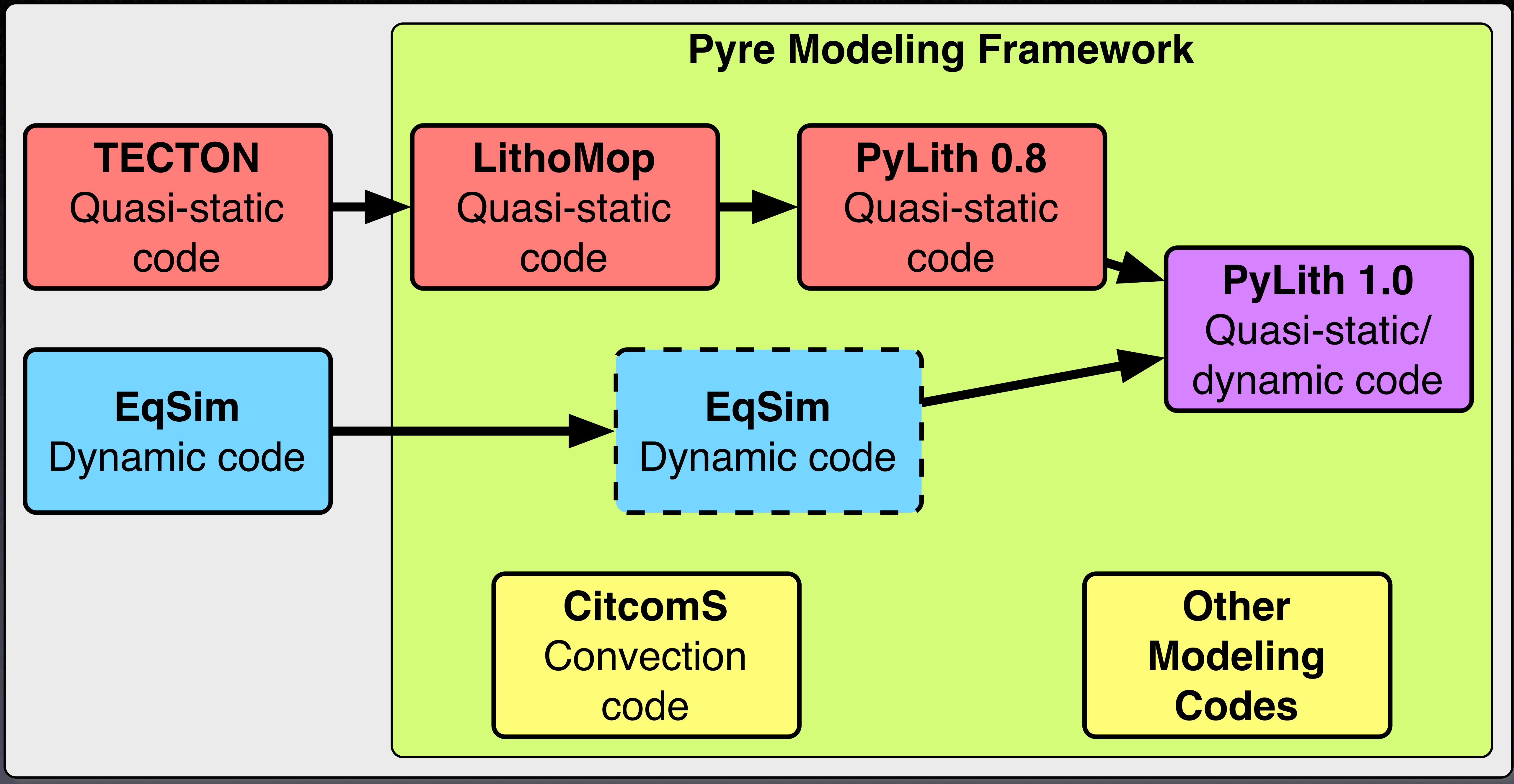
Why We Keep Changing the Name of TECTON/LithoMop/EqSim/PyLith (We promise we will never change the name again)

Charles Williams
Rensselaer Polytechnic Institute

Other People Involved in Code Development

- Matt Knepley (CIG/ANL)
  - *PETSc/Sieve development and implementation, parallelization.*
- Brad Aagaard (USGS)
  - *External packages, merging of LithoMop and EqSim into PyLith.*
- Leif Strand (CIG)
  - *Build system, SVN repository.*
- Michael Aivazis (CIG/CACR)
  - *Pyre/pythia.*

Overview of Code Development Status



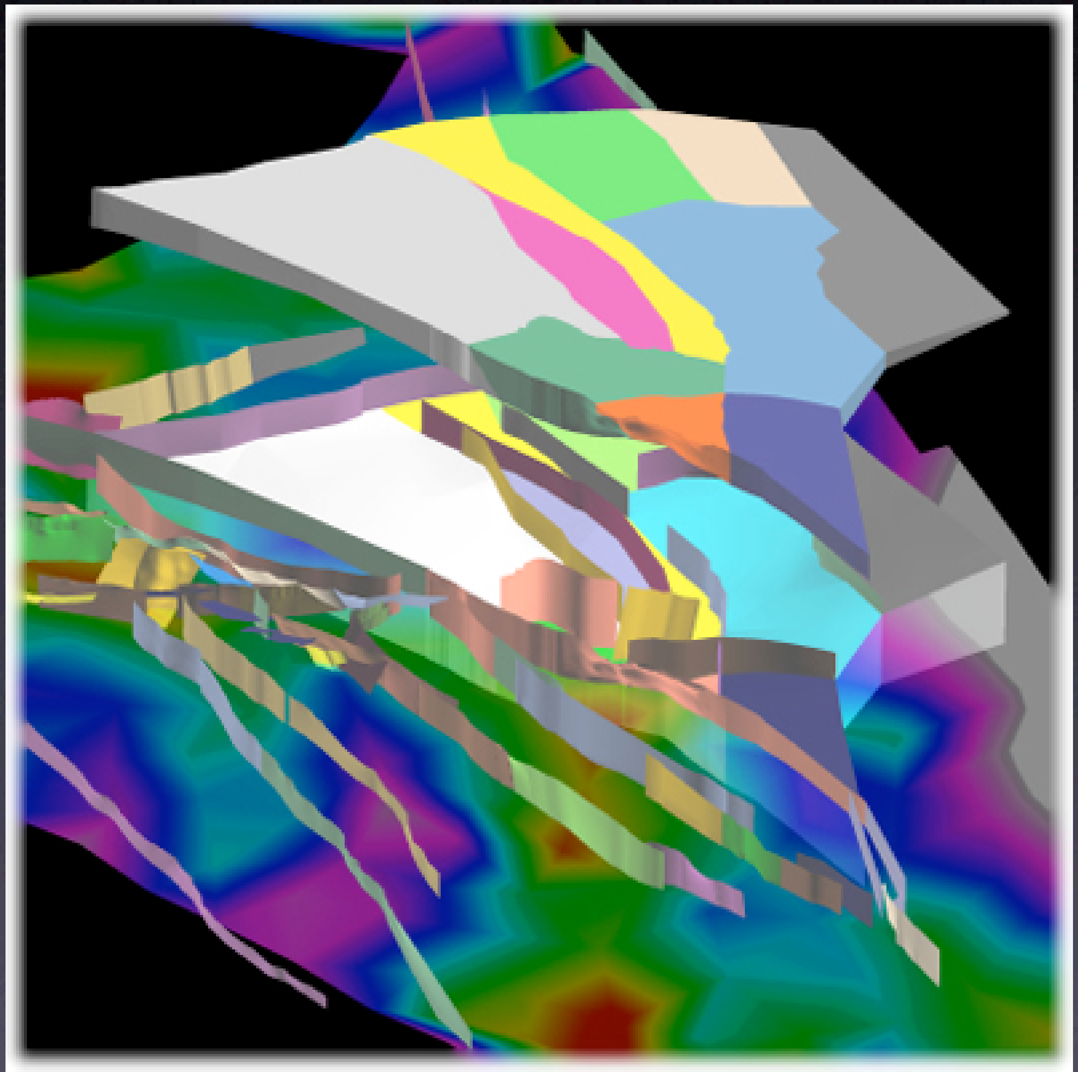
What Types of Problems Would We Like To Solve?

