rsfmodel - A Frictional Modeling Tool for Fault and Laboratory Data Analysis

J.R. Leeman[1], Ryan May[2], Chris Marone[1], Demian Saffer[1]

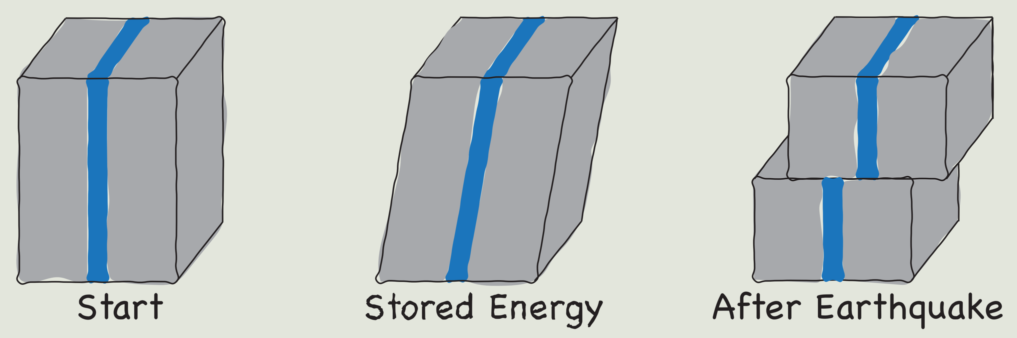
[1]Penn State Department of Geoscience, [2]UCAR/Unidata

