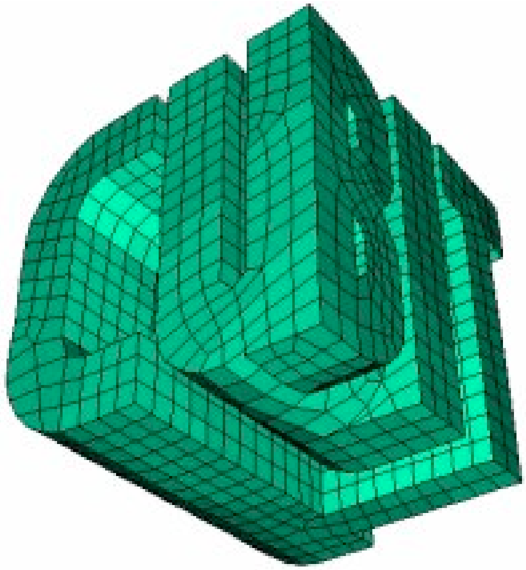
Introduction to CUBIT
A quick tutorial and some simple examples

2009 Numerical Modeling of Crustal Deformation and Earthquake Faulting Workshop - Golden, CO

June 22-26



Sandia National Laboratories

Cornell University

Why use CUBIT?

- Easy connection to Pylith
- Graphical interface and/or scripting (including python)
- Variety of meshing types and approaches

- Platforms
  - Linux RedHat 9.0 32- and 64-bit
  - Windows 2000/XP
  - Mac OS X

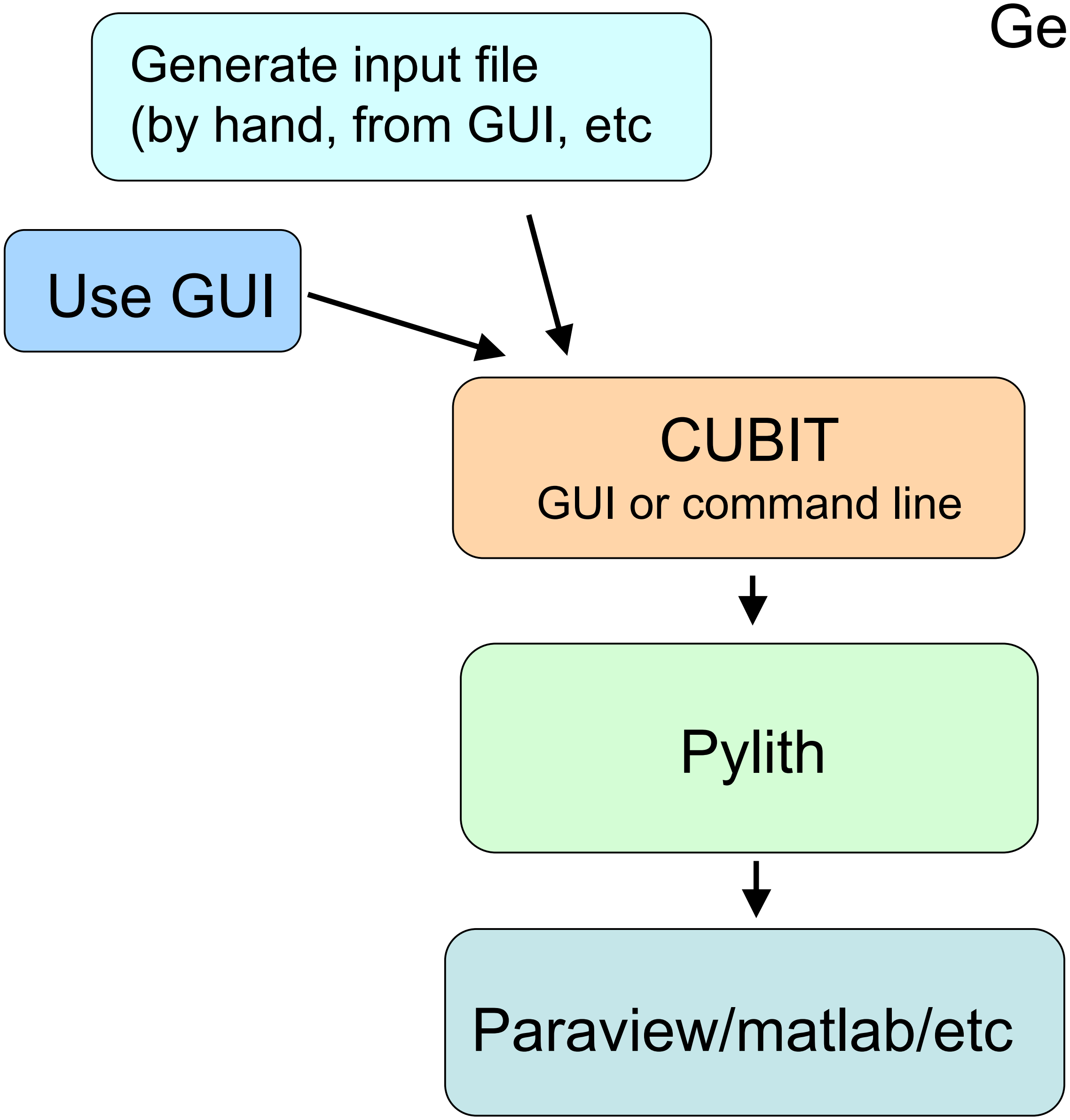
- http://malla.sandia.gov/cubit/index.html
  - $300, downloads and updates/support for 5 years

- Acknowledgements/other resources
  - Last year - Emanuele Casarotti
    - now at Istituto Nazionale di Geofisica e Vulcanologia
  - Online CUBIT info
    - http://cubit.sandia.gov
    - majordomo@scico.sandia.gov
    - Documentation, tutorials (online and ppt w/ auxiliary files)
  - Other NMCDEF participants

Examples from today:

-> Short-Term Crustal Dynamics
->Work Area
->Benchmarks
-> CUBIT examples

Pylith Workflow



Generate input file
(by hand, from GUI, etc

Use GUI

CUBIT
GUI or command line

Pylith

Paraview/matlab/etc

Geological info

- low (toy model)
- high (topo, digitized surfaces, etc)

Outline

- Our specific application / why we like CUBIT
- Walk-through interface features
  - Types of entities and meshing
  - How to built things
  - How to find help
- Examples
  - Mostly simple, but a couple more complicated ones from Emanuel Casarotti (building a subduction zone, loading topography)

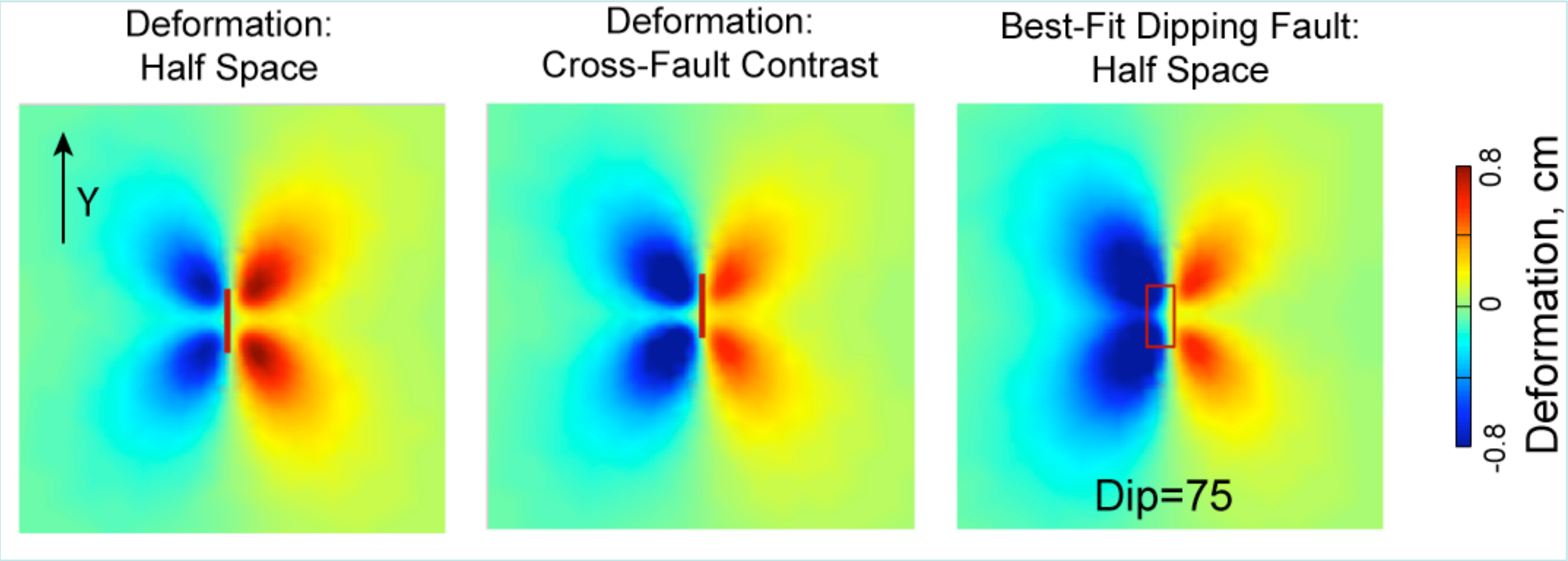
Cornell University

Our Implementations

Need: scriptable, flexible meshing approach requiring minimal interaction for large numbers of runs with slightly different conditions

