
simex_platform Documentation

Release 0.4.1

Carsten Fortmann-Grote, Sergey Yakubov

Nov 06, 2020

CONTENTS

1	Acknowledgements	1
2	Introduction	3
3	Usage	5
4	Contribute	7
5	Support	9
6	License	11
7	Contents	13
7.1	Acknowledgements	13
7.2	Introduction	13
7.3	Installation	14
7.4	Testing	19
7.5	Usage	19
7.6	Examples	19
7.7	Contribute	26
7.8	Support	26
7.9	License	26
7.10	API Manual	26
7.11	Data format documentation	72
	Python Module Index	77
	Index	79

ACKNOWLEDGEMENTS

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 654220 and No. 823852.

INTRODUCTION

SIMEX platform provides a software platform for simulation of experiments at advanced laser and x-ray light sources. All aspects of a typical experiment, the photon source, light transport through optics elements in the beamline, interaction with a target or sample, scattering from the latter, photon detection, and data analysis can be modelled. A simulation can contain, one, several, or all of these parts.

As an example, consider a coherent imaging experiment using x-ray free electron laser (XFEL) radiation: A molecule, cluster, or nanocrystal is irradiated by highly brilliant, ultrashort x-ray pulses, these scatter from the sample get detected in an x-ray detector.

The x-ray photons will ionize and destroy the sample after a very short time, typically of the order of a few tens to a few hundreds of femtoseconds ($1 \text{ fs} = 1 \times 10^{-15} \text{ s}$). By using x-ray pulses that are even shorter, of the order of a few femtoseconds, this radiation damage can, at least partly, be avoided, i.e. the sample is probed before destruction. The scattered photons are registered in a area pixel detector and the scattering (diffraction) pattern can be analyzed to infer structural information about the sample, i.e. the 3D electron density and the position of atoms within the molecule.

SIMEX platform provides scriptable python user interfaces, SimEx API, to a number of advanced simulation codes for the various stages of the experiment: Photon Source, Photon Propagation, Photon-Matter Interaction, Photon Diffraction and Scattering, Photon Detection, and Photon Data Analysis. Additionally, `simex_platform` defines data interfaces such that the involved simulation codes can “talk” to each other. E.g. output from a photon source calculation can be fed into the photon propagation simulation.

The SimEx API is open-source, but some of the interfaced simulation codes are not. In such cases, the user has to acquire the simulation code and install on his system.

USAGE

The intended usage scheme of SIMEX platform is that of a python module in either an interactive (i)python session or a python script. Hence, the first thing to do is to import the SimEx API module.

```
>>> import SimEX
```

A good starting point for finding examples on how to use SIMEX platform modules are the tests under `Tests/python/unittest/SimExTest/`. Each Calculator is tested, and the test-suite `PhotonExperimentSimulation/PhotonExperimentSimulationTest.py` contains some SIMEX platform workflow examples for complete start-to-end simulations.

CONTRIBUTE

- Source Code: <https://github.com/PaNOSC-ViNYL/SimEx/>
- Issue Tracker: <https://github.com/PaNOSC-ViNYL/SimEx/issues>

**CHAPTER
FIVE**

SUPPORT

If you are having issues, please let us know at carstendotgroteatxfeldoteu .

LICENSE

The project is licensed under the GPL open source license version 3.

CONTENTS

7.1 Acknowledgements

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No. 654220 and No. 823852.

7.2 Introduction

SIMEX platform provides a software platform for simulation of experiments at advanced laser and x-ray light sources. All aspects of a typical experiment, the photon source, light transport through optics elements in the beamline, interaction with a target or sample, scattering from the latter, photon detection, and data analysis can be modelled. A simulation can contain, one, several, or all of these parts.

As an example, consider a coherent imaging experiment using x-ray free electron laser (XFEL) radiation: A molecule, cluster, or nanocrystal is irradiated by highly brilliant, ultrashort x-ray pulses, these scatter from the sample get detected in an x-ray detector.

The x-ray photons will ionize and destroy the sample after a very short time, typically of the order of a few tens to a few hundreds of femtoseconds ($1 \text{ fs} = 1 \times 10^{-15} \text{ s}$). By using x-ray pulses that are even shorter, of the order of a few femtoseconds, this radiation damage can, at least partly, be avoided, i.e. the sample is probed before destruction. The scattered photons are registered in a area pixel detector and the scattering (diffraction) pattern can be analyzed to infer structural information about the sample, i.e. the 3D electron density and the position of atoms within the molecule.

SIMEX platform provides scriptable python user interfaces, SimEx API, to a number of advanced simulation codes for the various stages of the experiment: Photon Source, Photon Propagation, Photon-Matter Interaction, Photon Diffraction and Scattering, Photon Detection, and Photon Data Analysis. Additionally, `simex_platform` defines data interfaces such that the involved simulation codes can “talk” to each other. E.g. output from a photon source calculation can be fed into the photon propagation simulation.

The SimEx API is open-source, but some of the interfaced simulation codes are not. In such cases, the user has to acquire the simulation code and install on his system.

7.3 Installation

Contents

- *Installation*
 - *From sources*
 - * *Download*
 - * *Software dependencies*
 - *Python dependencies*
 - *Non-python dependencies*
 - *Module specific dependencies*
 - * *Install only the python SimEx API*
 - * *Install the backengine modules along with the SimEx python API*
 - *Configuration via cmake*
 - *Module selection*
 - *Build the documentation*
 - *Developer install*
 - *Create deb packages*
 - *Building the library*
 - *Installation*
 - * *Environment settings*
 - * *Troubleshooting*
 - *cmake fails to resolve dependency but the library is installed*
 - *gomp/iomp not found / MKL not found*

7.3.1 From sources

To build and install from the source code directly.

Download

First obtain the source code by either cloning the repository:

```
$> git clone https://github.com/PaNOSC-ViNYL/SimEx
```

or by downloading and extracting the [zip](#) archive.

After downloading (and extracting), switch into the top level directory:

```
$> cd SimEx
```

Software dependencies

Python dependencies

The python dependencies are listed in `$SIMEX_ROOT/requirements.txt`, reproduced here:

Note that these requirements contain all dependencies of all modules. A customized build with only selected Modules may well get away with fewer dependencies installed. The cmake build system will check all dependencies for each individual Module. Cmake will abort the configuration if missing dependencies are detected for a given Module.

The python dependencies can be installed via `pip`:

```
$> python -m pip install -r requirements.txt
```

Except for `xraydb`, all python requirements can also be installed through `conda`.

Non-python dependencies

The following libraries/executables should be installed on your system:

- cmake 3.12+
- libgsl0-dev
- libfftw3-dev or INTEL-MKL
- wget
- hdf5 (version $\geq 1.10.4$) + header files
- python3.5+
- libbz2-dev
- unzip
- C/C++ and Fortran compilers, e.g. gcc

Module specific dependencies

- boost (version > 1.54) + header files (only needed for Detector simulations with the XCSITPhotonDetector Module)
- libfftw3-dev or INTEL-MKL (needed to run WPG/SRW wavefront propagation)

For conda, we provide an `environment.yml` file to create a simex environment:

```
$> conda env create --file environment.yml
```

See below for how to install the dependencies and backengines into the same environment.

NOTE 1 (Intel(R) MKL(R)): If you want to link against Intel(R) MKL(R), make sure that the Intel(R) MKL(R) environment variables are set. This is typically done by running one of the scripts in `$INTEL_HOME/bin/`, where `$INTEL_HOME` is the root directory of the Intel(R) MKL(R) installation, e.g. `/opt/intel/2015`.

NOTE 2 (BOOST): Sometimes `boost_mpi` is not built although all libraries (default) where requested as per `project.conf` in the boost build directory. Append a “using mpi ;” to that file (without the quotes) to enforce building `boost_mpi`.

NOTE 3 (BOOST): It has been observed that newer versions of boost (>1.61), if linked against mpich, require libmpich.so.12, which might not be available on all systems, especially not completely updated clusters. Use boost.1.60 or lower if this problem occurs. You can find out by running ldd on libboost_mpi.so.

Install only the python SimEx API

If, for whatever obscure reason, you only need the SimEx python library (the Calculators and Parameters), simply run:

```
$> cd Sources/python
$> python -m pip install [-e] .
```

The `-e` option is recommended for developers: Changes in `Sources` would be reflected in the imported python module.

Install the backengine modules along with the SimEx python API

The SimEx python API and backengine modules (aka Modules) can be installed by following the steps below.

The install process has three stages: configuration, building, and installing.

Configuration via cmake

This step requires a dedicated build directory. Create one, and change into it:

```
$> mkdir build
$> cd build
```

Configuration is done by issuing the command `cmake ...` `cmake` accepts numerous command line arguments. To list them all along with their defaults, run:

```
$> cmake -LAH
```

To set a flag/argument to a non-default value, it is appended to the `cmake` command. E.g. to set the installation prefix (path under which all SimEx libraries and executables will be installed):

```
$> cmake .. -DCMAKE_INSTALL_PREFIX=/path/to/some/directory
```

Note the capital “D” before the actual flag.

Module selection

As of version 0.5, no Module is installed by default. To switch to the old behaviour and install all Modules, set the flag `USE_MODULES_DEFAULT`:

```
$> cmake .. -DUSE_MODULES_DEFAULT=ON + further flags and arguments]
```

To keep the new behaviour AND select individual modules, append each selected module with a `-DUSE_` prefix. E.g. to activate the propagation Module based on WPG:

```
$> cmake .. -DUSE_wpg=ON
```

To activate the SingFELPhotonDiffraction:

```
$> cmake .. -DUSE_SingFELPhotonDiffractor=ON
```

By setting the installation prefix to `$CONDA_PREFIX`, one can install the backengines and the simex library into the same environment.

ii. Wave propagation with OpenMP:

```
$> cmake .. -DUSE_MODULES_DEFAULT=ON -DUSE_wpg=OFF
```

In this example, all but the wpg module will be installed.

Build the documentation

This will build this documentation using the source code at `Sources/doc`:

```
$> cmake -DBUILD_DOC=ON ..
```

Developer install

This is recommended for SimEx developers. In this way, you will be able to run the unittests without having to recompile:

```
$> cmake -DCMAKE_INSTALL_PREFIX=..
```

Create deb packages

This will create `.deb` packages:

```
$> cmake -DPACKAGE_MAKE=ON -DCMAKE_INSTALL_PREFIX=/usr ..
$> make package
```

Probably you will have to call `cmake` two times because for some unknown reason CMake creates `.tgz` archives in the first time.

The package can then be installed system-wide along with all necessary dependencies:

```
$> dpkg -i <package_name>
$> apt-get install -f
```

on another computer with Debian based OS. Simex will be installed in `/usr/...`, Tests are installed in `/usr/share/simex/...` and should be system-wide available. Calling `dpkg` with `--instdir` option allows to change installation dir. In this case `simex_vars.sh` should be modified manually to set paths correctly.

Building the library

After successful completion of cmake, just type:

```
$> make
```

On machines with more than 1 CPU, compilation can be sped up with:

```
$> make -jN
```

where N is the number of CPUs to consume.

An example build & installation script is provided (install.sh). It might need manual adjustment as indicated.

Installation

Finally, after make returns, install the compiled software into the installation directory:

```
$> make install
```

Make sure that the user has write access to the installation directory, or use:

```
$> sudo make install
```

Environment settings

simex_platform is a python library, hence python needs to be aware of its location. To this end, run the command:

```
$> source <install_prefix>/bin/simex_vars.sh
```

after installation.

Troubleshooting

cmake fails to resolve dependency but the library is installed

On some systems cmake fails to find the paths for some of the third party libraries like boost. If this should be the case, consult the corresponding FindXXX.cmake scripts in the CMake directory and in your system's configuration for how to help cmake find these libraries. An example for how to specify paths for boost is given in the install.sh script that comes with the sources.

gomp/iomp not found / MKL not found

If compiling with Intel compilers and/or using MKL, run this command before cmake:

```
$> source `which compilervars.sh` intel64
```

7.4 Testing

Testing the installation.

By default, the installation also creates the unittest suite. You can switch this option off by appending the flag `-DINSTALL_TESTS=OFF` to the cmake command.

It is advised to run the test suite to check your installation in your source file path:

```
$> ./get_testdata.sh
$> cd Tests/python/unittest
$> python Test.py -v 2>&1 | tee Test.log
```

This will run the entire test suite and pipe the output to the file `Test.log`. A final test report is appended.

7.5 Usage

The intended usage scheme of SIMEX platform is that of a python module in either an interactive (i)python session or a python script. Hence, the first thing to do is to import the SimEx API module.

```
>>> import SimEX
```

A good starting point for finding examples on how to use SIMEX platform modules are the tests under `Tests/python/unittest/SimExTest/`. Each Calculator is tested, and the test-suite `PhotonExperimentSimulation/PhotonExperimentSimulationTest.py` contains some SIMEX platform workflow examples for complete start-to-end simulations.

