

---

# **mechmat Documentation**

***Release 0.2.4***

**Jelle Spijker**

**Apr 21, 2020**



## CONTENTS:

<b>1</b>	<b>mechmat</b>	<b>1</b>
1.1	Features . . . . .	1
1.2	Credits . . . . .	1
<b>2</b>	<b>Installation</b>	<b>3</b>
2.1	Stable release . . . . .	3
2.2	From sources . . . . .	3
<b>3</b>	<b>Usage</b>	<b>5</b>
<b>4</b>	<b>mechmat</b>	<b>7</b>
4.1	mechmat package . . . . .	7
<b>5</b>	<b>Contributing</b>	<b>19</b>
5.1	Types of Contributions . . . . .	19
5.2	Get Started! . . . . .	20
5.3	Pull Request Guidelines . . . . .	20
5.4	Tips . . . . .	21
5.5	Deploying . . . . .	21
<b>6</b>	<b>Credits</b>	<b>23</b>
6.1	Development Lead . . . . .	23
6.2	Contributors . . . . .	23
<b>7</b>	<b>History</b>	<b>25</b>
7.1	0.1.0 (2019-03-29) . . . . .	25
7.2	0.1.4 (2019-05-11) . . . . .	25
7.3	0.2.0 (2019-05-25) . . . . .	25
7.4	0.2.1 (2019-05-25) . . . . .	25
7.5	0.2.2 (2019-05-31) . . . . .	26
7.6	0.2.3 (2019-05-31) . . . . .	26
7.7	0.2.4 (2019-07-09) . . . . .	26
<b>8</b>	<b>Indices and tables</b>	<b>27</b>
	<b>Python Module Index</b>	<b>29</b>
	<b>Index</b>	<b>31</b>



## 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 Subpackages

**mechmat.core package**

**Submodules**

**mechmat.core.chainable module**

```
class mechmat.core.chainable.Chainable(**kwargs)
```

Bases: object

” A linked attribute class

```
link_attr (attr, transform, **kwargs)
```

Link a Linked attribute against another Linked attribute.

**Args:** attr (str): Attribute name transform: The function which provides the transform. **\*\*kwargs:** the transform function keywords where the value is either a str (if the attribute can be obtained from the own instance) or a tuple containing the other instance and attribute name.

```
linked_transforms (attr)
```

```
set_guard (attr, unit=None, rng=None, doc=None)
```

Set the guard descriptor unit and range, this is usually set in the `__init__()` function

**Args:** attr (str): The guard attribute to be set unit (ureg.Unit): The unit in which guarded inputs are to be converted rng (tuple, list, np.array): The range [low, high] against which to test doc (str): dosctring

```
unlink_attr (attr, transform)
```

```
class mechmat.core.chainable.Guarded
```

Bases: object

Descriptor guarding Linked attributes

```
static cite_value (value)
```

```
static in_range (value, rng)
```

is value in specified range

**Args:** value: value to be tested rng: tuple or list to be tested against

**Returns:** true when in range otherwise false

**class** mechmat.core.chainable.Message  
Bases: set  
Linked Message

## mechmat.core.errors module

**exception** mechmat.core.errors.OutOfRangeError (*value, rng, property*)  
Bases: ValueError  
Raised when trying to set an out-of-range value

## Module contents

### mechmat.principal package

#### Submodules

#### mechmat.principal.core module

mechmat.principal.core.reciprocal (*value*)  
mechmat.principal.core.sub (*\*\*kwargs*)  
mechmat.principal.core.add (*\*\*kwargs*)  
mechmat.principal.core.mul (*\*\*kwargs*)  
mechmat.principal.core.div (*\*\*kwargs*)  
**class** mechmat.principal.core.Interp (*kind='cubic', cite=None, \*\*kwargs*)  
Bases: object

#### mechmat.principal.crossarrhenius module

mechmat.principal.crossarrhenius.arrhenius\_shift (*temperature, arrhe-  
nius\_activation\_energy, tempera-  
ture\_ref*)  
mechmat.principal.crossarrhenius.relaxation\_time (*relaxation\_time\_ref, arrhenius*)  
mechmat.principal.crossarrhenius.viscosity\_dynamic (*shear\_rate, zero\_shear\_viscosity,  
relaxation\_time,  
shear\_thinning\_const*)  
mechmat.principal.crossarrhenius.zero\_shear\_viscosity (*arrhenius,  
zero\_shear\_viscosity\_ref*)

#### mechmat.principal.density module

mechmat.principal.density.from\_specific\_weight (*specific\_weight*)  
Args: specific\_weight:  
Returns:

## mechmat.principal.geometry module

`mechmat.principal.geometry.distance` (*point\_1*, *point\_2*)

Returns the distance between two points.

