Reconciling the Geological Framework (GFM) with the Community Fault Model (CFM)

Laurent Montesi[1]; Andreas Plesch[2]; John Shaw[2]
[1]University of Maryland, College Park (montesi@umd.edu); [2]Harvard University

1. Abstract

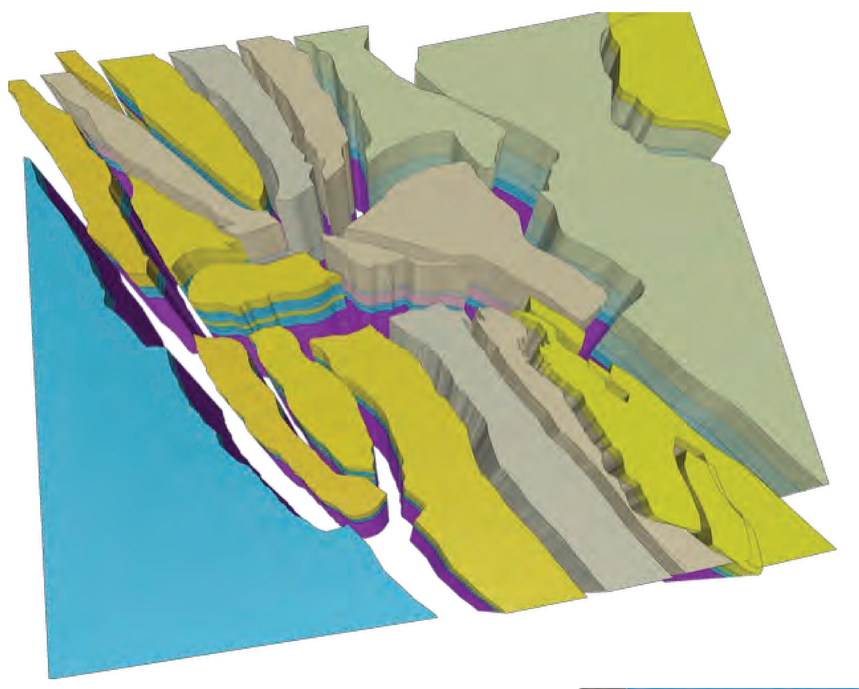
The Geological Framework (GFM) describes California as a collection of lithotectonic blocks, each with an ascribed geological column through the crust. This model intends to provide sufficient lithological information to assign material properties, such as rheological parameters, throughout the lithosphere, with the ultimate objective of informing large-scale models of fault loading and lithosphere evolution. To achieve that objective, additional material properties must also be ascribed to the interfaces between blocks. While several of these boundaries are linked to aeromagnetic anomalies and other geological contacts, some correspond to known faults. Thus, there is an opportunity to link the GFM block boundaries to the Community Fault Model (CFM) as a step towards an integrated model of the region.

We present here preliminary results, focusing on the development of Python modules allowing this model to be built semi-automatically. The purpose of these modules is to enable quick model rebuilding when the CFM is updated, or when information is available from an alternative fault database. We want the utilities to be open-source and require no specific commercial license. This project required defining basic operations, including translating tsurfs to a Python CFM class, extracting the edges of these surfaces for subsequent mesh building, building a new surface between these meshes to fill gaps between mapped faults, as the GFM blocks must be watertight, cutting CFM surfaces at the intersection between blocks, handling non-fault surfaces in the same framework, and extending surfaces at depth under a series of assumptions. Moreover, we are developing a logic tree describing the GFM blocks, the surfaces bounding them, and the operations necessary to construct their boundaries. These facilities will help generate alternative versions of the GFM based on geological choices and extend that model to the entirety of California.

In parallel, we updated the existing GFM 1.0 volumetric mesh to conform to CFM 7.0 where bounding faults were improved from CFM 5.3, by building a GOCAD structural model typically used in reservoir characterization. This process includes selecting fault and non-fault boundaries, merging and extending faults, defining fault-to-fault relationships, and adding "horizons" (topography, seismogenic depth, Moho, LAB). The resulting, well defined structural model is used to resample region bounding surfaces for regularized 2D and 3D meshing.

