

1 Tutorials for the GRASS geocomputation engine

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

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

DOI: [10.xxxxxx/draft](https://doi.org/10.xxxxxx/draft)

Software

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

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Published: unpublished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](#)).

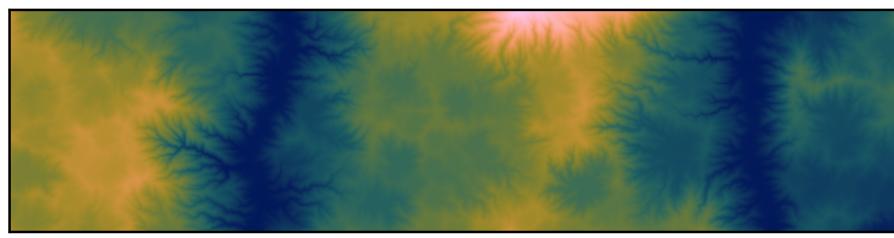


Figure 1: An example of fractal terrain generated with GRASS and eroded with LandLab from a [tutorial](#) on scientific modeling. This tutorial demonstrates a seamless workflow for scientific modeling in Python, showing how gridded data can be passed as arrays between GRASS, NumPy ([Harris et al., 2020](#)), and Landlab ([Barnhart et al., 2020](#)).

6 Summary

7 This collection of tutorials is an introduction to the GRASS geospatial processing engine.
8 GRASS is an open source computational engine for spatiotemporal data management,
9 analysis, modeling, and simulation ([GRASS Development Team et al., 2025](#); [Neteler &](#)
10 [Mitasova, 2008](#)). As an engine that can be integrated in data science pipelines with shell
11 scripting, Python, R, Jupyter, and Colab, there are many ways to use GRASS. While
12 GRASS already had extensive documentation of individual processing tools, tutorials
13 were needed to introduce the many ways to interface with the tools and combine them
14 into computational workflows ([Figure 1](#)). These open education tutorials - which cover
15 integrations, core features, and disciplinary applications - were developed as part of an
16 effort to grow the GRASS community. The tutorials are built with Quarto and are deployed
17 as webpages that are paired, when appropriate, with Jupyter computational notebooks.
18 The tutorials are available at <https://grass-tutorials.osgeo.org> under both the GNU Free
19 Documentation License v1.2 or later and the Creative Commons Attribution-ShareAlike
20 4.0 International License.

21 Statement of Need

22 As GRASS has grown from its roots as a geographic information system ([Westervelt, 2004](#)),
23 it has evolved into a geocomputational engine with many interfaces. As an engine, it can
24 be integrated in geospatial data science pipelines using shell scripting, application pro-
25 gramming interfaces, tangible interfaces ([Petrasova et al., 2018](#)), computational notebooks
26 ([Haedrich et al., 2023](#)), cloud computing environments ([Neteler et al., 2019; White et al.,](#)
27 [2023](#)), or high performance computing environments. While GRASS is well documented
28 with books, a user manual, developer manuals, and a wiki, it lacked official tutorials. Over

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

33 The design and implementation of the new official tutorials for GRASS was based on
34 experience teaching university courses and conference workshops using open educational
35 resources. Over the last decade, the GRASS community has developed many open
36 educational resources, experimenting with delivery via web documents, computational
37 notebooks, and cloud computing services. Online tutorials for GRASS have been built
38 from source in HTML (Petras et al., 2015), built from Markdown with a static site
39 generator (Harmon, 2020), included Jupyter notebooks (Haedrich et al., 2023), and used
40 cloud computing services such as Binder (Petrasova & Petras, 2019), Whole Tale (Andreou,
41 2023b), and Google Colab (Andreou, 2023a). Petras et al. (2015) used a modular structure
42 with tabs to teach the core interfaces for GRASS – the GUI, CLI, and Python API –
43 separating explanatory text introducing geospatial concepts from software specific text
44 for each interface. This scaffolding helps learners to focus on concepts, while building
45 their skills with increasingly complex interfaces. Haedrich et al. (2023) developed the
46 GRASS–Jupyter integration to incorporate more scripting into a graduate-level course on
47 geospatial computing and simulation. The package extends the existing GRASS Python
48 APIs with data visualization and management tools for the Jupyter environment. The
49 new course materials include Jupyter Notebooks that combine tutorials and assignments,
50 allowing students to write and modify code, interact with examples, and explain their
51 reasoning in markdown, all within a single document. Based on these experiences, our
52 design principles for the new tutorials include teaching geospatial concepts discretely
53 from software specifics to encourage spatial thinking, supporting live coding to encourage
54 computational thinking, and using an open source publishing system to build documents
55 from plain text tracked with version control.

