Crustal Deformation Modeling Tutorial

Spontaneous Rupture with PyLith

Brad Aagaard
Matthew Knepley
Charles Williams

COMPUTATIONAL INFRASTRUCTURE for GEODYNAMICS

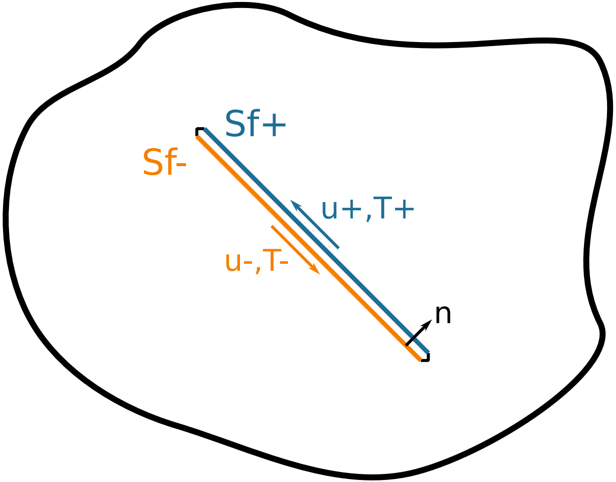
June 27, 2017

Concepts Covered in this Session

- Quasistatic simulations with spontaneous fault rupture driven by aseismic creep
- Fault constitutive models
  - Slip-weakening
  - Dieterich-Ruina rate-state friction w/ageing law
- Nonlinear solver parameters
- Initial fault traction perturbations

Fault Interface

Fault tractions couple deformation across interface



Governing Equations

Terms in governing equation associated with fault

- Tractions on fault surface are analogous to boundary tractions

$$\ldots + \underbrace{\int_{S_T} \vec{\phi} \cdot \vec{T}\, dS}_{\text{Neumann BC}} - \underbrace{\int_{S_{f+}} \vec{\phi} \cdot \vec{l}\, dS}_{\text{Fault +}} + \underbrace{\int_{S_{f-}} \vec{\phi} \cdot \vec{l}\, dS}_{\text{Fault -}} \ldots = 0$$

- Relationship between slip and relative displacement

$$\int_{S_f} \vec{\phi} \cdot (\underbrace{\vec{d}}_{\text{Slip}} - \underbrace{(\vec{u}_+ - \vec{u}_-)}_{\text{Relative Disp.}})\, dS = 0$$

Fault Constitutive Model

Fault constitutive model places constraints on Lagrange multipliers

- Shear components of Lagrange multipliers limited by fault constitutive model

$$l_{shear} \leq T_{friction} \tag{1}$$

- Fault friction depends on cohesion, coefficient of friction, and normal traction

$$T_{friction} = \begin{cases} T_{cohesion} - \mu_f T_{normal} & T_{normal} \leq 0 \\ T_{cohesion} & T_{normal} > 0 \end{cases} \tag{2}$$

- Compression $\Rightarrow$ no interpenetration, opening $\Rightarrow$ free surface

$$T_{normal} u_{normal} = 0 \tag{3}$$

Solution Algorithm

Solution requires "friction sensitivity" solve in addition to nonlinear solve

1. Perform nonlinear iteration assuming no additional slip
2. Check to see if fault constitutive model is satisfied
3. If not satisfied, estimate slip required to reduce traction
    1. Extract subset of system associated with the fault

$$\begin{pmatrix} \boldsymbol{K}_{n^+n^+} & 0 & \boldsymbol{L}_p^T \\ 0 & \boldsymbol{K}_{n^-n^-} & -\boldsymbol{L}_p^T \\ \boldsymbol{L}_p & -\boldsymbol{L}_p & 0 \end{pmatrix} \begin{pmatrix} \vec{u}_{n^+} \\ \vec{u}_{n^-} \\ \vec{l}_p \end{pmatrix} = \begin{pmatrix} \vec{b}_{n^+} \\ \vec{b}_{n^-} \\ \vec{b}_p \end{pmatrix} \quad (4)$$

