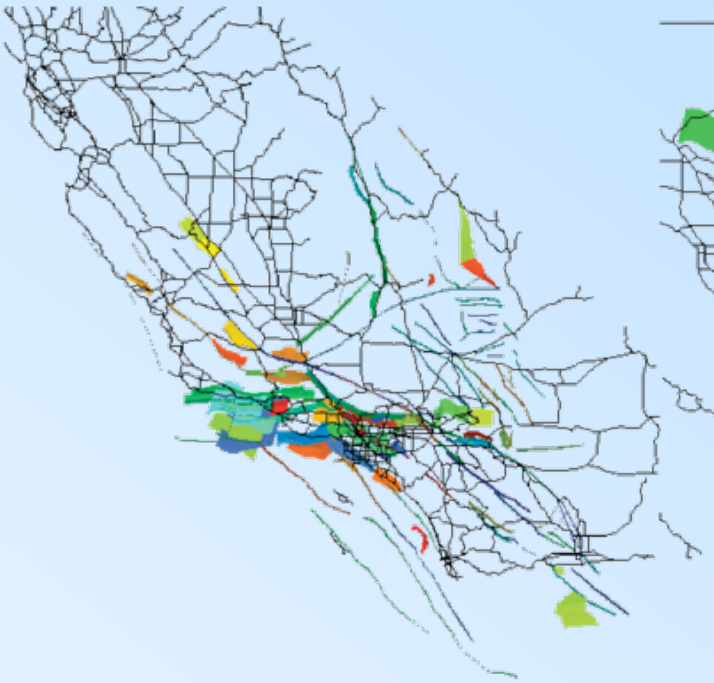
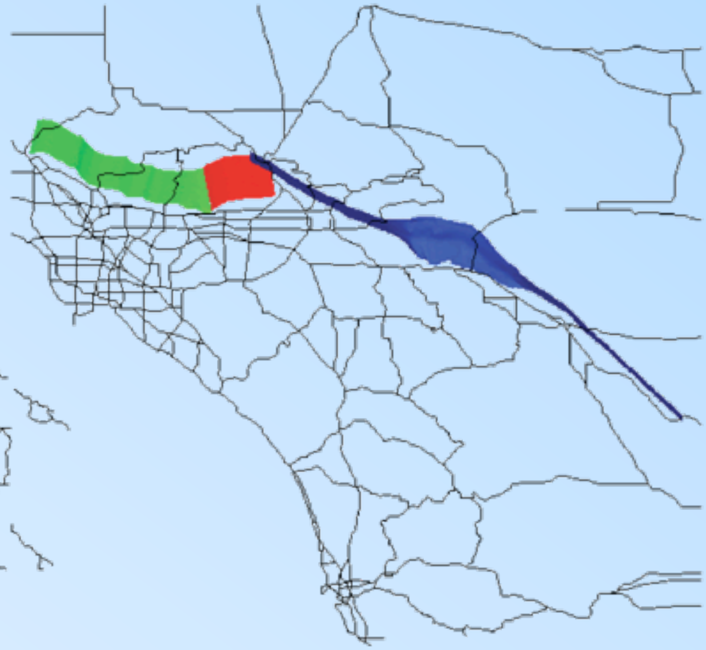
SCEC Community Fault Model Southern California



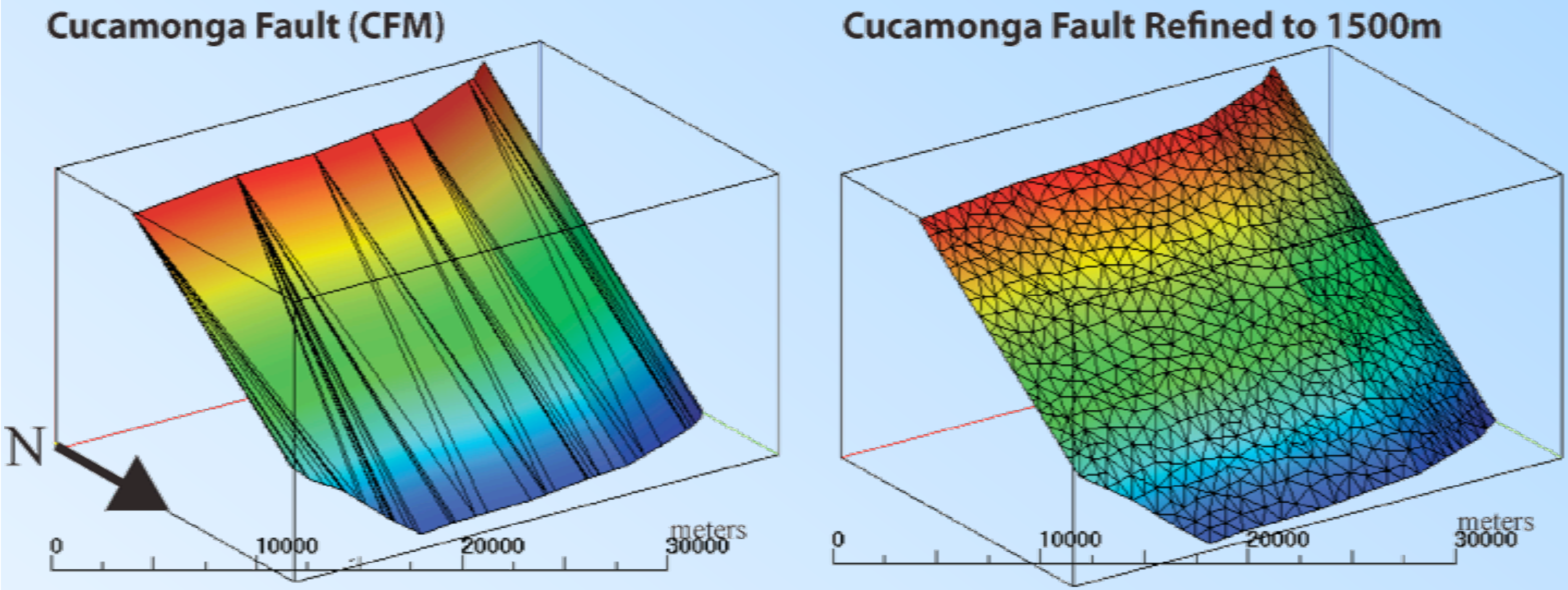
154 faults of the CFM (colors) plotted with major roads (black lines) of California for orientation.

SCEC CFM, San Andreas Sierra Madre, Cucamunga Faults

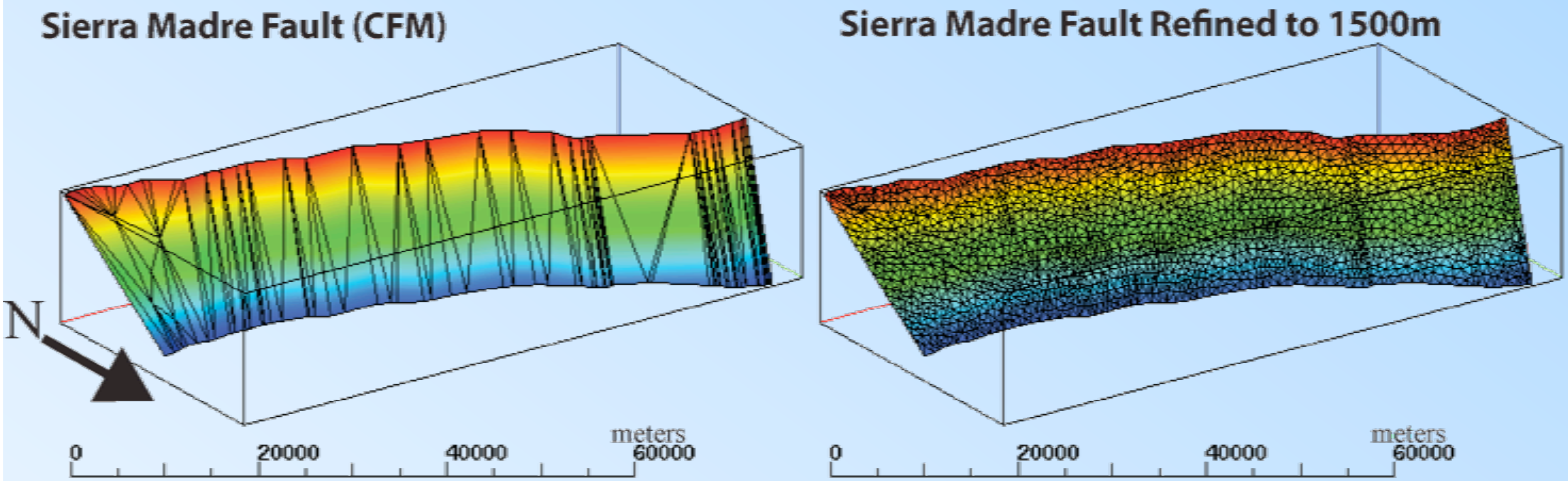


Three faults from the CFM that will be used to demonstrate the method of developing a fault conforming finite element mesh.