7.6 Examples

7.6.1 Wavefront propagation using the SimEx Calculator “WavePropagator”

In this notebook, we will demonstrate how to propagate a simple Gaussian wavefront between two planes through vacuum. This is the simplest case of coherent wavefront propagation.

```
[1]: %load_ext autoreload
```

```
[2]: %autoreload 2
```

```
[3]: # Import all SimEx modules
from SimEx.Calculators.WavePropagator import WavePropagator
from SimEx.Calculators.GaussianPhotonSource import GaussianPhotonSource
from SimEx.Parameters.WavePropagatorParameters import WavePropagatorParameters
from SimEx.Parameters.GaussWavefrontParameters import GaussWavefrontParameters

from SimEx.Utilities.Units import electronvolt, meter, joule, radian
import numpy

initializing ocelot...
```

Setup the initial wavefront

We first create a wavefront with a Gaussian intensity distribution. To this end, we use the `GaussianPhotonSource` Calculator and its corresponding parameter class, the `GaussWavefrontParameters`. Looking up the documentation of these two classes gives us the information needed to create the instances:

```
[4]: GaussWavefrontParameters?
```

```
[5]: GaussianPhotonSource?
```

We first describe the wavefront parameters which will then be used to create the wavefront itself:

```
[4]: wavefront_parameters = GaussWavefrontParameters(photon_energy=8.0e3*electronvolt,
                                                    photon_energy_relative_bandwidth=1e-3,
                                                    beam_diameter_fwhm=1.0e-4*meter,
                                                    pulse_energy=2.0e-6*joule,
                                                    number_of_transverse_grid_points=400,
                                                    number_of_time_slices=12,
                                                    )
```

Now, we use the just created `wavefront_parameters` to initialize the Photon Source.

```
[5]: photon_source = GaussianPhotonSource(wavefront_parameters, input_path="/dev/null",
    ↪output_path="initial_wavefront.h5")
```

Let's calculate the initial wavefront and visualize it:

```
[6]: photon_source.backengine()

beam waist radius from divergence angle = 8.493e-05
beam waist radius from fwhm = 8.493e-05
```

We can retrieve the wavefront data from the calculator:

```
[7]: wavefront = photon_source.data
```

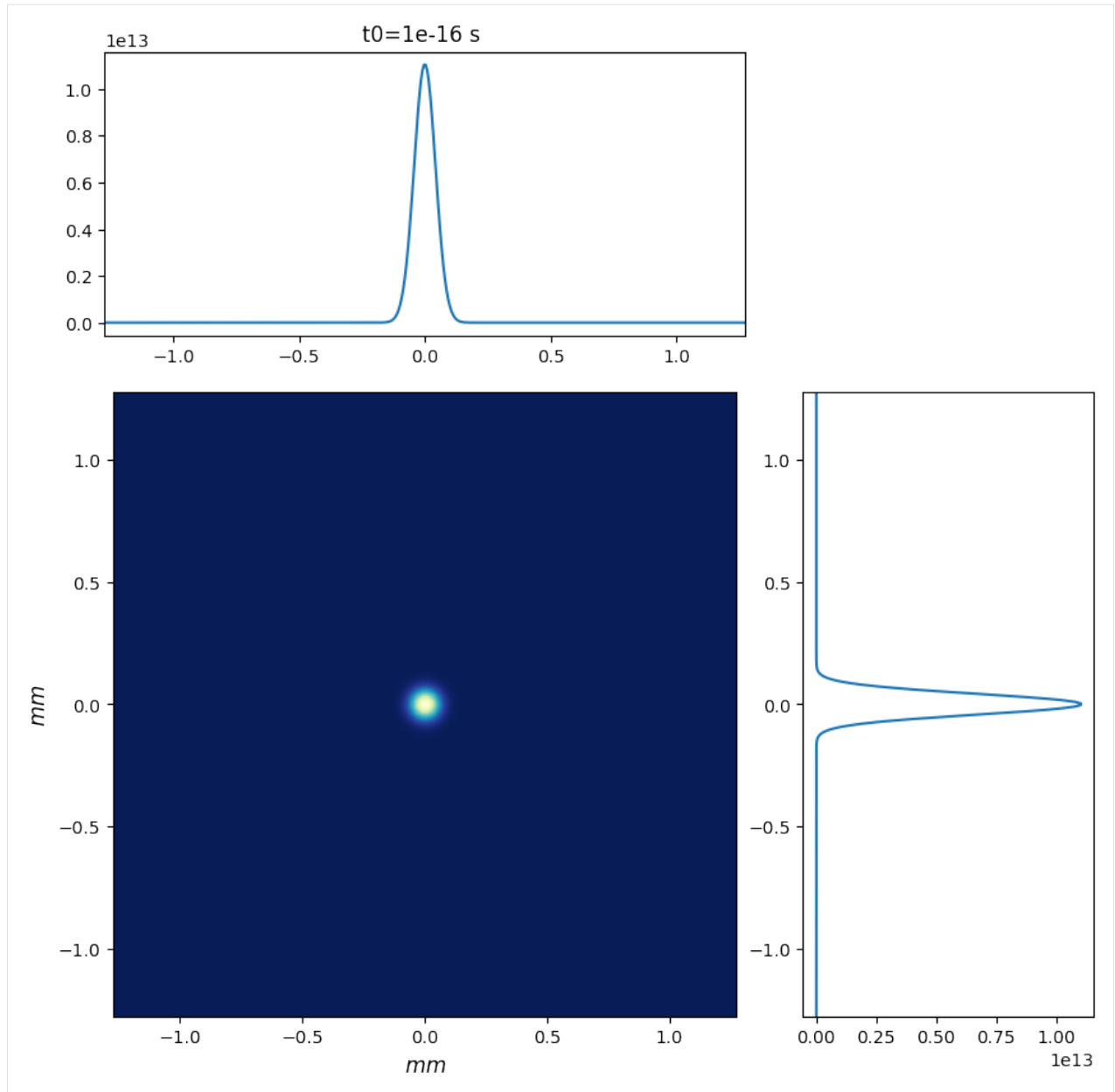
Let's visualize the wavefront using the WPG utilities.

```
[8]: from wpg import wpg_util_wf as wpg_utils
```

First plot the intensity distribution

```
[9]: wpg_utils.plot_intensity_map(wavefront)

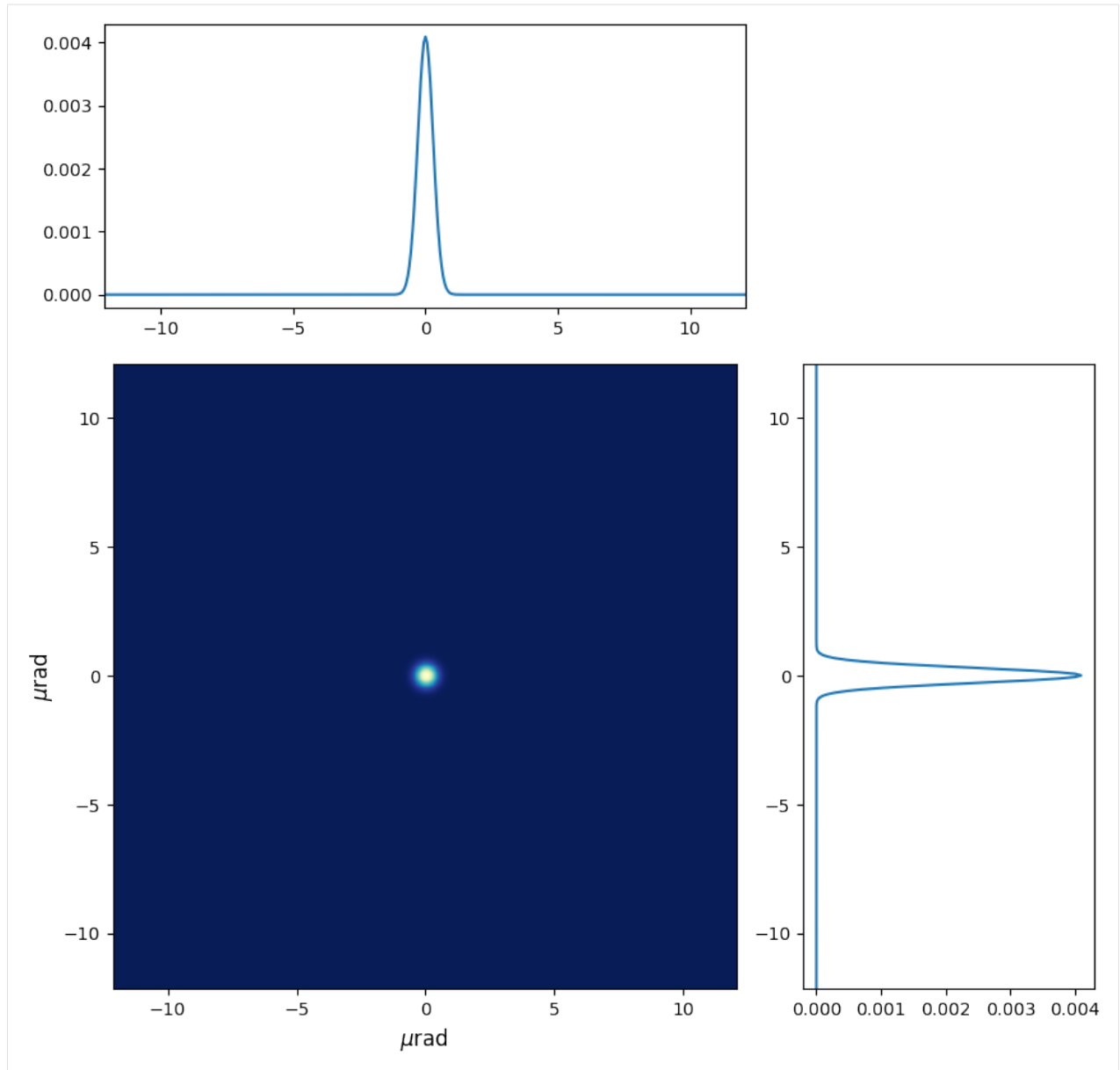
R-space
(400,) (400,)
FWHM in x = 1.019e-04 m.
FWHM in y = 1.019e-04 m.
```

Plot intensity distribution in q-space

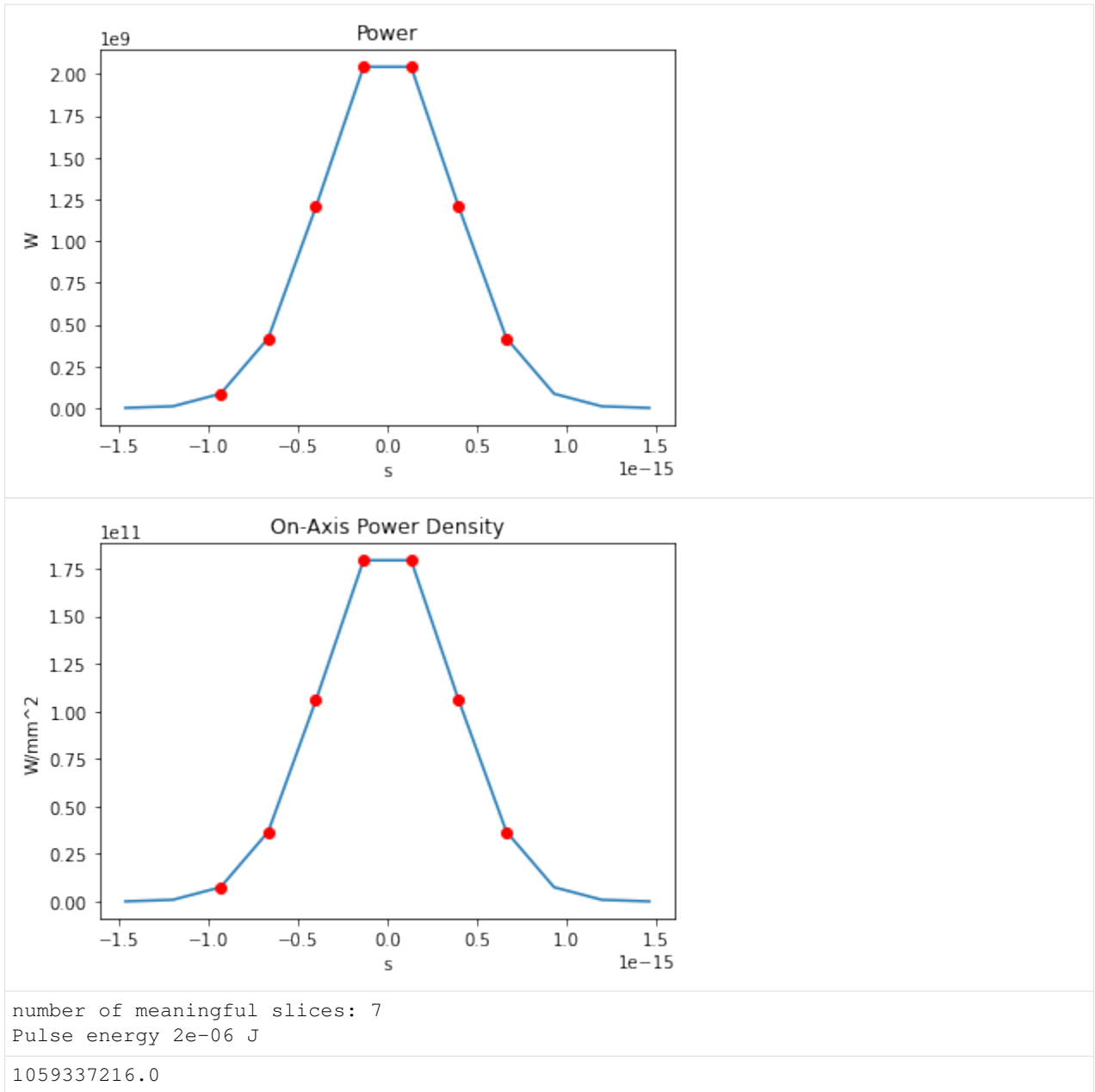
```
[10]: wpg_utils.plot_intensity_qmap(wavefront)
```

```
Q-space
{'fwhm_x': 6.657123479604694e-07, 'fwhm_y': 6.657123479604694e-07}
Q-space
(400,) (400,)
```



Plot the power as a function of time integrated over the transverse dimensions

```
[11]: wpg_utils.integral_intensity(wavefront)
```



Plot the power spectrum

```
[12]: import wpg
```

Check the sampling quality

```
[13]: print(wpg_utils.check_sampling(wavefront))
```

```
WAVEFRONT SAMPLING REPORT
+-----+-----+-----+-----+-----+-----+-----+
|x/y      |FWHM   |px      |ROI      |R        |Fzone   |px*7    |px*10   |
+-----+-----+-----+-----+-----+-----+-----+
|Horizontal|1.019e-04|6.386e-06|2.548e-03|0.000e+00|0.000e+00|4.470e-05|6.386e-05|
|Vertical  |1.019e-04|6.386e-06|2.548e-03|0.000e+00|0.000e+00|4.470e-05|6.386e-05|
```

(continues on next page)

(continued from previous page)

```

+-----+-----+-----+-----+-----+-----+-----+-----+
Horizontal Fresnel zone extension NOT within [7,10]*pixel_width -> Check pixel width."
Vertical Fresnel zone extension NOT within [7,10]*pixel_height -> Check pixel width."
Horizontal ROI !> 3* FWHM(x) -> Increase ROI width (x).
Horizontal ROI !> 3* FWHM(y) -> Increase ROI height (y).
Focus sampling: FWHM > 10*px

END OF REPORT

```

Ok, we're happy with our initial wavefront and will now proceed to setup the beamline. Our beamline will consist just of a stretch of vacuum over 10 m distance.

Estimate the beam size after 100 m propagation through empty space

Before we launch the numerical propagation machinery, let's estimate the beam size to be expected after propagating through 10 m of vacuum.

