Geological Data Visualization
Week 2

Andrew Pellett
6/12/08

Related Work

There are a number of applications and even scripting languages made specifically to visualize large amounts of data. Some examples include the Geoscience Interactive Databases (GEOID) project at Cornell University, the NCAR Command Language (NCL), and Visu. Links to these projects are, respectively:

http://atlas.geo.cornell.edu/geoid/geoid.html
http://www.ncl.ucar.edu/overview.shtml
http://www.agu.org/eos_elec/97237e.html

GEOID interfaces with some commercial GIS software, ArcInfo, and provides a relatively simplistic GUI. GEOID retains an internal database of data, and it's unclear whether it is possible to process new data sets.

NCL is a powerful language maintained by the Computational and Information Systems Laboratory at the National Center for Atmospheric Research (NCAR). NCL can handle input from a wide variety of file types, including netCDF, and is capable of producing graphics at high resolutions and in many different formats. NCL includes commands for many different kinds of data manipulation and analysis, including interpolation, empirical orthogonal functions, and 1,2, and 3 dimensional visualizations.

Visu is another powerful tool for geological data visualization. Visu provides a graphical user interface with an extensive set of parameters required for visualization generation. Visu, like NCL, can handle netCDF, and is capable of outputting images in a variety of standard image formats such as JPG, GIF, and TGA. Visu doesn't include any sort of data analysis or manipulation tools.

Progress

A C module for parsing netCDF files has been completed. Given a struct describing what is desired from the netCDF file, the function returns a data structure containing the desired data as well as pertinent metadata. The next step is to visualize the data. Currently, libgd is being used to explore drawing each region of like data as a color mapped polygon. Once this is done, a separate module could be written to create an animation from the set of still frames. The development of C methods for this project has been suspended in the interest of completing the MATLAB version first.

The existing MATLAB visualization code, written by Sean Birkel, uses built-in MATLAB functions to create the color maps, and is more useful than the C code, currently. A graphical user interface (GUI) is being developed to allow the user to specify some of the input and output parameters of the existing code. The GUI will allow the code to be used by people who aren't experienced in programming.