**Args:** *point\_1*: Scalar or vector of point 1 *point\_2*: Scalar or vector of point 2

**Returns:** Scalar of the distance between *point\_2* and *point\_1*

`mechmat.principal.geometry.halfway` (*point\_1*, *point\_2*)

## mechmat.principal.shear\_rate module

`mechmat.principal.shear_rate.circle` (*V\_dot*, *r*)

” The apparent shear rate for a melt flowing through a circle is defined as

$$\dot{\gamma}_a = \frac{4\dot{V}}{\pi R^3}$$

Source: Rao, Natti S. Basic Polymer Engineering Data. Cincinnati, Ohio, USA: Hanser, 2017.

**Args:** *V\_dot*: Volumetric\_flow in  $[L^3 t^{-1}]$  *r*: Radius in  $[L^1]$

**Returns:** Apparent shear rate in  $[t^{-1}]$

`mechmat.principal.shear_rate.annulus` (*V\_dot*, *r\_i*, *r\_o*)

The apparent shear rate for a melt flowing through an annulus is defined as

$$\frac{6\dot{V}}{\pi (r_o + r_i) (r_o - r_i)^2}$$

Source: Rao, Natti S. Basic Polymer Engineering Data. Cincinnati, Ohio, USA: Hanser, 2017.

**Args:** *V\_dot*: Volumetric\_flow in  $[L^3 t^{-1}]$  *r\_i*: inner radius in  $[L^1]$  *r\_o*: inner radius in  $[L^1]$

**Returns:** Apparent shear rate in  $[t^{-1}]$

## mechmat.principal.specific\_weight module

`mechmat.principal.specific_weight.from_density` (*density*)

**Args:** *density*:

Returns:

## mechmat.principal.thermal module

`mechmat.principal.thermal.specific_heat_capacity` (*thermal\_conductivity*, *density*, *thermal\_diffusivity*)

`mechmat.principal.thermal.thermal_conductivity` (*thermal\_diffusivity*, *specific\_heat\_capacity*, *density*) *spe-*

`mechmat.principal.thermal.thermal_diffusivity` (*thermal\_conductivity*, *specific\_heat\_capacity*, *density*) *spe-*

The rate of transfer of heat of a material from the hot end to the cold end.

**Args:** *thermal\_conductivity*: *k* *specific\_heat\_capacity*: *c<sub>p</sub>* *density*: *ρ*

Returns:

## mechmat.principal.twodomaintaitpvt module

```
mechmat.principal.twodomaintaitpvt.get_specific_volume(p, v_0, v_t, B)
mechmat.principal.twodomaintaitpvt.get_B(T, b_3, b_4, b_5)
mechmat.principal.twodomaintaitpvt.switch_m_s(T, T_t, s, m)
mechmat.principal.twodomaintaitpvt.get_T_t(p, b_5, b_6)
mechmat.principal.twodomaintaitpvt.get_v_0(T, b_1, b_2, b_5)
mechmat.principal.twodomaintaitpvt.get_v_t(p, T, T_t, b_5, b_7, b_8, b_9)
```

## Module contents

### mechmat.properties package

#### Subpackages

#### mechmat.properties.elastic\_deformation package

## Module contents

### mechmat.properties.flow package

#### Submodules

#### mechmat.properties.flow.flow module

```
class mechmat.properties.flow.flow.Flow(**kwargs)
    Bases: mechmat.properties.flow.mass_flow.MassFlow, mechmat.properties.flow.
    volume_flow.VolumeFlow
```

#### mechmat.properties.flow.mass\_flow module

```
class mechmat.properties.flow.mass_flow.MassFlow(**kwargs)
    Bases: mechmat.core.chainable.Chainable

    density
        Descriptor guarding Linked attributes

    massflow
        Descriptor guarding Linked attributes

    specific_volume
        Descriptor guarding Linked attributes

    specific_weight
        Descriptor guarding Linked attributes
```