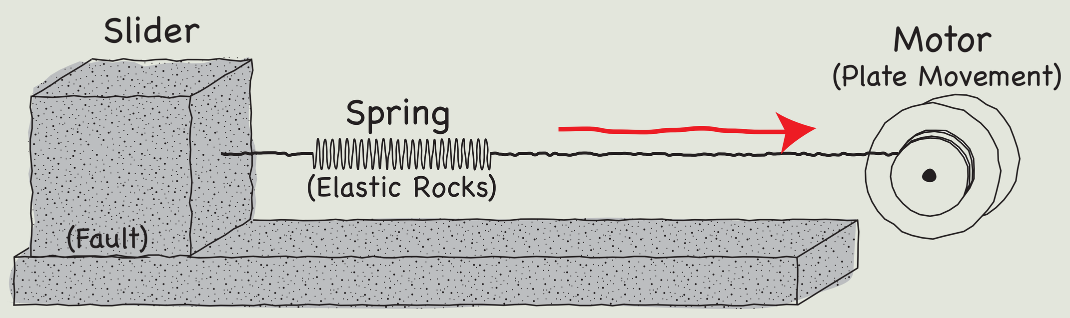
www.rsfmodel.com

@geo_leeman
www.johnrleeman.com

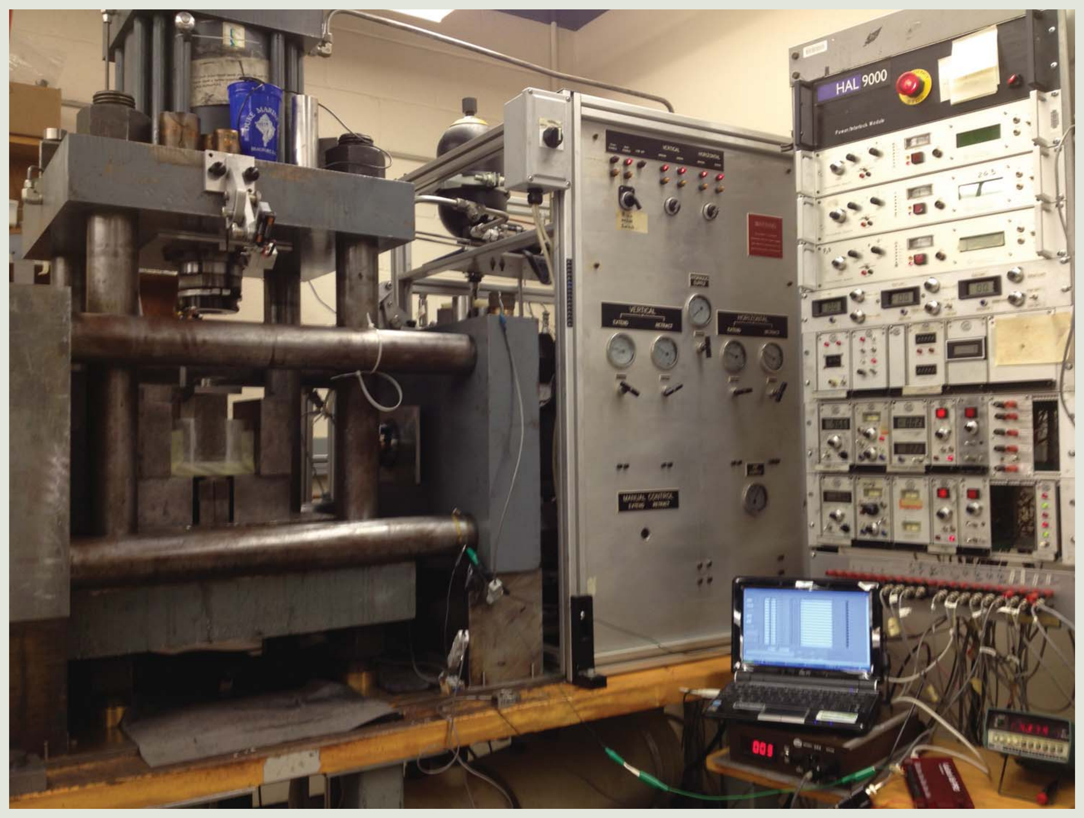
Friction and Earthquakes

Earthquakes are a result of stored elastic energy in the Earth's crust. The crust is strained by the constant motion of Earth's 15-20 tectonic plates at 2.5 -15 cm/yr. This stored strain energy must be released. Energy releases can be continuous sliding (creep) or acute bursts of energy (earthquakes). An important factor in determining how a fault will fail, and if it presents a seismic hazard, is the frictional properties of the wall rock and the ground up fault gouge.





To better understand earthquake processes, we use physical and numerical models of fault zone processes. The simplest physical model is known as the "spring-slider model" (above) in which a spring stores strain energy from a motor and a rock slider represents the fault. We can conduct more controlled experiments in modern, servo-controlled testing apparatus such as the Penn State Biaxial Shear Apparatus (right).

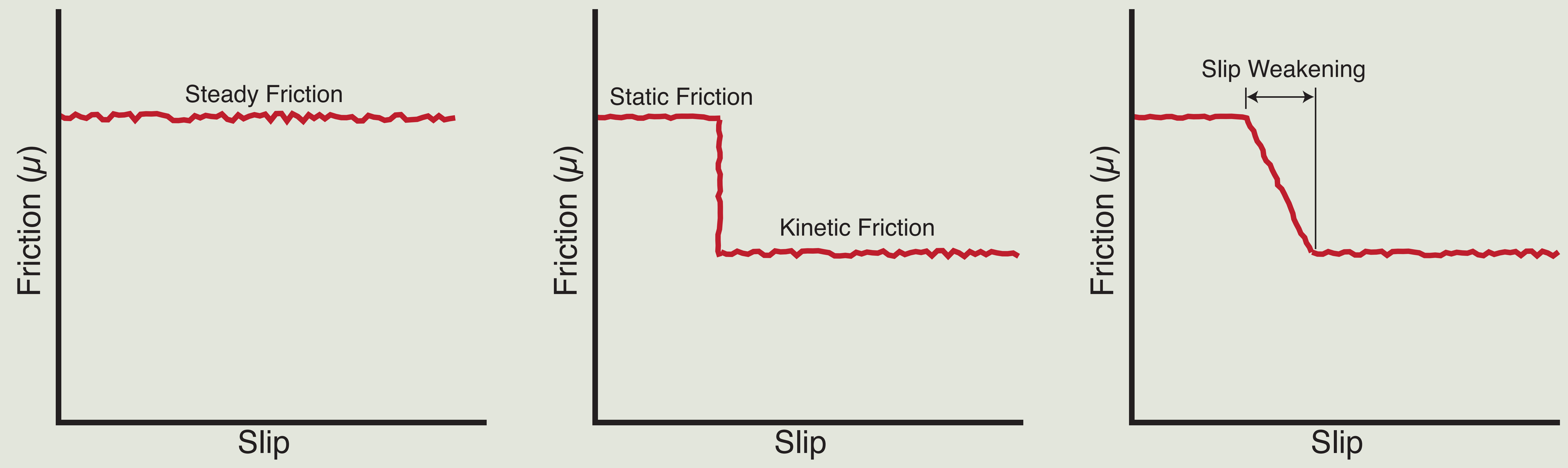


Control and recording sub-systems on the apparatus are run via LabView. Initial data analysis is accomplished with the in-house tool "XLook". After data reduction to engineering units, each scientist begins to work in their favorite processing environment to model and analyze their results. Python has been gaining traction, but suffers from the lack of field-specific tools that are available. Here we try to provide a useful tool to complete one of the most common tasks faced by the friction experimentalist.

