The Journal of Open Source Software

Topography-based surface water modeling in Julia, with support for infiltration and temporal developments

Odd A. Andersen [1]

1 SINTEF Digital, Dept. of Mathematics and Cybernetics, Norway

DOI: 10.21105/joss.07785

Software

- Review
- Repository
- Archive

Editor: Mehmet Hakan Satman

Reviewers:

- @tukss
- @mkitti

Submitted: 01 January 2025
Published: 01 May 2025

Summary

SWIM (Surface Water Integrated Modeling) is a Julia software package for static modeling and prediction of surface water and urban flooding based on analysis of terrain topography, terrain properties, and infrastructure.

SWIM consists of a collection of algorithms for analyzing terrain, identifying watershed boundaries, and providing a better understanding of how water accumulates and moves across the landscape. Such analyses are valuable for various purposes, including water resource management, flood modeling and mitigation, and environmental planning.

The algorithms are based on the assumption of infinitesimal flow and the identification of *spill points*. Spill-point analyses are highly computationally efficient compared to tools based on numerical simulation, making it easy to work interactively and test out various scenarios and measures. SWIM offers unique functionality, such as simplified infiltration models for both permeable and impermeable surfaces, as well as the calculation of time series that model how water accumulates or drains over time without resorting to computationally intensive numerical time-stepping approaches.

Statement of need

The damage caused by intense rainfall in urban areas can be extremely costly. The stormwater problem is also becoming more severe due to population growth, urban densification, and increasingly frequent extreme weather events. The need for long-term adaptation and planning is therefore increasingly urgent, as is the need for adequate digital tools.

Limitation on and lack of data often makes modeling and prediction of stormwater flow related to intensive rainfall challenging (Skaugen et al., 2020). Mathematical approaches to evaluate uncertainties (Beven, 2003; Zhang et al., 2019) typically rely on the ability to run large numbers of simulations and scenarios. However, use of complex hydrological simulators (DHI, 2025; Langevin, 2017) are computationally very expensive and may render the analysis intractable for many applications and potential users.

In urban areas with extensive impervious surfaces, topography is typically the primary driver of stormwater flow patterns and local accumulation of surface water. GIS-based models operate on a simplified premise where surface flow is solely determined by topography, and can provide an attractive modeling alternative (ESRI, 2024; Scalgo, 2025; Skaugen & Onof, 2014). Such models are generally easy to set up and run, and require considerably fewer computational resources than complex hydrological simulators do. However, drawbacks include:

- Typically no ability to handle temporal developments or infiltration.
- Sensitive to data resolution and uncertainties.

- Lack of open-source alternatives adaptable to particular user or researcher needs.

In response to these shortcomings, SWIM was developed as a computationally efficient, flexible open-source prototyping software library for stormwater modeling. It builds upon a GIS-based modeling approach, while enabling extension and generalization of the methodology.

Functionality

SWIM is implemented in Julia (Bezanson et al., 2017), with computational performance as a key goal. Input data consists solely of raster data (2D matrices), for topography, terrain features, infrastructure and weather data. Functionality includes:

- Static analysis, including delineating hierarchies of traps (lakes) and intermittent rivers, identifying corresponding watersheds, and estimating flow intensities across terrain.
- Integrating terrain characteristics into the analysis, including permanent water bodies (rivers, lakes, ocean), buildings, obstacles and drains.
- Simplified infiltration model that supports both permeable and impermeable surfaces.
- Dynamic analysis, including terrain response to precipitation events and infiltration over time, and routing of waters as ponds overflow.

Principle of topography-based analysis

Spill regions and flow graph

Although input data consists of raster grids, the terrain and its properties is internally represented as directed graphs, which form the basis of analysis.

The flow graph describes how water flows over the terrain from one node (grid cell) to the next. To create this graph from a digital elevation model (DEM), SWIM uses a deterministic eight-node (D8) single-flow direction (SFD) algorithm (Wilson et al., 2008) which generates a tree-structured, generally disconnected flow graph shown in Figure 1. Using the D8 algorithm, flow from a given node is always directed towards its steepest downhill neighbor, if any.