## mechmat.properties.flow.volume\_flow module

```
class mechmat.properties.flow.volume_flow.VolumeFlow (**kwargs)
    Bases:      mechmat.properties.geometry.surface.Surface,      mechmat.properties.
               geometry.vector.Segment

    volume_flow
        Descriptor guarding Linked attributes
```

## Module contents

## mechmat.properties.geometry package

## Submodules

## mechmat.properties.geometry.geometry module

```
class mechmat.properties.geometry.geometry.Geometry (**kwargs)
    Bases:      mechmat.properties.geometry.vector.Vector,      mechmat.properties.
               geometry.vector.Segment,      mechmat.properties.geometry.surface.Surface,
               mechmat.properties.geometry.volume.Volume
```

## mechmat.properties.geometry.surface module

```
class mechmat.properties.geometry.surface.Surface (**kwargs)
    Bases: mechmat.core.chainable.Chainable

    cross_section
        Descriptor guarding Linked attributes
```

## mechmat.properties.geometry.vector module

```
class mechmat.properties.geometry.vector.Segment (**kwargs)
    Bases: mechmat.core.chainable.Chainable

    distance
        Descriptor guarding Linked attributes

    point_1
        Descriptor guarding Linked attributes

    point_2
        Descriptor guarding Linked attributes

class mechmat.properties.geometry.vector.Vector (**kwargs)
    Bases: mechmat.core.chainable.Chainable

    coordinate
        Descriptor guarding Linked attributes
```

## mechmat.properties.geometry.volume module

```
class mechmat.properties.geometry.volume.Volume (**kwargs)
    Bases: mechmat.core.chainable.Chainable

    volume
        Descriptor guarding Linked attributes
```

## Module contents

### mechmat.properties.mass package

#### Submodules

## mechmat.properties.mass.mass module

```
class mechmat.properties.mass.mass.Mass (**kwargs)
    Bases: mechmat.core.chainable.Chainable

    density
        Descriptor guarding Linked attributes

    mass
        Descriptor guarding Linked attributes

    specific_volume
        Descriptor guarding Linked attributes

    specific_weight
        Descriptor guarding Linked attributes
```

## Module contents

### mechmat.properties.plastic\_deformation package

#### Module contents

### mechmat.properties.pressure package

#### Submodules

## mechmat.properties.pressure.pressure module

```
class mechmat.properties.pressure.pressure.Pressure (**kwargs)
    Bases: mechmat.core.chainable.Chainable

    pressure
        Descriptor guarding Linked attributes
```



## Module contents

### mechmat.properties.shearing package

#### Submodules

#### mechmat.properties.shearing.shearing module

**class** `mechmat.properties.shearing.shearing.Shearing(**kwargs)`

Bases: `mechmat.core.chainable.Chainable`

**shear\_rate**

Descriptor guarding Linked attributes

## Module contents

### mechmat.properties.specific\_volume package

#### Submodules

#### mechmat.properties.specific\_volume.twodomaintaitpvt module

**class** `mechmat.properties.specific_volume.twodomaintaitpvt.TwoDomainTaitpvt(**kwargs)`

Bases: `mechmat.core.chainable.Chainable`

The modified 2-domain Tait pvT model is used to determine the density of the material as a function of the temperature and pressure. This variation impacts on many aspects of the flow simulation.

The 2-domain Tait pvT model is given by the following equations:

$$v(T, p) = v_0(T) \left[ 1 - C \ln \left( 1 + \frac{p}{B(T)} \right) \right] + v_t(T, p)$$

where:

- $v(T, p)$  is the specific geometry at temperature and pressure
- $v_0$  is the specific geometry at zero gauge pressure
- $T$  is the temperature
- $p$  is the pressure
- $C$  is a constant
- $B$  accounts for the pressure sensitivity of the material

**The input for fully specified state:**

- `b_1s`
- `b_1m`
- `b_2s`
- `b_2m`
- `b_3s`

- b\_3m
- b\_4s
- b\_4m
- b\_5
- b\_6
- b\_7
- b\_8
- b\_9
- temperature
- pressure

