Crustal Deformation Modeling Tutorial

Overview of CUBIT

Brad Aagaard
Rowena Lohman
Charles Williams
Matthew Knepley

CIG COMPUTATIONAL INFRASTRUCTURE for GEODYNAMICS

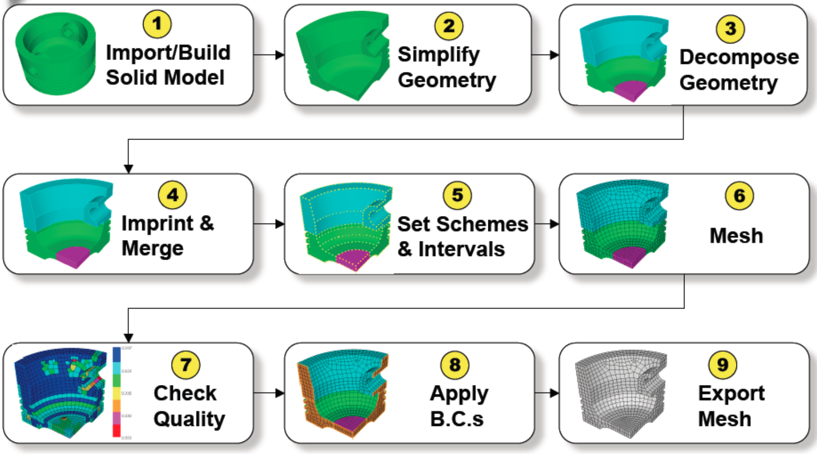
June 18, 2012

CUBIT, `cubit.sandia.gov`

Finite-element mesh generator

- Developed by Sandia National Laboratory
  - GUI interface reduces learning curve
  - Relatively inexpensive
  - Scripting via Python and journal files
  - Trial version available from `www.csimsoft.com`
- Geometry engine
  - Construct domain geometry using simple, yet powerful tools
  - Spline surfaces and curves
  - Some support for mesh surfaces (e.g., triangular facets)
- 2-D and 3-D structured and unstructured meshes
  - Triangular cells
  - Quadrilateral cells
  - Tetrehedral cells
  - Hexahedral cells

The Basic CUBIT Process



1 Import/Build Solid Model

2 Simplify Geometry

3 Decompose Geometry

4 Imprint & Merge

5 Set Schemes & Intervals

6 Mesh

7 Check Quality

8 Apply B.C.s

9 Export Mesh

Generating a Mesh

General workflow for using CUBIT with PyLith

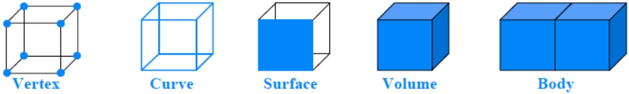
1. Create the geometry
    - Build geometry from bottom-up (points-curves-surfaces-volumes)
    - Import geometry from other geologic models
2. Set the discretization size meshing scheme
3. Mesh the geometry
4. Specify the boundary conditions and materials
5. Export the mesh

Using CUBIT

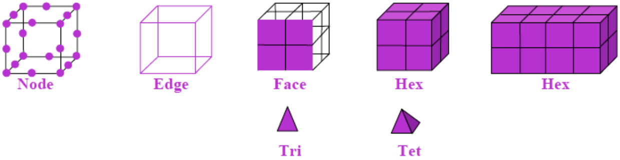
- GUI
  - Interactive mesh construction
  - Constructing journal files and Python scripts
- Command line
  - Can be run in batch mode and with or without graphics
  - Useful for generating large and/or complex meshes
- Journal files and APREPRO
  - Enables scripting with variables and units
  - Limited to CUBIT commands and a few simple functions
- Python scripts
  - Generate CUBIT commands
  - Full capabilities of Python programming language
  - Restricted to Python interpreter distributed with CUBIT

