

Infrastructure Enhancements to the Community Rheology Model and its Explorer

Laurent Montesi,¹ Andreas Plesch², John Shaw²

¹University of Maryland, Department of Geological, Environmental, and Planetary Sciences

²Harvard University, Department of Earth and Planetary Sciences

Project objectives

Community Earth Models (CEMs) development remains a hallmark of SCEC activities. Curated by the SCEC community, these models provide vetted information useful for various activities related to earthquake hazards. Of particular interest for this project, the Community Rheology Model (CRM, Hearn et al. 2020, 2026) is constructed on top of the Geological Framework (GFM, Oskin et al. 2017; Hearn et al. 2026). It provides consistently defined flow law parameters (L. G. J. Montesi et al. 2021) that can be used to understand strength variations over the model's scope (Izquierdo et al. 2019; L. G. J. Montesi 2023; Perry and Montesi 2024) and helps interpret viscosity estimates obtained by postseismic creep, lake unloading, or other analysis (Burgmann and Dresen 2008; Dickinson-Lovell et al. 2018; Luttrell et al. 2007).

However, the CRM does not currently provide information directly related to fault zones. One reason is the lack of generally agreed upon constitutive relations that can be linked to geological information regarding the faults, or even the absence of information regarding the material in fault zones (Ault et al. 2025). Additionally, the GFM is not yet set up to link information with the surfaces of the model, only the volumes.

This project addresses this last point by creating explicit links between the GFM and the Community Fault Model (CFM, Plesch et al. 2007, 2024) through a library of open-source modular functions. The CFM segments are often separated by gaps typically justified by geological or geophysical data. However, continuum models of the regions require GFM domains to be closed. Therefore, we spent most of our efforts developing modules that automatically close these gaps. In this way, it is possible to construct a complex outline with a series of small commands. Although we do not yet have a full set of functionalities, the modules described below provide the foundation for a complete utility.

Library access

The modules developed are available at a dedicated GitHub repository (L. Montesi [2025] 2026). All the modules are collected into a single python script, `GFM_utilities.py`. The repository

includes a Jupyter notebook `construct_GFM.ipynb` that illustrates the use of the provided modules using several example fault segments from the CFM.

Dependencies were kept to a minimum. The utilities need PyVista (Sullivan and Kaszynski 2019), NumPy (Harris et al. 2020), and matplotlib (Hunter 2007).

The repository also includes a Todo list for the authors and potential collaborators to keep track of work to be completed.

Library contents

Library contents are described by the Readme file reproduced below:

- **read_CFM** Import a tsurf generated by GOCAD. The general workflow has several step:
 - **CFM_TS** Import header, vertices, and triangles, based on `python_geoprobe` [python_geoprobes](#)
 - **CFM_connections** Define the connections between vertices
 - **CFM_edges** Define the surface's outline and break it into four segments
 - Optional: pass two opposite edges if they are already known.
 - **segment_angle** A function calculating the angle between successive segments.
 - **CFM_flip_segment** If needed, flip the segment definitions so that the first vertex of the first segment is at the surface and in the south.
- **CFM_join** Make a new surface joining two outline segments. Two linking options are provided: “direct” simply links the edges of the segments, while “smooth” starts by cutting each segment where they are closest to the other’s edge.
 - **find_closest_edge** As the name indicates...
- **CFM_Tjunction** Cut a surface where it is close to the edge of a different one; Add a surface joining the stem of the T to the cut. Note that the clipping is done using `pyVista`
 - **surf_to_CFM** convert a `pyVista` object to our CFM object
- **CFM_link** smoothes over a stepover
 - `link_mode = 'direct'` (default) simply join the overlapping edge
 - `link_mode = 'smooth'` cuts the surfaces where the stepover begins, and then join the edges
- **CFM_merge** merge two surfaces, removing duplicate vertices
 - Optionally: Give the indices of the duplicate edges as a tuple.
- **CFM_view** Makes a technical visualization with the triangles and the outline segments for a list of CFM surfaces
 - **get_n_color** As the name indicates. Thanks, Sourabh Shubham.
 - Optional arguments
 - `elevation, azimuth`: for viewing
 - `x_lim, y_lim, z_lim`: limit the plot
 - `show_outline`: the outline is useful for planning surface operations but can makes the view too busy

- surf_color: set to False for a wireframe look
- save_id, fig_size

Illustrations

The principal objective of these modules is to load CFM surfaces and manipulate them to build progressively an outline of the GFM blocks. Figures 1 and 2 illustrate two steps necessary to build progressively the northern boundary of the GFM Mojave block using the Carrizo of the San Andreas Fault and the eastern and western segments of the Garlock Fault System