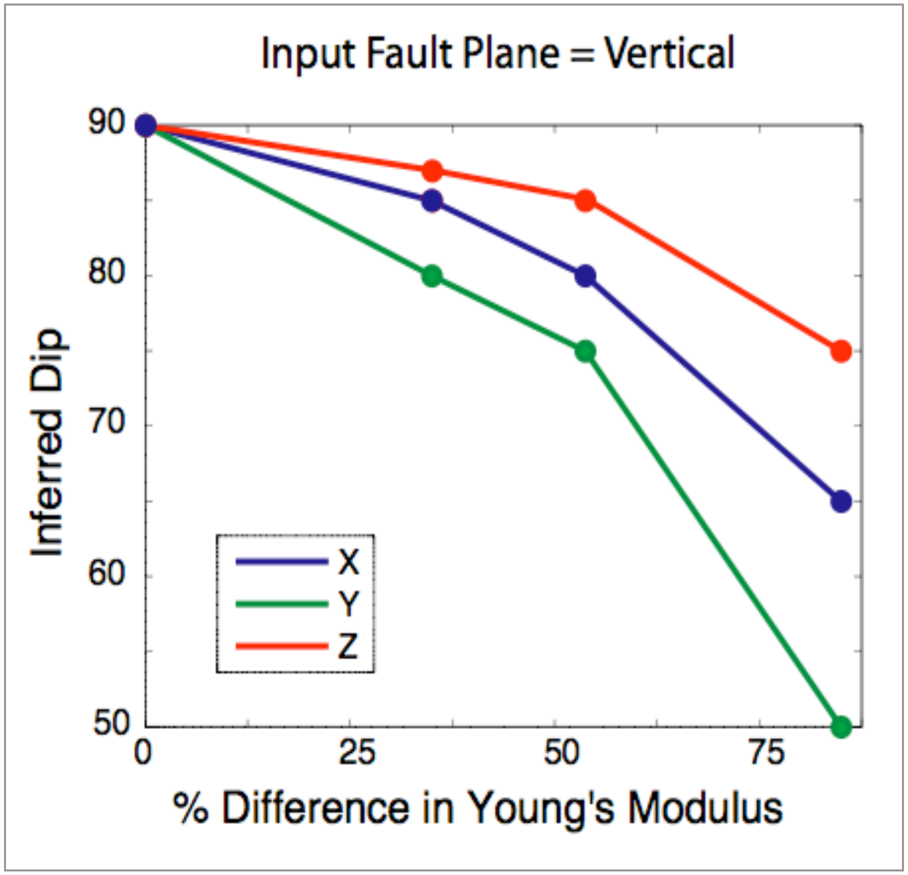
- Green's functions for fault slip inversions
  - Requires BIG mesh or semi-autonomous generation of meshes for each fault patch
- Assessing effects of crustal rigidity variations
  - Need to model faults with range of orientations, depths, etc.
  - Must worry about features introduced by inadequate meshing

Sensitivity test: Strike-slip



Can't fit asymmetric deformation with vertical fault

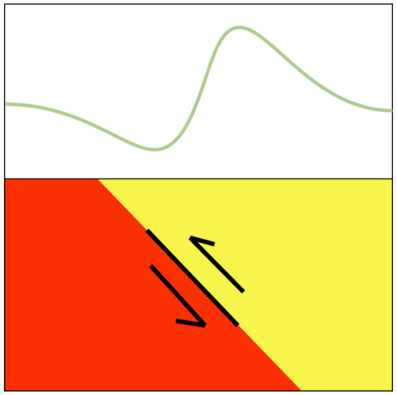
Cross-Fault Contrast Tests

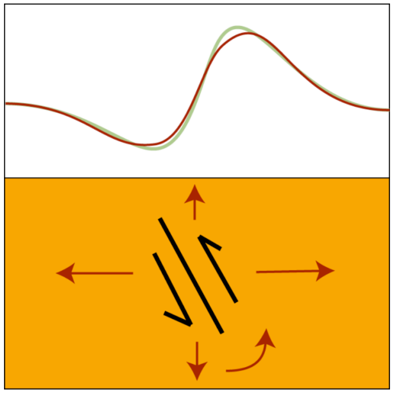


- Retrieve input geometry when contrast=0
- Sensitivity depends on viewing and earthquake geometry

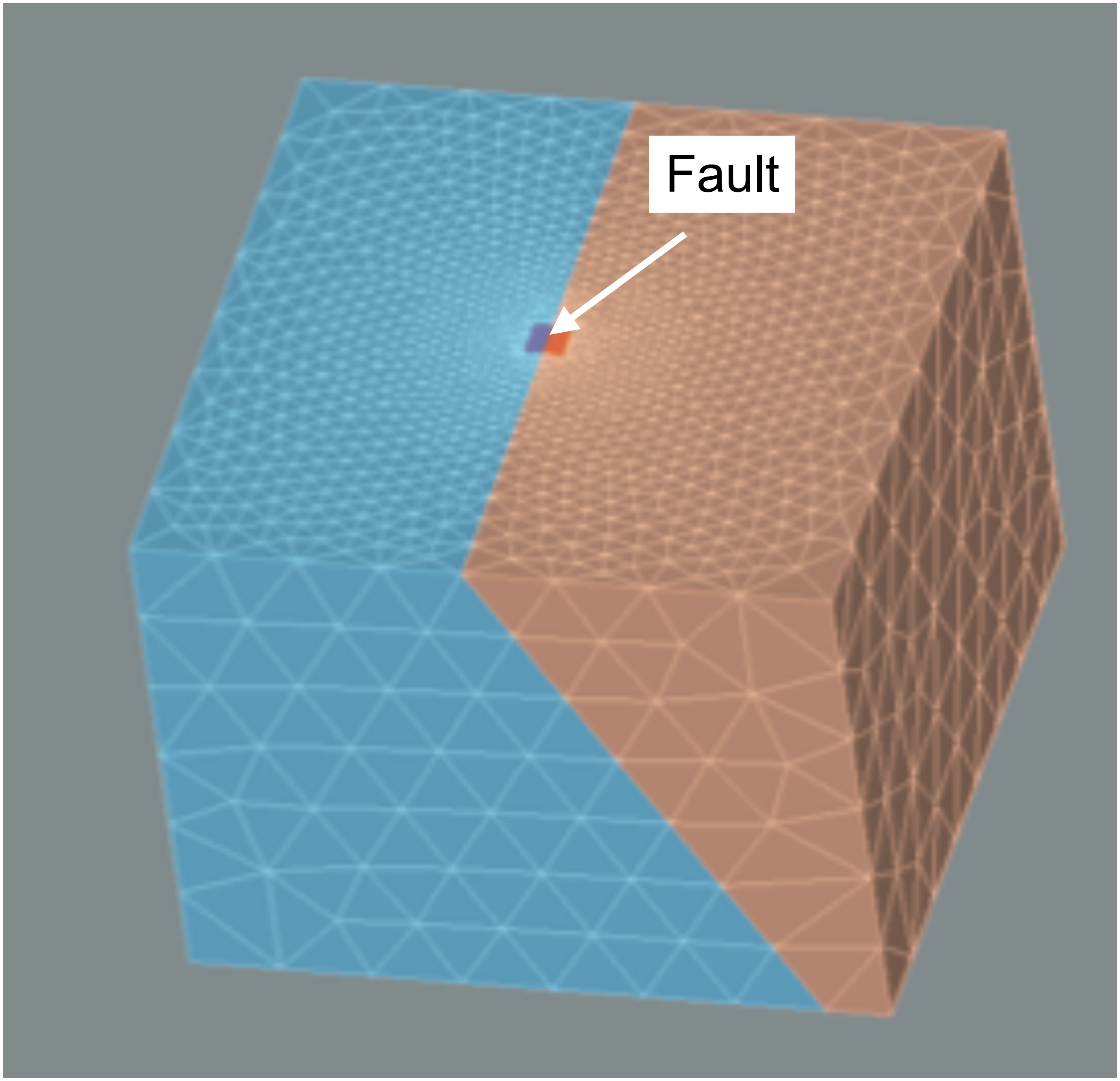
Examples: Sensitivity Tests

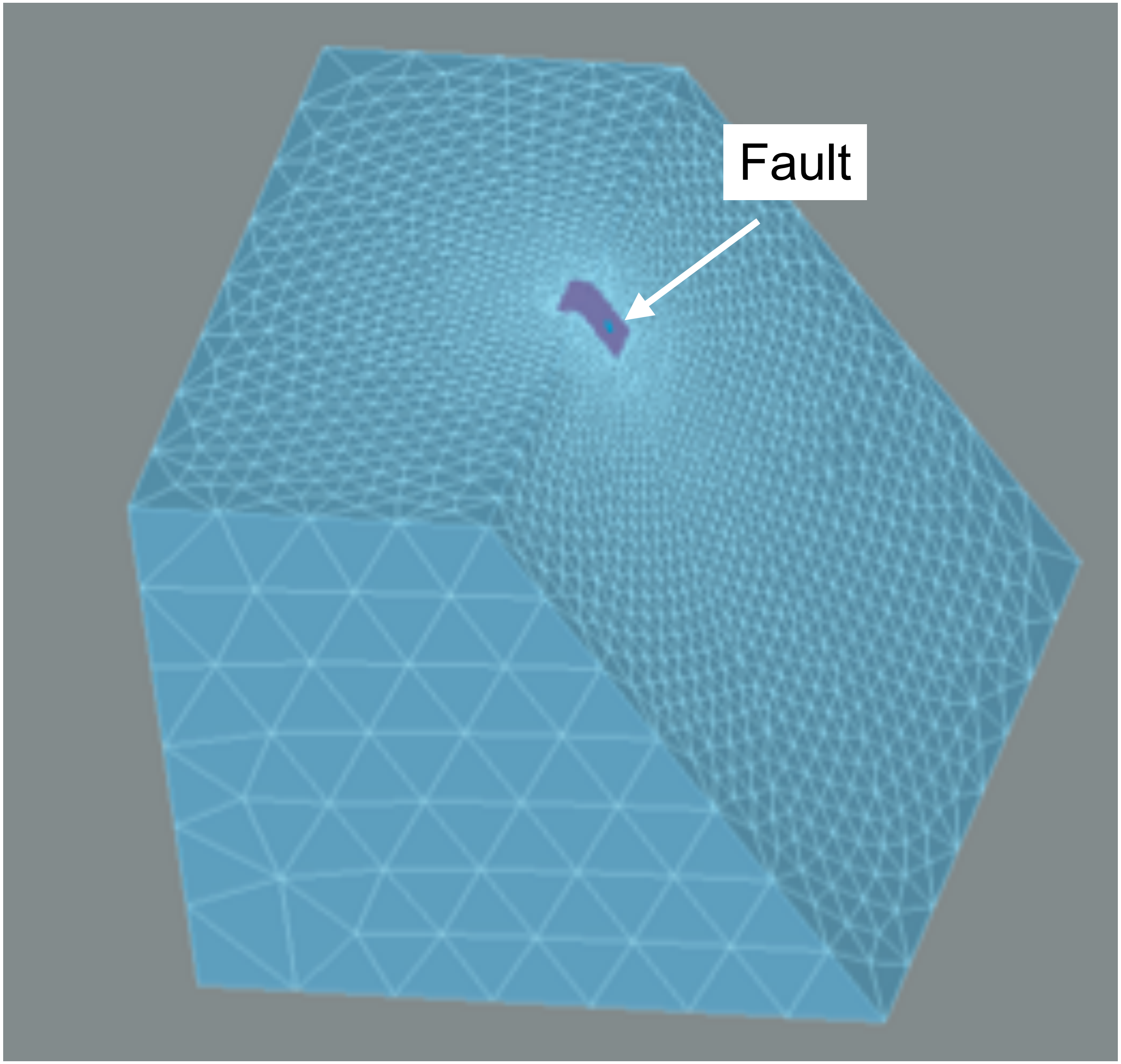
- Goal: For generic settings, what is inversion sensitivity?
  - Generate synthetic data using cross-fault contrast (slow)
  - Invert using elastic half space (fast)
  - Assess potential bias: Inferred fault dip





