

# PyLith Modeling Tutorial

## Troubleshooting PyLith Simulations

Brad Aagaard



June 10, 2019

# What parameters are available?

Parameters are specified as a hierarchy of components and properties

- Components (Facilities) are the object building blocks  
Appendix B of the PyLith manual lists all of the components
  - Problem **TimeDependent**
  - Boundary conditions **DirichletTimeDependent**
  - Faults **FaultCohesiveKin**
  - Materials **Elasticity**
  - Solution observers **OutputSolnBoundary**
  - Readers **MeshIOCubit**
- Properties are the basic types
  - String **mat\_viscoelastic.spatialdb**
  - Integer **4**
  - Float **2.3**
  - Dimensioned quantity **2.5\*year**
  - Array of Strings, Integers, or Floats **[0, 0, 1]**

# Parameter Files

Simple syntax for specifying parameters for properties and components

```
# Syntax
[pylithapp.COMPONENT.SUBCOMPONENT] ; Inline comment
COMPONENT = OBJECT
PARAMETER = VALUE

# Example
[pylithapp.mesh_generator] ; Header indicates path of mesh_generator in hierarchy
reader = pylith.meshio.MeshIOCubit ; Use mesh from CUBIT/Trelis
reader.filename = mesh_quad4.exo ; Set filename of mesh.
reader.coordsys.space_dim = 2 ; Set coordinate system of mesh.

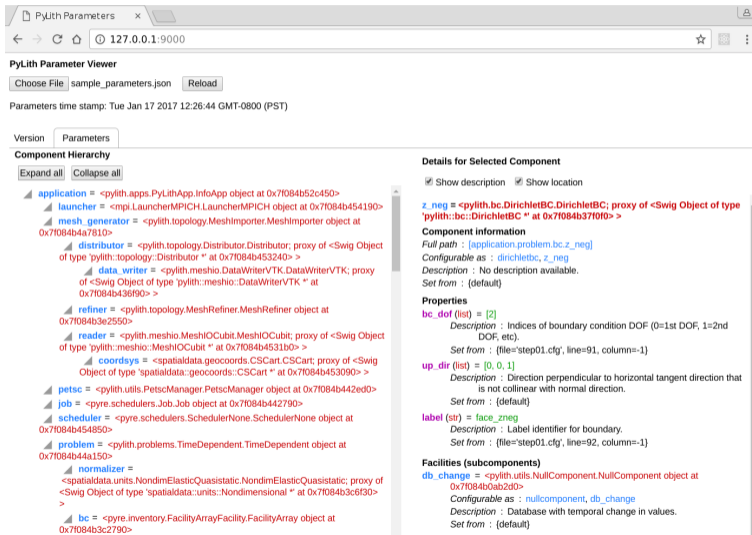
[pylithapp.problem.solution_outputs.output] ; Set output format
writer = pylith.meshio.DataWriterHDF5
writer.filename = axialdisp.h5

[pylithapp.problem]
bc = [x_neg, x_pos, y_neg] ; Create array of boundary conditions
bc.x_neg = pylith.bc.DirichletTimeDependent ; Set type of boundary condition
bc.x_pos = pylith.bc.DirichletTimeDependent
bc.y_neg = pylith.bc.DirichletTimeDependent

[pylithapp.problem.bc.x_pos] ; Boundary condition for +x
constrained_dof = [0] ; Constrain x DOF
label = edge_xpos ; Name of nodeset from CUBIT/Trelis
db_auxiliary_field = spatialdata.spatialdb.SimpleDB ; Set type of spatial database
db_auxiliary_field.label = Dirichlet BC +x edge
db_auxiliary_field.iohandler.filename = axial_disp.spatialdb ; Filename for database
```

# Parameters Graphical User-Interface

```
cd parametersgui; ./pylith_paramviewer
```



PyLith Parameter Viewer

Choose File sample\_parameters.json Reload

Parameters time stamp: Tue Jan 17 2017 12:26:44 GMT-0800 (PST)

Version Parameters

Component Hierarchy

Expand all Collapse all

- application = <pylith.apps.PyLithApp.InfoApp object at 0x7f084b52c450>
  - launcher = <mpi.LauncherMPICH.LauncherMPICH object at 0x7f084b454190>
  - mesh\_generator = <pylith.topology.MeshImporter.MeshImporter object at 0x7f084b4a7810>
    - distributor = <pylith.topology.Distributor.Distributor; proxy of <Swig Object of type 'pylith::topology::Distributor \*' at 0x7f084b453240> >
      - data\_writer = <pylith.meshio.DataWriterVTK.DataWriterVTK; proxy of <Swig Object of type 'pylith::meshio::DataWriterVTK \*' at 0x7f084b436f90> >
      - refiner = <pylith.topology.MeshRefiner.MeshRefiner object at 0x7f084b3e2550>
      - reader = <pylith.meshio.MeshIOCube.MeshIOCube; proxy of <Swig Object of type 'pylith::meshio::MeshIOCube \*' at 0x7f084b4531b0> >
        - coordsys = <spatialdata.geocoords.CSCart.CSCart; proxy of <Swig Object of type 'spatialdata::geocoords::CSCart \*' at 0x7f084b453090> >
    - petsc = <pylith.utils.PetscManager.PetscManager object at 0x7f084b442ed0>
    - job = <pyre.schedulers.Job.Job object at 0x7f084b442790>
    - scheduler = <pyre.schedulers.SchedulerNone.SchedulerNone object at 0x7f084b454850>
    - problem = <pylith.problems.TimeDependent.TimeDependent object at 0x7f084b44a150>
      - normalizer = <spatialdata.units.NondimElasticQuasistatic.NondimElasticQuasistatic; proxy of <Swig Object of type 'spatialdata::units::Nondimensional \*' at 0x7f084b3c6f30> >
      - bc = <pyre.inventory.FacilityArrayFacility.FacilityArray object at 0x7f084b3c2790>

# Parameters Graphical User-Interface

Case study: `examples/2d/box/step02_sheardisp`

- 1 Generate the JSON file with the parameters

```
cd examples/2d/box  
pylithinfo step02_sheardisp.cfg
```

- 2 Start the web-server (start at your top-level PyLith directory)

```
cd parametersgui  
./pylith_paramviewer
```

- 3 Point your web browser to `http://127.0.0.1:9000`

- 4 Load the parameter file

# Show values of parameters using the command line

Case study: `examples/2d/box/step02_sheardisp`

- Components and properties for given component `--help`  
`step02_sheardisp.cfg` [pylithapp.problem.bc.y\_neg]  
`shell` `pylith step02.cfg --problem.bc.y_neg.help`
- Current components of a given component `--help-components`  
`step02_sheardisp.cfg` [pylithapp.problem.bc.y\_neg]  
`shell` `pylith step02_sheardisp.cfg --problem.bc.y_neg.help-components`
- Current properties of a given component `--help-properties`  
`step02_sheardisp.cfg` [pylithapp.problem.bc.y\_neg]  
`shell` `pylith step02_sheardisp.cfg --problem.bc.y_neg.help-properties`

# What about a GUI for editing parameters?

On the wish list but will require time or a developer

- Parameter viewer → editor
  - See possible choices for components and properties
  - Basic validation of parameters
  - ⇒ Generate JSON schema from component specifications
  - ⇒ Translate JSON schema into GUI
- Export parameters to single file
  - Facilitates archiving parameters used in given simulation

# Troubleshooting Examples

See [examples/troubleshooting/nofaults-2d](#)

**Introduce common (and a few uncommon) errors into 2d/box input files**