- Stress/strain evolution over multiple earthquake cycles.
  - *Complex geometries, rheologies, and boundary conditions.*
  - *Realistic fault behavior on multiple time scales.*
- Large-scale models of inter/intra-plate stresses.
  - *Large spatial scales, spherical geometry.*
  - *Contributions from gravity and lithosphere/mantle coupling.*
- Volcanically-induced stresses (intrusion, emplacement, etc.).
  - *Complex structure and plastic/viscoplastic rheology.*

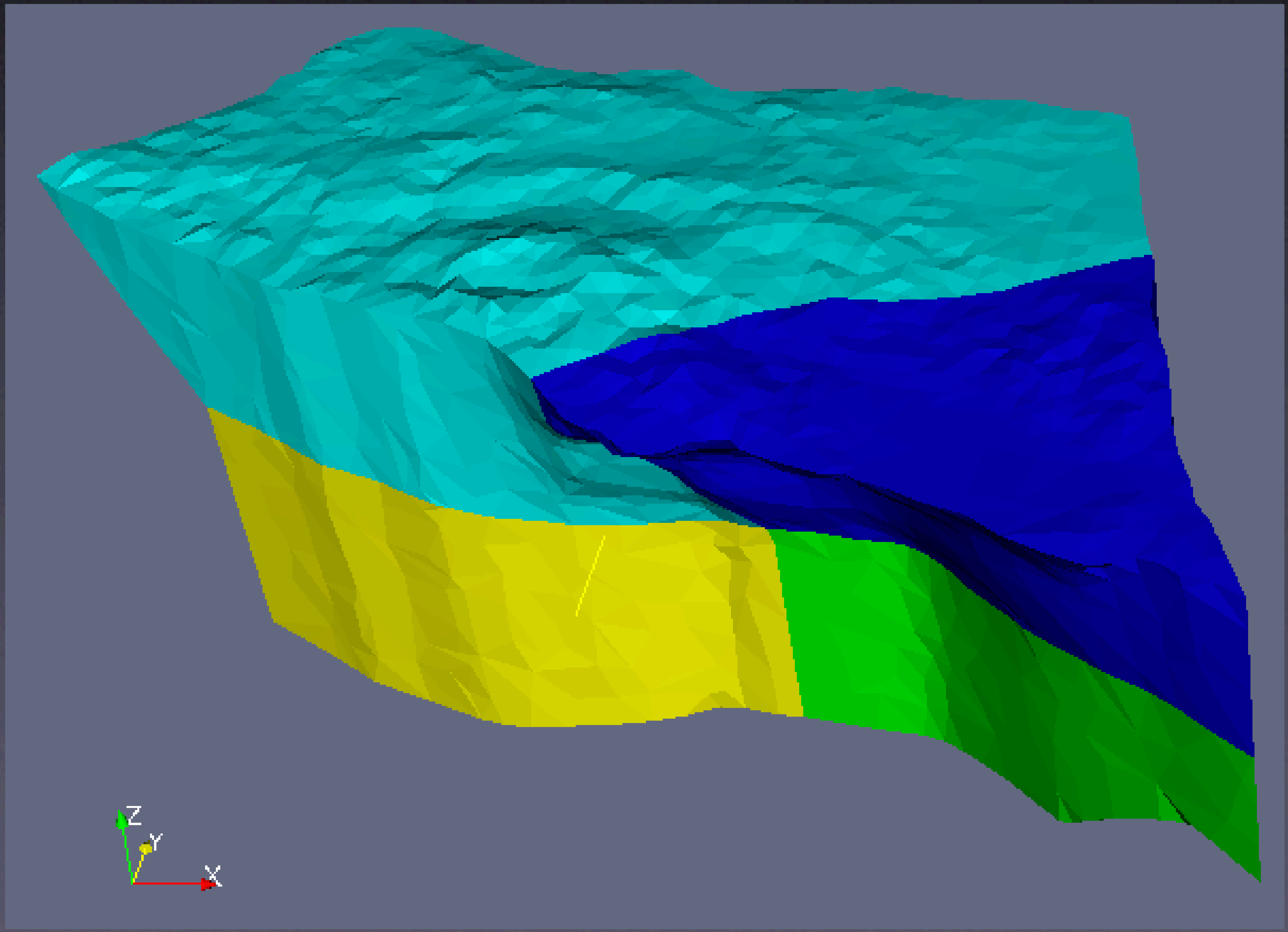
Modeling of Multiple Earthquake Cycles

- Very large mesh.
- Complex geometry and BC.
- Need good fault representations.

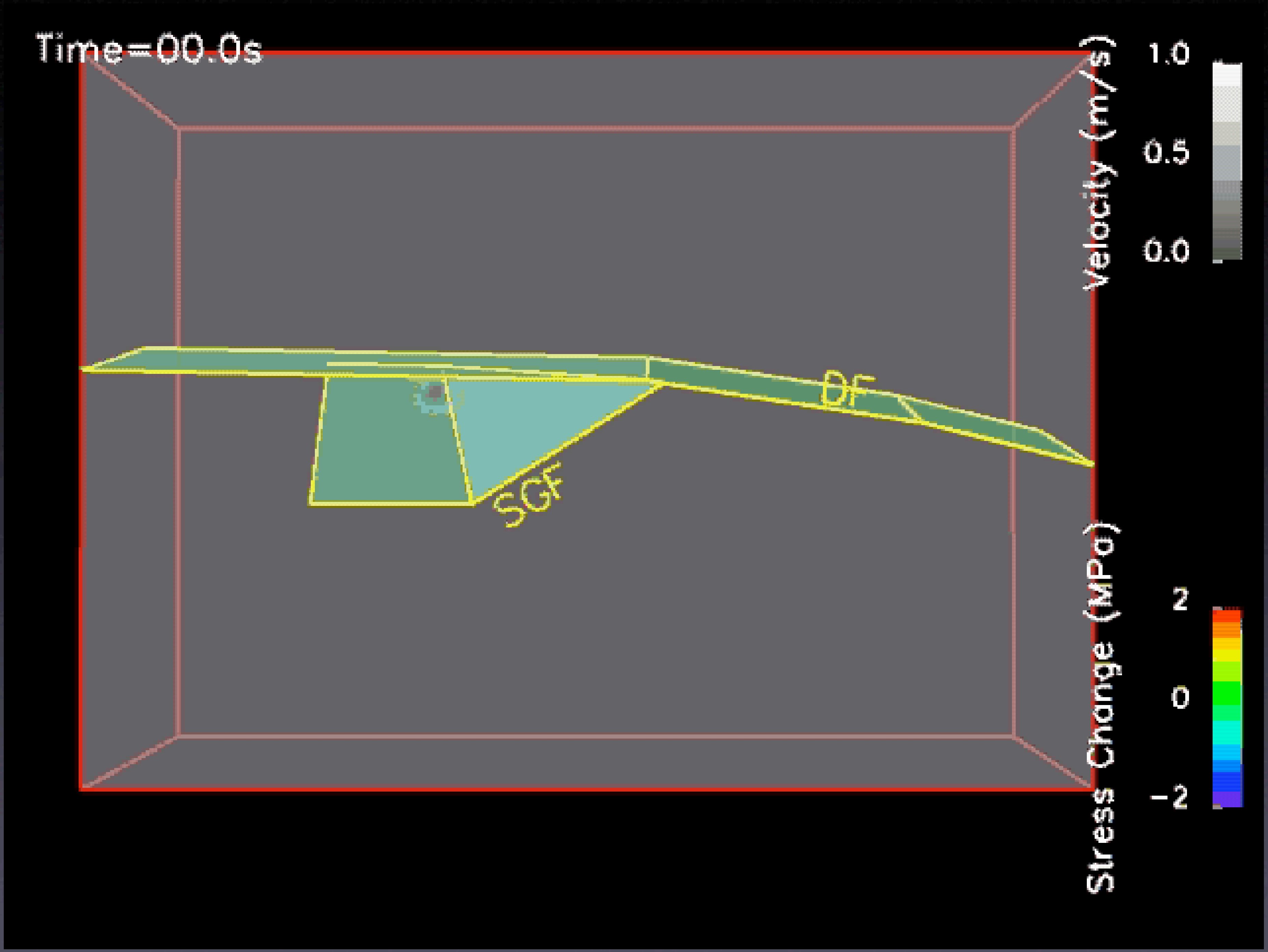
View of CBM for Mojave region from
http://structure.harvard.edu/cfm/blockmodel.jpg



Moderate-resolution mesh (56,472 nodes; 342, 998 elements) created by Carl Gable, LANL.