FE calculations using Pylith, mesh with CUBIT


Fault

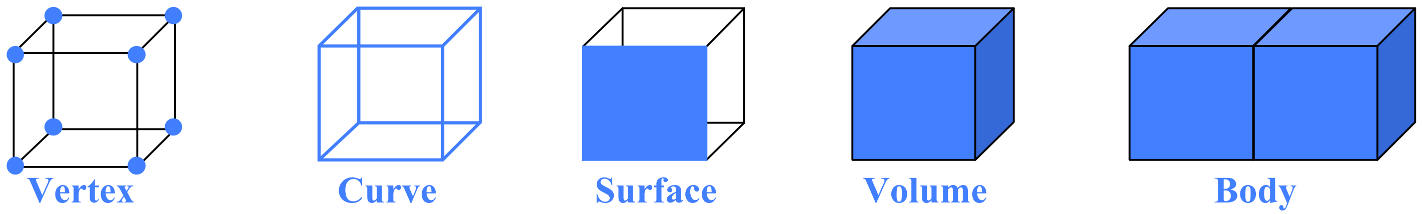

Fault

Using CUBIT

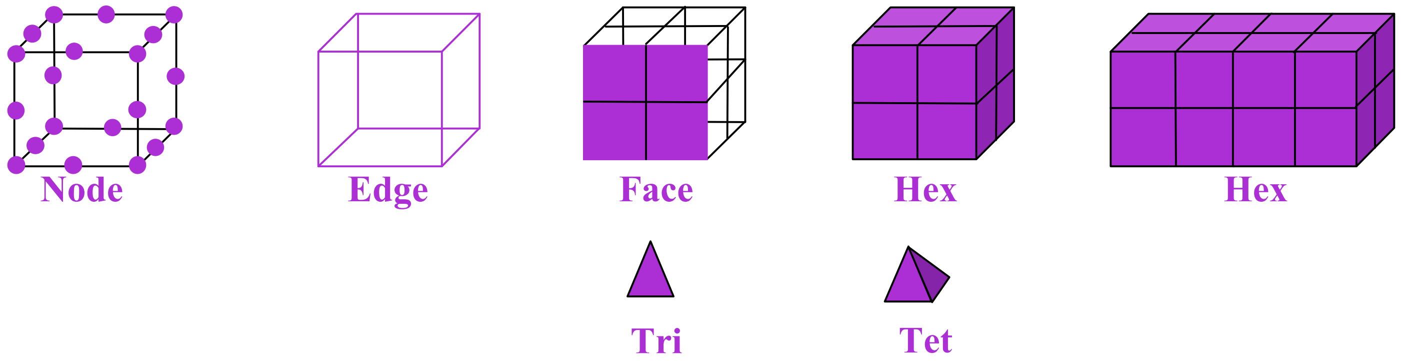
1. Creating the geometry (curve-surface-volume)
2. Setting the mesh interval sizes and meshing schemes
3. Meshing the geometry
4. Specifying the boundary conditions
5. Exporting the mesh

Entity Types in CUBIT

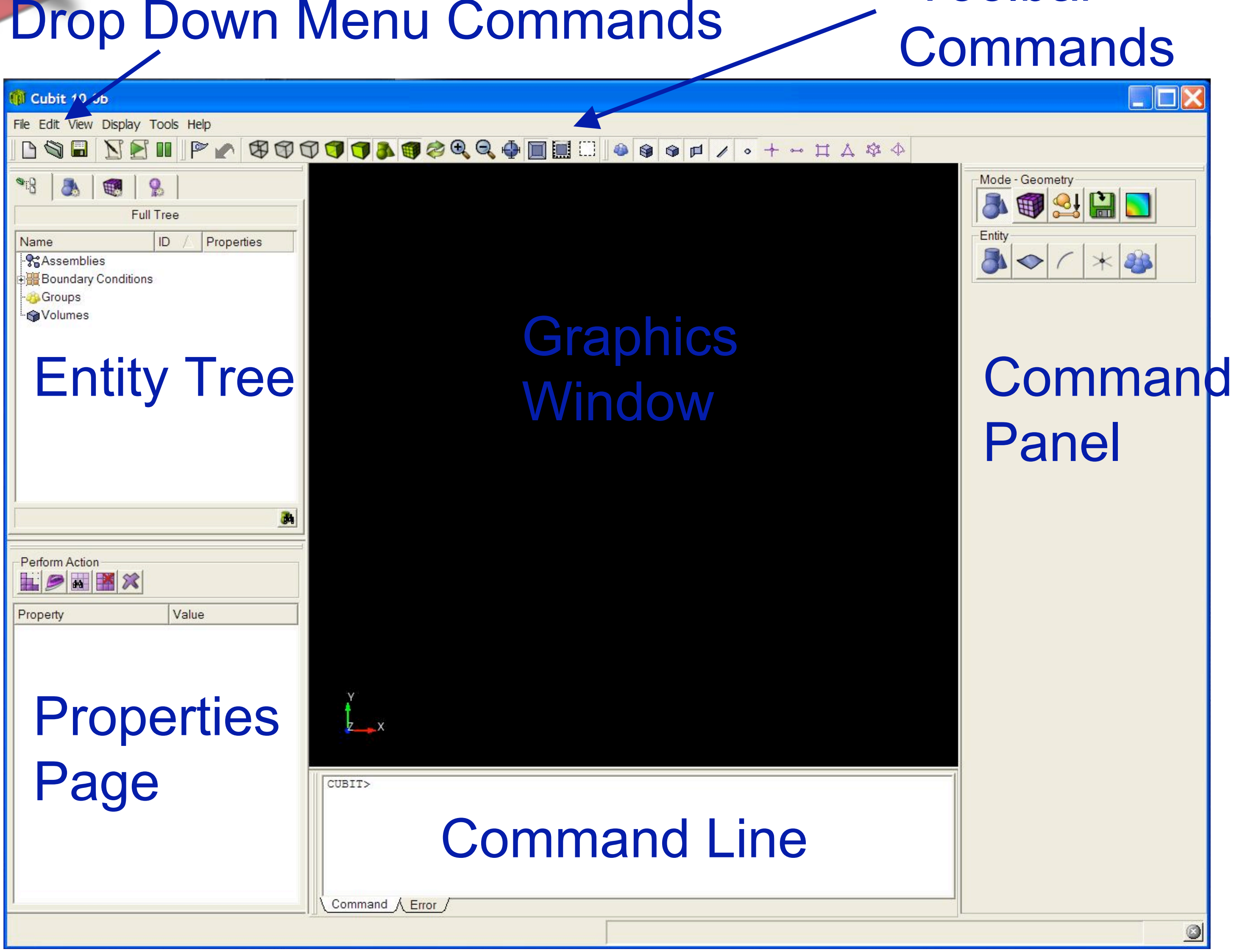
Geometry Entities in CUBIT



Mesh Entities, which approximate geometry entities of same dimension



CUBIT Meshes Vertices First, Then Curves, Then Surfaces, Then Volumes (Advancing Front Paradigm)