    2. Perturb Lagrange multipliers to satisfy friction criterion
    3. Inner solve to get slip producing Lagrange multiplier perturbation

$$\boldsymbol{K}_{n^+n^+} \cdot \partial\vec{u}_{n^+} = -\boldsymbol{L}_p^T \cdot \partial\vec{l}_p, \quad (5)$$

$$\boldsymbol{K}_{n^-n^-} \cdot \partial\vec{u}_{n^-} = \boldsymbol{L}_p^T \cdot \partial\vec{l}_p, \quad (6)$$

$$\partial\vec{d}_p = \partial\vec{u}_{n^+} - \partial\vec{u}_{n^-}. \quad (7)$$

4. Repeat

Coming in PyLith v3.x

New fault friction formulation

- Change meaning of Lagrange multiplier for fault friction
- Recompute Jacobian when switching from locked to sliding
- No “friction sensitivity” solve required
- **Much faster convergence in nonlinear solve**

Friction and Nonlinear Solver Parameters

Solver tolerances are very important

- Dynamic (spontaneous rupture) fault parameters

  zero_tolerance Iterative solver is not exact, so need threshold to detect nucleation of slip.

  zero_tolerance_normal Suppress fault opening for near zero values of slip.

- Linear solver must converge to tighter tolerance than fault **zero_tolerance** for fault to “lock”

  ksp_rtol Set to very small value to force absolute convergence

  ksp_atol Must be smaller than fault **zero_tolerance**

- Nonlinear solver tolerance should not be smaller than fault **zero_tolerance**

  snes_rtol Set to very small value to force absolute convergence

  snes_atol Must be larger than fault **zero_tolerance**

Friction and Nonlinear Solver Parameters

Parameters from a typical example (see examples)

```
[pylithapp.problem.interfaces.fault]
zero_tolerance = 1.0e-9
zero_tolerance_normal = 1.0e-9

[pylithapp.petsc]
# Linear solver tolerances
ksp_rtol = 1.0e-20
ksp_atol>/p> = 1.0e-10

# Nonlinear solver tolerances
<p>snes_rtol = 1.0e-20
snes_atol = 1.0e-8

# Set preconditioner for friction sensitivity solve
friction_pc_type = asm
friction_sub_pc_factor_shift_type = nonzero
```

Fault Constitutive Models

PyLith contains some of the more popular fault constitutive models

Static	Constant coefficient of friction
Slip-Weakening	Friction decreases with slip to a lower limit
Time-Weakening	Time replaces slip in slip-weakening friction model
Rate-State	Dieterich-Ruina rate-state friction with ageing law

Some additional, less popular, fault-constitutive models with combinations of slip-weakening and time-weakening are available for use in the SCEC Dynamic Rupture benchmarks.

Static Friction

Fault has constant coefficient of friction

- Coefficient of friction

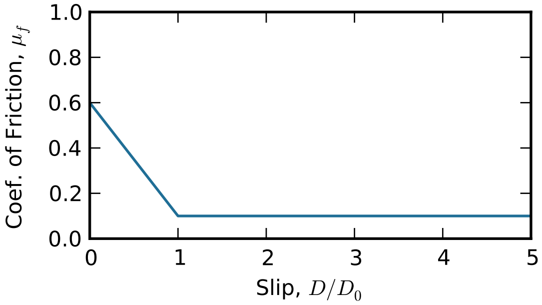
$$\mu_f = \mu_{static} \tag{8}$$

- Slip continues once threshold shear traction is reached
- No stick-slip behavior
- Generally only used in static simulations

Slip-Weakening Friction

Fault weakens with slip until it reaches a lower limit

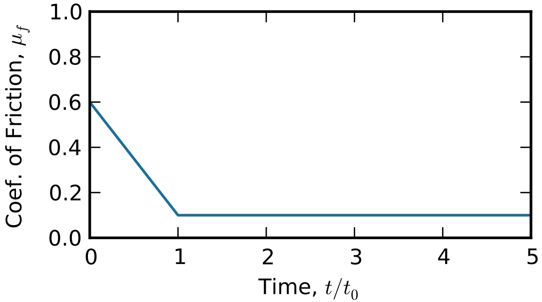
$$\mu_f = \begin{cases} \mu_{dynamic} + (1 - \frac{D}{D_0})(\mu_{static} - \mu_{dynamic}) & D \leq D_0 \\ \mu_{dynamic} & D > D_0 \end{cases} \quad (9)$$



