
CLT Calculation for Python Documentation

Release 0.0.5

Axel Schmidt

Jan 28, 2020

Contents:

1	CLT Calculation for Python	1
1.1	Features	1
1.2	Credits	1
2	Installation	3
2.1	Stable release	3
2.2	From sources	3
3	Usage	5
3.1	Setting up materials	5
3.2	Set up plies/layers	5
3.3	Arrange plies to laminate	6
3.4	Output material data	6
4	clt_py	7
4.1	clt_py package	7
5	Contributing	9
5.1	Types of Contributions	9
5.2	Get Started!	10
5.3	Pull Request Guidelines	11
5.4	Tips	11
5.5	Deploying	11
6	Credits	13
6.1	Development Lead	13
6.2	Contributors	13
7	History	15
7.1	Unreleased	15
7.2	0.0.5 (2020-01-24)	15
7.3	0.0.4 (2020-01-24)	15
7.4	0.0.3 (2020-01-24)	16
7.5	0.0.2 (2020-01-22)	16
8	Indices and tables	17
	Python Module Index	19

CLT Calculation for Python

This packages provides methodes for calculating fiber-reinforced materials.

- Free software: MIT license
- Documentation: <https://clt-py.readthedocs.io>.

1.1 Features

- TODO

1.2 Credits

This package was created with [Cookiecutter](#) and the [audreyr/cookiecutter-pypackage](#) project template.

2.1 Stable release

To install CLT Calculation for Python, run this command in your terminal:

```
$ pip install clt_py
```

This is the preferred method to install CLT Calculation for Python, 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 CLT Calculation for Python can be downloaded from the [Github repo](#).

You can either clone the public repository:

```
$ git clone git://github.com/axelschmidt91/clt_py
```

Or download the [tarball](#):

```
$ curl -OJL https://github.com/axelschmidt91/clt_py/tarball/master
```

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

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


To use CLT Calculation for Python in a project:

```
import clt_py
```

or:

```
from clt_py import *
```

or any other format you prefer.

3.1 Setting up materials

```
# Matrix material:
matMat = IsotropicMaterial(rho=1, E=1, v=0.25)
# Fiber material:
matFib = AnisotropicMaterial(rho=2, v_para_ortho=0.25, E_para=10, E_ortho=2, G=3)

# Composite Material:
mat_FRM = FiberReinforcedMaterialUD(matFib=matFib, matMat=matMat)
```

3.2 Set up plies/layers

```
ply0 = Ply(mat_FRM, rotation=0, thickness=1)
plyP45 = Ply(mat_FRM, rotation=45, thickness=1)
plyN45 = Ply(mat_FRM, rotation=-45, thickness=1)
```

3.3 Arrange plies to laminate

```
laminate = Laminate()

laminate.addPly(plyP45)
laminate.addPly(ply0)
laminate.addPly(plyN45)
```

3.4 Output material data

Calculation of complete stiffness matrix (ABD-matrix) **[#CLT]**.

```
np.set_printoptions(precision=3, suppress=True)
print(laminate.get_stiffnessMatrix())
```

4.1 clt_py package

Top-level package for CLT Calculation for Python.

4.1.1 Submodules

4.1.2 clt_py.clt_py module

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/axelschmidt91/clt_py/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

CLT Calculation for Python could always use more documentation, whether as part of the official CLT Calculation for Python 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/axelschmidt91/clt_py/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 *clt_py* for local development.

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

```
$ git clone git@github.com:your_name_here/clt_py.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 clt_py
$ cd clt_py/
$ 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 clt_py 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.org/axelschmidt91/clt_py/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_clt_py
```

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.

6.1 Development Lead

- Axel Schmidt <schmidt.axel.91@gmail.com>

6.2 Contributors

None yet. Why not be the first?

All notable changes to this project will be documented in this file.

The format is based on [Keep a Changelog](<https://keepachangelog.com/en/1.0.0/>).

7.1 Unreleased

7.2 0.0.5 (2020-01-24)

- **changes:**
 - version number
 - long_description_content_type in setup.py
 - travis API token fixed

7.3 0.0.4 (2020-01-24)

- **changed:**
 - requirements_dev.txt to updated packades
 - requirements.txt
- **fixed:**
 - upload to pypi with API Token

7.4 0.0.3 (2020-01-24)

- First release on PyPI.
- add class Material2D - base class for all materials used in this package.
- add class IsotropicMaterial - Material2D class for isotropic materials.
- add class AnisotropicMaterial - Material2D class for anisotropic materials.
- add class FiberReinforcedMaterialUD - Material2D class that combines two materials.
- add class Ply - generates defined ply/layer of Material2D.
- add class Laminate - combines multiple Ply instances to composite laminate.
- **dev setup:**
 - add .pre-commit-config.yaml - formatting with black
 - changed travis - add `black --check` on 'before_script' method
 - changed requirements_dev.txt
 - add requirements.txt
 - add tests for classes
 - fix tox python 3.8 and flake8 checking
 - add flake8 ignore: E501, W503, F841
- update mail address
- **docs:**
 - changed usage.rst - new Version update to use package

7.5 0.0.2 (2020-01-22)

- setup of package
- first ideas of structure

CHAPTER 8

Indices and tables

- `genindex`
- `modindex`
- `search`

C

`clt_py`, [7](#)

C

`clt_py` (*module*), [7](#)