Finite Element Meshing of the SCEC Community Fault Model: Methods and Algorithms

Carl W. Gable — gable@lanl.gov, Los Alamos National Laboratory, Earth and Environmental Sci. Division, MS T003 LANL, Los Alamos, NM 87545
Brad H. Hager — bhhager@mit.edu, Massachusetts Institute of Technology, Dept. of Earth, Atmospheric and Planetary Sciences, Cambridge, MA 02139
Jingning Lu — johnnylu@mit.edu, Massachusetts Institute of Technology, Dept. of Earth, Atmospheric and Planetary Sciences, Cambridge, MA 02139
Charles A. Williams — willic3@rpi.edu, Rensselaer Polytechnic Institute, Dept. of Earth Environmental Sciences Science Center 1W19, Troy, NY 12180

Los Alamos NATIONAL LABORATORY EST. 1943
Massachusetts Institute of Technology
Rensselaer

Abstract

Complex models of fault interactions and crustal deformation have been developed for Southern California. **The Southern California Earthquake Center (SCEC)** has sponsored the creation of the **Community Fault Model (CFM)**, http://epicenter.usc.edu/cmeportal/cmodels.html, a 3D representation of active faults in southern California that are deemed capable of generating moderate to large earthquakes. In order to integrate the **CFM** into finite element models of stress and strain, **SCEC** has also supported development of mesh generation algorithms and tools to create computational meshes that capture **CFM** geometry. A series of simple benchmark meshes have been created and made available to the community (see URL), as well as more complex meshes that incorporate the **CFM** geometry for the faults associated with Landers and Hector Mine earthquakes. General methods that take advantage of algorithms for conforming Delaunay tetrahedral meshing of a planar straight line complex (the fault triangulation) are described and their application to meshing over one hundred faults from the **CFM** are presented. These methods have general application to meshing of other non-manifold geometries. The results of visco-elastic crustal deformation calculations that utilize these finite element meshes illustrate applications. Geodynamics modeling, from data integration, conceptual model development, model construction, algorithm development, and computations and analysis, has become complex enough to necessitate the compartmentalization and specialization of some tasks.

BACKGROUND AND GOALS

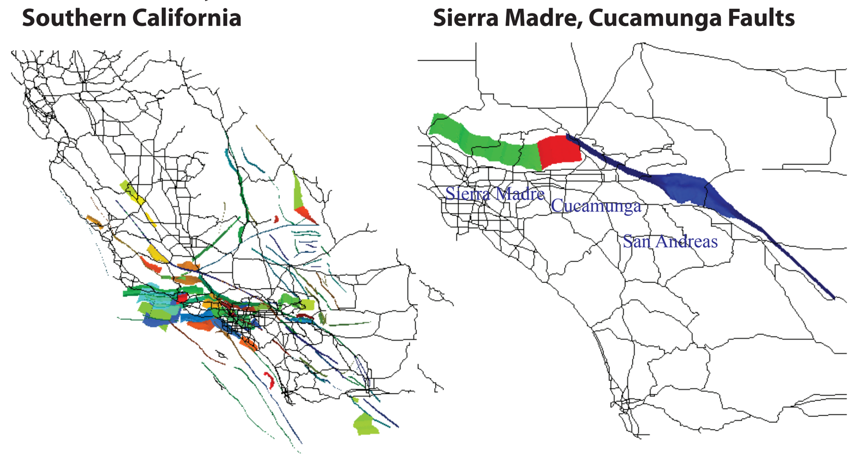
SCEC - Southern California Earthquake Center
CFM - Community Fault Model

What is SCEC / CFM?

"The Unified Structural Representation Focus Area of SCEC has developed a new, community-based 3D model of active faults in southern California designed for use in fault systems analysis, seismic hazards assessment, and the SCEC Community Velocity Model."
From: http://structure.harvard.edu/cfm/themodel.html

The form of the **CFM** is a set of 3D triangulated surfaces in that represents the best estimate of fault surfaces. In some cases the CFM will have a 'base case' and 'alternative' representations for a fault surface.

Our **goal** is to utilize the **CFM** as input and create high fidelity finite element models of crustal deformation that incorporate the geometric information in the **CFM**.



154 faults of the CFM (colors) plotted with major roads (black lines) of California for orientation. *Three faults from the CFM that will be used to demonstrate the method of developing a fault conforming finite element mesh.*

Why ?

The finite element mesh is used in viscoelastic stress/strain calculations to study stress transfer, crustal deformation and crustal rheoology. Geometric complexity can be an important aspect of the system.