Stress Changes Due to Dynamic Rupture Propagation

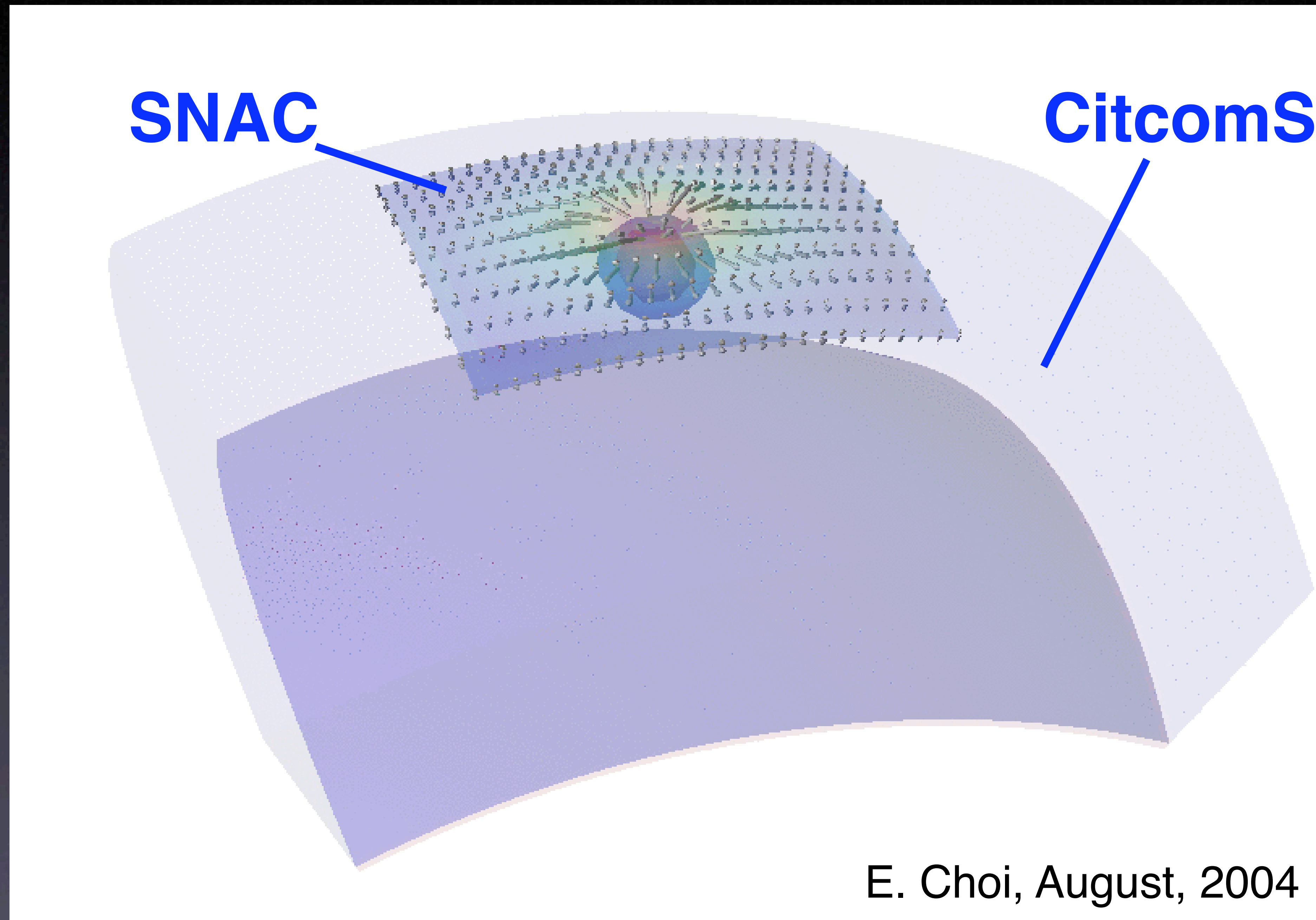


Movie of 2002 *M* 7.9 Denali earthquake provided by Brad Aagaard, USGS.

- Quasi-static solution is needed to provide realistic initial conditions for dynamic solution.
- At end of dynamic solution, there are regions of high shear stress, which would likely be reduced by viscoelastic relaxation (quasi-static).

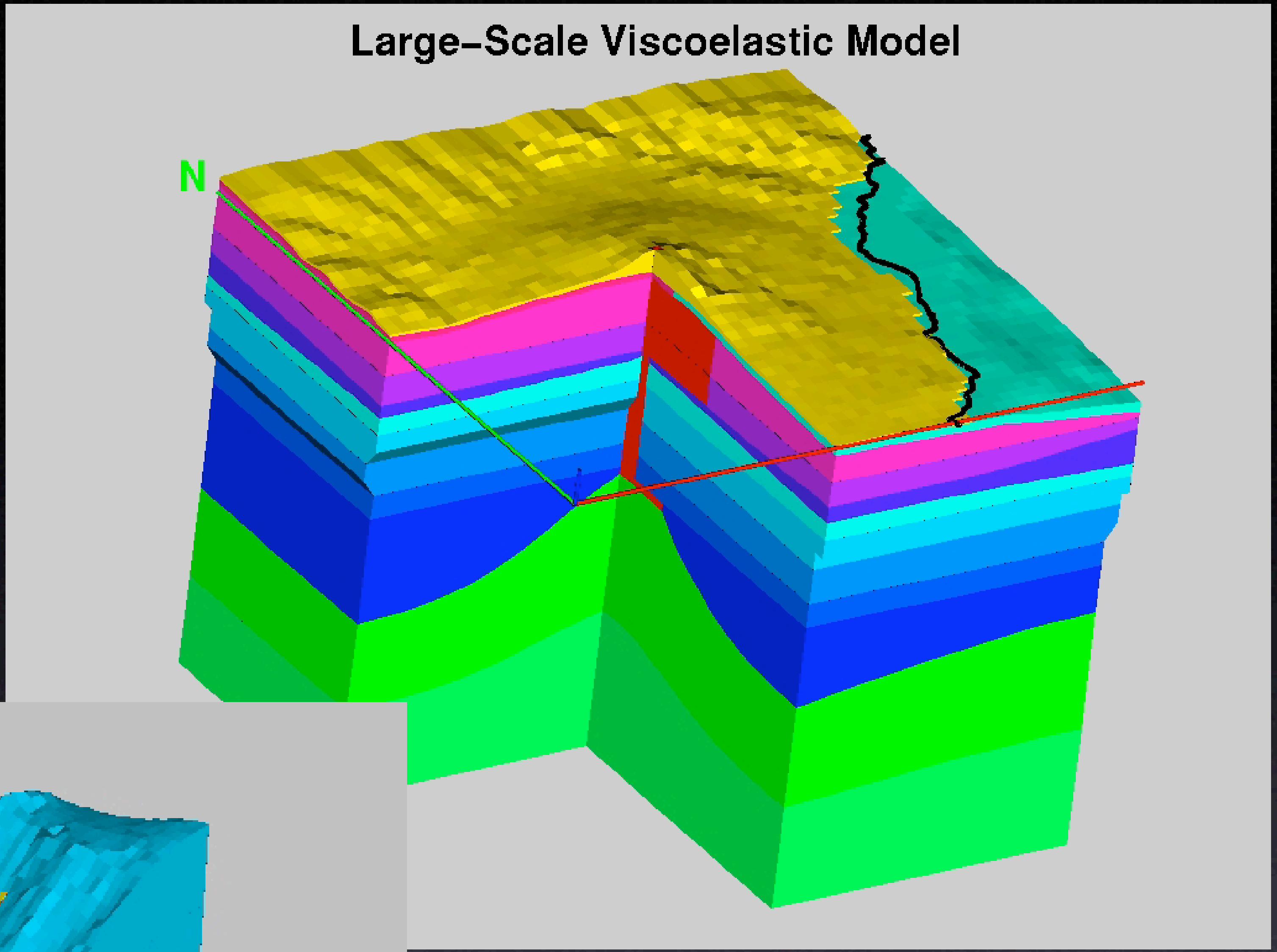
Large-Scale Problems in Lithospheric Dynamics