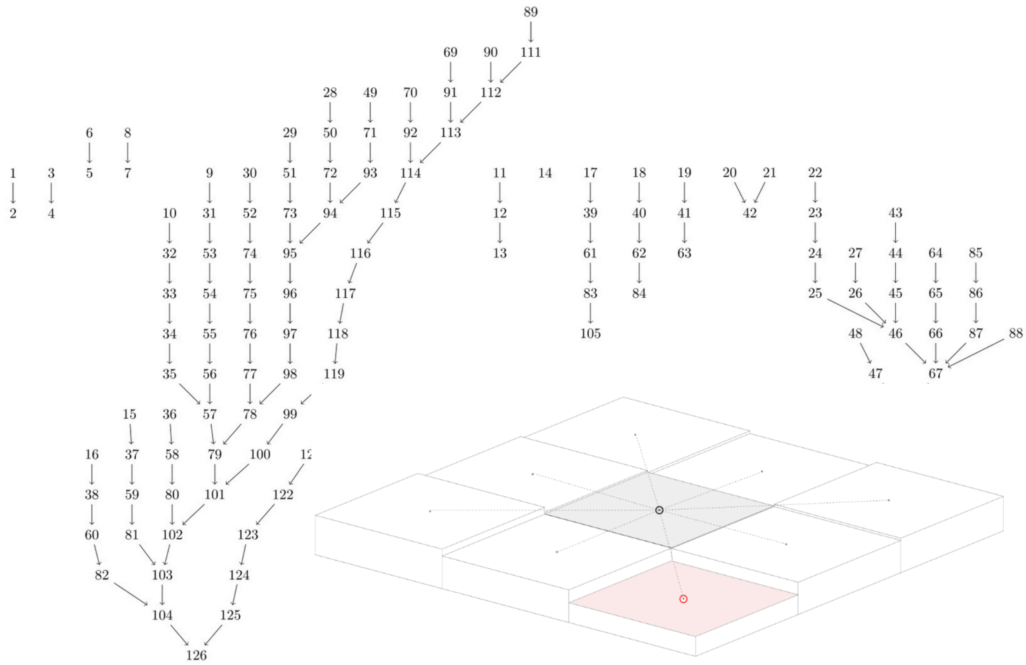

Figure 1: Spill graph generated from a small terrain grid. Inlet: selecting the steepest downstream neighbor (pink) of a cell (gray) using the D8 algorithm.

Once generated, most analysis is done directly on the flow graph using standard graph concepts and algorithms. For instance, root nodes represent accumulation points in the terrain, connected components represent the associated watersheds, and flow intensity at a given node is obtained by integrating precipitation over all its upstream nodes.

Topological structure of lakes and rivers

The spill graph serves as input to define a higher-level graph of traps ("ponds", "lakes") and the intermittent stream connecting them. Traps are delineated by identifying *spill points* (lowest elevation node along the boundary of the catchment area for a given accumulation point). A trap will fill up until its water level reaches the spillpoint, at which point it will overflow. Small traps typically coalesce into larger traps as they fill up, thereby defining a hierarchy of sub-traps and super-traps. An example of a *trap structure graph* representing both the upstream/downstream and the sub-trap/super-trap relationships between traps is shown in Figure 2.

The Journal of Open Source Software

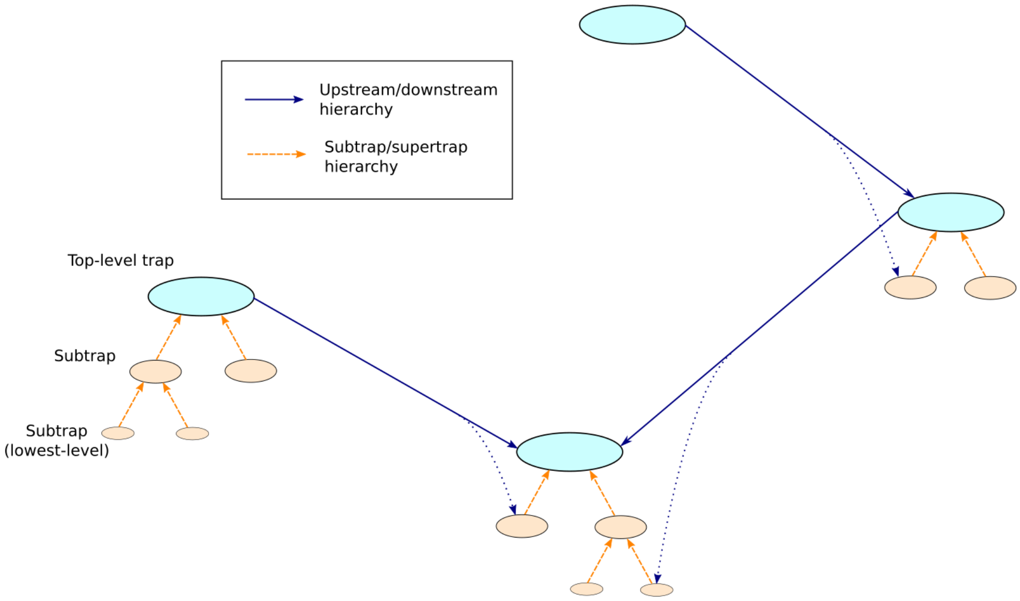

Figure 2: An example of a trap structure graph, illustrating how smaller traps coalesce into larger traps, and how upstream traps overflow into downstream traps. An upstream trap is always a top-level trap spilling into a lowest-level sub-trap downstream (indicated with dashed blue lines)

