GMG: An open source two-dimensional geophysical modelling GUI

22 May 2018

Summary

For decades, forward modelling of potential field data, such as gravity and magnetic anomalies, has been common practice within the geophysics community as a means of constraining subsurface structure. Many software packages (both freely available and commercially licenced) exist for preforming such modelling. However, most, if not all of these packages, suffer from at least one major draw-back, such as: (1) being closed source; (2) not allowing for the calculation of both gravity and magnetic anomalies; (3) Not facilitating the integration of complimentary data within the modelling environment and (4) being programmed in such a way that the software is cumbersome for integrating within an academic research project due to, for example, (5) being a single platform release (usually Windows only), (6) having poor I/O functionality and (7) poor documentation.

GMG is an open-source Python package primarily intended as an interactive, user-friendly two-dimensional geophysical forward modelling GUI that resolves all of the draw-backs listed above. Both gravity and magnetic anomalies can be computed along a 2D profile consisting of subsurface bodies defined as any number of 2D polygons. Moreover, functions for displaying complementary data within the modelling environment, such as exploration well logs and seismic data, are provided. `GMG` has been designed with a minimalist user-interface and simple I/O in order to enhance usability and it is expected to be useful for both researchers and teaching of exploration geophysics. Most importantly `GMG` is fully open source. Hence, providing an environment where users can add new functionality and optimise processes such that the software will naturally become more useful and streamlined over time.

`GMG` makes extensive use of functions from the Scientic Computing in Python (SciPy) package (Oliphant 2007). In particular, NumPy (Van Der Walt, Colbert, and Varoquaux 2011) data structures are used for data handling and computational efficiency. Matplotlib (Hunter 2007) plotting tools are employed for displaying and interacting with graphics. The GUI is implemented using the

wxWidgets GUI toolkit, wxPython (Rappin and Dunn 2006). Further, software dependencies include Fatiando a Terra (Uieda, Oliveira Jr, and Barbosa 2013), from which, the function fatiando.polygon is used for handling model layers and ObsPy (Beyreuther et al. 2010), from which, the seismic plotting function obspy.read is used for loading seismic data. The algorithms for calculating the gravity and magnetic anomalies are from (Bott 1969) and (Talwani and Heirtzler 1964) respectively. The source code for `GMG` is stored on github at: https://github.com/btozer/gmg

Acknowledgements

GMG was conceived at the University of Oxford and with contributions from Brook Keats.

References

Beyreuther, M., R. Barsch, L. Krischer, T. Megies, Y. Behr, and J. Wassermann. 2010. "ObsPy: A Python Toolbox for Seismology." *Seismological Research Letters* 81 (3): 530–33. doi:10.1785/gssrl.81.3.530.

Bott, M H P. 1969. "GRAVN." *Durham Geophysical Computer Specification* 1 (7).

Hunter, John D. 2007. "Matplotlib: A 2D graphics environment." *Computing in Science and Engineering* 9 (3): 99–104. doi:10.1109/MCSE.2007.55.

Oliphant, T. E. 2007. "Python for Scientific Computing Python Overview." *Computing in Science and Engineering Engineering*, 10–20.

Rappin, N, and R Dunn. 2006. *wxPython in Action*. Manning Publications. doi:10.1017/CBO9781107415324.004.

Talwani, M, and J. R. Heirtzler. 1964. "Computation of magnetic anomalies caused by two dimensional bodies of arbitrary shape." In *Computers in the Mineral Industries, Part 1*, 9:464–80.

Uieda, Leonardo, Vanderlei C. Oliveira Jr, and Valéria C.F. Barbosa. 2013. "Modeling the Earth with Fatiando a Terra." *Proceedings of the 12th Python in Science Conference*, no. Scipy: 90–96.

Van Der Walt, Stéfan Stéfan, S. Chris Colbert, and Gaël Gaël Gaël Varoquaux. 2011. "The NumPy array: A structure for efficient numerical computation." *Computing in Science and Engineering* 13 (2): 22–30. doi:10.1109/MCSE.2011.37.