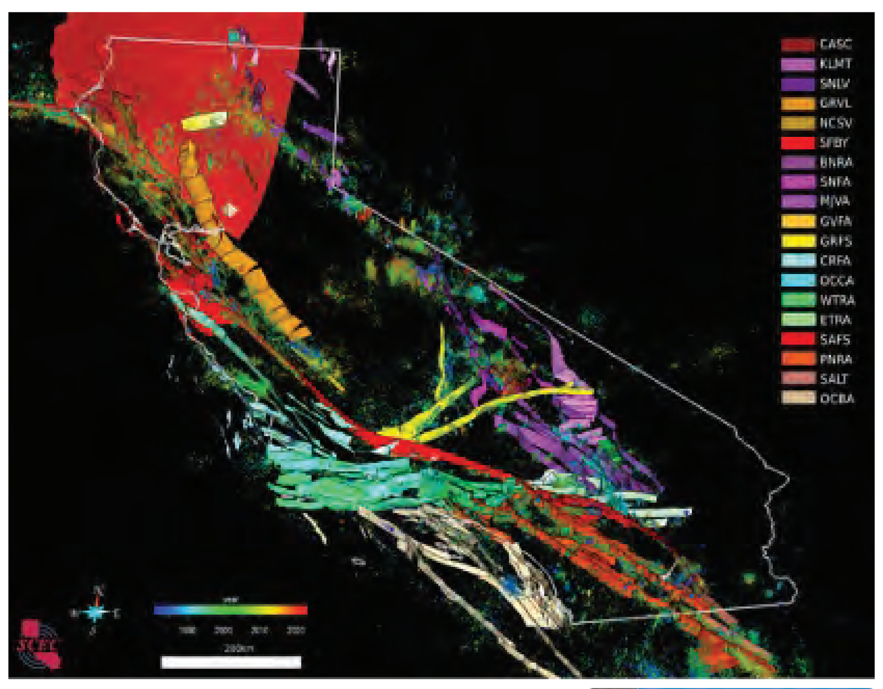
2. Current Representations

The **GFM 1.0** comprises 22 blocks and describes the lithologies expected at various depth in each block



DOI 10.5281/zenodo.4579626

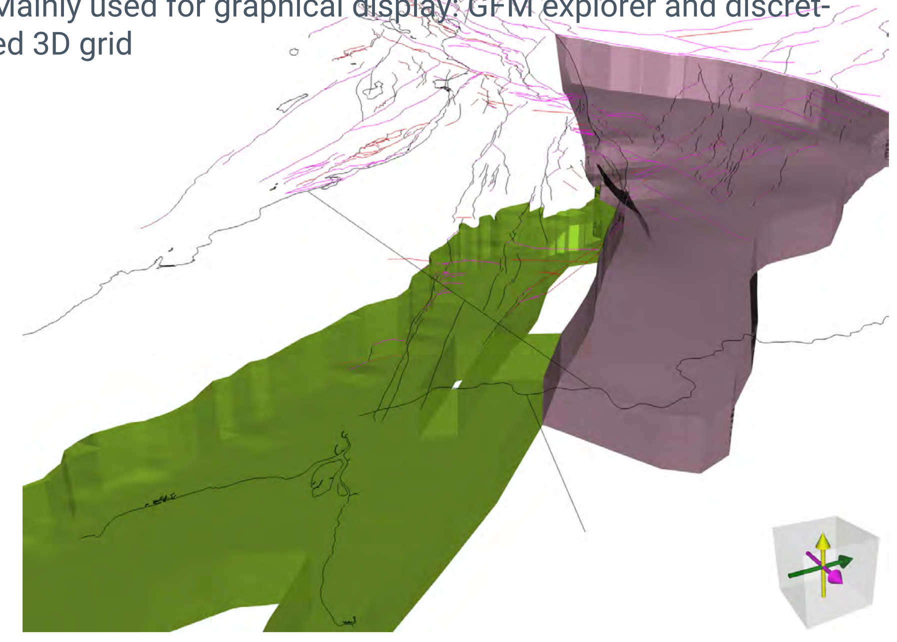
The **CFM 7.0** comprises 556 faults over all of California. Faults surfaces are typically built by hand with variable strike and dip.