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

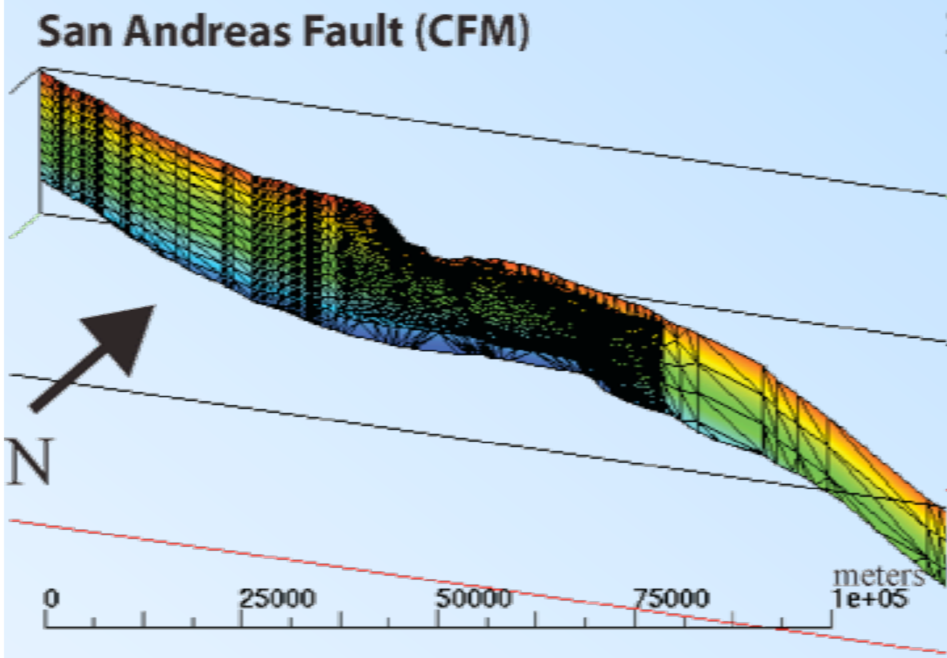


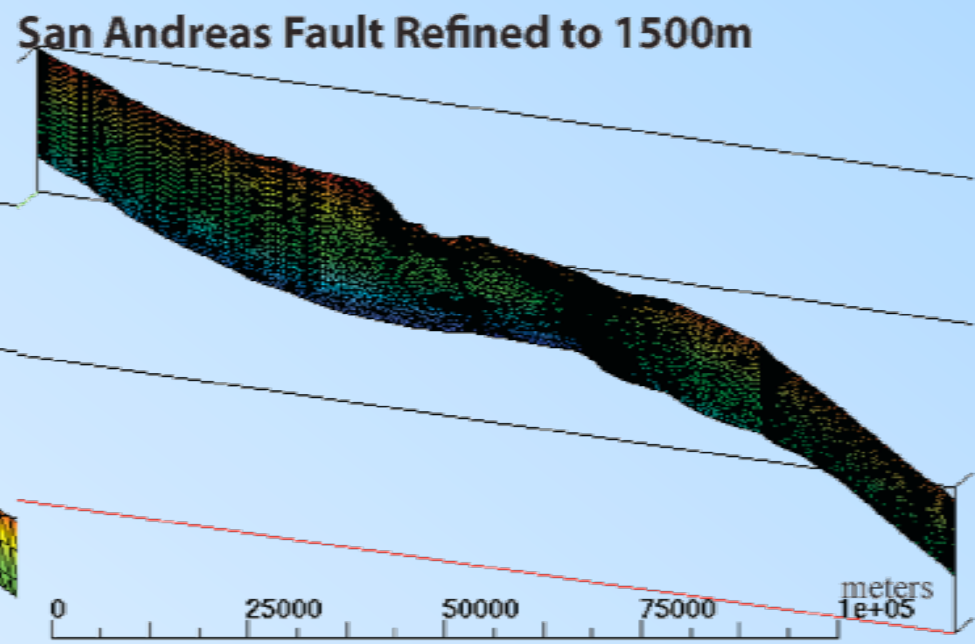
Step 1) Modify CFM surfaces to improve aspect ratio of triangles and refine them to a desired edge length scale (L)



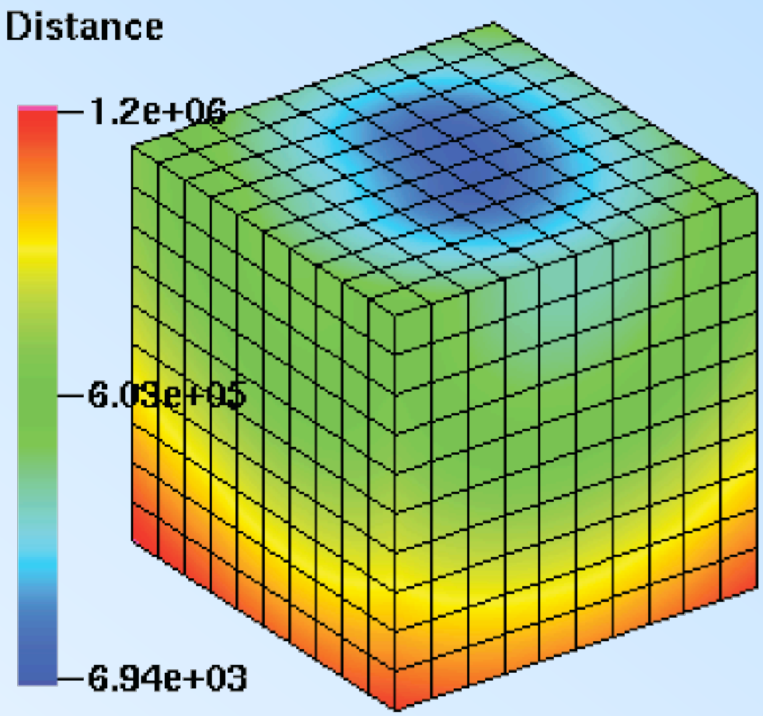
http://meshing.lanl.gov/proj

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





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

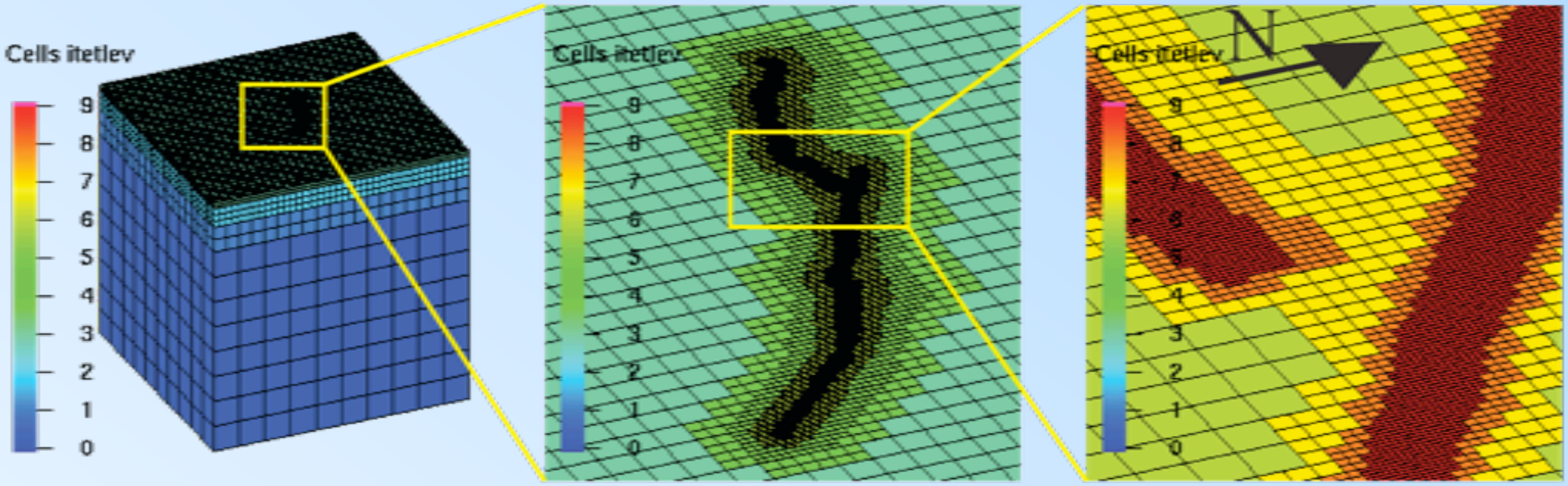


Background mesh with 10 x10x10 blocks and resolution dx=dy=dz=100km. Color scale represents distance (m) from fault triangulations to each node of the mesh.

- Build Hex mesh
- Build or import fault surface
- Compute distance field
  - `compute / distance_field / mo1 / mo2 / dfield`

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

•Octree refine based on distance field



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

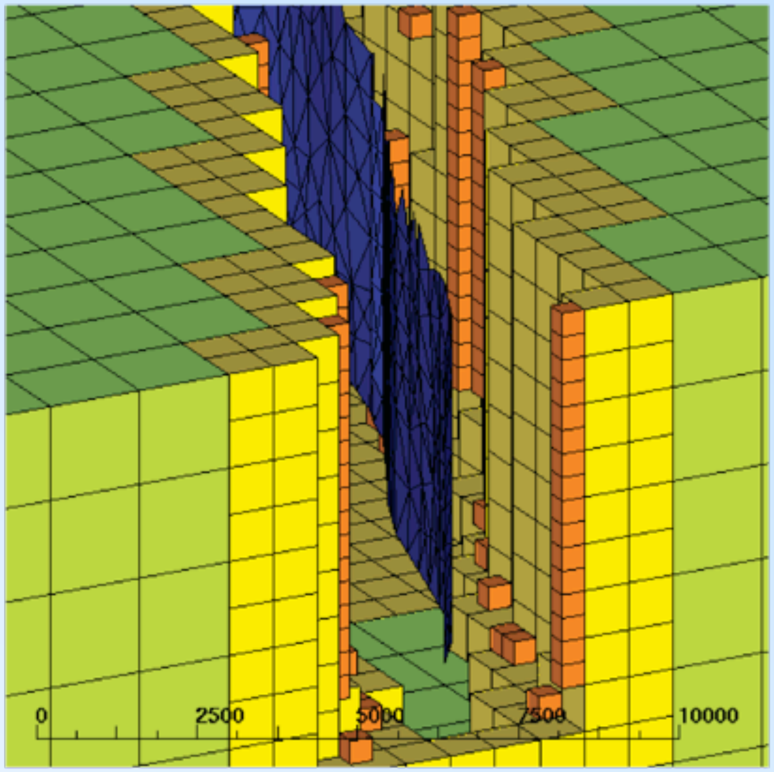
Close view of the area near the San Andreas fault (right) and the end of the Cucamonga fault. Note that refinement insures that any element has neighbors that are either the same, one higher or one lower level of refinement. This insures a resolution gradient that is never greater than 2.