The concepts are illustrated on a simple synthetic surface in Figure 3. On the upper row, the green and red traps are sub-traps of the orange trap, which again is an upstream trap to the blue one. On the lower row, we see how water gradually accumulates, making the sub-traps coalesce, and finally how water spills over to the downstream trap.

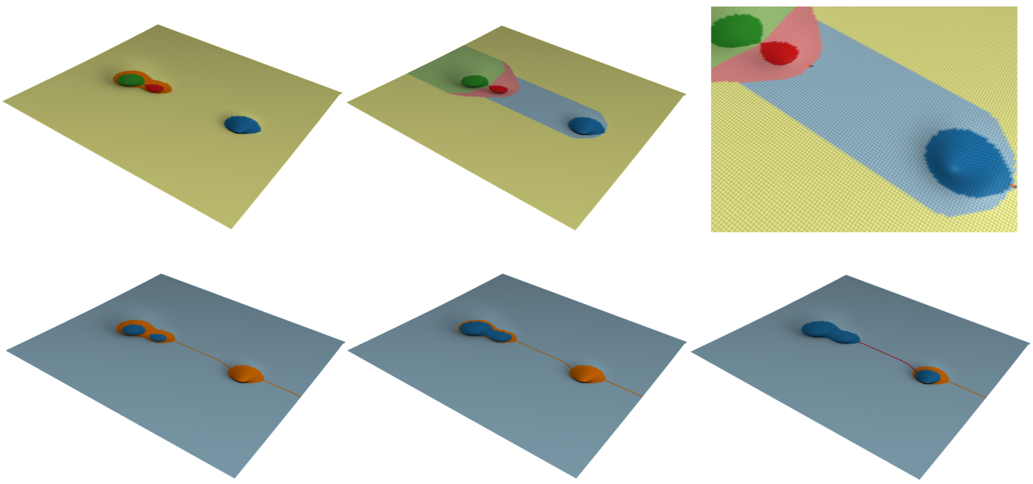

Figure 3: Small synthetic example of sub-traps, super-traps and downstream traps. The upper row shows the four traps (red, green, orange and blue) with their respective watersheds. The spill points can be seen as small black dots on the rightmost figure. The lower row illustrates the process of traps filling up, coalescing and spilling over.

The Journal of Open Source Software

Infiltration and temporal evolution

The routing of water across the terrain at any given time depends on the current state of each trap (whether it is still filling up or spilling over), current precipitation rate (which may vary spatially), and the rate of infiltration at each point. Infiltration at each node happens at a fixed rate (zero for impervious regions) capped by the node's current inflow rate. For a given weather scenario, specified infiltration rates and an initial status (typically empty) for each trap, it is possible to construct a sequence of events that identifies the moment in time when each trap fills up, and how this influences the subsequent routing of water. This allows the user to determine the surface water flow and and extract water content of each trap at any given moment, how long it takes for each trap to fill up, where water currently is accumulating, and more.

Example

The following brief example demonstrates the application of some basic SWIM functionality on real data. Figure 4 presents a DEM of the Vulkan neighborhood around the Akerselva river in central Oslo, along with four stencils which serves as additional input to the analysis, all provided as raster grids with the same resolution.

