Crustal Deformation Modeling Tutorial

Spontaneous Rupture via Fault Friction

Brad Aagaard
Matthew Knepley
Charles Williams

COMPUTATIONAL INFRASTRUCTURE for GEODYNAMICS

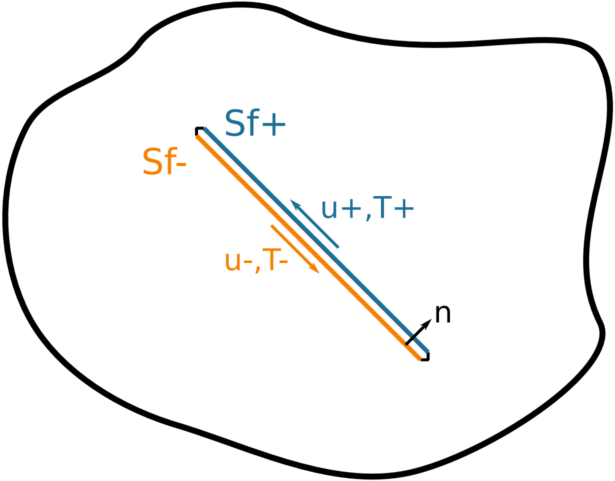
June 24, 2014

Concepts Covered in this Session

- PyLith simulations with spontaneous fault rupture
  - Quasi-static simulations
  - Dynamic simulations
- Fault constitutive models
  - Static friction
  - Slip-weakening
  - Dieterich-Ruina rate-state friction w/ageing law
- Nonlinear solver parameters
- Absorbing boundaries in dynamic simulations
- Time-dependent Dirichlet BC
- Initial and time-dependent 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 interpenetation, 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

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

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

$$\overline{K}_{n^+n^+} \cdot \partial\vec{u}_{n^+} = -\overline{L}_p^T \cdot \partial\vec{l}_p, \tag{5}$$

$$\overline{K}_{n^-n^-} \cdot \partial\vec{u}_{n^-} = \overline{L}_p^T \cdot \partial\vec{l}_p, \tag{6}$$

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

4. Repeat

PyLith v2.1

New fault friction formulation for next version of PyLith

- 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 has a `zero_tolerance` parameter
- 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-11

[pylithapp.petsc]
# Linear solver tolerances
ksp_rtol = 1.0e-20
ksp_atol = 1.0e-12

# Nonlinear solver tolerances
snes_rtol = 1.0e-20
snes_atol = 1.0e-10

# 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

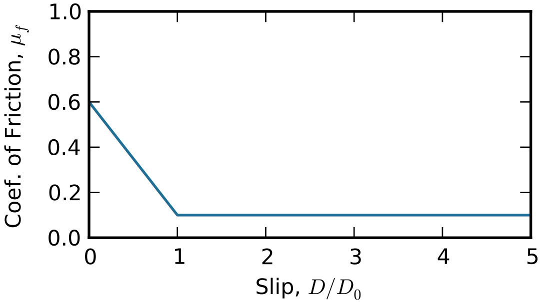
$$\mu_f = \mu_{static} \tag{8}$$

- 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

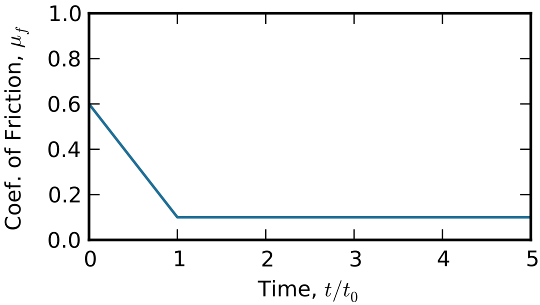
$$\mu_f = \begin{cases} \mu_{dynamic} + (1 - \frac{D}{D_0})(\mu_{static} - \mu_{dynamic}) & D \leq D_0 \\ \mu_{dynamic} & D > D_0 \end{cases} \quad (9)$$