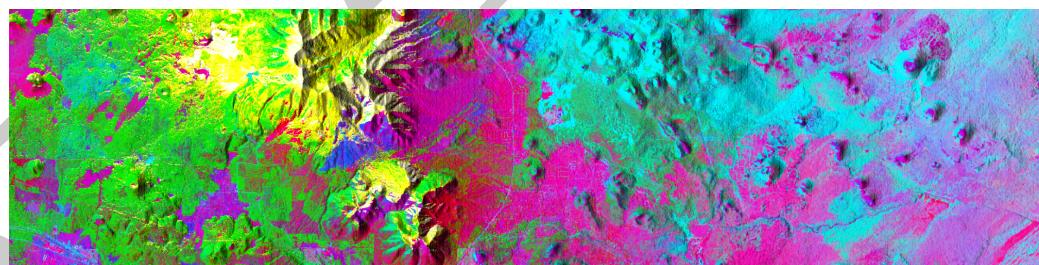


Figure 2: An example of image fusion of principal components analysis of multi-band images of the San Francisco volcanic field from a [tutorial](#) introducing the basics of remote sensing in GRASS. This tutorial demonstrates how to process and visualize multi-band remote sensing imagery.

56 Description

57 Learning Objectives

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

64 Instructional Design

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

- 67 • as modules for reuse and remixing,
- 68 • as worked examples to reduce cognitive load,
- 69 • as interactive lessons for active learning and engagement,
- 70 • as scaffolded prose and code to structure learning,
- 71 • and as computable content to teach computational thinking.

72 Drawing on the education benefits of computational notebooks (Barba et al., 2022), the
 73 tutorials introduce geocomputational concepts through worked examples that synthesize
 74 prose explanations, graphics, and executable code. The tutorials, which range from
 75 introductory to advanced, have a modular design for reuse and remixing so that learners
 76 can choose their own course of study and teachers can select modules for their lesson
 77 plans. The tutorials build in complexity from introductory to core to disciplinary modules.
 78 A set of getting-started tutorials introduce different ways to interface with the GRASS
 79 engine. The core tutorials cover important concepts such as geovisualization (Figure 2),
 80 map algebra (Figure 3), geospatial modeling, and the temporal framework. Disciplinary
 81 tutorials demonstrate applications for GRASS in domains such as climatology, ecology,
 82 hydrology, geomorphology, and terrain generation (Figure 4). The disciplinary tutorials
 83 build engagement by working through applications in the learner’s domain and thus
 84 motivate further exploration. Throughout the tutorials, different ways to interface with
 85 GRASS are presented as tabs in code blocks, so that learners can work their way through
 86 the same tutorial repeatedly using increasingly challenging interfaces – building proficiency
 87 first with the graphical user interface (GUI), then with the command line interface (CLI),
 88 and finally with the Python or R application programming interfaces (API).