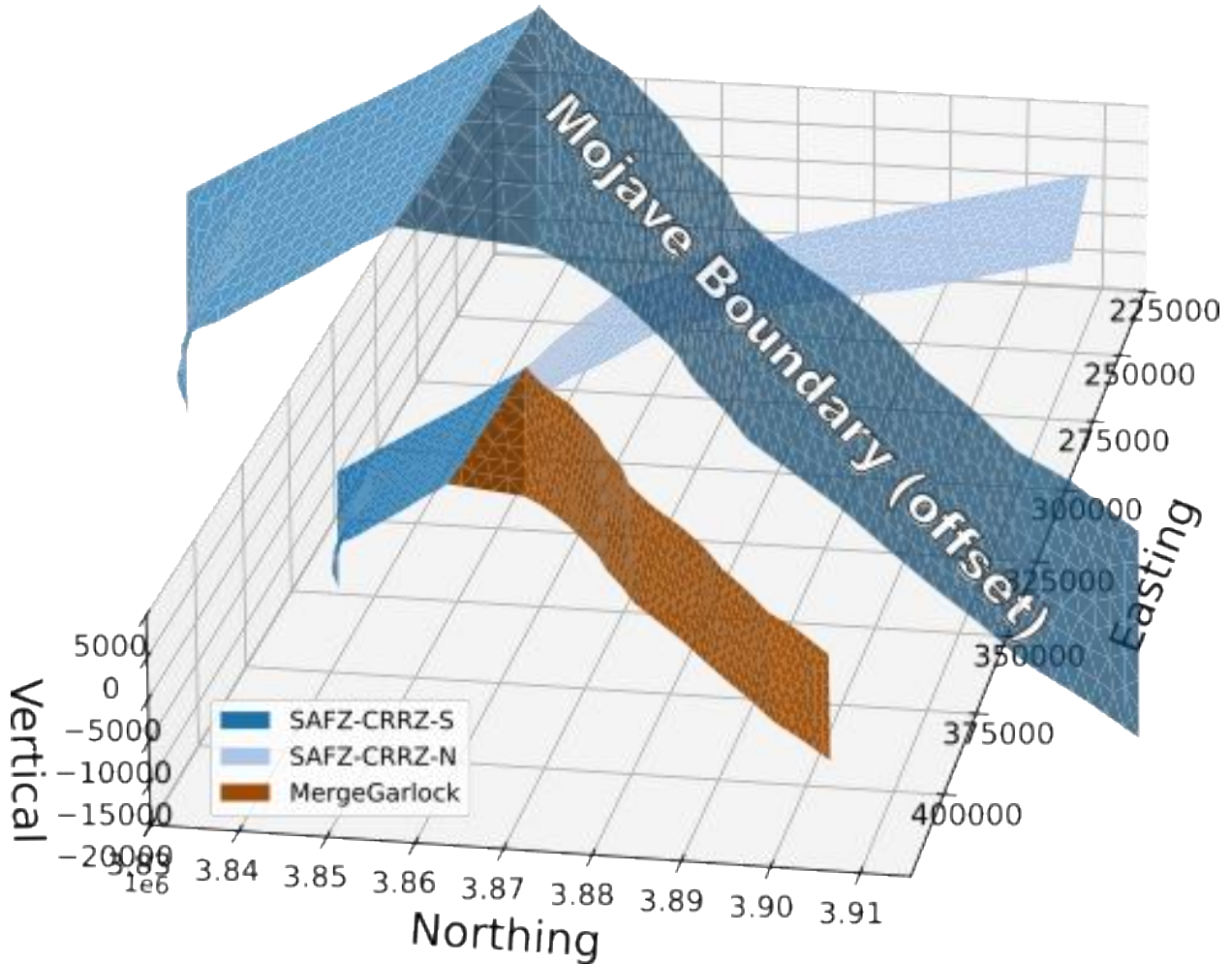


Figure 1: The Garlock Fault meets the San Andreas Fault Carrizo segment (CRRZ) at the Western end of the Mojave blocks. To represent this outline, it was first necessary to cut the CRRZ segment into a northern and southern portion. The cut is defined as projection of the western edge of the Garlock segment unto CRRZ. A new surface had to be added to Garlock to fill the gap between these surfaces, forming a new “MergeGarlock” surface. The Mojave boundary, shown as magnified semitransparent surface offset from its actual location, is the junction between the southern part of CRRZ and the MergeGarlock surface.