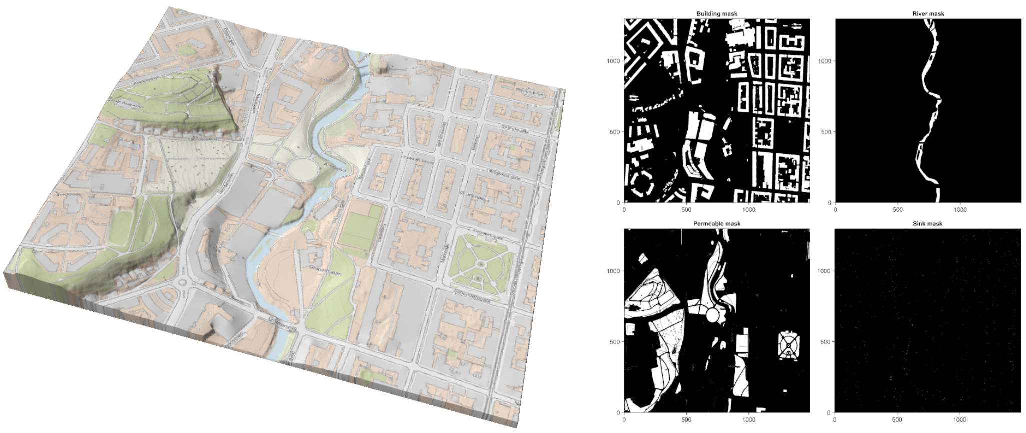

Figure 4: Vulkan area, central Oslo. The four stencils on the right represent infrastructure, water bodies (rivers), permeable surfaces and point sinks.

The end state of a scenario with constant precipitation until all traps have been filled is shown in Figure 5. Here, all traps and the intermittent streams connecting them are shown on the left plot. The right plot shows a cutout where the logarithm of flow intensity has been visualized.

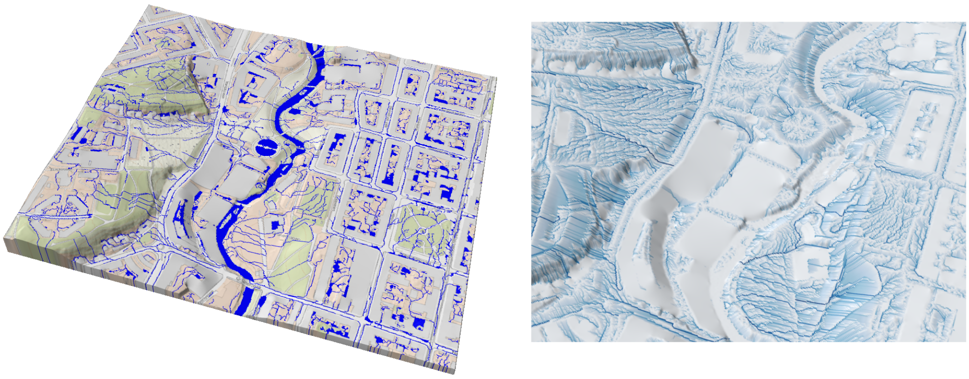

Figure 5: Vulkan area end state in a modeled precipitation scenario.

For additional and more in-depth examples, the reader is encouraged to have a look at the online documentation.

Acknowledgments

The data used in the example above was originally obtained from Kartverket (the Norwegian Mapping Authority) under the Creative Commons Attribution 4.0 International (CC BY 4.0) license.

References

Beven, K. J. (2003). *Rainfall - runoff modelling: The primer*. Wiley. ISBN: 9780470866719

Bezanson, J., Edelman, A., Karpinski, S., & Shah, V. B. (2017). Julia: A fresh approach to numerical computing. *SIAM Review*, *59*(1), 65–98. https://doi.org/10.1137/141000671

DHI. (2025). *MIKE SHE*. https://support.dhigroup.com/download/MIKE-latest/

ESRI. (2024). *ArcGIS pro* (Version 3.4). https://www.esri.com/en-us/arcgis/products/arcgis-pro/overview

Langevin, H., C. D. (2017). *MODFLOW 6 modular hydrologic model: United States Geological Survey (USGS) software*. https://doi.org/10.5066/F76Q1VQV

Scalgo. (2025). *Scalgo live*. https://scalgo.com/en-US/

Skaugen, T., Lawrence, D., & Ortega, R. Z. (2020). A parameter parsimonious approach for catchment scale urban hydrology – which processes are important? *Journal of Hydrology X*, *8*, 100060. https://doi.org/10.1016/j.hydroa.2020.100060

Skaugen, T., & Onof, C. (2014). A rainfall-runoff model parameterized from GIS and runoff data. *Hydrological Processes*, *28*(15), 4529–4542. https://doi.org/10.1002/hyp.9968

Wilson, J. P., Aggett, G., Yongxin, D., & Lam, C. S. (2008). Water in the landscape: A review of contemporary flow routing algorithms. In Q. Zhou, B. Lees, & G. Tang (Eds.), *Advances in digital terrain analysis* (pp. 213–236). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-77800-4_12

Zhang, M.-D., Zhang, J., & Song, Y. (2019). Preliminary research and application of MIKE SHE model in Jialingjiang River Basin. *IOP Conference Series: Earth and Environmental Science*, *304*. https://doi.org/10.1088/1755-1315/304/2/022088