Astronomical Data Analysis Software and Systems XIX **O14.3**
ASP Conference Series, Vol. XXX, 2009
Y. Mizumoto, K.-I. Morita, and M. Ohishi, eds.

Status of GDL - GNU Data Language

A. Coulais

LERMA, Obs. de Paris, ENS, UPMC, UCP, CNRS, Paris, France

M. Schellens[1]

J. Gales

Goddard Space Flight Center, Greenbelt, MD, USA

S. Arabas

Institute of Geophysics, University of Warsaw, Poland

M. Boquien

University of Massachusetts, Dep. of Astronomy, Amherst, MA, USA

P. Chanial

P. Messmer, D. Fillmore

Tech-X GmbH, Zurich, Switzerland; Tech-X Corp, Boulder, CO, USA

O. Poplawski

Colorado Div. (CoRA) of NorthWest Res. Ass. Inc., Boulder, CO, USA

S. Maret

LAOG, Obs. de Grenoble, UJF, CNRS, Grenoble, France

G. Marchal[2], N. Galmiche[2], T. Mermet[2]

Abstract. Gnu Data Language (GDL) is an open-source interpreted language aimed at numerical data analysis and visualisation. It is a free implementation of the Interactive Data Language (IDL) widely used in Astronomy. GDL has a full syntax compatibility with IDL, and includes a large set of library routines targeting advanced matrix manipulation, plotting, time-series and image analysis, mapping, and data input/output including numerous scientific data formats. We will present the current status of the project, the key accomplishments, and the weaknesses - areas where contributions are welcome!

[1]Head of the project

[2]Former students at LERMA CNRS and Observatoire de Paris

1. Dependencies

GDL is written in C++ and can be compiled on systems with GCC ($\geq$ 3.4) and X11 or equivalents. The code, under GNU GPL, is hosted by SourceForge. The library routines make use of numerous open-source libraries including: readline, the GNU Scientific Library (GSL), the PLplot plotting library, a Fourier transform package (FFTw), and others. Since recently (GDL 0.9rc4 release) GDL features multi-threaded matrix operations if compiled using an OpenMP-enabled compiler (e.g. GCC $\geq$ 4.2.)

PLplot and GSL are the only mandatory dependencies.

Data input/output is managed using ImageMagick, NetCDF, HDF and HDF5 libraries. FITS files can be read and written using the Astron Library. GDL features a Python bridge (Python code can be called from GDL, GDL can be compiled as a Python module).

2. Platforms, packages and compilation

GDL runs on most recent Linux and *BSD systems, and also on Mac OSX and OpenSolaris. x86 and x86_64 are the key supported architectures, successful compilations on other architectures have been reported.

Pre-compiled packaged versions of GDL are available for several operating systems including Mac OSX (e.g. via Macports), Debian and Ubuntu, Fedora and Red Hat, Gentoo, ArchLinux and FreeBSD.

The source code compiles smoothly on most Linux distributions, *BSD, recent OSX[3] versions and some other UNIX systems (e.g. OpenSolaris), as long as some caveats are avoided.

Since the pre-compiled packages happen to be out of date and don't include the latest bug-corrections and improvements, we strongly advice to try compiling GDL from source. The source code can be obtained from SourceForge where GDL development is hosted. The most recent additions to GDL are readily available at the CVS repository [4].

3. Useful libraries

Large parts of the Astron Library are working well in GDL, including the FITS part. The MPFIT library (Robust non-linear least squares curve fitting based on MINPACK-1) does work with GDL. IDL save files can be read and written using the external CMSVLIB library. Success have been reported with the Wavelet Library.

Due to limited number of graphical keywords currently available, and also due to the limited achievements of the Postscript output, some famous pub-

[3] On Mac OSX, which is an OS GDL does have significant number of requests from users, success have been reported on 10.4, 10.5 and 10.6. A tutorial for compilation on OSX which try to include the last tricks can be found here: `http://aramis.obspm.fr/~coulais/IDL_et_GDL/GDLonOSX_10.5.6.html`

[4] see `http://sf.net/projects/gnudatalanguage/develop` for details

lic libraries for complex graphical outputs (X11 and PS) are clearly limited in functionality.