Drop Down Menu Commands

Toolbar Commands

Entity Tree

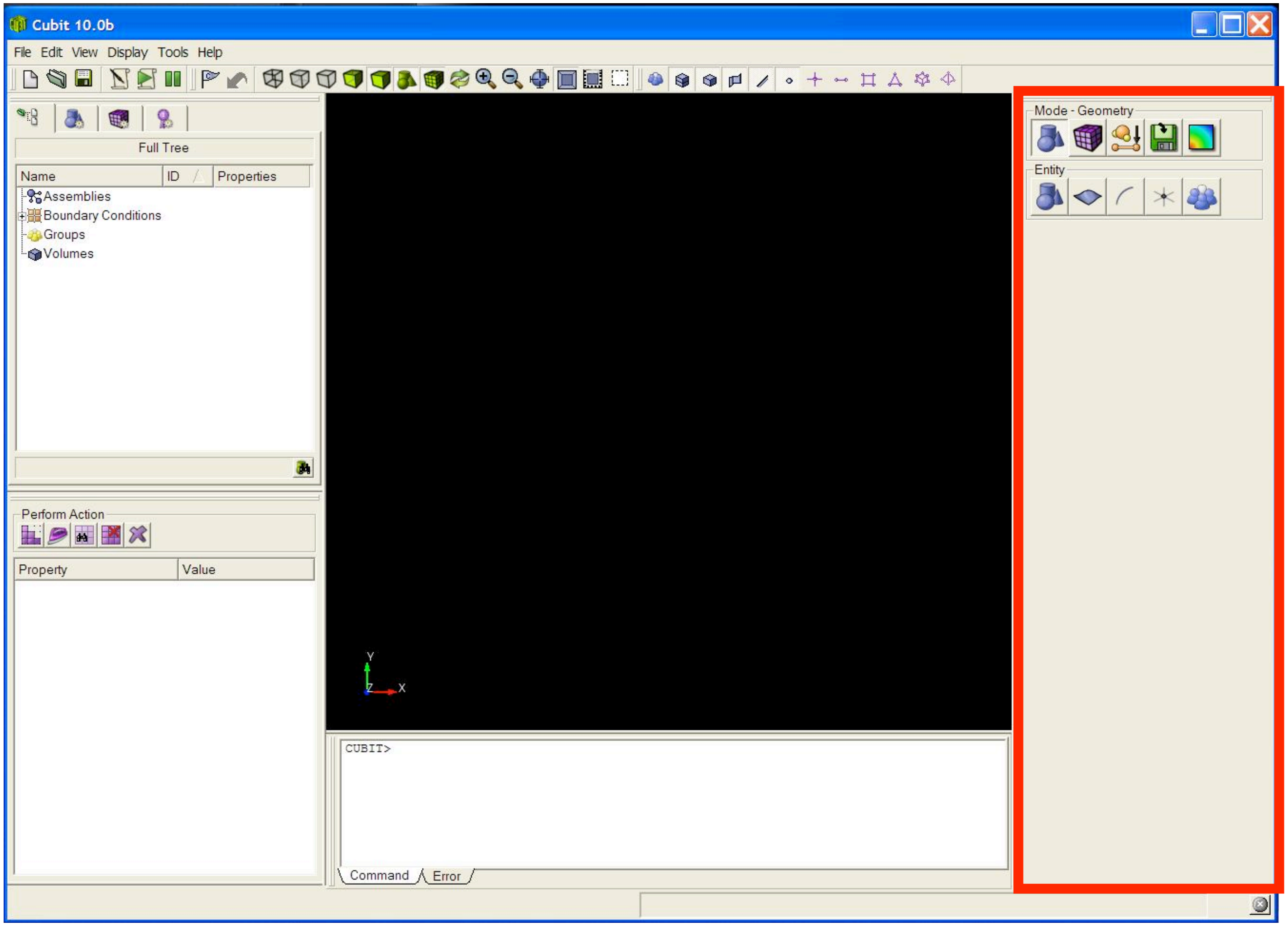
Graphics Window

Command Panel

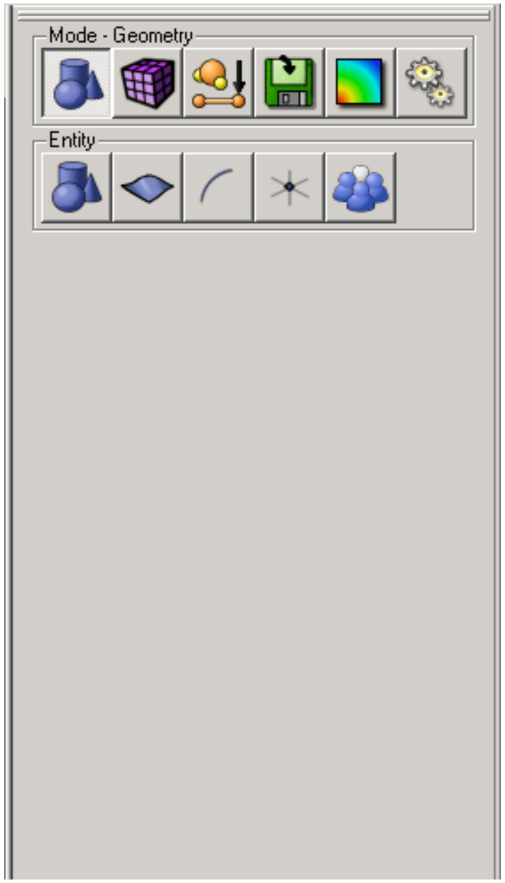
Properties Page

Command Line

The Command Panel



Operation Mode Buttons



Press an Icon to enter a new mode

- Geometry: Create, modify, cleanup…

- Mesh: Intervals, schemes, smoothing…

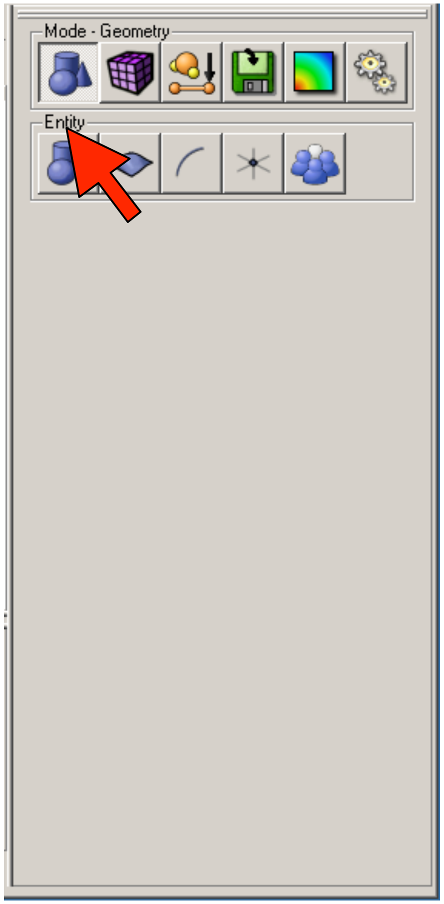
- Properties: Nodesets, sidesets, blocks

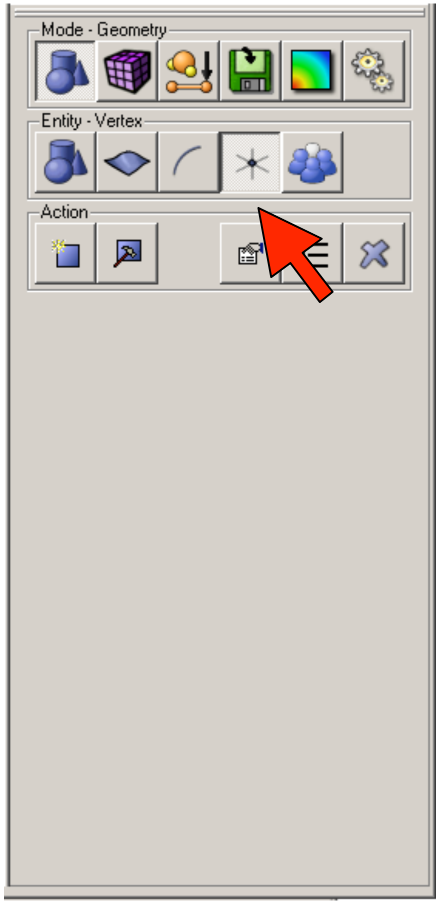
- Analysis Setup: Export mesh

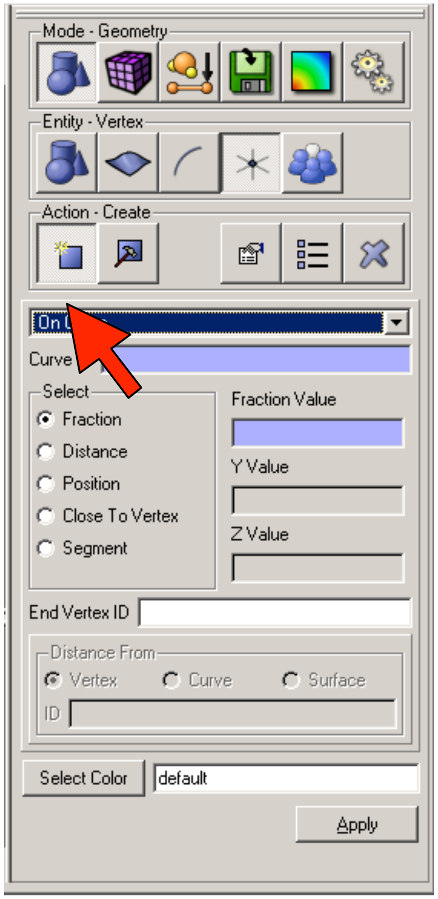
- Post Processing: Customizable shortcut

Operation Mode Buttons

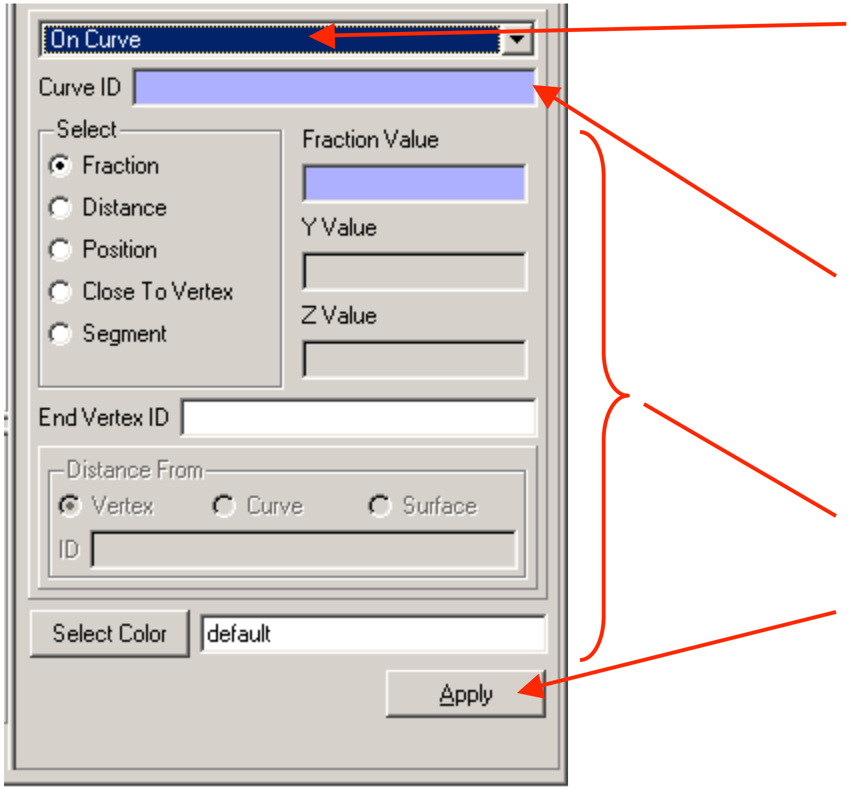
Each Button press takes you to a lower level







Typical Dialog Layout



On Curve

Curve ID

Select: ● Fraction ○ Distance ○ Position ○ Close To Vertex ○ Segment

Fraction Value

Y Value

Z Value

End Vertex ID

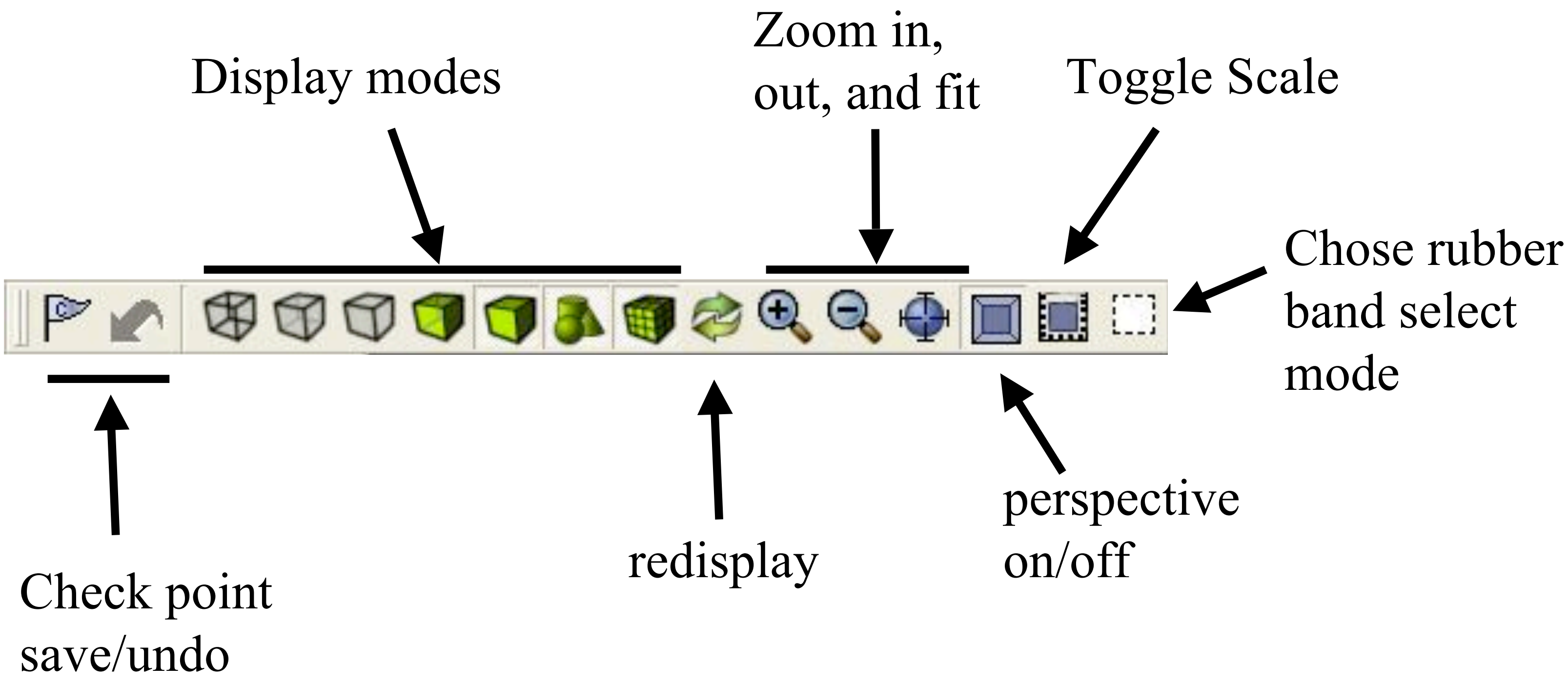
Distance From: ● Vertex ○ Curve ○ Surface