The beam waist size (rms width of the E-field distribution, or distance from beam axis where intensity drops to $1/e^2$ of its on-axis value) is given by the expression

$w(z) = w_0 \sqrt{1 + \left(\frac{z}{R}\right)^2}$, with the Rayleigh length $R = \pi w_0^2 / \lambda$ is the central wavelength of the laser (ca., 0.15 nm at 8 keV.)

Our Gaussian source distribution has a FWHM of $10^{-4}m$. The FWHM and the beam waist radius as defined above are related to each other through $\text{FWHM}(z) = \sqrt{2 \ln 2} w(z) \simeq 1.177 w(z)$

Beam waist at $z=0$: $w_0 = 100 \mu\text{m} / 1.177 \simeq 84.9 \mu\text{m}$.

The Rayleigh length for our Gaussian beam is $R = \frac{\pi w_0^2}{0.15 \text{nm}} \simeq 151 \text{m}$.

At $z = 100 \text{m}$, we therefore expect a FWHM of $100 \mu\text{m} \cdot \sqrt{1 + (100/151)^2} \simeq 119.9 \mu\text{m}$.

For numerical reasons, we have to set the radius of curvature to a finite value. Here we choose to set it to a value corresponding approximately to a distance of 1 m from the source position, $z = 1 \text{m}$.

```
[14]: wavefront.params.Rx = wavefront.params.Ry = 150**2
```

Setup the beamline

```
[21]: from wpg import Beamline, optical_elements, srwlib
```

```
[16]: beamline = Beamline()
```

```
[17]: free_space = optical_elements.Drift(_L=100, _treat=1)
```

```
[18]: free_space_propagation_parameters = optical_elements.Use_PP(semi_analytical_
↳treatment=1)
```

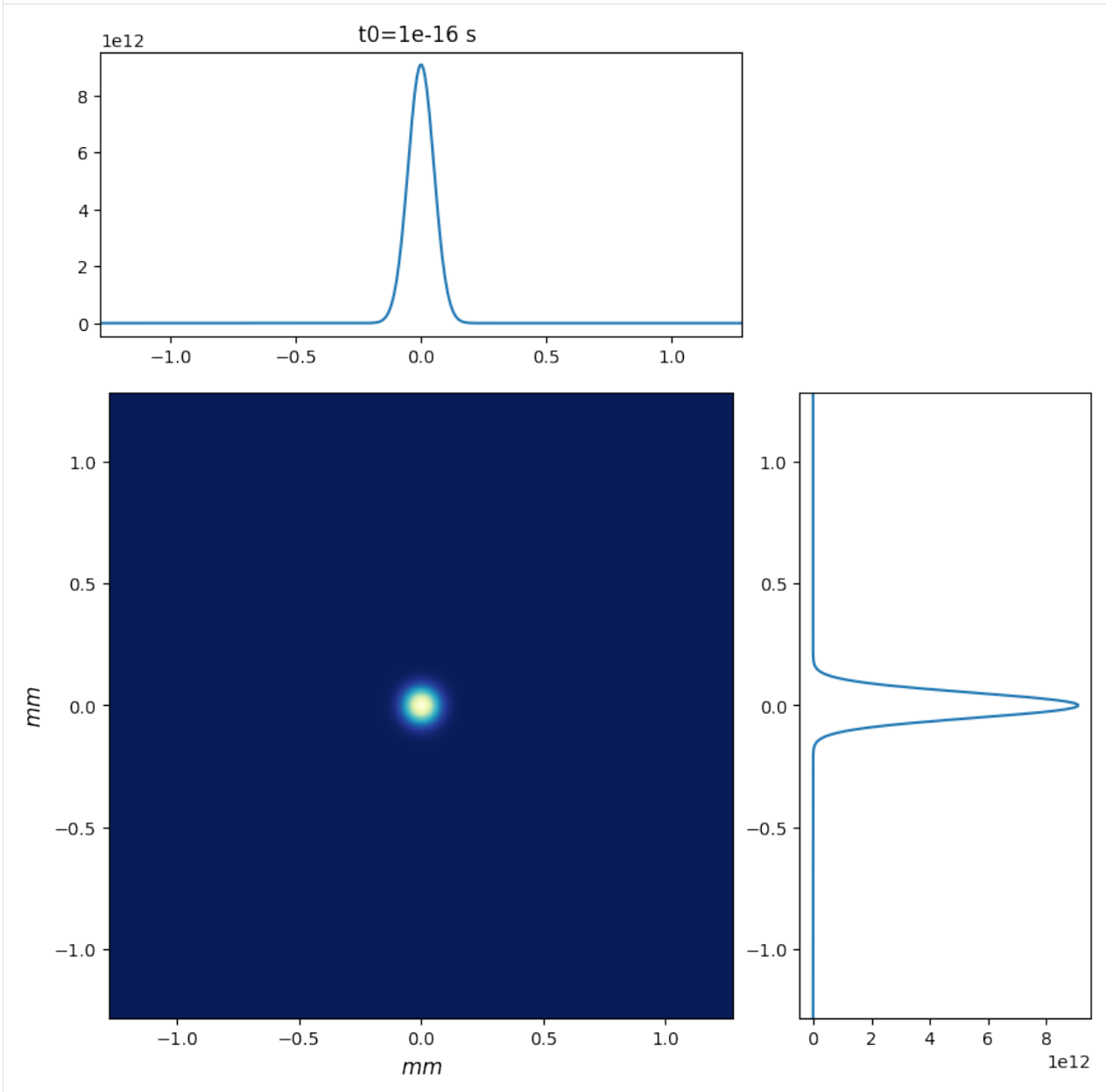
```
[19]: beamline.append(free_space, free_space_propagation_parameters)
```

```
[20]: srwlib.srwlib.SetRepresElecField(wavefront._srwl_wf, 'f')
beamline.propagate(wavefront)
srwlib.srwlib.SetRepresElecField(wavefront._srwl_wf, 't')
```

```
[20]: <wpg.srw.srwlib.SRWLWfr at 0x2b83161af750>
```

```
[22]: wpg_utils.plot_intensity_map(wavefront)
```

```
R-space
(400,) (400,)
FWHM in x = 1.152e-04 m.
FWHM in y = 1.152e-04 m.
```



The result is $\text{FWHM} = 115.2 \mu\text{m}$ at $z = 100$ m. Considering that the FWHM is measured by counting pixels in the

intensity profile, this is in rather good agreement with the analytical calculation above.

7.7 Contribute

- Source Code: <https://github.com/PaNOSC-ViNYL/SimEx/>
- Issue Tracker: <https://github.com/PaNOSC-ViNYL/SimEx/issues>

7.8 Support

If you are having issues, please let us know at carstendotgroteatxfeldoteu .

7.9 License

The project is licensed under the GPL open source license version 3.

7.10 API Manual

7.10.1 SimEx packages

SimEx.Analysis package

SimEx.Analysis.AbstractAnalysis module

module AbstractAnalysis Hosts the abstract base class for all analysis classes.

class `SimEx.Analysis.AbstractAnalysis.AbstractAnalysis` (*input_path=None*)
Bases: `object`

Class AbstractAnalysis Abstract base class for all data analysis classes.

Parameters `input_path` (*str*) – Path to hdf5 file holding the data to analyze. Single file or directory.

classmethod `dumpLoader` (*fname*)
Creates calculator object from a dump file

Parameters `fname` – path to the dump file.

Returns Created calculator object.

Raises `RuntimeError` – if cannot create object.

dumpToFile (*fname*)
dump class instance to file.

Parameters `fname` – Path to file to dump.

property `input_path`
Query for the input file path(s).

classmethod `runFromCLI()`

Method to start calculator computations from command line.

Returns exit with status code

`SimEx.Analysis.AbstractAnalysis.checkAndSetBaseCalculator` (*var=None, default=None*) *de-*

Check if passed object is an AbstractAnalysis instance. If non is given, set to given default.

Parameters

- **var** – The object to check.
- **default** – The default to use.

Returns The checked object.

Raises `RuntimeError` – if no valid Analysis object was given.

SimEx.Analysis.DiffractionAnalysis module

module `DiffractionAnalysis` Module that hosts the DiffractionAnalysis class.

class `SimEx.Analysis.DiffractionAnalysis.DiffractionAnalysis` (*input_path=None, pattern_indices=None, poissonize=True, mask=None*)

Bases: `SimEx.Analysis.AbstractAnalysis.AbstractAnalysis`

Class `DiffractionAnalysis` Class that implements common data analysis tasks for diffraction data.

Parameters

- **input_path** (*str*) – Name of file or directory that contains data to analyse.
- **pattern_indices** (*int || sequence of int || "all"*) – Identify which patterns to include in the analysis (default “all”).
- **poissonize** (*bool*) – Whether to add Poisson noise to the integer photon numbers (default True).
- **mask** (*numpy.array*) – Mask to multiply on each pattern.

Example `pattern_indices` `pattern_indices=1`

`pattern_indices=[1,2,3]`

`pattern_indices=range(1,10)`

`pattern_indices="all"`

animatePatterns (*output_path=None, logscale=False, offset=0.1*)

Make an animated gif out of the given patterns.

Parameters

- **output_path** (*str*) – Where to save the animated gif.
- **logscale** (*bool*) – Whether to apply logarithmic scaling to the z axis (color).
- **offset** (*float*) – Offset to apply if logarithmic scaling is on.

Raises `IOError` – File exists or parent directory not found.

property mask

Query the mask.

property npattern

get the number of the selected patterns in this analysis object

numpyPattern (*operation=None*)

Return the pattern after operation over the patterns defined in DiffractionAnalysis class.

Parameters **operation** (*python function*) – Operation to apply to selected patterns (default none).

Note operation Operation must accept a 3D numpy.array as first input argument and the “axis” keyword-argument. Operation must return a 2D numpy.array. Axis will always be chosen as axis=0.

Example operation numpy.mean, numpy.std, numpy.sum

property parameters

patternGenerator ()

Yield an iterator over a given pattern sequence from a diffraction file.

property pattern_indices

Query pattern indices attribute.

property patterns_iterator

plotPattern (*operation=None, logscale=False, offset=0.1, symlog=False, *argv, **kwargs*)

Plot a pattern.

Parameters

- **operation** (*python function*) – Operation to apply to selected patterns (default numpy.sum).
- **logscale** (*bool*) – Whether to plot the intensity on a logarithmic scale (z-axis) (default False).
- **offset** (*float*) – Offset to apply if logarithmic scaling is on.

Note operation Operation must accept a 3D numpy.array as first input argument and the “axis” keyword-argument. Operation must return a 2D numpy.array. Axis will always be chosen as axis=0.

Example operation numpy.mean, numpy.std, numpy.sum

plotRadialProjection (*operation=None, logscale=False, offset=1e-05, unit='q_nm^-1'*)

Plot the radial projection of a pattern.

Parameters

- **operation** (*python function*) – Operation to apply to selected patterns (default numpy.sum).
- **logscale** (*bool*) – Whether to plot the intensity on a logarithmic scale (z-axis) (default False).

Note operation Operation must accept a 3D numpy.array as first input argument and the “axis” keyword-argument. Operation must return a 2D numpy.array. Axis will always be chosen as axis=0.

Example operation numpy.mean, numpy.std, numpy.sum

:param unit:can be “q_nm^-1”, “q_A^-1”, “2th_deg”, “2th_rad”, “r_mm”. :type unit: str

property poissonize

Query whether to read data with (True) or without (False) Poisson noise.

property qMap

q of each pixel

shannonPixelPhoton (*resolution*)

Get the average number of photons per shannon pixel

Parameters **resolution** (*float*) – The full periodic resolution (Å) for shannon pixels

property solidAngles

Solid angle of each pixel

statistics ()

Get statistics of photon numbers per pattern (mean and rms) over selected patterns and plot a histogram.

SimEx.Analysis.DiffractionAnalysis.**azimuthalIntegration** (*pattern*, *parameters*,
unit='q_nm^-1')

SimEx.Analysis.DiffractionAnalysis.**diffractionParameters** (*path*)

Extract beam parameters and geometry from given file or directory.

Parameters **path** (*str*) – Path to file that holds the parameters to extract.

SimEx.Analysis.DiffractionAnalysis.**photonStatistics** (*stack*)

SimEx.Analysis.DiffractionAnalysis.**plotImage** (*pattern*, *logscale*=False, *offset*=0.1, *symlog*=False, **argv*, ***kwargs*)

Workhorse function to plot an image

Parameters

- **logscale** (*bool*) – Whether to show the data on logarithmic scale (z axis) (default False).
- **offset** (*float*) – Offset to apply if logarithmic scaling is on.
- **symlog** (*bool*) – If logscale is True, to show the data on symlogarithmic scale (z axis) (default False).

Returns the handles of figure and axis

Return type figure,axis

SimEx.Analysis.DiffractionAnalysis.**plotRadialProjection** (*pattern*, *parameters*,
logscale=True, *offset*=1e-05, *unit*='q_nm^-1')

Perform integration over azimuthal angle and plot as function of radius.

:param unit: can be “q_nm^-1”, “q_A^-1”, “2th_deg”, “2th_rad”, “r_mm”. :type unit: str

SimEx.Analysis.DiffractionAnalysis.**plotResolutionRings** (*parameters*, *rings*=(10, 5.0, 3.5), *half*=True)

Show resolution rings on current plot.

Parameters

- **parameters** (*dict*) – Parameters needed to construct the resolution rings.
- **rings** (*list*) – the rings shown on the figure
- **half** (*bool*) – show half period resolution (True, default) or full period resolution (False)

SimEx.Analysis.DiffractionAnalysis.**totalNPattern** (*input_path*)

get the number of the diffraction patterns in the h5file

SimEx.Analysis.XFELPhotonAnalysis module

module XFELPhotonAnalysis Module that hosts the XFELPhotonAnalysis class.

class SimEx.Analysis.XFELPhotonAnalysis.XFELPhotonAnalysis (*input_path=None*)

Bases: *SimEx.Analysis.AbstractAnalysis.AbstractAnalysis*

Class XFELPhotonAnalysis Class that implements common data analysis tasks for wavefront (radiation field) data.

Parameters *input_path* (*str*) – Name of file or directory that contains data to analyse.

animate (*qspace=False, logscale=False*)

Generate an animated gif from the wavefront data.

property intensity

Query for the intensity.

numpyTotalPower (*spectrum=False, all=False*)

Method to dump meaningful total power.