4. Contributions

Contributions are very welcome. A large number of bug-reports and improvements are coming from anonymous users, posting directly to the contributors or posting on the SourceForge forum. We really appreciate code donation under GNU GPL. There is a strong demand for more packaging effort. Development is clearly driven by end users (you), except for very difficult tasks (MEDIAN code development and testing was paid, based on T. Huang et al. (1979)). Please do not hesitate to report bugs, regressions, compilation problems, feature requests (e.g. missing keywords), or other comments – you can do it also anonymously at GDL SourceForge page (bug tracker or forum). Please do help in testing GDL by using the CVS version. Finally, please do not be impatient: GDL is developed by a team of volunteers – often busy with other tasks.

5. Weakness

GDL suffers from several weakness, the main one maybe is that none of the main contributors are full time on the project.

- **Documentation** Because of (1) the huge amount of on-line documentation concerning IDL, (2) the goal to be as close as possible to IDL features, (3) we use many external features (e.g. mathematical functions in GSL) we lack a clear pure GDL documentation. Nevertheless, there are some useful documents available on the web, covering different aspects of the project, from installations problems to usage in dedicated field.
- **Widgets** Widgets are not available in GDL now. An implementation based on wxWidgets library is being developed but it is still in very early stage of development.
- **Graphical outputs** The main weakness now is the Postscript output. The current plotting features do not support publication-quality figures. Contributions on this part will be very welcome. Performance of simple plots (PLOT, OPLOT, PLOTS) in X11 using PLplot are comparable to IDL ones. SURFACE and TV are not as fast, especially trough network. For CONTOUR and SURFACE, PLplot does not provide exact equivalent to IDL outputs.
- **Obsolete packages** Another clear problem comes from the gap between the CVS version and the versions available in main Linux distributions. The closest version now is the Fedora one, which usually include important corrections in the CVS.

6. Usage

As for many free software packages, we don't know if GDL is widely used or not, by who and in which field (Astronomy, Geophysics, Medicine ...) but we have some indicators that this project does have users ! We received messages

worldwide, mostly from people working in Astronomy. The greater the number of available features and the simpler the compilation procedure, the less the number of messages.

The known examples of usage include: a master course at the Paris Observatory (∼20 students per year), a master course at the University of Warsaw, refereed papers where computations were done with GDL (e.g. : Koleva & al 2009), computations for MSc and PhD theses.

There are some documented examples of experimental extensions to GDL written for specific research purposes. Jaffey et al. (2008) described a web-based interactive data analysis tool based on GDL, while Jaffey et al. (2009) presented application of a GPU-accelerated version of GDL in solar-physics calculations.

7. Conclusion

The core components of GDL (i.e. interpreter, library routines API, key data manipulation and plotting functionality) are stable and do not pose efficiency problems (no significant discrepancy from IDL performance). Large number of routines are available, several widely used external libraries (Astron, MPfit) can be used in GDL.

We hope to consolidate the users community, to gather feedback in form of bug reports, feature requests, test routines, documentation and patches (several GDL modules have been provided by scientists who wrote the functions for their own work).

The current status is stable and complete enough for numerous applications but still much work is needed. The major axis on development for the next year are:

- aggregating a more efficient community
- giving pre-compiled versions in major Linux distributions closer to the CVS version
- finishing a whole test suite (like the GSL one) to avoid regression and bugs
- having a wider set of graphical keywords
- better Postscript output (or other format(s) for graphical outputs) for publications
- development of documentation

Acknowledgments. We are grateful to many anonymous contributors who spend time and energy on testing and improving GDL. We appreciate all the feedback we got so far from GDL users. Alain C. would like to thanks his former students and also LERMA for financial support.

References

Huang, T., Yang, G. & Tang, G. 1979, IEEE TASSP, 27, 1, 1318

Jaffey, A., Cheung, M.C.M. & Kobashi, A. 2008, American Geophysical Union, Fall Meeting

Jaffey, A., Kobashi, A. & Cheung, M.C.M. 2009, NVIDIA Research Summit `http://www.nvidia.com/content/GTC/posters/89_Cheung_GPU_Acceleration_of_the_Scientific_Data.pdf`

Koleva M., Prugniel Ph., Bouchard A., & Wu, Y. 2009, A&A 501, 3