ID

Select Color: default

- **Drop Down Menu**
  - Select the type of operation (sub-action).
- **ID Input Field**
  - You can type IDs here, or fill the box by picking
- **Command Options Input**
- **Execute Button**
  - Click button or hit alt-a to execute the command.

Display Tool Bar



Display modes

Zoom in, out, and fit

Toggle Scale

Chose rubber band select mode

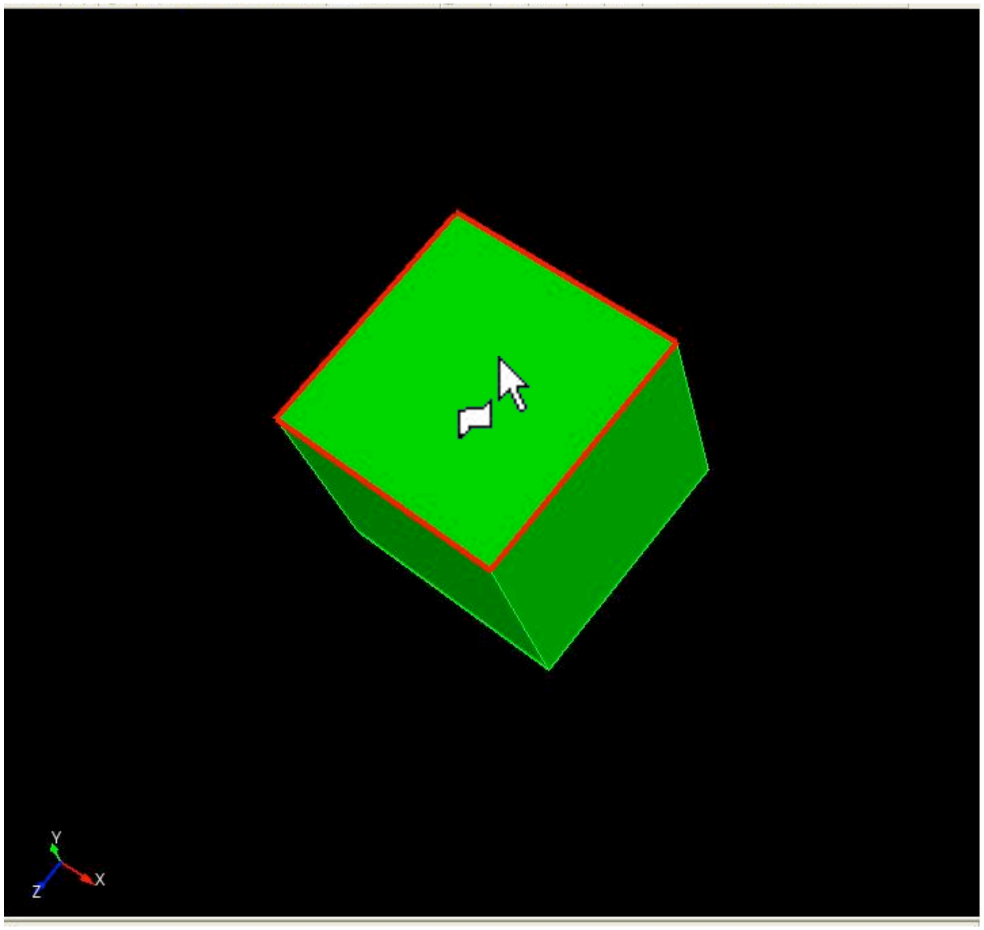
Check point save/undo

redisplay

perspective on/off

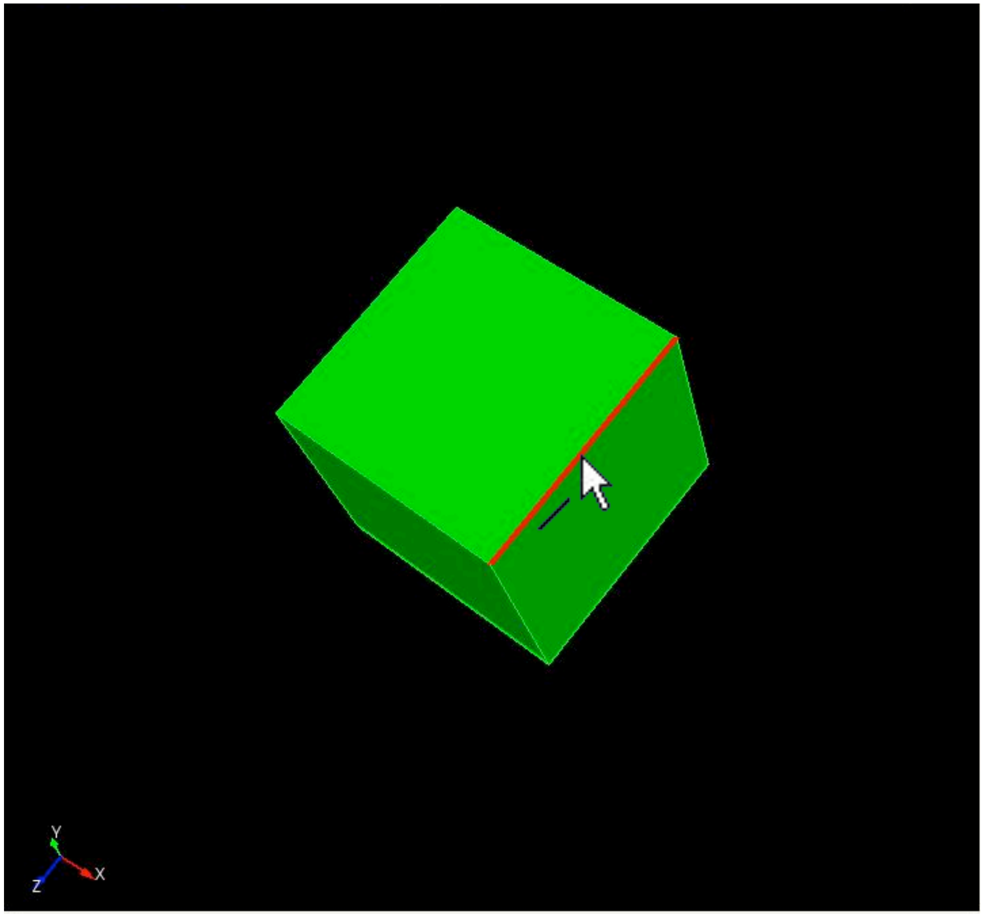
Selecting Surfaces in the Graphics Window

Move cursor to a surface. The bounding curves of the surface are highlighted and cursor indicates surface type.



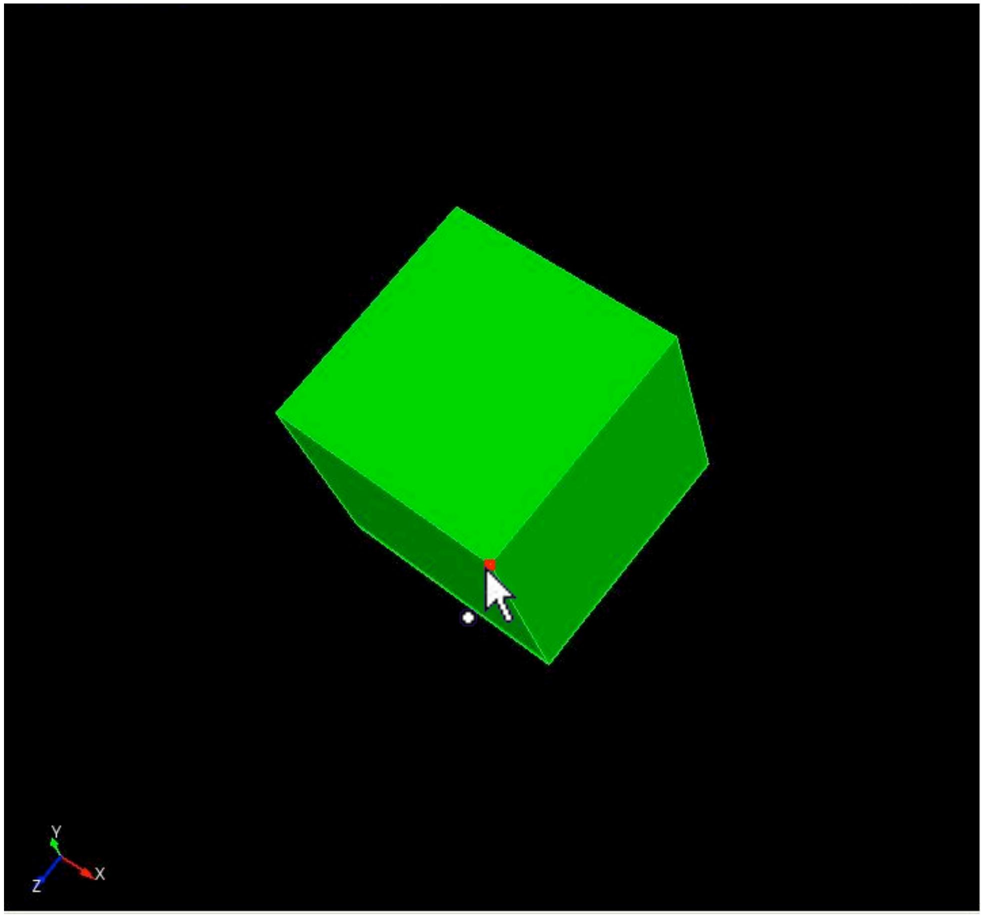
Selecting Curves in the Graphics Window

Move the cursor to a curve. The curve is highlighted and the cursor indicates curve type.

