Noname manuscript No.
(will be inserted by the editor)

A Non-stiff Summation-By-Parts Finite Difference Method for Earthquake Cycle Simulations with Full Dynamic Effects

Tobias W. Harvey · Brittany A. Erickson · Jeremy E. Kozdon

Received: date / Accepted: date

Abstract We present a high-order finite difference method for earthquake cycle simulations within complex geometries that incorporates full dynamic effects and is provably stable. The method is developed for the two-dimensional anti-plane strain problem where a rate-and-state frictional fault is embedded in a heterogeneous elastic half-space. To overcome challenges imposed by the large range of spatial and temporal scales, we use the fault slip rate as a threshold for switching between two different numerical solvers. During the interseismic phase, we use a quasi-static approximation where large linear systems of equations must be solved to allow for large time steps. Previous approaches to simulating the co-seismic phase enforced a rate-and-state friction law that resulted in a stiff system of equations that required a special, semi-implicit time-stepping method. Here, we apply a newly developed non-stiff method that is compatible with the interseismic method and allows for traditional explicit time stepping methods. Our simulations are verified by rigorous convergence tests and code-comparison exercises, and are an important step towards advancing earthquake cycle simulations incorporating more realistic physics.

J. E. Kozdon
Department of Applied Mathematics,
Naval Postgraduate School,
833 Dyer Road,
Monterey, CA 93943–5216
E-mail: jekozdon@nps.edu

B. A. Erickson and T. W. Harvey
Computer and Information Science
1202 University of Oregon
1477 E. 13th Ave.
Eugene, OR 97403–1202
E-mail: {bae, tharvey2}@uoregon.edu

1 Introduction and Literature Review

Simulations of multiple earthquake cycles on complex geometries will allow geophysicists to answer questions about how stress fields from previous earthquakes will affect future earthquakes on the same fault and neighboring faults, and to make hazard assessments using information about earthquake recurrence times. Historically, solving multi-cycle models at high resolution has been computationally infeasible due to the range of spatial and temporal scales, so modelers have resorted to addressing different phases of a cycle individually, or to using approximations over large domains. Previous models could be divided into two categories: dynamic rupture simulations and earthquake simulators. Dynamic rupture simulations model co-seismic behavior where fault mechanics are coupled with momentum balance in the off-fault volume, producing rupture dynamics over timescales of seconds in regional domains. Alternatively, earthquake simulators, which are commonly used for hazard analysis, run simulations for tens of thousands of years, over hundreds of thousands of km, but make many simplifying physical assumptions (see for instance Richards-Dinger and Dieterich (2012)). As Lapusta and Rice (2003) point to, the main reason for this division becomes obvious when considering the different spatial and temporal scales that must be resolved in a single simulation. In the spatial domain, fault depths and lengths range from tens to thousands of kilometers, while nucleation zones are realistically only a few meters. Temporally, tectonic loading occurs at rates of millimeters per year, resulting in ruptures that occur at hundred to thousand year increments, yet slip velocities during rupture occur at rates of meters per second, and ruptures last for tens of seconds. A back of the envelope calculation shows that numerically resolving the smallest spatio-temporal scales in a simulation over the largest scales of interest is completely computationally infeasible. Hence, we can investigate short co-seismic periods through dynamic rupture simulation, or large scale (but with simplified physics) through earthquake simulators.

More recently, the Sequences of Earthquakes and Aseismic Slip (SEAS) project has been attempting to address these shortcomings by running simulations on regional sized faults over thousands of years, while maintaining accurate modeling principles. Over the past two decades, many SEAS type simulations have been developed such as Lapusta et al. (2000), Barbot (2019), Segall and Bradley (2012) etc. Many of these simulators use the boundary element method (BEM) to discretize the governing physical equations, and BEM is considered the defacto method for solving many SEAS like problems. However, the BEM method can only solve problems in isotropic linear elastic volumes, ultimately limiting the modeler to simpler problems, when it is known that anisotropy and inelasticity play a crucial role in earthquake cycles (Best Mckay et al. (2019), Erickson et al. (2017)). As an alternative, Erickson and Dunham (2014) lays the ground work for a Finite Difference (FD) framework for SEAS problems, which do not suffer from these limitations. They spatially discretize a quasi-static approximation of the dynamic problem using Summation-by-Parts (SBP) FD operators and impose boundary conditions through simultaneous-approximation-terms (SATs). Their approximation neglects inertial terms and instead estimates

the kinetic energy lost out of the fault with a radiation damping term. For time-stepping, they derive an adaptive method that allows for large time-steps to be taken during periods of slow tectonic loading. While their method overcomes the limitations of the BEM method, Thomas et al. (2014) show that there are significant discrepancies between simulations with full dynamics and radiation damping approximations, especially when additional physics such as flash heating is added in the dynamic phase. It is therefore important that FD solvers incorporate full dynamics as well.

In order to simulate full dynamic effects in the co-seismic period a two-solver approach can be taken. During the interseismic period when wave effects are unimportant a quasi-dynamic solver like the one in Erickson and Dunham (2014) can be used, and a dynamic solver can be switched to right before rupture. Duru et al. (2019) takes this approach, using Erickson and Dunham (2014), and then in the co-seismic period impose a rate-state friction law through a Neumann boundary condition using SATs. Rate-and-state laws give traction on a fault in terms of slip velocity, and in 1D a linearized law reduces to:

$$\frac{\partial u}{\partial x} = -\alpha \dot{u}, \tag{1}$$

where traction is $\frac{\partial u}{\partial x}$, slip rate is $\dot{u}$, and α is a time-varying coefficient. As Kozdon et al. (2021a) point out, α can range over many orders of magnitude during an earthquake, ultimately leading to numerical stiffness. Duru et al. (2019) addresses this stiffness with a custom time-stepping scheme in which fault conditions are integrated implicitly, and the volume is integrated with a second-order-accurate explicit scheme. Kozdon et al. (2012) takes an alternative approach by considering characteristic variables on the boundary, namely

$$\dot{u} - \frac{\partial u}{\partial x} = R\left(\dot{u} + \frac{\partial u}{\partial x}\right), R = \frac{1-\alpha}{1+\alpha}, \tag{2}$$

where the incoming characteristic is $\dot{u} - \frac{\partial u}{\partial x}$ and the outgoing characteristic is $\dot{u} + \frac{\partial u}{\partial x}$, and $-1 \leq R \leq 1$ is the reflection coefficient. They first show that nonlinear friction laws can be rewritten in terms of characteristic variables, and then remove the stiffness from the characteristic formulation with the equations of motion in first order form. Another advantage of the characteristic formulation is that it supports Neumann ($R = 1$), Dirichlet ($R = -1$), and non-reflecting ($R = 0$) boundary conditions, all with the same equation, which proves useful for earthquake simulations. While the removal of the stiffness in Kozdon et al. (2012) is a step in the right direction, it is only done for the first order form of the equations where stress and velocity must be integrated. This does not allow for a seamless transition between solvers, since Erickson and Dunham (2014) take a displacement-based approach. The first order stress/velocity formulation also requires more variables to be integrated. Kozdon et al. (2021a) revisits the characteristic formulation and removes the stiffness from the second order form. This allows for a generic explicit time-stepper of arbitrary-order accuracy on the dynamic problem, and an easy transition between the quasi-static and dynamic problems.

