




# GNU Data Language 1.0: a free/libre and open-source drop-in replacement for IDL/PV-WAVE

Jeongbin Park<sup>1</sup>, Gilles Duvert<sup>2</sup>, Alain Coulais<sup>3,4</sup>, Gregory V. Jung<sup>5</sup>, Sylwester Arabas<sup>6,7</sup>, Brian Barker<sup>8</sup>, Takeshi Enomoto<sup>9</sup>, Sylvain Flinois<sup>10</sup>, Oliver Gressel<sup>11</sup>, Tomas Hillberg<sup>12</sup>, Thibault Huillet<sup>13</sup>, Jan Kohnert<sup>14</sup>, Orion Poplawski<sup>15</sup>, Eloi Rozier de Linage<sup>3</sup>, Remi A. Solàs<sup>16</sup>, Luke Stagner<sup>17</sup>, Ole Streicher<sup>11</sup>, James Tappin<sup>18</sup>, Thierry Thomas<sup>19</sup>, Jingwei Wang<sup>20</sup>, and Christian Wimmer<sup>14</sup>

**1** School of Biomedical Convergence Engineering, Pusan National University, Republic of Korea **2** Univ. Grenoble Alpes, CNRS, IPAG, 38000 Grenoble, France **3** LERMA, Observatoire de Paris, CNRS, Paris, France **4** AIM, CEA, CNRS, Université Paris-Saclay, Université Paris Diderot, Sorbonne Paris Cité, Gif-sur-Yvette, France **5** Independent Researcher, USA **6** Jagiellonian University, Kraków, Poland **7** University of Illinois at Urbana-Champaign, USA **8** Ursinus College, PA, USA **9** Kyoto University, Japan **10** Kumullus, Paris, France **11** Leibniz-Institut für Astrophysik Potsdam (AIP), Potsdam, Germany **12** Department of Astronomy, Stockholm University, Sweden **13** Independent Researcher, France **14** Independent Researcher, Germany **15** NorthWest Research Associates, Boulder, CO, USA **16** Independent Researcher, Norway **17** General Atomics, San Diego, CA, USA **18** RAL Space, STFC Rutherford Appleton Laboratory, UK **19** EDF, Lyon, France **20** MINES ParisTech, Paris, France

DOI: [10.21105/joss.04633](https://doi.org/10.21105/joss.04633)

## Software

- [Review](#) 
- [Repository](#) 
- [Archive](#) 

Editor: [George K. Thiruvathukal](#)



## Reviewers:

- [@mgalloy](#)
- [@mohawk2](#)

Submitted: 21 June 2022

Published: 20 December 2022

## License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#)).

## Summary

We present GNU Data Language (GDL), an open-source free incremental compiler for programs written in Interactive Data Language (IDL<sup>\*</sup>) and Precision Visuals - Workstation Analysis and Visualization Environment (PV-WAVE<sup>\*\*</sup>), two computer languages used for scientific data analysis. GDL is highly compatible with the IDL and PV-WAVE and aims to run any existing IDL codes without any modifications. GDL comes with its dedicated IDE GDL Workbench and Jupyter kernel `gdL_kernel` to provide a comfortable development environment. In addition, GDL supports interoperability with Python. GDL is freely available at <https://gnudatalanguage.github.io/>.

<sup>\*</sup> IDL is a registered trademark of [L3HARRIS](#).

<sup>\*\*</sup> PV-WAVE is a product of [Perforce](#).

## Statement of Need

Interactive Data Language (IDL) is a commercial, domain-specific language used for processing data in various scientific fields, including astronomy, geosciences, biology, hyperspectral, medical imaging (Burrage et al. (2021), Coates et al. (2019), Xiao et al. (2022), Lee et al. (2022), Castro-Tirado et al. (2021)), and even COVID-19 research (Chen et al. (2021)). Although IDL is not so widely used these days for scientific computing as the rise of modern computing languages, many legacy IDL codes are still being used these days. Undoubtedly, porting such legacy codes to modern languages would require a huge effort, it is much more economic to run them with IDL. To run such IDL codes without any financial constraints, there have been several efforts to create a freeware clone of IDL, including Fawlt Language (<https://www.flxpert.hu/fl/>). Fawlt Language is a very successful implementation as it supports nearly 100% of the latest IDL 8.0 syntax. However, Faulty Language is neither

open-source nor free as in free speech and thus users cannot report or fix problems in its source code. Neither IDL nor FL fulfill modern requirements for freely reproducible research.

The GDL project is an international effort to create a free software clone of Interactive Data Language (IDL) or Precision Visuals - Workstation Analysis and Visualization Environment (PV-WAVE), preserving the capability to run the vast body of scientific legacy codes without any technical, legal or financial constraints. GDL has been developed for over four decades with public funding throughout academic institutions around the world. The initial development was done by **Marc Schellens** almost 20 years ago (commit history preserved on GitHub dates back to 2004) and since then a continuous development has been carried out by an evolving team of volunteer contributors – both freelance and affiliated with academic institutions (A. Coulais et al. (2010), A. Coulais et al. (2012), A. Coulais et al. (2014), Alain Coulais et al. (2019), Duvert et al. (2020)). As a result, we have recently announced GDL 1.0.