How does friction evolve?

Friction evolution can be viewed from any number of levels of complexity. In the broadest sense, friction is a single value describing the ratio of shear force to normal force on an interface. Very early on, it became apparent that friction is not a single, unchanging value. First approximations (still used in some situations) defined static and kinetic coefficients of friction. Friction relations have continued to become more complex as observations of frictional healing, slip weakening, and rate-dependence have been incorporated into the models.

The current frictional model used in rock mechanics is the rate-and-state frictional relation in which friction is an evolving quantity in a system that has the potential to become frictionally unstable.



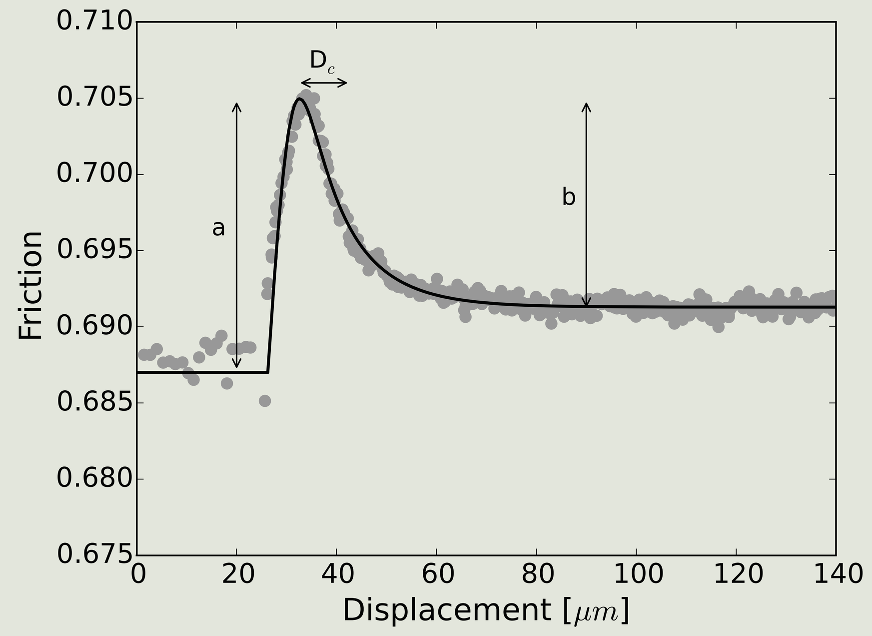
Rate-and-State Friction

The rate-and-state relation describes friction in terms of a direct effect in response to a perturbation, and an evolution effect that incorporates 'memory' responses of friction. The state variable(s) evolution can be described by one or more of several proposed relationships.

$$\mu = \mu_0 + a \ln\left(\frac{V_{\text{slider}}}{V_0}\right) + \sum_{i=1}^{n} b_i \ln\left(\frac{V_0 \theta_i}{D_{ci}}\right)$$

$$\frac{d\mu}{dt} = k\left(V_{\text{slider}} - V_0 \exp\left[\frac{\mu - \mu_0 - \sum_{i=1}^{n} b_i \ln(\frac{V_0\theta_i}{D_{ci}})}{a}\right]\right)$$

Coefficients *a* and *b* describe how friction evolves: *a* characterizes the initial response due to the rate change, *b* scales the state dependent evolution (right figure). For a material with *(a-b) < 0* the friction decreases as velocity increases, called rate-weakening. Materials with *(a-b) > 0* are rate-strengthening, and when *(a-b) = 0* the material is said to be velocity neutral. Samples can slide in a stable manner, exhibit stick-slip behavior, or reside in the continuum known as slow slip between the two. Stable sliding is analogous to the deep movement of a subducting plate below the seismogenic zone, while stick-slip failure is a common model for earthquake recurrence.