The main contribution of this work is the merging of the quasi-static time-stepping approach in Erickson and Dunham (2014), a new spatial discreteization of the quasi-static problem from Kozdon et al. (2021b), and the dynamic approach of Kozdon et al. (2021a) to simulate multiple earthquake cycles with high-order accuracy on complex geometries. In section 2, we introduce 1D SBP-operators. In section 3, we show that a characteristic formulation in 1D of the dynamic problem is non-stiff. In section 4, describe the governing equations of a 2D cycle simulation. In section 5, we explain how the governing equations of a 2D simulation can be transformed to solve the problem on complex geometries. In section 6, we introduce 2D SBP-operators. In section 7, we describe our solution method to the multi-cycle problem. In section 8, we present the results of our simulations, and in section 9 we give some conclusions.

2 1D Summation-By-Parts (SBP) Operators and Simultaneous Approximation Terms (SATs)

SBP operators are finite difference operators that discretely approximate derivatives, and mimic integration-by-parts identities. In conjunction with boundary enforcement through SATs, these operators allow for a discrete energy estimate that can be used to prove discrete stability as an analogue to the stability of a continuous problem.

Let the domain $0 \leq r \leq 1$ be partitioned with $N+1$ equally spaced nodes so that the distance between each node is $h = 1/N$. We denote the projection of a function u on the resulting grid points as $\boldsymbol{u}^T = \left[u_0, u_1 \cdots u_{n+1}\right]$. We also define the restriction operator $\boldsymbol{e}_k^T$, which takes a grid function to its value at $r(hk)$, as a vector of zeros except for a one at index k. While sometimes these operators are useful, sometimes they are cumbersome, so we use them interchangeably with $\boldsymbol{e}_k^T \boldsymbol{u} = u_k$. A first derivative operator $\boldsymbol{D}_1$ is called SBP if:

$$\boldsymbol{D}_1 = \boldsymbol{H}^{-1}\boldsymbol{Q} \approx \frac{\partial}{\partial r} \tag{3}$$

and

$$\boldsymbol{Q} + \boldsymbol{Q}^T = \begin{bmatrix} -1 & & & \\ & 0 & & \\ & & \ddots & \\ & & & 0 \\ & & & & 1 \end{bmatrix}, \tag{4}$$

where $\boldsymbol{H}$ is diagonal and defines a norm $||u||_H^2 = \boldsymbol{u}^T \boldsymbol{H} \boldsymbol{u}$. It also has the nice property of being a quadrature approximation. This definition gives the identity:

$$\boldsymbol{u}^T \boldsymbol{D}_1 \boldsymbol{v} = u_N v_N - u_0 v_0 - \boldsymbol{u}^T \boldsymbol{D}_1^T \boldsymbol{v}, \tag{5}$$

which is the discrete analog to the integration-by-parts identity:

$$\int_0^1 u \frac{\partial v}{\partial r} dx = uv|_0^1 - \int_0^1 \frac{\partial u}{\partial r} v \tag{6}$$

.

A variable coefficient $C(r)$ second derivative operator $\boldsymbol{D}_2^C$ is called SBP if:

$$\boldsymbol{D}_2^C = \boldsymbol{H}^{-1}(-\boldsymbol{A}^C + C_N \boldsymbol{e}_N \boldsymbol{b}_N^T - C_0 \boldsymbol{e}_0 \boldsymbol{b}_0^T) \approx \frac{\partial}{\partial r}\left(C(r)\frac{\partial}{\partial r}\right), \tag{7}$$

where $\boldsymbol{A}^C$ is a symmetric positive definite matrix with standard center difference operators and a damping operator for spurious oscillations. Mattsson and Nordström (2004) give a full explanation of the advantages of using $\boldsymbol{A}^C$ over the application of a first derivative operator twice. $\boldsymbol{b}_k^T$ computes an approximation to the first derivative at grid point k, and is not necessarily the first and last row of $\boldsymbol{D}_1$ (Mattsson and Parisi (2010)). (7) then leads to the identity

$$\boldsymbol{u}^T \boldsymbol{H} \boldsymbol{D}_2^C \boldsymbol{v} = C_N(\boldsymbol{e}_N^T \boldsymbol{u})\boldsymbol{b}_N^T \boldsymbol{v} - C_0(\boldsymbol{e}_0^T \boldsymbol{u})\boldsymbol{b}_0^T \boldsymbol{v} - \boldsymbol{u}^T \boldsymbol{M}^C \boldsymbol{v} \tag{8}$$

,

which is the discrete analog to the continuous identity

$$\int_0^1 u \frac{\partial}{\partial r}\left(\mu \frac{\partial v}{\partial r}\right) dx = uv|_0^1 - \int_0^1 \frac{\partial u}{\partial r}\mu \frac{\partial v}{\partial r}. \tag{9}$$

SATs are used to enforce boundary conditions weakly. Instead of injecting boundary data over the nodal values, these nodal values relax towards the boundary data over timescales that decrease with grid spacing. These terms in 1D correspond to inhomogeneous ODEs in time that depend on a penalty parameter which determines stability.

3 Dynamic 1D Example Problem

In order to show that the method in Kozdon et al. (2021a) produces a non-stiff system of ODEs, we apply both a standard SBP-SAT discretization and the new method to the 1D wave equation with characteristic and Neumann boundary conditions:

$$\rho \ddot{u} = \frac{\partial}{\partial x}\left(\mu \frac{\partial u}{\partial x}\right) \qquad x \in [0,1] \tag{10a}$$

$$\dot{u} + \tau = R(\dot{u} - \tau) \qquad x = 1 \tag{10b}$$

$$\tau = 0 \qquad x = 0 \tag{10c}$$

$$u(x,0) = I(x) \qquad x \in [0,1] \tag{10d}$$

Where dots denote derivatives in time, μ and ρ are spatially varying parameters, u is the particle displacement, $0 \leq R \leq 1$ is the reflection coefficient, τ is the traction computed as $\tau = n\mu\frac{\partial u}{\partial x}$ where n is a normal direction (-1 or 1), and I is an initial condition.

(10b) is a characteristic boundary condition. We will refer to $\dot{u} + \tau$ as the incoming characteristic, and $\dot{u} - \tau$ as the outgoing characteristic for the following

reason. For the moment, assuming μ and ρ are constant in space, a diagonalization of (10a) in first order form leads to a system of advection equations, one moving the quantity $\dot{u} + \tau$ into the domain, and the other moving $\dot{u} - \tau$ out of the domain. Therefore (10b) can be thought of as enforcing that some fraction R of the outgoing characteristic is sent back in the domain. We also note that with spatially variable μ and ρ, $\dot{u}+\tau$ and $\dot{u}-\tau$ are not the exact characteristics, so (10b) becomes an approximation to a true characteristic boundary condition.

As mentioned earlier, for different values of R we can produce different boundary conditions. Namely, $R = 1$ is a free-surface condition, $R = 0$ is a non-reflecting condition, and $R = -1$ is a zero-Dirichlet condition. This is extremely important since our multi-cycle simulation uses all three conditions.

A traditional SBP-SAT discretization of this problem is detailed in Mattsson et al. (2009), namely

$$\boldsymbol{\rho}\ddot{\boldsymbol{u}} = \boldsymbol{D}_2^{\mu}\boldsymbol{u} - \boldsymbol{H}^{-1}\alpha_0\boldsymbol{e}_0(\mu_0\boldsymbol{b}_0^T\boldsymbol{u}) + \alpha_N\boldsymbol{e}_N((1+R)(\mu_N\boldsymbol{b}_N^T\boldsymbol{u}) + (1-R)\dot{\boldsymbol{u}}_N) \quad (11)$$

In Mattsson et al. (2009) they derive bounds on α_0 and α_N for stability, which in the case of a characteristic boundary condition on the right side forces $\alpha_N = \frac{-1}{1+R}$. Plugging this value into (11) produces a term with $\frac{1-R}{1+R}$, so that when R is close to -1 a very large negative eigenvalue exists, leading to numerical stiffness.