Building a Finite Element Mesh Using Community Fault Model (CFM) Input

The CFM represents faults as a set of triangulated surfaces. The goal is to build a 3D tetrahedral finite element mesh with internal faces that conforms to the fault surfaces. This is accomplished with the following steps:

Step 1) Modify CFM surfaces to improve aspect ratio of triangles and refine them to a desired edge length scale (L).

Step 2) Build a background mesh that fills the computational volume of interest with variable resolution elements such that in the volume surrounding the faults, the edge length of the volume mesh elements are close to the length scale (L) of the fault triangles.

Step 3) Remove all elements from the 3D mesh within distance L of the fault surfaces.

Step 4) Combine the point distribution from the fault surfaces and the 3D volume mesh and Delaunay tetrahedral mesh from point distribution. This takes advantage of the properties of Delaunay triangulations and theory proved in: Murphy, M, D Mount, CW Gable, "A point-placement Strategy for conforming Delaunay tetrahedralization", J. Computational Geometry 2001. The resulting mesh will have tetrahedra such that the interior triangular faces conform to the fault surface triangulations.

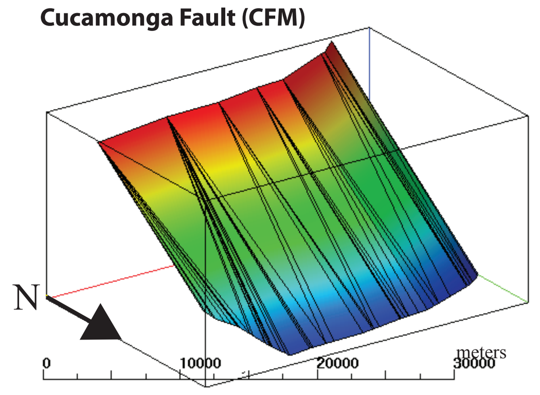
Step 5) Improve mesh quality with a combination of smoothing (node movement), reconnections, refinement and derefinement.

Step 6) Compute various attributes necessary for model set-up, initial conditions and boundary conditions.

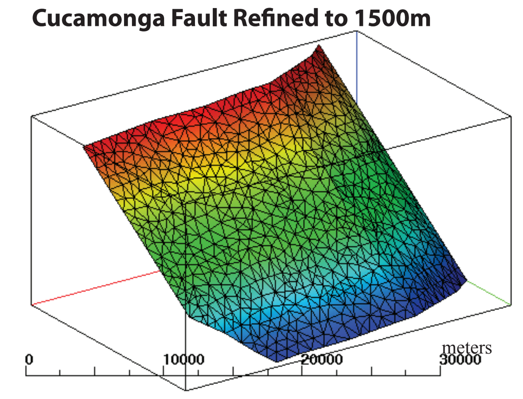
Methods and Results

Step 1) Modify CFM surfaces to improve aspect ratio of triangles and refine them to a desired edge length scale (L).

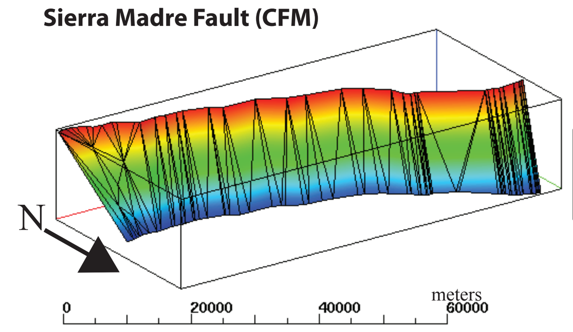
Cucamonga Fault (CFM)



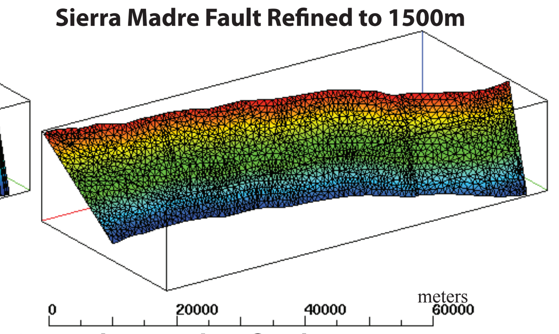
Cucamonga Fault Refined to 1500m



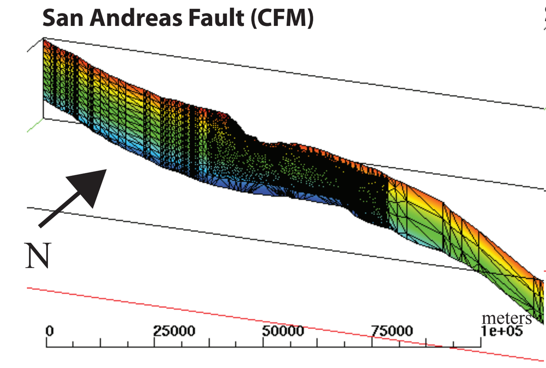
Sierra Madre Fault (CFM)