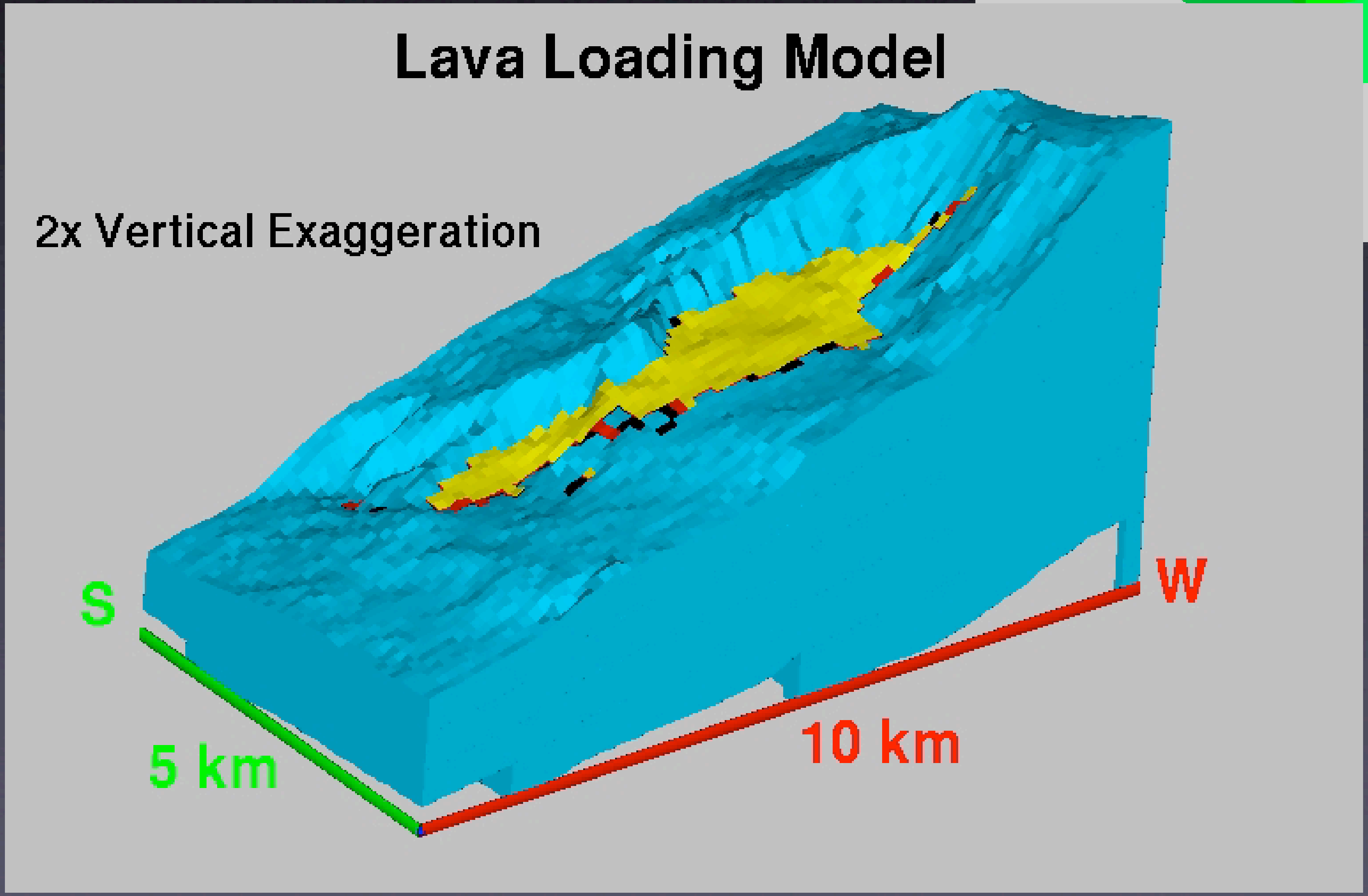
- Contributions from mantle convection.
- Spherical geometry.
- Very large meshes.



Volcanic Deformation

- Complex geometry and structure.
- Complex rheologies.





What if We Want to Perform an Inversion?

- Need an easy mechanism for replacing a portion of the forward problem (e.g., BC or material properties) without re-entering all simulation information.

- Need an easy method for sampling the results at specified times and locations for computation of cost functions.

Sample Problem

Inversion for material properties and/or BC of the CBM using geodetic data.

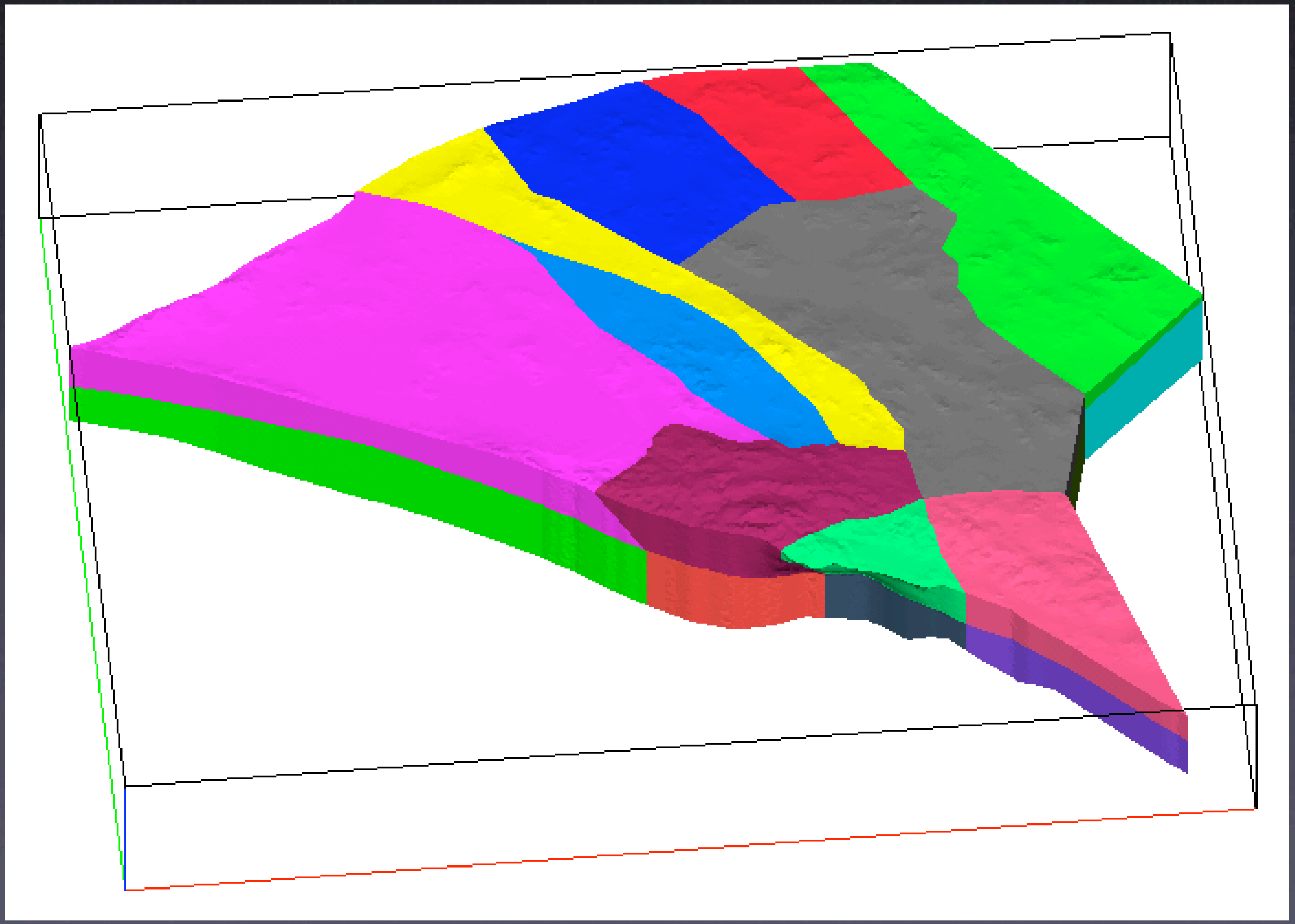
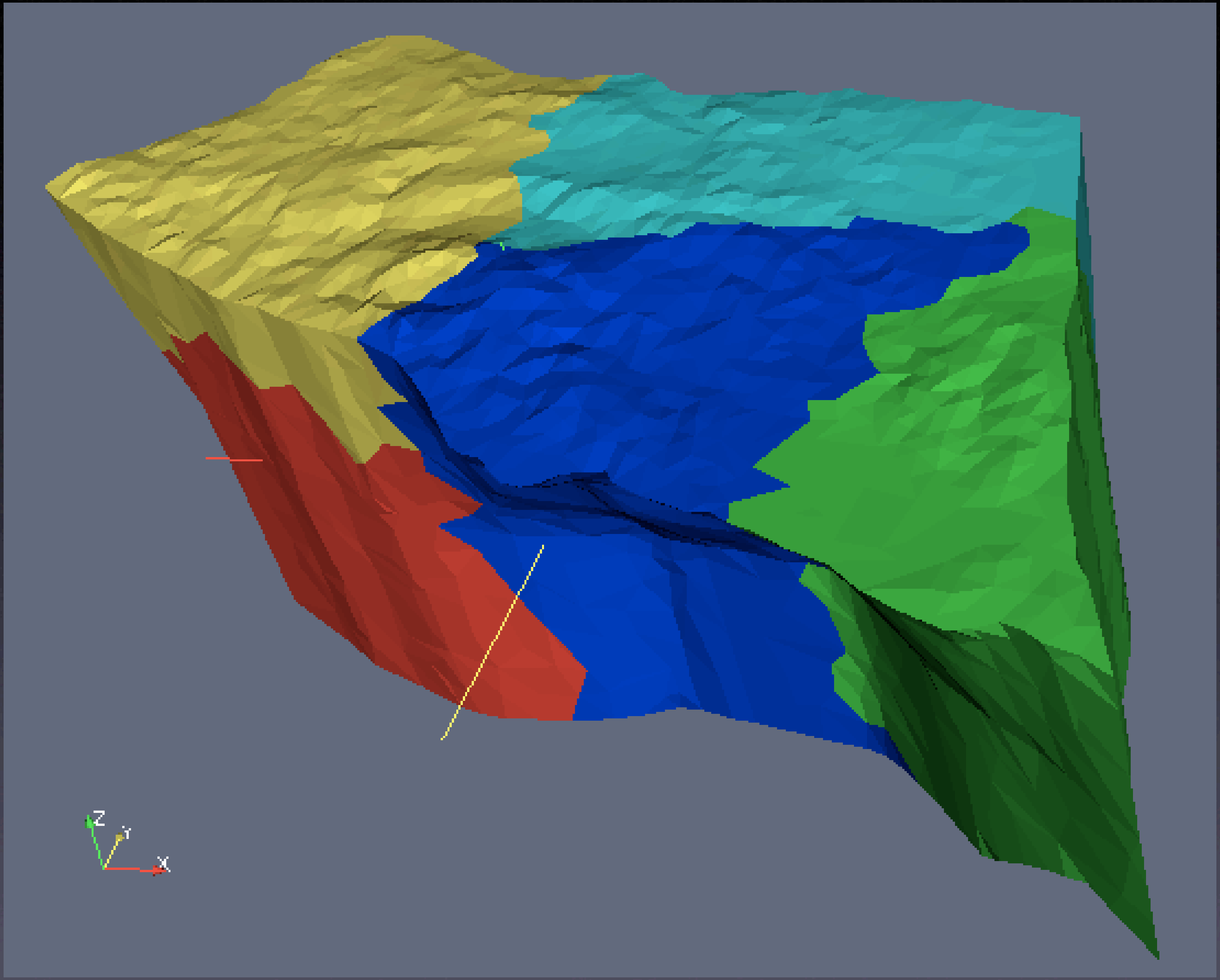


