

Tutorials for the GRASS geocomputation engine

Brendan A. Harmon¹, Veronica Andreo², Anna Petrasova³, Vaclav Petras³, Caitlin Haedrich³, Corey White³, and Laura Belica³

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Published: unpublished

License

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

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.

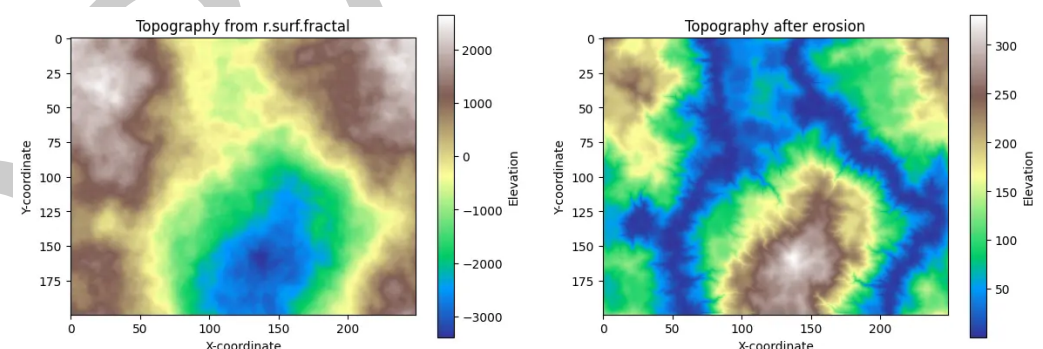


Figure 1: Tutorial on using GRASS, NumPy, and Landlab for Scientific Modeling. This tutorial demonstrates a seamless workflow for scientific modeling in Python, showing how gridded data can be passed as arrays between GRASS, NumPy (Harris et al., 2020), and Landlab (Barnhart et al., 2020).

Statement of Need

As GRASS has grown from its roots as a desktop 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 a user manual, developer manuals, and a wiki, it lacked official tutorials. Over the years, the community

28 developed many tutorials across different platforms, but as these are independently maintained,
29 many have become outdated and obsolete. The current roadmap for GRASS – established
30 in 2024 – calls for official new tutorials to encourage community growth and demonstrate
31 integrations in data science pipelines.

32 The design and implementation of the new official tutorials for GRASS was based on experience
33 teaching university courses and conference workshops using open educational resources. Over
34 the last decade, the GRASS community has developed many open educational resources,
35 experimenting with delivery via web documents, computational notebooks, and cloud computing
36 services. Online tutorials for GRASS have been built from source in HTML (Petráš et al.,
37 2015), built from Markdown rendered with Hugo, included Jupyter notebooks (Haedrich et
38 al., 2023), and used cloud computing services such as Google Colab. Petráš et al. (2015)
39 used a modular structure with tabsets to teach the core interfaces for GRASS – the GUI, CLI,
40 and Python API – separating explanatory text introducing geospatial concepts from software
41 specific text for each interface. This scaffolding helps learners to focus on concepts, while
42 building their skills with increasingly complex interfaces. Haedrich et al. (2023) developed
43 a GRASS–Jupyter integration to incorporate more scripting into a graduate-level course on
44 Geospatial Computing and Simulation. The package extends the existing GRASS Python APIs
45 with data visualization and management tools for the Jupyter environment. The new course
46 materials include Jupyter Notebooks that combine tutorials and assignments, allowing students
47 to write and modify code, interact with examples, and explain their reasoning in markdown,
48 all within a single document. Based on these experiences, our design principles for the new
49 tutorials include teaching geospatial concepts discretely from software specifics to encourage
50 spatial thinking, supporting live coding to encourage computational thinking, and using an
51 open source publishing system to build documents from plain text tracked with version control.

52 Description

53 Learning Objectives

54 This collection of tutorials was designed to teach geocomputational thinking using the GRASS
55 geoprocessing engine. To introduce computational approaches (Council, 2010; Weintrop et al.,
56 2016) to thinking about space and time, the tutorials cover the fundamentals of geoprocessing
57 with GRASS, integrations of GRASS into data science pipelines, and disciplinary applications
58 of GRASS. The tutorials were designed for self-study by learners of all levels, integration into
59 courses, and deployment in workshops.

60 Instructional Design

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

- 63 ■ as modules for reuse and remixing,
- 64 ■ as worked examples to reduce cognitive load,
- 65 ■ as interactive lessons for active learning and engagement,
- 66 ■ as scaffolded prose and code to structure learning,
- 67 ■ and as computable content to teach computational thinking.

68 Drawing on the education benefits of computational notebooks (Barba et al., 2022), the
69 tutorials introduce geocomputational concepts through worked examples that synthesize prose
70 explanations, graphics, and executable code. The tutorials, which range from introductory
71 to advanced, have a modular design for reuse and remixing so that learners can choose their
72 own course of study and teachers can select modules for their lesson plans. The tutorials
73 build in complexity from introductory to core to disciplinary modules. A set of “Getting
74 Started” tutorials introduce different ways to interface with the GRASS engine. The core
75 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 application programming interface (API).

Implementation

This collection of tutorials was published using Quarto 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 undergo rigorous review, tutorials are maintained and updated as GRASS evolves, and issues are promptly addressed. 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	Language
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
	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

Modules	Tutorials	Level	Language
Disciplinary	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
	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	Language
Integrations	Unleash the power of GRASS GIS	5	Beg - Adv	En
	GRASS for Remote Sensing data processing with Jupyter Notebooks	1	Adv	En
Core	NCSU Geospatial Modeling and Analysis Course	13	Beg - Int	En
	Geoprocessingo.com GRASS GIS	1	Beg - Int	Pt
	Tutoriales de GRASS GIS en grasswiki	4	Beg - Int	Es
	GISMentors	30	Beg - Adv	En & Cs
	Deforestation study using GRASS GIS	1	Beg	En
	Teledetección, OBIA y series de tiempo	5	Beg - Int	Es
	GIS for Designers	12	Beg - Int	En
	OpenGeoHub 2019: GRASS GIS for environmental monitoring and disease ecology	2	Beg - Int	En
	Processing lidar and UAV point clouds	1	Beg - Int	En
	Physically-based hydrologic modeling using GRASS GIS r.topmodel	1	Int	En
	Spatio-temporal data handling and visualization	1	Int	En
	Ecodiv.earth tutorials	16	Beg - Adv	En
	Urban growth modeling with FUTURES	1	Adv	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>
- 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., Willing, 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., Istanbuluoglu, 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>
- Council, N. R. (2010). *Report of a workshop on the scope and nature of computational thinking*. The National Academies Press. <https://doi.org/10.17226/12840>
- 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>
- 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>
- 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>
- 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

153 in the cloud. *Environmental Modelling & Software*, 167, 105767. <https://doi.org/https://doi.org/10.1016/j.envsoft.2023.105767>
154

DRAFT