In terms of compatibility, GDL aims at a full compatibility with modern IDL language specification, including partial support of the IDL 8 specification and above. In addition, GDL is partially compatible with PV-WAVE, another data analysis framework forked from IDL and sharing parts of IDL syntax and library interface. GDL aims to closely mimic the behaviour of the IDL compiler and libraries distributed by Harris Geospatial Solutions, so that the existing IDL scripts to be compatible with GDL without any modifications. The development has been done by relying on IDL and PV-WAVE documentation which has been publicly available on the IDL and PV-WAVE proprietors' websites, IDL documentation which had been publicly available on NASA websites (currently available at the Internet Archive, see [https://web.archive.org/web/20090423093625/http://idlastro.gsfc.nasa.gov/idl\\_html\\_help/](https://web.archive.org/web/20090423093625/http://idlastro.gsfc.nasa.gov/idl_html_help/)), and several published books which describes IDL syntax and library routines (Fanning (2003), Bowman (2005), Gumley (2010), Galloy (2015)).

For developer convenience, GDL comes with its own integrated development environment (IDE), GDL Workbench, based on Eclipse Rich Client Platform (RCP), to aid users to simply edit IDL scripts and run them with GDL. In addition, there is an ongoing effort to maintain a GDL Jupyter kernel `gdL_kernel` providing GDL data analysis and plotting functionalities via familiar interactive notebook interface.

GDL is interoperable with Python, by featuring a bi-directional Python bridge offering access to IDL/GDL code from Python and vice versa.

GDL, GDL Workbench and `gdL_kernel` are free/libre and open-source software publicly available under the terms of the GNU General Public License v2 or above.

## Highlights in GDL 1.0

From GDL 1.0, the plotting part has been completely revamped based on wxWidgets by default for all platforms, which provides widget support and also guarantees the same behavior across different operating systems. GDL 1.0 also provides improved file format support, including reading/writing geoTIFF and writing HDF files. From the release of 1.0, the GDL project is automatically compiled and released via Github Actions, providing source code distribution and a convenient installer for Windows systems. In addition to new features, many performances and threading optimizations were also done. Map projections support in GDL is also much more mature compared to earlier major releases of GDL. Finally, the transition from SourceForge to GitHub that preceded the release of GDL 1.0 helped to consolidate the community efforts, streamline development, and support workflows.

## Design and Dependencies

GDL interpreter is written in C++ using the ANTLR framework. The library routines are written either in C++ or in GDL itself. Build and test automation is handled with CMake.

Continuous integration is set up with GitHub Actions.

GDL command-line interface is built on top of readline and ncurses. Basic array-handling, numerical processing, and I/O functionalities are implemented using Eigen, GSL, FFTW, OpenMP and zlib. Graphical output (screen and files) and widget handling is realised using pyplot and wxWidgets. Support for various file formats is implemented using netCDF, HDF4, HDF5, Shapelib, Magick++ or GraphicsMagick and Expat. Map projections are handled using PROJ. An evolving list of all project dependencies is maintained at the project website and can also be extracted from the CMake configuration files.

## Availability

GDL can be installed via the package managing system on Linux (Arch, Debian, Fedora, Gentoo, Mageia, Ubuntu), FreeBSD and macOS (Homebrew and Macports). On Windows, we provide a precompiled Windows binary, wrapped as a convenient installer based on the Nullsoft Scriptable Install System (NSIS). The source code of GDL, GDL Workbench, and `gdl_kernel` is available at our GitHub repository (<https://github.com/gnudatalanguage>). The weekly unstable builds can be obtained from <https://github.com/gnudatalanguage/gdl/releases>.

## Support and Contributions

GDL resources are being catalogued at the GDL website (<https://gnudatalanguage.github.io/>). The preferred and effective way to report requests for support, missing features or bugs present is through the GitHub issue tracker (<https://github.com/gnudatalanguage/gdl/issues>).

To streamline review and incorporation of code contributions, pull requests on GitHub are preferred. All contributed code must comply with the GNU General Public License.

## Author Contributions

JP has served as release manager for GDL 1.0 and has been the key contributor to Windows OS support, the GDL Workbench IDE for GDL and the continuous integration setup. GD has contributed a major rework of widgets and plotting subsystems to the 1.0 release. GD and AC have been the key maintainers and developers of GDL over the last decade. GVJ has contributed support for Windows OS, newer IDL datatypes, and library routines. LS started the `gdl_kernel` project. BB, SF, OG, THi, THu, JK, ERdL, JT, JW, and RAS contributed library routines and tests. OS, TT, OP, and TE maintain GDL packages. SA had been an active contributor in years 2009-2015 and has since kept contributing to project maintenance. JP and SA wrote the paper draft, and all the other authors have reviewed and confirmed the text.