Parameters

- **spectrum** (*bool*) – Whether to dump the power density in energy domain (True) or time domain (False, default).
- **all** (*bool*) – True to extract all the points, False to extract only meaningful points

plotIntensityMap (*qspace=False, logscale=False*)

Plot the integrated intensity as function of x,y or qx, qy on a colormap.

Parameters

- **qspace** (*bool*) – Whether to plot the reciprocal space intensity map (default False).
- **logscale** (*bool*) – Whether to plot the intensity on a logarithmic scale (z-axis) (default False).

plotOnAxisPowerDensity (*spectrum=False*)

Method to plot the on-axis power density.

Parameters **spectrum** (*bool*) – Whether to plot the power density in energy domain (True) or time domain (False, default).

plotTotalPower (*spectrum=False*)

Method to plot the total power.

Parameters **spectrum** (*bool*) – Whether to plot the power density in energy domain (True) or time domain (False, default).

property wavefront

Query for the wavefront.

SimEx.Analysis.XFELPhotonAnalysis.**calculate_fwhm** (*wfr*)

Calculate FWHM of the beam calculating number of point bigger then max/2 throughgt center of the image

Parameters **wfr** – wavefront

Returns {'fwhm_x':fwhm_x, 'fwhm_y': fwhm_y} in [m]

SimEx.Analysis.XFELPhotonAnalysis.**mask_nans** (*a, replacement=0.0*)

Find nans in an array and replace. :param a: Array to mask. :type a: numpy.array

Parameters **replacement** (*numeric*) – The value to replace nans.

Returns The array of booleans indicating which values of a are nan, numpy.isnan(a)

Return type numpy.array(dtype=bool)

SimEx.Analysis.XMDYNPhotonMatterAnalysis module

module XMDYNPhotonMatterAnalysis Hosting utilities to analyse and visualize photon-matter trajectories generated by XMDYN.

class SimEx.Analysis.XMDYNPhotonMatterAnalysis.XMDYNPhotonMatterAnalysis (*input_path=None, snapshot_indices=None, elements=None, sample_path=None*)

Bases: *SimEx.Analysis.AbstractAnalysis.AbstractAnalysis*

Class XMDYNPhotonMatterAnalysis Class to encapsulate diagnostics of photon matter interaction trajectories.

Parameters

- **input_path** (*str*) – Path to the data to analyze.
- **snapshot_indices** (*list*) – Snapshot (indices or IDs) to analyze (Default: All).
- **elements** (*list*) – Which elements to include in the analysis (Default: All).

animate ()

Generate an animation of atom trajectories and their ionization.

property elements

Query the elements to include.

property input_path

Query the input path.

load_snapshot (*snapshot_index*)

Load snapshot data from hdf5 file into memory.

load_trajectory ()

Load the selected snapshots and extract data to analyze.

number_of_snapshots ()

Get number of valid snapshots.

plot_charge ()

Plot the average number of electrons per atom per atomic species as function of time.

plot_displacement ()

Plot the average displacement per atomic species as function of time.

plot_energies ()

Plot the evolution of MD energies over the simulation time.

property sample_path

Query the sample path.

property snapshot_indices

Query the snapshot indices.

SimEx.Analysis.XMDYNPhotonMatterAnalysis.**calculate_displacement** (*snapshot*, *r0*, *sample*)

Calculate the average displacement per atomic species in a snapshot.

Parameters

- **snapshot** (*dict*) – The snapshot to analyze
- **r0** (*numpy.array (shape=(Natoms, 3))*) – Unperturbed positions of the sample atoms.

CHECKME: Can't we read r0 from the sample dict?

Parameters sample (*dict*) – Sample data

SimEx.Analysis.XMDYNPhotonMatterAnalysis.**calculate_ion_charge** (*snapshot*, *sample*)

Calculate the remaining electric charge per atomic species of a given snapshot.

Parameters

- **snapshot** (*dict*) – The snapshot to analyze
- **sample** (*dict*) – The sample data.

SimEx.Analysis.XMDYNPhotonMatterAnalysis.**load_sample** (*sample_path*)

Load a sample file into memory.

SimEx.Analysis.XMDYNPhotonMatterAnalysis.**read_h5_dataset** (*path*, *dataset*)

Read a dataset from hdf5 file.

SimEx.Calculators package

SimEx.Calculators.AbstractBaseCalculator module

module AbstractBaseCalculator Hosting the base class of all Calculators.

class SimEx.Calculators.AbstractBaseCalculator.**AbstractBaseCalculator** (*parameters=None*, *in-put_path=None*, *out-put_path=None*)

Bases: SimEx.AbstractBaseClass.AbstractBaseClass

Class AbstractBaseCalculator Abstract class for all simulation calculators.

Parameters

- **parameters** (*dict || AbstractCalculatorParameters*) – Parameters of the calculation (not data).
- **input_path** (*str*) – Path to hdf5 file holding the input data.
- **output_path** (*str*) – Path to hdf5 file for output.

abstract backengine ()

Method to call the backengine for the calculator. To be implemented on the derived classes.

classmethod dumpLoader (*fname*)

Creates calculator object from a dump file

Parameters fname – path to the dump file.

Returns Created calculator object.

Raises `RuntimeError` – if cannot create object.

dumpToFile (*fname*)

dump class instance to file.

Parameters *fname* – Path to file to dump.

abstract expectedData ()

Query for the data fields expected by this calculator.

property input_path

Query for the input file path(s).

property output_path

Query for the output file path(s).

property parameters

Query for the control parameters of the calculator.

abstract providedData ()

Query for the data fields provided by this calculator.

classmethod runFromCLI ()

Method to start calculator computations from command line.

Returns exit with status code

abstract saveH5 ()

`SimEx.Calculators.AbstractBaseCalculator.checkAndSetBaseCalculator` (*var=None, de-fault=None*)

Check if passed object is an `AbstractBaseCalculator` instance. If non is given, set to given default.

Parameters

- **var** – The object to check.
- **default** – The default to use.

Returns Te checked photon source object.

Raises `RuntimeError` – if no valid `BaseCalculator` was given.

`SimEx.Calculators.AbstractBaseCalculator.checkAndSetIO` (*io*)

Check the passed io path/filenames and set appropriately.

`SimEx.Calculators.AbstractBaseCalculator.checkAndSetParameters` (*parameters*)

Utility to check if the ‘parameters’ argument is valid input.

Parameters *parameters* (*dict* or `AbstractCalculatorParameters`) – The parameters object to check.

Returns The checked parameters object.

SimEx.Calculators.AbstractPhotonAnalyzer module

module AbstractPhotonAnalyzer Module for AbstractPhotonAnalyzer

class SimEx.Calculators.AbstractPhotonAnalyzer.**AbstractPhotonAnalyzer** (*parameters=None, in-put_path=None, out-put_path=None*)

Bases: *SimEx.Calculators.AbstractBaseCalculator.AbstractBaseCalculator*

Class AbstractPhotonAnalyzer Abstract base class for all PhotonAnalyzers.

Parameters

- **parameters** (*dict* || *AbstractCalculatorParameters*) – Parameters of the calculation (not data).
- **input_path** (*str*) – Path to hdf5 file holding the input data.
- **output_path** (*str*) – Path to hdf5 file for output.

SimEx.Calculators.AbstractPhotonAnalyzer.**checkAndSetPhotonAnalyzer** (*var=None, de-fault=None*)

Check if passed object is an AbstractPhotonAnalyzer instance. If non is given, set to given default.

Parameters

- **var** – The object to check.
- **default** – The default to use.

Returns The checked photon source object.

Raises **RuntimeError** – if no valid PhotonAnalyzer was given.

SimEx.Calculators.AbstractPhotonDetector module

module AbstractPhotonDetector Hosts the base class for all PhotonDetectors.

class SimEx.Calculators.AbstractPhotonDetector.**AbstractPhotonDetector** (*parameters=None, in-put_path=None, out-put_path=None*)

Bases: *SimEx.Calculators.AbstractBaseCalculator.AbstractBaseCalculator*

Class AbstractPhotonDetector Abstract base class for all PhotonDetectors.

SimEx.Calculators.AbstractPhotonDetector.**checkAndSetPhotonDetector** (*var=None, de-fault=None*)

Check if passed object is an AbstractPhotonDetector instance. If non is given, set to given default.

Parameters

- **var** – The object to check.
- **default** – The default to use.

Returns The checked photon source object.

Raises `RuntimeError` – if no valid PhotonDetector was given.

SimEx.Calculators.AbstractPhotonDiffractor module

module AbstractPhotonDiffractor Hosts the base class for all Diffractors.

```
class SimEx.Calculators.AbstractPhotonDiffractor.AbstractPhotonDiffractor (parameters=None,
                                                                    in-
                                                                    put_path=None,
                                                                    out-
                                                                    put_path=None)
```

Bases: *SimEx.Calculators.AbstractBaseCalculator.AbstractBaseCalculator*

Abstract base class for all PhotonDiffractors.

Parameters

- **parameters** (*dict* || *AbstractCalculatorParameters*) – Parameters of the calculation (not data).
- **input_path** (*str*) – Path to hdf5 file holding the input data.
- **output_path** (*str*) – Path to hdf5 file for output.

```
SimEx.Calculators.AbstractPhotonDiffractor.checkAndSetPhotonDiffractor (var=None,
                                                                    de-
                                                                    fault=None)
```

Check if passed object is an AbstractPhotonDiffractor instance. If non is given, set to given default.

@param var : The object to check. @param default : The default to use. @return : The checked photon source object. @throw : RuntimeError if no valid PhotonDiffractor was given.

SimEx.Calculators.AbstractPhotonInteractor module

module AbstractPhotonInteractor Hosts the base class for all PhotonInteractors.

```
class SimEx.Calculators.AbstractPhotonInteractor.AbstractPhotonInteractor (parameters=None,
                                                                    in-
                                                                    put_path=None,
                                                                    out-
                                                                    put_path=None,
                                                                    sam-
                                                                    ple_path=None)
```

Bases: *SimEx.Calculators.AbstractBaseCalculator.AbstractBaseCalculator*

Class AbstractPhotonInteractor Abstract base class for all PhotonInteractors.

Parameters

- **parameters** (*dict* || *AbstractCalculatorParameters*) – Parameters of the calculation (not data).
- **input_path** (*str*) – Path to hdf5 file holding the input data.
- **output_path** (*str*) – Path to hdf5 file for output.
- **sample_path** (*str*) – Path to file containing the sample atom coordinates (at least atom type and cartesian coordinates).

SimEx.Calculators.AbstractPhotonInteractor.**checkAndSetPhotonInteractor** (*var=None, de-fault=None*)

Check if passed object is an AbstractPhotonInteractor instance. If non is given, set to given default.

@param *var* : The object to check. @param *default* : The default to use. @return : The checked photon source object. @throw : RuntimeError if no valid PhotonInteractor was given.

SimEx.Calculators.AbstractPhotonPropagator module

module AbstractPhotonPropagator Module holding the AbstractPhotonPropagator abstract class.

class SimEx.Calculators.AbstractPhotonPropagator.**AbstractPhotonPropagator** (*parameters=None, in-put_path=None, out-put_path=None*)

Bases: *SimEx.Calculators.AbstractBaseCalculator.AbstractBaseCalculator*

Class AbstractPhotonPropagator Base class for all PhotonPropagators.

Parameters

- **parameters** (*dict* // *AbstractCalculatorParameters*) – Parameters of the calculation (not data).
- **input_path** (*str*) – Path to hdf5 file holding the input data.
- **output_path** (*str*) – Path to hdf5 file for output.

expectedData ()

Return the list of expected data sets with full paths.

Returns List of strings, e.g. ['/data/data1', '/params/params1']

providedData ()

Return the list of provided data sets with full paths.

Returns List of strings, e.g. [/data/data1, /params/params1']

SimEx.Calculators.AbstractPhotonPropagator.**checkAndSetPhotonPropagator** (*var=None, de-fault=None*)

Check if passed object is an AbstractPhotonPropagator instance. If non is given, set to given default.

Parameters

- **var** – The object to check
- **default** – The default to use

Returns The checked photon source object

Raises **RuntimeError** – if no valid PhotonPropagator was given.

SimEx.Calculators.AbstractPhotonSource module

module AbstractPhotonSource Hosting the base class for all PhotonSources.

```
class SimEx.Calculators.AbstractPhotonSource.AbstractPhotonSource (parameters=None,  
in-  
put_path=None,  
out-  
put_path=None)
```

Bases: *SimEx.Calculators.AbstractBaseCalculator.AbstractBaseCalculator*

Class AbstractPhotonSource Abstract base class for all PhotonSources.

Parameters

- **parameters** (*dict* || *AbstractCalculatorParameters*) – Parameters of the calculation (not data).
- **input_path** (*str*) – Path to hdf5 file holding the input data.
- **output_path** (*str*) – Path to hdf5 file for output.

expectedData ()

Return the list of expected data sets with full paths.

Returns List of strings, e.g. [/data/data1, /params/params1']

providedData ()

Return the list of provided data sets with full paths.

Returns List of strings, e.g. [/data/data1, /params/params1']

```
SimEx.Calculators.AbstractPhotonSource.checkAndSetPhotonSource (var=None, de-  
fault=None)
```

Check if passed object is an AbstractPhotonSource instance. If non is given, set to given default.

Parameters

- **var** – The object to check.
- **default** – The default to use.

Returns The checked photon source object.

Throw RuntimeError if no valid PhotonSource was given.

SimEx.Calculators.ComptonScatteringCalculator module

module ComptonScatteringCalculator Hosts the ComptonScatteringCalculator class.

```
class SimEx.Calculators.ComptonScatteringCalculator.ComptonScatteringCalculator (parameters=None,  
in-  
put_path=None,  
out-  
put_path=None)
```

Bases: *SimEx.Calculators.AbstractPhotonDiffraction.AbstractPhotonDiffraction*

Class ComptonScatteringCalculator Class representing a Compton scattering calculator.

:param parameters : Parameters for the ComptonScatteringCalculator. :type parameters : dict

Parameters

- **input_path** (*str*) – Path to input data for Compton Scattering calculations (Default: 'xrts_in.h5').
- **output_path** (*str*) – Path to output data for Compton Scattering calculations (Default: 'xrts_out.h5').

backengine ()

This method drives the backengine xrts.

property data

Query for the field data.

expectedData ()

Query for the data expected by the Diffractor.

providedData ()

Query for the data provided by the Diffractor.

saveH5 ()

`SimEx.Calculators.ComptonScatteringCalculator.checkAndSetParameters` (*parameters*)
 Utility to check if the parameters dictionary is ok .