The method from Kozdon et al. (2021a) also uses SBP operators, but enforces boundary conditions with flux terms τ^* and u^*. Fluxes act very similarly to SATs, but through the two degrees of freedom given by the fluxes conditions on both traction and displacement can be imposed simultaneously. Using this method, the resulting discretization is

$$\begin{aligned}\ddot{\boldsymbol{u}} &= \boldsymbol{D}_2^{\mu}\boldsymbol{u} + \boldsymbol{H}^{-1}\boldsymbol{e}_N(\tau_N^* - \boldsymbol{b}_N^T\boldsymbol{u}) + \boldsymbol{H}^{-1}\boldsymbol{e}_0(\tau_0^* + \boldsymbol{b}_0^T\boldsymbol{u}) \\ &- \boldsymbol{H}^{-1}\boldsymbol{b}_N(u_0^* - \boldsymbol{e}_N^T\boldsymbol{u}) + H^{-1}b_0(u_0^* - \boldsymbol{e}_0^T\boldsymbol{u}).\end{aligned}$$

This formulation allows us to simultaneously impose conditions on both traction τ and displacement u through the fluxes on the boundaries. We enforce that the fluxes satisfy (10b), and that the outgoing characteristic is maintained:

$$\dot{u}_i^* + \tau_i^* = R_i(\dot{u}^* - \tau^*) \quad (12a)$$

$$\dot{u}_i^* - \tau_i^* = \dot{\boldsymbol{u}}_i - \tau_i, \quad (12b)$$

where $i = 0, N$, and the numerical traction τ_i is computed with the boundary derivative and a penalty parameter, namely

$$\tau_i = n_i b_i^T\boldsymbol{u} + \alpha(u_i^* - \boldsymbol{u}_i). \quad (13)$$

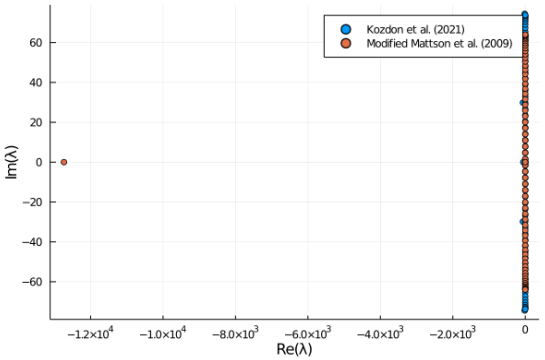
Solving the linear system (12) for the fluxes results in:

$$\dot{u}_i^* = \frac{1+R_i}{2}\left(\dot{\boldsymbol{u}}_i - \tau_i\right) \quad (14a)$$

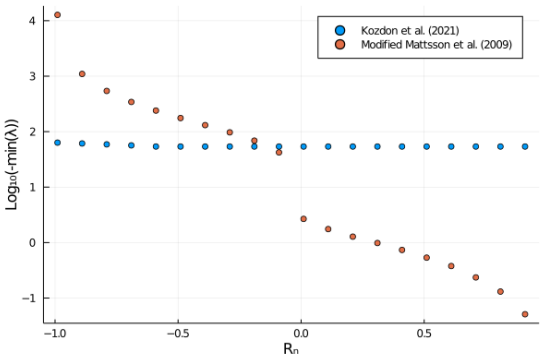
$$\tau_i^* = \frac{1-R_i}{2}\left(\dot{\boldsymbol{u}}_i - \tau_i\right). \quad (14b)$$

To enforce the Neumann condition (10c) we set $R_0 = 1$, and to impose (10b) we make $R_N = R$. Additionally, we note that imposing characteristic boundary conditions results in another set of equations (14a) which must be integrated to yield u^*.

In Figure 1a, we show the eigenvalues of the semi-discrete systems of both methods when $R_N = .99$. The removal of the large negative eigenvalue in the standard discretization signals that the new method produces a non-stiff system. Additionally, sweeping through the values of R in Figure 1b shows that the method produces small eigenvalues regardless of the value of R.



(a) Spectrum of 33 node second-order discretization with $R_N = -.99$



(b) The same discretization as (a), but varying R from -.99 to 1

Fig. 1: Comparison of the eigenvalue spectra between the Kozdon et al. (2021a) and the modified Mattsson et al. (2009) methods.

4 2D Earthquake Problem

The new dynamic method furnishes a non-stiff system of ODEs, even when a rate-state law is imposed (Kozdon et al. (2012)), making it feasible to add a

fully dynamic phase to a cycle simulation. In this next section we describe an anti-plane cycle problem.

We consider a homogeneous, isotropic, linear elastic half space $(x, y, z) \in (-\infty, \infty) \times (-\infty, \infty) \times (0, \infty)$. The $y = 0$ plane is taken to be Earth's free surface, and a strike-slip fault is embedded in the $x = 0$ plane. We make the simplifying assumption of anti-plane strain, which means that non-zero displacements $u(x, y, t)$ only occur in the z direction. Secondly, due to the anti-symmetry of the solution over the $x = 0$ plane we consider only the positive half space. The problem is then formulated as an initial-boundary value problem (IBVP) governed by conservation of momentum:

$$\rho\ddot{u} = \frac{\partial \sigma_{xz}}{\partial x} + \frac{\partial \sigma_{yz}}{\partial y} \qquad (x, y) \in (0, \infty) \times (0, \infty), \tag{15}$$

where ρ is the density of the volume material. We enforce Hookes law which relates stresses σ_{xz} and σ_{yz} to strains with shear modulus μ:

$$\sigma_{xz} = \mu \frac{\partial u}{\partial x} \qquad \sigma_{yz} = \mu \frac{\partial u}{\partial y}. \tag{16}$$

Along the fault we impose a rate-and-state friction law down to a depth of W kilometers, below which we impose a slow slip condition, as can be seen in Figure 2. Together these conditions form a boundary condition on $x = 0$:

$$\tau(y, t) = F(V, \psi) \qquad x = 0, 0 \le y \le W \tag{17a}$$

$$\dot{\psi}(y, t) = G(V, \psi) \qquad x = 0, 0 \le y \le W \tag{17b}$$

$$V(y, t) = V_p \qquad x = 0, y \ge W, \tag{17c}$$

where τ is defined by a background stress τ^0 added to stress caused by deformation $\Delta\tau$ on the fault. Equation (17a) relates the shear stress on the fault τ to frictional strength F via an experimentally motivated friction law. The slip velocity is $V = 2\dot{u}(0, y, t)$ and ψ is a state variable. ψ evolves according to (17b). F and G take the forms

$$F(V, \psi) = \sigma_n a \sinh^{-1}\left(\frac{V}{2V_0} e^{\frac{\psi}{a}}\right) \tag{18a}$$

$$G(V, \psi) = \frac{bV_0}{D_c} e^{\frac{f_0 - \psi}{b} - \frac{|V|}{V_0}} \tag{18b}$$

Here, σ_n is the normal stress on the fault, V_0 is a steady state background slip rate on the fault, a and b are parameters determined by the material properties of the rock, and D_c is a characteristic slip distance over which the fault returns to steady state sliding after a perturbation in slip velocity.