## Acknowledgements

GDL development benefited from code contributions, bug reports, and feedback from numerous other developers and users.

GDL development had been hosted at Sourceforge over 2003-2018 and has moved to Github.

## References

Bowman, K. P. (2005). *An introduction to programming with IDL interactive data language*. Elsevier. <http://worldcat.org/oclc/1285831627>

- Burrage, M. K., Shanmuganathan, M., Zhang, Q., Hann, E., Popescu, I. A., Soundarajan, R., Chow, K., Neubauer, S., Ferreira, V. M., & Piechnik, S. K. (2021). Cardiac stress T1-mapping response and extracellular volume stability of MOLLI-based T1-mapping methods. *Scientific Reports*, 11(1), 1–13. <https://doi.org/10.1038/s41598-021-92923-4>
- Castro-Tirado, A. J., Østgaard, N., Göğüş, E., Sánchez-Gil, C., Pascual-Granado, J., Reglero, V., Mezentsev, A., Gabler, M., Marisaldi, M., Neubert, T., & others. (2021). Very-high-frequency oscillations in the main peak of a magnetar giant flare. *Nature*, 600(7890), 621–624. <https://doi.org/10.1038/s41586-021-04101-1>
- Chen, D., Zheng, Q., Sun, L., Ji, M., Li, Y., Deng, H., & Zhang, H. (2021). ORF3a of SARS-CoV-2 promotes lysosomal exocytosis-mediated viral egress. *Developmental Cell*, 56(23), 3250–3263. <https://doi.org/10.1016/j.devcel.2021.10.006>
- Coates, M. I., Tietjen, K., Olsen, A. M., & Finarelli, J. A. (2019). High-performance suction feeding in an early elasmobranch. *Science Advances*, 5(9), eaax2742. <https://doi.org/10.1126/sciadv.aax2742>
- Coulais, Alain, Duvert, G., Jung, G., Arabas, S., Flinois, S., & Lounis, A. S. (2019). GDL - GNU Data Language 0.9.9. In P. J. Teuben, M. W. Pound, B. A. Thomas, & E. M. Warner (Eds.), *Astronomical data analysis software and systems XXVII* (Vol. 523).
- Coulais, A., Schellens, M., Arabas, S., Lenoir, M., Noreksal, L., & Erard, S. (2012). Space Missions: Long Term Preservation of IDL-based Software using GDL. In P. Ballester, D. Egret, & N. P. F. Lorente (Eds.), *Astronomical data analysis software and systems XXI* (Vol. 461).
- Coulais, A., Schellens, M., Duvert, G., Park, J., Arabas, S., Erard, S., Roudier, G., Hivon, E., Mottet, S., Laurent, B., Pinter, M., Kasradze, N., & Ayad, M. (2014). Scaling GDL for Multi-cores to Process Planck HFI Beams Monte Carlo on HPC. In N. Manset & P. Forshay (Eds.), *Astronomical data analysis software and systems XXIII* (Vol. 485).
- Coulais, A., Schellens, M., Gales, J., Arabas, S., Boquien, M., Chaniel, P., Messmer, P., Fillmore, D., Poplawski, O., Maret, S., Marchal, G., Galmiche, N., & Mermet, T. (2010). Status of GDL - GNU Data Language. In Y. Mizumoto, K.-I. Morita, & M. Ohishi (Eds.), *Astronomical data analysis software and systems XIX* (Vol. 434). <https://doi.org/10.48550/arXiv.1101.0679>
- Duvert, G., Coulais, A., & Schellens, M. (2020). GDL Today: Reaching a Viable Alternative to IDL. In P. Ballester, J. Ibsen, M. Solar, & K. Shortridge (Eds.), *Astronomical data analysis software and systems XXVII* (Vol. 522). <https://doi.org/10.48550/arXiv.1909.02371>
- Fanning, D. W. (2003). *IDL programming techniques* (2nd ed.). Fanning Software Consulting. <http://worldcat.org/oclc/249962947>
- Galloy, M. D. (2015). *Modern IDL: A guide to IDL programming*. <http://worldcat.org/oclc/1000154387>
- Gumley, L. E. (2010). *Practical IDL programming: Creating effective data analysis and visualization applications*. Morgan Kaufmann. <http://worldcat.org/oclc/900611723>
- Lee, H.-J., Jo, Y.-J., Kim, S., Kim, D., Kim, J.-M., Choi, D., Jo, H.-Y., Bak, J., Park, S.-Y., Jeon, W., & others. (2022). Transboundary aerosol transport process and its impact on aerosol-radiation-cloud feedbacks in springtime over northeast asia. *Scientific Reports*, 12(1), 1–10. <https://doi.org/10.1038/s41598-022-08854-1>
- Xiao, X., Wang, L., Wang, Z., & Wang, Z. (2022). Superheating of grain boundaries within bulk colloidal crystals. *Nature Communications*, 13(1), 1–9. <https://doi.org/10.1038/s41467-022-29254-z>