eqstochsim Documentation

Release 0.0.1

James Holt

May 24, 2021

CONTENTS:

CHAPTER

ONE

EQSTOCHSIM

A package to showcase the stochastic method to compute earthquake ground motions.

- Free software: MIT license
- Documentation: https://eqstochsim.readthedocs.io.

1.1 Features

- TODO

1.2 Credits

This package was created with Cookiecutter and the audreyr/cookiecutter-pypackage project template.

CHAPTER

TWO

INSTALLATION

2.1 Stable release

To install eqstochsim, run this command in your terminal:

```
$ pip install eqstochsim
```

This is the preferred method to install eqstochsim, as it will always install the most recent stable release.

If you don't have pip installed, this Python installation guide can guide you through the process.

2.2 From sources

The sources for eqstochsim can be downloaded from the Github repo.

You can either clone the public repository:

```
$ git clone git://github.com/sgjholt/eqstochsim
```

Or download the tarball:

```
$ curl -OJL https://github.com/sgjholt/eqstochsim/tarball/master
```

Once you have a copy of the source, you can install it with:

```
$ python setup.py install
```

CHAPTER

THREE

USAGE

To use eqstochsim in a project:

```
import eqstochsim
```

CHAPTER

FOUR

EQSTOCHSIM

4.1 eqstochsim package

4.1.1 Submodules

4.1.2 eqstochsim.eqphysics module

Physical earthquake source parameters.

This file contains the functions that compute physical earthquake source parameters assuming .

This file can also be imported as a module and contains the following functions:

- fc
- mo_from_mw
- mw

eqstochsim.eqphysics.**fc**(*vs: float*, *sd: float*, *mo: float*, *c: float = 0.49*) → float

The formula for corner frequnecy assuming constant stress drop (Aki, 1967; Brune, 1970, 1971). The default assumes SI units for all parameters, if using cgs override c=0.49 to c=4.9E6 (e.g., Boore, 2003).

Parameters

- **vs** (*float*) – The shear-wave velocity at the rupture source (m [default] or km).
- **sd** (*float*) – The total stress drop () of the rupture (Pa [default] or Bar).
- **mo** (*float*) – The total scalar seismic moment (M0) release of the rupture (N m [default] or dyne-cm)
- **c** (*float*) – A scaling constant (assuming self-similarity) relating stress drop and corner frequency (SI [default] or cgs units).

Returns The theoretical corner frequency of a double-couple point source earthquake.

Return type float

eqstochsim.eqphysics.**mo_from_mw**(*mw: float*, *c: float = 6.0333*) → float

A formula for converting moment magnitude (Hanks and Kanamori, 1979) to seismic moment. The default assumes SI units for seismic moment, if using cgs override c=6.0333 to c=10.7 (Boore, 2003).

Parameters

- **mw** (*float*) – The moment magnitude.
- **c** (*float*) – A constant that maps log10-seismic moment to moment magnitude (SI [default] or cgs units).

Returns The seismic moment for a given Moment Magnitude.

Return type float

eqstochsim.eqphysics.**moment_scaling**(*vs: float = 3500.0*, *rho: float = 2600.0*, *ro: float = 1000.0*, *fs: float = 2.0*, *rp: float = 0.55*) → float

The displacement to moment scaling factor (log10 units) at the source. The constants for rp are taken

Parameters

- **vs** (*float*) – The shear-wave velocity at the rupture source (m [default] or km).
- **rho** (*float*) – The shear-wave density at the rupture source (kgm^-3 [default] or gcm^-3).
- **ro** (*float*) – The reference distance from the source (m[default]=1000 or km).
- **fs** (*float*) – Free surface amplification factor (2 for SH waves).
- **rp** (*float*) – The radiation partitioning coefficient (Boore and Boatwright) https://pubs.geoscienceworld.org/ssa/bssa/article/74/5/1615/118641/Average-body-wave-radiation-coefficients

Returns The scaling parameter from long period displacement to seismic moment.

Return type float

eqstochsim.eqphysics.**mw**(*mo: float*, *c: float = 6.0333*) → float

A formula for converting moment magnitude (Hanks and Kanamori, 1979) to seismic moment. The default assumes SI units for seismic moment, if using cgs override c=6.0333 to c=10.7 (Boore, 2003).

Parameters

- **mw** (*float*) – The moment magnitude.
- **c** (*float*) – A constant that maps log10-seismic moment to moment magnitude (SI [default] or cgs units).

Returns The moment magnitude for a given seismic moment.

Return type float

eqstochsim.eqphysics.**sd**(*fc: float*, *mo: float*, *vs: float*, *k: float = 0.37*) → float

The formula for stress drop assuming a circular crack model (Eshelby, 1957, Aki, 1967; Brune, 1970, 1971). k depends on wave type (P or S). Default value is for S-waves. The default assumes SI units for all parameters.

Parameters

- **fc** (*float*) – The radially averaged corner frequency of the source spectrum (Hz).
- **mo** (*float*) – The total scalar seismic moment (M0) release of the rupture (N m [default] or dyne-cm)
- **vs** (*float*) – The shear-wave velocity at the rupture source (m [default] or km).
- **k** (*float*) – A scaling constant that depends on wave type (P or S). S is the default value.