`SimEx.Calculators.ComptonScatteringCalculator.fermihalf` (*x, sgn*)
 Series approximation to the $F_{1/2}(x)$ or $F_{-1/2}(x)$ Fermi-Dirac integral. Credits: Greg von Winckel <http://www.scientificpython.net/pyblog/approximate-fermi-dirac-integrals>

SimEx.Calculators.CrystFELPhotonDiffractor module

module CrystFELPhotonDiffractor Module that holds the CrystFELPhotonDiffractor class.

class `SimEx.Calculators.CrystFELPhotonDiffractor.CrystFELPhotonDiffractor` (*parameters=None, in-put_path=None, out-put_path=None*)

Bases: `SimEx.Calculators.AbstractPhotonDiffractor.AbstractPhotonDiffractor`

Class CrystFELPhotonDiffractor Represents simulation of photon diffraction by crystals using CrystFEL.oattern_sim.

Parameters

- **parameters** (`CrystFELPhotonDiffractorParameters`) – Parameters of the calculation (not data).
- **input_path** (*str*) – Path to hdf5 file holding the input data.
- **output_path** (*str*) – Path to hdf5 file for output.

backengine ()

This method drives the backengine CrystFEL.pattern_sim.

computeNTasks ()

Calculate the number of MPI tasks as function of available resources and assigned cpus per task.

property data

Query for the field data.

expectedData ()

Query for the data expected by the Diffractor.

providedData ()

Query for the data provided by the Diffractor.

saveH5 ()

Method to save the output to a file. Creates links to h5 files that all contain only one pattern.

saveH5_geom ()

Method to save the output to a file. Creates links to h5 files that all contain only one pattern.

SimEx.Calculators.DMPhasing module

module DMPhasing Module that holds the DMPhasing class.

class SimEx.Calculators.DMPhasing.**DMPhasing** (*parameters=None, input_path=None, output_path=None*)

Bases: *SimEx.Calculators.AbstractPhotonAnalyzer.AbstractPhotonAnalyzer*

Class DMPhasing Encapsulates photon diffraction analysis for electron density reconstruction from oriented 3D diffraction patterns.

Parameters

- **parameters** (*DMPhasingParameters instance*) – Phasing parameters.
- **input_path** (*str*) – Path to input data (file or directory).
- **output_path** (*str*) – Path to output data (file or directory).

backengine ()

Start the actual calculation.

Returns 0 if the DM run was successful, 1 if not.

property data

Query for the field data.

Returns The stored 3D electron density map.

expectedData ()

Query for the data expected by the Analyzer.

Returns A list of strings telling which datasets are expected to be found in the input file.

providedData ()

Query for the data provided by the Analyzer.

Returns A list of strings indicating which datasets are produced in the output data.

saveH5 ()

SimEx.Calculators.DMPhasing.**show_support** (*support*)

SimEx.Calculators.EMCCaseGenerator module

module EMCCaseGenerator Hosting the EMCCaseGenerator class that sets up a EMC run.

class SimEx.Calculators.EMCCaseGenerator.**EMCCaseGenerator** (*runLog=None*)

Bases: `object`

Class EMCCaseGenerator Encapsulates one EMC case.

Parameters `runLog` (*str, default None (don't save log.)*) – Flag that indicates where to save the runtime log.

createTestScatteringGeometry ()

ATTENTION: Untested! Contains recipe to create, diffract and show a low-pass-filtered, random particle contrast.

If particle and diffraction parameters are not given, then default ones are used: `particleRadius = 5.9` (num. of pixels; good results if number is `x.9`, where `x` is an integer), `damping = 1.5` (larger damping=larger Debye-Waller factor), `frac = 0.5` (frac. of most intense realspace voxels forced to persist in iterative particle generation), `pad = 1.8` (extra voxels to pad on 3D particle density to create support for phasing), `radius = numpy.floor(particleRadius) + numpy.floor(pad)` (half length of cubic volume that holds particle), `size = 2*radius + 1` (length of cubic volume that holds particle).

diffractTestCase (*inMaxScattAngDeg=45.0, inSigma=6.0, inQminNumShannon-Pix=1.4302966531242025*)

ATTENTION: Untested! Requires `makeMonster()` to first be called, so that particle density is created.

Function `diffract()` needs the maximum scattering angle to the edge of the detector, the sampling rate of Shannon pixels (`inSigma=6` means each Shannon pixel is sampled by roughly 6 pixels), and the central missing data region has a radius of `inQminNumShannonPix` (in units of Shannon pixels).

Variables redefined here: `z = cotangent of maximum scattering angle`, `sigma = sampling rate on Shannon pixels`, `qmax = number of pixels to edge of detector`, `numPixToEdge = same as qmax`, `detectorDist = detector-particle distance (units of detector pixels)`, `beamstop = voxel positions of central disk of missing data on detector`, `detector = pixel position of 2D area detector (projected on Ewald sphere)`, `intensities = 3D Fourier intensities of particle`.

makeTestParticleAndSupport (*inParticleRadius=5.9, inDamping=1.5, inFrac=0.5, inPad=1.8*)

ATTENTION: Untested! Recipe for creating random, “low-passed-filtered binary” contrast by alternating binary projection and low-pass-filter on an random, 3D array of numbers.

Variables defined here: `support = spherical particle support (whose radius is less than particleRadius given)`, `density = 3D particle contrast`, `supportPositions = voxel position of support used in phasing`.

placePixel (*ii, jj, zL*)

Gives (`qx, qy, qz`) position of pixels on Ewald sphere when given as input the (`x, y, z`)=(`ii, jj, zL`) position of pixel in the diffraction laboratory. The latter is measured in terms of the size of each realspace pixel.

Parameters

- `ii` (*int*) – Pixel index in x direction
- `jj` (*int*) – Pixel index in y direction
- `zL` (*float*) – Distance of pixel from detector in units of detector size.

readGeomFromDetectorFile (*fn='detector.dat'*)

Read `qx, qy, qz` coordinates of detector and beamstop from `detector.dat`.

readGeomFromPhotonData (*fn, thisProcess*)

Extract detector geometry from S2E photon files.

Parameters `fn` (*str*) – Path to file to read.

showDensity ()

ATTENTION: Untested! Shows particle density as an array of sequential, equal-sized 2D sections.

showDetector ()

Shows detector pixels as points on scatter plot; could be slow for large detectors.

showLogIntensity (*inSection=0*)

ATTENTION: Untested! Show a particular intensities section of Fourier intensities. Sections range from -qmax to qmax.

showLogIntensitySlices ()

ATTENTION: Untested! Shows Fourier intensities as an array of sequential, equal-sized 2D sections. Maximum intensities set to logarithm of maximum intensity in 3D Fourier volume.

writeAllOutputToFile (*supportFileName='support.dat', densityFileName='density.dat', detectorFileName='detector.dat', intensitiesFileName='intensity.dat'*)

ATTENTION: Untested! Convenience function for writing output

Parameters

- **supportFileName** (*str, default 'support.dat'*) – Path to file for support data.
- **densityFileName** (*str, default 'density.dat'*) – Path to file for density data.
- **detectorFileName** (*str, default 'detector.dat'*) – Path to file for detector data.
- **intensitiesFileName** (*str, default 'intensities.dat'*) – Path to file for intensities data.

writeDensityToFile (*filename='density.dat'*)

Write electron density to a file. ATTENTION: Untested! :param filename: Path to file. :type filename: str

writeDetectorToFile (*filename='detector.dat'*)

Writes computed detector and beamstop coordinates to output.

Parameters filename (*str*) – Path of file where to write the detector data.

writeSparsePhotonFile (*fileList, outFN, outFNH5Avg, thisProcess, numProcesses*)

Convert dense S2E file format to sparse EMC photons.dat format.

Parameters

- **fileList** (*List of str.*) – List of files to convert into one sparse photon file.
- **outFN** (*str, default 'photons.dat'*) – Filename of sparse photon file.
- **outFNH5Avg** (*str*) – Filename for averaged photon file.

writeSupportToFile (*filename='support.dat'*)

Write the support range to a file. ATTENTION: Untested! :param filename: Path to file. :type filename: str

SimEx.Calculators.EMCCaseGenerator.**cluster_two_means** (*vals*)

SimEx.Calculators.EMCCaseGenerator.**find_two_means** (*vals, v0, v1*)

SimEx.Calculators.EMCCaseGenerator.**load_intensities** (*ref_file*)

SimEx.Calculators.EMCCaseGenerator.**support_from_autocorr** (*auto, qmax, thr_0, thr_1, kl=1, write=True*)

SimEx.Calculators.EMCCaseGenerator.**zero_neg** (*x*)

SimEx.Calculators.EMCOrientation module

module EMCOrientation Module that holds the EMCOrientation class.

class SimEx.Calculators.EMCOrientation.**EMCOrientation** (*parameters=None, input_path=None, output_path=None, tmp_files_path=None, run_files_path=None*)

Bases: *SimEx.Calculators.AbstractPhotonAnalyzer.AbstractPhotonAnalyzer*

Class EMCOrientation Representing photon data analysis for orientation of 2D diffraction patterns to a 3D diffraction volume.

Parameters

- **parameters** (*EMCOrientationParameters instance*) – Parameters for the EMC orientation calculator.
- **input_path** (*str*) – Path to directory holding input data for EMC.
- **output_path** (*str*) – Path to file where output data will be stored.
- **tmp_files_path** (*str*) – Path to directory where temporary files will be stored.
- **run_files_path** (*str*) – Path to directory where run data will be stored, in particular the sparse photons file ‘photons.dat’ and ‘detector.dat’.

Note If ‘run_files_path’ is an existing directory that contains data from a previous EMC run, the current run will append to the existing data. A consistency check is performed.

backengine ()

Starts EMC simulations in parallel in a subprocess

computeNTasks ()

property data

Query for the field data.

expectedData ()

Query for the data expected by the Analyzer.

providedData ()

Query for the data provided by the Analyzer.

property run_files_path

saveH5 ()

property tmp_files_path

SimEx.Calculators.EstherPhotonMatterInteractor module

module EstherPhotonMatterInteractor Module that holds the EstherPhotonMatterInteractor class.

class SimEx.Calculators.EstherPhotonMatterInteractor.**EstherPhotonMatterInteractor** (*parameters=, in-put_path=N, out-put_path=N*)

Bases: *SimEx.Calculators.AbstractPhotonInteractor.AbstractPhotonInteractor*

Class EstherPhotonMatterInteractor Class interfacing the Esther Radiation-Hydrodynamics simulation backengine.

Parameters

- **parameters** (`EstherPhotonMatterInteractorParameters`) – Parameters for the EstherPhotonMatterInteractor.
- **input_path** (`str`) – Path to the input data for this calculator.
- **output_path** (`str`) – Path to write output data generated by this calculator to.

backengine ()

This method drives the esther backengine code.

property data

Query for the field data.

expectedData ()

Query for the data expected by the Diffractor.

providedData ()

Query for the data provided by the Diffractor.

saveH5 ()

Method to save the data to a file.

SimEx.Calculators.FEFFFotonMatterInteractor module

module FEFFFotonMatterInteractor Holds the FEFFFotonMatterInteractor class.

class `SimEx.Calculators.FEFFFotonMatterInteractor.FEFFFotonMatterInteractor` (`parameters=None`, `input_path=None`, `output_path=None`)

Bases: `SimEx.Calculators.AbstractPhotonInteractor`, `AbstractPhotonInteractor`

Class FEFFFotonMatterInteractor Interface class for photon-matter interaction calculations using the FEFF code.

Parameters

- **parameters** (`FEFFFotonMatterInteractorParameters`) – Parameters that govern the PMI calculation.
- **input_path** (`str`) – Location of data needed by the PMI calculation.
- **output_path** (`str`) – Where to store the data generated by the PMI calculation.

backengine ()

This method drives the backengine code.

property data

Query for the field data.

expectedData ()

Query for the data expected by the Interactor.

property path_to_executable

Query the path to the feff executable.

providedData ()
Query for the data provided by the Interactor.

saveH5 ()

property working_directory
Query the working directory

SimEx.Calculators.GenesisPhotonSource module

module GenesisPhotonSource Module that holds the GenesisPhotonSource class.

class SimEx.Calculators.GenesisPhotonSource.**GenesisPhotonSource** (*parameters=None, in-put_path=None, out-put_path=None*)

Bases: *SimEx.Calculators.AbstractPhotonSource.AbstractPhotonSource*

Class GenesisPhotonSource Representing a x-ray free electron laser photon source using the Genesis backend.

:param parameters : Photon source parameters. :type parameters: dict

Parameters

- **input_path** (*str*) – The path to the input data for the photon source.
- **output_path** – The path where to save output data.

Note input_path Accepts a native genesis beam file or openPMD conform hdf5.

backendengine ()
Method to call the backendengine for the calculator. To be implemented on the derived classes.

property data
Query for the field data.

saveH5 ()

SimEx.Calculators.IdealPhotonDetector module

module IdealPhotonDetector Module that holds the IdealPhotonDetector class.

class SimEx.Calculators.IdealPhotonDetector.**IdealPhotonDetector** (*parameters=None, in-put_path=None, out-put_path=None*)

Bases: *SimEx.Calculators.AbstractPhotonDetector.AbstractPhotonDetector*

Class IdealPhotonDetector Class representing an ideal photon detector.

This is just a dummy calculator that copies input data to output data.

Parameters

- **parameters** (*dict*) – Parameters for the Detector simulation.
- **input_path** (*str*) – Path to input files for the detector simulation.
- **output_path** (*str*) – Path where output is stored.

backengine ()
This method drives the backengine code.

property data
Query for the field data.

expectedData ()
Query for the data expected by the Interactor.

providedData ()
Query for the data provided by the Interactor.

saveH5 ()

SimEx.Calculators.PlasmaXRTSCalculator module

module PlasmaXRTSCalculator Module that holds the PlasmaXRTSCalculator class.

class SimEx.Calculators.PlasmaXRTSCalculator.**PlasmaXRTSCalculator** (*parameters=None, in-put_path=None, out-put_path=None*)

Bases: *SimEx.Calculators.AbstractPhotonDiffraction.AbstractPhotonDiffraction*

Class PlasmaXRTSCalculator Represents a plasma x-ray Thomson scattering calculation.

