Project Report on "Tests of a New Automated Method for Remeshing the CFM for Use by Earthquake Simulators"

INTRODUCTION

This project develops and tests a semi-automated method to convert any raw triangulated version of the Community Fault Model (CFM), into a remeshed form that is appropriate for use by earthquake simulators. The automated method we have tested was proposed at a 2017 workshop (http://www.scec.org/workshops/2017/cfm) and involves fitting a spline surface to each fault segment of the CFM. After fitting, the spline surface could be meshed at any desired resolution. This is a process that should be capable of automation, an assertion we have been testing in this pilot project. A version of the spline-fit fault surface could either include the Earth surface topography that exists in the CFM or each spline could be cut at its intersection with a horizontal plane at sea-level. For half space-based earthquake simulation methods the latter is necessary, but in the future if Greens' functions were generated using the finite element method, then the topography could be included. Our plan has been to test this spline-based approach by remeshing nine faults of the CFM and carrying the process all the way through to using them in an earthquake simulator. The idea of choosing nine faults is that we will work on the faults in order of difficulty, starting with 3 faults that we anticipate will be easy to remesh (e.g. approximately planar vertical faults, non-intersecting), three that will be intermediate in difficulty (e.g. complex non planar geometry), and three that will be challenging (e.g. have multiple subsurface intersections and are non-planar).

ACTIVITIES AND APPROACHES UNDERTAKEN

Fault selection (and manual-meshing times for first two categories by Scott Marshall using 3DMove and some Perl scripts).

Nine faults were chosen by John Shaw, Andreas Plesch, and Scott Marshall. They are:

EASY SINGLE FAULT CATEGORY

1) Fontana Seismic Zone *(31 minutes by hand)*; CFM Name: PNRA-CRSF-USAV-Fontana_Seismicity_lineament-CFM1; Comments: This is a simple approximately square shaped planar fault that has no intersections. The only complication is that that the native mesh is rather poor in triangle quality.

2) Garlock Western Segment *(30 minutes by hand)*; CFM Name: GRFS-GRFZ-WEST-Garlock_fault-CFM5; Comments: Garlock West is close to planar and the native CFM mesh is coarse, but pretty decent. The only complication is that the western end connects to the SAF (SAFS-SAFZ-1857-San_Andreas_fault_rupture-CFM5).

3) Southern San Cayetano Fault *(35 minutes by hand)*; CFM Name: WTRA-NCVS-VNTB-Southern_San_Cayetano_fault-steep-JHAP-CFM5; Comments: This is our one dip-slip fault in the "easy" list. The S. San Cayetano fault just touches the San Cayetano fault at one point and it links to the west with the Ventura fault (WTRA-NCVS-VNTB-Ventura_fault-steep-CFM5) but, otherwise, it intersects no other faults.

MODERATELY DIFFICULT FAULTS CATEGORY

1) Northridge *(20 minutes by hand)*; CFM Name: WTRA-ORFZ-SFNV-Northridge-Frew_fault-CFM2; Comments: This is the Northridge thrust, which has a complex 3D geometry. This is included to make sure the automated algorithm can handle a complex non-planar blind fault.

2) Santa Susana *(18 minutes by hand)*; CFM Name: WTRA-SSFZ-MULT-Santa_Susana_fault-CFM1; Comments: The Santa Susana fault has a corrugated geometry with some rather complex tears in the near surface and a complex surface fault trace.

3) SAF (Banning segment) *(36 minutes by hand)* & Southern SAF (Coachella segment); CFM Names: SAFS-SAFZ-MULT-Garnet_Hill_fault_strand-CFM4 SAFS-SAFZ-COAV-Southern_San_Andreas_fault-CFM4; Comments: These two faults should link together. This is included to see if we can get these two faults to connect. Like most connected faults in the CFM, the meshes to not exactly connect in 3D space, so our algorithm should be able to deal with this.