Los Alamos NATIONAL LABORATORY EST. 1943

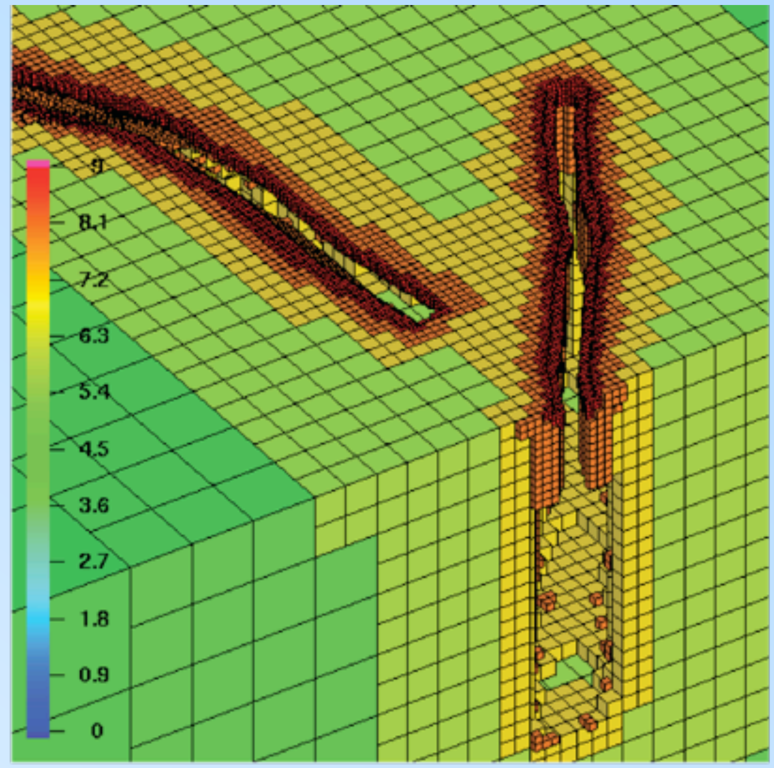
http://meshing.lanl.gov/proj

NNSA

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.

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

Find where elements in hex mesh are intersected by fault mesh

```
intersect_elements / mo_hex / mo_tri / in_flt
```

Make an element set of intersected elements

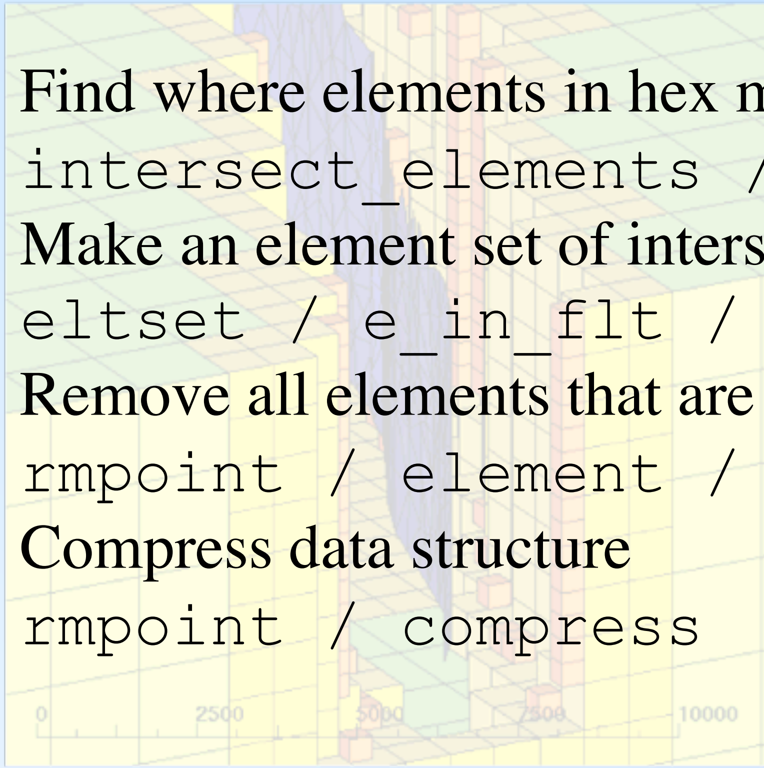
```
eltset / e_in_flt / in_flt / gt / 0
```

Remove all elements that are members of the element set

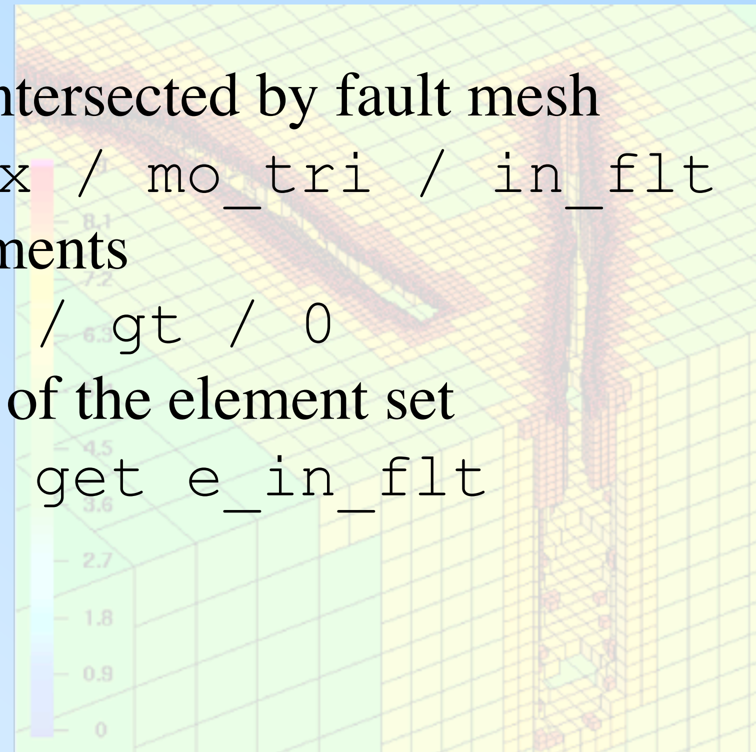
```
rmpoint / element / eltset get e_in_flt
```

Compress data structure

```
rmpoint / compress
```

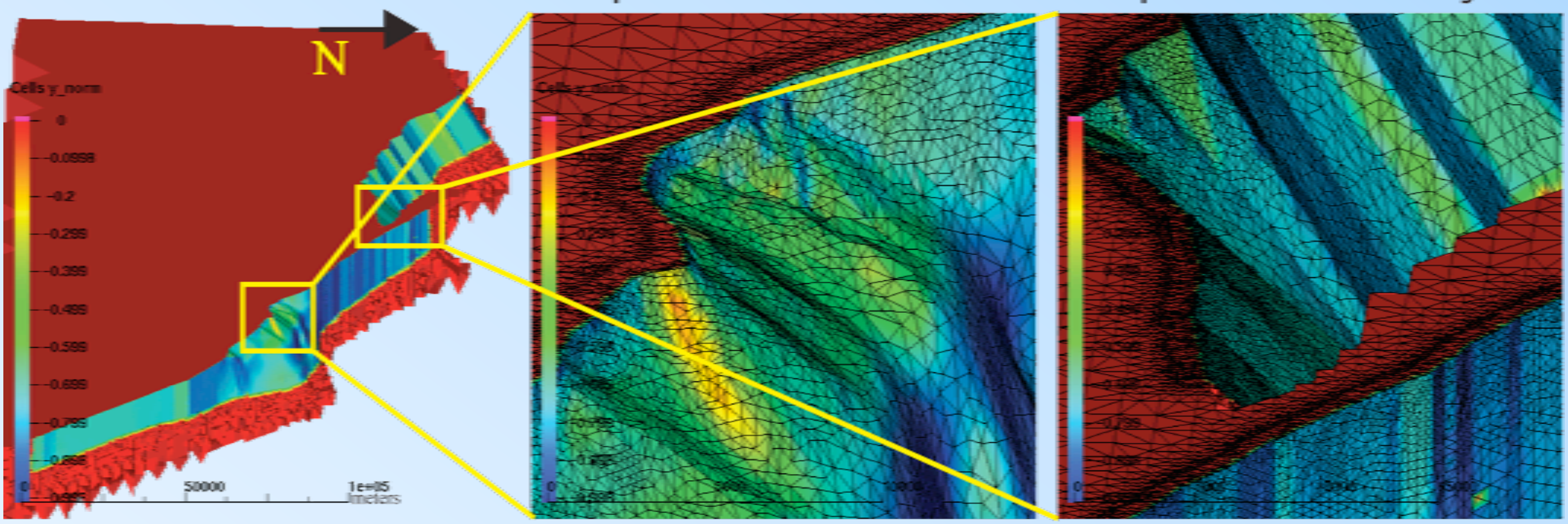


Fault triangulation (blue) inside octree refined mesh with elements near the fault surface removed.



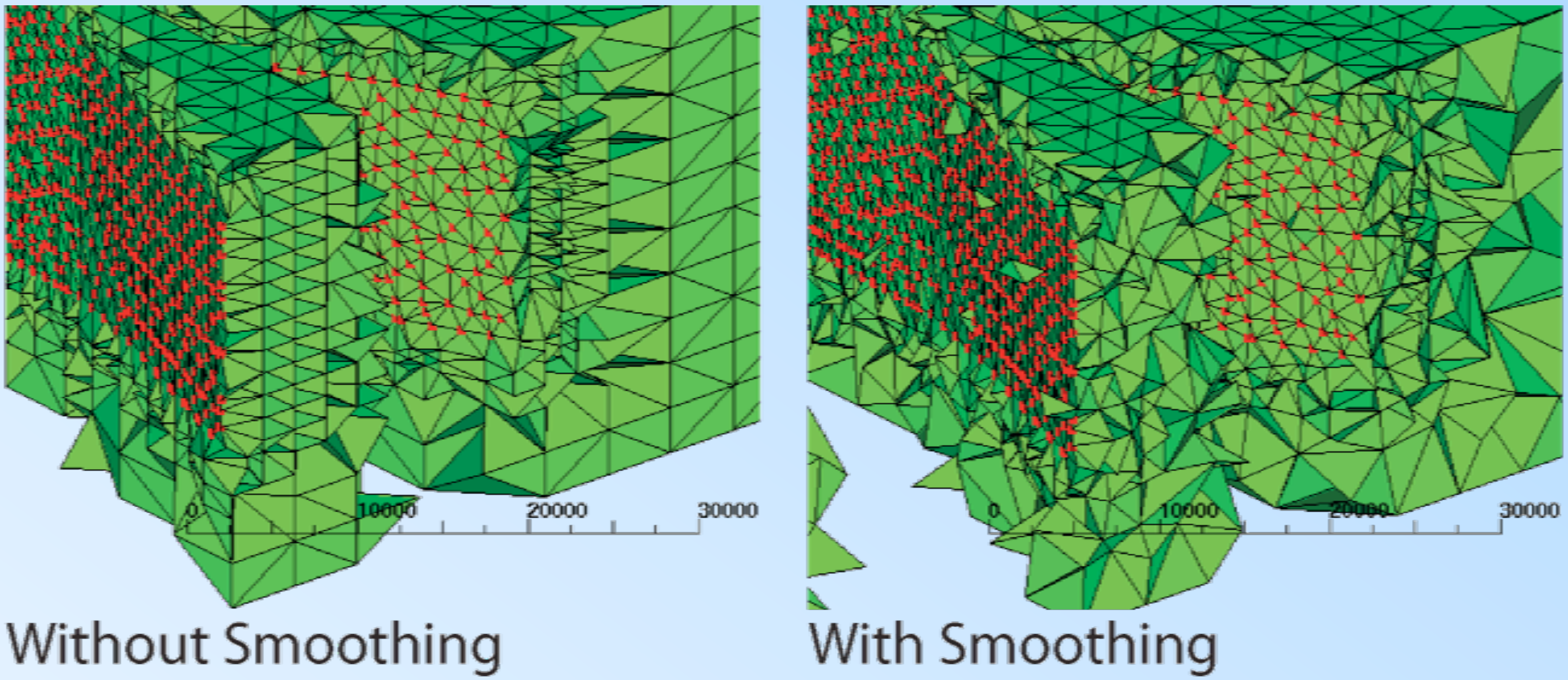
Cut away view after elements near the fault surfaces have been removed.