Parameters

- **parameters** (*PlasmaXRTSCalculatorParameters*) – Parameters for the PlasmaXRTSCalculator.
- **input_path** (*str*) – Path to the input data for this calculator.
- **output_path** (*str, default 'xrts.h5'*) – Where to write output data generated by this calculator.

backengine ()
This method drives the backengine xrts.

property data
Query for the field data.

expectedData ()
Query for the data expected by the Diffraction.

providedData ()
Query for the data provided by the Diffraction.

saveH5 ()
Method to save the data to a file.

property static_data
Query for the static data.

SimEx.Calculators.PlasmaXRTSCalculator.**checkAndSetParameters** (*parameters*)
Utility to check if the parameters dictionary is ok .

Parameters parameters (*AbstractCalculatorParameters*) – The parameters to check.

SimEx.Calculators.PlasmaXRTSCalculator.**extractDate** (*pattern_string, text*)
Workhorse function to get a pattern from text using a regular expression. :param pattern_string: The regex pattern to find. :type pattern_string: str (argument to re.compile)

Parameters `text (str)` – The string from which to extract the date.
:~return The date. :~rtype: float

SimEx.Calculators.S2EReconstruction module

module S2EReconstruction Module that holds the S2EReconstruction class.

class `SimEx.Calculators.S2EReconstruction.S2EReconstruction` (`parameters=None`,
`input_path=None`,
`output_path=None`)

Bases: `SimEx.Calculators.AbstractPhotonAnalyzer.AbstractPhotonAnalyzer`

Class S2EReconstruction Class representing photon data analysis for electron density reconstruction from 2D diffraction patterns.

Wraps the EMC orientation module and the DM phasing module.

Parameters

- **parameters** (`dict`) – The parameters for the reconstruction.
- **input_path** (`str`) – Path for input data.
- **output_path** (`str`) – Path where to write output to.

Example parameters `parameters={'EMC_Parameters' : EMCOrientationParameters(), 'DM_Parameters' : DMPhasingParameters()} # Use default parameters for EMC and DM.`

backengine ()
Run the EMC and DM backengine executables.

property data
Query for the field data.

expectedData ()
Query for the data expected by the Analyzer.

providedData ()
Query for the data provided by the Analyzer.

saveH5 ()

SimEx.Calculators.SingFELPhotonDiffractor module

module SingFELPhotonDiffractor Module that holds the SingFELPhotonDiffractor class.

class `SimEx.Calculators.SingFELPhotonDiffractor.SingFELPhotonDiffractor` (`parameters=None`,
`in-`
`put_path=None`,
`out-`
`put_path=None`)

Bases: `SimEx.Calculators.AbstractPhotonDiffractor.AbstractPhotonDiffractor`

Class SingFELPhotonDiffractor Representing scattering from a molecular sample into a detector plane.

Parameters

- **parameters** (*dict* || *SingFELPhotonDiffractorParameters*) – Parameters of the calculation (not data).
- **input_path** (*str*) – Path to hdf5 file holding the input data.
- **output_path** (*str*) – Path to hdf5 file for output.

backengine ()

This method drives the backengine singFEL.

computeNTasks ()

property data

Query for the field data.

expectedData ()

Query for the data expected by the Diffractor.

providedData ()

Query for the data provided by the Diffractor.

saveH5 ()

SimEx.Calculators.XCSITPhotonDetector module

module XCSITPhotonDetector Hosts the XCSITPhotonDetector class.

class SimEx.Calculators.XCSITPhotonDetector.**XCSITPhotonDetector** (*parameters=None, in-put_path=None, out-put_path=None*)

Bases: *SimEx.Calculators.AbstractPhotonDetector.AbstractPhotonDetector*

Class XCSITPhotonDetector Wraps detector simulations with XCSIT.

Parameters parameters – Parameters of the calculator such as the type of detector :type parameters: XCSITPhotonDetectorParameters

Parameters

- **input_path** (*str*) – Path to the hdf5 file holding the input data.
- **output_path** (*str*) – Path pointing to the path for output

backengine ()

Executes the simulation of the particle and charge simulation.

expectedData ()

Query for the data fields expected by this calculator.

getChargeData ()

getInteractionData ()

getPhotonData ()

property input_path

Query for the input file path(s).

property output_path

Query for the output file path(s).

providedData ()
Query for the data fields provided by this calculator.

saveH5 ()
Save the results in a file

SimEx.Calculators.XFELWavePropagator module

module WavePropagator Module that holds the WavePropagator class.

class SimEx.Calculators.WavePropagator.**WavePropagator** (*parameters=None, input_path=None, output_path=None*)

Bases: *SimEx.Calculators.AbstractPhotonPropagator.AbstractPhotonPropagator*

Class WavePropagator Represents coherent wavefront propagation using the WPG wrapper for SWR.

Parameters

- **parameters** (*WavePropagatorParameters instance.*) – Parameters for the photon propagation.
- **input_path** (*str, default 'FELsource/'*) – Location of input data for photon propagation.
- **output_path** (*str, default 'prop/'*) – Location of propagation output data.

backengine ()
Starts WPG simulations in parallel in a subprocess

computeNTasks ()

property data
Query for the field data.

saveH5 ()

SimEx.Calculators.XFELPhotonSource module

module XFELPhotonSource Module that holds the XFELPhotonSource class.

class SimEx.Calculators.XFELPhotonSource.**XFELPhotonSource** (*parameters=None, input_path=None, output_path=None*)

Bases: *SimEx.Calculators.AbstractPhotonSource.AbstractPhotonSource*

Class XFELPhotonSource Class representing a x-ray free electron laser photon source.

:param parameters : Photon source parameters. :type parameters: dict

Parameters

- **input_path** (*str, default FELsource_in.h5*) – The path to the input data for the photon source.
- **output_path** – The path where to save output data.

backengine ()
Method to call the backengine for the calculator. To be implemented on the derived classes.

property data

Query for the field data.

saveH5 ()**SimEx.Calculators.XMDYNDemoPhotonMatterInteractor module**

module XMDYNDemoPhotonMatterInteractor Module that holds the XMDYNDemoPhotonMatter-Interactor class.

class SimEx.Calculators.XMDYNDemoPhotonMatterInteractor.**PMIDemo**

Bases: `object`**f_dbase_zq2id** (*a_z*, *a_q*)**f_dbase_setup** ()**f_init_random** ()**f_load_pulse** (*a_prop_out*)**f_load_sample** (*sample_path*)**f_load_snp_content** (*a_fp*, *a_snp*)**f_load_snp_from_dir** (*path_to_snapshot*)

Load xmdyn output from an xmdyn directory.

Parameters

- **path** (*str*) – The directory path to xmdyn output.
- **snapshot_index** (*int*) – Which snapshot to load.

Returns The snapshot data.**Return type** `dict`**f_load_snp_XXX** (*a_real*, *a_snp*)**f_num_snp_XXX** (*all_real*)**f_rotate_sample** ()**f_s2e_setup** ()**f_save_data** (*dset*, *data*)**f_save_info** ()**f_save_snp** (*a_snp*)**f_system_setup** ()**f_time_evolution** ()

class SimEx.Calculators.XMDYNDemoPhotonMatterInteractor.**XMDYNDemoPhotonMatterInteractor** (*par*

in-
put,
out-
put,
sam
ple,

Bases: `SimEx.Calculators.AbstractPhotonInteractor`, `AbstractPhotonInteractor`

Class XMDYNDemoPhotonMatterInteractor Interface class for photon-matter interaction calculations using the demo version of the XMDYN code.

Parameters

- **parameters** (*dict*) – Parameters that govern the PMI calculation.
- **input_path** (*str*) – Location of data needed by the PMI calculation (Laser source wavefront data).
- **output_path** (*str*) – Where to store the data generated by the PMI calculation.
- **sample_path** (*str*) – Location of the sample/target geometry file. Can be either a simS2E sample file or a pdb file. Specifying a pdb will first check if it's present in a database, if not, it will issue a query for the basename of the file to the RCSB protein data bank.

backengine ()

This method drives the backengine code.

property data

Query for the field data.

expectedData ()

Query for the data expected by the Interactor.

providedData ()

Query for the data provided by the Interactor.

saveH5 ()

```
SimEx.Calculators.XMDYNDemoPhotonMatterInteractor.f_eval_disp(a_snp, a_r0,  
                                                             a_sample)
```

```
SimEx.Calculators.XMDYNDemoPhotonMatterInteractor.f_eval_numE(a_snp, a_sample)
```

```
SimEx.Calculators.XMDYNDemoPhotonMatterInteractor.f_h5_out2in(src, dest, *args)
```

```
SimEx.Calculators.XMDYNDemoPhotonMatterInteractor.f_hdf5_simple_read(self,  
                                                                       a_file,  
                                                                       a_dataset)
```

```
SimEx.Calculators.XMDYNDemoPhotonMatterInteractor.f_md_step(r, v, m, dt)
```

```
SimEx.Calculators.XMDYNDemoPhotonMatterInteractor.f_pmi_diagnostics_help()
```

```
SimEx.Calculators.XMDYNDemoPhotonMatterInteractor.load_ff_database()
```

```
SimEx.Calculators.XMDYNDemoPhotonMatterInteractor.s2e_gen_randrot_quat(quat,  
                                                                           rot-  
                                                                           mat)
```

```
SimEx.Calculators.XMDYNDemoPhotonMatterInteractor.s2e_rand_orient(r, mat)
```

SimEx.Calculators.XMDYNPhotonMatterInteractor module

module XMDYNPhotonMatterInteractor Module that holds the XMDYNPhotonMatterInteractor class.

class `SimEx.Calculators.XMDYNPhotonMatterInteractor.XMDYNPhotonMatterInteractor` (*parameters=None*, *input_path=None*, *output_path=None*, *sample_path=None*, *seed=1*, *root_path=None*)

Bases: `SimEx.Calculators.AbstractPhotonInteractor.AbstractPhotonInteractor`

Class XMDYNPhotonMatterInteractor Wrapper class for photon-matter interaction calculations using the XMDYN code.

Parameters

- **parameters** (*dict*) – Parameters that govern the PMI calculation.
- **input_path** (*str*) – Location of data needed by the PMI calculation (Laser source wavefront data).
- **output_path** (*str*) – Where to store the data generated by the PMI calculation.
- **sample_path** (*str*) – Location of the sample/target geometry file. Can be either a simS2E sample file or a pdb file. Specifying a pdb will first check if it's present in a database, if not, it will issue a query for the basename of the file to the RCSB protein data bank.
- **root_path** (*str*) – Path to a root directory from which to restart a (previously failed) simulation.

backengine ()

This method drives the backengine code.

property data

Query for the field data.

expectedData ()

Query for the data expected by the Interactor.

load_snp_from_dir (*path_to_snapshot*)

Load xmdyn output from an xmdyn directory.

Parameters

- **path** (*str*) – The directory path to xmdyn output.
- **snapshot_index** (*int*) – Which snapshot to load.

Returns The snapshot data.

Return type `dict`

property parameters

Query the calculator parameters.

Returns The parameters of this Calculator.

providedData ()

Query for the data provided by the Interactor.

property root_path

Get the path to the restart data.

property sample_path

Get the path to the sample geometry file.

saveH5 ()

setup_hierarchy (*h5_handle*)

Create all datagroups in the output hdf5 file.

Parameters *h5_handle* (*h5py.File*) – File handle to writable hdf5 file.

`SimEx.Calculators.XMDYNPhotonMatterInteractor.h5_out2in` (*src, dest, *args*)

SimEx.Parameters package

Submodules

SimEx.Parameters.AbstractCalculatorParameters module

module AbstractCalculatorParameters Hosting the abstract base class for all parameter classes.

class `SimEx.Parameters.AbstractCalculatorParameters.AbstractCalculatorParameters` (**kwargs)

Bases: `SimEx.AbstractBaseClass.AbstractBaseClass`

Abstract class for all calculator parameters.

Parameters ****kwargs** – key=value pairs for calculator specific parameters.

property `cpus_per_task`

Query for the number of cpus per task.

property `forced_mpi_command`

Query for the number of cpus per task.

property `gpus_per_task`

Query for the number of gpus per task.

property `nodes_per_task`

Query for the number of nodes per task.

SimEx.Parameters.AbstractPhotonDiffractorParameters module

module AbstractPhotonDiffractorParameters Hosts the abstract base class for all PhotonDiffractors.

class `SimEx.Parameters.AbstractPhotonDiffractorParameters.AbstractPhotonDiffractorParameters`

Bases: `SimEx.Parameters.AbstractCalculatorParameters.AbstractCalculatorParameters`

Class AbstractPhotonDiffractorParameters Abstract base class for all PhotonDiffractors.

Parameters

- **sample** (*str*) – Location of file that contains the sample definition (pdb or crystfel format)

- **uniform_rotation** (*bool*, *default True*) – Whether to perform uniform sampling of rotation space.
- **number_of_diffraction_patterns** (*int*, *default 1*) – Number of diffraction patterns to calculate from each trajectory.
- **beam_parameters** (*str*) – Path of the beam parameter file.
- **detector_geometry** (*DetectorGeometry*) – The detector geometry for the simulated scattering experiment.
- **kwargs** – Key-value pairs to pass to the parent class.

property beam_parameters

Query for the 'beam_parameters' parameter.

property detector_geometry

Query for the 'detector_geometry' parameter.

property number_of_diffraction_patterns

Query for the 'number_of_diffraction_patterns_file' parameter.

property sample

Query the 'sample' parameter.

property uniform_rotation

Query for the 'uniform_rotation' parameter.

SimEx.Parameters.CrystFELPhotonDiffractorParameters module

module CrystFELPhotonDiffractorParameters Module that holds the CrystFELPhotonDiffractorParameters class.

class SimEx.Parameters.CrystFELPhotonDiffractorParameters.**CrystFELPhotonDiffractorParameters**

Bases: *SimEx.Parameters.AbstractPhotonDiffractorParameters*,
AbstractPhotonDiffractorParameters

Class CrystFELPhotonDiffractorParameters Encapsulates parameters for the CrystFELPhotonDiffractor.

