The Journal of Open Source Software

Tutorials for the GRASS geocomputation engine

Brendan A. Harmon [1], **Veronica Andreo** [2], **Anna Petrasova** [3], **Vaclav Petras** [3], **Caitlin Haedrich** [3], **Corey White** [3], and **Laura Belica** [3]

1 Louisiana State University, United States **2** Instituto Gulich, Argentina **3** North Carolina State University, United States

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Open Journals

Reviewers:

- @openjournals

Submitted: 01 January 1970
Published: unpublished

Summary

This collection of tutorials is an introduction to the GRASS geospatial processing engine. GRASS is an open source computational engine for spatiotemporal data management, analysis, modeling, and simulation (GRASS Development Team et al., 2025; Neteler & Mitášová, 2008). As an engine that can be integrated in data science pipelines with shell scripting, Python, R, Jupyter, and Colab, there are many ways to use GRASS. While GRASS already had extensive documentation of individual processing tools, tutorials were needed to introduce the many ways to interface with the tools and combine them into computational workflows (Figure 1). These open education tutorials - which cover integrations, core features, and disciplinary applications - were developed as part of an effort to grow the GRASS community. The tutorials are built with Quarto and are deployed as webpages paired with Jupyter computational notebooks. The tutorials are available at https://grass-tutorials.osgeo.org under both the GNU Free Documentation License v1.2 or later and the Creative Commons Attribution-ShareAlike 4.0 International License.

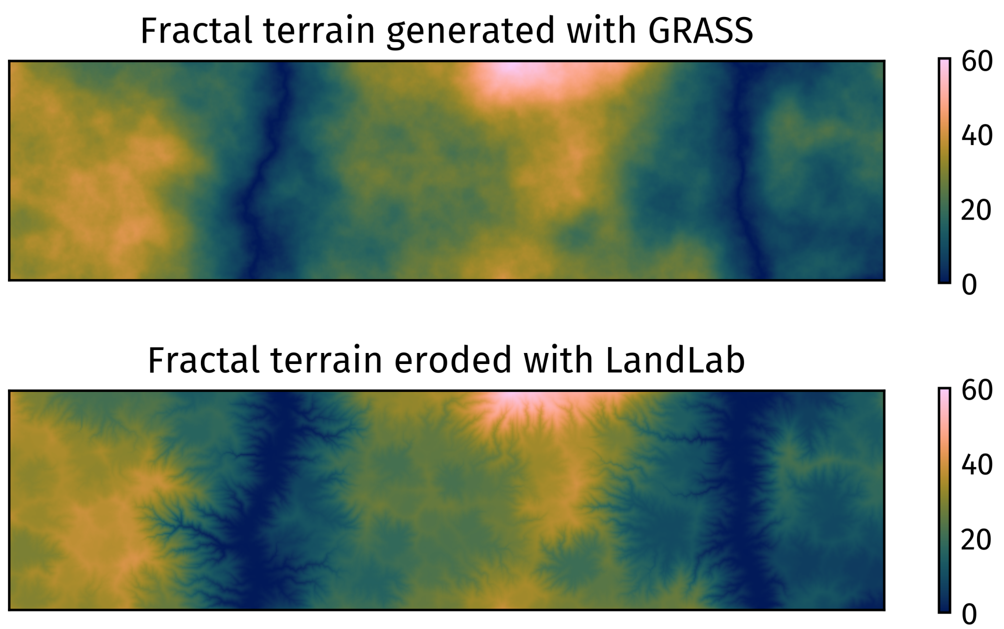

Figure 1: Tutorial on using GRASS, NumPy, and Landlab for Scientific Modeling. This tutorial demonstrates a seamless workflow for scientific modeling in Python, showing how gridded data can be passed as arrays between GRASS, NumPy (Harris et al., 2020), and Landlab (Barnhart et al., 2020).

Statement of Need

As GRASS has grown from its roots as a geographic information system (Westervelt, 2004), it has evolved into a geocomputational engine with many interfaces. As an engine, it can be integrated in geospatial data science pipelines using shell scripting, application programming interfaces, tangible interfaces (Petrášová et al., 2018), computational notebooks (Haedrich et al., 2023), cloud computing environments (Neteler et al., 2019; White et al., 2023), or high performance computing environments. While GRASS is well documented with books, a user manual, developer manuals, and a wiki, it lacked official tutorials. Over the years, the community developed many tutorials across different platforms, but as these are independently maintained, many have become outdated and obsolete. The current roadmap for GRASS – established in 2024 – calls for official new tutorials to encourage community growth and demonstrate integrations in data science pipelines.