**b\_1m**

Descriptor guarding Linked attributes

**b\_1s**

Descriptor guarding Linked attributes

**b\_2m**

Descriptor guarding Linked attributes

**b\_2s**

Descriptor guarding Linked attributes

**b\_3m**

Descriptor guarding Linked attributes

**b\_3s**

Descriptor guarding Linked attributes

**b\_4m**

Descriptor guarding Linked attributes

**b\_4s**

Descriptor guarding Linked attributes

**b\_5**

Descriptor guarding Linked attributes

**b\_6**

Descriptor guarding Linked attributes

**b\_7**

Descriptor guarding Linked attributes

**b\_8**

Descriptor guarding Linked attributes

**b\_9**

Descriptor guarding Linked attributes

**specific\_volume\_transition\_temperature**

Descriptor guarding Linked attributes

**specific\_volume\_zero\_gauge\_pressure**

Descriptor guarding Linked attributes

**temperature\_transition**  
Descriptor guarding Linked attributes

## Module contents

### mechmat.properties.thermal package

#### Submodules

#### mechmat.properties.thermal.conductivity module

**class** `mechmat.properties.thermal.conductivity.ThermalConductivity` (\*\*kwargs)  
Bases: `mechmat.core.chainable.Chainable`

**heat\_transfer\_coeff**  
Descriptor guarding Linked attributes

**thermal\_conductance**  
Descriptor guarding Linked attributes

**thermal\_conductivity**  
Descriptor guarding Linked attributes

**thermal\_diffusivity**  
Descriptor guarding Linked attributes

**thermal\_insulance**  
Descriptor guarding Linked attributes

**thermal\_resistance**  
Descriptor guarding Linked attributes

**thermal\_resistivity**  
Descriptor guarding Linked attributes

**thermal\_transmittance\_convection**  
Descriptor guarding Linked attributes

**thermal\_transmittance\_radiation**  
Descriptor guarding Linked attributes

#### mechmat.properties.thermal.thermal module

**class** `mechmat.properties.thermal.thermal.Thermal` (\*\*kwargs)  
Bases: `mechmat.core.chainable.Chainable`

**specific\_heat\_capacity**  
Descriptor guarding Linked attributes

**temperature**  
Temperature of a material

**temperature\_melt**  
Descriptor guarding Linked attributes

**temperature\_vapor**  
Descriptor guarding Linked attributes

**thermal\_conductivity**

Descriptor guarding Linked attributes

**thermal\_diffusivity**

Descriptor guarding Linked attributes

**thermal\_expansion\_coeff**

Descriptor guarding Linked attributes

## Module contents

### mechmat.properties.viscosity package

#### Submodules

#### mechmat.properties.viscosity.crossarrhenius module

**class** `mechmat.properties.viscosity.crossarrhenius.CrossArrhenius` (*\*\*kwargs*)

Bases: `mechmat.core.chainable.Chainable`

The model is based on the assumption that the fluid flow obeys the Arrhenius equation for molecular kinetics.

$$\eta(T, \dot{\gamma}) = \frac{\eta_0(T)}{1 + (\lambda(T)\dot{\gamma})^a}$$

where:

- $\eta_0(T_{\text{ref}})$  zero shear rate viscosity at reference temperature
- $\lambda(T_{\text{ref}})$  “relaxation time” at reference temperature
- $a$  “shear-thinning” constant
- $E_a$  Arrhenius activation energy
- $R$  gas constant
- $T$  temperature
- $T_{\text{ref}}$  reference temperature

**The input for fully specified state:**

- temperature
- temperature\_cross\_arrhenius\_ref
- arrhenius\_activation\_energy
- relaxation\_time\_ref
- shear\_rate
- shear\_thinning\_const
- viscosity\_zero\_shear\_rate\_ref

**arrhenius\_activation\_energy**

Descriptor guarding Linked attributes

**relaxation\_time**

Descriptor guarding Linked attributes

**relaxation\_time\_ref**  
Descriptor guarding Linked attributes

**shear\_thinning\_const**  
Descriptor guarding Linked attributes

**temperature\_cross\_arrhenius\_ref**  
Descriptor guarding Linked attributes

**viscosity\_zero\_shear\_rate**  
Descriptor guarding Linked attributes

**viscosity\_zero\_shear\_rate\_ref**  
Descriptor guarding Linked attributes

## Module contents

## Module contents

### 4.1.2 Submodules

#### 4.1.3 mechmat.material module

`mechmat.material.material_factory(*args, flow=False, **kwargs)`

Material instance facotry

**Args:** *\*args*: Chainable sub-propperties flow: is the material a continium flowing  $\frac{dm}{dt}$  *\*\*kwargs*:

Returns:

