The FastScape software stack: reusable tools for landscape evolution modelling

Benoit Bovy¹, Jean Braun¹, Guillaume Cordonnier², Raphael Lange¹, and Xiaoping Yuan¹
¹Helmholtz Centre Potsdam GFZ German Research Centre for Geosciences
²CGL, ETH Zurich

The name “FastScape” has been used to describe a landscape evolution model as well as a set of efficient algorithms to simulate various processes of erosion, transport and deposition (e.g., fluvial, hillslope and marine). We also use this name for a set of software components (https://github.com/fastscape-lem) aimed at making those models and algorithms readily accessible to a wide range of users, from experts in landscape evolution modelling to scientists, researchers and teachers in the broader Earth science community. Those software components are organised as a stack where each level has a distinct scope. At the bottom of this stack, “fastscapelib-fortran” is the original, full-featured implementation of the FastScape model, which provides a Fortran API as well as Python bindings. Its successor “fastscapelib” is a library written in modern C++ that directly exposes the FastScape algorithms (e.g., flow-routing, depression-resolving, channel erosion, hillslope diffusion) through basic APIs in C++, Python and potentially other languages such as R or Julia in the future. Built on top of those core libraries, “fastscape” is a high-level yet flexible tool that helps anyone who wants to quickly build, extend or simply run FastScape model variants in a user-friendly, interactive environment. Through its xarray-centric interface, it is deeply integrated with the rest of the Python scientific ecosystem, therefore offering great capabilities at user’s fingertips for pre/post-processing, visualisation and simulation management. One of our primary concern is following good practices (API design, testing, documentation, distribution...) while developing each of these tools. We show through a gallery of examples how the FastScape software stack has been used in research and outreach projects. We plan to provide better integration with other tools for topographic analysis/modelling (e.g., Landlab, LSDTopotools) in the future and we also greatly encourage contributions from the broader community.