CUBIT
Brett Clark
Sandia National Labs

PHYSICS-BASED MODELING IN DESIGN & DEVELOPMENT
FOR U.S. DEFENSE CONFERENCE
Denver, Colorado
November, 2011
What is CUBIT?

- Advanced Meshing Algorithms
- Small surfaces
- New local topology created
- Geometry Cleanup/Simplification
- Model Attribution
- Geometry and Mesh Generation Toolkit
- Mesh Quality
- GUI and customization
- Geometry Creation
- Geometry Decomposition
- CAD Interface
- ACIS, PRO/E, CATIA, MBG
- CAD Abstraction Layer
- Commercial Geometry Kernels
- Meshing Algorithms
The CUBIT Project

- CUBIT started as a research platform for unstructured quad and hex meshing in the early 90s
- late 90s GUI was added
- early 2000s product focus increased to improve robustness

Team Personnel
- Sandia
- Elemental Technologies, Inc.
- Independent Contractors
- Caterpillar
- University Students

Past and Present

Collaborators

Industry
- Goodyear
- Caterpillar
- TechnoSoft

Universities
- Brigham Young University
- Carnegie Mellon University
- California Institute of Technology

DoD
- Office of Naval Research
- ALEGRA

National Labs
- Lawrence Livermore National Laboratory
- Los Alamos National Laboratory
- Idaho National Laboratory

Technical Conferences
- 9th US National Congress on Computational Mechanics
- International Meshing Roundtable

CUBIT Repeat Users 2002-2011
Some Distinguishing Strengths

- **Complex Hex Meshing**
- **Scripting**
- **Geometry Clean-up & Defeaturing**
- **Contiguous Assembly Meshing**
- **Decomposition Tools**
- **Local Hex Refinement**
Complex Hex Meshing
Structured vs. Unstructured

Structured
1. Interior node valence is constant.
   ie. number of elements at each interior node=4
2. Meshing algorithm relies on specific topology constraints.
   ie. number of sides=4

Unstructured
1. Interior node valence varies.
   ie. number of elements at each node=3,4,5…
2. Meshing algorithm applies to arbitrary topology
   ie. number of sides is arbitrary
3D?

Mapped Meshing

Sweeping
(2.5 D)
Complex 3D?

Block-Structured Meshing

Partition & Sweeping
More complex solids can be meshed by first doing manual partitioning into several sweepable sub-solids.
More complex solids can be meshed by first doing manual partitioning into several sweepable sub-solids.
More complex solids can be meshed by first doing manual partitioning into several sweepable sub-solids.
More complex solids can be meshed by first doing manual partitioning into several sweepable sub-solids.
More complex solids can be meshed by first doing manual partitioning into several sweepable sub-solids.
Partitioning & Sweeping Very Complex Solids

Any geometry, regardless of complexity, can be meshed by first decomposing it into sweepable sub-solids. Decomposition step of complex solids requires tedium, experience, and creativity and often lots of time.
Hex Meshing Example: Thermomechanical Modeling of Back-End-of-the-Line 3D Interconnects

BEOL 3D via chain, (a) plan-view of exposed wires, and (b) cross section view of multi-layered chain.

Figure 4: Mesh of a 3-link via chain; (a) isometric cutaway view, and (b) cross-section of W-filled via surrounded by Ti and TiN linings and SiO₂

3D virtual geometry of a via chain. The as-drafted geometry is modified to resemble the fabricated geometry.
Assembly Meshing and Gaps/Overlaps/Misalignments

Right clicking on an overlap pair in the list and choosing “Draw Pair” will draw the two volumes with the overlap and shade the region of overlap.

Tool for quickly finding and fixing gaps, overlaps, and misalignments in CAD assembly models.

Selecting a solution will show the volume that the overlap region will be subtracted from and clicking “Execute” will do the subtraction.
Local Hexahedral Refinement

- Fully conforming hexahedral refinement
- User selects target elements or geometry to refine
- CUBIT creates all-hex transitions between refined and coarse regions

Geological model locally refined near surface and at faults
CUBIT Scripting Capability

- Journal Files
  - CUBIT Command Syntax
    - CUBIT Automatically echoes commands to create journal file
    - User can create and play journal files
  - APREPRO language permits basic programming and parameterization of variables

- Python Scripting
  - Full python interpreter included in CUBIT
  - More control than journal file
  - CUBIT Python Interface includes extensive API to CUBIT functions
  - Permits Custom GUI Creation with QT
Automated Geometry Defeaturing

Press "Detect Small Features" button to run diagnostics

List of geometric entities that fail the diagnostic test. Left click to see solutions for an entity. Right click to zoom, point to, fly-in, etc. List of specific solutions for selected entity. Left click to see a preview of the solution. Right click to get help or execute the solution (or press Execute button)
Tools for Decomposition