#### 4.1.4 mechmat.polymer module

```
class mechmat.polymer.PolyLacticAcid(**kwargs)
    Bases: mechmat.properties.thermal.polymer.ThermalPolymer, mechmat.
    properties.viscosity.crosswlf.CrossWLF, mechmat.properties.specific_volume.
    twodomaintaitpvt.TwoDomainTaitpvT

class mechmat.polymer.PolyLacticAcidThermoplasticPolyUrethane(**kwargs)
    Bases: mechmat.properties.thermal.polymer.ThermalPolymer, mechmat.
    properties.viscosity.crosswlf.CrossWLF, mechmat.properties.specific_volume.
    twodomaintaitpvt.TwoDomainTaitpvT

class mechmat.polymer.PolyLacticAcidThermoplasticStarch(**kwargs)
    Bases: mechmat.properties.thermal.polymer.ThermalPolymer, mechmat.
    properties.viscosity.crosswlf.CrossWLF, mechmat.properties.specific_volume.
    twodomaintaitpvt.TwoDomainTaitpvT

class mechmat.polymer.Polyamide66CrossWLF(**kwargs)
    Bases: mechmat.properties.thermal.polymer.ThermalPolymer, mechmat.properties.
    viscosity.crosswlf.CrossWLF

class mechmat.polymer.Polycarbonate_CrossWLF(**kwargs)
    Bases: mechmat.properties.thermal.polymer.ThermalPolymer, mechmat.properties.
    viscosity.crosswlf.CrossWLF
```

```
class mechmat.polymer.PolypropyleneCrossWLF(**kwargs)
    Bases: mechmat.properties.thermal.polymer.ThermalPolymer, mechmat.properties.
    viscosity.crosswlf.CrossWLF

class mechmat.polymer.PolystyreneCrossWLF(**kwargs)
    Bases: mechmat.properties.thermal.polymer.ThermalPolymer, mechmat.properties.
    viscosity.crosswlf.CrossWLF
```

### 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](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

### 7.3 0.2.0 (2019-05-25)

- Removed the need for a metaclass
- Observer pattern implemented as Chainable class
- Guarded descriptor added
- Modular materials, allows for mix and match of different models
- Two-domain-Tait-pvt added
- Cross-Arrhenius model added

### 7.4 0.2.1 (2019-05-25)

- property models, functions and values are cited by source

## 7.5 0.2.2 (2019-05-31)

- Couple of bug-fixes
- dir shows only user variables
- Interpolated function added (measurement data, can now be used)
- Serialization using dill is now possible
- repr string simplified
- Cross-WLF model added
- Some example materials added: PLA, PLA-TPU, PLA-TPS

## 7.6 0.2.3 (2019-05-31)

- Added multiple thermal properties

## 7.7 0.2.4 (2019-07-09)

- Bug fix citations

## INDICES AND TABLES

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





## PYTHON MODULE INDEX

### m

mechmat, 18  
mechmat.core, 8  
mechmat.core.chainable, 7  
mechmat.core.errors, 8  
mechmat.material, 17  
mechmat.polymer, 17  
mechmat.principal, 10  
mechmat.principal.core, 8  
mechmat.principal.crossarrhenius, 8  
mechmat.principal.density, 8  
mechmat.principal.geometry, 9  
mechmat.principal.shear\_rate, 9  
mechmat.principal.specific\_weight, 9  
mechmat.principal.thermal, 9  
mechmat.principal.twodomaintaitpvt, 10  
mechmat.properties, 17  
mechmat.properties.elastic\_deformation, 10  
mechmat.properties.flow, 11  
mechmat.properties.flow.flow, 10  
mechmat.properties.flow.mass\_flow, 10  
mechmat.properties.flow.volume\_flow, 11  
mechmat.properties.geometry, 12  
mechmat.properties.geometry.geometry, 11  
mechmat.properties.geometry.surface, 11  
mechmat.properties.geometry.vector, 11  
mechmat.properties.geometry.volume, 12  
mechmat.properties.mass, 12  
mechmat.properties.mass.mass, 12  
mechmat.properties.plastic\_deformation, 12  
mechmat.properties.pressure, 13  
mechmat.properties.pressure.pressure, 12  
mechmat.properties.shearing, 13  
mechmat.properties.shearing.shearing, 13  
mechmat.properties.specific\_volume, 15  
mechmat.properties.specific\_volume.twodomaintaitpvt, 13  
mechmat.properties.thermal, 16  
mechmat.properties.thermal.conductivity, 15  
mechmat.properties.thermal.thermal, 15  
mechmat.properties.viscosity, 17  
mechmat.properties.viscosity.crossarrhenius, 16