Time-Weakening Friction

Fault weakens with time until it reaches a lower limit

$$\mu_f = \begin{cases} \mu_{dynamic} + (1 - \frac{t}{t_0})(\mu_{static} - \mu_{dynamic}) & t \leq t_0 \\ \mu_{dynamic} & t > t_0 \end{cases} \quad (10)$$

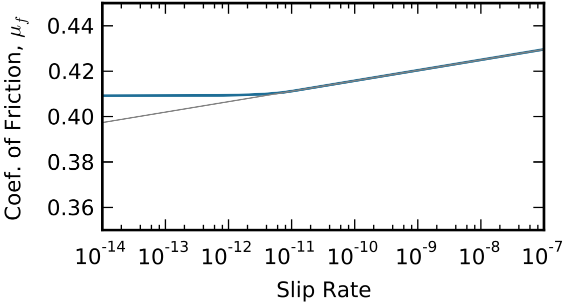


Rate-State Friction with Ageing Law

Dieterich-Ruina rate-state friction with ageing evolution law

$$\mu_f = \begin{cases} \mu_0 + a\ln(\frac{V}{V_0}) + b\ln(\frac{V_0\theta}{L}) & V \geq V_{linear} \\ \mu_0 + a\ln(\frac{V_{linear}}{V_0}) + b\ln(\frac{V_0\theta}{L}) - a(1 - \frac{V}{V_{linear}}) & V < V_{linear} \end{cases} \quad (11)$$

$$\frac{d\theta}{dt} = 1 - \frac{V\theta}{L} \quad (12)$$



Spontaneous Rupture Parameters

Overview of principal components

FaultCohesiveDyn	Fault object for spontaneous rupture
FrictionModel	Fault constitutive model
TractPerturbation	Prescribed spatial and/or temporal variation in fault tractions
SolverNonlinear	Quasi-static simulations with spontaneous rupture require nonlinear solver

Spontaneous Rupture Parameters

Example of fault parameters in a .cfg file

```
[pylithapp.timedependent.interfaces]
fault = pylith.faults.FaultCohesiveDyn

[pylithapp.timedependent.interfaces.fault]
friction = pylith.friction.StaticFriction
friction.label = Static friction

friction.db_properties = spatialdata.spatialdb.UniformDB
friction.db_properties.label = Static friction
friction.db_properties.values = [friction-coefficient,cohesion]
friction.db_properties.data = [0.6,0.0*Pa]

traction_perturbation = pylith.faults.TractPerturbation
traction_perturbation.db_initial = spatialdata.spatialdb.SimpleDB
traction_perturbation.db_initial.label = Initial fault tractions
traction_perturbation.db_initial.iohandler.filename = spatialdb/tractions.spatialdb
```

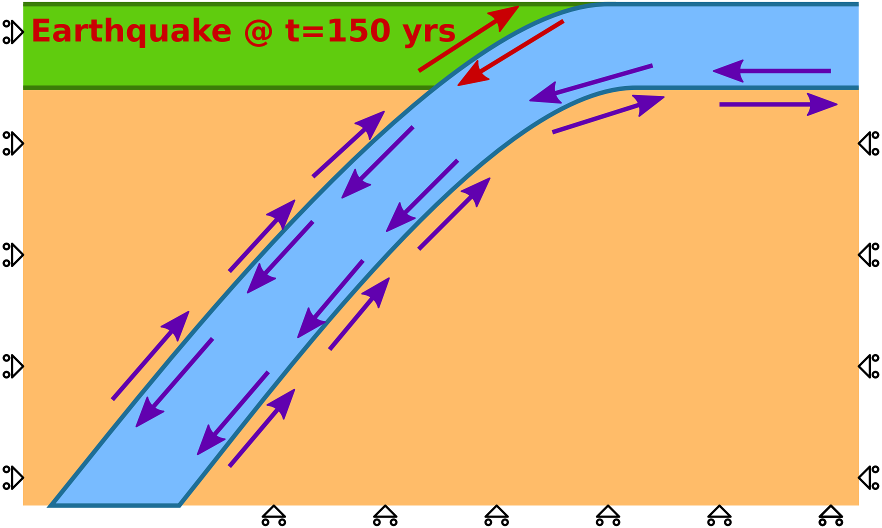
Quasi-static Spontaneous Ruptures

Step 5 in `examples/3d/subduction` does not work yet, and will likely take many minutes to run when it does work.

New examples in `examples/2d/subduction`. Earthquake cycle with spontaneous rupture driven by subducting slab.