Image of Mojave portion of CBM from Carl Gable, LANL.

What Code Features Will Be Needed to Solve These Problems?

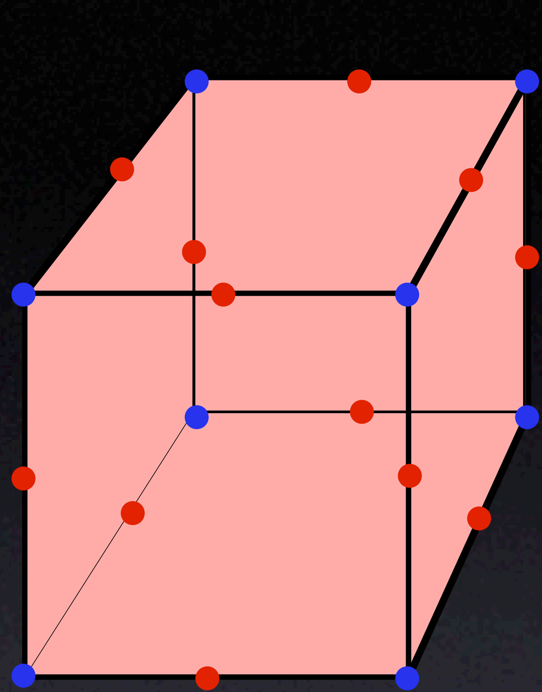
- Able to deal with very large meshes (parallel code).
- Able to represent complex structures and geometries.
- Able to model different types of fault behavior.
- Able to couple with other codes.
- Modular design.
  - *Extensible and adaptable physics, geometries, material models, etc.*
  - *Ability to replace portions of a simulation 'on the fly'.*

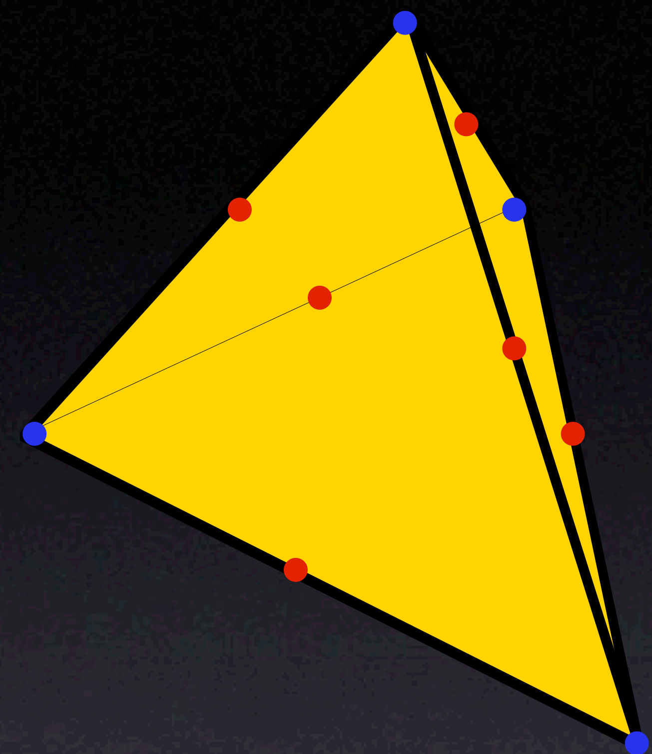
Parallelization Using PETSc and Sieve



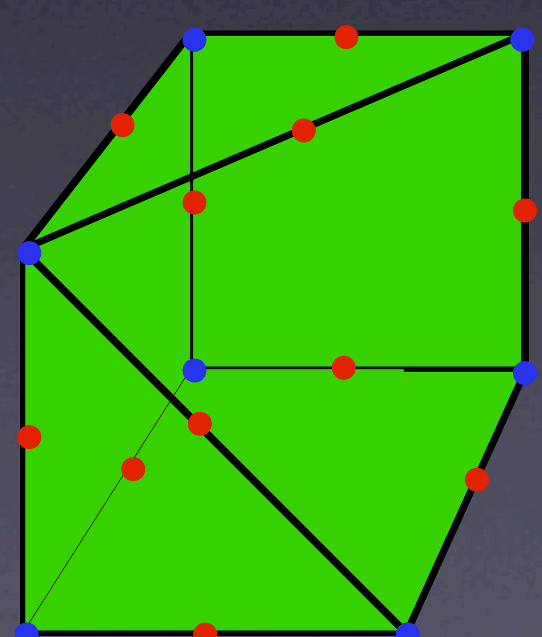
- Mesh represented in terms of hierarchical *covering* relations.
- Sieves are inherently parallel, eliminating need for constructs such as "ghost nodes".
- Sieves make parallelization much easier.
- PETSc/Sieve can import meshes from a variety of meshing packages, perform partitioning and refinement, and provide parallel data structures that may be used directly by PyLith.

