Crustal Deformation Modeling

Overview of workflow for typical research problem

Legend

- CIG
- Free
- Open Source
- Commercial
- Available
- Planned

- Gocad
- LaGriT
- PyLith
- ParaView
- Earth Vision
- CUBIT
- GeoFEST
- Mayavi2
- TetGen
- Abaqus
- Visit
- NetGen
- Matlab
- OpenDX
- Iris Explorer
- Fledermaus

Computational Infrastructure for Geodynamics
Ingredients for Running PyLith

- Simulation parameters

- Finite-element mesh
  - Mesh exported from LaGriT
  - Mesh exported from CUBIT
  - Mesh constructed by hand (PyLith mesh ASCII format)

- Spatial databases for physical properties, boundary conditions, and rupture parameters
  - SCEC CVM-H or USGS Bay Area Velocity model
  - Simple ASCII files
Spatial Databases

User-specified field/value in space

- **Examples**
  - Uniform value for Dirichlet (0-D)
  - Piecewise linear variation in tractions for Neumann BC (1-D)
  - SCEC CVM-H seismic velocity model (3-D)

- Generally independent of discretization for problem

- **Available spatial databases**
  - **UniformDB** Optimized for uniform value
  - **SimpleDB** Simple ASCII files (0-D, 1-D, 2-D, or 3-D)
  - **SCECCVMH** SCEC CVM-H seismic velocity model v5.3
  - **ZeroDispDB** Special case of UniformDB
Features in PyLith 1.5

Enhancements and new features in blue

- Time integration schemes and elasticity formulations
  - Implicit for quasi-static problems (neglect inertial terms)
    - Infinitesimal strains
    - Small strains
  - Explicit for dynamic problems
    - Infinitesimal strains with sparse system Jacobian
    - Infinitesimal strains with lumped system Jacobian
    - Small strains with sparse system Jacobian

- Bulk constitutive models
  - Elastic model (1-D, 2-D, and 3-D)
  - Linear and Generalized Maxwell viscoelastic models (3-D)
  - Power-law viscoelastic model (3-D)
  - Linear Maxwell viscoelastic model (2-D)
  - Drucker-Prager elastoplastic model (3-D)
Features in PyLith 1.5 (cont.)

Enhancements and new features in blue

- Boundary and interface conditions
  - Time-dependent Dirichlet boundary conditions
  - Time-dependent Neumann (traction) boundary conditions
  - Absorbing boundary conditions
  - Kinematic (prescribed slip) fault interfaces w/multiple ruptures
  - Dynamic (friction) fault interfaces
  - Time-dependent point forces
  - Gravitational body forces

- Fault constitutive models
  - Static friction
  - Linear slip-weakening
  - Dieterich-Ruina rate and state friction w/ageing law
Features in PyLith 1.5 (cont.)

Enhancements and new features in blue

- Automatic and user-controlled time stepping
- Ability to specify initial stress state
- Importing meshes
  - LaGriT: GMV/Pset
  - CUBIT: Exodus II
  - ASCII: PyLith mesh ASCII format (intended for toy problems only)
- Output: VTK files
  - Solution over volume
  - Solution over surface boundary
  - State variables (e.g., stress and strain) for each material
  - Fault information (e.g., slip and tractions)
- Automatic conversion of units for all parameters
PyLith 1.5: Under-the-hood Improvements

- Additional cleanup of C++ code
- Optimization of several modules
  - Mesh distribution among processors
  - Integration of elasticity terms
- Ability to use algebraic multigrid preconditioners
Time-Dependent Boundary Conditions

Dirichlet, Neumann, and Point Forces

\[ f(\vec{x}) = \]

\[ f_0(\vec{x}) + \quad \text{db}_\text{initial} \]
\[ \dot{f}_1(\vec{x})(t - t_1(\vec{x})) + \quad \text{db}_\text{rate} \]
\[ f_2(\vec{x})a(t - t_2(\vec{x})) \quad \text{db}_\text{change} \]