Connect nodes to form a Delaunay tetrahedral mesh that conforms to the fault surfaces



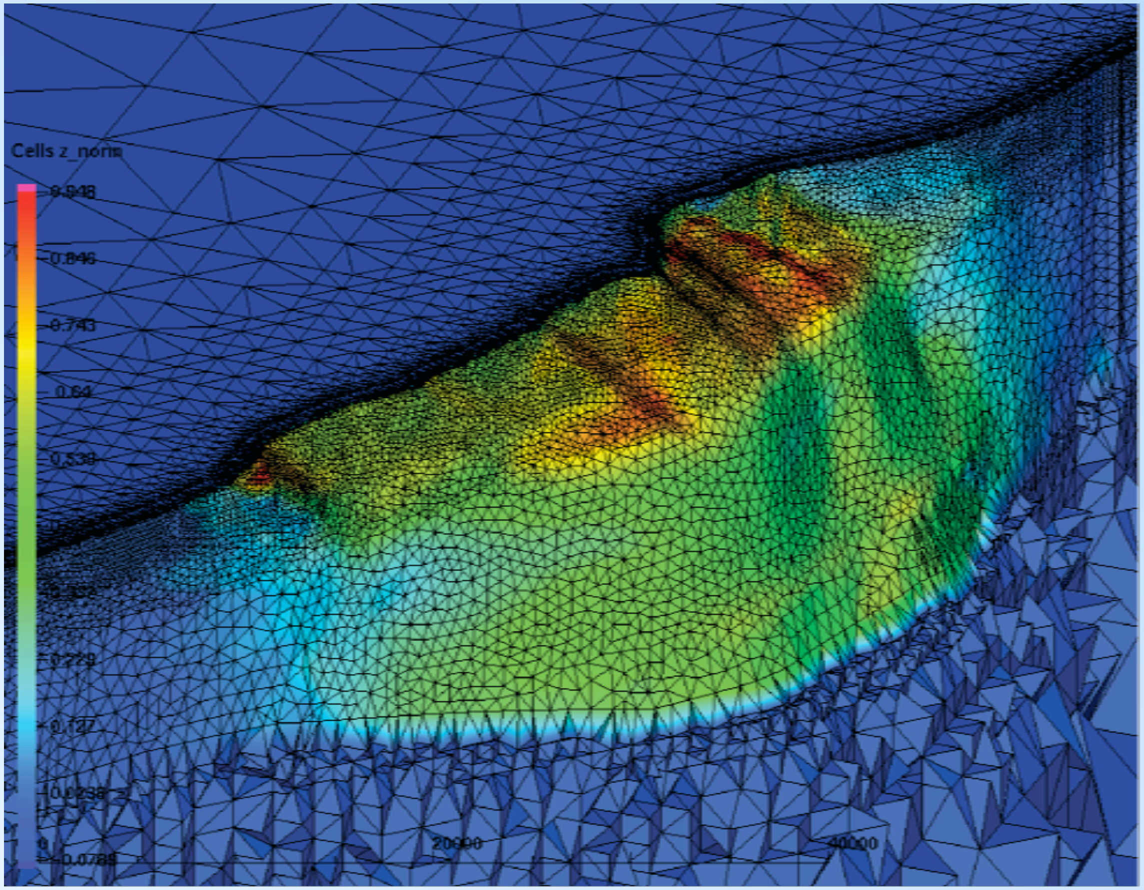
Tetrahedral mesh with elements north of the faults removed. Fault surfaces are colored by the y component of the fault surface normal vector. The lower 900km of the mesh have been removed for viewing. See the theory section for a description of why the fault surfaces emerge from a Delaunay tetrahedralization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

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.

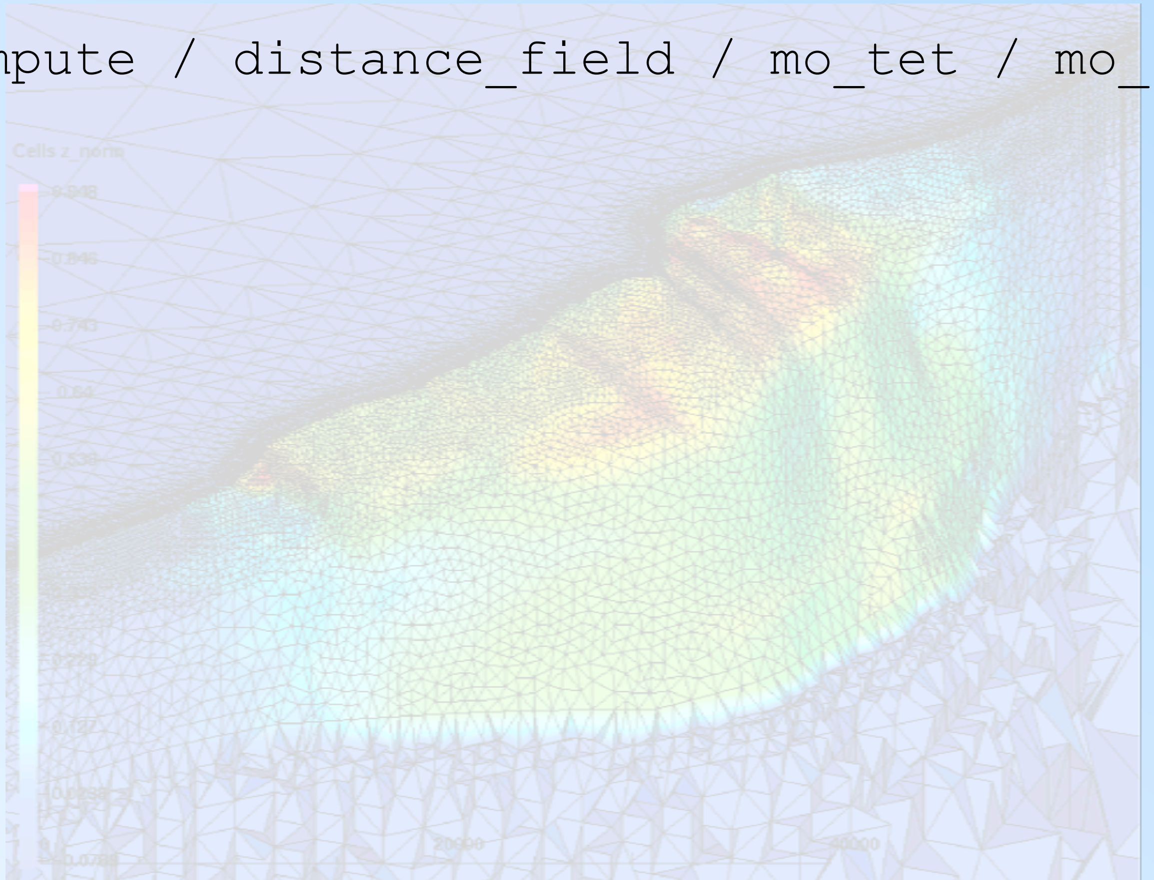
Compute attributes necessary for setup, initial and boundary conditions



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.

Compute attributes necessary for setup, initial and boundary conditions

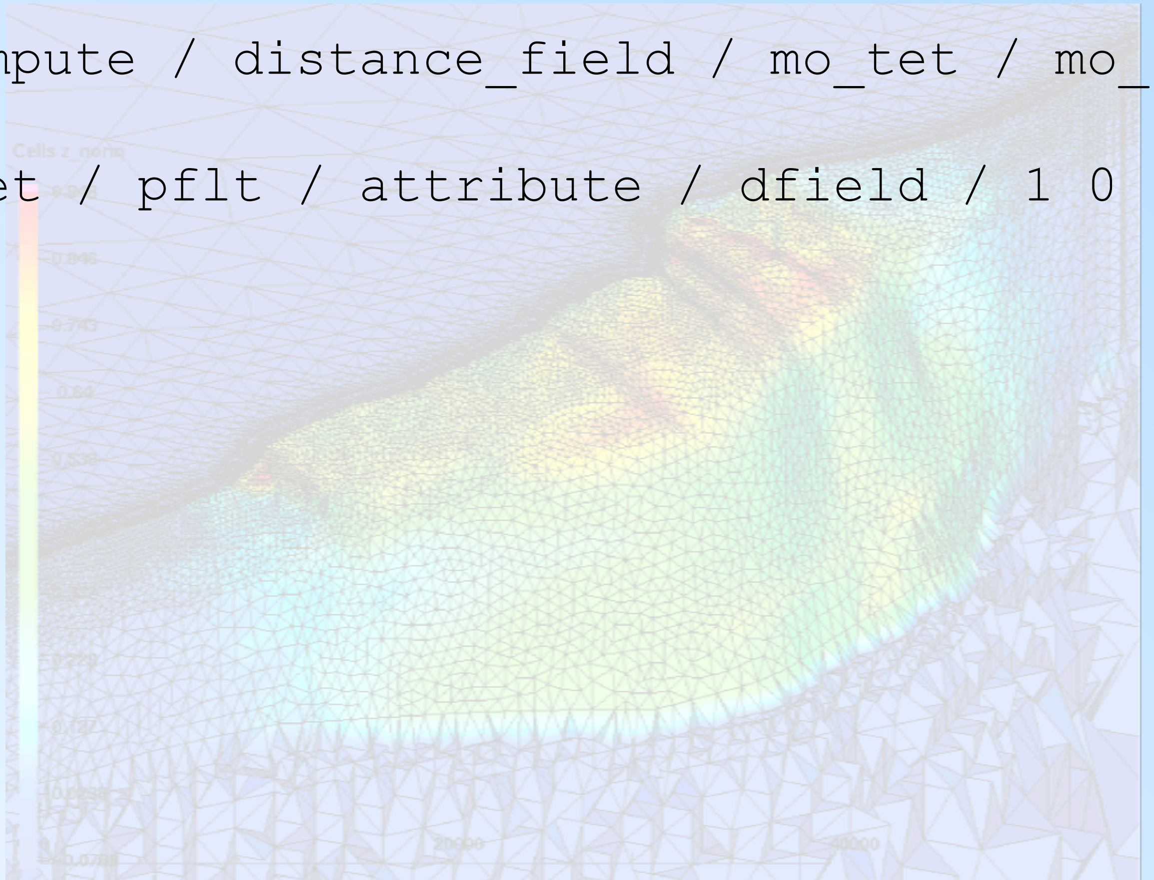
```
compute / distance_field / mo_tet / mo_fault / dfield
```