The sign of $b - a$ determines if the fault is velocity strengthening $(b - a < 0)$ or weakening $(b - a > 0)$ (Segall (2010)). Within velocity strengthening zones rupture is inhibited as the fault strength increases with slip. The opposite is true of velocity weakening zones. In order to facilitate rupture at the surface and slow slip at depth, we make a vary along the fault with:

$$a(y) = \begin{cases} a_0 & 0 \le y < H_w \\ a_0 + (a_{\max} - a_0)(y - H_w)/H_t & H \le y < H_w + H_t \\ a_{\max} & H_w + H_t \le y < W \end{cases} \tag{19}$$

For the values of all the parameters above see Table 1.

Parameter	Value
ρ	2.67 Mg/m^3
μ	32.038 Gpa
σ_n	50 MPa
a_0	.010
$a_{\max}$	.025
b_0	.025
D_c	.008 m
V_p	10^{-9} m/s
V_0	10^{-6} m/s
f_0	.6
H_w	15 km
H_t	3 km
W	40 km

Table 1: Parameter values for BP1

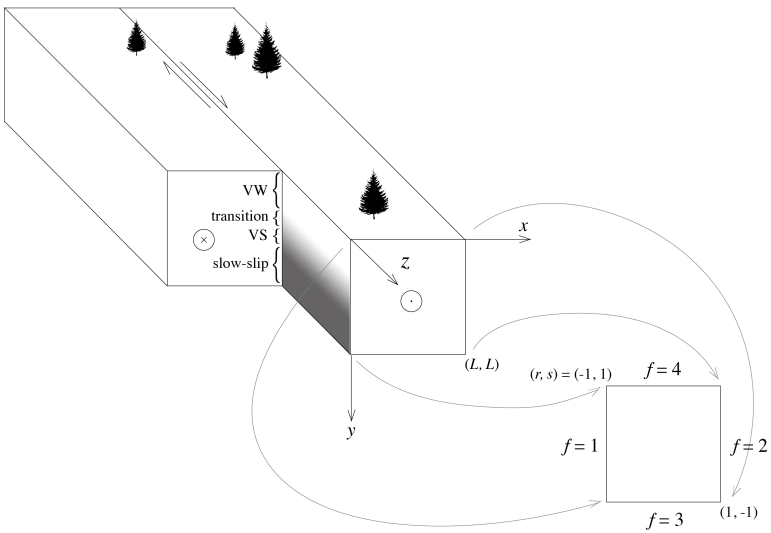


Fig. 2: The strike-slip fault is embedded vertically; anti-plane shear motion is assumed, defining the computational domain with four faces.

5 Domain Transformation

Some sections of the domain in (15) require higher spatial resolution that others. In Rice et al. (2001) they derive a length scale h^* that must be resolved for rupture to occur. Since rupture occurs at depths above 18km, it is computationally efficient to apply a grid stretching below these depths, where h^* does not need to be resolved. To do this we apply a coordinate transformation between the physical domain $x(r,s)$ and $y(r,s)$ and a logical domain $r(x,y), s(x,y) \in (-1,1)\times(-1,1)$. All finite difference computations are done on the logical domain, and are transformed back to the physical domain if need be. Letting J denote the Jacobian determinant

$$J = \frac{\partial x}{\partial r}\frac{\partial y}{\partial s} - \frac{\partial y}{\partial r}\frac{\partial x}{\partial s}, \tag{20}$$

the isotropic problem (15) in physical space can be recast as an anisotropic problem in the computational domain under a transformation as:

$$J\rho\ddot{u} = \left[\tfrac{\partial}{\partial r}\ \tfrac{\partial}{\partial s}\right] C \begin{bmatrix} \frac{\partial u}{\partial r} \\ \frac{\partial u}{\partial s} \end{bmatrix}, \tag{21}$$

where C is a now a (2×2) matrix-valued function. For a full proof of (21) and for the explicit form of C see appendix A.

Dirichlet, Neumann and characteristic boundary conditions will need to be imposed on the computational domain to simulate BP1. Dirichlet conditions are unaffected by a coordinate transformation, but Neumann boundary conditions such as:

$$\tau = \sigma_{xz}n_1 + \sigma_{yz}n_2 = g_n \tag{22}$$

where n is the normal vector along a physical face, becomes

$$\hat{\tau} = \hat{n}^T C \begin{bmatrix} \frac{\partial u}{\partial r} \\ \frac{\partial u}{\partial s} \end{bmatrix} = S_J g_n \tag{23}$$

in logical space, where $\hat{n}$ is the normal vector along the computational face and S_J is the surface Jacobian. S_J satisfies the relationships:

$$S_J n_1 = J\frac{\partial r}{\partial x}\hat{n}_1 + J\frac{\partial s}{\partial x}\hat{n}_2 \tag{24a}$$

$$S_J n_2 = J\frac{\partial r}{\partial y}\hat{n}_1 + J\frac{\partial s}{\partial y}\hat{n}_2 \tag{24b}$$

.

Characteristic boundary conditions have a traction term in them and therefore also use (23) as seen later.

6 2D SBP Operators

In this section we describe the construction of the 2D operators. Describing 2D operators poses some notational complications. We therefore hold the following conventions: vector subscripts indicate an indexing into that vector. For matrix operators a subscript denotes the direction in which that operator acts. A normal superscript denotes a grid line if there is a colon, otherwise it denotes the face f that matrix is acting on. If there are parentheses around the superscript it denotes that the matrices' construction is dependent on the superscripted function. The operators are usually constructed via the Kronecker product:

$$\boldsymbol{A} \otimes \boldsymbol{B} = \begin{bmatrix} a_{11}\boldsymbol{B} & \dots & a_{1n}\boldsymbol{B} \\ \vdots & \ddots & \vdots \\ a_{m1}\boldsymbol{B} & \dots & a_{mn}\boldsymbol{B} \end{bmatrix} \tag{25}$$

We describe the second derivative operators on the block $(r, s) = [-1, 1] \times [-1, 1]$, where faces 1 and 2 are the right and left faces, and faces 3 and 4 are the top and bottom faces (see Figure 2). We let the domain be discretized with $(N+1)\times(N+1)$ evenly spaced grid points a distance of $h = 2/N$ apart. The projection of u onto the grid is again denoted $\boldsymbol{u}^T = \big[\big[\boldsymbol{u}_{00} \dots \boldsymbol{u}_{0N}\big] \dots \big[\boldsymbol{u}_{N0} \dots \boldsymbol{u}_{NN}\big]\big]$, where $\boldsymbol{u}_{kl} \approx u(kh, lh)$ and is stored as a vector with r being the fastest index.

We define the face restriction operators as:

$$\boldsymbol{L}^1 = \boldsymbol{I} \otimes \boldsymbol{e}_0^T \qquad \boldsymbol{L}^2 = \boldsymbol{I} \otimes \boldsymbol{e}_N^T \qquad \boldsymbol{L}^3 = \boldsymbol{e}_0^T \otimes \boldsymbol{I} \qquad \boldsymbol{L}^4 = \boldsymbol{e}_0^T \otimes \boldsymbol{I} \tag{26}$$

where $\boldsymbol{I}$ is the $(N+1)\times(N+1)$ identity matrix. More generally, the restriction to a single grid line in the r and s directions, respectively, are

$$\boldsymbol{L}^{l:} = \boldsymbol{e}_l^T \otimes \boldsymbol{I} \qquad \boldsymbol{L}^{:l} = \boldsymbol{I} \otimes \boldsymbol{e}_l^T. \tag{27}$$

In order to construct $\boldsymbol{A}_{rr}^{C_{rr}}$ and $\boldsymbol{A}_{ss}^{C_{ss}}$ we must construct individual one-dimensional second derivative matrices for each grid line with varying coefficients C and place them in the correct block. Expanding a single second derivative matrix with the Kronecker product and the identity matrix only works in the constant coefficient case.