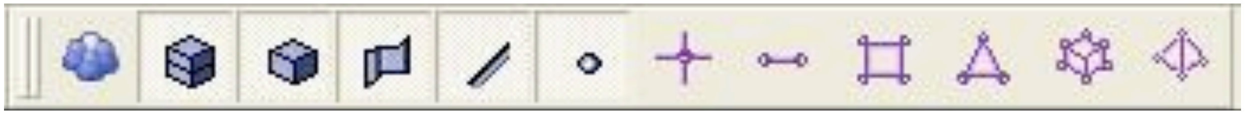


Selecting Vertices in the Graphics Window

Move the cursor to a vertex. The vertex is highlighted and the cursor indicates vertex type.



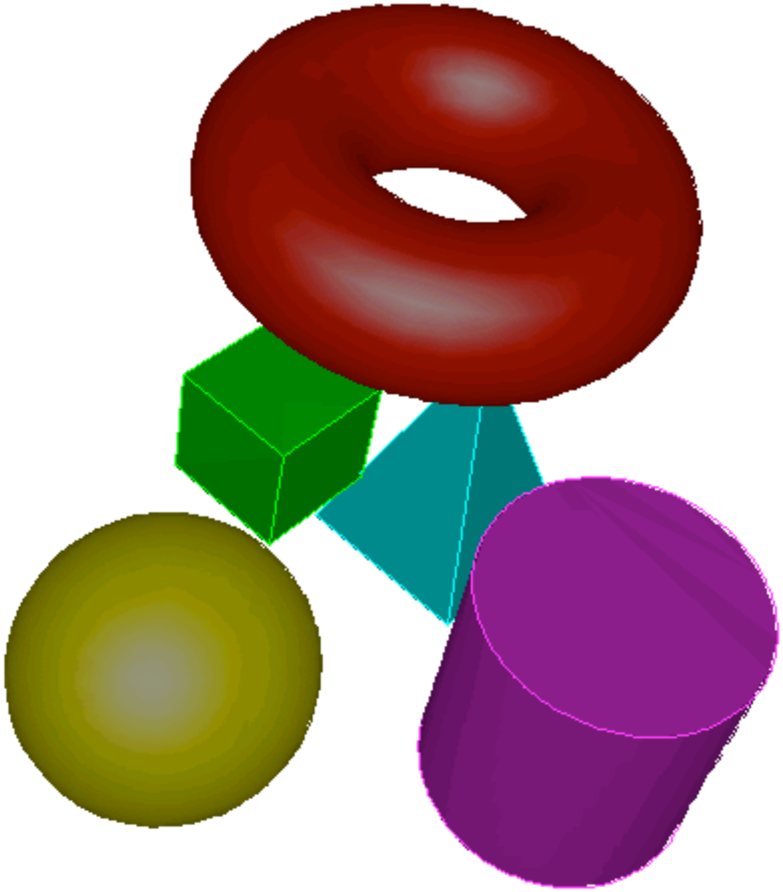
Entity Selection Filter



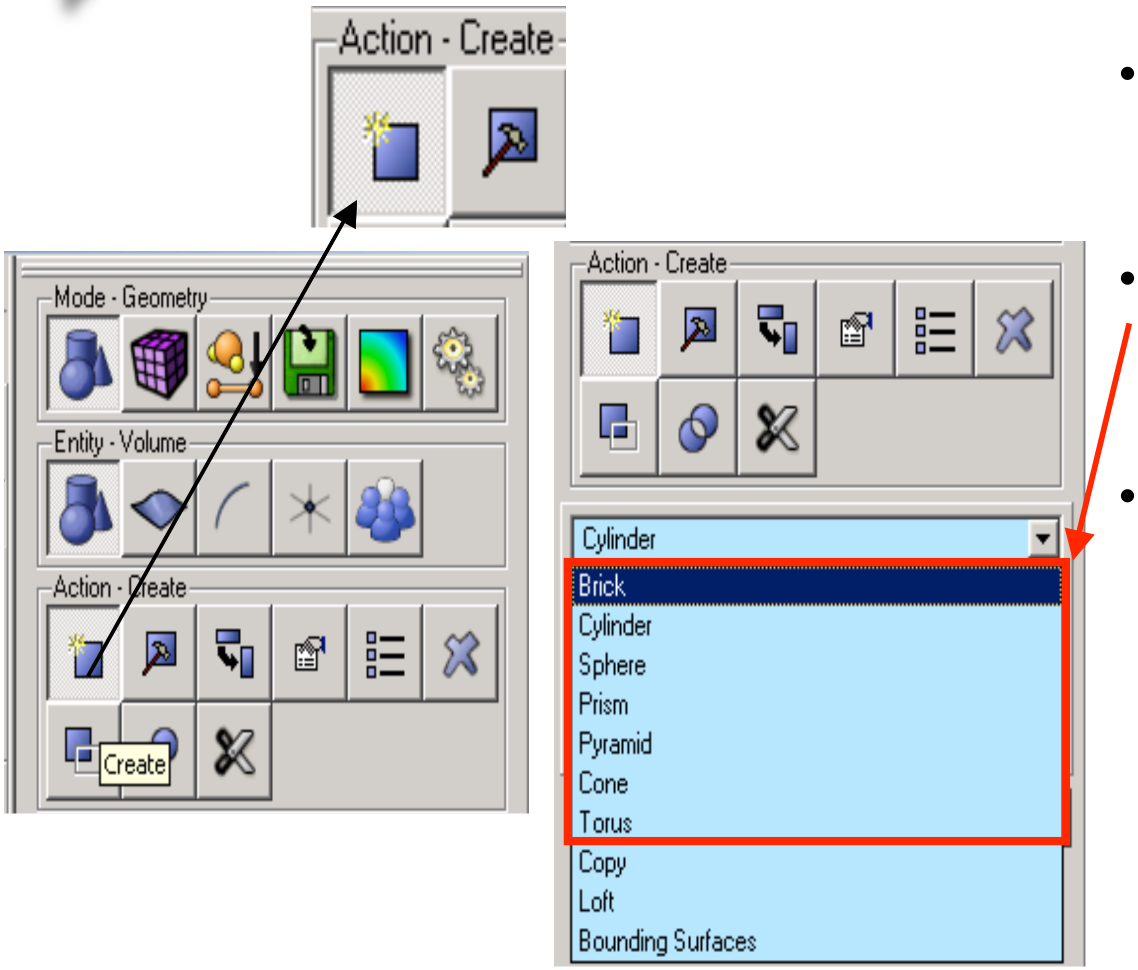
- **Toolbar buttons toggle entity types that will be included in pre-selection**
- **Default**
  - Volume
  - Surface
  - Curve
  - Vertex
- **Active ID Input field “hijacks” pre-selection so that only the expected entity type is selectable**

Geometry Primitive Creation

- Many analytic geometry types may be created in CUBIT
- Useful when creating geometry from scratch, and in decomposition



Create Button



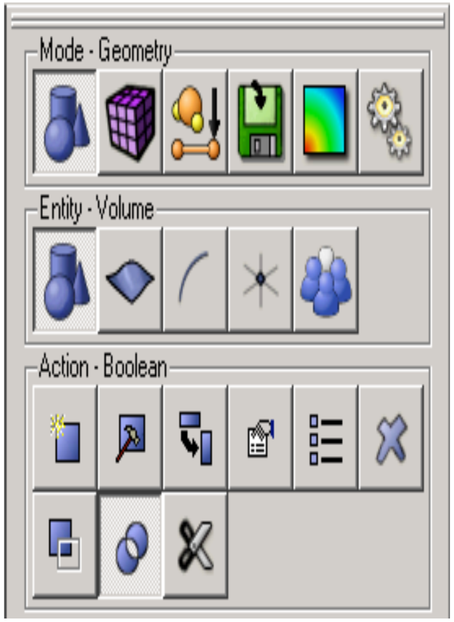
- **Geometry Primitives are accessed with the Create button**
- **Seven primitive types are currently available**
- **For command line syntax:**
  - `CUBIT> help create`

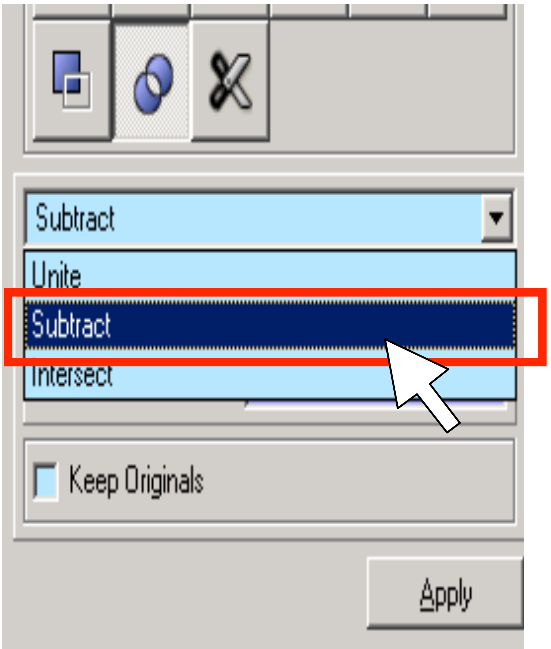
Geometry Booleans

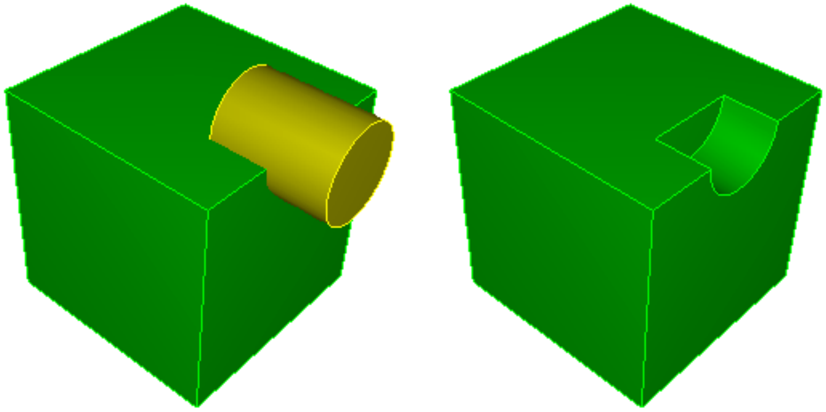
- **Geometry Booleans define the shape of a Body based on overlapping regions**
  - Subtract
    - **Remove regions of overlap**
  - Intersect
    - **Delete all except regions of overlap**
  - Unite
    - **Combine all regions**