Entity Types in CUBIT

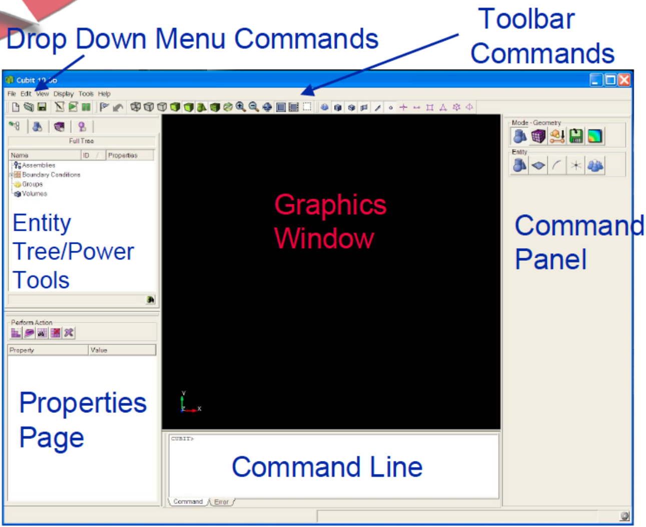
Geometry Entities in CUBIT



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/Power Tools

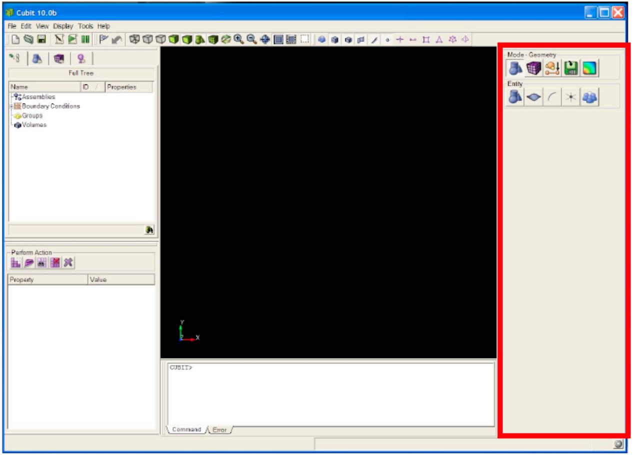
Graphics Window

Command Panel

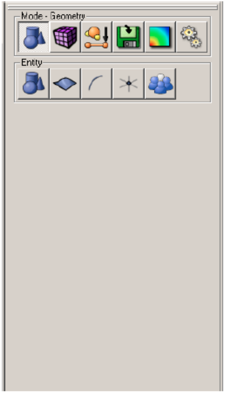
Properties Page

Command Line

The Command Panel



Operation Mode Buttons



Press an Icon to enter a new mode

- Geometry: Create, modify, cleanup…

- Mesh: Intervals, schemes, smoothing…

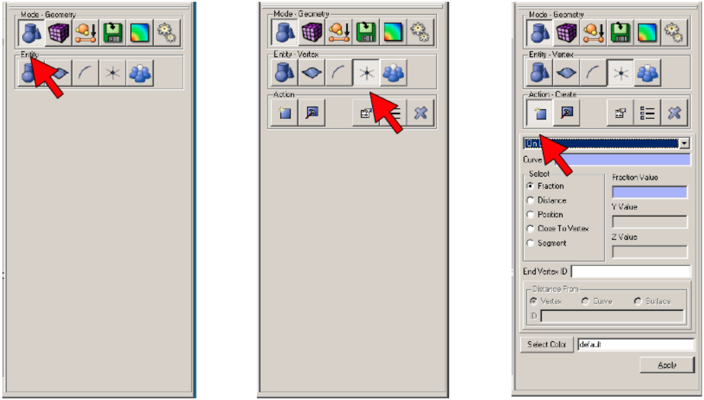
- Properties: Nodesets, sidesets, blocks