## A

add() (in module *mechmat.principal.core*), 8  
annulus() (in module *mechmat.principal.shear\_rate*), 9  
arrhenius\_activation\_energy (in module *mechmat.principal.crossarrhenius*), 8  
arrhenius\_shift() (in module *mechmat.principal.crossarrhenius*), 8

## B

b\_1m (in module *mechmat.principal.specific\_volume.twodomaintaitpvt*), 14  
b\_1s (in module *mechmat.principal.specific\_volume.twodomaintaitpvt*), 14  
b\_2m (in module *mechmat.principal.specific\_volume.twodomaintaitpvt*), 14  
b\_2s (in module *mechmat.principal.specific\_volume.twodomaintaitpvt*), 14  
b\_3m (in module *mechmat.principal.specific\_volume.twodomaintaitpvt*), 14  
b\_3s (in module *mechmat.principal.specific\_volume.twodomaintaitpvt*), 14  
b\_4m (in module *mechmat.principal.specific\_volume.twodomaintaitpvt*), 14  
b\_4s (in module *mechmat.principal.specific\_volume.twodomaintaitpvt*), 14  
b\_5 (in module *mechmat.principal.specific\_volume.twodomaintaitpvt*), 14  
b\_6 (in module *mechmat.principal.specific\_volume.twodomaintaitpvt*), 14  
b\_7 (in module *mechmat.principal.specific\_volume.twodomaintaitpvt*), 14  
b\_8 (in module *mechmat.principal.specific\_volume.twodomaintaitpvt*), 14  
b\_9 (in module *mechmat.principal.specific\_volume.twodomaintaitpvt*), 14

## C

Chainable (class in *mechmat.core.chainable*), 7  
circle() (in module *mechmat.principal.shear\_rate*), 9

cite\_value() (in module *mechmat.core.chainable.GuardedStaticMethod*), 7  
coordinate (in module *mechmat.properties.geometry.vector.Vector*), 11  
cross\_section (in module *mechmat.properties.geometry.surface.Surface*), 11  
CrossArrhenius (class in *mechmat.properties.viscosity.crossarrhenius*), 16

## D

density (in module *mechmat.properties.flow.mass\_flow.MassFlow*), 10  
density (in module *mechmat.properties.mass.mass.Mass*), 12  
distance (in module *mechmat.properties.geometry.vector.Segment*), 11  
distance() (in module *mechmat.principal.geometry*), 9  
div() (in module *mechmat.principal.core*), 8  
Flow (class in *mechmat.properties.flow.flow*), 10  
from\_density() (in module *mechmat.principal.specific\_weight*), 9  
from\_specific\_weight() (in module *mechmat.principal.specific\_weight*), 8

## G

Geometry (class in *mechmat.properties.geometry.geometry*), 11  
get\_B() (in module *mechmat.principal.twodomaintaitpvt*), 10  
get\_specific\_volume() (in module *mechmat.principal.twodomaintaitpvt*), 10  
get\_T\_t() (in module *mechmat.principal.twodomaintaitpvt*), 10  
get\_v\_0() (in module *mechmat.principal.twodomaintaitpvt*), 10  
get\_v\_t() (in module *mechmat.principal.twodomaintaitpvt*), 10

Guarded (class in mechmat.core.chainable), 7

## H

halfway() (in module mechmat.principal.geometry), 9

heat\_transfer\_coeff (mechmat.properties.thermal.conductivity.ThermalConductivity attribute), 15