DIFFICULT FAULTS CATEGORY

1) Santa Monica Steep & Santa Monica Thrust; CFM Names: WTRA-SFFS-SMMT-Santa_Monica_fault-steep-CFM5 WTRA-SBTS-SMMT-Santa_Monica_thrust_fault-CFM1; Comments: These both comprise the

Santa Monica fault zone. One CFM segment is steep with tears at the surface, the other is a shallow dipping thrust. Getting the intersections between these surfaces will be a challenge.
2) Newport-Inglewood, San Joaquin Hills, & Compton; CFM Names: PNRA-NIRC-LABS-Newport-Inglewood_fault-dip_w_splays-split-CFM5 PNRA-CSTL-SJQH-San_Joaquin_Hills_fault-truncated-CFM3 PNRA-CEPS-LABS-Compton-Los_Alamitos_fault-CFM2; Comments: These three faults intersect at depth. The NI-San Joaquin Hills intersection is pretty straight forward. The intersection with Compton is complex, as Compton has offset the NI fault resulting in a tear in the surface. This is included because it will be a challenge to get these intersections correct.
3) The Puente Hills Thrust (and everything it intersects); CFM Names: too many to list; Comments: If our meshing tool can re-mesh this complex set of faults, it can probably handle anything.

Spline Fitting. Ossian O'Reilly developed a surface tool to fit B-splines to fault surfaces. We settled on the use of B-splines because they are widely used in CAD (computer aided design) software and consequently many tools and techniques exist for working with them. Our surface-fitting tool, together with a technical description of how it functions, is available at https://github.com/ooreilly/splinefit. The approach used by the tool is to project the CFM mesh onto the best fitting plane to the fault and after the projection, divide the fault boundary into four segments and then fit each boundary segment with a B-spline curve using a linear least squares approach. Each boundary maps to the side of a rectangle. The fitted surface is constructed by deforming the rectangle so that it conforms to the previously obtained boundary curves, the interior mesh points moving appropriately in the plane of the rectangle. Since the original surface is in general not flat, it is also necessary to deform the fitted surface in the surface normal direction to obtain a good approximation of the original shape. We used linear least squares for this fitting step as well. After developing the tool, Ossian tried it on many of the nine fault surface groups and made improvements to the tool as he encountered issues with certain of the faults.

Meshing of the Spline-defined Surfaces. Michael Barall used Cubit to mesh the fault surfaces that Ossian had described with splines. Michael wrote a conversion tool to import the output from the B-spline surface fitting tool into the Cubit mesh generator. This conversion tool, available at https://github.com/mbarall/cspline, supports a minimal subset of the IGES exchange file format for CAD data. Michael investigated various approaches to modify the meshes in situations where faults intersected, as discussed in Figure 1.

Manual remeshing of the Fault Surfaces. Scott Marshall started with the original CFM surfaces in our "Easy" and "Moderately Difficult" groups and manually remeshed them, keeping track of the time involved, in order to compare the effort involved in doing this manually with that involved with eventually using our developing "automatic" spline-based method. While this method produces meshes of good quality with no issues for modelers, it is time consuming (~30 mins per fault) and if a new resolution were needed, the entire processes would need to be redone.

RESULTS

In summary, we have had significant success in developing a spline-based method for remeshing the CFM, but some significant problems remain, most of which are likely soluble with more effort than we were able to devote at the level of support and time available through this grant. Because Cubit can cut the spline-defined surfaces at zero depth, we can generate meshes without as well as with the Earth's topography as needed. We have been successful in remeshing the three faults in the "easy simple fault" category. Of the other six faults, we have focused much of our attention on two of the faults in the "moderately difficult faults" category, the Northridge thrust and the Garnet Hill fault strand within the Banning segment of the SAF. The non-planar and irregular outline of the Northridge thrust and the fault intersections involved in the Garnet Hill strand of the SAF have presented us with problems which we believe we may have solutions for, but have not yet completely solved. Consequently, we have not yet moved to the three faults in the "difficult faults" category, because the problems they present require at a minimum being able to deal with the problems presented by the "moderately difficult faults" category.

The CFM representation of the faults comes in two forms, the native description that involves triangles that vary greatly in size and shape as a consequence of being derived from a wide range of original geological and geophysical fault data sets, and a semi-regularized description that uses more equi-dimensional triangles of similar size that have been constructed by John Shaw and Andreas Plesch as part of SCEC's CFM support grant. They do this by applying a finite element program mesh algorithm available in the GoCAD software package. We have found that the regularized description provides better input for the spline-fitting stage of our automated remeshing process. It is important to note that the regularized description of the CFM is itself not suitable as input to earthquake simulators because it involves a non-planar Earth's surface and because intersecting faults are represented independently and therefore inconsistently. Namely, adjacent faults may interpenetrate one another and/or have gaps in situations where they should intersect perfectly, whereas they both should have meshes with conforming nodes.

