PyLith Simple Example Problem: Dike + Magma Chamber + Topography

CIG COMPUTATIONAL INFRASTRUCTURE for GEODYNAMICS

www.geodynamics.org

Charles Williams (GNS Science)

Adrian Shelley (Victoria University Wellington)

Steps in Running a PyLith Simulation

- Define necessary geometry
- Create a mesh (we will use CUBIT today)
- Set up simulation parameters (.cfg files)
- Create necessary spatial databases
- Run simulation
  - Can run in parallel by typing

  pylith xxx.cfg --nodes=2
- Analyze/visualize/postprocess results

Defining Geometry/Creating Mesh

- Create NURBS surface representing topography
  - Python script to subsample DEM and create CUBITjournal files
  - Run CUBIT to create NURBS surface and export it for later use
- Import NURBS surface into CUBIT
- Create planar dike and spherical magma chamber within CUBIT
- Generate and export mesh

Set up Simulation Parameters

- Create pylithapp.cfg file
  - Automatically read by PyLith
  - Contains parameters common to a number of simulations (mesh, far-field BC, etc.)
- Create two_source_dike_slip.cfg file
  - Contains parameters specific to a particular problem (number of sources, material properties, etc.)
  - Add this on the command-line when running PyLith

Create Necessary Spatial Databases

- Material property databases
  - outer_const_elastic.spatialdb: Elastic properties for outer material
  - ruapehu_const_elastic.spatialdb: Elastic properties for material near volcano
- Boundary condition databases
    - magma_pressure.spatialdb: Pressure BC inside chamber
    - dikeslip.spatialdb: Amount of opening for dike
    - sliptime.spatialdb: Time at which dike opening starts

Run Simulation

- Run simulation by typing:

  ```
  pylith two_source_dike_slip.cfg --nodes=2
  ```

- Generally useful to redirect output to a log file (e.g., on bash):

  -
    ```
    pylith two_source_dike_slip.cfg 2>&1 | tee run.log
    ```

Visualize Results

- We will use ParaView to examine the various HDF5 files
- Within ParaView, make sure you open the *.xmf files
  - The *.xmf files describe the HDF5 layout so that ParaView can read the HDF5 files