Parameters

- **sample** (*str*) – Location of file that contains the sample definition (pdb or crystfel format)
- **uniform_rotation** (*bool*, *default True*) – Whether to perform uniform sampling of rotation space.
- **powder** (*bool*) – Whether to sum all patterns to generate a simulated powder diffraction pattern.
- **intensities_file** (*str*) – Location of file that contains intensities and phases at reciprocal lattice points. See CrystFEL documentation for more info. Default: Constant intensities and phases across entire sample.
- **crystal_size_min** (*PhysicalQuantity with unit of length (meter)*) – Minimum crystal size.
- **crystal_size_max** (*PhysicalQuantity with unit of length (meter)*) – Maximum crystal size.
- **poissonize** (*bool*) – Whether to add Poisson noise to pixel values.
- **number_of_background_photons** (*int*) – Add this number of Poisson distributed photons uniformly over the detector surface (default 0).
- **number_of_diffraction_patterns** (*int*, *default 1*) – Number of diffraction patterns to calculate from each trajectory.

- **suppress_fringes** (*bool*) – Whether to suppress subsidiary maxima beyond first minimum of the shape transform (default False).
- **beam_parameters** (*str*) – Path of the beam parameter file.
- **detector_geometry** (*str*) – Path of the beam geometry file.
- **kwargs** – Key-value pairs to pass to the parent class.

property crystal_size_max

Query the ‘crystal_size_max’ parameter.

property crystal_size_min

Query the ‘crystal_size_min’ parameter.

property intensities_file

Query the ‘intensities_file’ parameter.

property number_of_background_photons

Query the ‘number_of_background_photons’ parameter.

property poissonize

Query the ‘poissonize’ parameter.

property powder

Query the ‘powder’ parameter.

property suppress_fringes

Query the ‘suppress_fringes’ parameter.

SimEx.Parameters.DMPhasingParameters module

module DMPhasingParameters Module that holds the DMPhasingParameters class.

```
class SimEx.Parameters.DMPhasingParameters.DMPhasingParameters (number_of_trials=None,
                                                                    num-
                                                                    ber_of_iterations=None,
                                                                    averag-
                                                                    ing_start=None,
                                                                    leash=None,
                                                                    num-
                                                                    ber_of_shrink_cycles=None,
                                                                    parame-
                                                                    ters_dictionary=None,
                                                                    **kwargs)
```

Bases: *SimEx.Parameters.AbstractCalculatorParameters*
AbstractCalculatorParameters

Class DMPhasingParameters Encapsulates parameters for the DMPhasing analyzer.

Parameters

- **number_of_trials** (*int>0, default 500*) – How many trials to run in each iteration.
- **number_of_iterations** (*int>0, default 50*) – Maximum number of DM iterations.
- **averaging_start** – Start averaging after this many runs.
- **leash** (*float>0, default 0.2*) – DM leash parameter.

- **number_of_shrink_cycles** (*int*>0, *default 10*) – DM shrink cycles.

property averaging_start

Query for the ‘averaging_start’ parameter.

property leash

Query for the ‘leash’ parameter.

property number_of_iterations

Query for the ‘number_of_iterations’ parameter.

property number_of_shrink_cycles

Query for the ‘number_of_shrink_cycles’ parameter.

property number_of_trials

Query for the ‘number_of_trials’ parameter.

SimEx.Parameters.DetectorGeometry module

module DetectorGeometry Module holding the DetectorGeometry class.

class SimEx.Parameters.DetectorGeometry.**DetectorGeometry** (*panels=None*,
***kwargs*)

Bases: *SimEx.Parameters.AbstractCalculatorParameters*
AbstractCalculatorParameters

Class representing the detector geometry.

Constructor of the DetectorGeometry class.

Parameters

- **panels** (*list, tuple, or instance of DetectorPanel*) – Single or list of detector panels that constitute the detector.
- **kwargs** (*dict*) – Key-value pairs to be passed to the parent class constructor.

property panels

serialize (*stream=None, caller=None*)

Serialize the geometry.

Parameters stream (*File like object.*) – The stream to write the serialized geometry to (default sys.stdout).

class SimEx.Parameters.DetectorGeometry.**DetectorPanel** (*ranges=None*,
pixel_size=None, energy_response=None,
photon_response=None, distance_from_interaction_plane=None,
distance_offset=None,
fast_scan_xyz=None,
slow_scan_xyz=None,
corners=None, saturation_adu=None, mask=None,
good_bit_mask=None,
bad_bit_mask=None, saturation_map=None, badregion_flag=None,
***kwargs*)

Bases: SimEx.AbstractBaseClass.*AbstractBaseClass*

Class DetectorPanel Represents one detector panel (contiguous array of pixels, i.e. not separated by gaps).

Parameters **ranges** (*Dictionary*) – The minimum and maximum values pixel numbers on the respective transverse axis.

“:example ranges: {“fast_scan_min” : 11, “fast_scan_max” : 20, “slow_scan_min” : 1, “fast_scan_max” : 20} # First axis from 11 to 20 and second axis from 1 to 20.”

Parameters

- **pixel_size** (*PhysicalQuantity with unit meter.*) – The physical size of the pixel (assuming quadratic shape) (SI units).
- **energy_response** (*PhysicalQuantity with unit 1/eV (adu_per_eV)*) – Number of detector units (ADU) arising from one eV.
- **photon_response** (*float*) – Number of detector units (ADU) arising from one photon.
- **distance_from_interaction_plane** (*PhysicalQuantity with unit meter.*) – Distance in z of this panel from the plane of interaction (transverse plane that contains the sample).
- **distance_offset** (*PhysicalQuantity with unit meter.*) – Offset from distance_from_interaction_plane.
- **fast_scan_xyz** (*str*) – Formula that lab frame coordinates to panel axes.
- **slow_scan_xyz** (*str*) – Formula that lab frame coordinates to panel axes.
- **corners** (*dict*) – [x,y] coordinates of lower left pixel of this panel in the globale detector geometry.
- **saturation_adu** (*float.*) – Saturation level for this panel.
- **mask** (*numpy.array of same shape as panel data.*) – Mask to apply to this panel.
- **good_bit_mask** – Bitmask indicating the good pixels
- **bad_bit_mask** – Bitmask indicating the bad pixels.
- **saturation_map** (*numpy.array of same shape as panel data.*) – Pixel map indicating saturated pixels.
- **badregion_flag** (*bool*) – Flag to indicate this panel as a bad region.

Example corners corners={“x” : -10, “y” : 10 }

property bad_bit_mask

Query the panel bad_bit_mask.

property badregion_flag

Query the panel badregion_flag.

property corners

Query the panel cornes.

property distance_from_interaction_plane

Query the panel distance_from_interaction_plane.

property distance_offset

Query the panel distance_offset.

property energy_response

Query the panel energy_response.

property fast_scan_xyz
Query the panel fast_scan_xyz.

property good_bit_mask
Query the panel good_bit_mask.

property mask
Query the panel mask.

property photon_response
Query the panel photon_response.

property pixel_size
Query the panel pixel_size.

property ranges
Query the panel ranges.

property saturation_adu
Query the panel saturation_adu.

property saturation_map
Query the panel saturation_map.

property slow_scan_xyz
Query the panel slow_scan_xyz.

SimEx.Parameters.DetectorGeometry.**detectorGeometryFromFile** (*input_file*)

SimEx.Parameters.EMCOrientationParameters module

module EMCOrientationParameters Hosts the EMCOrientationParameters class.

class SimEx.Parameters.EMCOrientationParameters.**EMCOrientationParameters** (*initial_number_of_quaternions*, *max_number_of_quaternions*, *max_number_of_iterations*, *min_error=None*, *beam-stop=None*, *de-tailed_output=None*, *parameters_dictionary=None*, ***kwargs*)

Bases: *SimEx.Parameters.AbstractCalculatorParameters*, *AbstractCalculatorParameters*

Class EMCOrientationParameters Class representing parameters for the EMCOrientation analyzer.

Parameters

- **initial_number_of_quaternions** (*int* ($0 < n < 10$), *default 1*) – Number of quaternions to start the EMC algorithm.
- **max_number_of_quaternions** (*int* ($initial_number_of_quaternions < n \leq 10$), *default initial_number_of_quaternions + 1*) – Maximum number of quaternions for the EMC algorithm.

- **min_error** (*float* (>0), *default* $1.e-6$) – Relative convergence criterion (Go to next quaternion is relative error gets below this value.)
- **max_number_of_iterations** (*int* (>0), *default* 100) – Stop the EMC algorithm after this number of iterations.
- **beamstop** (*bool*, *default* *True*) – Whether to apply a “center + strip” beamstop to the pixel map.
- **detailed_output** (*bool*, *default* *True*) – Whether to write detailed info to log.

property beamstop

Query for the ‘beamstop’ parameter.

property detailed_output

Query for the ‘detailed_output’ parameter.

property initial_number_of_quaternions

Query for the ‘initial_number_of_quaternions’ parameter.

property max_number_of_iterations

Query for the ‘max_number_of_iterations’ parameter.

property max_number_of_quaternions

Query for the ‘max_number_of_quaternions’ parameter.

property min_error

Query for the ‘min_error’ parameter.

SimEx.Parameters.EstherPhotonMatterInteractorParameters module

module EstherPhotonMatterInteractorParameters Hosting the parameter class for the EstherPhotonMatterInteractor.

class SimEx.Parameters.EstherPhotonMatterInteractorParameters.**EstherPhotonMatterInteractor**

Bases: *SimEx.Parameters.AbstractCalculatorParameters*.
AbstractCalculatorParameters

Class EstherPhotonMatterInteractorParameters representing parameters for the Esthe Hydrocode Calculator.

Parameters

- **ablator** (*str*) – The ablating material (“Al” | “CH” | “Diamond” | “Kapton” | “Mylar”)
- **ablator_thickness** – The ablator thickness (micrometers)
- **sample** (*str*) – The sample material (from list of materials)
- **sample_thickness** (*float*) – The sample thickness (micrometers)
- **layer1** (*str*) – The layer1 material (from list of materials)
- **layer2** (*str*) – The layer2 material (from list of materials)
- **window** (*str*) – The window material (LiF | SiO2 | Diamond)
- **window_thickness** (*float*) – The window thickness, if using window (micrometers)
- **laser_pulse** (*str*) – Pulse type (“flat” | “ramp” | “other”)
- **laser_pulse_duration** (*float*) – Pulse duration of the pump laser (ns)
- **laser_wavelength** (*float*) – Laser wavelength (nm)
- **laser_intensity** (*float*) – Laser intensity (TW/cm2)

- **run_time** (*float*) – Simulation run time (ns)
- **delta_time** (*float*) – Time steps resolution (ns)
- **force_passage** (*boolean*) – Expert option to force passage of simulation through minor errors
- **without_therm_conduc** (*boolean*) – Expert option to use without thermal conductivity options
- **rad_transfer** (*boolean*) – Expert option to use radiative transfer

property ablator

Query for the ablator type.

property ablator_thickness

Query for the ablator thickness.

checkConsistency ()**property delta_time**

Query for simulation time resolution (delta t ns)

property laser_intensity

Query for laser intensity

property laser_pulse

Query for laser pulse type

property laser_pulse_duration

Query for laser pulse duration

property laser_wavelength

Query for the laser wavelength type.

property layer1

Query for the layer1 type.

property layer1_thickness

Query for the layer1 thickness type.

property layer2

Query for the layer2 type.

property layer2_thickness

Query for the layer2 thickness type.

property number_of_layers

Query for the number of layers.

property run_time

Query for simulation run time

property sample

Query for the sample type.

property sample_thickness

Query for the sample thickness type.

property window

Query for the window type.

property window_thickness

Query for the window thickness type.

`SimEx.Parameters.EstherPhotonMatterInteractorParameters.checkAndSetAblator` (*ablator*)
Utility to check if the ablator exists in the EOS database.

Parameters `ablator` – The ablator material to check.

Returns The ablator choice after being checked.

Raises `ValueError` – ablator not in [“CH”, “Al”, “Diamond”, “Mylar”, “Kapton”].

`SimEx.Parameters.EstherPhotonMatterInteractorParameters.checkAndSetAblatorThickness` (*ablator_thi*)
Utility to check that the ablator thickness is > 5 um and < 100 um

`SimEx.Parameters.EstherPhotonMatterInteractorParameters.checkAndSetDeltaTime` (*delta_time*)
Utility for checking the simulation delta time (resolution) is valid

`SimEx.Parameters.EstherPhotonMatterInteractorParameters.checkAndSetLaserIntensity` (*laser_intens*)
Utility to check that the laser intensity is valid.

`SimEx.Parameters.EstherPhotonMatterInteractorParameters.checkAndSetLaserPulse` (*laser_pulse*)
Utility to check that the laser pulse type is correct.

`SimEx.Parameters.EstherPhotonMatterInteractorParameters.checkAndSetLaserPulseDuration` (*laser_pu*)
Utility to check that the laser pulse duration is valid.

`SimEx.Parameters.EstherPhotonMatterInteractorParameters.checkAndSetLaserWavelength` (*laser_wave*)
Utility to check that the laser wavelength is correct.

`SimEx.Parameters.EstherPhotonMatterInteractorParameters.checkAndSetLayer1` (*layer1*)
Utility to check if the layer1 is in the list of known EOS materials

`SimEx.Parameters.EstherPhotonMatterInteractorParameters.checkAndSetLayer1Thickness` (*layer1_thic*)
Utility to check that the layer1 thickness is in permitted range set by Esther.

`SimEx.Parameters.EstherPhotonMatterInteractorParameters.checkAndSetLayer2` (*layer2*)
Utility to check if the layer2 is in the list of known EOS materials

`SimEx.Parameters.EstherPhotonMatterInteractorParameters.checkAndSetLayer2Thickness` (*layer2_thic*)
Utility to check that the layer2 thickness is in permitted range set by Esther.

`SimEx.Parameters.EstherPhotonMatterInteractorParameters.checkAndSetNumberOfLayers` (*number_of_l*)
Utility to check if the number of layers is reasonable.

Parameters `number_of_layers` – The number of layers to check

Returns Checked number of layers

Raises `ValueError` – not (1 < number_of_layers <= 4)

`SimEx.Parameters.EstherPhotonMatterInteractorParameters.checkAndSetRunTime` (*run_time*)
Utility for checking the simulation run time is valid

`SimEx.Parameters.EstherPhotonMatterInteractorParameters.checkAndSetSample` (*sample*)
Utility to check if the sample is in the list of known EOS materials

`SimEx.Parameters.EstherPhotonMatterInteractorParameters.checkAndSetSampleThickness` (*sample_thi*)
Utility to check that the sample thickness is in permitted range set by Esther.