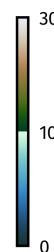
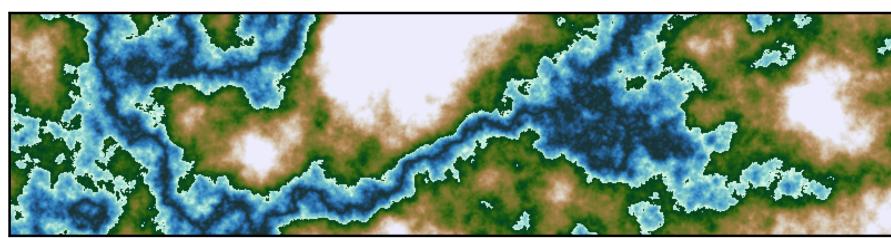


Figure 3: An example of synthetic terrain generated using map algebra from a [tutorial](#) introducing the basics of map algebra in GRASS. This tutorial demonstrates local algebraic operations using the raster map calculator, focal operations using nearest neighbors analysis, zonal operations using conditional statements with the raster map calculator, and global operations using raster metadata.

89 Implementation

90 To reach a broad audience, the tutorials in this collection are published as web documents
 91 for immediate, easy access via web browsers. When appropriate, web documents are accom-
 92 panied by a downloadable computational notebook, encouraging interactivity, engagement,
 93 and geocomputational thinking. The tutorials – which are built and deployed using the
 94 Quarto scientific publishing system (Allaire et al., 2025) – are written in Markdown with
 95 YAML frontmatter. Tutorials are composed in Markdown for human-readable source code,
 96 efficient version control, executable code blocks for different interfaces, rendering in multi-
 97 ple formats, and reproducibility. As this open education project aims to teach different
 98 ways to interface with GRASS, executable code for multiple relevant interfaces such as the
 99 GUI, CLI, Python, or R can be included in tutorials as tabssets. Once tutorials have been

¹⁰⁰ written, they are reviewed by the GRASS Development Team, rendered as web documents
¹⁰¹ and Jupyter notebooks, and deployed to an Open Source Geospatial Foundation website.
¹⁰² The source code for the tutorials is built in the GitHub repository <https://github.com/OS-Geo/grass-tutorials> and deployed to the website <https://grass-tutorials.osgeo.org> using
¹⁰³ GitHub Actions.
¹⁰⁴

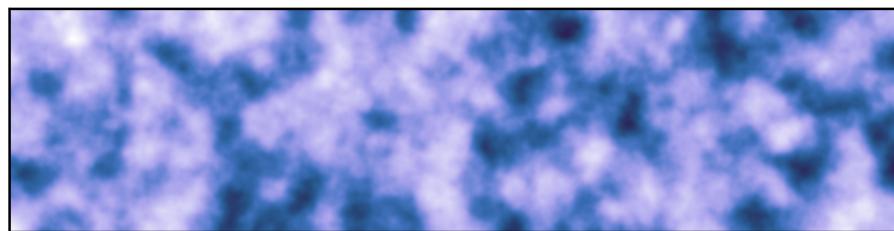


Figure 4: An example of fractional Brownian motion from a [tutorial](#) introducing procedural noise in GRASS. This tutorial demonstrates stochastic functions for procedurally generating data.

¹⁰⁵ Content

¹⁰⁶ This official collection of tutorials is maintained by the GRASS Development Team as part
¹⁰⁷ of the documentation for the GRASS geocomputational engine ([Table 1](#)). This ensures
¹⁰⁸ that tutorials are standardized, undergo rigorous review, and are maintained and updated
¹⁰⁹ as GRASS evolves. The website also includes a curated collection of external tutorials
¹¹⁰ that are hosted and maintained by their creators ([Table 2](#)).