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.

Compute attributes necessary for setup, initial and boundary conditions

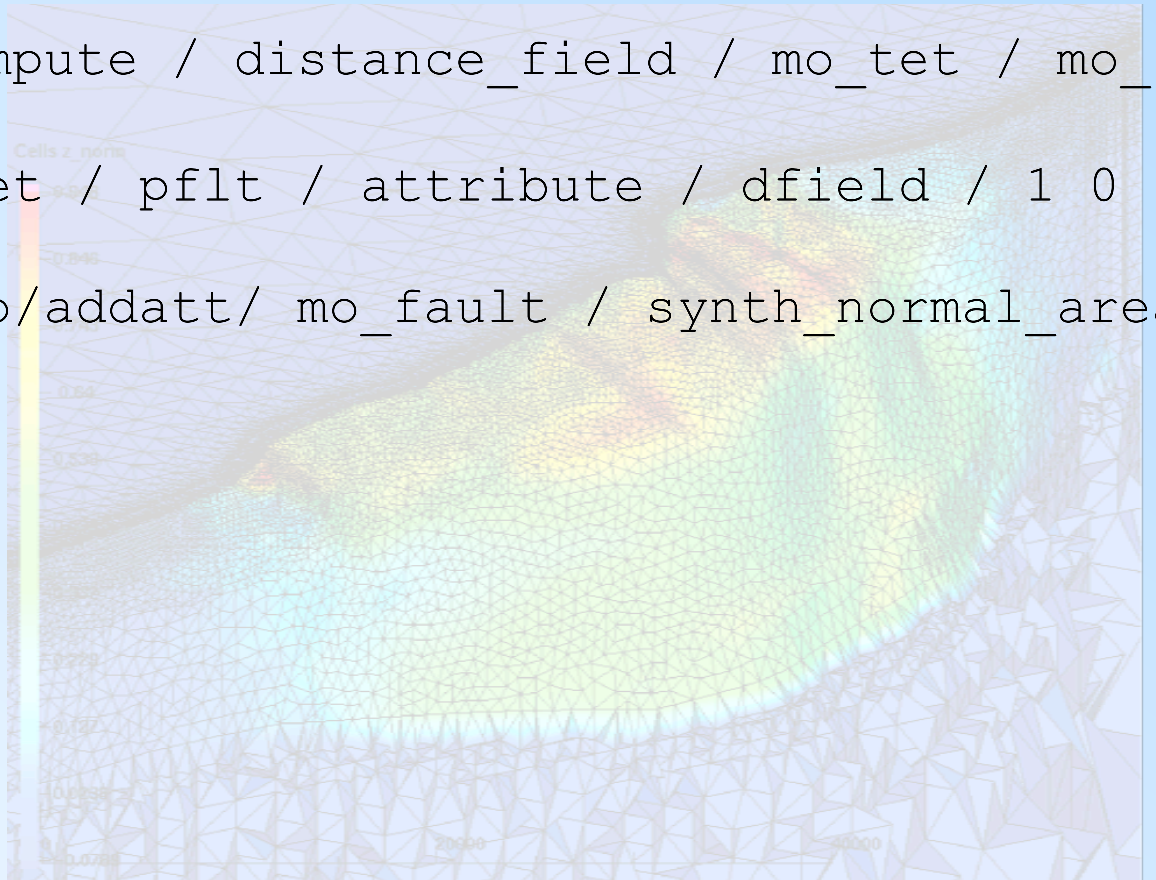
```
compute / distance_field / mo_tet / mo_fault / dfield

pset / pflt / attribute / dfield / 1 0 0 / 1.0 / lt
```



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.

Compute attributes necessary for setup, initial and boundary conditions

```
compute / distance_field / mo_tet / mo_fault / dfield

pset / pflt / attribute / dfield / 1 0 0 / 1.0 / lt

cmo/addatt/ mo_fault / synth_normal_area
```



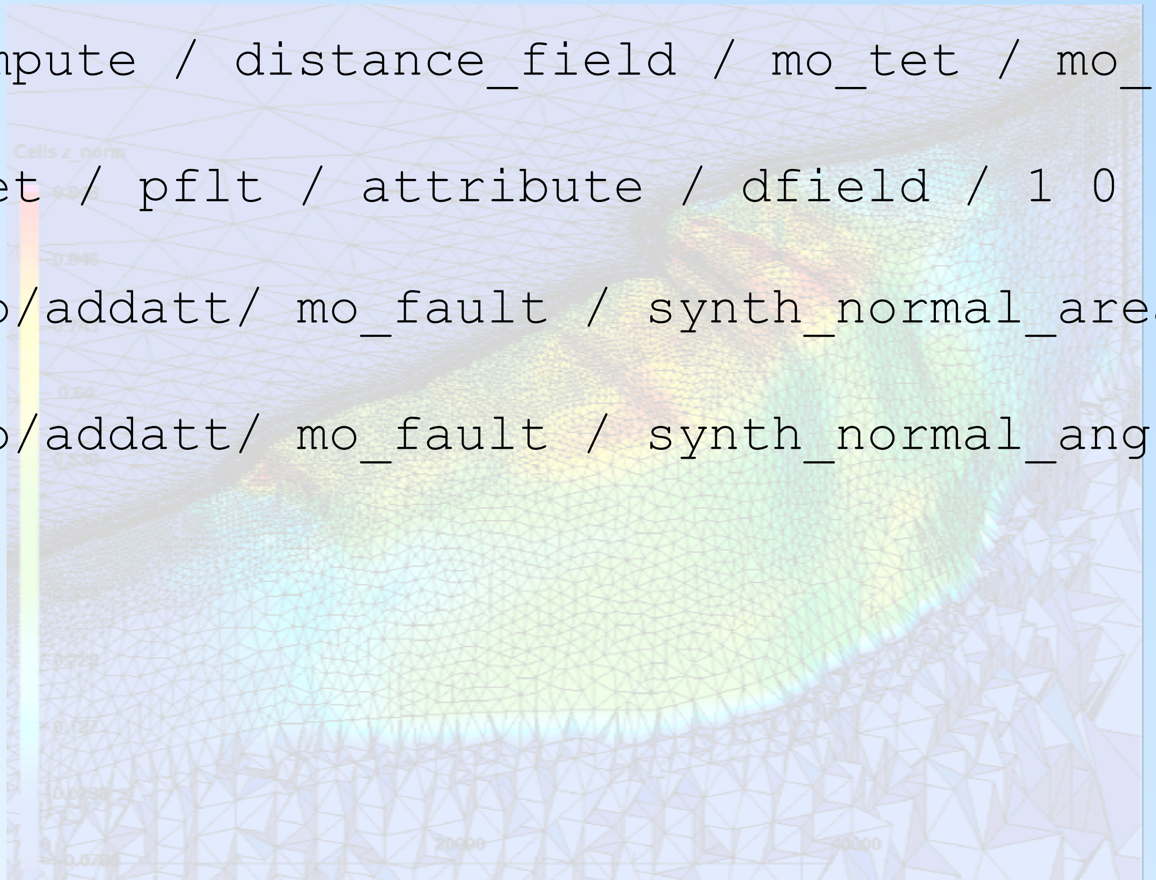
e.g. The normal vector to each node of fault surfaces is computed and output for use in setting boundary conditions. The Z component is show on the San Andreas fault. Z_norm=1 is a horizontal surface, Z_norm=0 is a vertical surface.

Compute attributes necessary for setup, initial and boundary conditions

```
compute / distance_field / mo_tet / mo_fault / dfield

pset / pflt / attribute / dfield / 1 0 0 / 1.0 / lt

cmo/addatt/ mo_fault / synth_normal_area
```

or

```
cmo/addatt/ mo_fault / synth_normal_angle
```



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.