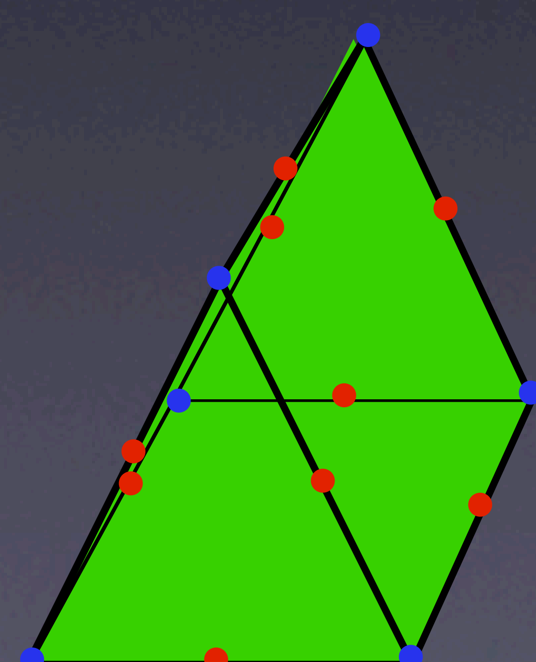
Geometric Flexibility Through Multiple Element Types

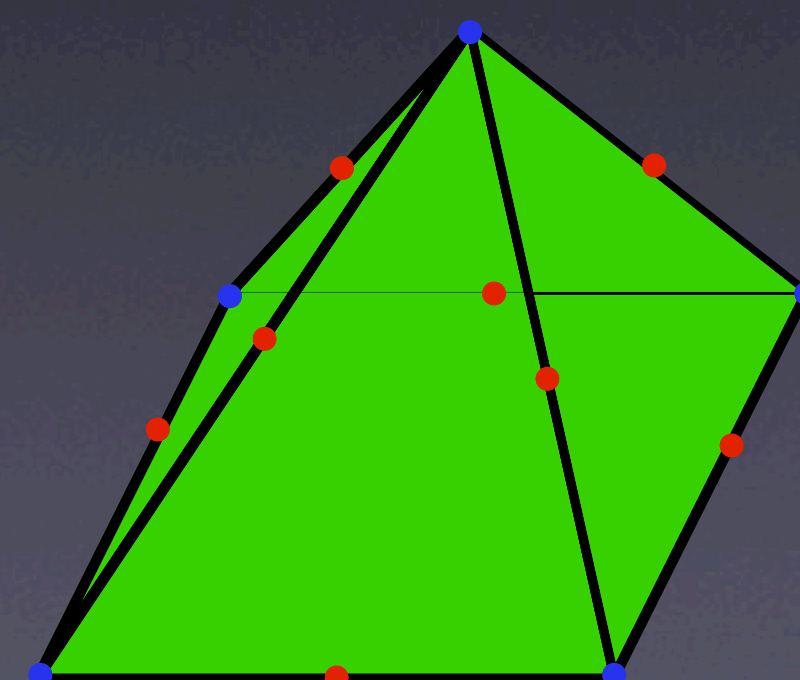




- Multiple element types provide compatibility with many meshing packages.
- Different element types may be combined for greater flexibility.
- We will use the FIAT (FInite element Automatic Tabulator) package to use arbitrary element types of any order.