Table 1: Official GRASS tutorials

Modules	Tutorials	Level	Lang.
Integrations	Get started with GRASS GUI	Beg	En
	Get started with GRASS & Python in Jupyter Notebooks	Beg	En
	Get started with GRASS in Google Colab	Beg	En
	Get started with GRASS in Jupyter Notebooks on Windows	Beg	En
	Quick comparison: R and Python GRASS interfaces	Int	En
	Get started with GRASS & R: the rgrass package	Adv	En
	Basics of map algebra	Beg	En
	Making plots with GRASS	Beg	En
	Visualizing and modeling terrain from DEMs in GRASS	Beg	En & Pt
	Introduction to remote sensing with GRASS	Beg	En
Core	Making thematic maps	Beg	En
	Introduction to time series in GRASS	Int	En
	Temporal subset, import and export	Int	En
	Temporal aggregations	Adv	En
	Temporal algebra	Adv	En
	Temporal accumulation	Adv	En
	Temporal gap-filling	Adv	En
	Temporal query with vector data	Adv	En
	Modeling movement in GRASS	Adv	En & Pt
	Basic earthworks	Beg	En
Disciplinary	Gully modeling	Beg	En
	Coastal infrastructure	Beg	En

Modules	Tutorials	Level	Lang.
	Terrain synthesis	Int	En
	Procedural noise	Int	En
	Hydro-flattening a digital elevation model	Int	En
	Using GRASS, NumPy, and Landlab for scientific modeling	Int	En
	fasterRaster: faster raster processing in R Using GRASS	Int	En
	Estimating wind fetch	Adv	En
	Parallelization of overland flow simulation	Adv	En

Table 2: External tutorials

Modules	Tutorials	No.	Level	Lang.
Integrations	Unleash the power of GRASS	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
	Geoprocessamento com GRASS	1	Beg - Int	Pt
Disciplinary	Tutoriales de GRASS en grasswiki	4	Beg - Int	Es
	GISMentors	30	Beg - Adv	En & Cs
	Deforestation study using GRASS	1	Beg	En
	Teledetección, OBIA y series de tiempo	5	Beg - Adv	Es
	GIS for designers	12	Beg - Int	En
	GRASS 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 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

¹¹¹ Acknowledgements

¹¹² The development of the tutorial platform and the majority of the tutorials presented in
¹¹³ this collection were supported by the U.S. National Science Foundation under Award
¹¹⁴ No. [2303651](#). This grant directly funded the development of tutorials and also funded
¹¹⁵ mentor support for community contributors working on tutorial content. Two tutorials
¹¹⁶ received additional partial support from NSF Award No. [2322073](#) granted to Natrx,
¹¹⁷ Inc. Another tutorial was developed with combined support from NSF Award No. 2322073
¹¹⁸ and USDA NRCS Award No. NR233A750023C043. We thank the GRASS development
¹¹⁹ team and the broader GRASS community for their continued support and contributions
¹²⁰ to this educational effort.