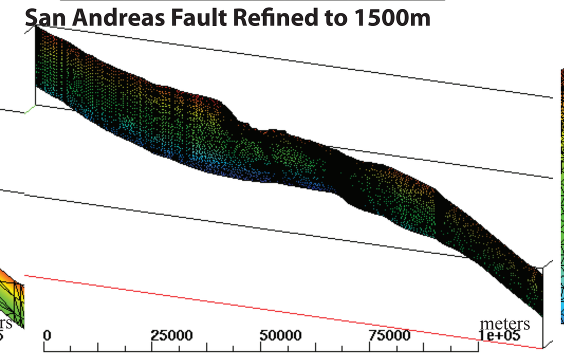
Sierra Madre Fault Refined to 1500m



San Andreas Fault (CFM)

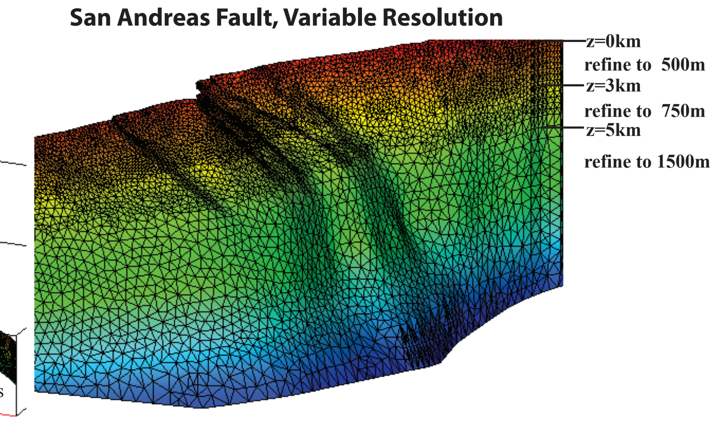


San Andreas Fault Refined to 1500m

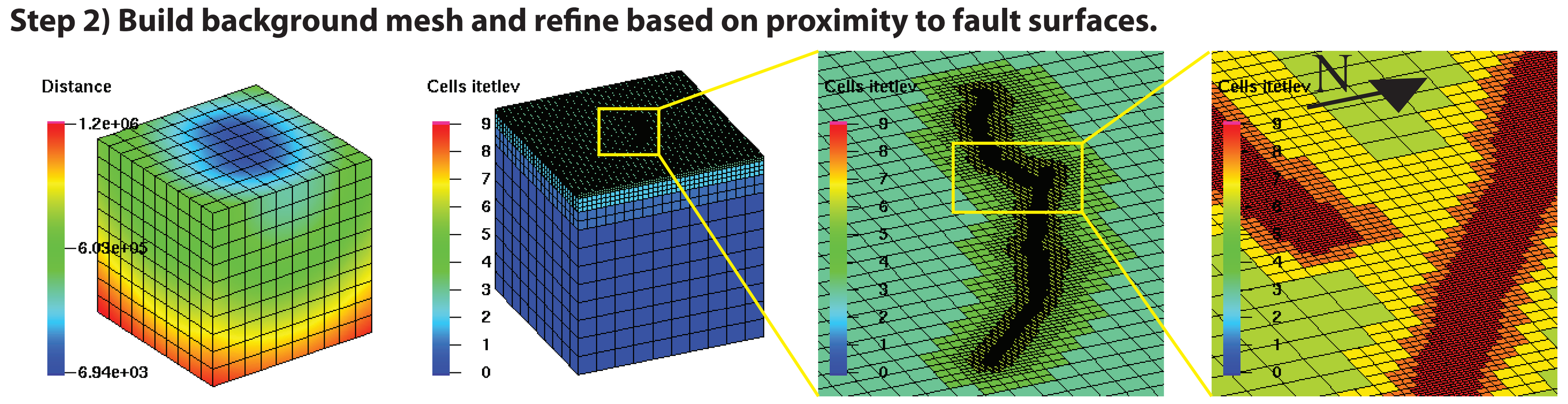


Three faults from the CFM are *'massaged'*. The faults are the San Andreas, Sierra Madre and Cucamonga. The surfaces are colored by elevation (z coordinate). The central San Andreas has considerably more geometric detail. The other faults are more planar in character. The massage algorithm refines any edge longer than the specified value, L=1500m in this case, and also derefines the sliver triangles by merging nodes as long as merging does not modify the geometry beyond a user specified value. The smoothing algorithm allows nodes to move along the surface in order to improve triangle aspect ratio.