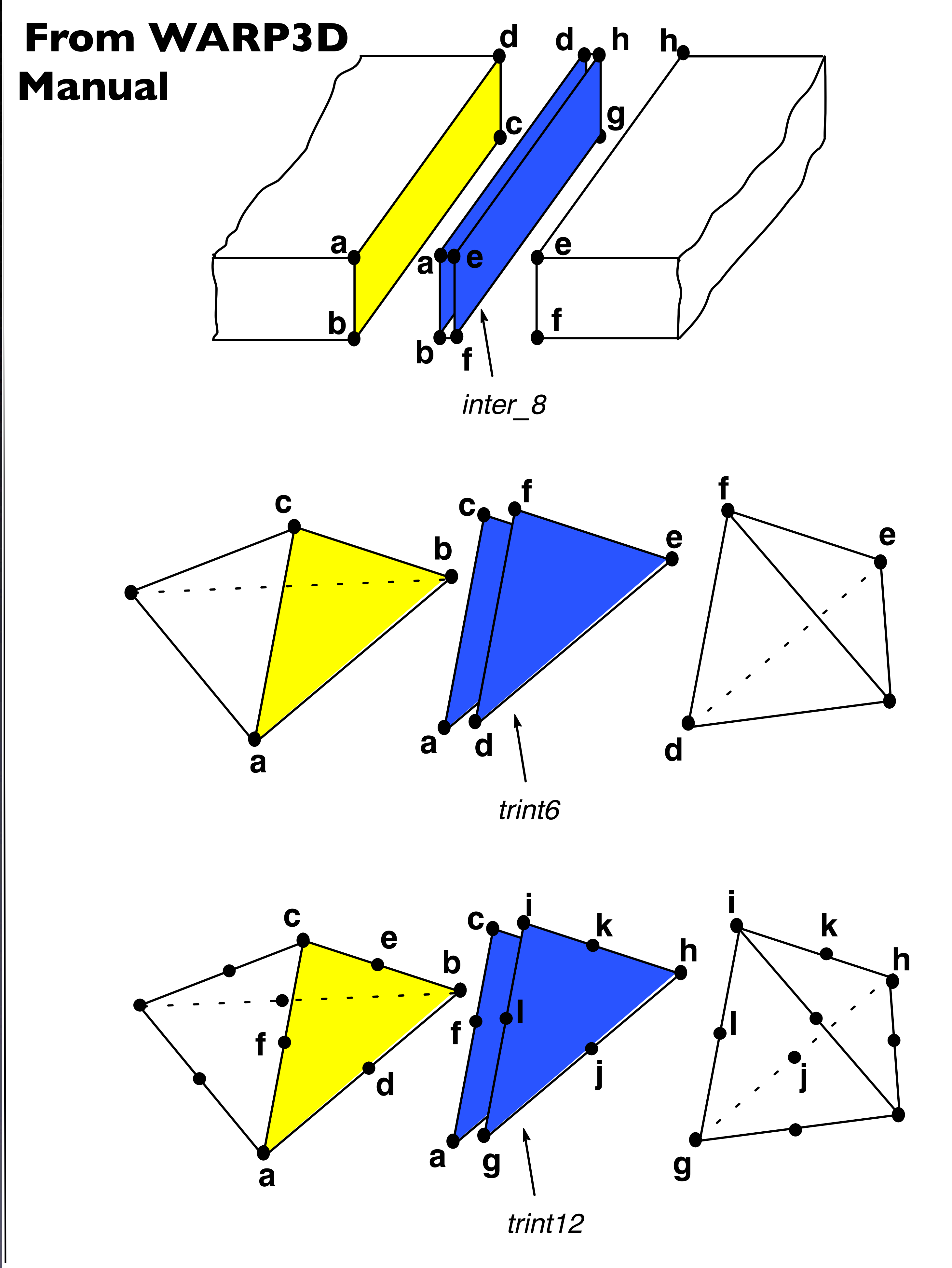






Faults

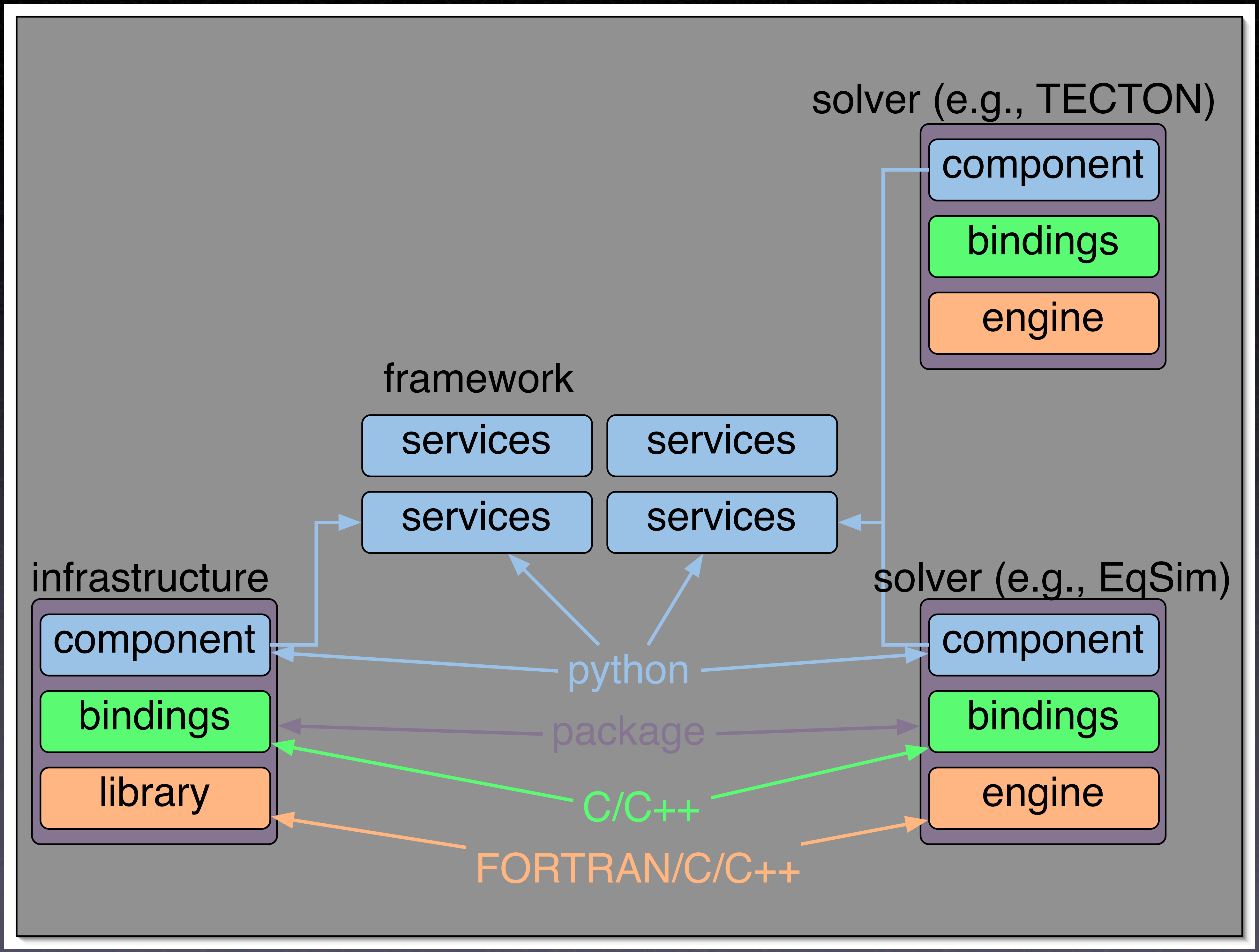
- Presently:
  - *Kinematic case: 'Split nodes'*
  - *'Dynamic' case: 'Slippery nodes'*
- Soon - cohesive elements:
  - *Kinematic case: Specified differential slip.*
  - *'Dynamic' case: Slip determined by element constitutive relationship.*
- Eventually - discontinuous elements:
  - *Faults can cut through elements for kinematic or dynamic case.*



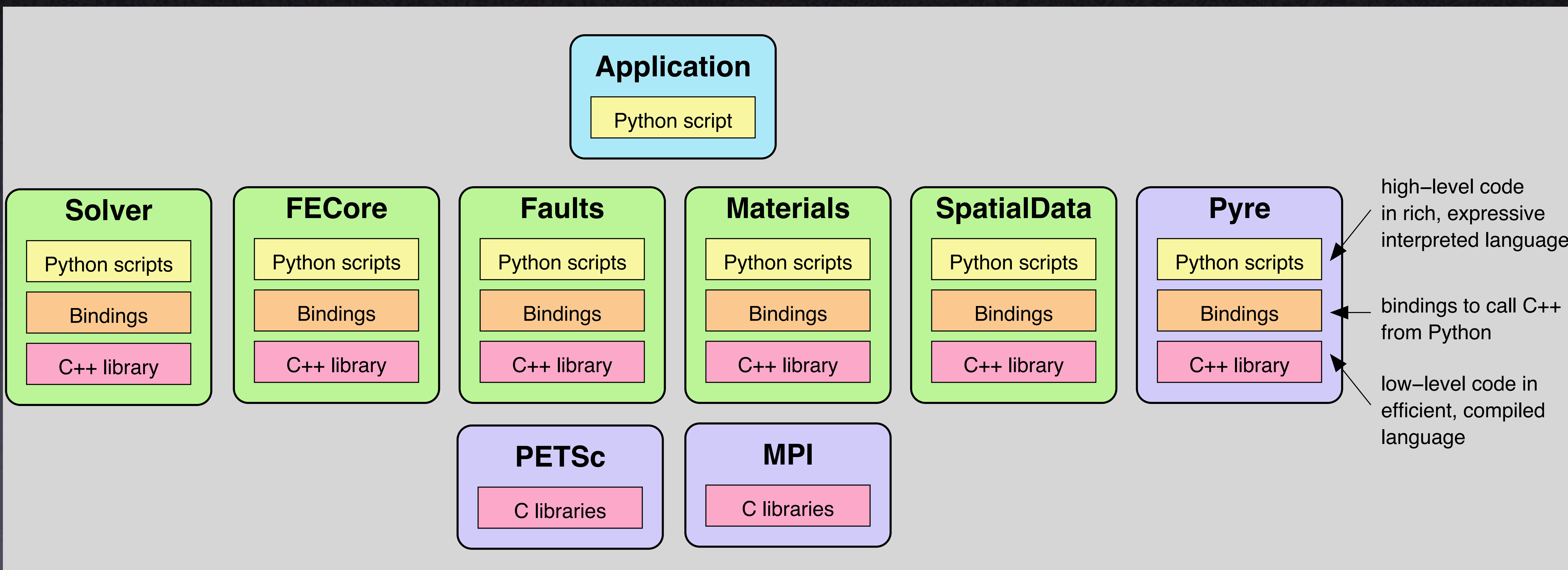
Pyrization -- Use of the Pyre Simulation Framework

- Full Pyrization implies:
  - *Full use of Pyre facilities.*
  - *Completely modular code design.*
  - *Use of interchangeable components.*
  - *Easy scripting ability, allowing usage of whichever components are needed for a particular problem.*