- Analysis Setup: Export mesh

- Post Processing: Customizable shortcut

Operation Mode Buttons

Each Button press takes you to a lower level



On

Curve

Select

● Fraction

○ Distance

○ Position

○ Close To Vertex

○ Segment

Fraction Value

Y Value

Z Value

End Vertex ID

Distance From

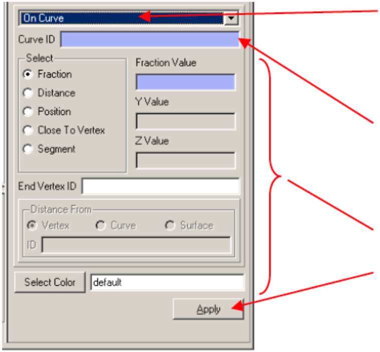
● Vertex ○ Curve ○ Surface

ID

Select Color default

Apply

Typical Dialog Layout



On Curve

Curve ID

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

Fraction Value

Y Value

Z Value

End Vertex ID

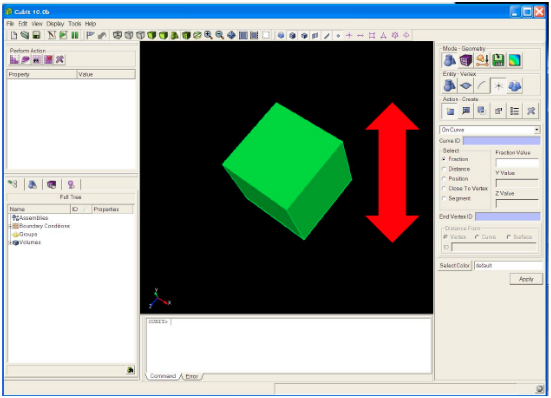
Distance From: ● Vertex ○ Curve ○ Surface

ID

Select Color: default

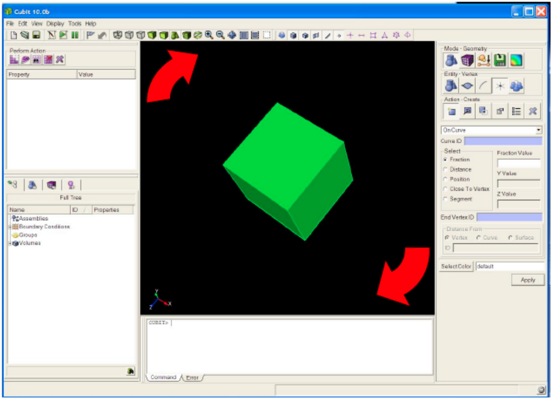
- **Drop Down Menu**
  - Select the type of operation (sub-action).
- **ID Input Field**
  - You can type IDs here, or fill the box by picking
- **Command Options Input**
- **Execute Button**
  - Click button or hit alt-a to execute the command.

Zooming



To zoom in and out, move the mouse into the graphics window, hold the right mouse button down, and move the mouse pointer vertically.

Rotate



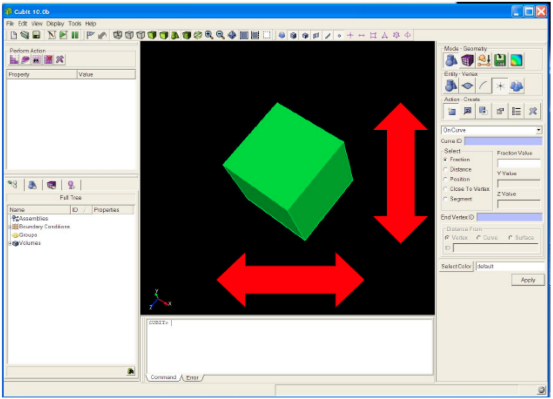
To rotate the model about an axis normal to the screen, move the mouse near the edge of the graphics window, hold the middle mouse button down, and move the mouse pointer along the edge of the window

LOCKHEED MARTIN

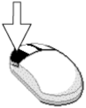
Sandia National Laboratories

CIG COMPUTATIONAL INFRASTRUCTURE for GEODYNAMICS

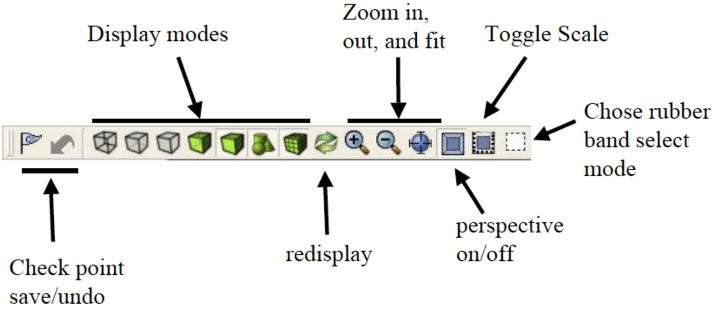
Panning



To pan, move the mouse into the graphics window, hold the left mouse button down, and move the mouse pointer horizontally or vertically.