Compute attributes necessary for setup, initial and boundary conditions

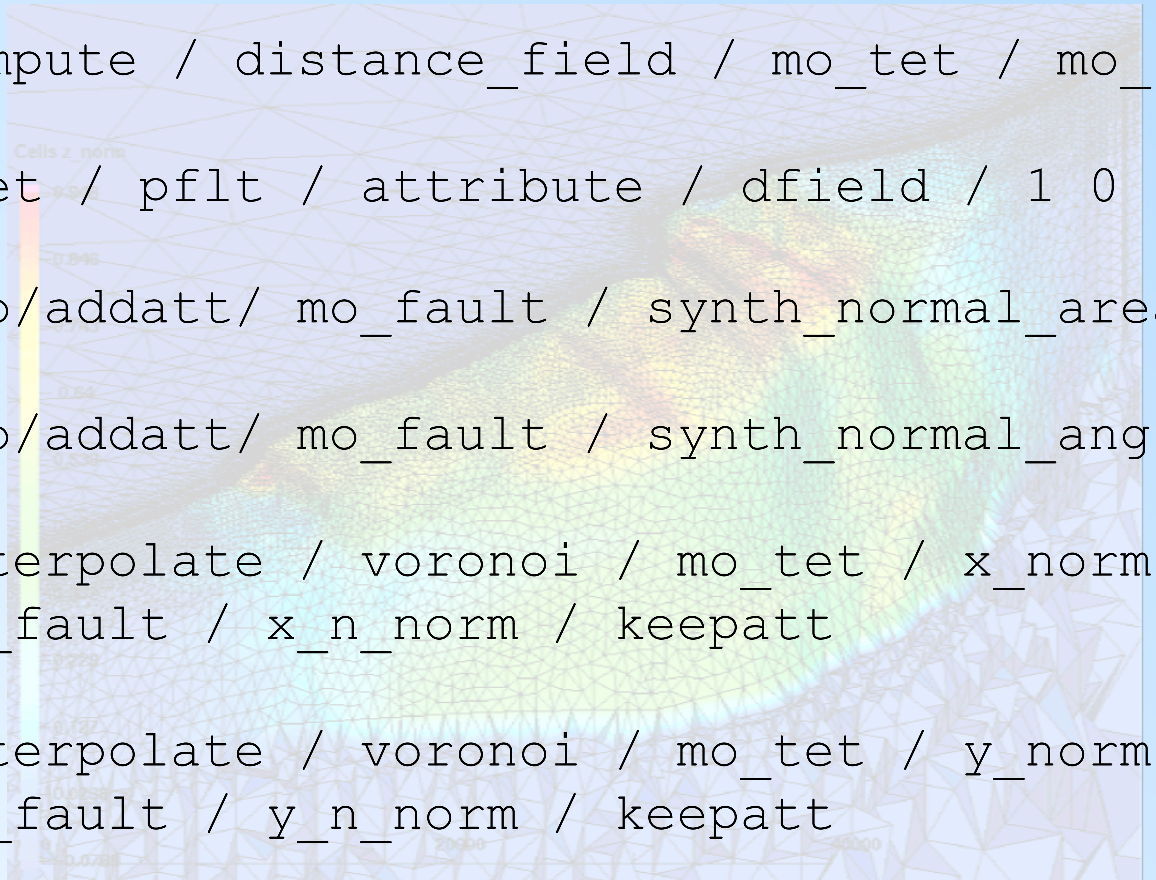
```
compute / distance_field / mo_tet / mo_fault / dfield

pset / pflt / attribute / dfield / 1 0 0 / 1.0 / lt

cmo/addatt/ mo_fault / synth_normal_area
```

or

```
cmo/addatt/ mo_fault / synth_normal_angle

interpolate / voronoi / mo_tet / x_norm / pset get pflt /
mo_fault / x_n_norm / keepatt
```

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.

Compute attributes necessary for setup, initial and boundary conditions



```
compute / distance_field / mo_tet / mo_fault / dfield

pset / pflt / attribute / dfield / 1 0 0 / 1.0 / lt

cmo/addatt/ mo_fault / synth_normal_area
```

or

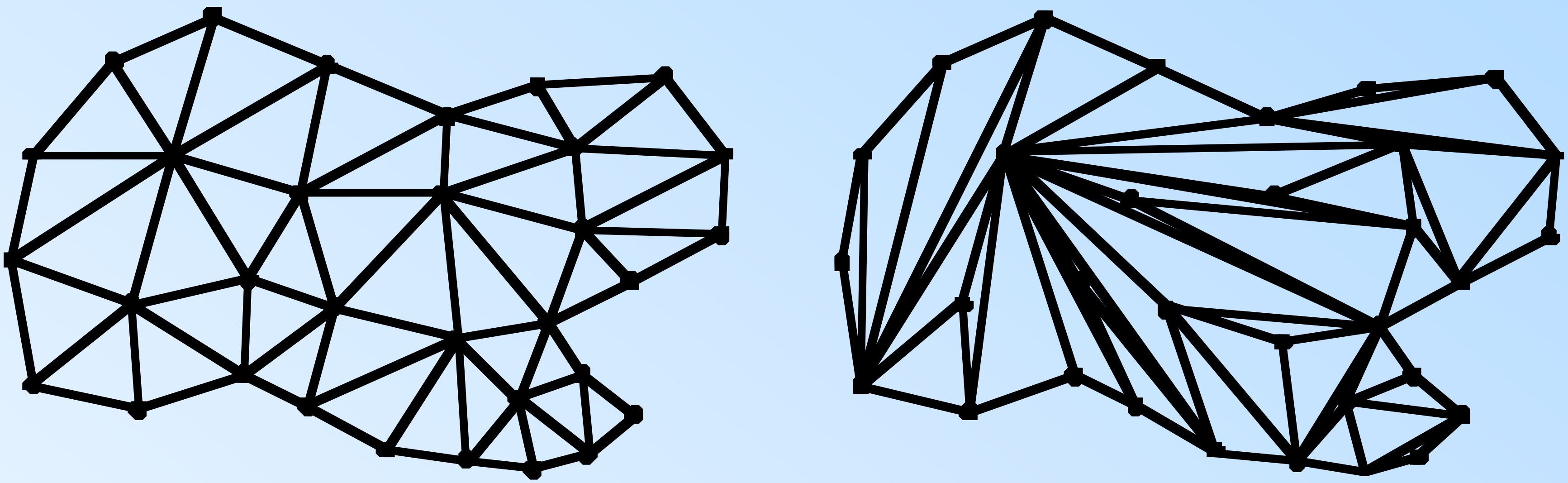
```
cmo/addatt/ mo_fault / synth_normal_angle

interpolate / voronoi / mo_tet / x_norm / pset get pflt /
mo_fault / x_n_norm / keepatt

interpolate / voronoi / mo_tet / y_norm / pset get pflt /
mo_fault / y_n_norm / keepatt

interpolate / voronoi / mo_tet / z_norm / pset get pflt /
mo_fault / z_n_norm
```

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.

Why does this method work?

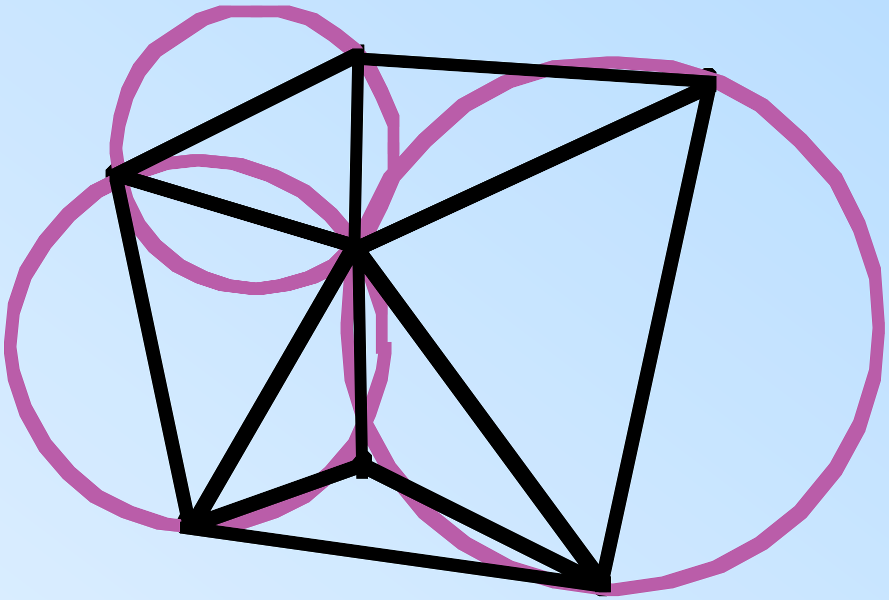
How to connect a point distribution?

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.

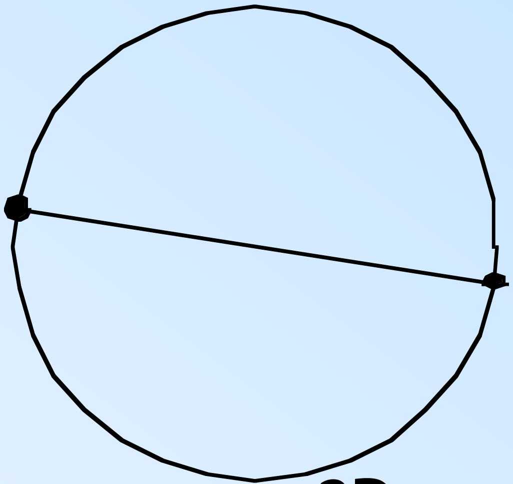
Why does this method work?



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

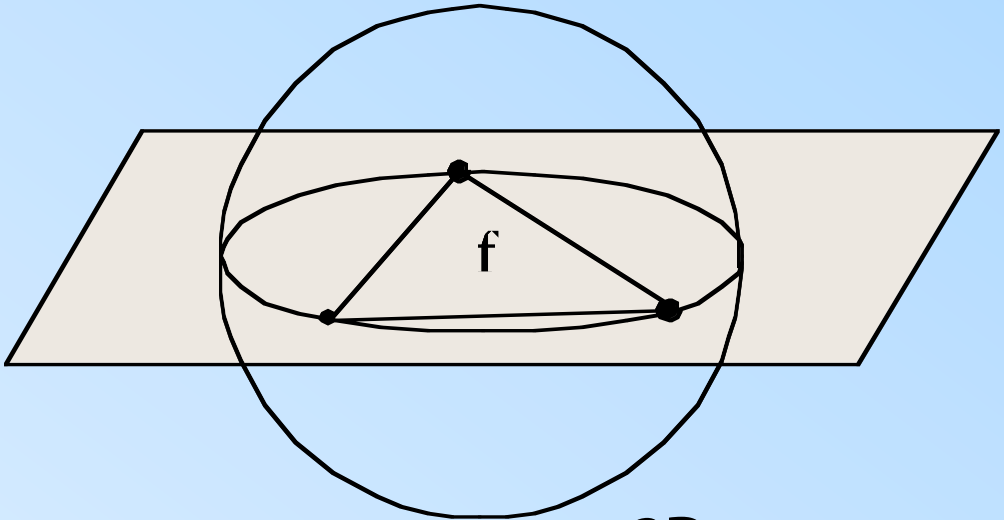
Why does this method work?