## I

in\_range() (mechmat.core.chainable.Guarded static method), 7

Interp (class in mechmat.principal.core), 8

## L

link\_attr() (mechmat.core.chainable.Chainable method), 7

linked\_transforms() (mechmat.core.chainable.Chainable method), 7

## M

Mass (class in mechmat.properties.mass.mass), 12

mass (mechmat.properties.mass.mass.Mass attribute), 12

MassFlow (class in mechmat.properties.flow.mass\_flow), 10

massflow (mechmat.properties.flow.mass\_flow.MassFlow attribute), 10

material\_factory() (in module mechmat.material), 17

mechmat (module), 18

mechmat.core (module), 8

mechmat.core.chainable (module), 7

mechmat.core.errors (module), 8

mechmat.material (module), 17

mechmat.polymer (module), 17

mechmat.principal (module), 10

mechmat.principal.core (module), 8

mechmat.principal.crossarrhenius (module), 8

mechmat.principal.density (module), 8

mechmat.principal.geometry (module), 9

mechmat.principal.shear\_rate (module), 9

mechmat.principal.specific\_weight (module), 9

mechmat.principal.thermal (module), 9

mechmat.principal.twodomaintaitpvt (module), 10

mechmat.properties (module), 17

mechmat.properties.elastic\_deformation (module), 10

mechmat.properties.flow (module), 11

mechmat.properties.flow.flow (module), 10

mechmat.properties.flow.mass\_flow (module), 10

mechmat.properties.flow.volume\_flow (module), 11

mechmat.properties.geometry (module), 12

mechmat.properties.geometry.geometry (module), 11

mechmat.properties.geometry.surface (module), 11

mechmat.properties.geometry.vector (module), 11

mechmat.properties.geometry.volume (module), 12

mechmat.properties.mass (module), 12

mechmat.properties.mass.mass (module), 12

mechmat.properties.plastic\_deformation (module), 12

mechmat.properties.pressure (module), 13

mechmat.properties.pressure.pressure (module), 12

mechmat.properties.shearing (module), 13

mechmat.properties.shearing.shearing (module), 13

mechmat.properties.specific\_volume (module), 15

mechmat.properties.specific\_volume.twodomaintaitpvt (module), 13

mechmat.properties.thermal (module), 16

mechmat.properties.thermal.conductivity (module), 15

mechmat.properties.thermal.thermal (module), 15

mechmat.properties.viscosity (module), 17

mechmat.properties.viscosity.crossarrhenius (module), 16

Message (class in mechmat.core.chainable), 7

mul() (in module mechmat.principal.core), 8

## O

OutOfRangeError, 8

## P

point\_1 (mechmat.properties.geometry.vector.Segment attribute), 11

point\_2 (mechmat.properties.geometry.vector.Segment attribute), 11

Polyamide66CrossWLF (class in mechmat.polymer), 17

Polycarbonate\_CrossWLF (class in mechmat.polymer), 17

PolyLacticAcid (class in mechmat.polymer), 17

PolyLacticAcidThermoplasticPolyUrethane (class in mechmat.polymer), 17

PolyLacticAcidThermoplasticStarch (class in mechmat.polymer), 17  
 PolypropyleneCrossWLF (class in mechmat.polymer), 17  
 PolystyreneCrossWLF (class in mechmat.polymer), 18  
 Pressure (class in mechmat.properties.pressure.pressure), 12  
 pressure (mechmat.properties.pressure.pressure.Pressure attribute), 12

## R

reciprocal() (in module mechmat.principal.core), 8  
 relaxation\_time (mechmat.properties.viscosity.crossarrhenius.CrossArrhenius attribute), 16  
 relaxation\_time() (in module mechmat.principal.crossarrhenius), 8  
 relaxation\_time\_ref (mechmat.properties.viscosity.crossarrhenius.CrossArrhenius attribute), 16