To do this it is useful to define $\boldsymbol{C}_{ij} = \operatorname{diag}(\boldsymbol{c}_{ij})$ where $\boldsymbol{c}_{ij}$ is the projection of $C_{ij}(r, s)$ onto the grid, and denote the coefficients along the individual grid lines as

$$\boldsymbol{C}_{ij}^{:l} = \operatorname{diag}(C_{ij}(0, hl), \dots, C_{ij}(hN, hl)) \qquad \boldsymbol{C}_{ij}^{k:} = \operatorname{diag}(C_{ij}(hk, 0), \dots, C_{ij}(hk, hN)). \tag{28}$$

The interior second derivative operators can then be defined as the sum of 1D operators along each grid line:

$$\boldsymbol{A}_{rr}^{(C_{rr})} = (\boldsymbol{H} \otimes \boldsymbol{I}) \left[\sum_{l=0}^{N} \left(\boldsymbol{L}^{:l}\right)^T \boldsymbol{A}_2^{\left(\boldsymbol{C}_{rr}^{:l}\right)} \boldsymbol{L}^{:l} \right] \tag{29a}$$

$$\boldsymbol{A}_{ss}^{(C_{ss})} = (\boldsymbol{I} \otimes \boldsymbol{H}) \left[\sum_{k=0}^{N} \left(\boldsymbol{L}^{k:}\right)^T \boldsymbol{A}_2^{\left(\boldsymbol{C}_{ss}^{k:}\right)} \boldsymbol{L}^{k:} \right] \tag{29b}$$

Operator	Dimensions
$\boldsymbol{e}_k$, $\boldsymbol{b}_k$	$(N+1)\times 1$
$\boldsymbol{Q}$, $\boldsymbol{D}_1$,$\boldsymbol{D}_2$,$\boldsymbol{H}$,$\boldsymbol{C}^{\cdot}$	$(N+1)\times(N+1)$
$\boldsymbol{L}^{\cdot}$,$\boldsymbol{B}^f$	$(N+1)\times(N+1)^2$
$\boldsymbol{A}_{ij}^{C_{ij}}$, $\boldsymbol{D}_{ij}^{C_{ij}}$, $\boldsymbol{C}_{ij}$	$(N+1)^2\times(N+1)^2$

Table 2: Dimensions of Operators

and the mixed derivative operators as:

$$\boldsymbol{A}_{rs}^{(C_{rs})} = (\boldsymbol{I}\otimes\boldsymbol{Q}^T)\boldsymbol{C}_{rs}(\boldsymbol{Q}\otimes\boldsymbol{I}) \tag{29c}$$
$$\boldsymbol{A}_{sr}^{(C_{sr})} = (\boldsymbol{Q}^T\otimes\boldsymbol{I})\boldsymbol{C}_{sr}(\boldsymbol{I}\otimes\boldsymbol{Q}) \tag{29d}$$

.

The boundary derivatives parallel to a face f are given with the one-dimensional first derivative operators $\boldsymbol{D}_1$,

$$\boldsymbol{B}_s^1 = \boldsymbol{e}_0^T\boldsymbol{D}_1\otimes\boldsymbol{I} \tag{30a}$$
$$\boldsymbol{B}_s^2 = \boldsymbol{e}_N^T\boldsymbol{D}_1\otimes\boldsymbol{I} \tag{30b}$$
$$\boldsymbol{B}_r^3 = \boldsymbol{I}\otimes\boldsymbol{e}_0^T\boldsymbol{D}_1 \tag{30c}$$
$$\boldsymbol{B}_r^4 = \boldsymbol{I}\otimes\boldsymbol{e}_N^T\boldsymbol{D}_1 \tag{30d}$$

and those perpendicular to the boundary using the boundary first derivative operators $\boldsymbol{b}_0$ and $\boldsymbol{b}_N$ from the second derivative operator:

$$\boldsymbol{B}_r^1 = \boldsymbol{I}\otimes\boldsymbol{b}_0^T \tag{31a}$$
$$\boldsymbol{B}_r^2 = \boldsymbol{I}\otimes\boldsymbol{b}_N^T \tag{31b}$$
$$\boldsymbol{B}_s^3 = \boldsymbol{b}_0^T\otimes\boldsymbol{I} \tag{31c}$$
$$\boldsymbol{B}_s^4 = \boldsymbol{b}_N^T\otimes\boldsymbol{I}. \tag{31d}$$

The full 2D second derivative operators with boundary closures are then defined as:

$$\boldsymbol{D}_{ij}^{(C_{ij})} = (\boldsymbol{H}\otimes\boldsymbol{H})^{-1}\left[-\boldsymbol{A}_{ij}^{(C_{ij})} + \sum_{f=2i-1}^{2i}\hat{n}_i^f(\boldsymbol{L}^f)^T\boldsymbol{H}\boldsymbol{C}^f\boldsymbol{B}_j^f\right]. \tag{32}$$

To better clarify how the operators work we list the dimensions of each operator in table 2.

7 Numerical Method

We implement two solvers, one for the interseismic period and one for the co-seismic period and link them together through a specified switching criterion. At the end of a phase, each solver passes their "final" data to the other solver as initial data, until the simulated time period has ended.

7.1 Quasi-Static Solver for Interseismic Period

The necessity of using a quasi-static solver during the interseismic period can be seen by considering the wave speed $c_s = \sqrt{\mu/\rho} = 3.464$ m/s. In order to fully resolve wave propagation a time-step on the order of h/c_s seconds is needed. This is an upper bound not taking into account the time-stepping method or grid transformation. This is computationally infeasible for simulations of thousands of years. Therefore, during the interseismic period, where wave propagation effects matter little, we neglect inertial terms in (15) and approximate the stress drop on the fault due to kinetic energy lost to the volume with a radiation damping term ηV, where $\eta = \mu/2c_s$. These considerations result in the following time dependent boundary value problem:

$$
\begin{aligned}
0 &= \left[\tfrac{\partial}{\partial r} \ \tfrac{\partial}{\partial s}\right] C \begin{bmatrix} \frac{\partial u}{\partial r} \\ \frac{\partial u}{\partial s} \end{bmatrix} && (r,s) \in (0,1)\times(0,1) && \text{(33a)}\\
\hat{\tau} &= S_J[F(V,\psi) + \eta V - \tau^0] && f = 1, 0 \le s \le s(0,W) && \text{(33b)}\\
\dot{\psi} &= G(V,\psi) && f = 1, 0 \le s \le s(0,W) && \text{(33c)}\\
V &= V_p && f = 1, s(0,W) \le s && \text{(33d)}\\
\frac{\partial u}{\partial t} &= V_p/2 && f = 2 && \text{(33e)}\\
\hat{\tau} &= 0 && f = 3 && \text{(33f)}\\
\hat{\tau} &= 0 && f = 4 && \text{(33g)}\\
& && && \text{(33h)}
\end{aligned}
$$

Here, (33g), (33f), and (33e) correspond to the surface of the Earth, a free surface at depth, and a remote slow loading condition, respectively.

