Heiner Igel, and E Schmedes. Evidence for rainfall-triggered earthquake activity. *Geophysical Research Letters*, 33(19), 2006.

[4] EK Montgomery-Brown, David R Shelly, and Paul A Hsieh. Snowmelt-triggered earthquake swarms at the margin of long valley caldera, california. *Geophysical Research Letters*, 46(7):3698–3705, 2019.

[5] ChiChing Liu, Alan T Linde, and I Selwyn Sacks. Slow earthquakes triggered by typhoons. *Nature*, 459(7248):833–836, 2009.

[6] Gillian R Foulger, Miles P Wilson, Jon G Gluyas, Bruce R Julian, and Richard J Davies. Global review of human-induced earthquakes. *Earth-Science Reviews*, 178:438–514, 2018.

[7] Youssef MA Hashash, Jeffrey J Hook, Birger Schmidt, I John, and Chiang Yao. Seismic design and analysis of underground structures. *Tunnelling and underground space technology*, 16(4):247–293, 2001.

[8] Jack Moehle. *Seismic design of reinforced concrete buildings*. McGraw-Hill Education, 2015.

[9] Mark D Zoback and Steven M Gorelick. Earthquake triggering and large-scale geologic storage of carbon dioxide. *Proceedings of the National Academy of Sciences*, 109(26):10164–10168, 2012.

[10] William L Ellsworth, Domenico Giardini, John Townend, Shemin Ge, and Toshihiko Shimamoto. Triggering of the pohang, korea, earthquake (m w 5.5) by enhanced geothermal system stimulation. *Seismological Research Letters*, 90(5):1844–1858, 2019.

[11] Katie M Keranen and Matthew Weingarten. Induced seismicity. *Annual Review of Earth and Planetary Sciences*, 46:149–174, 2018.

[12] Saumik Dana and Mary F Wheeler. Augmented lagrangian for treatment of hanging nodes in hexahedral meshes. *arXiv preprint arXiv:1809.04031*, 2018.

[13] Saumik Dana and Karthik Reddy Lyathakula. Uncertainty quantification in friction model for earthquakes using bayesian inference. *arXiv preprint arXiv:2104.11156*, 2021.

[14] S. Dana and M. F. Wheeler. Convergence analysis of fixed stress split iterative scheme for anisotropic poroelasticity with tensor biot parameter. *Computational Geosciences*, 22(5):1219–1230, 2018.

[15] S. Dana and M. F. Wheeler. Convergence analysis of two-grid fixed stress split iterative scheme for coupled flow and deformation in heterogeneous poroelastic media. *Computer Methods in Applied Mechanics and Engineering*, 341:788–806, 2018.

[16] Saumik Dana and Mary F Wheeler. Design of convergence criterion for fixed stress split iterative scheme for small strain anisotropic poroelastoplasticity coupled with single phase flow. *arXiv preprint arXiv:1912.06476*, 2019.

[17] Saumik Dana and Mary F Wheeler. An efficient algorithm for numerical homogenization of fluid filled porous solids: part-i. *arXiv preprint arXiv:2002.03770*, 2020.

[18] Saumik Dana and Birendra Jha. A fault slip model to study earthquakes due to pore pressure perturbations. *arXiv preprint arXiv:2104.06257*, 2021.

[19] Saumik Dana and Mary F Wheeler. A machine learning accelerated fe^2 homogenization algorithm for elastic solids. *arXiv preprint arXiv:2003.11372*, 2020.

[20] Saumik Dana, Benjamin Ganis, and Mary F. Wheeler. A multiscale fixed stress split iterative scheme for coupled flow and poromechanics in deep subsurface reservoirs. *Journal of Computational Physics*, 352:1–22, 2018.

[21] Saumik Dana, Shriram Srinivasan, Satish Karra, Nataliia Makedonska, Jeffrey D Hyman, Daniel O'Malley, Hari Viswanathan, and Gowri Srinivasan. Towards real-time forecasting of natural gas production by harnessing graph theory for stochastic discrete fracture networks. *Journal of Petroleum Science and Engineering*, 195:107791, 2020.

[22] Saumik Dana, Joel Ita, and Mary F Wheeler. The correspondence between voigt and reuss bounds and the decoupling constraint in a two-grid staggered algorithm for consolidation in heterogeneous porous media. *Multiscale Modeling & Simulation*, 18(1):221–239, 2020.

[23] Saumik Dana. A simple framework for arriving at bounds on effective moduli in heterogeneous anisotropic poroelastic solids. *arXiv preprint arXiv:1912.10835*, 2019.