DOI 10.5281/zenodo.4651667

Our inspiration: GFM v.1.0 3D

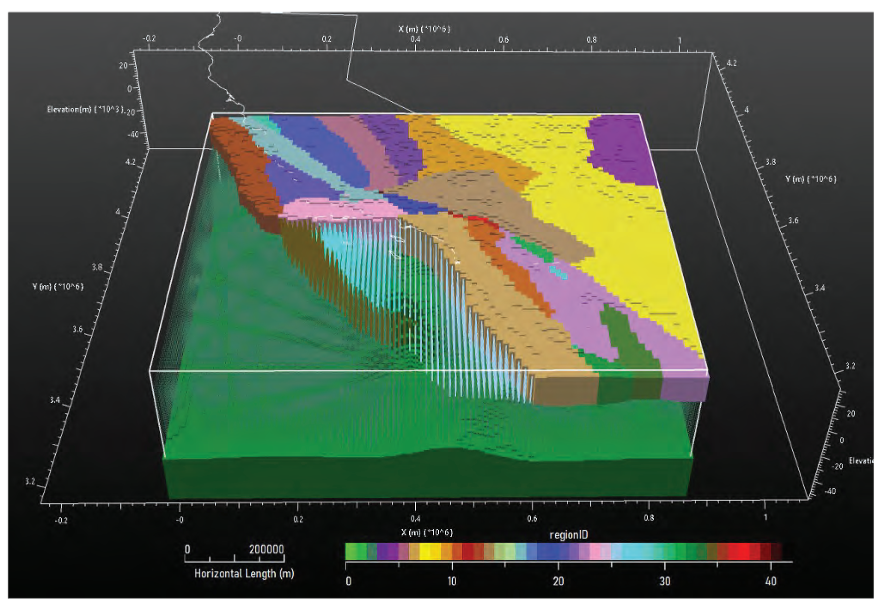
_ Watertight blocks corresponding to GFM lithotectonic units
_ Boundaries conforming to CFM 5.2 faults where appropriate
_ Mainly used for graphical display: GFM explorer and discretized 3D grid



http://moho.scec.org/GFM_web/web/viewer.php

What will be different

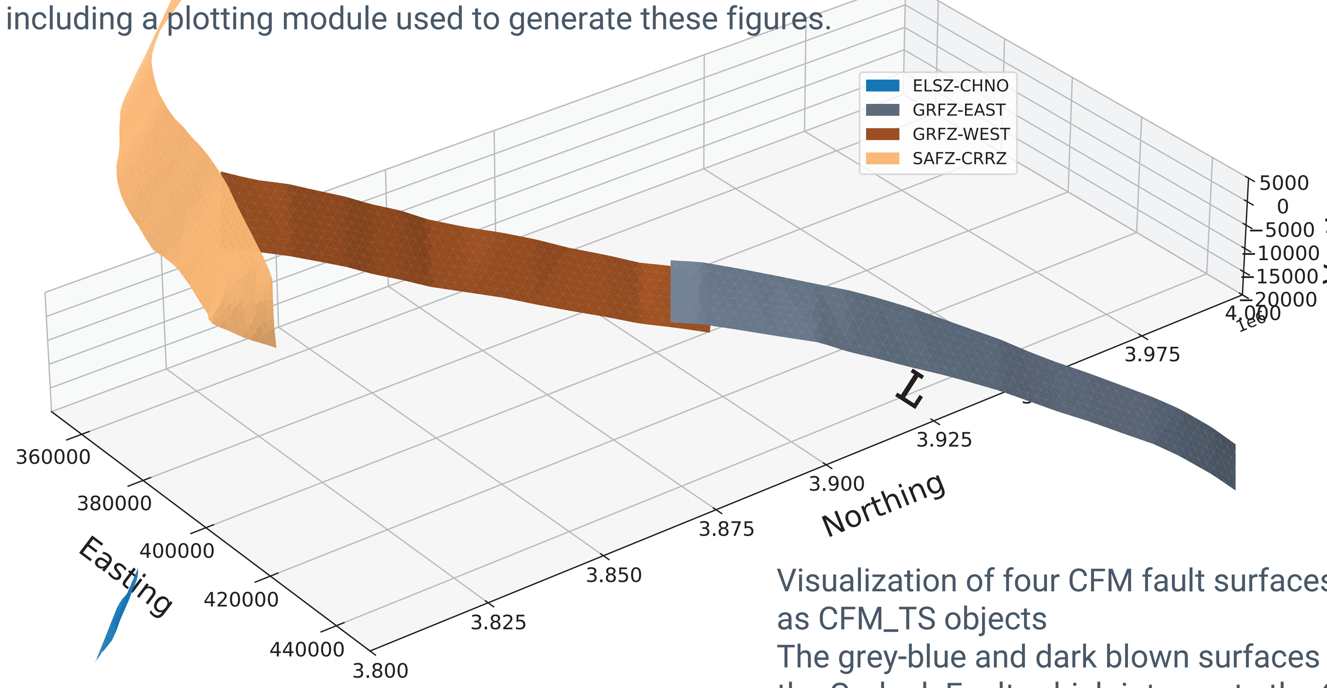
_ New model generated with minimal human input and using open-source tools so it can be reproduced and updated as new information becomes available



