

# Crustal Deformation Modeling Tutorial

PyLith: Introduction to Fault Friction

Brad Aagaard  
Matthew Knepley  
Charles Williams



June 18, 2012

# Governing Equations

Terms in governing equation associated with fault

- Traction on fault surface are analogous to boundary tractions

$$\dots + \int_{S_T} \vec{\phi} \cdot \vec{T} dS - \int_{S_{f+}} \vec{\phi} \cdot \vec{T} dS + \int_{S_{f-}} \vec{\phi} \cdot \vec{T} dS \dots = 0 \quad (1)$$

- Relationship between slip and relative displacement

$$\int_{S_f} \vec{\phi} \cdot (\vec{d} - (\vec{u}_+ - \vec{u}_-)) dS = 0 \quad (2)$$

# Governing Equations (cont.)

Express weighting function  $\vec{\phi}$ , displacement field  $\vec{u}$ , Lagrange multipliers (fault tractions)  $\vec{l}$ , and fault slip  $\vec{d}$  as linear combinations of basis functions,

$$\vec{\phi} = \bar{N}_m \cdot \vec{a}_m \quad (3)$$

$$\vec{u} = \bar{N}_n \cdot \vec{u}_n \quad (4)$$

$$\vec{l} = \bar{N}_p \cdot \vec{l}_p \quad (5)$$

$$\vec{d} = \bar{N}_p \cdot \vec{d}_p \quad (6)$$

- Lagrange multiplier (fault traction) terms:

$$\dots - \int_{S_{f+}} \bar{\mathbf{N}}_m^T \cdot \bar{\mathbf{N}}_p \cdot \vec{l}_p dS + \int_{S_{f-}} \bar{\mathbf{N}}_m^T \cdot \bar{\mathbf{N}}_p \cdot \vec{l}_p dS = \vec{0} \quad (7)$$

- Constraint equation

$$\int_{S_f} \bar{\mathbf{N}}_p^T \cdot \left( \bar{\mathbf{N}}_p \cdot \vec{d}_p - \bar{\mathbf{N}}_{n+} \cdot \vec{u}_{n+} + \bar{\mathbf{N}}_{n-} \cdot \vec{u}_{n-} \right) dS = \vec{0} \quad (8)$$

# Fault Constitutive Model

Fault constitutive model places constraints on Lagrange multipliers

- Shear components of Lagrange multipliers limited by fault constitutive model

$$I_{shear} \leq T_{friction} \quad (9)$$

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

$$T_{friction} = \begin{cases} T_{cohesion} - \mu_f T_{normal} & T_{normal} \leq 0 \\ T_{cohesion} & T_{normal} > 0 \end{cases} \quad (10)$$

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

$$T_{normal} u_{normal} = 0 \quad (11)$$

# 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

$$\begin{pmatrix} \bar{K}_{n^+n^+} & 0 & \bar{L}_p^T \\ 0 & \bar{K}_{n^-n^-} & -\bar{L}_p^T \\ \bar{L}_p & -\bar{L}_p & 0 \end{pmatrix} \begin{pmatrix} \vec{u}_{n^+} \\ \vec{u}_{n^-} \\ \vec{l}_p \end{pmatrix} = \begin{pmatrix} \vec{b}_{n^+} \\ \vec{b}_{n^-} \\ \vec{b}_p \end{pmatrix} \quad (12)$$

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

$$\bar{K}_{n^+n^+} \cdot \partial \vec{u}_{n^+} = -\bar{L}_p^T \cdot \partial \vec{l}_p, \quad (13)$$

$$\bar{K}_{n^-n^-} \cdot \partial \vec{u}_{n^-} = \bar{L}_p^T \cdot \partial \vec{l}_p, \quad (14)$$

$$\partial \vec{d}_p = \partial \vec{u}_{n^+} - \partial \vec{u}_{n^-}. \quad (15)$$

- 4 Repeat

# Fault Constitutive Models

PyLith contains some of the most 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

# Static Friction

Fault has constant coefficient of friction

- Coefficient of friction

$$\mu_f = \mu_{static} \quad (16)$$

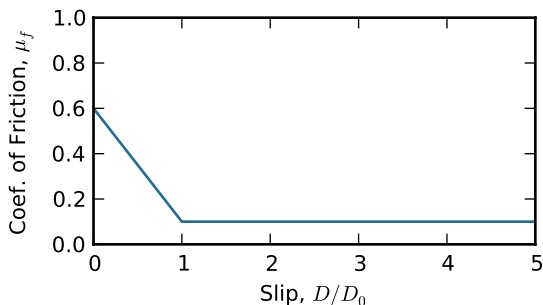
- Slip continues once threshold shear traction is reached
- No stick-slip behavior
- Generally only used in static simulations