## S

Segment (class in mechmat.properties.geometry.vector), 11  
 set\_guard() (mechmat.core.chainable.Chainable method), 7  
 shear\_rate (mechmat.properties.shearing.shearing.Shearing attribute), 13  
 shear\_thinning\_const (mechmat.properties.viscosity.crossarrhenius.CrossArrhenius attribute), 17  
 Shearing (class in mechmat.properties.shearing.shearing), 13  
 specific\_heat\_capacity (mechmat.properties.thermal.thermal.Thermal attribute), 15  
 specific\_heat\_capacity() (in module mechmat.principal.thermal), 9  
 specific\_volume (mechmat.properties.flow.mass\_flow.MassFlow attribute), 10  
 specific\_volume (mechmat.properties.mass.mass.Mass attribute), 12  
 specific\_volume\_transition\_temperature (mechmat.properties.specific\_volume.twodomaintaitpvt.TwoDomainTaitpvt attribute), 14  
 specific\_volume\_zero\_gauge\_pressure (mechmat.properties.specific\_volume.twodomaintaitpvt.TwoDomainTaitpvt attribute), 14  
 specific\_weight (mechmat.properties.flow.mass\_flow.MassFlow attribute), 10

specific\_weight (mechmat.properties.mass.mass.Mass attribute), 12  
 sub() (in module mechmat.principal.core), 8  
 Surface (class in mechmat.properties.geometry.surface), 11  
 switch\_m\_s() (in module mechmat.principal.twodomaintaitpvt), 10  
 temperature (mechmat.properties.thermal.thermal.Thermal attribute), 15  
 temperature\_cross\_arrhenius\_ref (mechmat.properties.viscosity.crossarrhenius.CrossArrhenius attribute), 17  
 temperature\_melt (mechmat.properties.thermal.thermal.Thermal attribute), 15  
 temperature\_transition (mechmat.properties.specific\_volume.twodomaintaitpvt.TwoDomainTaitpvt attribute), 14  
 temperature\_vapor (mechmat.properties.thermal.thermal.Thermal attribute), 15  
 Thermal (class in mechmat.properties.thermal.thermal), 15  
 thermal\_conductance (mechmat.properties.thermal.conductivity.ThermalConductivity attribute), 15  
 thermal\_conductivity (mechmat.properties.thermal.conductivity.ThermalConductivity attribute), 15  
 thermal\_conductivity (mechmat.properties.thermal.thermal.Thermal attribute), 15  
 thermal\_conductivity() (in module mechmat.principal.thermal), 9  
 thermal\_diffusivity (mechmat.properties.thermal.conductivity.ThermalConductivity attribute), 15  
 thermal\_diffusivity (mechmat.properties.thermal.thermal.Thermal attribute), 16  
 thermal\_diffusivity() (in module mechmat.principal.thermal), 9  
 thermal\_expansion\_coeff (mechmat.properties.thermal.thermal.Thermal attribute), 16  
 thermal\_insulation (mechmat.properties.thermal.conductivity.ThermalConductivity attribute), 15  
 thermal\_resistance (mechmat.properties.thermal.conductivity.ThermalConductivity attribute), 15

*attribute*), 15

`thermal_resistivity` (*mechmat.properties.thermal.conductivity.ThermalConductivity attribute*), 15

`thermal_transmittance_convection` (*mechmat.properties.thermal.conductivity.ThermalConductivity attribute*), 15

`thermal_transmittance_radiation` (*mechmat.properties.thermal.conductivity.ThermalConductivity attribute*), 15

`ThermalConductivity` (class in *mechmat.properties.thermal.conductivity*), 15

`TwoDomainTaitpvT` (class in *mechmat.properties.specific\_volume.twodomaintaitpvt*), 13

## U

`unlink_attr()` (*mechmat.core.chainable.Chainable method*), 7

## V

`Vector` (class in *mechmat.properties.geometry.vector*), 11

`viscosity_dynamic()` (in module *mechmat.principal.crossarrhenius*), 8

`viscosity_zero_shear_rate` (*mechmat.properties.viscosity.crossarrhenius.CrossArrhenius attribute*), 17

`viscosity_zero_shear_rate_ref` (*mechmat.properties.viscosity.crossarrhenius.CrossArrhenius attribute*), 17

`Volume` (class in *mechmat.properties.geometry.volume*), 12

`volume` (*mechmat.properties.geometry.volume.Volume attribute*), 12

`VolumeFlow` (class in *mechmat.properties.flow.volume\_flow*), 11

`volume_flow` (*mechmat.properties.flow.volume\_flow.VolumeFlow attribute*), 11

## Z

`zero_shear_viscosity()` (in module *mechmat.principal.crossarrhenius*), 8