Towards a Computational Geodynamics Visualization and Data Analysis Framework

John Naliboff, Sascha Brune, Thilo Wrona, Guillaume Duclaux, Chris Havlin, Dave May

A key challenge in computational geodynamics is the development of robust and efficient software for visualization, information extraction and analysis of large 3D data sets. At present, a wide range of tools are used for visualization (e.g. ParaView, VisIt, GMT) and data analysis (e.g. Python and Matlab), but ultimately, quantitative comparison to observational data sets remains a major problem.

Despite the availability of very efficient geodynamic models, there is no automated way to answer questions like the following ones: How many faults does a model feature at any given time? How many plumes or subduction zones? What is their azimuth or dip, their length, area, or volume? In addition, for large 3D data sets, substantial work is often required to extract and reformat data subsections that can be processed on smaller machines or with software lacking parallel processing capabilities. In many cases, individual research groups likely develop similar workflows and toolkits to address these issues, but no platform exists to easily integrate this work into one or more existing community packages.

Given these issues, a robust and open-source community visualization and data analysis package supporting a wide range of functionality, data formats and types would be of great benefit to the broader geodynamics community. However, the challenges associated with developing such software for a wide range of geodynamic applications are multifold, and include:

(1) The ability to efficiently read massive 3D data sets derived from a wide range of data formats and styles (i.e., structured versus unstructured meshes, model geometry).
(2) A simple, programmable interface to data analysis and plotting routines, which can be easily adapted for a wide range of simulation or observational data sets.
(3) Integration with existing image processing and machine learning algorithms
(4) Building robust documentation and an example applications suite
(5) Designing the underlying code structure in a manner that allows new users to easily add new features, which can in turn be integrated back into the main repository.

Using our research and software development efforts in recent years as a guide, we believe a community visualization and data analysis software package built on existing python functionality can overcome all of these challenges. For example, the python visualization package yt (https://yt-project.org) currently supports parallel reading, rendering and analysis of a wide range of data types through a clean and user-friendly programmable interface. Similarly, existing python libraries for image processing and machine learning have been successfully applied to a wide range of Earth science applications, including, for example, automatic detection and extraction of faults within simulations of lithospheric deformation.

Here, we propose that CIG builds on existing python-based software and development efforts (e.g., Scipy, Matlplotlib, skimage, yt) to produce a new visualization and data analysis package, which (1) supports a wide range of data types, (2) supports parallel loading and rendering of massive 3D data sets, (3) utilizes a modular design and robust documentation to promote community development, and (4) provides an interface to existing python packages for image processing and machine learning, which are critical for comparing geodynamic models to observational data sets. Development of the software packages should include both participants from the visualization community and a wide range of computational geoscientists, which will ensure broad usability of the core functionality and growth of a stable development team. Development efforts should be supported by yearly hackathons following the model of existing CIG codes, while user-support and outreach can be achieved thoroughly through annual training workshops and online forums.