# Slip-Weakening Friction

Fault weakens with slip until it reaches a lower limit

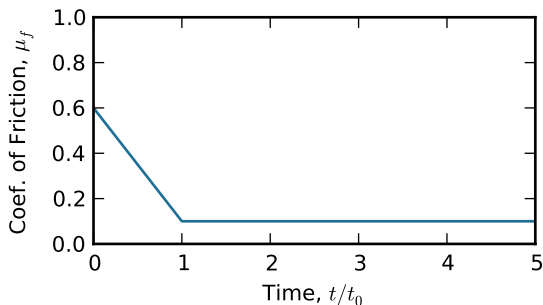
$$\mu_f = \begin{cases} \mu_{dynamic} + (1 - \frac{D}{D_0})(\mu_{static} - \mu_{dynamic}) & D \leq D_0 \\ \mu_{dynamic} & D > D_0 \end{cases} \quad (17)$$



# Time-Weakening Friction

Fault weakens with time until it reaches a lower limit

$$\mu_f = \begin{cases} \mu_{dynamic} + (1 - \frac{t}{t_0})(\mu_{static} - \mu_{dynamic}) & t \leq t_0 \\ \mu_{dynamic} & t > t_0 \end{cases} \quad (18)$$

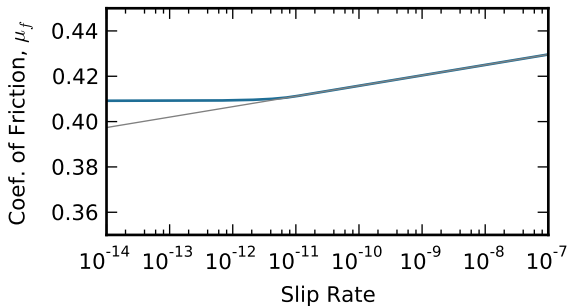


# Rate-State Friction with Ageing Law

Dieterich-Ruina rate-state friction with ageing evolution law

$$\mu_f = \begin{cases} \mu_0 + a \log\left(\frac{V}{V_0}\right) + b \log\left(\frac{V_0 \theta}{L}\right) & V \geq V_{linear} \\ \mu_0 + a \log\left(\frac{V_{linear}}{V_0}\right) + b \log\left(\frac{V_0 \theta}{L}\right) - a\left(1 - \frac{V}{V_{linear}}\right) & V < V_{linear} \end{cases} \quad (19)$$

$$\frac{d\theta}{dt} = 1 - \frac{V\theta}{L} \quad (20)$$



# Spontaneous Rupture Parameters

Overview of principal components

<b>FaultCohesiveDyn</b>	Fault object for spontaneous rupture
<b>FrictionModel</b>	Fault constitutive model
<b>TractPerturbation</b>	Prescribed spatial and/or temporal variation in fault tractions
<b>SolverNonlinear</b>	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
```

# Spontaneous Rupture Parameters

## Overview of principal features

Files are in `src/pylith/examples/3d/hex8`

**Step10** Static simulation, static friction w/o slip

**Step11** Static simulation, static friction w/slip

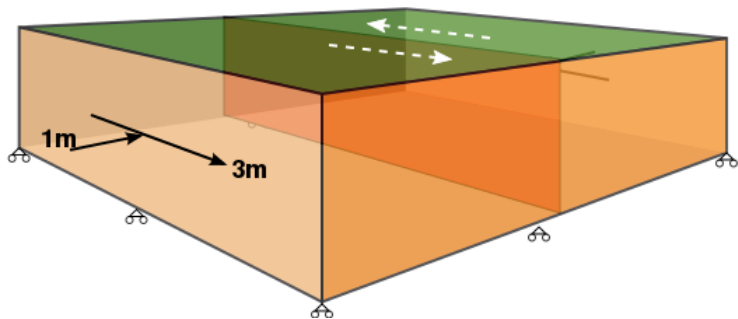
**Step12** Quasi-static simulation, static friction w/slip

**Step13** Quasi-static simulation, slip-weakening w/stick-slip

**Step14** Quasi-static simulation, rate-state w/stick-slip

# Step11

Static simulation, static friction w/slip



# Step13

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