State Evolution Relations

Slowness (Dieterich) $$\frac{d\theta}{dt} = 1 - \frac{V_{\text{slider}}\theta}{D_c}$$

Slip (Ruina) $$\frac{d\theta}{dt} = -\frac{V_{\text{slider}}\theta}{D_c}\ln\left(\frac{V_{\text{slider}}\theta}{D_c}\right)$$

PRZ $$\frac{d\theta}{dt} = 1 - \left(\frac{V_{\text{slider}}\theta}{2D_c}\right)^2$$

Nagata $$\frac{d\theta}{dt} = 1 - \frac{V_{\text{slider}}\theta}{D_c} - \frac{c}{b}\theta\frac{d\mu}{dt}$$

rsfmodel Toolkit

The rsfmodel toolkit provides a forward modeling environment to enable modelers and experimentalists to predict the frictional response of materials. Previously this has been done with widespread non-standardized programs with little or no regular regression testing and inter-program results comparison. This alone has likely resulted in misfit data and interpretations.

Modeling of experimental data, as well as numerical hypothesis testing are essential to the community and can now be completed on a common platform of experimentation. The toolkit also includes a variety of commonly produced plots and can produce simple ASCII output to feed into any other analysis tools.

We are not aware of any tool that currently allows an arbitrary number of state variables, mixing of state relations, or the definition of unique state variable relations.

Capabilities

- Forward modeling of friction with an arbitrary velocity profile
- Modeling with *n* state variables
- Mix and match state evolution relations
- Define new state evolution relations
- Produce standard plots

