RESSPyLab Documentation

Release 0.1.3

Albano de Castro e Sousa

Sep 02, 2017

Contents

Contents:

CHAPTER 1

RESSPyLab

Welcome to the Resilient Steel Structures Laboratory (RESSLab) Python Library.

The RESSLab is a research laboratory at École Polytechnique Fédérale de Lausanne (EPFL).

The library is currently under testing phase.

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

Features

- Implicit integration scheme for non-linear uniaxial Voce and Chaboche metal plasticity
- Newton Trust-Region (NTR) with Singular Value Decomposition (SVD) preconditioning solver
- Voce and Chaboche material parameter estimation with NTR-SVD

Credits

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

CHAPTER 2

Installation

Stable release

To install RESSPyLab, run this command in your terminal:

```
$ pip install RESSPyLab
```

This is the preferred method to install RESSPyLab, 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.

From sources

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

You can either clone the public repository:

```
$ git clone git://github.com/AlbanoCastroSousa/RESSPyLab
```

Or download the tarball:

```
$ curl  -OL https://github.com/AlbanoCastroSousa/RESSPyLab/tarball/master
```

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

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

CHAPTER 3

Usage

To use RESSPyLab in a project:

```
import RESSPyLab
```

The current version includes primarily functions to perform material parameter calibration of the Voce and Chaboche (VC) metal plasticity model.

There are three main useful functions in the library

```
def NTR_SVD_Solver(f,gradF,Hf,x_0):
        # f - function taking an array "x" of n floats as an argument and returning␣
↪the real value of f
        # gradF - function taking an array "x" of n floats as an argument and␣
↪returning the gradient of f, an array sized n
        # Hf - function taking an array "x" of n floats as an argument and returning␣
↪Hessian of f, array sized n by n
        # x_0 - initial starting point; array sized n
        # x_min - local minimum of f; array sized n
        #...
        return x_min

def VCcurve(x,test):
        # x - set of parameters for the Voce and Chaboche material model
        # The order in x should be the following:
        # x=[E, sy0, Qinf, b, C_1, gamma_1, C_2, gamma_2, ..., ..., C_k, gamma_k]
        #
        # test is a pandas dataframe with columns named 'e_true', for the true strain␣
↪and 'Sigma_true', for the true stress
        # simCurve is a pandas dataframe containing the simulated curve with the VC␣
↪model. Integration is conducted with the discretization in "test"
        #...
        return simCurve

def VCopt(x_0,listTests):
        # This function performs parameter optimization of the VC model for an␣
↪ensemble of experimental data.
```

```
        #
        # listTests is a python list containg pandas dataframes of multiple "test"
        return x_min
```

A working example using a Jupyter notebook, along its data files can be found on gitHub.

CHAPTER 4

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:

Types of Contributions

Report Bugs

Report bugs at https://github.com/AlbanoCastroSousa/RESSPyLab/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.

Fix Bugs

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

Implement Features

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

Write Documentation

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

Submit Feedback

The best way to send feedback is to file an issue at https://github.com/AlbanoCastroSousa/RESSPyLab/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 :)

Get Started!

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

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

   ```
   $ git clone git@github.com:your_name_here/RESSPyLab.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 RESSPyLab
   $ cd RESSPyLab/
   $ 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 RESSPyLab tests
   $ python setup.py test or py.test
   $ 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.

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 2.6, 2.7, 3.3, 3.4 and 3.5, and for PyPy. Check https://travis-ci.org/AlbanoCastroSousa/RESSPyLab/pull_requests and make sure that the tests pass for all supported Python versions.

Tips

To run a subset of tests:

```
$ python -m unittest tests.test_RESSPyLab
```

CHAPTER 5

Credits

Development Lead

- Albano de Castro e Sousa <albano.sousa@epfl.ch>

Contributors

None yet. Why not be the first?

CHAPTER 6

History

0.1.0 (2017-09-01)

- First release on PyPI.

CHAPTER 7

Indices and tables

- genindex
- modindex
- search