

# <sup>1</sup> Tutorials for the GRASS geocomputation engine

<sup>2</sup> **Brendan A. Harmon**  <sup>1</sup>, **Veronica Andreo**  <sup>2</sup>, **Anna Petrasova**  <sup>3</sup>, **Vaclav Petras**  <sup>3</sup>, **Caitlin Haedrich**  <sup>3</sup>, **Corey White**  <sup>3</sup>, and **Laura Belica**  <sup>3</sup>

<sup>4</sup> <sup>1</sup> Louisiana State University, United States  <sup>2</sup> Instituto Gulich, Argentina  <sup>3</sup> North Carolina State University, United States 

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

## Software

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

---

Editor: [Open Journals](#) 

Reviewers:

- [@openjournals](#)

Submitted: 01 January 1970

Published: unpublished

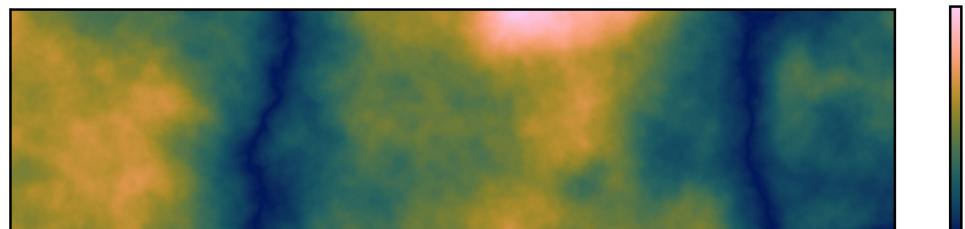
License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#)).  
<sup>17</sup>  
<sup>18</sup>  
<sup>19</sup>  
<sup>20</sup>

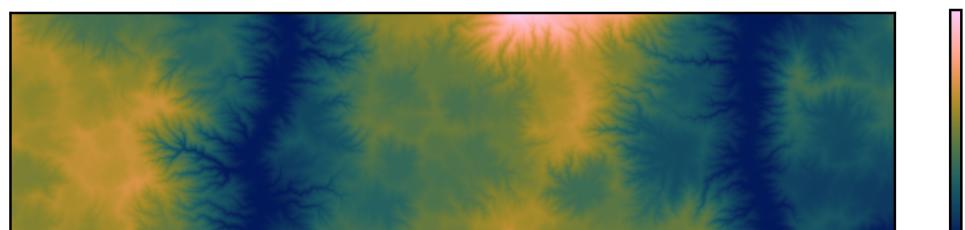
## <sup>6</sup> Summary

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

Fractal terrain generated with GRASS



Fractal terrain eroded with LandLab



**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](#)).

## 20 Statement of Need

21 As GRASS has grown from its roots as a geographic information system (Westervelt, 2004), it  
22 has evolved into a geocomputational engine with many interfaces. As an engine, it can be  
23 integrated in geospatial data science pipelines using shell scripting, application programming  
24 interfaces, tangible interfaces (Petrášová et al., 2018), computational notebooks (Haedrich  
25 et al., 2023), cloud computing environments (Neteler et al., 2019; White et al., 2023), or  
26 high performance computing environments. While GRASS is well documented with books, a  
27 user manual, developer manuals, and a wiki, it lacked official tutorials. Over the years, the  
28 community developed many tutorials across different platforms, but as these are independently  
29 maintained, many have become outdated and obsolete. The current roadmap for GRASS  
30 – established in 2024 – calls for official new tutorials to encourage community growth and  
31 demonstrate 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., 2015),  
37 built from Markdown with a static site generator (Harmon, 2020), included Jupyter notebooks  
38 (Haedrich et al., 2023), and used cloud computing services such as Binder (Petrášová & Petráš,  
39 2019), Whole Tale (Andrej, 2023b), and Google Colab (Andrej, 2023a). Petráš et al. (2015)  
40 used a modular structure with tabs to teach the core interfaces for GRASS – the GUI, CLI,  
41 and Python API – separating explanatory text introducing geospatial concepts from software  
42 specific text for each interface. This scaffolding helps learners to focus on concepts, while  
43 building their skills with increasingly complex interfaces. Haedrich et al. (2023) developed  
44 a GRASS–Jupyter integration to incorporate more scripting into a graduate-level course on  
45 Geospatial Computing and Simulation. The package extends the existing GRASS Python APIs  
46 with data visualization and management tools for the Jupyter environment. The new course  
47 materials include Jupyter Notebooks that combine tutorials and assignments, allowing students  
48 to write and modify code, interact with examples, and explain their reasoning in markdown,  
49 all within a single document. Based on these experiences, our design principles for the new  
50 tutorials include teaching geospatial concepts discretely from software specifics to encourage  
51 spatial thinking, supporting live coding to encourage computational thinking, and using an  
52 open source publishing system to build documents from plain text tracked with version control.

## 53 Description

### 54 Learning Objectives

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

### 61 Instructional Design

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

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

67        ▪ as scaffolded prose and code to structure learning,  
 68        ▪ and as computable content to teach computational thinking.