Sufficient but not necessary condition for Conforming Delaunay



2D

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



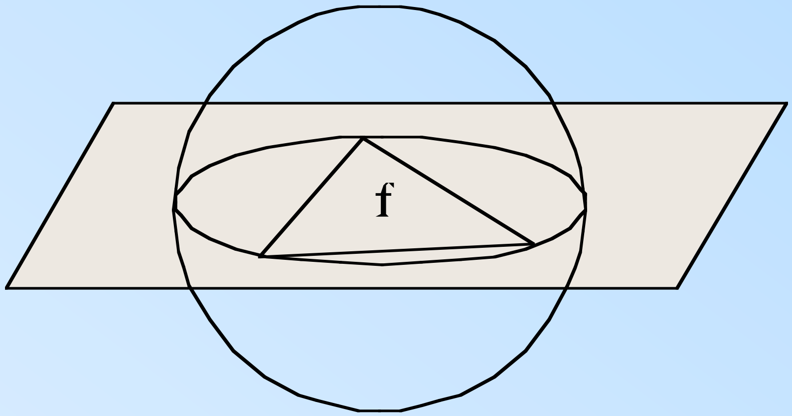
3D

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

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

Why does this method work?

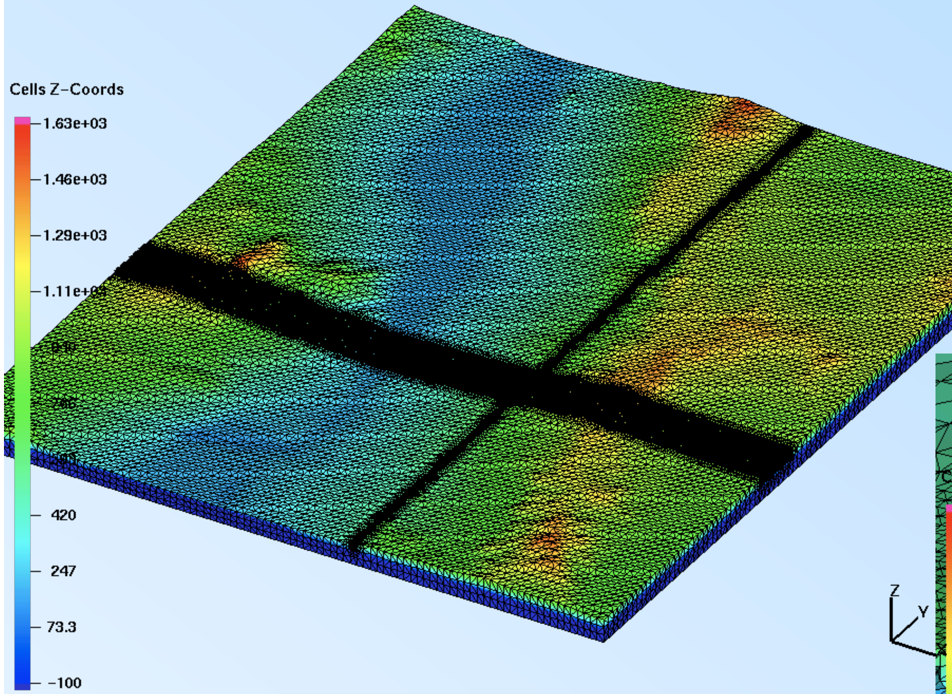
Conforming Delaunay Tetrahedralizations

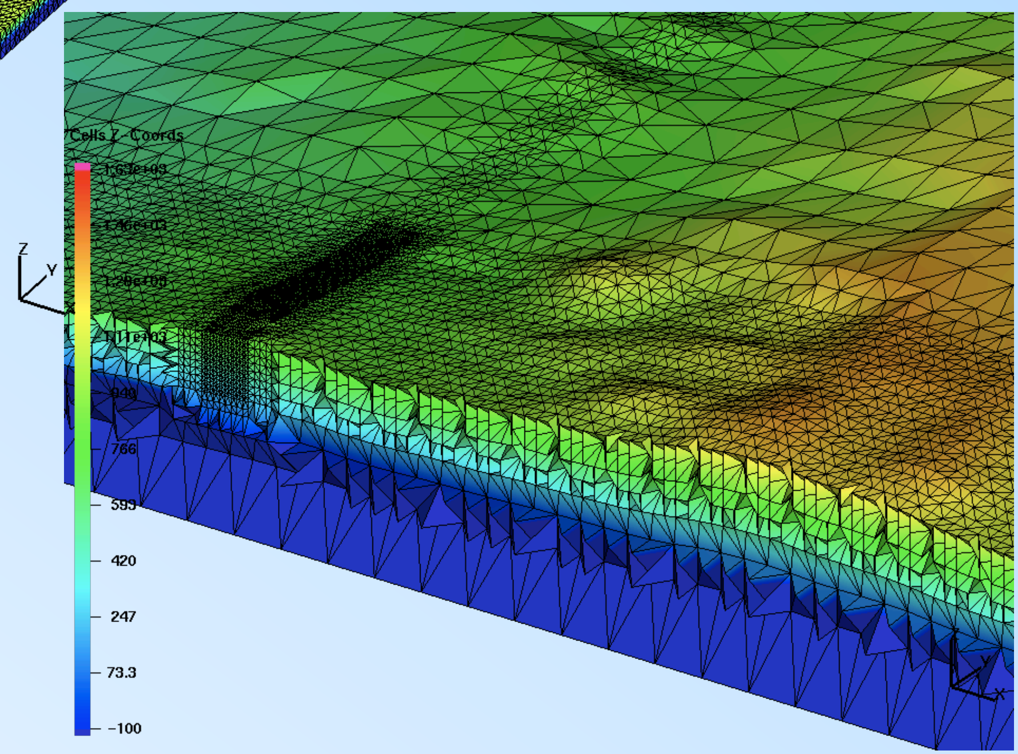


Lemma: A triangular face f of a triangulated surface with vertex set V is a face in the Delaunay tetrahedralization of V if and only if there exists a sphere passing through the verticies of f containing no points of V in its interior.