Subtract

- **Removes regions that overlap**





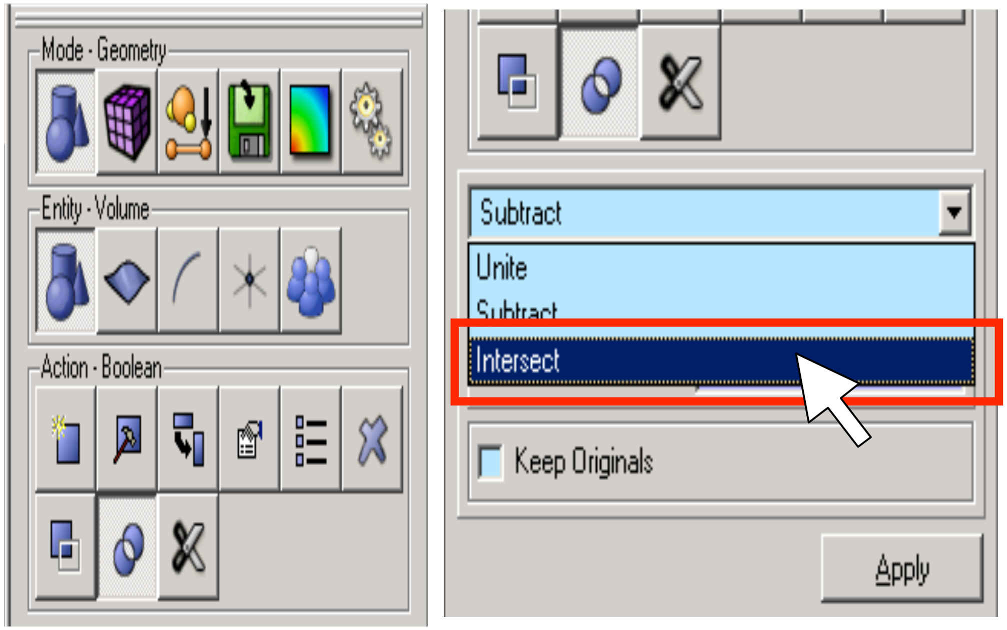


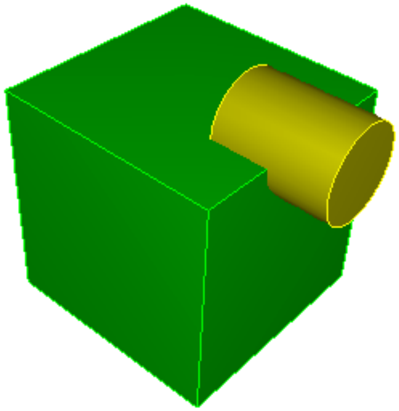
Before After

```
CUBIT> subtract body 2 from 1
```

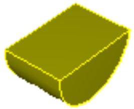
Intersect

- **Removes regions that don't overlap**





Before



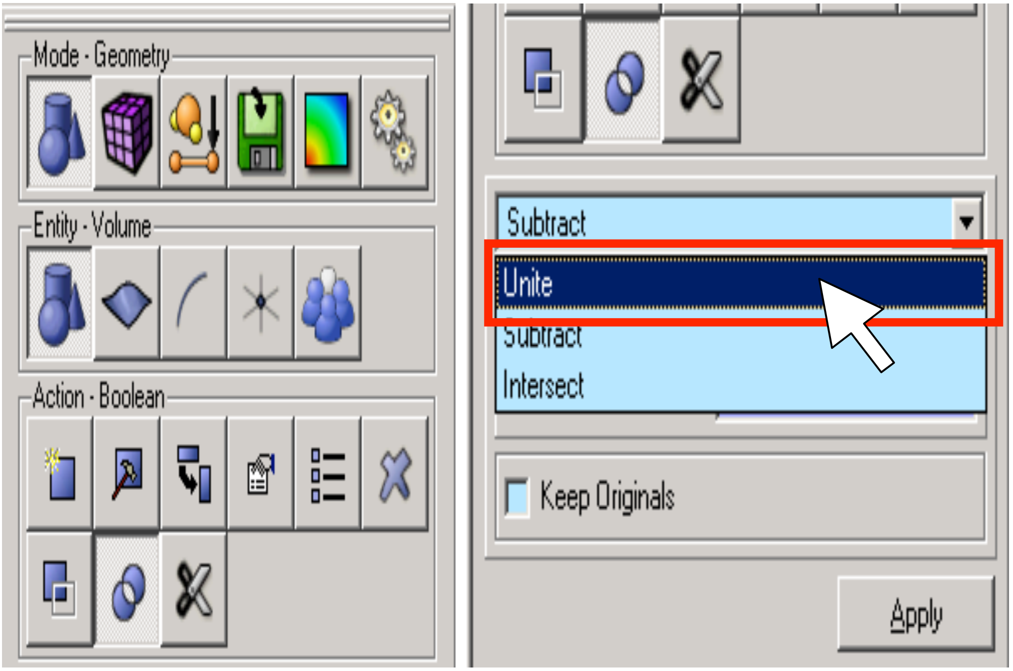
After

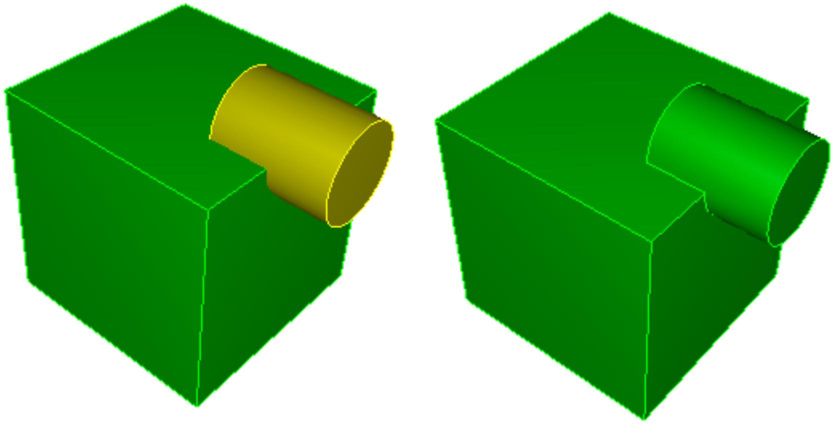
```
CUBIT> intersect body 1 2
```

Unite

Run geom_test

- **Combines all regions into one Body**





```
CUBIT> unite body 1 2
```

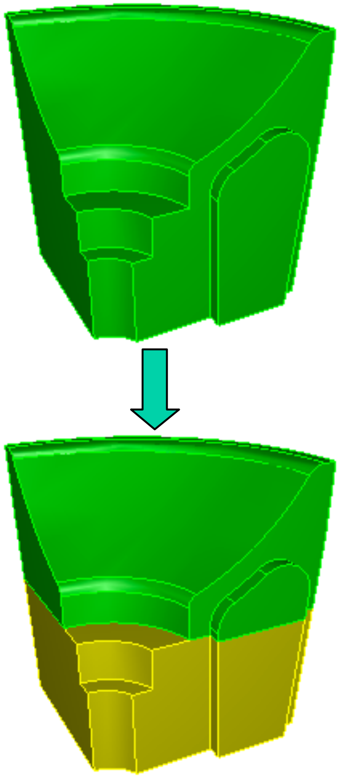
Importing Geometry

- **Previously created geometry may be imported from CAD files**
  - ACIS
  - STEP
  - IGES
  - Pro /E (limited availability)
- **Geometry translators may be used to import unsupported CAD formats**
  - pro2acis

Note: can also use “brute force” and build nodes, surfaces, then volumes

Webcutting

- **Webcutting slices 1 Body into 2 Bodies**
- **Many methods to determine where to make the slice**
  - Plane
  - Cylinder
  - Extended Surface
  - Intersection with “Tool” Body



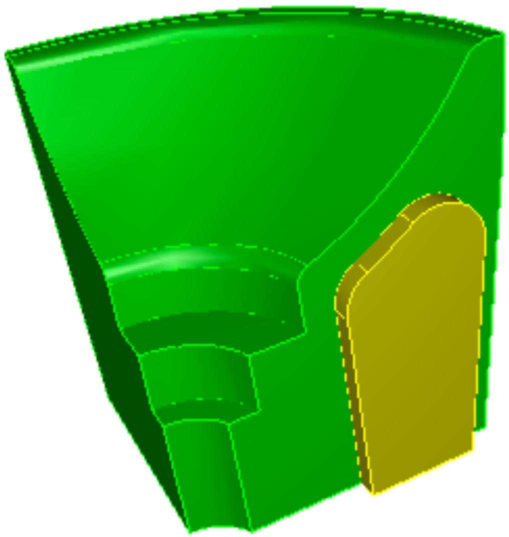
For command line syntax:

```
CUBIT> help webcut
```

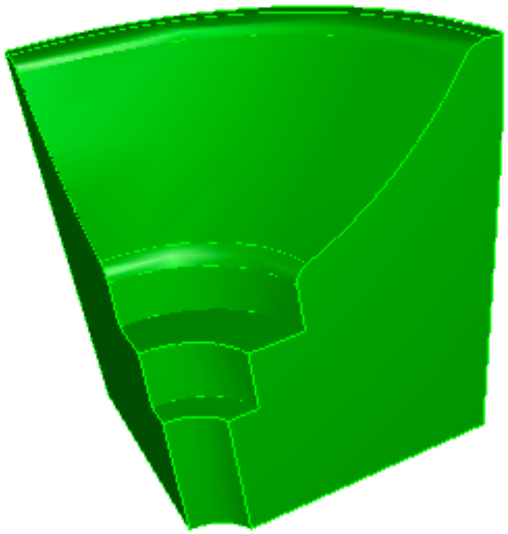
Imprinting

- Modifies a Body based on what it touches
- Splits existing Curves and Surfaces at points of contact
- Imprinting is a necessary step to allow adjacent Bodies to share common boundaries

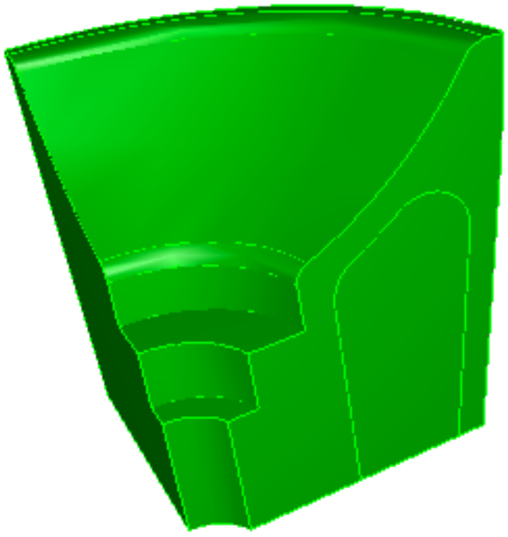
Imprinting



Body 1 and 2



Body 1
before imprinting



Body 1
after imprinting

Merging

- **Adjacent Surfaces, Curves, and Vertices are replaced with a single entity**
- **Merged entities belong to more than one parent**
- **Merging allows mesh to be shared at common boundaries**
  - Otherwise - have two surfaces in the same spot with different names/mesh

Run subduction example

Geometry - My notes

- Can select multiple entities at once in many ways
  - `draw volume all with x_coord > 0`
  - `curve all in volume 1 3 5 visibility off`
- Use tree view and info panel to find names/numbers/geometrical information (or python)

- Make sure to:
  - "reset" between tests/runs
  - Merge/Imprint all entities before meshing!