First, focusing on the problem at any single time-step, we discretize (33a) subject to Dirichlet and Neumann boundary conditions. This requires a divergence operator and enforcement of boundary conditions through SAT terms. SAT terms are both dependent on $\boldsymbol{u}$, and boundary data. We therefore decompose the SATs into $\boldsymbol{b}^f$ vectors which contain any boundary data, and the $\boldsymbol{K}$ matrix where we insert any $\boldsymbol{u}$ dependent coefficients from the SAT vectors. This results in the system of linear equations:

$$
0 = (\boldsymbol{D}_{rr}^{(C_{rr})} + \boldsymbol{D}_{rs}^{(C_{rs})} + \boldsymbol{D}_{sr}^{(C_{sr})} + \boldsymbol{D}_{ss}^{(C_{ss})} + \boldsymbol{K})\boldsymbol{u} + \sum_{f=1}^{4} \boldsymbol{b}^f, \tag{34}
$$

where the sum of $\boldsymbol{D}_{ij}^{(C_{ij})}$'s approximates the divergence. For the explicit form of $\boldsymbol{b}^f$ and $\boldsymbol{K}$ and a stability proof see Kozdon et al. (2021b), and Erickson and Dunham (2014). (34) is a system of linear equations which is solved for particle displacements $\boldsymbol{u}$ within the domain. This linear system solve, using a Cholesky decomposition, is embedded in the full time-stepping algorithm.

To evolve the system forward in time we integrate the boundary conditions governing the fault then add them back into $\boldsymbol{b}^f$. The full time-stepping algorithm can be summarized as follows:

1. Use the current slip on the fault $\boldsymbol{\delta} = 2\boldsymbol{u}$ (initially $\boldsymbol{\delta} = \boldsymbol{0}$) to set a displacement boundary condition on face 1. Also, compute the boundary conditions (33g), (33f), and (33e) using (23).
2. Generate the linear system (34) where each of the boundary conditions in 1 determines the form of $\boldsymbol{b}^f$ and solve it for $\boldsymbol{u}$.
3. Use these new displacements to compute $\hat{\tau}$ in (33b). Then solve (33b), a nonlinear equation, at every point along the fault for V. We do this using a bracketed Newton method.
4. Integrate V and (33d) using an explicit adaptive RK4 method to give a new slip $\boldsymbol{\delta}$ along the fault. Repeat for the next time-step.

7.2 Dynamic Solver for Co-Seismic Period

Right before rupture we switch to a dynamic solver; the method is similar to the one laid out in section 2, but is extended to two dimensions with the fault on face 1 again. The full problem during the dynamic phase is:

$$
\begin{aligned}
J\rho\ddot{u} &= \left[\tfrac{\partial}{\partial r}\ \tfrac{\partial}{\partial s}\right] C \begin{bmatrix} \frac{\partial u}{\partial r} \\ \frac{\partial u}{\partial s} \end{bmatrix} && (r,s) \in (0,1)\times(0,1) && \text{(35a)}\\
\hat{\tau} &= S_J[F(V,\psi) - \tau^0] && f = 1, 0 \le s \le s(0,W) && \text{(35b)}\\
\dot{\psi} &= G(V,\psi) && f = 1, 0 \le s \le s(0,W) && \text{(35c)}\\
V &= V_p && f = 1, s(0,W) \le s && \text{(35d)}\\
\hat{Z}\dot{u} + \hat{\tau} &= R^f(\hat{Z}\dot{u} - \hat{\tau}) && f = 2,3,4 && \text{(35e)}
\end{aligned}
$$

.

Off-fault boundaries are handled with characteristic conditions (35e), where $\hat{Z} = \sqrt{\mu\rho}$ is the shear impedance. We impose $(R^2, R^3, R^4) = (0, 1, 0)$, so that face 3 is a free surface, and faces 2 and 4 are non-reflecting.

A discretization of (35a) using the method from Kozdon et al. (2021a) yields the system:

$$
\begin{aligned}
\boldsymbol{J\rho\ddot{u}} &= (\boldsymbol{D}_{ij}^{(C_{ij})})\boldsymbol{u} \\
&+ \sum_{f=1}^{4} (\boldsymbol{H}\otimes\boldsymbol{H})^{-1}(\boldsymbol{L}^f)^T\boldsymbol{H}^f(\hat{\boldsymbol{\tau}}^{*f} - \hat{n}_i^f\boldsymbol{C}_{ij}^f\boldsymbol{B}_j^f\boldsymbol{u}) \\
&- \sum_{f=1}^{4} (\boldsymbol{H}\otimes\boldsymbol{H})^{-1}(\boldsymbol{B}_j^f)^T\hat{n}_i^f\boldsymbol{C}_{ij}^f\boldsymbol{H}^f(\boldsymbol{u}^{*f} - \boldsymbol{L}^f\boldsymbol{u})
\end{aligned}, \qquad \text{(36)}
$$

where summation over i and j is implied. We again enforce the same conditions on the fluxes for characteristic conditions as we did in 1D on faces 2, 3, and 4:

$$\dot{\boldsymbol{u}}^{*f} = \frac{1+R^f}{2}\left(\boldsymbol{L}^f\dot{\boldsymbol{u}} - \hat{\boldsymbol{\tau}}^f\right) \tag{37a}$$

$$\boldsymbol{\tau}^{*f} = \frac{1-R^f}{2}\left(\boldsymbol{L}^f\dot{\boldsymbol{u}} - \hat{\boldsymbol{\tau}}^f\right) \tag{37b}$$

.

The numerical traction $\hat{\boldsymbol{\tau}}^f$ on a face is computed with

$$\hat{\boldsymbol{\tau}}^f = \hat{n}_i^f \boldsymbol{C}_{ij}^f \boldsymbol{B}_j^f \boldsymbol{u} + (\hat{n}_i^f \boldsymbol{C}_{ij}^f \hat{n}_i^f \boldsymbol{\Gamma}^f)(\boldsymbol{u}^{*f} - \boldsymbol{L}^f \boldsymbol{u}), \tag{38}$$

where $\boldsymbol{\Gamma}^f$ is a matrix storing penalty parameters for face f derived in Kozdon et al. (2021a) and (Almquist and Dunham 2020), and we again assume summation over i and j. To enforce the rate-state friction law we impose that the traction flux obeys the friction law, and that the outgoing characteristic is maintained along all points of face 1. At a single node this amounts to:

$$\tau^{*1} = S_J^1[F(2\dot{u}^{*1}, \psi) - \tau^0] \tag{39a}$$

$$\hat{Z}\dot{u}^{*1} - \tau^{*1} = \hat{Z}L^1\dot{u} - \hat{\tau}^1. \tag{39b}$$

To solve this system for $\dot{u}^{*1}$ we root-find (with a bracketed Newton method) on the equation:

$$0 = S_J[F(2\dot{u}^{*1}, \psi) - \tau^0] + \hat{\tau}^1 + \hat{Z}(\dot{u}^{*1} - L^1\dot{u}). \tag{40}$$

For state evolution (35c) we enforce that

$$\dot{\boldsymbol{\psi}} = G(2\dot{\boldsymbol{u}}^{*1}, \boldsymbol{\psi}). \tag{41}$$

We then apply a generic RK4 method to (36) $\dot{\boldsymbol{u}}^{*1}$, (37a), and (41) to step forward in time.