3. Surface handling modules

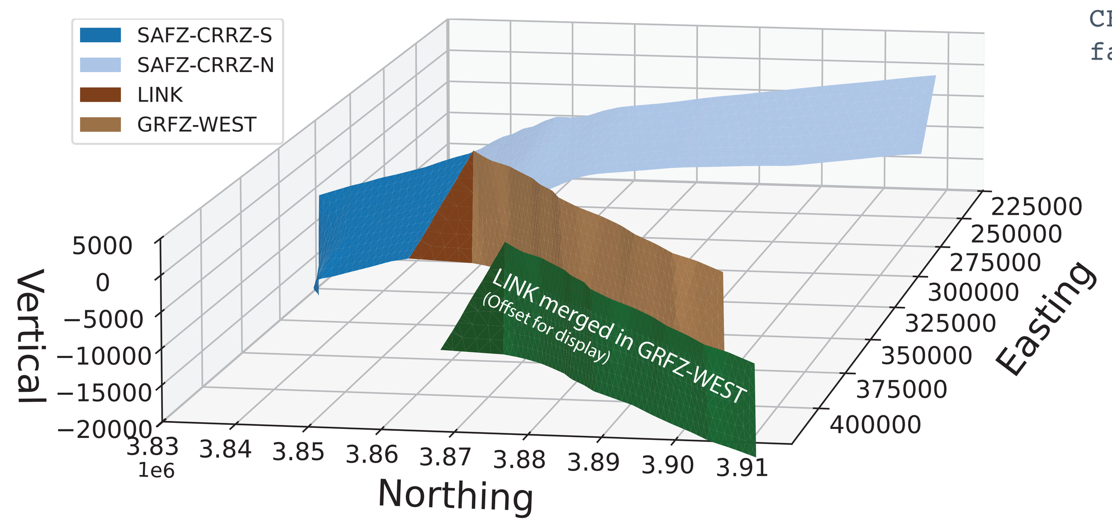
All new surface handling modules are written in Python. Join the project at `https://github.com/montesi/SCEC_GFM_Utilities` **if you want to help!**

The CFM is distributed as a collection of GOCAD T-surf files. The files are parsed and stored in a specific class, CFM_TS, with a function modified from python_geoprobe. The surfaces are described as a list of points in three dimensions and triangles supported by these points. There are no gaps or surface elements other than triangles.
Several operations are needed to generate surfaces that will eventually form a water-tight block boundary. CFM_link joins two surfaces separated by a stepover. CFM_Tjunction cuts a fault into separate segments where it intersects a second fault and generates a new segment where joining the cut line and the intersecting fault. These modules occasionally used PyVista commands.
Additional utilities help implementing these operations. They include CFM_edges, which defines the outline of a CFM_TS surface and CFM_merge, which adds triangles from one surface to another.
The package also including a plotting module used to generate these figures.