Display Tool Bar


Display modes

Zoom in, out, and fit

Toggle Scale

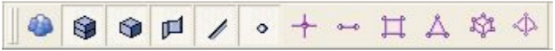
Chose rubber band select mode

Check point save/undo

redisplay

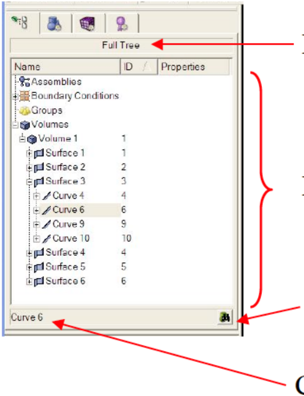
perspective on/off

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**

Using the Tree View



Full Tree

Name	ID	Properties
Assemblies		
Boundary Conditions		
Groups		
Volumes		
Volume 1	1	
Surface 1	1	
Surface 2	2	
Surface 3	3	
Curve 4	4	
Curve 6	6	
Curve 9	9	
Curve 10	10	
Surface 4	4	
Surface 5	5	
Surface 6	6	

Curve 6

List Type

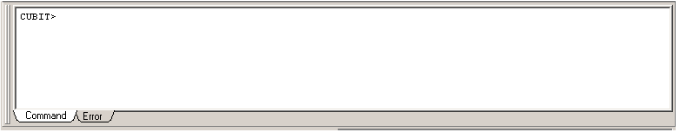
List View

Expand tree to show selected entity

Current Selection

Entering Commands

- All commands can be entered in the command window
- Partial words OK
- Can use general ID ranges
  - draw curve 1 to 5 except 4
  - draw curve in volume 2



```
CUBIT>
```

Basic Aprepro Syntax

- Aprepro expressions are wrapped in curly braces
- Aprepro evaluated first, results inserted into command:

```
Brick X {10}
```

And

```
Brick X {5*2}
```

Are Equivalent To

```
Brick X 10
```

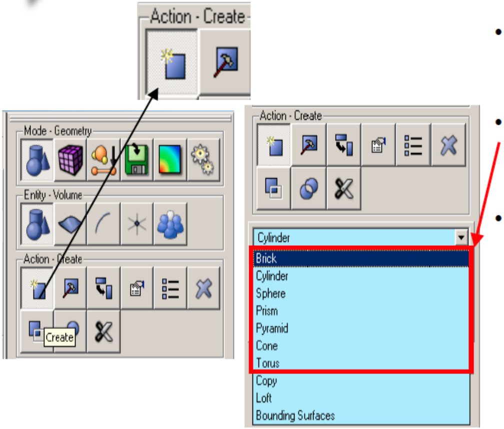
Types of Functions

- **Math Functions**
  - sin(num), cos(num), asin(num), etc…
  - sqrt(num), exp(num), log(num), ln(num), etc…
- **String Manipulation Functions**
  - Quote(string), toupper(string), tolower(string)
- **Utility Functions**
  - Print(string), PrintError(string)
  - FileExists(string), HasFeature(string)

Example 7 – Row Of Bricks

```
# ***Set parameters***
# {num_bricks=5}
# {brick_size=1}
#
# ***Create the bricks***
# {Loop(num_bricks)}
 Brick Width {brick_size}
# {EndLoop}
#
# ***Scoot them into a line***
# {cur_brick = 1}
# {Loop(num_bricks)}
 Volume {cur_brick} move {(cur_brick-1)*brick_size}
 #{cur_brick++}
# {EndLoop}
```

Create Button



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

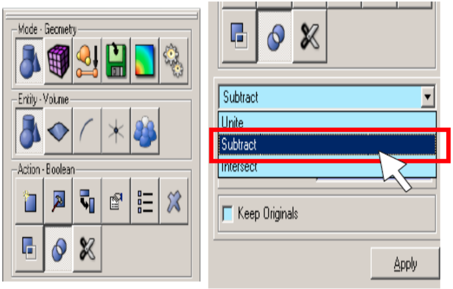
LOCKHEED MARTIN

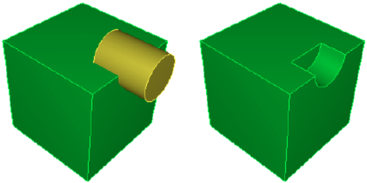
Sandia National Laboratories

COMPUTATIONAL INFRASTRUCTURE for GEODYNAMICS

Subtract

- Removes regions that overlap





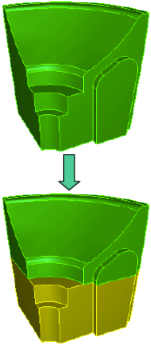
Before

After

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

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



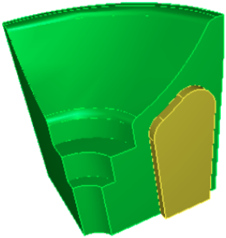
For command line syntax:

```
CUBIT> help webcut
```