Returns The theoretical stress drop in MPa.

Return type float

4.1.3 eqstochsim.eqstochsim module

4.1.4 eqstochsim.filters module

Time domain filters for random noise.

This file contains the functions for filtering the random noise of the stochastic models.

This file can also be imported as a module and contains the following functions:

eqstochsim.filters.**exp_filt**()

4.1.5 eqstochsim.models module

Time domain filters for random noise.

This file contains the functions of the constituant models that are combined to compute the stochastic ground motion model for arbitrary earthquake scenarios.

This file can also be imported as a module and contains the following functions:

motion_factor The scaling of ground motion to displacement, velocity or acceleration.

source_scf A generic single corner frequency model for a seismic source.

eqstochsim.models.**f_dep_attenuation**(*f: numpy.ndarray*, *a: float*, *Q: float*, *R: float*, *b: float*) → numpy.ndarray

Frequency dependent attenuation model (log base 10 representation).

Parameters

- **f** (*np.ndarray*) – The frequencies to compute attenuation over [Hz].
- **a** (*float*) – The frequency dependent factor for attenuation.
- **Q** (*float*) – The frequency independent quality factor.
- **R** (*float*) – The propagation distance [km or m].
- **b** (*float*) – Average seismic velocity along the path [km or m].

eqstochsim.models.**f_idep_attenutation**(*f: numpy.ndarray*, *Q: float*, *R: float*, *b: float*) → numpy.ndarray

Frequency independent attenuation model (log base 10 representation).

Parameters

- **f** (*np.ndarray*) – The frequencies to compute the attenuation for.
- **Q** (*float*) – The frequency independent quality factor.
- **R** (*float*) – The propagation distance in km or m.
- **b** (*float*) – Average seismic velocity along the path in km of m.

eqstochsim.models.**motion_factor**(*f: numpy.ndarray*, *motion: str = 'disp'*) → Optional[numpy.ndarray]

A function to scale the source spectrum to the desired motion parameter. Add to differentiate and subtract to integrate.

eqstochsim.models.**single_geospreading**(*R: Union[float, numpy.ndarray]*, *p=1*)

eqstochsim.models.**source_scf**(*f: numpy.ndarray*, *llpsp: float*, *fc: float*, *gam: float*, *n: float*) → numpy.ndarray

Generic single corner frequency model for the far-field displacement spectrum of an arbitrary seismic source as a function of frequency (log base 10 representation).

Parameters **llpsp** – Log10 amplitude of the long period plateau (e.g. log10[M0])

4.1.6 Module contents

Top-level package for eqstochsim.

CHAPTER

FIVE

CONTRIBUTING

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

You can contribute in many ways:

5.1 Types of Contributions

5.1.1 Report Bugs

Report bugs at https://github.com/sgjholt/eqstochsim/issues.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

5.1.2 Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with "bug" and "help wanted" is open to whoever wants to implement it.

5.1.3 Implement Features

Look through the GitHub issues for features. Anything tagged with "enhancement" and "help wanted" is open to whoever wants to implement it.

5.1.4 Write Documentation

eqstochsim could always use more documentation, whether as part of the official eqstochsim docs, in docstrings, or even on the web in blog posts, articles, and such.

5.1.5 Submit Feedback

The best way to send feedback is to file an issue at https://github.com/sgjholt/eqstochsim/issues.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

5.2 Get Started!

Ready to contribute? Here's how to set up *eqstochsim* for local development.

1. Fork the *eqstochsim* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/eqstochsim.git
```

3. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your fork for local development:

```
$ mkvirtualenv eqstochsim
$ cd eqstochsim/
$ python setup.py develop
```

4. Create a branch for local development:

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

   Now you can make your changes locally.

5. When you're done making changes, check that your changes pass flake8 and the tests, including testing other Python versions with tox:

```
$ flake8 eqstochsim tests
$ python setup.py test or pytest
$ tox
```

   To get flake8 and tox, just pip install them into your virtualenv.

6. Commit your changes and push your branch to GitHub:

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

7. Submit a pull request through the GitHub website.

5.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.
3. The pull request should work for Python 3.5, 3.6, 3.7 and 3.8, and for PyPy. Check https://travis-ci.com/sgjholt/eqstochsim/pull_requests and make sure that the tests pass for all supported Python versions.

5.4 Tips

To run a subset of tests:

```
$ pytest tests.test_eqstochsim
```

5.5 Deploying

A reminder for the maintainers on how to deploy. Make sure all your changes are committed (including an entry in HISTORY.rst). Then run:

```
$ bump2version patch # possible: major / minor / patch
$ git push
$ git push --tags
```

Travis will then deploy to PyPI if tests pass.

CHAPTER

SIX

CREDITS

6.1 Development Lead

- James Holt

6.2 Contributors

None yet. Why not be the first?

CHAPTER

SEVEN

HISTORY

7.1 0.0.1 (2021-02-05)

- First release on PyPI.

CHAPTER

EIGHT

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

e

INDEX