Pyre Architecture

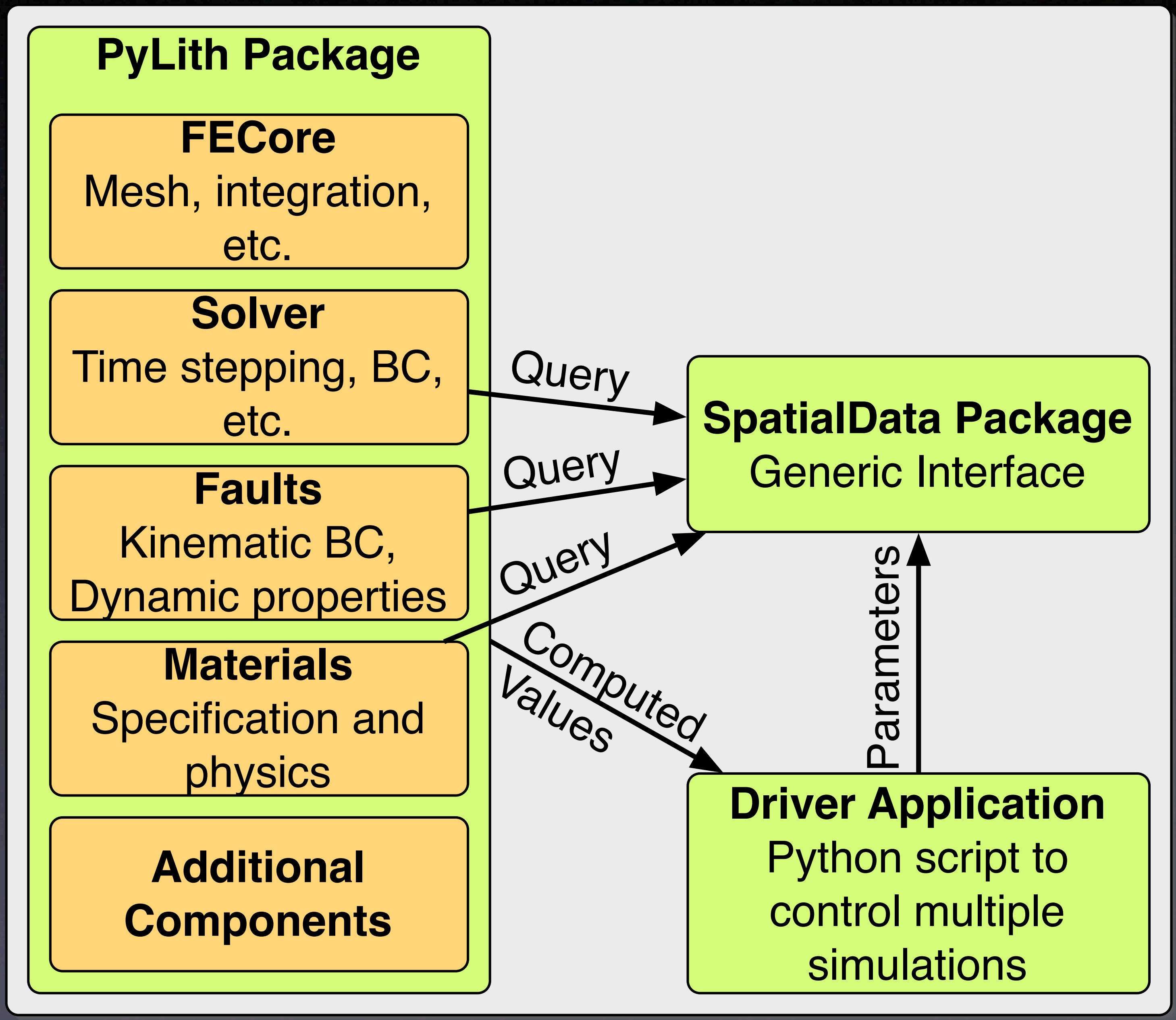


Structure of PyLith 1.0 Package



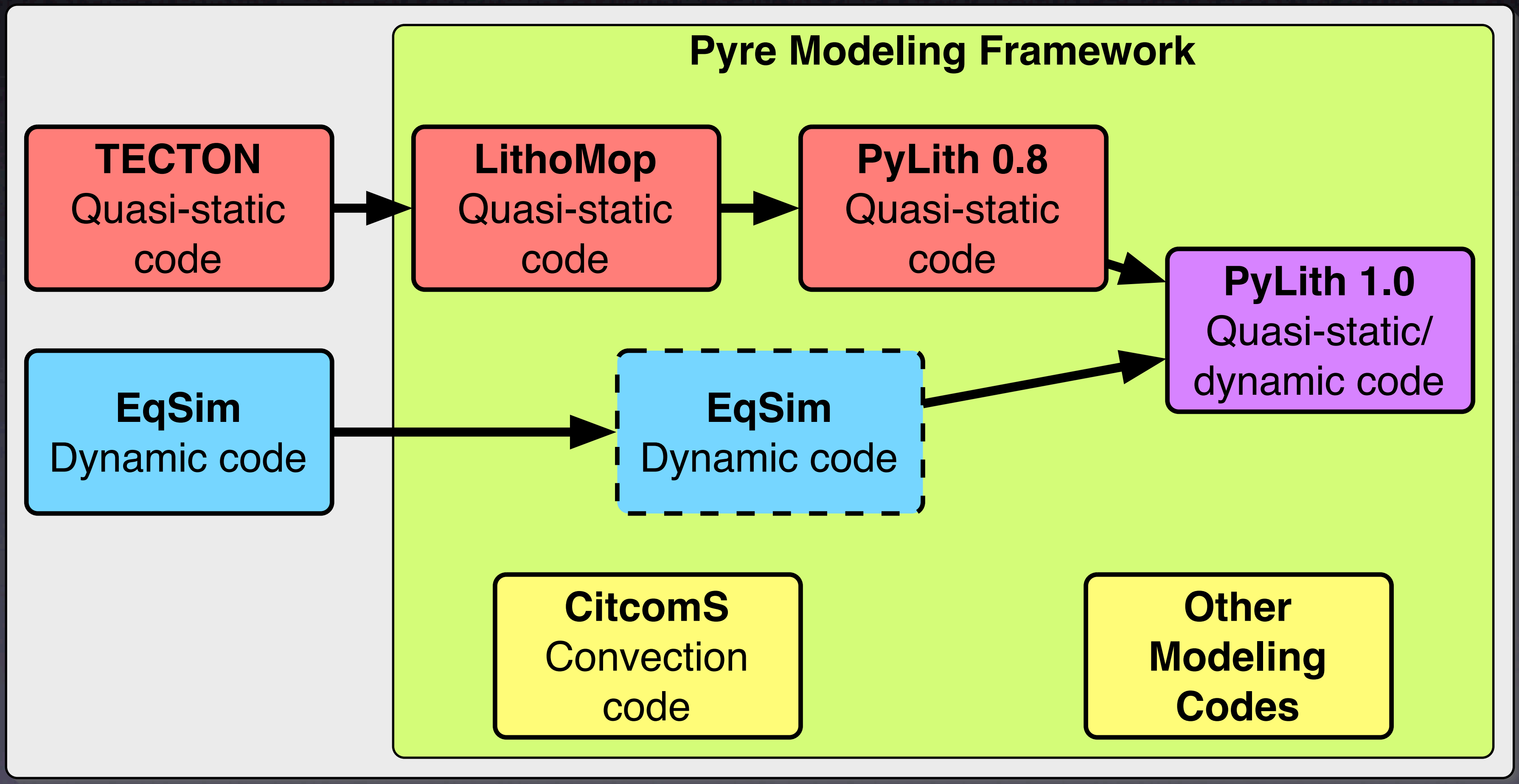
Sample Setup for an Inversion

- Python script can compute cost functions based on computed values at specified times/locations.
- Script can also generate sets of parameters to be used by the code, replacing only a subset of the simulation information.



Overview of Code Development Status

- Merging of LithoMop and EqSim will facilitate simulation of problems involving interseismic and coseismic time scales.
- Integration of both codes into Pyre framework will facilitate coupling with other codes in the framework.
- Modular design will simplify the addition of new physics, material models, geometries, etc.



How Desired Features Are Being Addressed

- Wide range of spatial and temporal scales.
  - *Parallelization using PETSc/Sieve.*
  - *Merging LithoMop + EqSim -> PyLith.*
  - *Allowing code coupling using Pyre.*
- Complex structures and geometries.
  - *Multiple element types.*
  - *Use Sieve for interfacing with meshing packages.*
- Different types of geometries (e.g., spherical).
  - *Most components are independent of geometry.*
  - *Use of FIAT provides basis functions for any geometry.*
  - *Modular design will facilitate addition of new geometries.*

How Desired Features Are Being Addressed (cont.)

- Faults and complex rheologies.
  - *Modular design simplifies addition of new rheologies.*
  - *Cohesive elements with easily-defined constitutive relations.*
- Adaptable to different problems with different physics.
  - *Integration into Pyre framework will eventually allow coupling with other codes in the framework.*
  - *Modular design allows easy extensibility.*

User Feedback

- Bug reports and feature requests:

  http://www.geodynamics.org/roundup

- Please provide feedback on the user manual.

Primary Code Features

Feature	TECTON	LithoMop	PyLith 0.8	PyLith 1.0
Faults	Split/Slippery	Split/Slippery	Split/Slippery	Cohesive/ Discontinuous
Solution	Quasi-static	Quasi-static	Quasi-static	Quasi-static/ Dynamic
Element types	Hex	10 types	Tet/Hex	Many
Complex rheologies	Limited	No	Planned	Planned
Pyrization	No	Limited	Limited	Full
Parallel	No	No	Yes	Planned
Large strain	Yes	No	No	Planned
Gravitational prestresses	Limited	Yes	Yes	Planned
Non-Cartesian geometry	No	No	No	Planned

Primary Advantages of PyLith 1.0

- Full Pyrization:
  - *Code coupling abilities and use of Simulation Controller.*
  - *Powerful scripting abilities and runtime component replacement.*
- Dramatically improved input facilities.
  - *Much better integration with meshing packages.*
  - *Spatial database to specify material properties and BC.*
- Solution of both quasi-static and dynamic problems.
- Better use of Sieve and Sieve data structures.
  - *Faster code development.*
  - *More efficient and scalable code.*

How to Best Allocate Our Resources?

- Bug fixes, testing, and addition of missing features for PyLith 0.8.
  - *Make sure that all existing features work properly.*
  - *Add hex elements, slippery nodes, traction BC, more material models, large deformation, etc.?*
- Development of PyLith 1.0.
  - *Initial version could be released in late 2006/early 2007, depending on how much time is devoted to development.*

Project Support and Further Information

- SCEC
  - *Fault systems workshops, initial funding for code development.*
- GeoFramework project
  - *Development of Pyre/Pythia framework.*
- NSF ITR program
  - *Funding for code development.*
- Computational Infrastructure for Geodynamics CIG
  - *Infrastructure and code development assistance.*
- Further information: www.geodynamics.org.