- `Step05` Slip-weakening friction
- `Step06` Rate-state friction

2-D Subduction Zone Description



Step 5: Tour of Input Files

`pylithapp.cfg` Parameters (mostly) common to Steps 1–6

`step05.cfg` Parameters specific to Step 5

`fault_slabtop_slipweakening.spatialdb` Friction properties spatial database

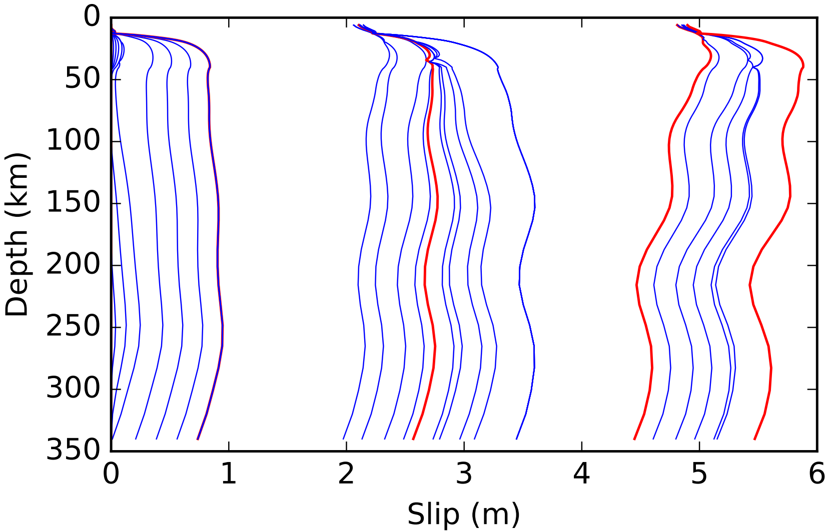
`fault_slabtop_tractions.spatialdb` Fault tractions spatial database

Run the simulation:

```
pylith step05.cfg >& step05.log &
tail -f step05.log
```

Step 5: Slip Profiles Versus Time

Earthquake rupture causes slip over entire fault



Step 6: Tour of Input Files

`step06.cfg` Parameters specific to Step 5

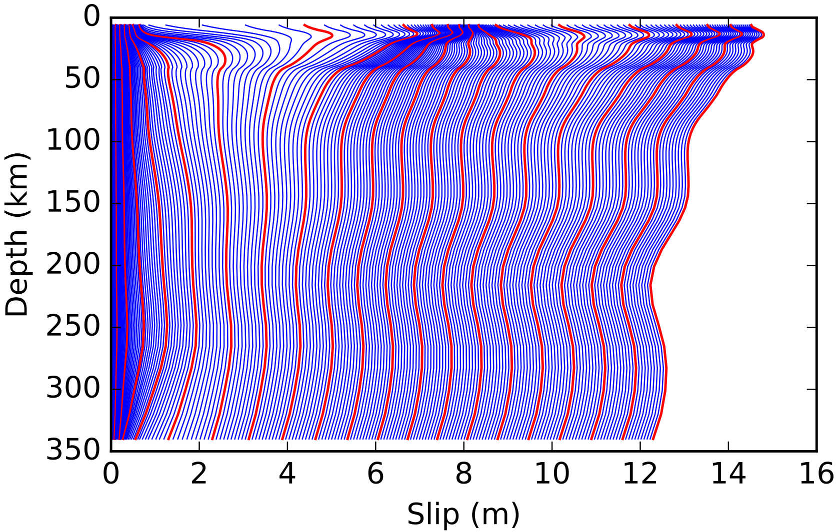
`fault_slabtop_ratestate.spatialdb` Friction properties spatial database

Run the simulation:

```
pylith step06.cfg >& step05.log &
tail -f step06.log
```

Step 6: Slip Profiles Versus Time

Earthquake rupture causes slip over entire fault



Spontaneous Rupture Tips

Fault friction is inherently highly nonlinear

- Spontaneous rupture often localizes stresses, requiring very high resolution meshes around fault.
- Friction parameters from the laboratory are usually not numerically tractable.
- You often need to regularize the friction model to obtain numerically stable solutions.
  - Increase slip/time over which friction coefficient evolves.
  - Reduce difference between “yield” stress and sliding stress.
  - Reduce time step and discretization size.