¹²¹ References

- ¹²² Allaire, J. J., Teague, C., Scheidegger, C., Xie, Y., Dervieux, C., & Woodhull, G. (2025).
¹²³ *Quarto* (Version 1.8). <https://doi.org/10.5281/zenodo.5960048>
- ¹²⁴ Andreo, V. (2023a). *Procesamiento y análisis de series temporales con GRASS GIS*.
¹²⁵ <https://veroandreo.github.io/cursograssgis/>
- ¹²⁶ 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
- ¹²⁸ Barba, L. A., Barker, L. J., Blank, D. S., Brown, J., Downey, A., George, T., Heagy,
¹²⁹ L. J., Mandli, K., Moore, J. K., Lippert, D., Niemeyer, K., Watkins, R., West, R.,
¹³⁰ Wickes, E., Willling, C., & Zingale, M. (2022). *Teaching and Learning with Jupyter*.
¹³¹ <https://doi.org/10.6084/m9.figshare.19608801.v1>
- ¹³² Barnhart, K. R., Hutton, E. W., Tucker, G. E., Gasparini, N. M., Istanbulluoglu, E., Hobley,
¹³³ D. E., Lyons, N. J., Mouchene, M., Nudurupati, S. S., Adams, J. M., & Bandaragoda,
¹³⁴ C. (2020). Landlab v2. 0: A software package for earth surface dynamics. *Earth
¹³⁵ Surface Dynamics*, 8(2), 379–397. <https://doi.org/10.5194/esurf-8-379-2020>
- ¹³⁶ GRASS Development Team, Landa, M., Neteler, M., Metz, M., Petrasova, A., Petras,
¹³⁷ V., Clements, G., Zigo, T., Larsson, N., Kladivova, L., Haedrich, C., Blumentrath, S.,
¹³⁸ Andreo, V., Cho, H., Gebbert, S., Nartiss, M., Kudrnovsky, H., Delucchi, L., Zambelli,
¹³⁹ P., ... Bowman, H. (2025). *GRASS GIS* (Version 8.4.0). <https://doi.org/10.5281/zenodo.4621728>
- ¹⁴¹ Haedrich, C., Petras, V., Petrasova, A., Blumentrath, S., & Mitasova, 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>
- ¹⁴⁴ Harmon, B. (2020). *GIS for designers*. <https://baharmon.github.io/gis-for-designers>
- ¹⁴⁵ Harris, C. R., Millman, K. J., Walt, S. J. van der, Gommers, R., Virtanen, P., Cournapeau,
¹⁴⁶ D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S.,
¹⁴⁷ Kerkwijk, M. H. van, Brett, M., Haldane, A., Río, J. F. del, Wiebe, M., Peterson, P., ...
¹⁴⁸ Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, 585(7825), 357–362.
¹⁴⁹ <https://doi.org/10.1038/s41586-020-2649-2>
- ¹⁵⁰ 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>
- ¹⁵³ Neteler, M., Gebbert, S., Tawalika, C., Bettge, A., Benelcadi, H., Löw, F., Adams, T.,
¹⁵⁴ & Paulsen, H. (2019). Actinia: Cloud based geoprocessing. *Proceedings of the 2019
¹⁵⁵ Conference on Big Data from Space (BiDS'2019)*, 41–44. <https://doi.org/10.5281/zenodo.2631917>
- ¹⁵⁷ Neteler, M., & Mitasova, H. (2008). *Open source GIS: A GRASS GIS approach*. Springer.

- 158 <https://doi.org/10.1007/978-0-387-68574-8>
- 159 Petras, V., Petrasova, A., Harmon, B. A., Meentemeyer, R. K., & Mitasova, H. (2015).
160 Integrating free and open source solutions into geospatial science education. *ISPRS
161 International Journal of Geo-Information*, 4(2), 942–956. [https://doi.org/10.3390/
163 ijgi4020942](https://doi.org/10.3390/
162 ijgi4020942)
- 164 Petrasova, A., Harmon, B., Petras, V., Tabrizian, P., & Mitasova, H. (2018). *Tangi-
165 ble Modeling with Open Source GIS* (2nd ed.). Springer. [https://doi.org/10.1007/
167 978-3-319-89303-7](https://doi.org/10.1007/
166 978-3-319-89303-7)
- 168 Petrasova, A., & Petras, V. (2019). *FUTURES model introduction using jupyter notebook*.
169 <https://github.com/ncsu-landscape-dynamics/futures-model-intro-notebook>
- 170 Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky,
171 U. (2016). Defining computational thinking for mathematics and science classrooms.
172 *Journal of Science Education and Technology*, 25(1), 127–147.
- 173 Westervelt, J. (2004, July). GRASS roots. *Proceedings of the FOSS/GRASS User
174 Conference*. https://grass.osgeo.org/files/westervelt2004_GRASS_roots.pdf
- 175 White, C. T., Petrasova, A., Petras, V., Tateosian, L. G., Vukomanovic, J., Mitasova,
176 H., & Meentemeyer, R. K. (2023). An open-source platform for geospatial participatory
177 modeling in the cloud. *Environmental Modelling & Software*, 167, 105767.
<https://doi.org/https://doi.org/10.1016/j.envsoft.2023.105767>