[24] Saumik Dana. System of equations and staggered solution algorithm for immiscible two-phase flow coupled with linear poromechanics. *arXiv preprint arXiv:1912.04703*, 2019.

[25] S. Dana. *Addressing challenges in modeling of coupled flow and poromechanics in deep subsurface reservoirs*. PhD thesis, The University of Texas at Austin, 2018.

[26] Saumik Dana, Xiaoxi Zhao, and Birendra Jha. Two-grid method on unstructured tetrahedra: Applying computational geometry to staggered solution of coupled flow and mechanics problems. *arXiv preprint arXiv:2102.04455*, 2021.

[27] Marion Y. Thomas, Nadia Lapusta, Hiroyuki Noda, and Jean-Philippe Avouac. Quasi-dynamic versus fully dynamic simulations of earthquakes and aseismic slip with and without enhanced coseismic weakening. *Journal of Geophysical Research: Solid Earth*, 119(3):1986–2004, 2014.

[28] C. Meng. Benchmarking defmod, an open source fem code for modeling episodic fault rupture. *Computers & Geosciences*, 100:10–26, 2017.

[29] Lei Jin and Mark D Zoback. Fully dynamic spontaneous rupture due to quasi-static pore pressure and poroelastic effects: An implicit nonlinear computational model of fluid-induced seismic events. *Journal of Geophysical Research: Solid Earth*, 123(11):9430–9468, 2018.

[30] Nathan M Newmark. A method of computation for structural dynamics. *Journal of the engineering mechanics division*, 85(3):67–94, 1959.

[31] B Aagaard, S Kientz, M Knepley, L Strand, and C Williams. Pylith user manual: version 2.1. 0, 2013.

[32] Brad T Aagaard, Matthew G Knepley, and Charles A Williams. A domain decomposition approach to implementing fault slip in finite-element models of quasi-static and dynamic crustal deformation. *Journal of Geophysical Research: Solid Earth*, 118(6):3059–3079, 2013.

[33] Ushnish Basu and Anil K. Chopra. Perfectly matched layers for time-harmonic elastodynamics of unbounded domains: theory and finite-element implementation. *Computer Methods in Applied Mechanics and Engineering*, 192(11-12):1337–1375, 2003.

[34] Richard Courant, Kurt Friedrichs, and Hans Lewy. On the partial difference equations of mathematical physics. *IBM journal of Research and Development*, 11(2):215–234, 1967.

[35] J. C. Jaeger and N. G. W. Cook. *Fundamentals of Rock Mechanics*. Chapman and Hall, London, 1979.

[36] X. Zhao and B. Jha. A new coupled multiphase flow–finite strain deformation–fault slip framework for induced seismicity. *J. Comput. Phys.*, 433:110178, 2021.

[37] S. Meguerdijian and B. Jha. Quantification of fault leakage dynamics based on leakage magnitude and dip angle. *Int. J. Numer. Anal. Methods Geomech.*, 2021.

[38] Birendra Jha and Ruben Juanes. Coupled multiphase flow and poromechanics: A computational model of pore pressure effects on fault slip and earthquake triggering. *Water Resources Research*, 50(5):3776–3808, 2014.

[39] K Jarrod Millman and Michael Aivazis. Python for scientists and engineers. *Computing in Science & Engineering*, 13(2):9–12, 2011.

[40] Nicolai M Josuttis. *The C++ standard library: a tutorial and reference*. Addison-Wesley, 2012.

[41] Teresa L Cottom. Using swig to bind c++ to python. *Computing in Science & Engineering*, 5(2):88–97, 2003.

[42] Stefan Van Der Walt, S Chris Colbert, and Gael Varoquaux. The numpy array: a structure for efficient numerical computation. *Computing in science & engineering*, 13(2):22–30, 2011.

[43] H. F. Wang. *Theory of Linear Poroelasticity with Applications to Geomechanics and Hydrogeology*. Princeton University Press, 2000.

[44] Grazia De Landro, Ortensia Amoroso, Guido Russo, and Aldo Zollo. 4d travel-time tomography as a tool for tracking fluid-driven medium changes in offshore oil–gas exploitation areas. *Energies*, 13(22):5878, 2020.

[45] José M. Carcione, Federico Da Col, Gilda Currenti, and Barbara Cantucci. Modeling techniques to study co2-injection induced micro-seismicity. *International Journal of Greenhouse Gas Control*, 42:246–257, 2015.