Time-Weakening Friction

Fault weakens with time until it reaches a lower limit

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

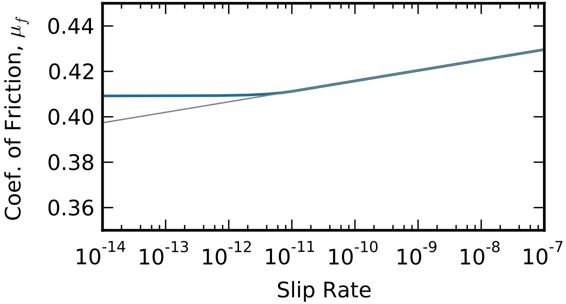


Rate-State Friction with Ageing Law

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
[pylithapp.timedependent.interfaces.fault.traction_perturbation]
db_initial = spatialdata.spatialdb.SimpleDB
db_initial.label = Initial fault tractions
db_initial.iohandler.filename = spatialdb/tractions.spatialdb
```

Static and Quasi-static Spontaneous Ruptures

Fault slips in response to loading from boundaries

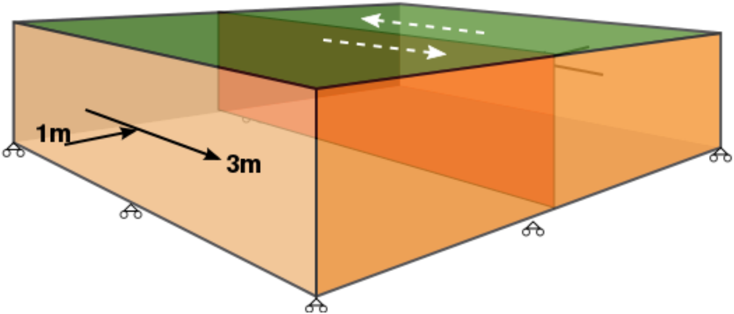
Files are in `examples/3d/hex8`

- `Step10` Static simulation, static friction w/o slip
- `Step11` Static simulation, static friction w/slip
- `Step12` Quasi-static simulation, static friction w/slip
- `Step13` Quasi-static simulation, slip-weakening w/stick-slip
- `Step14` Quasi-static simulation, rate-state w/stick-slip

```
pylith step10.cfg
```

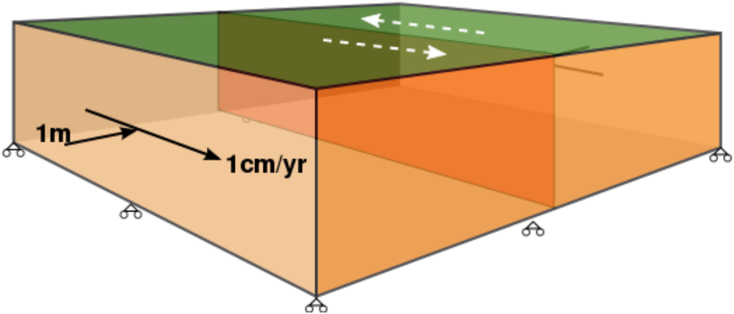
Step11

Static simulation, static friction w/slip



Step13

Quasi-static simulation, slip-weakening w/slip-slip



Dynamic Spontaneous Rupture

Fault slips in reponse to prescribed tractions

Files are in `examples/bar_shearwave/quad4`

`spontaneousrup_staticfriction` Static friction

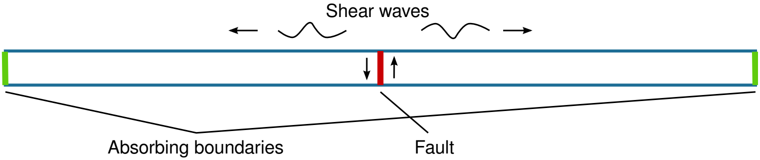
`spontaneousrup_slipweakening` Slip-weakening

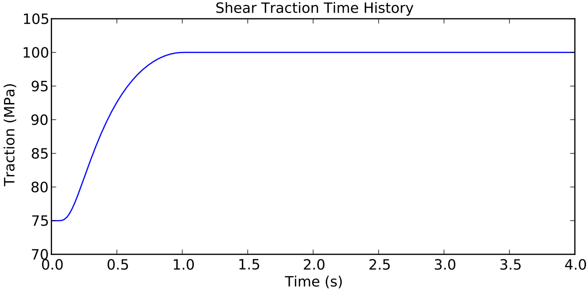
`spontaneousrup_ratestateageing` Rate-state w/ageing law

```
pylith spontaneousrup.cfg
spontaneousrup_staticfriction.cfg
```

Prescribed Traction Loads Fault

Dynamic simulation w/initial & temporal traction perturbation





COMPUTATIONAL INFRASTRUCTURE for GEODYNAMICS