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

Conform top of mesh to a DEM



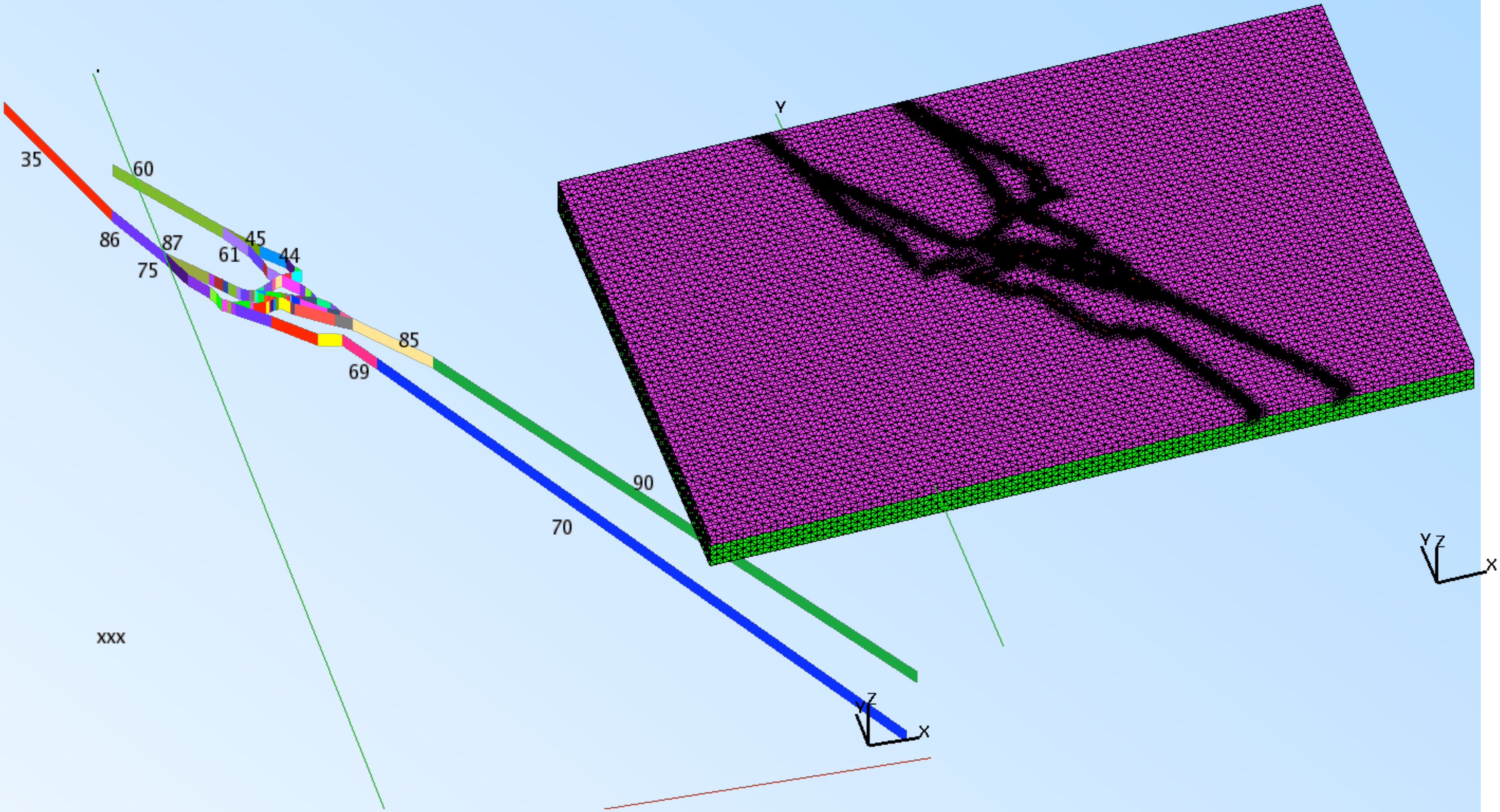


Los Alamos NATIONAL LABORATORY EST.1943

http://meshing.lanl.gov/proj

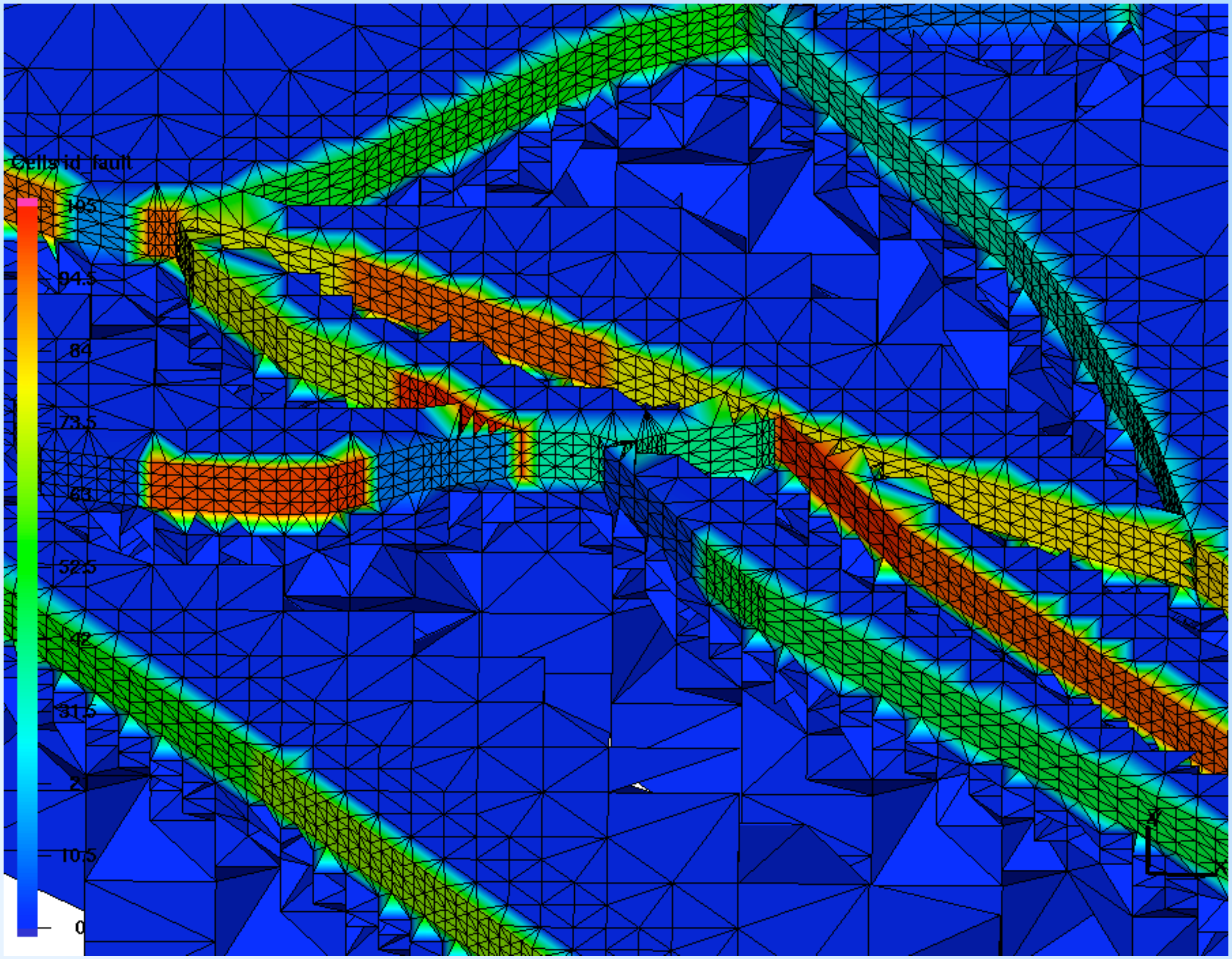
NNSA

Meshing Faults from Meade/Hager Model



http://meshing.lanl.gov/proj

Meshing Faults from Meade/Hager Model

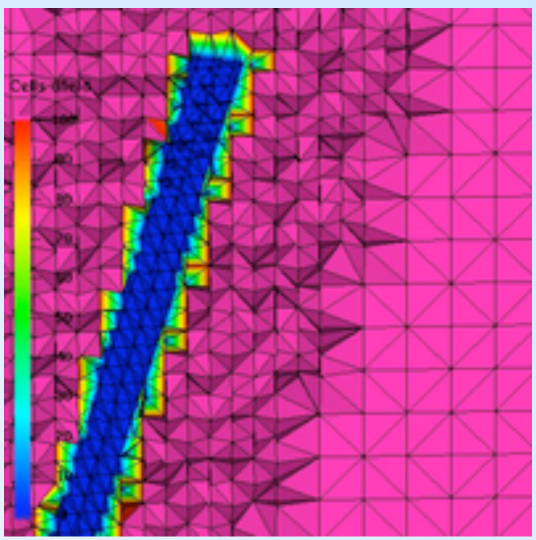


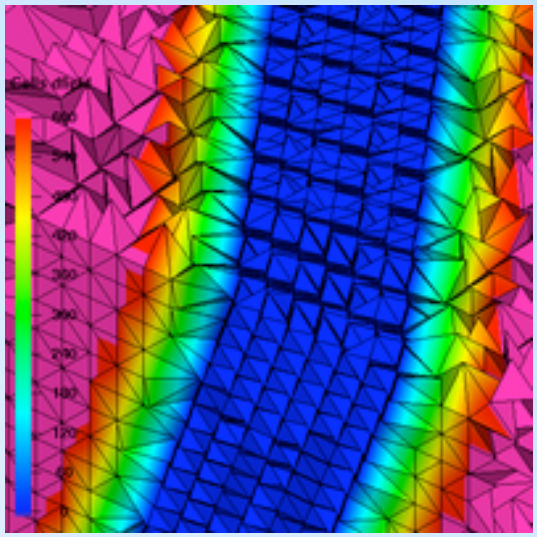
http://meshing.lanl.gov/proj

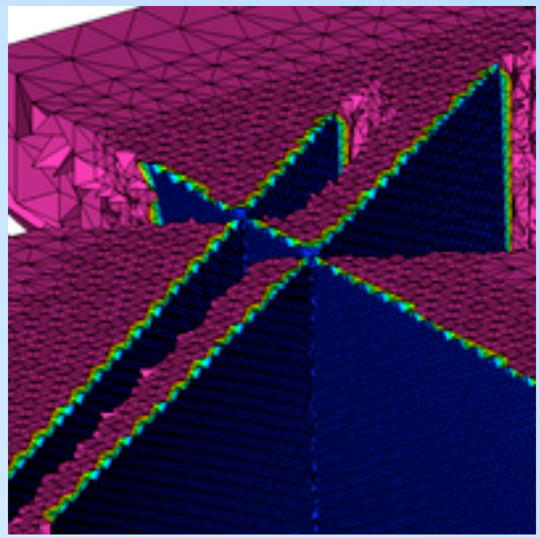
Other Examples for Crustal Deformation

http://meshing.lanl.gov/proj.html

http://meshing.lanl.gov/proj/crustal_dynamics_CFEM/
LaGriT_Mesh_Generation_Demos_CFEM.html



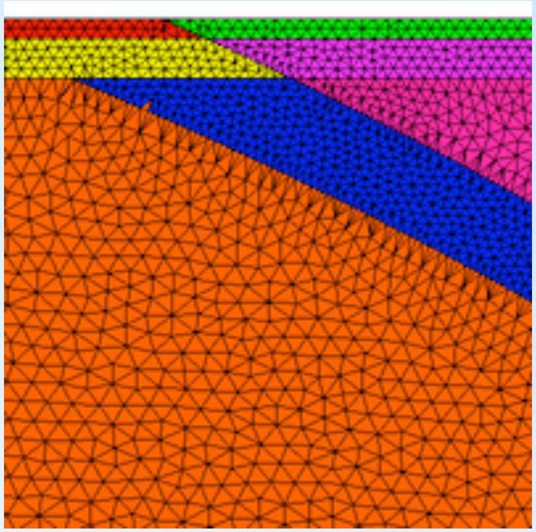


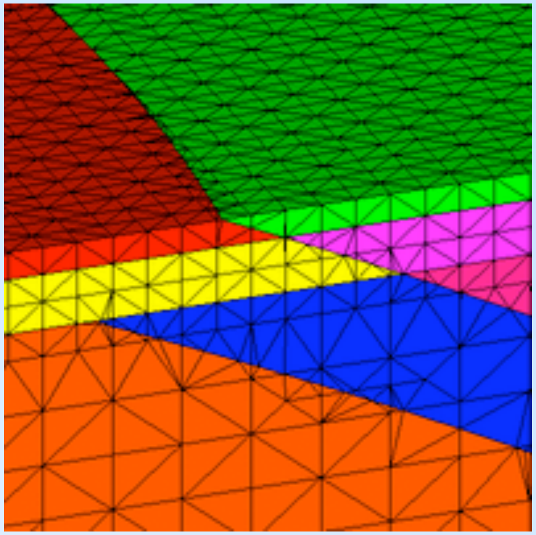


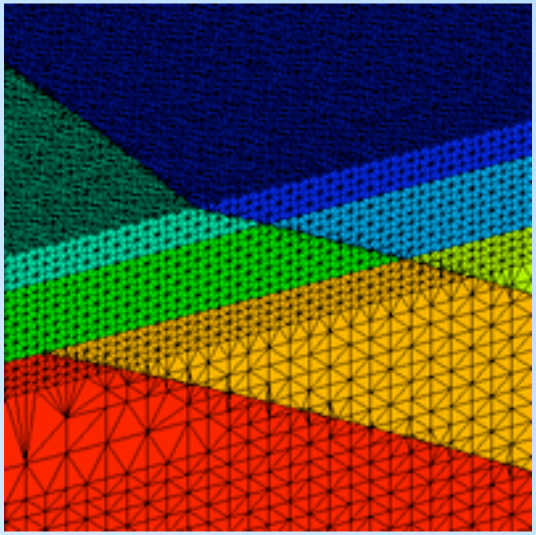
Other Examples for Crustal Deformation

http://meshing.lanl.gov/proj.html

http://meshing.lanl.gov/proj/crustal_dynamics_CFEM/
LaGriT_Mesh_Generation_Demos_CFEM.html







Other Examples for Crustal Deformation

http://meshing.lanl.gov/proj.html

http://meshing.lanl.gov/proj/crustal_dynamics_CFEM/
LaGriT_Mesh_Generation_Demos_CFEM.html