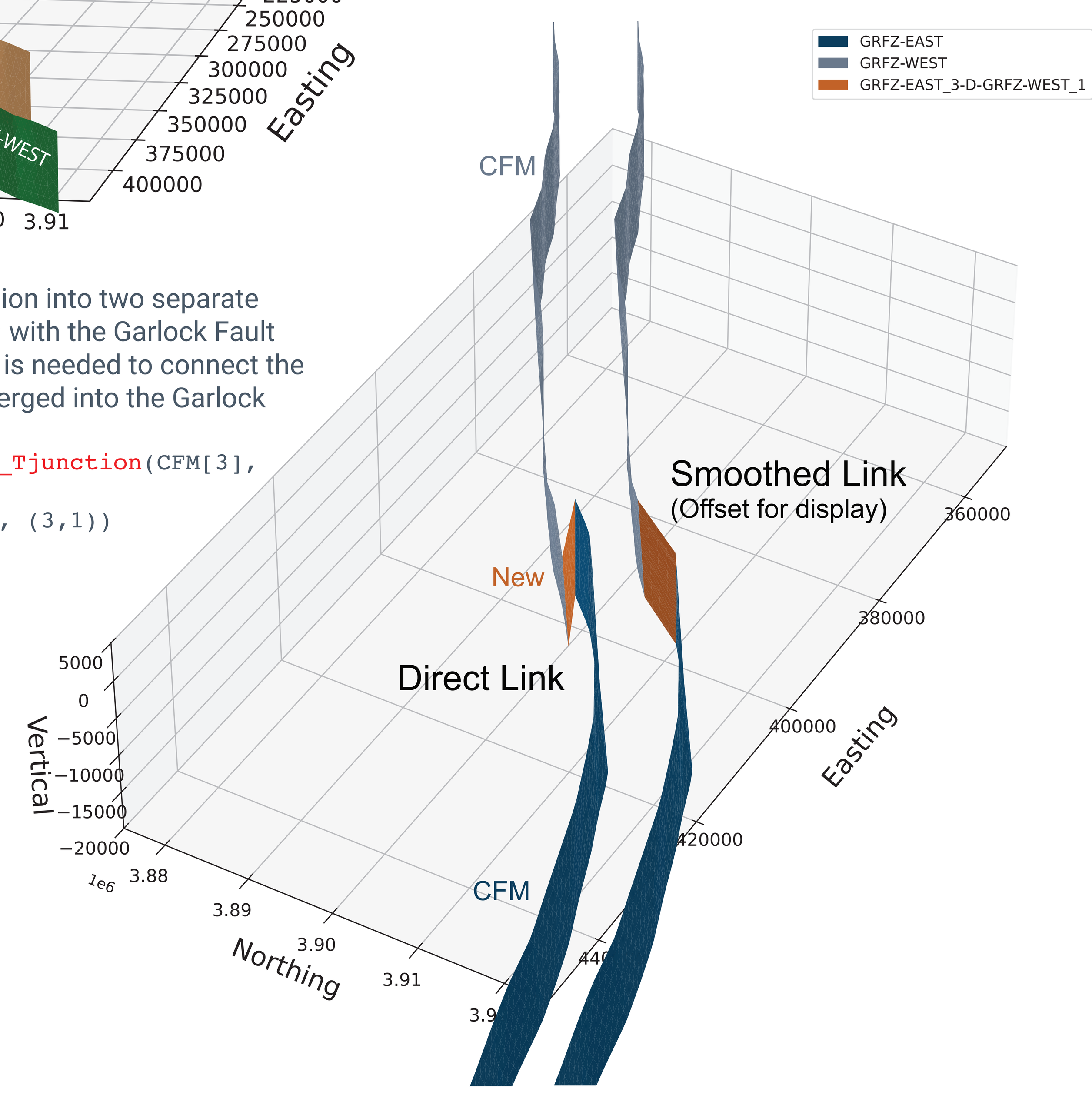
Visualization of four CFM fault surfaces imported into Python as CFM_TS objects
The grey-blue and dark blown surfaces are two segments of the Garlock Fault, which intersects the Carrizo section of the San Andreas Fault (light brown)

```
CFM_surface = read_CFM("GRFS-GRFZ-EAST-Garlock_-fault-CFM5_2000m.ts")
```