The design and implementation of the new official tutorials for GRASS was based on experience teaching university courses and conference workshops using open educational resources. Over the last decade, the GRASS community has developed many open educational resources, experimenting with delivery via web documents, computational notebooks, and cloud computing services. Online tutorials for GRASS have been built from source in HTML (Petráš et al., 2015), built from Markdown with a static site generator (Harmon, 2020), included Jupyter notebooks (Haedrich et al., 2023), and used cloud computing services such as Binder (Petrášová & Petráš, 2019), Whole Tale (Andreo, 2023b), and Google Colab (Andreo, 2023a). Petráš et al. (2015) used a modular structure with tabsets to teach the core interfaces for GRASS – the GUI, CLI, and Python API – separating explanatory text introducing geospatial concepts from software specific text for each interface. This scaffolding helps learners to focus on concepts, while building their skills with increasingly complex interfaces. Haedrich et al. (2023) developed a GRASS–Jupyter integration to incorporate more scripting into a graduate-level course on Geospatial Computing and Simulation. The package extends the existing GRASS Python APIs with data visualization and management tools for the Jupyter environment. The new course materials include Jupyter Notebooks that combine tutorials and assignments, allowing students to write and modify code, interact with examples, and explain their reasoning in markdown, all within a single document. Based on these experiences, our design principles for the new tutorials include teaching geospatial concepts discretely from software specifics to encourage spatial thinking, supporting live coding to encourage computational thinking, and using an open source publishing system to build documents from plain text tracked with version control.

Description

Learning Objectives

This collection of tutorials was designed to teach geocomputational thinking using the GRASS geoprocessing engine. To introduce computational approaches (National Research Council, 2010; Weintrop et al., 2016) to thinking about space and time, the tutorials cover the fundamentals of geoprocessing with GRASS, integrations of GRASS into data science pipelines, and disciplinary applications of GRASS. The tutorials were designed for self-study by learners of all levels, integration into courses, and deployment in workshops.

Instructional Design

In order to teach a computational approach to thinking about spatiotemporal phenomena through different interfaces to the GRASS engine, the tutorials were designed:

- as modules for reuse and remixing,
- as worked examples to reduce cognitive load,
- as interactive lessons for active learning and engagement,

- as scaffolded prose and code to structure learning,
- and as computable content to teach computational thinking.

Drawing on the education benefits of computational notebooks (Barba et al., 2022), the tutorials introduce geocomputational concepts through worked examples that synthesize prose explanations, graphics, and executable code. The tutorials, which range from introductory to advanced, have a modular design for reuse and remixing so that learners can choose their own course of study and teachers can select modules for their lesson plans. The tutorials build in complexity from introductory to core to disciplinary modules. A set of "Getting Started" tutorials introduce different ways to interface with the GRASS engine. The core tutorials cover important concepts such as geovisualization, map algebra, geospatial modeling, and the temporal framework. Disciplinary tutorials demonstrate applications for GRASS in domains such as climatology, ecology, hydrology, geomorphology. The disciplinary tutorials build engagement by working through applications in the learner's domain and thus motivate further exploration. Throughout the tutorials, different ways to interface with GRASS are presented as tabs in code blocks, so that learners can work their way through the same tutorial repeatedly using increasingly challenging interfaces – building proficiency first with the graphical user interface (GUI), then with the command line interface (CLI), and finally with the Python or R application programming interfaces (API).