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

## 85 Implementation

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

## 101 Content

102 This official collection of tutorials is maintained by the GRASS Development Team as part  
 103 of the documentation for the GRASS geocomputational engine (Table 1). This ensures that  
 104 tutorials are standardized, undergo rigorous review, and are maintained and updated as GRASS  
 105 evolves. The website also includes a curated collection of community contributed tutorials that  
 106 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.
Core	Get started with GRASS & R: the rgrass package	Advanced	En
	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
Disciplinary	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
Disciplinary	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
Disciplinary	Parallelization of overland flow simulation	Advanced	En
		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
	NCSU geospatial modeling and analysis course	13	Beginner - Intermediate	En
	Geoprocessamento com GRASS GIS	1	Beginner - Intermediate	Pt
Core	Tutoriales de GRASS GIS en grasswiki	4	Beginner - Intermediate	Es
	GISMentors	30	Beginner - Advanced	En & Cs
	Deforestation study using GRASS GIS	1	Beginner	En
	Teledetección, OBIA y series de tiempo	5	Beginner - Advanced	Es
Disciplinary	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

## 107 Acknowledgements

108 The initial development of these tutorials was partially supported by the U.S. National Science  
 109 Foundation under Grant [2303651](https://doi.org/10.303651).

## 110 References

- 111 Allaire, J. J., Teague, C., Scheidegger, C., Xie, Y., Dervieux, C., & Woodhull, G. (2025).  
 112 *Quarto* (Version 1.8). <https://doi.org/10.5281/zenodo.5960048>
- 113 Andreo, V. (2023a). *Procesamiento y análisis de series temporales con GRASS GIS*. <https://veroandreo.github.io/curso-grass-gis/>
- 114
- 115 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](https://veroandreo.github.io/grass_ncsu_2023/studio_index.html)
- 116
- 117 Barba, L. A., Barker, L. J., Blank, D. S., Brown, J., Downey, A., George, T., Heagy,  
 118 L. J., Mandli, K., Moore, J. K., Lippert, D., Niemeyer, K., Watkins, R., West, R.,  
 119 Wickes, E., Willling, C., & Zingale, M. (2022). *Teaching and Learning with Jupyter*.  
 120 <https://doi.org/10.6084/m9.figshare.19608801.v1>
- 121 Barnhart, K. R., Hutton, E. W., Tucker, G. E., Gasparini, N. M., Istanbulluoglu, E., Hobley,  
 122 D. E., Lyons, N. J., Mouchene, M., Nudurupati, S. S., Adams, J. M., & Bandaragoda, C.  
 123 (2020). Landlab v2. 0: A software package for earth surface dynamics. *Earth Surface  
 124 Dynamics*, 8(2), 379–397. <https://doi.org/10.5194/esurf-8-379-2020>
- 125 GRASS Development Team, Landa, M., Neteler, M., Metz, M., Petrášová, A., Petrás, V.,  
 126 Clements, G., Zigo, T., Larsson, N., Kladivová, L., Haedrich, C., Blumentrath, S., Andreo,  
 127 V., Cho, H., Gebbert, S., Nartišs, M., Kudrnovsky, H., Delucchi, L., Zambelli, P., ... Bowman,  
 128 H. (2025). *GRASS GIS* (Version 8.4.0). <https://doi.org/10.5281/zenodo.4621728>
- 129 Haedrich, C., Petrás, V., Petrášová, A., Blumentrath, S., & Mitášová, H. (2023). Integrating  
 130 GRASS GIS and jupyter notebooks to facilitate advanced geospatial modeling education.  
 131 *Transactions in GIS*, 27(3), 686–702. <https://doi.org/10.1111/tgis.13031>
- 132 Harmon, B. (2020). *GIS for designers*. <https://baharmon.github.io/gis-for-designers>
- 133 Harris, C. R., Millman, K. J., Walt, S. J. van der, Gommers, R., Virtanen, P., Cournapeau, D.,  
 134 Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., Kerkwijk,  
 135 M. H. van, Brett, M., Haldane, A., Río, J. F. del, Wiebe, M., Peterson, P., ... Oliphant,  
 136 T. E. (2020). Array programming with NumPy. *Nature*, 585(7825), 357–362. <https://doi.org/10.1038/s41586-020-2649-2>
- 138 National Research Council. (2010). *Report of a workshop on the scope and nature of  
 139 computational thinking*. The National Academies Press. <https://doi.org/10.17226/12840>
- 140 Neteler, M., Gebbert, S., Tawalika, C., Bettge, A., Benelcadi, H., Löw, F., Adams, T.,  
 141 & Paulsen, H. (2019). Actinia: Cloud based geoprocessing. *Proceedings of the 2019*

- 142      Conference on Big Data from Space (*BiDS'2019*), 41–44. <https://doi.org/10.5281/zenodo.2631917>
- 143
- 144      Neteler, M., & Mitášová, H. (2008). *Open source GIS: A GRASS GIS approach*. Springer. <https://doi.org/10.1007/978-0-387-68574-8>
- 145
- 146      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>
- 147
- 148
- 149      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>
- 150
- 151      Petrášová, A., & Petráš, V. (2019). *FUTURES model introduction using jupyter notebook*. <https://github.com/ncsu-landscape-dynamics/futures-model-intro-notebook>
- 152
- 153      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.
- 154
- 155
- 156      Westervelt, J. (2004, July). GRASS roots. *Proceedings of the FOSS/GRASS User Conference*. [https://grass.osgeo.org/files/westervelt2004\\_GRASS\\_roots.pdf](https://grass.osgeo.org/files/westervelt2004_GRASS_roots.pdf)
- 157
- 158      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>
- 159
- 160
- 161