The CFM_Tjunction module cut the Carrizo Section into two separate surfaces (light and dark blue) at the intersection with the Garlock Fault (light brown). A new surface (LINK, dark brown) is needed to connect the Garlock and San Andreas faults. LINK can be merged into the Garlock segment to produce the green surface.

```
SAF_south, SAF_north, Link_surface = CFM_Tjunction(CFM[3], CFM[2].segments[3])
MergedGarlock = CFM_merge(CFM[2], CFM[7], (3,1))
```



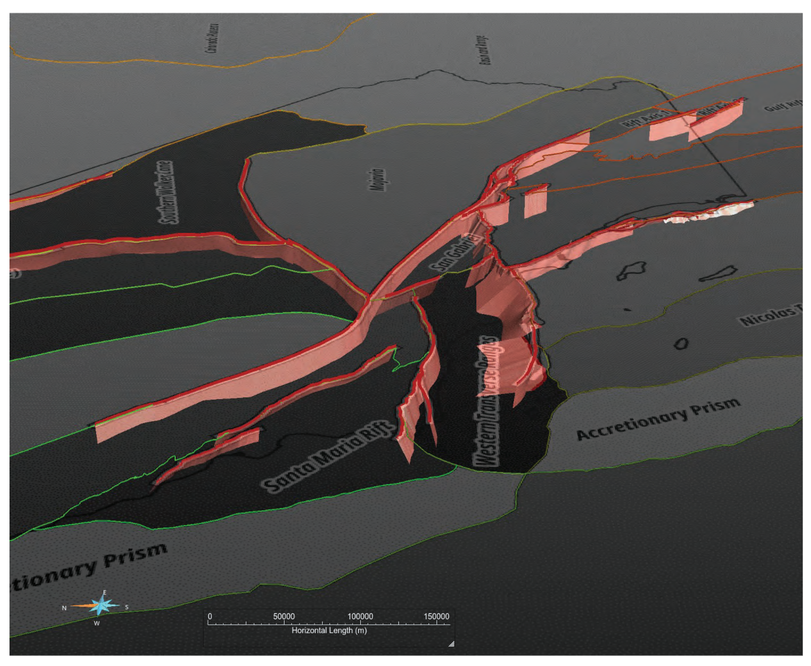
The gap formed by the step-over between the East and West Garlock Fault (light and dark blue) is closed by a new surface (brown) that either links the previously defined segment ends (left, Direct Link) or cuts across the region where the segments overlap (right, Smoothed Link)

```
s0,s1,joint=CFM_link(CFM[1], CFM[2], link_mode='direct')
s0,s1,joint=CFM_link(CFM[1], CFM[2], link_mode='smooth')
```

