
decaylanguage

Release 0.2.0

Jun 06, 2020

Contents

1	Installation	1
2	Usage	3
2.1	DecFiles	3
2.2	DecayLanguage	3
3	Reference	5
3.1	decaylanguage	5
3.2	decaylanguage.decay	5
4	Contributing	7
4.1	Bug reports	7
4.2	Documentation improvements	7
4.3	Feature requests and feedback	7
4.4	Development	8
5	Authors	9
6	Indices and tables	11
	Python Module Index	13
	Index	15

CHAPTER 1

Installation

At the command line:

```
pip install decaylanguage
```

Or:

```
pip install https://github.com/scikit-hep/decaylanguage.git
```


Particles are a key component when dealing with decays. Refer to the **'Particle package'** for how to deal with particles and PDG identification codes.

2.1 DecFiles

DecFiles describe decays. You can read files and query information about them.

2.2 DecayLanguage

The primary way to use `decaylanguage.decay` is through the module provided to read in a language file and produce an output. For example, for GooFit, call:

```
python -m decaylanguage.gooFit models/DtoKpipipi_v2.txt
```

You can pipe the output to a file.

Examples of interaction with the API directly are provided in the `/notebooks` folder, including svg diagrams of lines.

3.1 decaylanguage

3.2 decaylanguage.decay

3.2.1 decaylanguage.modeling.decay

A general base class representing decays.

class `decaylanguage.modeling.decay.ModelDecay` (*particle*, *daughters*=[], *name*=None)
This describes a decay very generally, with search and print features. Subclassed for further usage.

list_structure (*final_states*)

The structure in the form [(0,1,2,3)], where the dual-list is used for permutations for bose symmatrization.
So for `final_states=[a,b,c,c]`, `[a,c,[c,b]]` would be: [(0,2,3,1),(0,3,2,1)]

structure

The structure of the decay chain, simplified to only final state particles

3.2.2 decaylanguage.modeling.ampgentransform

3.2.3 decaylanguage.modeling.amplitudechain

A class representing a set of decays. Can be subclassed to provide custom converters.

class `decaylanguage.modeling.amplitudechain.AmplitudeChain` (*particle*, *daughters*=[], *line-shape*=None, *spinfactor*=None, *amp*=(1+0j), *err*=0j, *fix*=True, *name*=None)

This is a chain of decays (a “line”)

expand_lines (*linelist*)

Take a DecayTree -> list of DecayTrees with each dead-end daughter expanded to every possible combination. (recursive)

classmethod from_matched_line (*mat*)

This operates on an already-matched line.

Parameters *mat* – The groupdict output of a match

Returns A new amplitude chain instance

classmethod read_ampgen (*filename=None, text=None, grammar=None, parser='lalr', **kargs*)

Read in an ampgen file

Parameters

- **filename** – Filename to read
- **text** – Text to read (use instead of filename)

Returns array of AmplitudeChains, parameters, constants, event type

class decaylanguage.modeling.amplitudechain.**LS**

Line shapes supported (currently)

3.2.4 decaylanguage.modeling.goofit

This is a GooFit adaptor for amplitude chain.

class decaylanguage.modeling.goofit.**DecayStructure**

class decaylanguage.modeling.goofit.**GooFitChain** (*particle, daughters=[], line-shape=None, spinfactor=None, amp=(1+0j), err=0j, fix=True, name=None*)

classmethod read_ampgen (**args, **kargs*)

Read in an ampgen file

Parameters

- **filename** – Filename to read
- **text** – Text to read (use instead of filename)

Returns array of AmplitudeChains, parameters, constants, event type

class decaylanguage.modeling.goofit.**SF_4Body**

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

4.1 Bug reports

When [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.

4.2 Documentation improvements

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

4.3 Feature requests and feedback

The best way to send feedback is to file an issue at <https://github.com/scikit-hep/decaylanguage/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 code contributions are welcome :)

4.4 Development

To set up *decaylanguage* for local development:

1. Fork [decaylanguage](#) (look for the “Fork” button).
2. Clone your fork locally:

```
git clone git@github.com:your_name_here/decaylanguage.git
```

3. Create a branch for local development:

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

Now you can make your changes locally.

4. When you’re done making changes, run all the checks, doc builder and spell checker with `tox` one command:

```
tox
```

5. 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
```

6. Submit a pull request through the GitHub website.

4.4.1 Pull Request Guidelines

If you need some code review or feedback while you’re developing the code just make the pull request.

For merging, you should:

1. Include passing tests (run `tox`)¹.
2. Update documentation when there’s new API, functionality etc.
3. Add a note to `CHANGELOG.md` about the changes.
4. Add yourself to `AUTHORS.rst`.

4.4.2 Tips

To run a subset of tests:

```
tox -e envname -- py.test -k test_myfeature
```

To run all the test environments in *parallel* (you need to `pip install detox`):

```
detox
```

¹ If you don’t have all the necessary python versions available locally you can rely on Travis - it will run the tests for each change you add in the pull request.

It will be slower though ...

CHAPTER 5

Authors

- Henry Fredrick Schreiner III - <https://iscinumpy.gitlab.io>
- Eduardo Rodrigues - <https://github.com/eduardo-rodrigues/>

CHAPTER 6

Indices and tables

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

d

decaylanguage, 5

decaylanguage.modeling.ampgentransform,
5

decaylanguage.modeling.amplitudechain,
5

decaylanguage.modeling.decay, 5

decaylanguage.modeling.goofit, 6

A

AmplitudeChain (class in *decaylanguage.modeling.amplitudechain*), 5

D

decaylanguage (module), 5

decaylanguage.modeling.ampgentransform (module), 5

decaylanguage.modeling.amplitudechain (module), 5

decaylanguage.modeling.decay (module), 5

decaylanguage.modeling.goofit (module), 6

DecayStructure (class in *decaylanguage.modeling.goofit*), 6

E

expand_lines() (decaylanguage.modeling.amplitudechain.AmplitudeChain method), 6

F

from_matched_line() (decaylanguage.modeling.amplitudechain.AmplitudeChain class method), 6

G

GooFitChain (class in *decaylanguage.modeling.goofit*), 6

L

list_structure() (decaylanguage.modeling.decay.ModelDecay method), 5

LS (class in *decaylanguage.modeling.amplitudechain*), 6

M

ModelDecay (class in *decaylanguage.modeling.decay*), 5

R

read_ampgen() (decaylanguage.modeling.amplitudechain.AmplitudeChain class method), 6

read_ampgen() (decaylanguage.modeling.goofit.GooFitChain class method), 6

S

SF_4Body (class in *decaylanguage.modeling.goofit*), 6

structure (decaylanguage.modeling.decay.ModelDecay attribute), 5