7.2.1 Convergence of scheme

Since the addition of the dynamic solver is new and un-tested, we use the method of manufactured solutions (MMS) to verify our code base (Roache (1998)). MMS works as follows: We assume an analytic solution to the problem with added boundary and volume source terms (not included in the model). We then compute what those source terms would be in the formulation (35), and add it in as sources in our numerical scheme. We can then compute the error between our analytic solution and our numerical solution with the source terms. Our manufactured solution is:

$$u_e(x, y, t) = \gamma \sin(\pi(x + y - t)) - (\gamma\pi/L + \epsilon)t + (\gamma\pi/L + \epsilon)x \tag{42}$$

N	$\text{err}_h = \|\|\Delta \boldsymbol{u}\|\|_{\boldsymbol{H}}$	rate $\left(\log_2\left(\frac{\text{err}_{h/2}}{\text{err}_h}\right)\right)$
second-order		
16	9.831847e-02	NA
32	1.706659e-02	2.526288
64	3.241969e-03	2.396233
128	7.988851e-04	2.020810
256	1.991652e-04	2.004022
fourth-order		
16	2.431269e-02	NA
32	1.719735e-03	3.821451
64	5.624746e-05	4.934255
128	3.920012e-06	3.842858
256	1.960141e-07	4.321829
sixth-order		
16	8.394578e+00	NA
32	4.916485e-04	14.059543
64	7.509679e-06	6.032732
128	2.079501e-07	5.174442
256	3.160370e-08	2.718072

Table 3: error and convergence rates for the dynamic scheme

Source values can be directly added to the volume and the off-fault boundaries through source vectors. To inject MMS values into the fault we notice that given (42) we can solve (18a) for an exact state equation:

$$\psi_e = a \log\left(\frac{2V_0}{2\dot{u}_e}\right) \sinh\left(\frac{\tau_e}{a\sigma_n}\right) \tag{43a}$$

$$\dot{\psi}_e = \frac{\dot{\tau}_e}{\sigma_n} \coth\left(\frac{\tau_e}{a\sigma_n}\right) - \frac{a\ddot{u}_e}{\dot{u}_e} \tag{43b}$$

$$\tag{43c}$$

This allows us to add the source term:

$$S_\psi = \dot{\psi}_e - G(2\dot{u}_e, \psi_e) \tag{44}$$

to (41), so that if our code is working state $\boldsymbol{\psi}$ will evolve with $\dot{\psi}_e$, forcing the solution of (40) to return values of $2\dot{u}_e$.

We test our code on complex domains using a transfinite interpolation. We also remove the slow slip condition from face 1, make $a = .01$, and run our simulation for .1 seconds. More on the coordinate transformation can be found in appendix A. We compute the error between $\boldsymbol{u}$ and u_e, using the L^2 norm:

$$||\Delta \boldsymbol{u}||_{\boldsymbol{H}} = \sqrt{(\Delta \boldsymbol{u})^T \boldsymbol{J}(\boldsymbol{H} \otimes \boldsymbol{H})\Delta \boldsymbol{u}} \tag{45}$$

Where $\Delta \boldsymbol{u} = \boldsymbol{u} - u_e$ and u_e is evaluated at the grid points. Errors and convergence rates for second, fourth and sixth order operators are shown in Table 3.

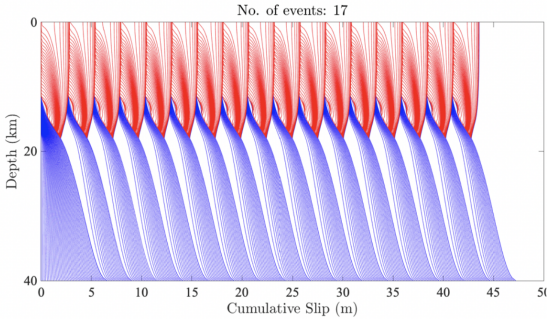
7.3 Switching Between Solvers

The two-solver method poses two problems when switching from one to the other: (1) how can the initial conditions for one solver be calculated from the final conditions of the other? (2) What is a good criteria for when to switch?

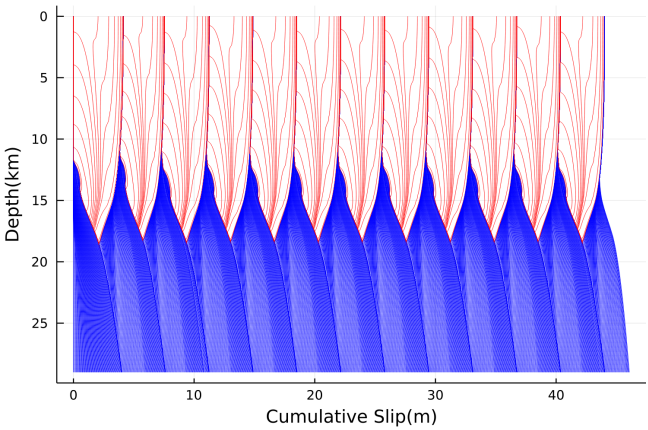
The dynamic solver needs initial conditions on particle displacement and velocity, as well as state. The static solver needs initial conditions on slip and state. Therefore, when switching from static to dynamic, a standard backwards difference in time is used to compute particle velocity, and the other data is already available. When switching from dynamic to static, slip on the fault can be computed as twice the particle velocity, and the other data is already available.

We tried a few switching criteria, and found the simulation was not very sensitive to different schemes. In the end, a threshold for switching of $\max(V) = .01$ m/s was used. Since it is desirable to not switch back and forth many times in the same cycle we also require that a period of 50 years must be surpassed in the inter-seismic phase for a switch to occur, and a period of 6 seconds in the co-seismic phase must be surpassed.

8 Results



(a) Quasi-static BEM simulation from Junle Jiang's Modeling group (Erickson et al. (2020)) with a 25m grid spacing



(b) Cumulative slip on fault in one of our simulations where $N = 850$, $l = .05$, $\bar{r} = \bar{s} = .78$, and $L = 160$km. See appendix A for the full domain transformation.

Fig. 3: Comparison of cumulative slip profiles between a (a) quasi-static BEM simulation and (b) our fully dynamic FD simulation. Both simulations run for 1500 years.

We run all of our simulations in the programming language Julia and all the codes are available at https://github.com/totorotoby/BP1-FD. The results of our simulation match BEM simulations of BP1 almost exactly as seen at https://strike.scec.org/cvws/cgi-bin/seas.cgi. While this is not the most exciting result, it is a good verification exercise before tackling more complicated problems that BEM simulators cannot consider.

In figure 3, we plot cumulative slip on the fault. Blue contours are plotted for every year, while red contours (during rupture) are plotted every second. Since

slip velocities in quasi-static simulations are unrealistically low, our simulation only has 12 events, while the BEM simulation has 17.

Ruptures nucleate at around a depth of 12km at approximately the transition between the velocity weakening and strengthening zones. There are a few critical physical length scales within this zone that must be resolved numerically. In Rice et al. (2001) they derive a length scale h^* that must be resolved for rupture to occur, and other even smaller spatial scales that need to be resolved are in Day (1982). h^* in BP1 is approximately 2km and it is recommended in Erickson et al. (2020) that h^* is resolved with 80 nodes, for a grid spacing of 25m. With our domain transformation we run our simulation with 850 nodes in each direction, and grid spacing of 50m in the critical nucleation zone, and see almost the exact same results as dynamic BEM models. We match results with about a quarter the number of total recommended grid points.

9 Conclusion

We have developed a two-solver method for earthquake cycle simulations in complex geometries. Both solvers use the SBP-SAT framework, and are provably stable. The method, during co-seismic periods, produces a non-stiff set of equations which can be integrated using an explicit time-stepper. The method during the inter-seismic period uses an adaptive time-stepping algorithm, which allows for large time steps. In conjunction these methods produce a computationally efficient scheme.

We verify the accuracy and stability of the new co-seismic solver with the method of manufactured solutions, and clarify the strategy for switching between solvers at the beginning and end of earthquakes.

This new method is applied to a anti-plane strike-slip problem. There is quantitative agreement between the new method, and previously established boundary element methods, even when we simulate on a coarser mesh than we expected to be necessary.