4. Boundaries in the GFM

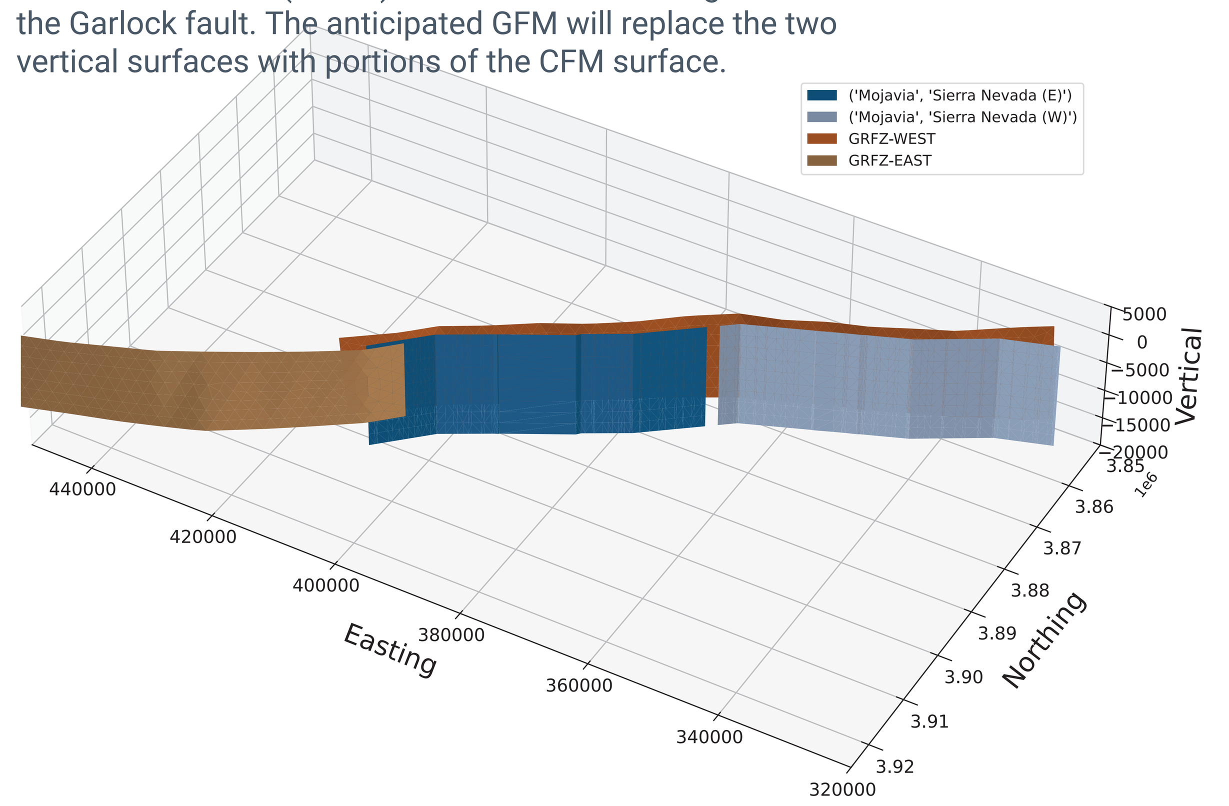
The envisioned GFM will consist of two datasets. The first describes blocks and their stratigraphy, as in the current GFM release. However, it will no longer list explicitly the latitude and longitudes of their outlines. Instead, it will include a list of block boundaries that are provided by the second dataset, avoiding redundancy. In that dataset, the boundaries will be given as a CFM_TS object with boundary-related information. Importantly, these boundaries will have consistent edges, so that the blocks are water tight, and will not be forced the be vertical. In that way, the GFM will be truly three-dimensional.

Whenever possible, the boundaries will be derived from the CFM. However, some boundaries are no recognized as faults. For these, we will preserve the coordinates in the current distribution of the GFM. Our objective is to automatically define the boundaries from the current GFM. Then, a script will replace GFM boundaries with CFM-derived surfaces, where reasonable. Human input will be limited to listing which CFM fault segments are needed, and any CFM operation needed to define these segments (e.g. keep only the portion of the San Andreas Fault Carrizo segment south of the intersection with the Garlock Fault). The sequence of operations will be provided as a script to so that the 3D GFM can be reproduced and updated as new information become available. Users can also generate alternative GFMs if desired.



Selected CFM 7.0 faults (red) superposed on the GFM block outlines (green) showing the mix of CFM and non-CFM surfaces needed to outlining the block boundaries in 3D

Comparison between the vertical GFM boundaries (blue) and CFM surfaces (brown) across the Eastern segment of the Garlock fault. The anticipated GFM will replace the two vertical surfaces with portions of the CFM surface.



Next steps. Do you want to help?

We welcome contributions to the python modules, such as new functionalities, streamlined algorithms, and improved robustness. Contact `montesi@umd.edu` and fork our github repository `https://github.com/montesi/SCEC_GFM_Utilities`
Next challenges include structuring the GFM block and boundary datasets, defining them as classes to add lithological information, extend surfaces with depth (remember they may intersect), and consider topography and relief at the Moho. FUN!
This effort is a first step towards generated an **integrated geological and geophysical model of California**. If thus is something you may be interested in, contact `montesi@umd.edu` to discuss the formation of a TAG or developing a proposal.

Acknowgments

This research was supported by the Statewide California Earthquake Center (Contribution No.14833). SCEC is funded by NSF Cooperative Agreement EAR-2225216 and USGS Cooperative Agreement G24AC00072-00.
SCEC Proposal 25312