
pytransport Documentation

Release 0.1

Royal Holloway

Nov 09, 2017

CONTENTS

1	Licence & Disclaimer	3
1.1	Licence	3
2	Authorship	5
3	Installation	7
3.1	Requirements	7
3.2	Installation	7
4	Conversion	9
5	Module Contents	11
5.1	pytransport.convert module	11
5.2	pytransport.elements module	12
5.3	pytransport.reader	13
6	Indices and tables	15
	Python Module Index	17
	Index	19

pytransport is a set of classes and functions to load MADX output as well as prepare Transport models. The package overall functions as a holder for any code required to load and manipulate Transport output data.

LICENCE & DISCLAIMER

pytransport copyright (c) Royal Holloway, University of London, 2017. All rights reserved.

1.1 Licence

This software is provided “AS IS” and any express or limit warranties, including, but not limited to, implied warranties of merchantability, of satisfactory quality, and fitness for a particular purpose or use are disclaimed. In no event shall Royal Holloway, University of London be liable for any direct, indirect, incidental, special, exemplary, or consequential damages arising in any way out of the use of this software, even if advised of the possibility of such damage.

AUTHORSHIP

The following people have contributed to pytransport:

- William Shields
- Jochem Snuerink
- Laurie Nevay
- Stuart Walker

INSTALLATION

3.1 Requirements

- pytransport was developed for the Python 2.7 series.

pytransport depends on the following Python packages not included with Python:

- matplotlib
- numpy
- scipy
- pymadx
- pybdsim

3.2 Installation

A *setup.py* file required for a correct python installation is currently under development.

Currently, we recommend the user clones the source repository and exports the parent directory to their PYTHONPATH environmental variable. This will allow Python to find pytransport.:

```
pwd
/Users/nevay/physics/rep
git clone http://bitbucket.org/jairhul/pytransport
ls
> pytransport
export PYTHONPATH=/Users/nevay/physics/rep

python
>>> import pytransport # no errors!
```


CONVERSION

Conversion.

MODULE CONTENTS

This documentation is automatically generated by scanning all the source code. Parts may be incomplete. A module for converting a TRANSPORT file into gmad for use in BDSIM.

To use: `>>> self = pytransport.convert.pytransport() >>> self.load_file(TRANSPORTfile) >>> self.convert()`

Will output:

filename.gmad filename_beam.gmad filename_components.gmad filename_options.gmad filename_samplers.gmad
filename_sequence.gmad

5.1 pytransport.convert module

```
class pytransport.convert.pytransport (inputfile, particle='proton', debug=False, distrType='gauss', gmad=True, gmadDir='gmad', madx=False, madxDir='madox', auto=True, dontSplit=False, keepName=False, combineDrifts=False, outlog=True)
```

Bases: `pytransport.elements.elements`

A module for converting a TRANSPORT file into gmad for use in BDSIM.

To use:

```
>>> self = pytransport.convert.pytransport(inputfile)
```

Will output the lattice in the appropriate format.

Parameters:

particle: string The particle type, default = 'proton'.

debug: boolean Output debug strings, default = False.

distrType: string The distribution type of the beam, default = 'gauss'. Can only handle 'gauss' and 'gausstwiss'. If madx output is specified, the madx beam distribution is 'madox'.

gmad: boolean Write the converted output into gmad format, default = True.

gmadDir: string Output directory for gmad format, default = 'gmad'

madox: boolean write the converted output into madx format, default = False.

madoxDir: string Output directory for madx format, default = 'madox'

auto: boolean Automatically convert and output the file, default = True.

keepName: boolean Keep original element name if present, default = False

combineDrifts: boolean Combine consecutive drifts into a single drift, default = False

outlog: boolean Output stream to a log file, default = True

AddLatticeToRegistry ()

Function that loops over the lattice, adds the elements to the element registry, and updates any elements that have fitted parameters.

ProcessAndBuild ()

Function to convert the registry elements into pybdsim format and add to the pybdsim builder.

UpdateElementsFromFits ()

create_beam ()

Function to prepare the beam for writing.

create_options ()

Function to set the Options for the BDSIM machine.

transport2gmad ()

Function to convert TRANSPORT file on a line by line basis.

write ()

5.2 pytransport.elements module

class pytransport.elements.elements

Bases: pytransport._General.functions

acceleration (*linedict*)

A Function that writes the properties of an acceleration element Only RF added for gmad, not for madx!

change_bend (*linedict*)

Function to change the direction of the dipole bend. Can be a direction other than horizontal (i.e != $n\pi$).

collimator (*linedict*)

A Function that writes the properties of a collimator element Only added for gmad, not for madx!

correction (*linedict*)

define_beam (*linedict*)

dipole (*linedict*)

drift (*linedict*)

printline (*linedict*)

quadrupole (*linedict*)

sextupole (*linedict*)

solenoid (*linedict*)

special_input (*linedict*)

unit_change (*linedict*)

Function to change the units (scaling) of various parameters.

5.3 pytransport.reader

class pytransport.reader.optics

class pytransport.reader.reader

getOptics (*file*, *type=None*)

INDICES AND TABLES

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

PYTHON MODULE INDEX

p

`pytransport`, [11](#)

`pytransport.convert`, [11](#)

`pytransport.elements`, [12](#)

`pytransport.reader`, [13](#)

INDEX

A

acceleration() (pytransport.elements.elements method), 12
AddLatticeToRegistry() (pytransport.convert.pytransport method), 12

C

change_bend() (pytransport.elements.elements method), 12
collimator() (pytransport.elements.elements method), 12
correction() (pytransport.elements.elements method), 12
create_beam() (pytransport.convert.pytransport method), 12
create_options() (pytransport.convert.pytransport method), 12

D

define_beam() (pytransport.elements.elements method), 12
dipole() (pytransport.elements.elements method), 12
drift() (pytransport.elements.elements method), 12

E

elements (class in pytransport.elements), 12

G

getOptics() (pytransport.reader.reader method), 13

O

optics (class in pytransport.reader), 13

P

println() (pytransport.elements.elements method), 12
ProcessAndBuild() (pytransport.convert.pytransport method), 12
pytransport (class in pytransport.convert), 11
pytransport (module), 11
pytransport.convert (module), 11
pytransport.elements (module), 12
pytransport.reader (module), 13

Q

quadrupole() (pytransport.elements.elements method), 12

R

reader (class in pytransport.reader), 13

S

sextupole() (pytransport.elements.elements method), 12
solenoid() (pytransport.elements.elements method), 12
special_input() (pytransport.elements.elements method), 12

T

transport2gmad() (pytransport.convert.pytransport method), 12

U

unit_change() (pytransport.elements.elements method), 12
UpdateElementsFromFits() (pytransport.convert.pytransport method), 12

W

write() (pytransport.convert.pytransport method), 12