San Andreas Fault, Variable Resolution



Step 2) Build background mesh and refine based on proximity to fault surfaces.



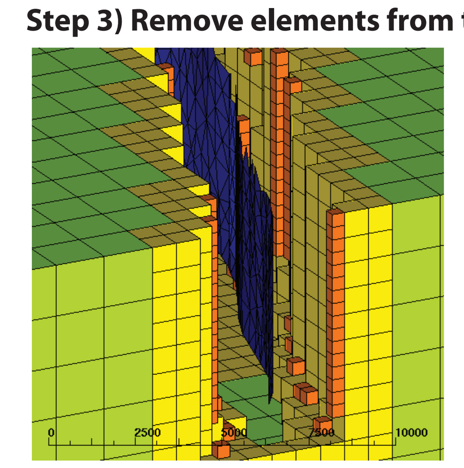
Background mesh with 10 x10x10 blocks and resolution dx=dy=dz=100km. Color scale represents distance (m) from fault triangulations to each node of the mesh.

Uniform mesh has been refined nine times using balanced octree refinement. Elements within a specified distance of the faults are progressivly split into eight elements. Color scale represents the refinement level.

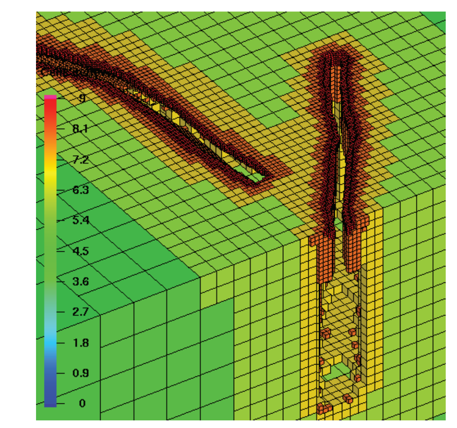
Closer view of the top surface near the faults. Octree refinement results in element dimension: level0=1.e5m, level1=5e4m, level2 = 2.5e4m, level3=1.25e4m, level4=6250m, level5=3125m, level6=1562.5m, level7=781.25m, level8=390.625m, 195.3125m

Close view of the area near the San Andreas fault (right) and the end of the Cucamonga fault. Note that refinement insures that any element has neighbors that are either the same, one higher or one lower level of refinement. This insures a resolution gradient that is never greater than 2.

Step 3) Remove elements from the 3D mesh within distance L of the fault surfaces.

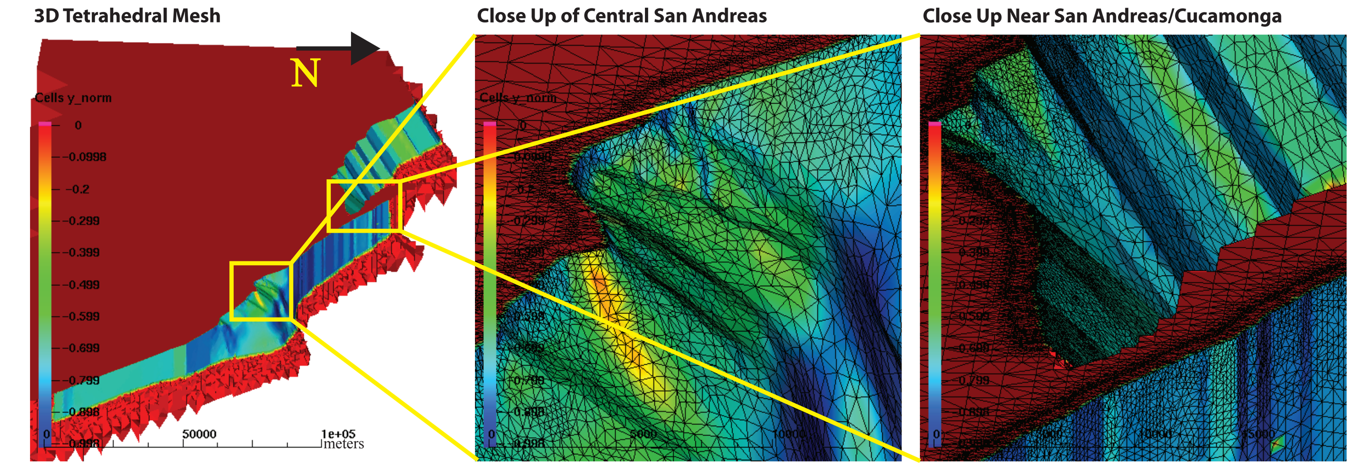


Fault triangulation (blue) inside octree refined mesh with elements near the fault surface removed.



