

Progress Report 2018–3–16

Since our last report in September, 2017 we have continued work on several parallel tracks:

1. We have maintained the obsolete GMT 4.5.x series, including a Jan. 1, 2018 v 4.5.17 release.
2. We have maintained the GMT 5.4 branch which is the current stable release branch, releasing 5.4.3 (the latest official release) on Jan. 1, 2018.
3. We are stabilizing GMT 6.0.0 β and experimenting with the prototype modern mode.
4. We are approaching the half-way point of the development of the GMT/Python package.
5. We have initiated discussions and planning regarding a potential move of GMT to CIG [<https://geodynamics.org>] for long-term hosting and maintenance.

Maintenance of 4.5.x

We will issue a final GMT 4-series release this June 1. This will be version 4.5.18 and its release will terminate further maintenance of the old GMT4 series, almost 5 years after GMT 5 was introduced. This final reference version can be used by those who have workflows that require GMT4. To assist users with any transition issues, we will add a new web page on our site that will summarize the various options users have to port any old scripts if they so wish. We will also clarify how users can install and switch between any number of GMT installations, which is helpful during a transition period. The new site will discuss various functions and scripts that we supply with GMT 5 to assist with such issues. GMT 5 was designed to be fully backwards compatible with GMT 4 and we will continue to respond to any issues filed that shows otherwise.

Maintenance of 5.4

The 5.4 branch is only used for bug fixes; no new features are added to 5.4.x, hence it is deemed our stable release. The fink and macports distributions are distributing the latest 5.4.3 release while most Linux distros remain hopelessly stuck farther back (e.g., 5.2.1). We released a small update to our country polygon data base DCW in January 2018 (1.1.3). This release fixes issues related to horizontal streaking lines for some countries internal boundaries. We anticipate releasing additional 5.4.x releases well into 2019 as GMT 6 goes through its development. Between 50 and 100 issues have been closed since our last report in September, 2017. We are about to release 5.4.4 which will simplify installation and building of the MB-system with GMT5 and 6.

GMT 6

All new development is directed at GMT 6. Note that this main activity is not just involving new features or hardening of existing code. It also includes complete rewrites to strengthen a particular feature or to simplify further maintenance. Such whole-sale rewrites are not committed on the GMT 5.4.x branch since it breaks with the coding conventions for that branch (and could also make it less stable). We have recently strengthened GMT with respect to 3-D views from any azimuth (the previous algorithm failed when viewed from some quadrants), we have improved the rasterization of PostScripts to rasters by using sub-pixeling to better render text (we had extensive discussions with the GhostScript developers to achieve this result), and we continue to stabilize and document GMT 6. We will be introducing a new movie module that will bring animations to the masses. This new module makes it simple to set up and generate GMT animations and building animated GIFs or MP4 video directly. We have submitted an EarthScope proposal of which a small

part is to support the development of new modules, such as movie and subplot.

On the testing front there has been a large increase in test scripts. As discussed, most bug reports are converted to failing test scripts and then added to the test suite. The scripts will not pass the test until solutions are found. Many new test scripts have been added; the total now is 776 scripts, of which ~18 are currently failing. Finally, we have made a few changes to make GMT 5 and 6 more compatible with GMT 4 by reintroducing the old-style arrow symbol in GMT 4 into GMT 6.

The other key direction for us is to simplify GMT. With users saying "Make GMT simpler, but don't change anything!" we are supplying simplifications under the modern mode framework. Streamlining options, improving parsing, clearer (and more) documentation are all part of this effort, but implementing and testing our prototype modern mode is where most of our development time (as different from maintenance time) goes.

Python package

The latest documentation for GMT/Python is automatically built and hosted on the website <http://www.gmtpython.xyz>. To ease new users into the Python interface, we have created an interactive online demo hosted at <http://try.gmtpython.xyz>. The demo uses the <https://mybinder.org> service to host temporary Jupyter notebooks with all required dependencies already installed. Users can experiment with the new library without having to download or install anything. We envision using this service as a low-barrier way to promote the GMT6 modern mode.

The library development has focused on improving and systematically testing the low-level wrapper module around the GMT C API (module `gmt.clib`). This module is used to call functions from the C API and communicate data between Python data types (numpy `ndarray` and pandas `Series`) and GMT data containers. Work on `gmt.clib` is approximately 70% done.

The high-level plotting API for GMT/Python has been completely redesigned. All plotting is now handled by the `gmt.Figure` class which includes two Python-specific functions:

1. `Figure.savefig`: A more convenient way of producing figure files than `psconvert`. It behaves like the `savefig` function in Python's `matplotlib` library.
2. `Figure.show`: Function to view a GMT figure in the Jupyter notebook.

The Jupyter notebook enables interactive visualizations through HTML and JavaScript. Support for viewing a `gmt.Figure` in the notebook as an interactive globe (similar to Google Earth) was recently implemented using the [NASA Web WorldWind](#) Javascript library. The globe visualization is automatically generated and inserted in the notebook by using `Figure.show(method='globe')`. An example can be seen on the [tutorial website](#).

A lot of effort has gone into ensuring code quality and attracting new volunteer contributors to the project. Continuous integration services automatically check code quality and test coverage of every change made to the code. The project now includes a [Contributing Guide](#) and a [Code of Conduct](#) to set clear expectations for behavior in our forums, possible repercussions, and a means of reporting violations. To help new contributors get started, we maintain a curated list of beginner friendly tasks through Github issues with the tags `'good first issue'` and `'help wanted'`. Through these efforts, we have already received [several contributions](#) from [Dongdong Tian](#), a PhD student at the University of Science and Technology of China. We are hoping to expand the number of contributors at the 2018 Scipy Conference.

Outreach

Both Paul and Leo attended the Fall AGU Meeting in New Orleans in December, 2017. We gave presentations on the prototype GMT 6 modern mode and a status report on the Python API. Leo's poster is available at doi:[10.6084/m9.figshare.5662411](https://doi.org/10.6084/m9.figshare.5662411) and the [source files are on Github](#), including a demo Jupyter notebook for GMT/Python and GMT6 command line. The demo can be executed online (no install required) at <http://agu2017demo.gmtpython.xyz>. Wessel's presentation can be found below.

Transition planning

We are taking the initial steps of developing a complete plan for how GMT repositories and their maintenance and further development might be migrated from SOEST to CIG (Computational Infrastructure for Geodynamics). These developments are expected to take up considerable time during the next 1–2 years and will necessarily involve the entire GMT team.

 [Wessel++_GMT6_AGUF2017.pdf](#) – Fall AGU 2017 talk on GMT 6 by Paul Wessel (1.99 MB)  Paul, 2018–03–09

21:25