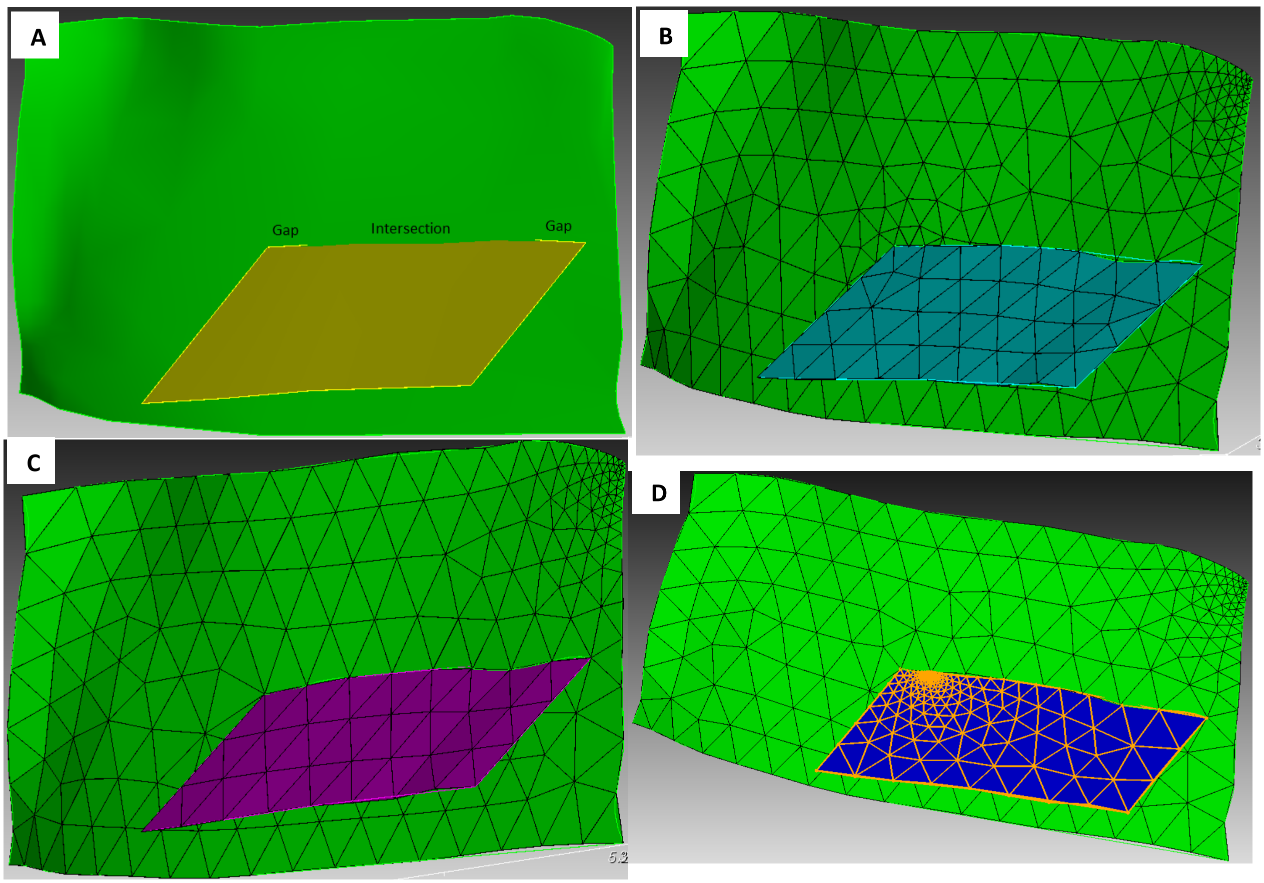

Figure 1. Views of the Garnet Hill fault from slightly different perspectives showing issues that arise when remeshing two faults that intersect imperfectly in the original CFM mesh. **A.** shows that the edge of the smaller fault does not lie on the surface of the larger fault. On the left and right there is a gap between them, whereas in the middle the smaller fault penetrates the larger one and the edge of the smaller fault is behind the larger one. **B.** shows the result of meshing each fault as it exists. Note the lack of conformance of the meshes on each fault at their intersection due to the gap. **C**. Shows the result of manually moving the small fault toward the large one 400 m so the gaps are eliminated, trimming the part of the small one that is behind the large one, and then meshing. The meshes conform nicely, but the fault geometry has been altered by moving the smaller one. **D.** This mesh results from adding a second even smaller surface that extends the small one beyond the large one. The very fine detail in this undesirable mesh results from its dealing with the intersection between the original small surface and the new even smaller surface. None of these solutions to the imperfect joining of the original small and large faults in the CFM is satisfactory, although we believe that an alternative solution at the spline-fitting stage of the process could solve this and other problems. We have not had time to fully investigate this alternative solution, which we describe in the text.