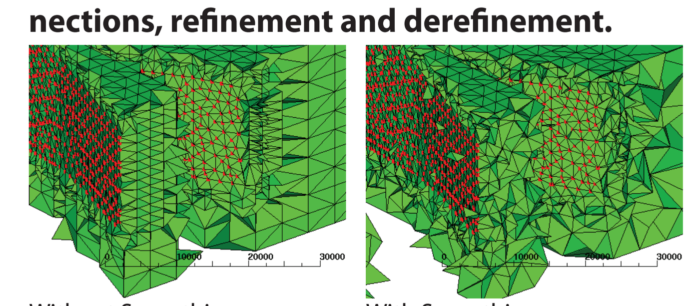
Cut away view after elements near the fault surfaces have been removed.

Step 4) Connect nodes to form a Delaunay tetrahedral mesh that conforms to the fault surfaces.



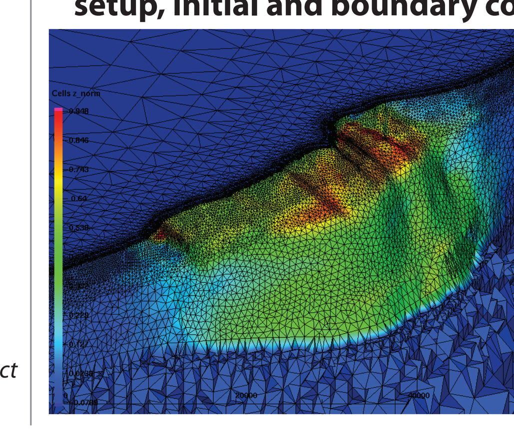
Tetrahedral mesh with elements north of the faults removed. Fault surfaces are colored by the y component of the fault surface normal vector. The lower 900km of the mesh have been removed for viewing. See the theory section for a description of why the fault surfaces emerge from a Delaunay tetrahedralization of the point distribution.

Step 5) Improve mesh quality with a combination of smoothing (node movement), reconnections, refinement and derefinement.



Smoothing and reconnection of node position improves element aspect ratio and creates a more isotropic mesh without changing fault and exterior node positions.

Step 6) Compute attributes necessary for setup, initial and boundary conditions.



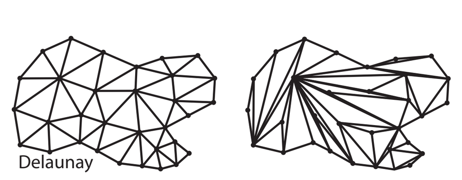
e.g. The normal vector to each node of fault surfaces is computed and output for use in setting boundary conditions. The Z component is show on the San Andreas fault. Z_norm=1 is a horizontal surface, Z_norm=0 is a vertical surface.

The Theory Behind the Approach

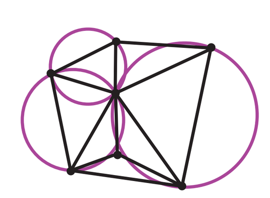
Murphy, M, D Mount, CW Gable, "A point-placement Strategy for conforming Delaunay tetrahedralization", J. Computational Geometry 2001

How to connect a point distribution?

There are many bad possibilities.

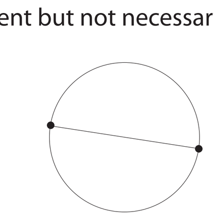


The Delaunay triangulation is a good option due to properties such as being the one which maximizes the minimum angle.

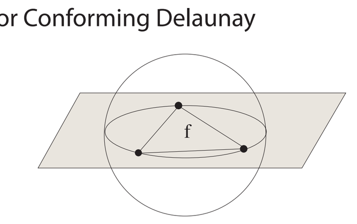


Delaunay Triangles (tetrahedra): The circumscribed circle (sphere) of any tri (tet) contains the 3 (4) points of the tri (tet) and no other points.