The new method is now being applied to simulate rupture patterns on a fault embedded in a generic sedimentary basin. We hope that this method will help understand when surface ruptures occur within sedimentary basins.

A Coordinate Transformation Equations

The entries of C in (21) are (Almquist and Dunham (2020)):

$$C_{ij}(r,s) = J\left(\frac{\partial i}{\partial x}\mu\frac{\partial j}{\partial x} + \frac{\partial i}{\partial y}\mu\frac{\partial j}{\partial y}\right) \tag{46}$$

The Coordinate Transformation used in the simulation pictured in Figure 3b is:

$$x(r,s) = \xi\tanh((r-1)/l) + \frac{\xi\tanh((-2)/l)+L}{2}(r-1) + L \tag{47a}$$

$$y(r,s) = \xi\tanh((s-1)/l) + \frac{\xi\tanh((-2)/l)+L}{2}(s-1) + L \tag{47b}$$

where:

$$\xi = \frac{L/2 - L\bar{r} - L}{2\tanh((\bar{r}-1)/l + \tanh(-2(\bar{r}-1)/l)} \tag{48}$$

The coordinate transformation for convergence tests in table 3 uses a transfinite interpolation to generate the necessary terms in (46)(Farrashkhalvat and Miles (2003)). The physical domain is defined by a rhombus with points $(x_1, y_1) = (0,0), (x_2, y_2) = (2,1), (x_3, y_3) = (1,2), (x_4, y_4) = (3,3)$.

References

Almquist, M., Dunham, E.M.: Non-stiff boundary and interface penalties for narrow-stencil finite difference approximations of the laplacian on curvilinear multiblock grids. Journal of Computational Physics **408**, 109,294 (2020). DOI https://doi.org/10.1016/j.jcp.2020.109294

Barbot, S.: Slow-slip, slow earthquakes, period-two cycles, full and partial ruptures, and deterministic chaos in a single asperity fault. Tectonophysics **768**, 228,171 (2019). DOI https://doi.org/10.1016/j.tecto.2019.228171

Best Mckay, M., Erickson, B.A., Kozdon, J.E.: A computational method for earthquake cycles within anisotropic media. Geophysical Journal International **219**(2), 816–833 (2019). DOI 10.1093/gji/ggz320

Day, S.M.: Three-dimensional finite difference simulation of fault dynamics: Rectangular faults with fixed rupture velocity. Bulletin of the Seismological Society of America **72**(3), 705–727 (1982). DOI 10.1785/BSSA0720030705

Duru, K., Allison, K.L., Rivet, M., Dunham, E.M.: Dynamic rupture and earthquake sequence simulations using the wave equation in second-order form. Geophysical Journal International **219**(2), 796–815 (2019). DOI 10.1093/gji/ggz319

Erickson, B.A., Dunham, E.M.: An efficient numerical method for earthquake cycles in heterogeneous media: Alternating subbasin and surface-rupturing events on faults crossing a sedimentary basin. Journal of Geophysical Research: Solid Earth **119**(4), 3290–3316 (2014). DOI https://doi.org/10.1002/2013JB010614

Erickson, B.A., Dunham, E.M., Khosravifar, A.: A finite difference method for off-fault plasticity throughout the earthquake cycle. Journal of the Mechanics and Physics of Solids **109**, 50–77 (2017). DOI https://doi.org/10.1016/j.jmps.2017.08.002

Erickson, B.A., Jiang, J., Barall, M., Lapusta, N., Dunham, E.M., Harris, R., Abrahams, L.S., Allison, K.L., Ampuero, J., Barbot, S., Cattania, C., Elbanna, A., Fialko, Y., Idini, B., Kozdon, J.E., Lambert, V., Liu, Y., Luo, Y., Ma, X., Best McKay, M., Segall, P., Shi, P., van den Ende, M., Wei, M.: The Community Code Verification Exercise for Simulating Sequences of Earthquakes and Aseismic Slip (SEAS). Seismological Research Letters **91**(2A), 874–890 (2020). DOI 10.1785/0220190248

Farrashkhalvat, M., Miles, J.: 4 - structured grid generation – algebraic methods. In: Farrashkhalvat, M., Miles, J. (eds.) Basic Structured Grid Generation, pp. 76–115. Butterworth-Heinemann, Oxford (2003). DOI https://doi.org/10.1016/B978-075065058-8/50004-6

Kozdon, J.E., Dunham, E.M., Nordström, J.: Interaction of waves with frictional interfaces using summation-by-parts difference operators: Weak enforcement of nonlinear boundary conditions. Journal of Scientific Computing **50**(2), 341–367 (2012). DOI 10.1007/s10915-011-9485-3

Kozdon, J.E., Erickson, B.A., Harvey, T.: A non-stiff summation-by-parts finite difference method for the wave equation in second order form: Characteristic boundary conditions and nonlinear interfaces (2021a)

Kozdon, J.E., Erickson, B.A., Wilcox, L.C.: Hybridized summation-by-parts finite difference methods. Journal of Scientific Computing **87**(3) (2021b). DOI 10.1007/s10915-021-01448-5

Lapusta, N., Rice, J.R.: Nucleation and early seismic propagation of small and large events in a crustal earthquake model. Journal of Geophysical Research: Solid Earth **108**(B4) (2003). DOI https://doi.org/10.1029/2001JB000793

Lapusta, N., Rice, J.R., Ben-Zion, Y., Zheng, G.: Elastodynamic analysis for slow tectonic loading with spontaneous rupture episodes on faults with rate- and state-dependent friction. Journal of Geophysical Research: Solid Earth **105**(B10), 23,765–23,789 (2000). DOI https://doi.org/10.1029/2000JB900250

Mattsson, K., Ham, F., Iaccarino, G.: Stable boundary treatment for the wave equation on second-order form. Journal of Scientific Computing **41**(3), 366–383 (2009). DOI 10.1007/s10915-009-9305-1

Mattsson, K., Nordström, J.: Summation by parts operators for finite difference approximations of second derivatives. Journal of Computational Physics - J COMPUT PHYS **199**, 503–540 (2004). DOI 10.1016/j.jcp.2004.03.001

Mattsson, K., Parisi, F.: Stable and accurate second-order formulation of the shifted wave equation. Communications in Computational Physics **7**(1), 103 (2010). DOI 10.4208/cicp.2009.08.135

Rice, J.R., Lapusta, N., Ranjith, K.: Rate and state dependent friction and the stability of sliding between elastically deformable solids. Journal of the Mechanics and Physics of Solids **49**(9), 1865–1898 (2001). DOI https://doi.org/10.1016/S0022-5096(01)00042-4. The JW Hutchinson and JR Rice 60th Anniversary Issue

Richards-Dinger, K., Dieterich, J.: Rsqsim earthquake simulator. Seismological Research Letters **83**, 983–990 (2012). DOI 10.1785/0220120105

Roache, P.: Verification and validation in computational science and engineering. 1 edn. Hermosa Publishers, Albuquerque, NM (1998)

Segall, P.: Earthquake and volcano deformation. Earthquake and Volcano Deformation **xxiii** (2010). DOI 10.1515/9781400833856

Segall, P., Bradley, A.M.: Slow-slip evolves into megathrust earthquakes in 2d numerical simulations. Geophysical Research Letters **39**(18) (2012). DOI https://doi.org/10.1029/2012GL052811

Thomas, M.Y., Lapusta, N., Noda, H., Avouac, J.P.: 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). DOI https://doi.org/10.1002/2013JB010615