**db_initial** Initial value (constant in time)

**db_rate** Constant rate of change (spatially variable start time)

**db_change** Time history (spatially variable amplitude and start time)
PyLith as a Hierarchy of Components

Components are the basic building blocks
PyLith as a Hierarchy of Components

PyLith Application and Time-Dependent Problem

**PyLithApp**

- **properties**
  - none

- **facilities**
  - mesh_generator

- problem

- petsc

**TimeDependent**

- **properties**
  - dimension

- **facilities**
  - normalizer

  - materials

  - bc

  - interfaces

  - gravity_field

  - formulation
PyLith as a Hierarchy of Components

Fault with kinematic (prescribed slip) earthquake rupture

FaultCohesiveKin

- properties
  - id
  - name
  - up_dir
  - normal_dir
- facilities
  - quadrature
  - eq_srcs
  - output

EqKinSrc

- properties
  - origin_time
- facilities
  - slip_function
PyLith Application Flow

PyLithApp

main()
    mesher.create()
    problem.initialize()
    problem.run()

TimeDependent (Problem)

initialize()
    formulation.initialize()

run()
    while (t < totalTime)
        dt = formulation.getTimeStep()
        formulation.prestep()
        formulation.step()
        formulation.poststep()

Implicit (Formulation)

initialize()

prestep()
    set constraints

step()
    calculate residual
    solve for displacement increment

poststep()
    update displacement field
    write output
Ingredients for Running PyLith

- Simulation parameters
  - .cfg ASCII files
  - pylithapp.cfg always read if it exists
  - Command line arguments

- Finite-element mesh
  - Mesh exported from LaGriT
  - Mesh exported from CUBIT
  - Mesh constructed by hand (PyLith mesh ASCII format)

- Spatial databases for physical properties, boundary conditions, and rupture parameters
Example: 3d/hex8 step01.cfg

Compression and shear via prescribed displacements
Example: 3d/hex8 step01.cfg
Example: 3d/hex8 step01.cfg

Input

- Simulation parameters
  - pylithapp.cfg
  - step01.cfg

- CUBIT Mesh:
  - mesh_hex8_1000m.mesh

- Spatial databases
  - mat_elastic.spatialdb
  - axialdisp.spatialdb

Output

- Displacement field
  - step01_t000000.vtk
  - step01-groundsurf_t000000.vtk

- State variables
  - Upper crust (elastic)
    - step01-statevars_info.vtk
      - physical properties
    - step01-statevars_t000000.vtk
      - stress and strain
  - Lower crust (elastic)
    - step01-statevars_info.vtk
      - physical properties
    - step01-statevars_t000000.vtk
      - stress and strain
Example: 3d/hex8 step06.cfg

Creep and repeated rupture on a strike-slip fault
Example: 3d/hex8 step06.cfg

Creep and repeated rupture on a strike-slip fault
Example: 3d/hex8 savageprescott.cfg

Input

- Simulation parameters
  - pylithapp.cfg
  - step06.cfg
- Mesh: mesh_hex8_1000m.exo
- Spatial databases
  - mat_elastic.spatialdb
  - mat_maxwell.spatialdb
  - finalsrip_rupture.spatialdb
  - sliptime.spatialdb
  - sliprate_creep.spatialdb

Output

- Displacement field
  - step06_tNNNN.vtk
  - step06-groundsurf_tNNNN.vtk
- State variables
  - Upper crust (elastic)
    - step06-upper_crust_info.vtk
    - step06-upper_crust_tNNNN.vtk
  - Lower crust (viscoelastic)
    - step06-lower_crust_info.vtk
    - step06-lower_crust_tNNNN.vtk
- Fault
  - step06-fault_info.vtk
  - step06-fault_tNNNN.vtk
Useful Tips/Tricks

- `pylithinfo [--verbose] [PyLith args]`
  Dumps all parameters with their current values to text file

- **Command line arguments**
  - `--help`
  - `--help-components`
  - `--help-properties`
  - `--petsc.start_in_debugger` (run in xterm)
  - `--nodes=N` (to run on N processors on local machine)

- **PyLith User Manual**

- **CIG Short-Term Tectonics mailing list**
  cig-short@geodynamics.org

- **CIG bug tracking system**
  http://www.geodynamics.org/roundup