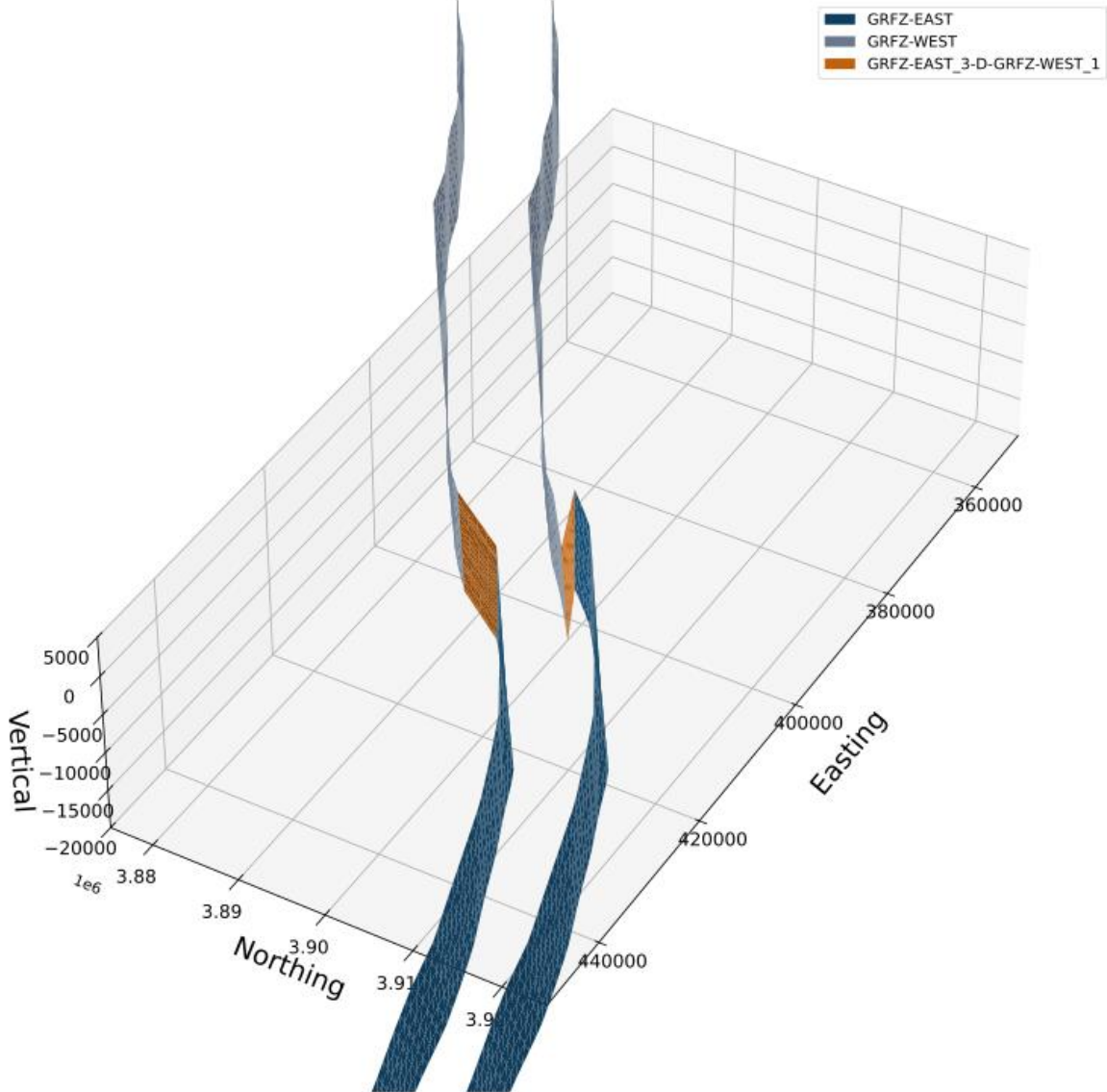


Figure 2: A stepover separates the eastern and western segments of the Garlock Fault. We provide two ways of closing the gap between these segments, either through a direct connection (right, in actual position) or as surface cutting across the stepover (left, offset from the actual position).

Future work

Although the newly developed modules help construct the outline of a fault bounded block, several functions must still be developed before it is possible to generate all the surfaces in the GFM in a manner consistent with the CFM.

- Several boundaries do not correspond to faults. When necessary, the utilities need to read the outline of a GFM block and generate a triangulated surface underneath it. This also requires access to a topographic file consistent with CFM surfaces.
- The surfaces need to be extended to the base of the GFM, that is, the base of the crust. It is also likely valuable to introduce intermediate stratigraphic levels based on the GFM layers.
- A related effort is underway by Crescent (Marfo and Bahavar [2025] 2026). That team chose to use JSON for defining surfaces rather than the Tsurf format inspired by python-geoprobe (Kington [2012] 2025) that we use here. It may be desirable to switch to JSON or provide means of converting one format into the other
- The outline of GFM blocks should be defined as a series of command using CFM segments where possible to automatically build GFM surface and update them as new versions of either the GFM or CFM become available.

