Coupling of Smooth Faceted Surface Evaluations in the SIERRA FEA Code

Timothy J. Tautges
Steven J. Owen
Sandia National Laboratories
University of Wisconsin-Madison

Mini-symposium on Computational Geometry for Mechanics and Applications
5th World Congress on Computational Mechanics
July 9, 2002
Outline

• Introduction
• CGM details
• Integration into SIERRA for h-refinement
• Examples
• Summary & future work
Introduction

• Actual representation of the spatial domain varies over the simulation process
  – Continuous representation: geometry (CAD)
  – Discretized representation: mesh
  – Other groupings (parallel decomposition, contact surfaces, shock interfaces, viz grouping)

• Currently, relationship between continuous & discretized representations is lost
  • However, many applications could use these relationships!
Introduction

• Solid geometry is widely used in simulation
  – Larger simulations making it possible to resolve small geometric features
  – Linking directly to design enables design-based simulation, iterations
  – Applications: adaptive refinement, flow over curved geometry, monte carlo transport, ...

• Geometry functionality found in CUBIT Mesh Generation Toolkit encapsulated in CGM libraries

• CGM being integrated into FE codes (GOMA, SIERRA) and MCNP_X monte carlo xport code
Why Is Integration Necessary?

Example: H-Refinement With Linear Facets

- SIERRA simulation of HE “cookoff”

- Geometric discontinuity leads to unphysical results!

*Courtesy of S. Bova, Sandia National Labs*
CGM - The Geometry Bus

- CGM is a set of libraries that provide non-manifold, solid model-based geometry modeling for analysis applications
The Common Geometry Module (CGM)

CUBIT

- Hex Improv.
- Adv. Hex Smoothing
- Automatic Algorithm Selection
- Submap
- Int. Map
- Pave
- Skew Control

CGM

Virtual Geometry, Topology & Mesh Interface

- Composite & Partition Geometry
- Merge Topology

Facet (CUBIT) | ACIS | Healing IGES STEP Local Ops | Pro E | SolidWorks | …
Facet-Based Smooth Surfaces
Requirements

• Wealth of previous work on smooth surface modeling on discrete facets
  – Hoppe et. al, Floater, Walton and Meek, etc.

• Requirements:
  – Build C1-continuous surfaces from patches of triangular facets
  – Use these surfaces as replacement or auxiliary surface representation for mesh generation and adaptation
  – Treat facet data from many sources, including graphics facets, mesh elements, and point cloud triangulations
  – Maximize code and data re-use in facet-based surface approximation
CGM Smooth Facets Implementation
G1-Continuous Surface from Tri Facets

- Quartic spline approximation gives G1-continuous surface across triangular facets

- Input: vertex coordinates, facet connectivity (, normals at vertices)
- Functions needed for meshing/evaluation: closest point, normal, derivatives
  - Derivatives approximated using differencing
Application-Based Interface to Facet Data
NxM Interface – NO!!

Meshing Algorithms

H-adaptivity

Boundary Conditions

...
CGM-Based Interface to Facet Data
Better…

- Meshing Algorithms
- H-adaptivity
- Boundary Conditions
- ...
Native (sub-CGM) Interface to Facet Data Best!

- Meshing Algorithms
- H-adaptivity
- Boundary Conditions
- Scanned data (ribosome)
- Imported mesh
- Analysis data (deformed drop)
- Mesh Database
- SIERRA
- Graphics Engine
- Facet Engine (CGM)
- Graphics facets

Goals (both important):
- minimize data duplication
- maximize code reuse
CGM Implementation
Facet Class Design

- CGM: abstract classes defining functions for topology traversal, point location, generic smooth surface functionality
- CGM Facet Data: local storage of facet data, functions
- SIERRA Interface: functions implemented using SIERRA mesh data classes (Fmwk_MeshObj)
SIERRA Faceted Surface Requirements

- **Fundamental requirement:** smooth faceted surface must pass through nodes (including displacements)