Implementation

This collection of tutorials was published as web documents paired with computational notebooks. To reach a broad audience, the tutorials are published as web documents for immediate, easy access via web browsers. When appropriate, web documents are accompanied by a downloadable computational notebook, encouraging interactivity, engagement, and geocomputational thinking. The tutorials – which are built and deployed using the Quarto scientific publishing system (Allaire et al., 2025) – are written in Markdown with YAML frontmatter. Tutorials are composed in Markdown for human-readable source code, efficient version control, executable code blocks for different interfaces, rendering in multiple formats, and reproducibility. As this open education project aims to teach different ways to interface with GRASS, executable code for multiple relevant interfaces such as the GUI, CLI, Python, or R can be included in tutorials as tabsets. Once tutorials have been written, they are reviewed by the GRASS Development Team, rendered as web documents and Jupyter notebooks, and deployed to an Open Source Geospatial Foundation website. The source code for the tutorials is built in the GitHub repository https://github.com/OSGeo/grass-tutorials and deployed to the website https://grass-tutorials.osgeo.org using GitHub Actions.

Content

This official collection of tutorials is maintained by the GRASS Development Team as part of the documentation for the GRASS geocomputational engine (Table 1). This ensures that tutorials are standardized, undergo rigorous review, and are maintained and updated as GRASS evolves. The website also includes a curated collection of community contributed tutorials that are hosted on external websites and maintained by their creators (Table 2).

Table 1: Official GRASS tutorials

Modules	Tutorials	Level	Lang.
Integrations	Get started with GRASS GUI	Beginner	En
	Get started with GRASS & Python in Jupyter Notebooks	Beginner	En
	Get started with GRASS in Google Colab	Beginner	En
	Get started with GRASS in Jupyter Notebooks on Windows	Beginner	En

Modules	Tutorials	Level	Lang.
	Get started with GRASS & R: the rgrass package	Advanced	En
Core	Basics of map algebra	Beginner	En
	Making plots with GRASS	Beginner	En
	Visualizing and modeling terrain from DEMs in GRASS	Beginner	En & Pt
	Introduction to remote sensing with GRASS	Beginner	En
	Quick comparison: R and Python GRASS interfaces	Intermediate	En
	Introduction to time series in GRASS	Intermediate	En
	Temporal subset, import and export	Intermediate	En
	Temporal aggregations	Advanced	En
	Temporal algebra	Advanced	En
	Temporal accumulation	Advanced	En
	Temporal gap-filling	Advanced	En
	Temporal query with vector data	Advanced	En
	Modeling movement in GRASS	Advanced	En & Pt
Disciplinary	Basic earthworks	Beginner	En
	Gully modeling	Beginner	En
	Coastal infrastructure	Beginner	En
	Terrain synthesis	Intermediate	En
	Procedural noise	Intermediate	En
	Hydro-flattening a digital elevation model	Intermediate	En
	Using GRASS, NumPy, and Landlab for scientific modeling	Intermediate	En
	Estimating wind fetch	Advanced	En
	Parallelization of overland flow simulation	Advanced	En

Table 2: Community contributed tutorials

Modules	Tutorials	No.	Level	Lang.
Integrations	Unleash the power of GRASS GIS	5	Beginner - Advanced	En
	GRASS for remote sensing data processing with Jupyter Notebooks	1	Advanced	En
Core	NCSU geospatial modeling and analysis course	13	Beginner - Intermediate	En
	Geoprocessamento com GRASS GIS	1	Beginner - Intermediate	Pt
	Tutoriales de GRASS GIS en grasswiki	4	Beginner - Intermediate	Es
	GISMentors	30	Beginner - Advanced	En & Cs
Disciplinary	Deforestation study using GRASS GIS	1	Beginner	En
	Teledetección, OBIA y series de tiempo	5	Beginner - Advanced.	Es
	GIS for designers	12	Beginner - Intermediate	En
	GRASS GIS for environmental monitoring and disease ecology	2	Beginner - Intermediate	En

Modules	Tutorials	No.	Level	Lang.
	Processing lidar and UAV point clouds	1	Beginner - Intermediate	En
	Physically-based hydrologic modeling using r.topmodel	1	Intermediate	En
	Spatio-temporal data handling and visualization	1	Intermediate	En
	Ecodiv.earth tutorials	16	Beginner - Advanced	En
	Urban growth modeling with FUTURES	1	Advanced	En

Acknowledgements

The initial development of these tutorials was partially supported by the U.S. National Science Foundation under Grant 2303651.

