xarray-topo Documentation

Release 0.0.9

xarray-topo Developers

Jun 21, 2017

Getting Started

xarray-topo is a Python package that extends xarray with a small set of topographic tools useful for processing Digital Elevation Models (DEMs) and/or outputs of Landscape Evolution Models (LEMs).

It also provides xarray-simlab compatible models and re-usable model components (i.e., processes) for landscape evolution modelling.

CHAPTER 1

Documentation index

Getting Started

- *Install xarray-topo*
- *Examples*

1.1 Install xarray-topo

1.1.1 Required dependencies

- Python 3.4 or later.
- numpy
- xarray (0.8.0 or later)
- numba (0.29.0 or later)

1.1.2 Optional dependencies

For landscape evolution modeling

- xarray-simlab

1.1.3 Install from source

There is currently no `xarray-topo` release available at PyPi or conda-forge.

Be sure you have the required dependencies (numpy and xarray) installed first. You might consider using conda to install them:

```
$ conda install xarray numba numpy pip -c conda-forge
```

A good practice (especially for development purpose) is to install the packages in a separate environment, e.g. using conda:

```
$ conda create -n topo_py36 python=3.6 xarray numba numpy pip -c conda-forge
$ source activate topo_py36
```

Then you can clone the `xarray-topo` git repository and install it using `pip` locally:

```
$ git clone https://gitext.gfz-potsdam.de/sec55-public/xarray-topo
$ cd xarray-topo
$ pip install .
```

For development purpose, use the following command:

```
$ pip install -e .
```

1.1.4 Import xarray-topo

To make sure that `xarray-topo` is correctly installed, try import it in a Python console:

```
>>> import xtopo
```

1.2 Examples

An example of using the landscape evolution models implemented in xarray-topo is shown the examples section of xarray-simlab's documentation.

Help & reference

- *What's New*
- *Contributor Guide*

1.3 What's New

No stable release yet.

1.4 Contributor Guide

1.4.1 Development environment

If you whish to contribute to the development of the code and/or the documentation, here are a few steps for setting a development environment.

Fork the repository and download the code

To further be able to submit modifications, it is preferable to start by forking the xarray-topo repository on GFZ GitLab (ext) (you need to have an account).

Then clone your fork locally:

```
$ git clone https://gitext.gfz-potsdam.de/your_name_here/xarray-topo
```

Alternatively, if you don't plan to submit any modification, you can clone the original xarray-topo git repository:

```
$ git clone https://gitext.gfz-potsdam.de/sec55-public/xarray-topo
```

Install

To install the dependencies, we recommend using the conda package manager with the conda-forge channel. For development purpose, you might consider installing the packages in a new conda environment:

```
$ conda create -n xarray-topo_dev python=3.6 numpy xarray -c conda-forge
$ source activate xarray-topo_dev
```

Then install xarray-topo locally using `pip`:

```
$ cd xarray-topo
$ pip install -e .
```

Run tests

Not yet available!

1.4.2 Contributing to code

Below are some useful pieces of information in case you want to contribute to the code.

Local development

Once you have setup the development environment, the next step is to create a new git branch for local development:

```
$ git checkout -b name-of-your-bugfix-or-feature
```

Now you can make your changes locally.

Submit changes

Once you are done with the changes, you can commit your changes to git and push your branch to your xarray-topo fork on GFZ GitLab:

```
$ git add .
$ git commit -m "Your detailed description of your changes."
$ git push origin name-of-your-bugfix-or-feature
```

(note: this operation may be repeated several times).

We you are ready, you can create a new merge request through the GFZ GitLab website (note that it is still possible to submit changes after your created a pull request).

Python versions

xarray-topo supports Python versions 3.4 and higher. It is not compatible with Python versions 2.x. We don't plan to make it compatible with Python 2.7.x unless there are very good reasons to do so.

Test

Not yet available!

Docstrings

Everything (i.e., classes, methods, functions…) that is part of the public API should follow the numpydoc standard when possible.

Coding style

The xarray-topo code mostly follows the style conventions defined in PEP8.

Source code checker

To check about any potential error or bad style in your code, you might want using a source chode checker like flake8. You can install it in your development environment:

```
$ conda install flake8 -c conda-forge
```

What's new entry

Every significative code contribution should be listed in the *What's New* section of this documentation under the corresponding version.

1.4.3 Contributing to documentation

xarray-topo uses Sphinx for documentation, hosted on http://readthedocs.org . Documentation is maintained in the RestructuredText markup language (`.rst` files) in `xarray-topo/doc`.

To build the documentation locally, first install requirements (for example here in a separate conda environment):

```
$ conda env create -n xarray-topo_doc -f doc/environment.yml
$ source activate xarray-topo_doc
```

Then build documentation with `make`:

```
$ cd doc
$ make html
```

The resulting HTML files end up in the `build/html` directory.

You can now make edits to rst files and run `make html` again to update the affected pages.

CHAPTER 2

License

3-clause ("Modified" or "New") BSD license.

xarray-topo is developped at the Earth Surface Process Modelling group of the GFZ Helmholtz Centre Potsdam.