
mechmat Documentation

Release 0.1.0

Jelle Spijker

Apr 21, 2020

CONTENTS:

1	mechmat	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
4	mechmat	7
4.1	mechmat package	7
5	Contributing	9
5.1	Types of Contributions	9
5.2	Get Started!	10
5.3	Pull Request Guidelines	10
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	0.1.0 (2019-03-29)	15
7.2	0.1.4 (2019-05-11)	15
8	Indices and tables	17
	Python Module Index	19
	Index	21

MECHMAT

Python package for the definition of materials used during mechanical engineering calculations

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

1.1 Features

- TODO

1.2 Credits

This package makes use of [pint](#) for unit safe calculations.

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

INSTALLATION

2.1 Stable release

To install mechat, run this command in your terminal:

```
$ pip install mechat
```

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

You can either clone the public repository:

```
$ git clone git://github.com/jellespijker/mechat
```

Or download the [tarball](#):

```
$ curl -OL https://github.com/jellespijker/mechat/tarball/master
```

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

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


USAGE

To use mechmat in a project:

```
import mechmat
```


4.1 mechmat package

4.1.1 Submodules

4.1.2 mechmat.linked module

```
class mechmat.linked.Linked (unit='dimensionless', rng=[-inf, inf], linked_properties={})  
    Bases: object  
  
    static argument_weight (visited, arg)  
  
    static get_sorted_functions (visited, args)  
  
    static in_range (value, rng)  
  
class mechmat.linked.MetaLinked (name, bases, attr_dict)  
    Bases: type
```

4.1.3 mechmat.material module

```
class mechmat.material.Material (**kwargs)  
    Bases: object  
  
    """ All materials inherit from this class. This class describes the basic properties, which all mater have, such as a  
    density, specific weight, and temperature.  
  
    CAS = None  
        str: Chemical Abstracts Service number  
  
    category = None  
        Category The Material category  
  
    density = None  
        Linked Density  $\rho$  in  $[M^1 L^{-3}]$   
  
    dump ()  
        """ Returns a YAML dump of the material  
  
    load (data)  
        """ Restores a YAML dump on the material  
  
    name = None  
        str: The common name of the material
```

short_name

str: Short name for the material. When it is not user specified, the *name* is used. When this consists of multiple words, the short name is build from all first letters. When the name consist of a single word, the first two letters are used

specific_heat_at_const_pressure = None

Linked Specific heat at constant pressure c_p in $[L^2T^{-1}t^{-2}]$

specific_volume = None

Linked Specific volume v in $[L^3M^{-1}]$

specific_weight = None

Linked Specific weight γ in $[M^1L^{-2}t^{-2}]$

temperature = None

Linked Temperature T in $[T]$

thermal_conductivity = None

Linked Thermal conductivity k in $[L^1M^1T^{-1}t^{-3}]$

thermal_diffusivity = None

Linked Thermal diffusivity α in $[L^2t^{-1}]$

4.1.4 mechmat.subject module

class mechmat.subject.**Subject** (*prop*, **observers*)

Bases: object

notify (*instance*, *key*)

register (**observers*)

send (*instance*, *key*)

4.1.5 Module contents

Top-level package for mechmat.

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://gitlab.com/pymech/mechmat/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 Gitlab 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 Gitlab issues for features. Anything tagged with “enhancement” and “help wanted” is open to whoever wants to implement it.

5.1.4 Write Documentation

mechmat could always use more documentation, whether as part of the official mechmat 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://gitlab.com/pymech/mechmat/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 *mechmat* for local development.

1. Fork the *mechmat* repo on Gitlab.
2. Clone your fork locally:

```
$ git clone git@gitlab.com:your_name_here/mechmat.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 mechmat
$ cd mechmat/
$ 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 mechmat 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 Gitlab:

```
$ 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 Gitlab 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 2.7, 3.4, 3.5 and 3.6, and for PyPy. Check https://travis-ci.org/jellespijker/mechmat/pull_requests and make sure that the tests pass for all supported Python versions.

5.4 Tips

To run a subset of tests:

```
$ py.test tests.test_mechmat
```

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:

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

Travis will then deploy to PyPI if tests pass.

CREDITS

6.1 Development Lead

- Jelle Spijker <spijker.jelle@gmail.com>

6.2 Contributors

None yet. Why not be the first?

HISTORY

7.1 0.1.0 (2019-03-29)

- First release on PyPI.

7.2 0.1.4 (2019-05-11)

- Multiple bug fixes
- Accepts Numpy arrays
- State factor for easy creation of material states
- State can now be set when initializing
- Expanded the base material properties
- Added support for Jupyter Markdown, LaTeX and HTML representation

INDICES AND TABLES

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

PYTHON MODULE INDEX

m

- `mechmat`, 8
- `mechmat.linked`, 7
- `mechmat.material`, 7
- `mechmat.subject`, 8

A

`argument_weight()` (*mechmat.linked.Linked static method*), 7

C

`CAS` (*mechmat.material.Material attribute*), 7
`category` (*mechmat.material.Material attribute*), 7

D

`density` (*mechmat.material.Material attribute*), 7
`dump()` (*mechmat.material.Material method*), 7

G

`get_sorted_functions()` (*mechmat.linked.Linked static method*), 7

I

`in_range()` (*mechmat.linked.Linked static method*), 7

L

`Linked` (*class in mechmat.linked*), 7
`load()` (*mechmat.material.Material method*), 7

M

`Material` (*class in mechmat.material*), 7
`mechmat` (*module*), 8
`mechmat.linked` (*module*), 7
`mechmat.material` (*module*), 7
`mechmat.subject` (*module*), 8
`MetaLinked` (*class in mechmat.linked*), 7

N

`name` (*mechmat.material.Material attribute*), 7
`notify()` (*mechmat.subject.Subject method*), 8

R

`register()` (*mechmat.subject.Subject method*), 8

S

`send()` (*mechmat.subject.Subject method*), 8
`short_name` (*mechmat.material.Material attribute*), 7

`specific_heat_at_const_pressure` (*mechmat.material.Material attribute*), 8
`specific_volume` (*mechmat.material.Material attribute*), 8
`specific_weight` (*mechmat.material.Material attribute*), 8
`Subject` (*class in mechmat.subject*), 8

T

`temperature` (*mechmat.material.Material attribute*), 8
`thermal_conductivity` (*mechmat.material.Material attribute*), 8
`thermal_diffusivity` (*mechmat.material.Material attribute*), 8