References

- Ault, Alexis, Sylvain Barbot, and Caroline Seyler. 2025. “2025 SCEC CRM Workshop.” <https://www.scec.org/events/2025-scec-crm-workshop/>.
- Burgmann, Roland, and Georg Dresen. 2008. “Rheology of the Lower Crust and Upper Mantle: Evidence from Rock Mechanics, Geodesy, and Field Observations.” *Annual Review of Earth and Planetary Sciences* 36: 531–67.
- Dickinson-Lovell, Haylee, Mong-Han Huang, Andrew M. Freed, Eric Fielding, Roland Bürgmann, and Christopher Andronicos. 2018. “Inferred Rheological Structure and Mantle Conditions from Postseismic Deformation Following the 2010 M w 7.2 El Mayor-Cucapah Earthquake.” *Geophysical Journal International* 213 (3): 1720–30.
- Harris, Charles R., K. Jarrod Millman, Stéfan J. van der Walt, et al. 2020. “Array Programming with NumPy.” *Nature* 585 (7825): 357–62. <https://doi.org/10.1038/s41586-020-2649-2>.
- Hearn, Elizabeth, Greg Hirth, Michael Oskin, Laurent Montesi, Mark Legg, and Alex Morelan. 2026. “The SCEC Community Rheology Model for the Southern California Lithosphere: Geological Framework, Ductile Flow Laws, and Preliminary Implications.” *Earth and Space Science* 13 (3): e2025EA004401. <https://doi.org/10.1029/2025EA004401>.
- Hearn, Elizabeth, Laurent Montesi, Mike Oskin, Greg Hirth, Wayne Thatcher, and Whitney Behr. 2020. “SCEC Community Rheology Model (CRM).” Zenodo, September 14. <https://doi.org/10.5281/zenodo.4579627>.
- Hunter, J. D. 2007. “Matplotlib: A 2D Graphics Environment.” *Computing in Science & Engineering* 9 (3): 90–95. <https://doi.org/10.1109/MCSE.2007.55>.
- Izquierdo, Kristel, Laurent G. J. Montesi, William Holt, Alireza Bahadori, and William Shinevar. 2019. “How Certain Are Regional Strength Envelopes in Southern California?” SCEC Contribution 9702. <https://www.scec.org/publication/9702>.

- Kington, Joe. (2012) 2025. *Joferkington/Python-Geoprobe*. Python. September 8, released October 25. <https://github.com/joferkington/python-geoprobe>.
- Luttrell, Karen, David Sandwell, Bridget Smith-Konter, Bruce Bills, and Yehuda Bock. 2007. “Modulation of the Earthquake Cycle at the Southern San Andreas Fault by Lake Loading.” *Journal of Geophysical Research* 112. <https://doi.org/10.1029/2006JB004752>.
- Marfo, William, and Manoch Bahavar. (2025) 2026. *Cascadiaquakes/CRESCENT-CFM-WEB*. HTML. January 22; CRESCENT, released April 10. <https://github.com/cascadiaquakes/CRESCENT-CFM-WEB>.
- Montesi, Laurent. (2025) 2026. *Montesi/SCEC_GFM_Uilities*. Jupyter Notebook. August 18, released May 6. https://github.com/montesi/SCEC_GFM_Uilities.
- Montesi, Laurent G. J. 2023. “Rheological Heterogeneity of the Southern California Lithosphere.” SCEC Contribution 12900. <https://www.scec.org/publication/12900>.
- Montesi, Laurent G. J., Greg Hirth, and Mike Oskin. 2021. “Predicted Ductile Rheology of Textured and Non-Textured Rocks in Southern California.” SCEC Contribution 11193. <https://www.scec.org/publication/11193>.
- Oskin, Mike, Whitney Behr, Alex E. Morelan, Andreas Plesch, and John H. Shaw. 2017. “Toward a Geologic Framework for the Community Rheology Model, with Focus on the Mojave Region.” SCEC Contribution 7520. <https://www.scec.org/publication/7520>.
- Perry, Megan, and Laurent G. Montesi. 2024. “Brittle-Ductile Transition Depth and Depth of Seismicity in Southern California: Agreement in Active Regions but Discrepancy Elsewhere | Statewide California Earthquake Center.” SCEC Contribution 13523. <https://central.scec.org/publication/13523>.
- Plesch, Andreas, Scott Marshall, and John Shaw. 2024. “SCEC Community Fault Model (CFM).” Zenodo, September. <https://doi.org/10.5281/zenodo.13685611>.
- Plesch, Andreas, John H. Shaw, Christine Benson, et al. 2007. “Community Fault Model (CFM) for Southern California.” *Bulletin of the Seismological Society of America* 97 (6): 1793–802. <https://doi.org/10.1785/0120050211>.
- Sullivan, C. Bane, and Alexander A. Kaszynski. 2019. “PyVista: 3D Plotting and Mesh Analysis through a Streamlined Interface for the Visualization Toolkit (VTK).” *Journal of Open Source Software* 4 (37): 1450. <https://doi.org/10.21105/joss.01450>.