Sufficient but not necessary condition for Conforming Delaunay

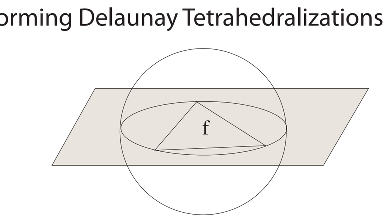


2D
The minimum diameter circle of every edge on the boundary is point-free.



3D
The minimum diameter sphere of every triangle on the boundary (fault) is point-free.

Conforming Delaunay Tetrahedralizations



Lemma: A triangular face f of a triangulated surface with vertex set V is a face in the Delaunay tetrahedralization of V if and only if there exists a sphere passing through the verticies of f containing no points of V in its interior.

Some of the Algorithms and Tools Used in This Project

LaGriT - Los Alamos Grid Toolbox (http://lagrit.lanl.gov http://meshing.lanl.gov)

LaGriT is a software package that provides mesh generation, mesh optimization and dynamic mesh maintenance in two and three dimensions for a variety of applications. A variety of techniques for distributing points within geometric regions are provided. The primary, but not only method of filling volumes with elements uses a Delaunay tetrahedralization algorithm that maintains multiple material interfaces. Data structures created to implement this algorithm are compact and powerful and include hybrid meshes of seven different element types (tet, hex, prism, pyramid, quadrilateral, triangle, line).

Mesh refinement, derefinement and smoothing modify the mesh for mesh quality improvement or variable resolution. Mesh refinement adds nodes to the mesh based on geometric criteria such as edge length or based on field variable value or gradient. Mesh elements may become distorted as mesh nodes move during a Lagrangian simulation. Mesh smoothing moves nodes to adapt the mesh to field variable measures, or improve element quality. During time dependent (Lagrangian moving mesh) simulation or when nodes are added as a result of refinement operations, mesh reconnection via a series of edge flips can be used to eliminate highly distorted elements.

LaGriT has three modes of use, 1) command line 2) batch driven via a control file 3) all modules can be called from C/Fortran programs. There is no GUI interface.

3D Octree Refinement An octree refine scheme starts with a 3D hexahedral and successively refines 1 hex into 8 hex elements by splitting along 3 topologically orthogonal planes. Additionally the octree is balanced so that any hex will only have neighbors that are the same, one higher or one lower level of refinement.

Delaunay Tetrahedralization Compute Delaunay tetrahedra from a 3D point distribution. Synthetic Normal - Compute normal vector to each node of a surface triangulations. Normals are not defined for nodes so two methods are imployed to approximate normal to nodes, area weighted normal and angle weighted normal. This information is output for use in PyLith for applying displacement boundary conditions to fault nodes.

Mesh to Mesh Intersections Elements can be identified by computing the intersection of one mesh object (e.g. computational mesh) with another mesh object (e.g. fault triangulation) to determine candidate elements for an action such as refinement. The algorithms work on element types point, line, triangle, quadralateral, tetrahedra, hexahedra.

Distance Field Calculation To compute the proximity of one object to another fast algorithms are implemented to compute the distance from nodes of one object (e.g. computational mesh) to the nearest point on another object (e.g. fault triangulation).

Mesh Quality Improvement A mesh *massage* algorithm combines smoothing, reconnection, refine and derefine in a single module to produce a mesh with the desired resolution and quality. These algorithms work on a multi-material mesh without damaging internal and external material interfaces. It can operate on a subset of the mesh if that is required.

Mesh Modification One can delete a set of elements from a mesh or two meshes can be merged.

Node/Element Attributes A mesh can have an arbitrary number of integer, real or character attributes associated with nodes or elements. Math operations can be performed on an attribute or between attributes.

Mesh Quality Various quality measures (volume, aspect ratio, dihedral angle, solid angle, edge length ratio, etc.) can be computed.

Download the software for FREE!
LaGriT - Los Alamos Grid Toolbox
http://lagrit.lanl.gov http://meshing.lanl.gov

Acknowledgement: This work has been funded by a grant from the Southern California Earthquake Center.

Applications:
G14A-08 Using Finite Element Meshes Derived from the SCEC Community Fault Model to Evaluate the Effects of Detailed Fault Geometry and Material Inhomogeneities, Williams, C A, Gable, C W, Hager, B H, Lu, J

G13A-0919 The QuakeSim GeoFEST modeling system - new features inspired by San Andreas Fault motion, Norton, C D, Parker, J W, Lyzenga, G, Lundgren, P, Gable, C W