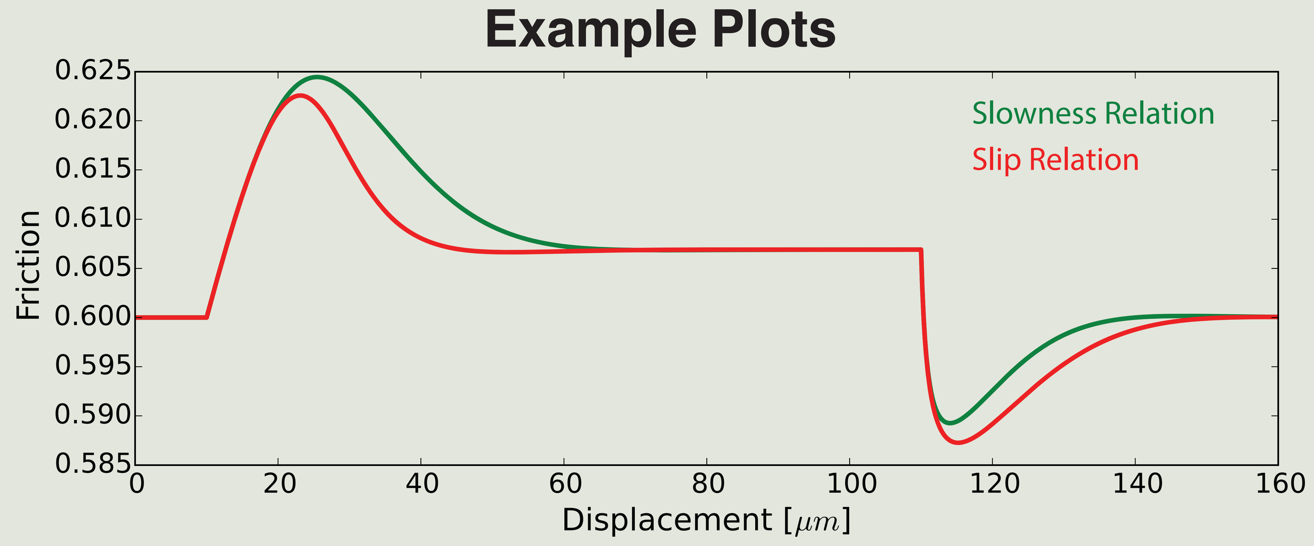
Future Enhancements

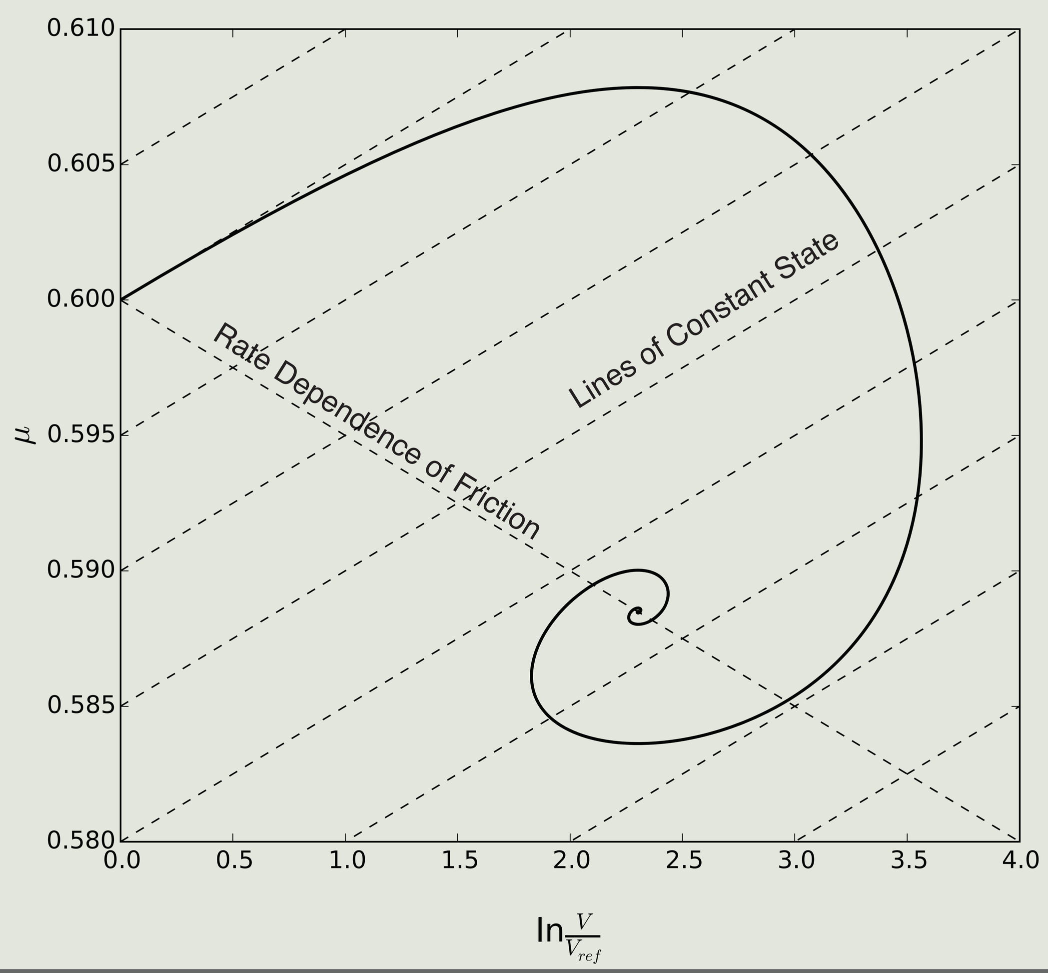
- Data based inversion with singular value decomposition (SVD)
- Data based inversion with evolutionary algorithms (i.e. CMA-ES)
- Model stochastic perturbation

Model Setup and Use

```
import numpy as np
import matplotlib.pyplot as plt
from rsfmodel import rsf

model = rsf.Model()

# Set model initial conditions
model.mu0 = 0.6 # Friction initial (at the reference velocity)
model.a = 0.005 # Empirical coefficient for the direct effect
model.k = 1e-3 # Normalized System stiffness (friction/micron)
model.v = 1. # Initial slider velocity, generally is vlp(t=0)
model.vref = 1. # Reference velocity, generally vlp(t=0)

state1 = rsf.DieterichState(model)
state1.b = 0.01  # Empirical coefficient for the evolution effect
state1.Dc = 10.  # Critical slip distance

model.state_relations = [state1] # Which state relation we want to use

# We want to solve for 40 seconds at 100Hz
model.time = np.arange(0,40.01,0.01)

# We want to slide at 1 um/s for 10 s, then at 10 um/s for 31
lp_velocity = np.ones_like(model.time)
lp_velocity[10*100:] = 10. # Velocity after 10 seconds is 10 um/s

# Set the model load point velocity, must be same shape as model.model_time
model.loadpoint_velocity = lp_velocity

# Run the model!
model.solve()

# Make the phase plot
rsf.phasePlot(model)

# Make a plot in displacement
rsf.dispPlot(model)
```

Example Plots





This material is based upon work supported by the National Science Foundation under Grant No. DGE1255832. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.