- **2 cases:**
  - **Static mesh:**
    - Smooth surface can use original points/facets
  - **Dynamic mesh:**
    - Smooth surface must use updated node positions, including new nodes from h-refinement
    - Faceted surface must be dynamic in point locations AND facets

- **SIERRA** h-adapted quads & tris, CGM tri facets & template-based face to facet(s) mapping
Refinement Templates

- Sierra surface elements may be decomposed into tris, and may have h-refined neighbors:

- CGM uses triangular facets
  - Sierra surface elements will have multiple facets

- Refinement templates can be used to avoid storing facet connectivity on every CGM facet:
  - Saves space (static tables & 2 or 3 int variables per Fmwk_MeshObj)
  - Indexing using ints, so it’s fast
  - Sierra functions/data still used for inter-element & unambiguous intra-element connectivity
4 Triangle Refinement Templates

htype = 0

htype = 1

htype = 2

htype = 3
6 Quadrilateral Refinement Templates

htype = 0

htype = 1

htype = 2

htype = 3

htype = 4

htype = 5

Sandia National Laboratories
Pseudo code

facet_edge->facets(facet_list)

- If (my SIERRA owner == FACE) // interior facet edge
  • Get owning facets from static fedge-facet tables
- Else // on SIERRA edge
  • Get SIERRA faces owning sierra edge
  • For each SIERRA face:
    - Get local rotation wrt refinement template for this face
    - Get facets from static fedge-facet tables
Example 1
Uniform H-Refinement on Cylinder

- Initial coarse hex mesh representing cylinder
- 2nd pass of uniform h-refinement
- 3rd pass of uniform h-refinement

- Uniform h-refinement with new nodes on boundary snapped to the smooth-faceted surfaces

- New node locations computed exactly on the cylinder (within tolerance)
Example 2
Uniform H-Refinement on Sphere

- Initial coarse hex mesh representing concentric spheres
- Uniform h-refinement with new nodes on boundary and at block interfaces snapped to the CGM geometry
- New node locations computed exactly on the spheres (within tolerance)
- 2nd pass of uniform h-refinement
- 3rd pass of uniform h-refinement

56 elements 448 elements
3584 elements 28,672 elements
Example 3
SIERRA “Crease” Test

- Demonstrates smooth surfaces defined from Sierra elements including discontinuities.
- Crease becomes a feature where the feature angle criteria \( f=135 \) is met, then two normals at facet vertices along the crease are defined.
- Changing the feature angle also changes the length of the crease.

For this test, 16 locations and normals (4X4 grid) are evaluated for every quad face.
Example 4
SIERRA “Warped” Test

• Definition of facet-based surfaces on internal and external material boundaries

• Test also shows definition of smooth surfaces across adjoining blocks
  – Angle-based
  – Normals aligned consistent with mesh
Open Issue: Quality Degeneration From Boundary-Only h-refinement

• Uniform refinement of concentric spheres, snapping new nodes to the boundary:

• Interior refinement should account for boundary snapping
Why Use CGM?

• **Common** interface to geometry in various formats:
  – Solid model
  – Facet-based (FE mesh, scanned/CT data, etc.)
  – Virtual (CUBIT-specific representation)
• Reliability/Support
  – CUBIT links it directly, unmodified
  – If it’s broken in CGM, it’s broken in CUBIT too
• Leverages future CUBIT geometry work
  – SolidWorks*, Pro/E** ports
  – Automatic detail suppression (using virtual geometry)
  – Facet-based geometry modeling & model acquisition
Summary & Future Work

• CGM has been integrated into SIERRA multi-physics code
  – Models smooth-faceted surfaces for static & deforming* meshes
  – Facet data evaluates SIERRA node positions/element connectivity directly
    • No data copies
  – Facet-based modeling code in CGM used without modification
    • Interface classes used for facet objects

• Support being developed for solid model distribution on parallel computers

• Research paths:
  – Investigate hierarchical data models
  – Dynamic meshes with lazy update of surface data