One issue we encountered is that our method of constructing the B-spline surface by deforming it resulted in problems during the mesh generation stage. The deformations tended to produce meshes of poor quality due to the mesh generator trying to follow the highly curved grid lines. We also encountered problems fitting boundary curves and surfaces due to the irregular position and sparse availability of data points. To prevent undesirable oscillations from developing due to solving ill-conditioned linear systems, we added regularization to the both the curve and surface fitting. We also added the option to include more data points in the fitting by inserting points on the surface of the original triangular mesh. With this option, it is important to be aware of that adding too many data points will resolve unrealistic features in the shape of the triangular mesh which are an artifact of the triangles used in whichever CFM representation is used as input.

At the stage of using Cubit to fit meshes to the spline surfaces we have found that poor meshes are often created due to the two types of problems referred to above, 1) the severe distortion of the inherently rectangular spline surface needed to fit a non-rectangular fault, and 2) the gaps and overlaps that result when two intersecting faults in the CFM do not intersect perfectly. The latter is illustrated in Figure 1.

We believe the potential solution to both of these problems is to make the following change in our procedures. Currently we define, as well as we are able, four "edges" of a CFM fault and map them to the four boundaries of a rectangular spline-defined surface, resulting in severe distortions of the spline description for faults that are non-rectangular. In this approach the entire fault is mapped onto the entire rectangle and the rectangle and fault mesh is then distorted so that the four no-longer rectangular edges lie along the four "edges" of the non-rectangular fault that is presented to the mesh generator in Cubit. Instead we propose to fit the fault by cutting out its shape from the best fitting plane. In this approach, the best fitting plane is deformed in the normal direction to approximate the fault mesh in this direction. The boundaries of the fault mesh are fitted using B-spline curves and these curves are projected onto the fitted surface itself. The projected boundary curve splits the surface into an inner and an outer part. The outer part is mostly flat, except for a user-defined threshold that extends the shape of the fault in the inner region to the outer region by an extrapolation. This extrapolation will allow us to close small gaps between intersecting faults. This is illustrated in Figure 2. The spline description of the rectangular surface will then include splines that describe the area of the projected fault as well as ones that define the surrounding mostly-flat area. The description of the fault to be passed to the mesh generator process will involve the rectangular B-spline surface as well as the B-spline curves that define the fault boundaries. Both Cubit and the IGES file format can accept such cutting curves.

This approach should solve both the problems described above. 1) For single faults without intersections with other faults, the spline description of the fault surface will not have the distortions of our current method. The fault-bounding curve can be used in Cubit to cut out only the fault-surface portion of the

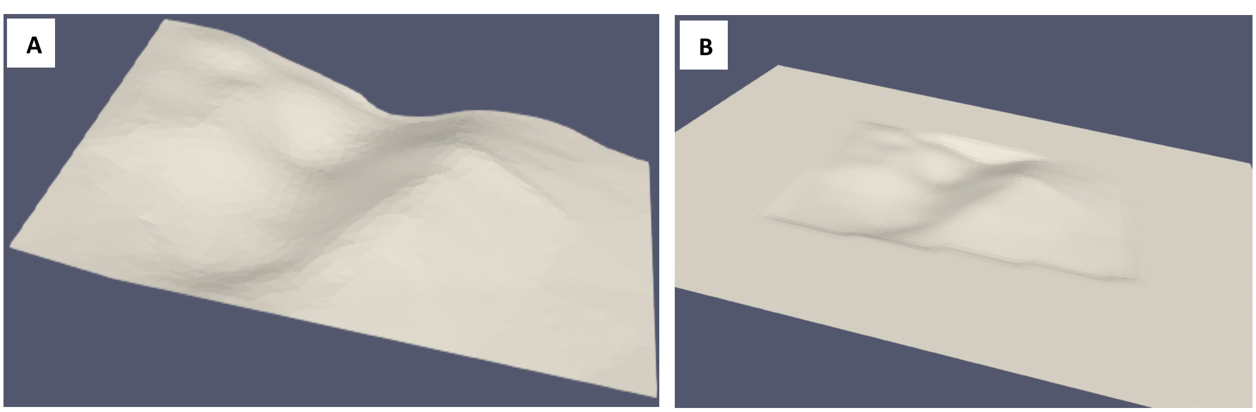

Figure 2. A. Original CFM mesh of the Northridge fault. **B.** Fitted B-spline surface using a plane. The final fitted model would be obtained by cutting away parts of the plane using fitted B-spline curves for the boundary of the original mesh.