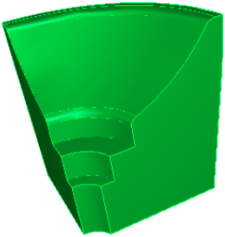
Imprinting

- 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

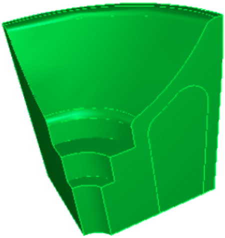
Imprinting



Body 1 and 2



Body 1
before imprinting



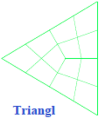
Body 1
after imprinting

Surface Meshing Schemes

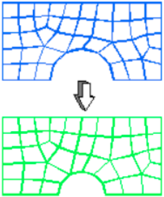
Computational Modeling Sciences Department

CUBIT Provides a number of different surface mesh schemes. Choosing the best scheme depends on the shape and number of curves in the surface. Your choice also depends on how you plan to mesh the volume.

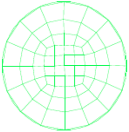
- **Surface Schemes**
  - Map
  - SubMap
  - Pave
  - Triangle Primitive
  - Circle Primitive
  - Pentagon Primitive
  - Morph/Mirror
  - TriMesh



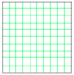
Triangle



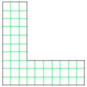
Copy/Morphing



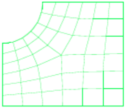
Circle



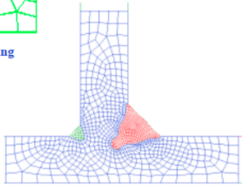
Map



SubMap



Pentagon



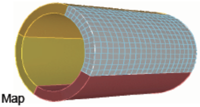
Pave

Volume Meshing Schemes

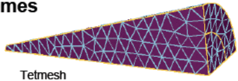
CUBIT Provides a number of different volume mesh schemes. Similar to Surface Scheme Selection, Volume Scheme Selection is based on the shape of the geometry.

Run mesh demo

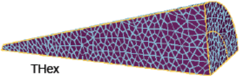
Example Volume Meshing Schemes



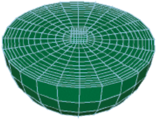
Map



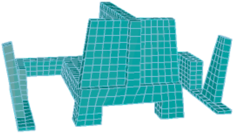
Tetmesh



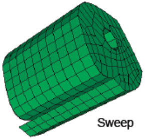
THex



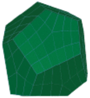
Sphere



Submap

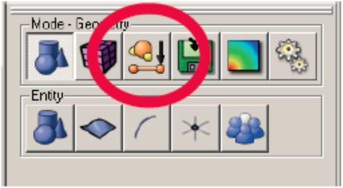


Sweep



Polyhedron

Materials and Boundary Conditions



- Blocks = materials

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

- Nodesets = groups of nodes for boundary conditions and faults

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

Exporting to PyLith

- Export mesh

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

- In `pylithapp.cfg`

  ```
  [pylithapp.mesh_generator]
  reader = pylith.meshio.MeshIOCubit
  reader.filename = out.exo
  ```

CUBIT Tips

- Can select multiple entities at once in many ways
  - `draw volume all with x_coord > 0.0`
  - `curve all in volume 1 to 3 5 visibility off`
- Use tree view and info panel (or Python) to find names/numbers/geometry information
- Make sure to
  - `reset` between tests/runs
  - `undo off` avoids bug in losing names of entities
  - Merge/Imprint all entities before meshing
- Journal files and APREPRO
  - Use journal files to save commands
  - Use APREPRO for units and variables
  - Every command is echoed in the command panel (copy/paste into journal files)

CUBIT Resources

- CUBIT website, `sandia.cubit.gov`
  - User manual
  - Tutorials (geometry with cones, prisms, spheres, etc)
  - Support via email
- CIG website
  - Slides from past CUBIT tutorials at Crustal Deformation Modeling workshops

    ```
    www.geodynamics.org
        → Community
           → Working Groups
              → Short-Term Crustal Dynamics
                 → Workshops
    ```
  - Example files
  - Some support for mesh surfaces (e.g., triangular facets)
- GeoCUBIT developed by Emanuele Casarotti
  - Python scripting of CUBIT for SPECFEM3D
  - Available from CIG code repository