CUBIT provides multiple possible solutions and user selects the one they want

Types of Decomposition Cuts
- Planar
- General Surface
- Tool Body
- Sweep Surface
- Sweep Curve
Current Technology Thrusts

- Hex Meshing from Volume Fractions
- Parallel Refinement
- Geometry Defeaturing For Meshing
- Moving Meshes, Mesh Morphing
- Integrated User Experience
Geometry Defeaturing
For Meshing
Automatic Geometry Defeaturing

Solid Model

Facet Representation
Automatic Geometry Defeaturing

Basic Operation – Edge Collapse

Initial facets

Final facets following collapse operations
Automatic Geometry Defetauring

Original Solid Model
(ACIS-based representation)

Final Solid Model
(Facet-based representation)
Defeaturing Tool
Sculptor is a new tool for generating meshes from volume fraction data generated from CTH for use in Sandia’s FEA codes.
Hex mesh constructed on eight processors and its timing data.

Detail from helix meshed at 2 different resolutions.

Scalability on fixed model size Shapes Diatom (2.26 Million Hexahedra)

- Ideal (1/n)

Time (Seconds)

Number of Processors
Close-up view of hex meshes generated at two different time steps of a simulated pipe bomb explosion.

Hex mesh of simulated grain microstructure with 15 different materials.
8 Processors, 4.19 million hexahedra. Hexes color-coded by processor.
Mesh Morphing for CAD
Parametric Shape Optimization

A representation of the procedure used for shape optimization. The same mesh is used on a new geometry description at each iteration.

Same mesh is morphed on to new geometry
Parallel Refinement with Geometry

Input Mesh
1,936 Hex

2 levels refinement
124k Hex

4 levels refinement
7.93M Hex

5 levels refinement
63.4M Hex
STK_Adapt is a Sandia (Trilinos) module which refines unstructured meshes. Input mesh is an Exodus II file. Had no knowledge of CAD. In previous releases, new nodes were positioned by evaluating element shape functions.

Previous Workflow

Zoom in of faceting artifacts from original mesh.
Sierra STK_ADAPT imports the coarse exo file and 3dm files. The mesh is partitioned to $N$ processors. The 3dm file is read into the OpenNURBS geometry evaluator kernel. Global uniform refinement is performed with new nodes projected to geometry.

**Step 1**

Cubit exports:

- A coarse mesh (exo file) and:
- A geometry file (3dm OpenNURBS)

**Step 2**

- 112 M hex elements
- 20 processors

Refined nodes are placed on original CAD geometry.
Example - Bone

- Input Mesh: 86k Tet
- Input Mesh: 5.52M Tet
- Input Mesh: 44.2M Tet
Integrated User Experience
The Design Through Analysis Process

Cubit

DART Workbench

Integrated Product

Future

Geometry
Meshing
Model Attribution
Input Deck Creation
Job Submission
Sim Data Management
Post Processing
The Eclipse Framework

Eclipse was chosen as the environment in which Cubit and the DART Workbench are being integrated. Eclipse provides a powerful plug-in environment ideal for integrating different pieces of functionality. The DART Workbench was already developed within Eclipse. Cubit’s GUI is built upon the Qt toolkit with Python as the glue between the Cubit GUI and the core functionality accessed through a command line interface. The Cubit GUI is being replicated within the new environment. During this process certain parts of the GUI are being redesigned and improved.
SAW
Sandia Analysis Workbench
SAW
Sandia Analysis Workbench
SAW
Sandia Analysis Workbench
As the Cubit GUI has been replicated in the Eclipse environment powerful tools for automatically generating GUI panels from XML have been developed. This has greatly facilitated the GUI development process and will allow a user-modifiable architecture for GUI development and modification in the future.

Cubit command defined with XML.

Automatically generated GUI from XML.
Multi-Model for V&V and Multi-Physics

Cubit has traditionally only supported a single model at a time. However, part of the integration effort will be to expand Cubit to use a client-server architecture which will facilitate working with multiple models at a time. This need has been identified for Verification and Validation studies as well as for multi-physics applications.

Multiple mesh types for multi-physics support.

Multiple mesh resolutions for a single CAD model for V&V studies.
Vision Going Forward: Collaboration!

- Open Source
  - The CUBIT product will be going open source under a very unrestrictive license (target date: end of FY12)
- Integration with the DART Workbench under Eclipse plug-in environment
  - Ideal environment for future collaboration
- Continue to modularize core CUBIT as plug-ins to facilitate sharing and collaboration
Contact Info

Brett Clark
Sandia National Labs
Phone: 505-844-0434
Email: bwclark@sandia.gov