References

Allaire, J. J., Teague, C., Scheidegger, C., Xie, Y., Dervieux, C., & Woodhull, G. (2025). *Quarto* (Version 1.8). https://doi.org/10.5281/zenodo.5960048

Andreo, V. (2023a). *Procesamiento y análisis de series temporales con GRASS GIS*. https://veroandreo.github.io/curso-grass-gis/

Andreo, V. (2023b). *Using satellite data for species distribution modeling with GRASS GIS and r*. https://veroandreo.github.io/grass_ncsu_2023/studio_index.html

Barba, L. A., Barker, L. J., Blank, D. S., Brown, J., Downey, A., George, T., Heagy, L. J., Mandli, K., Moore, J. K., Lippert, D., Niemeyer, K., Watkins, R., West, R., Wickes, E., Willling, C., & Zingale, M. (2022). *Teaching and Learning with Jupyter*. https://doi.org/10.6084/m9.figshare.19608801.v1

Barnhart, K. R., Hutton, E. W., Tucker, G. E., Gasparini, N. M., Istanbulluoglu, E., Hobley, D. E., Lyons, N. J., Mouchene, M., Nudurupati, S. S., Adams, J. M., & Bandaragoda, C. (2020). Landlab v2. 0: A software package for earth surface dynamics. *Earth Surface Dynamics*, *8*(2), 379–397. https://doi.org/10.5194/esurf-8-379-2020

GRASS Development Team, Landa, M., Neteler, M., Metz, M., Petrášová, A., Petráš, V., Clements, G., Zigo, T., Larsson, N., Kladivová, L., Haedrich, C., Blumentrath, S., Andreo, V., Cho, H., Gebbert, S., Nartišs, M., Kudrnovsky, H., Delucchi, L., Zambelli, P., … Bowman, H. (2025). *GRASS GIS* (Version 8.4.0). https://doi.org/10.5281/zenodo.4621728

Haedrich, C., Petráš, V., Petrášová, A., Blumentrath, S., & Mitášová, H. (2023). Integrating GRASS GIS and jupyter notebooks to facilitate advanced geospatial modeling education. *Transactions in GIS*, *27*(3), 686–702. https://doi.org/10.1111/tgis.13031

Harmon, B. (2020). *GIS for designers*. https://baharmon.github.io/gis-for-designers

Harris, C. R., Millman, K. J., Walt, S. J. van der, Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., Kerkwijk, M. H. van, Brett, M., Haldane, A., Río, J. F. del, Wiebe, M., Peterson, P., … Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

National Research Council. (2010). *Report of a workshop on the scope and nature of computational thinking*. The National Academies Press. https://doi.org/10.17226/12840

Neteler, M., Gebbert, S., Tawalika, C., Bettge, A., Benelcadi, H., Löw, F., Adams, T., & Paulsen, H. (2019). Actinia: Cloud based geoprocessing. *Proceedings of the 2019*

Conference on Big Data from Space (BiDS'2019)*, 41–44. https://doi.org/10.5281/zenodo.2631917

Neteler, M., & Mitášová, H. (2008). *Open source GIS: A GRASS GIS approach*. Springer. https://doi.org/10.1007/978-0-387-68574-8

Petráš, V., Petrášová, A., Harmon, B. A., Meentemeyer, R. K., & Mitášová, H. (2015). Integrating free and open source solutions into geospatial science education. *ISPRS International Journal of Geo-Information*, *4*(2), 942–956. https://doi.org/10.3390/ijgi4020942

Petrášová, A., Harmon, B., Petráš, V., Tabrizian, P., & Mitášová, H. (2018). *Tangible Modeling with Open Source GIS* (2nd ed.). Springer. https://doi.org/10.1007/978-3-319-89303-7

Petrášová, A., & Petráš, V. (2019). *FUTURES model introduction using jupyter notebook*. https://github.com/ncsu-landscape-dynamics/futures-model-intro-notebook

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology*, *25*(1), 127–147.

Westervelt, J. (2004, July). GRASS roots. *Proceedings of the FOSS/GRASS User Conference*. https://grass.osgeo.org/files/westervelt2004_GRASS_roots.pdf

White, C. T., Petrášová, A., Petráš, V., Tateosian, L. G., Vukomanovic, J., Mitášová, H., & Meentemeyer, R. K. (2023). An open-source platform for geospatial participatory modeling in the cloud. *Environmental Modelling & Software*, *167*, 105767. https://doi.org/https://doi.org/10.1016/j.envsoft.2023.105767