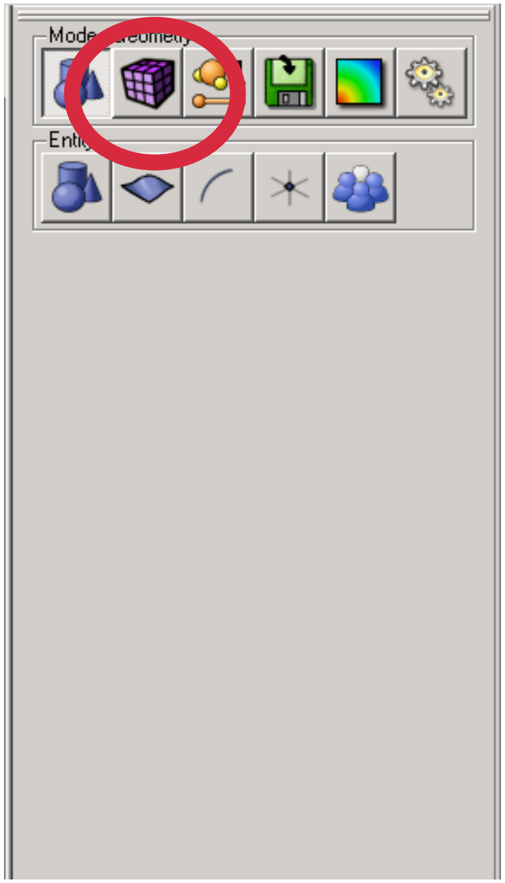
- Everything done in GUI shows up in command pane and history - save in journal file to repeat without getting carpal tunnel

- Bringing in points individually - not a pain if using a journal file/scripting

Meshing Notes

- Start with mesh node spacing curves and build up to volumes
  - Set mesh spacing, then "scheme", then apply meshing
- Can build tet or hex meshes
- Usually requires some iteration at first to find what works best
- Symmetrical volumes - form mesh on one surface and "sweep" around to rest of volume

Operation Mode Buttons



- Geometry: Create, modify, cleanup…

- Mesh: Intervals, schemes, smoothing…

- Properties: Nodesets, sidesets, blocks

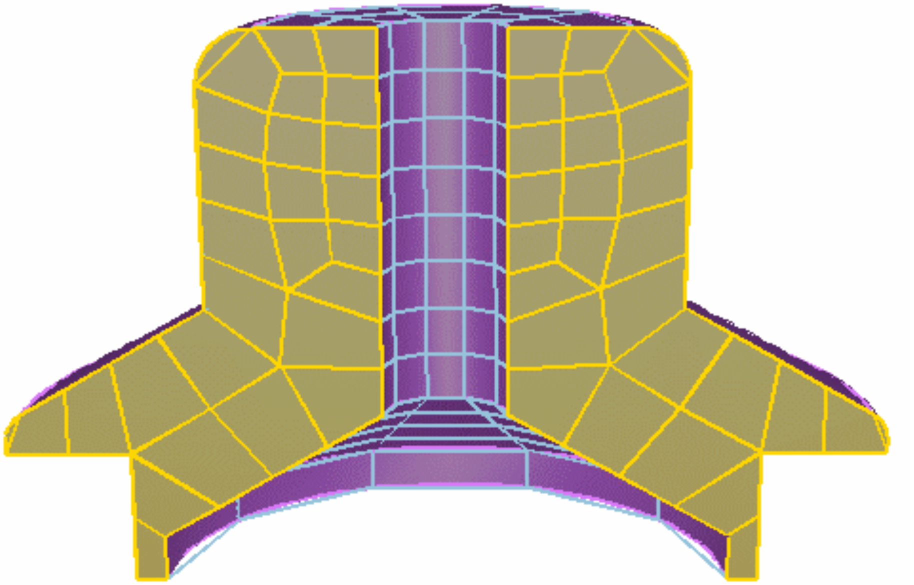
- Analysis Setup: Export mesh

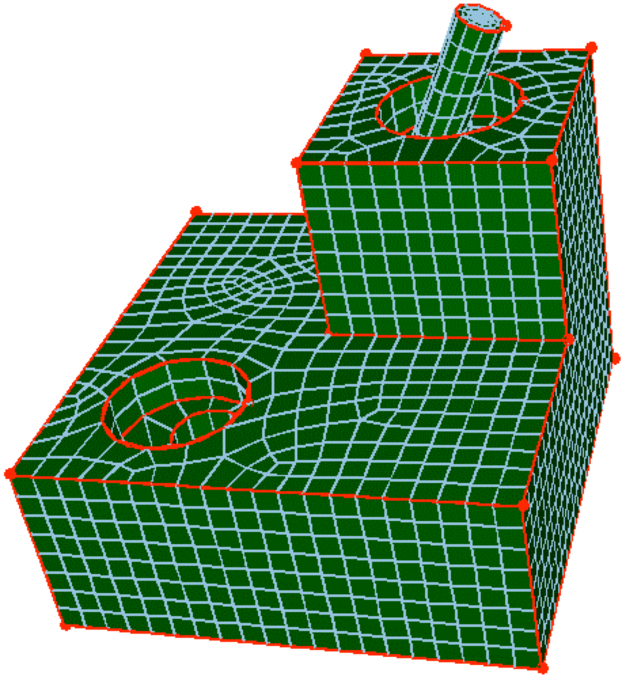
- Post Processing: Customizable shortcut

Sweep

Run mesh example

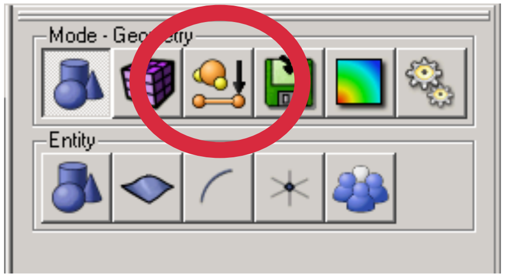
- **2.5d - may twist & turn**





Building Groups for Pylith

- Blocks = materials
- Nodesets = surfaces for boundary conditions

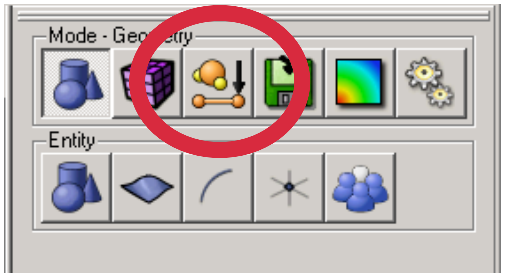


In CUBIT:

```
block 1 volume foot_inner
block 1 volume 1 to 8
block 1 name "foot_walls"
```

Building Groups for Pylith

- Blocks = materials
- Nodesets = surfaces for boundary conditions

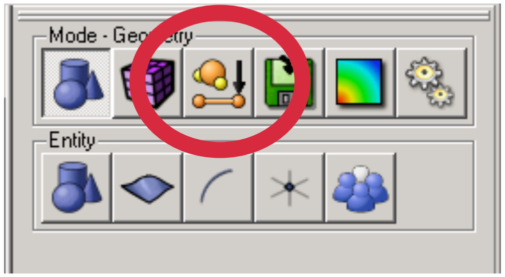


In CUBIT:

```
 group "fault" add node in fault_inner
nodeset 10 group fault
nodeset 10 name "fault"
```

Building Groups for Pylith

- Blocks = materials
- Nodesets = surfaces for boundary conditions



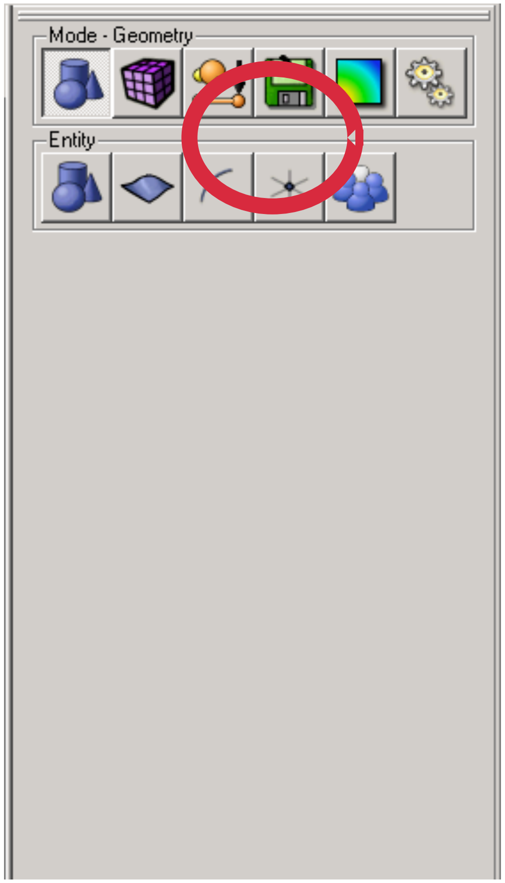
In dislocation.cfg (boundary conditions):

```
[pylithapp.timedependent.bc.x_neg]
fixed_dof = [0]
label = 12
db.label = Dirichlet BC on -x
```

Exporting to Pylith

- Export mesh with:

```
export mesh "out.exo" dimension 3
```



- In pylithapp.cfg

```
reader = pylith.meshio.MeshIOCubit

[pylithapp.mesh_generator.reader]
filename = out.exo
```

Summary/putting all together

- Bring in points from faults, topo, etc
- Build bodies that describe desired sceario
- Mesh (other refinement tools?)
- Define all sets of nodes (boundary conditions) and tets/hex (materials)
- Export to “myname”.exo
- Use in Pylith