spline description of the rectangle, and then Cubit can mesh only the remaining fault surface. 2) For faults that intersect others imperfectly, the spline description of the fault can be given by the spline description of the rectangle as cut out by the bounding curve, as well as any needed surrounding portions of the spline description of the rectangle that extends the fault up to the intersection with other faults. This eliminates any troublesome gaps without shifting the location of any faults. Because Cubit can compute intersections between spline surfaces, and construct a mesh that respects the intersections as is shown in Figure 1C, we can mesh properly located intersecting faults that will meet perfectly.

The times required for manual meshing of six of our nine faults is given in the list of faults above. It is too early to know how automated this system can become and so how much effort would be involved in doing the faults in an automated or nearly automated manner to compare with the manual times. Figure 3 shows a comparison of three meshes for the Northridge fault at different densities from three approaches.

Because we didn't get decent meshes for all of our faults we didn't try running the RSQSim earthquake simulator on them. We do not anticipate that this would be difficult to do.

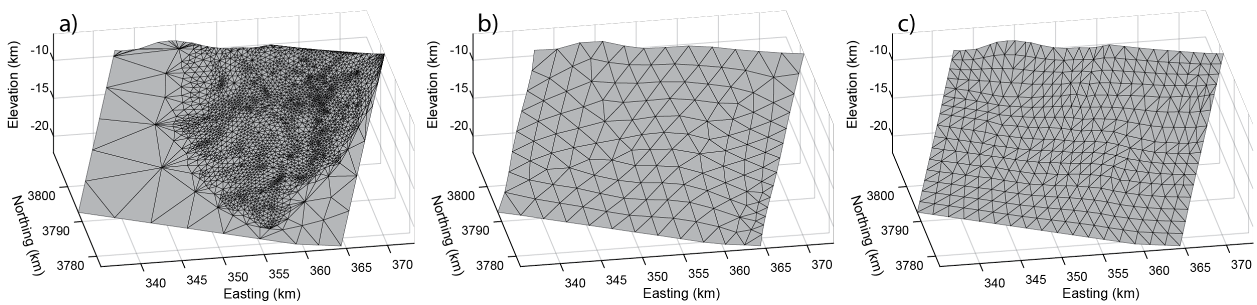

Figure 3. Three meshes of the blind Northridge thrust fault. **a)** Native CFM5.2 mesh. The heart shaped part of the fault with the finer mesh was defined using relocated hypocenters and is of higher-resolution than the rest of the fault. **b)** Our automated mesh produced by the spline-based method and Cubit, using part a as input. **c)** A mesh made manually by Scott Marshall starting from **a.**

FUTURE DEVELOPMENT USING SPLINE-BASED AUTOMATED REMESHING

We have not managed to create an automated procedure for meshing the CFM through this project. However, we have made considerable progress and as described we have a promising idea for improving the method so that it should create more satisfactory fault meshes. How automated this will allow the approach to become remains unclear. Whether we can do some of the remaining work gratis in our spare time is unclear. Exploring the method may require a subsequent small grant similar to this one.

Future EQ simulators are likely to need even higher mesh resolutions. Given the number of faults in the CFM, the more automated the process can become the better. Certainly our progress to date is sufficient that we would not propose manually remeshing the CFM for use by simulators at this time. With manual meshing, changing the density of the mesh or whether the top surface is at zero depth or follows the Earth's topography requires some repetition of previously completed work. Any such repetition is certainly much more work than simply selecting parameter values for input to an automatic system. In addition, if automated or nearly automated remeshing is possible it could hopefully be simple enough that it could be reapplied to any faults that are changed or added in future versions of the CFM, thereby satisfying the desire to make earthquake-simulator-compatible versions that are consistent with future releases of the CFM. Manual remeshing requires not only software tools to assist the process, it requires considerable experience to do it efficiently and is not work that anyone can or would want to do routinely. To our knowledge Scott Marshall and Michele Cooke's research groups are the only people that have the software tools and experience required to contemplate any massive manual remeshing.