`SimEx.Parameters.EstherPhotonMatterInteractorParameters.checkAndSetWindow` (*window*)
Utility to check that the window exists in the EOS database.

`SimEx.Parameters.EstherPhotonMatterInteractorParameters.checkAndSetWindowThickness` (*window_th*)
Utility to check that the window thickness is > 1 um and < 500 um

SimEx.Parameters.FEFFPhotonMatterInteractorParameters module

module Holds the FEFFPhotonMatterInteractorParameters class.

class SimEx.Parameters.FEFFPhotonMatterInteractorParameters.FEFFPhotonMatterInteractorParameters

Bases: *SimEx.Parameters.AbstractCalculatorParameters*,
AbstractCalculatorParameters

Class FEFFPhotonMatterInteractorParameters Interface class for photon-matter interaction calculations using the FEFF code.

Parameters

- **atoms** (*list // tuple*) – The atomic structure (Atom coordinates ([x,y,z] in Angstrom), element symbol, and potential index). If no potential index is given, all atoms of the same species will be assigned the default potential. The scattering atom must have the potential index 0.
- **potentials** (*list*) – The potentials to use.
- **edge** (*str*) – The edge to calculate (K, L1, L2, M1, M2, M3, ...). Default 'K'.
- **amplitude_reduction_factor** (*float*) – The amplitude reduction factor. Default 1.0
- **effective_path_distance** (*float*) – The maximum effective (half-path) distance in Angstrom. Translates to rpath parameter in feff.inp. Default 2.2 times nearest neighbor distance.

Example atoms ([[0.0, 0.0, 0.0], 'Cu', 0], [[0.0, 1.0, 1.2], 'O', 1], ...)

property amplitude_reduction_factor
Query method for amplitude_reduction_factor

property atoms
Query method for atoms

property edge
Query method for edge

property effective_path_distance
Query method for effective_path_distance

finalize()
Finalize the parameters. Check if all parameters are internally consistent.

property finalized
Query the finalization status.

property potentials
Query method for potentials

SimEx.Parameters.PhotonBeamParameters module

module PhotonBeamParameters Contains the PhotonBeamParameters class and associated functions.

class SimEx.Parameters.PhotonBeamParameters.**PhotonBeamParameters** (*photon_energy*,
beam_diameter_fwhm,
pulse_energy,
pho-
ton_energy_relative_bandwidth=None,
diver-
gence=None,
pho-
ton_energy_spectrum_type=None,
***kwargs*)

Bases: *SimEx.Parameters.AbstractCalculatorParameters*,
AbstractCalculatorParameters

Class PhotonBeamParameters Encapsulates the parameters of a photon beam.

Parameters

- **photon_energy** (*float*) – The mean photon energy in units of electronvolts (eV).
- **photon_energy_relative_bandwidth** (*float* (>0.0)) – The relative energy bandwidth
- **beam_diameter_fwhm** (*float*) – Beam diameter in units of metre (m).
- **pulse_energy** (*float*) – Total energy of the pulse in units of Joule (J).
- **divergence** (*float* ($0 < divergence < 2\pi$)) – Beam divergence angle in units of radian (rad).
- **photon_energy_spectrum_type** (*float*) – Type of energy spectrum (“SASE” | “tophat” | “twocolour”, default “SASE”).
- **kwargs** (*dict*) – Key-value pairs to be passed to the parent class constructor.

property beam_diameter_fwhm
 Query the ‘beam_diameter_fwhm’ parameter.

property divergence
 Query the ‘divergence’ parameter.

property photon_energy
 Query the ‘photon_energy’ parameter.

property photon_energy_relative_bandwidth
 Query the ‘photon_energy_relative_bandwidth’ parameter.

property photon_energy_spectrum_type
 Query the ‘photon_energy_spectrum_type’ parameter.

property pulse_energy
 Query the ‘pulse_energy’ parameter.

serialize (*stream=<_io.TextIOWrapper name='<stdout>' mode='w' encoding='UTF-8'>*)
 Serialize the object (write to a stream)

Parameters stream (*str* || *file*) – The stream to write to (default sys.stdout)

SimEx.Parameters.PhotonBeamParameters.**propToBeamParameters** (*prop_output_path*)
 Utility to setup a PhotonBeamParameters instance from propagation output.

SimEx.Parameters.PhotonMatterInteractorParameters module

module PhotonMatterInteractorParameters Module that holds the PhotonMatterInteractorParameter class.

class SimEx.Parameters.PhotonMatterInteractorParameters.**PhotonMatterInteractorParameters** (ro

Bases: *SimEx.Parameters.AbstractCalculatorParameters*, *AbstractCalculatorParameters*

Class PhotonMatterInteractorParameters Encapsulates parameters for the PhotonMatterInteractor calculator.

Parameters

- **rotation** – Rotation to apply to the sample atoms' positions (Default: no rotation).
- **calculate_Compton** (*bool, default False*) – Whether to calculate incoherent (Compton) scattering.
- **number_of_trajectories** (*int, default 1*) – Number of trajectories to simulate.
- **beam_parameters** (*PhotonBeamParameters*) – Parameters of the photon beam.
- **parameters_dictionary** (*dict*) – A legacy parameters dictionary (Default: None).

property beam_parameters

Query for the 'beam_parameters' parameter.

property calculate_Compton

Query for the 'calculate_Compton' parameter.

property number_of_trajectories

Query for the 'number_of_trajectories_file' parameter.

property rotation

Query for the 'rotation' parameter.

SimEx.Parameters.PlasmaXRTSCalculatorParameters module

module PlasmaXRTSCalculatorParameters Module that holds the PlasmaXRTSCalculatorParameters class.

```
class SimEx.Parameters.PlasmaXRTSCalculatorParameters.PlasmaXRTSCalculatorParameters (element.
    photon_ene
    scat-
    ter-
    ing_ang
    elec-
    tron_tem
    elec-
    tron_den
    ion_tem
    ion_cha
    mass_de
    de-
    bye_tem
    band_ga
    en-
    ergy_ra
    model_S
    model_S
    model_S
    model_S
    model_I
    model_I
    lfc=Non
    Sbf_nor
    source_
    source_
    **kwargs
```

Bases: `SimEx.Parameters.AbstractCalculatorParameters`, `AbstractCalculatorParameters`

Class PlasmaXRTSCalculatorParameters Encapsulates parameters for the plasma x-ray Thomson scattering calculator.

Parameters

- **elements** (*list* `[[element symbol, stoichiometric number, charge], ..]`, default `None`) – The chemical elements in the scattering target.
- **photon_energy** (*float*) – The central energy of incoming x-ray photons.
- **scattering_angle** (*float*) – The scattering angle.
- **electron_temperature** (*float*) – The temperature of the electron subsystems (units of eV).
- **electron_density** (*float*) – The electron number density (units of 1/cm³)
- **ion_temperature** (*float*) – The temperature of the ion subsystem (units of eV).
- **ion_charge** (*float*) – The average ion charge (units of elementary charge *e*).
- **mass_density** (*float*) – The mass density of the target (units of g/cm³).
- **debye_temperature** (*float*) – The Debye temperature (units of eV).
- **band_gap** (*float*, default `0`) – The band gap of the target (units of eV).
- **energy_range** (*dict*, default `0`) – The energy range over which to calculate the scattering spectrum.

- **model_Sii** (*str* ('DH' || 'OCP' || 'SOCP' || 'SOCPN') || *float*, *default* 'DH') – The model to use for the ion-ion structure factor.
- **model_See** (*str* ('RPA' || 'BMA' || 'BMA+sLFC'), *default* 'RPA') – The model of the dynamic (high frequency) part of the electron-electron structure factor.
- **model_Sbf** (*str* ('IA' || 'FA'), *default* 'IA') – The model for the bound-free structure factor.
- **model_IPL** (*str* ('SP' || 'EK') || *float*, *default* 'SP') – Model for ionization potential lowering.
- **model_Mix** (*str*, *default* None) – The model to use for mixing (of species).
- **lfc** (*float*, *default* 0.0) – The local field correction to use.
- **Sbf_norm** (*str* || *float*, *default* None) – How to normalize the bound-free structure factor.
- **source_spectrum** (*str*, *default* None) – Path to a file holding the x-ray probe energy spectrum.
- **source_spectrum_fwhm** (*float*) – The x-ray probe energy spectrum fwhm.

Example elements [['B', 1, 2], ['N', 1, 2]] for Boron-Nitride with both B and N two fold ionized (ion average).

Example elements [['C', 1, 4], ['H', 1, -1]] for Plastic with both four-fold ionized C and ionization of H calculated so that the given average ion charge comes out correct.

Example energy_range energy_range={'min' -100.0, 'max' 100, 'step' 0.5} to go from -100 eV to 100 eV in steps of 0.5 eV.

Example model_Sii Sii=1.5 to use a fixed value of Sii=1.5

Example model_IPL model_IPL=100.0 # Set the ionization potential difference (lowering) to 100 eV.

property Sbf_norm

Query for the norm of the bound-free structure factor.

property band_gap

Query for the band gap.

property debye_temperature

Query for the Debye temperature.

property electron_density

Query for the electron density.

property electron_temperature

Query for the electron temperature.

property elements

Query for the field data.

property energy_range

Query for the energy range.

property ion_charge

Query for the ion charge.

property ion_temperature

Query for the ion temperature.

property lfc

Query for the local field factor.

property mass_density

Query for the mass density.

property model_IPL

Query for the ionization potential lowering model.

property model_Mix

Query for the mixing model.

property model_Sbf

Query for the bound-free structure factor model.

property model_See

Query for the electron-electron (high-frequency) structure factor model.

property model_Sii

Query for the ion-ion structure factor model.

property photon_energy

Query for the photon energy.

property scattering_angle

Query for the scattering angle.

property source_spectrum

Query for the source spectrum identifier.

property source_spectrum_fwhm

Query for the source spectrum fwhm identifier.

`SimEx.Parameters.PlasmaXRTSCalculatorParameters.checkAndSetBandGap` (*band_gap*)

Utility to check if input is a valid bandgap.

@param band_gap: The bandgap to check. `
type` : double `
default` 0.0. @return : The checked bandgap.

`SimEx.Parameters.PlasmaXRTSCalculatorParameters.checkAndSetDebyeTemperature` (*debye_temperature*)

Utility to check if input is a valid Debye temperature.

@param debye_temperature : The Debye temperature to check. `
type` : double `
default` : 0.0 @return : The checked Debye temperature.

`SimEx.Parameters.PlasmaXRTSCalculatorParameters.checkAndSetDensitiesAndCharge` (*electron_density,*

ion_charge,
mass_density,
el-
e-
ments)

Utility to check input and return a set of consistent electron density, average ion charge, and mass density, if two are given as input.

`SimEx.Parameters.PlasmaXRTSCalculatorParameters.checkAndSetElectronTemperature` (*electron_temperature*)

Utility to check if input is a valid electron temperature.

@param electron_temperature : The electron temperature to check. `
type` : double @return : The checked electron temperature.

`SimEx.Parameters.PlasmaXRTSCalculatorParameters.checkAndSetElements` (*elements*)

Utility to check if input is a valid list of elements.

@param elements: The elements to check.
type elements: list @return: The checked list of elements.

SimEx.Parameters.PlasmaXRTSCalculatorParameters.**checkAndSetEnergyRange** (*energy_range*,
*elec-
tron_density=None*)

Utility to check if the photon energy range is ok. @param energy_range : The range to check.
type dict @return The checked photon energy range. @raise ValueError if not of correct shape.

SimEx.Parameters.PlasmaXRTSCalculatorParameters.**checkAndSetIonTemperature** (*ion_temperature*,
*elec-
tron_temperature=None*)

Utility to check if input is a valid ion temperature.

@param ion_temperature : The ion temperature to check.
type : double
default : Electron temperature. @return : The checked ion temperature.

SimEx.Parameters.PlasmaXRTSCalculatorParameters.**checkAndSetLFC** (*lfc*)

Utility to check if input is a valid local field correction factor.

@param lfc : The lfc to check.
type : double @return : The checked lfc.

SimEx.Parameters.PlasmaXRTSCalculatorParameters.**checkAndSetModelIPL** (*model*)

Utility to check if the model is a valid model for ionization potential lowering.

@param model : The model to check.
type : str or float @return : The checked model @raise ValueError if not a valid IPL model.

SimEx.Parameters.PlasmaXRTSCalculatorParameters.**checkAndSetModelMix** (*model_Mix*)

Utility to check if input is a valid mixing model.

@param model_Mix : The mixing model to check.
type : string @return : The checked mixing model.

SimEx.Parameters.PlasmaXRTSCalculatorParameters.**checkAndSetModelSbf** (*model*)

Utility to check if the model is a valid model for the bound-free (Compton) scattering feature.

@param model : The model to check.
type : str @return : The checked model @raise ValueError if not a string or not a valid Sbf model ('IA', 'HWF')

SimEx.Parameters.PlasmaXRTSCalculatorParameters.**checkAndSetModelSee** (*model*)

Utility to check if the model is a valid model for the high frequency (dynamic) feature.

@param model : The model to check.
type : str @return : The checked model @raise ValueError if not a string or not a valid See0 model ('RPA', 'BMA', 'BMA+sLFC', 'BMA+dLFC', 'LFC', 'Landen')

SimEx.Parameters.PlasmaXRTSCalculatorParameters.**checkAndSetModelSii** (*model*)

Utility to check if the model is a valid model for the Rayleigh (quasistatic) scattering feature.

@param model : The model to check.
type : str @return : The checked model @raise ValueError if not a string or not a valid Sii model ('RPA', 'DH',

SimEx.Parameters.PlasmaXRTSCalculatorParameters.**checkAndSetPhotonEnergy** (*energy*)

Utility to check if the photon energy is correct. @param energy : The energy to check. @return The checked energy.

SimEx.Parameters.PlasmaXRTSCalculatorParameters.**checkAndSetSbfNorm** (*Sbf_norm*)

Utility to check if input is a valid norm of the bound-free structure factor.

@param Sbf_norm : The norm to check.
type : string or double. @return : The checked norm.

SimEx.Parameters.PlasmaXRTSCalculatorParameters.**checkAndSetScatteringAngle** (*angle*)

Utility to check if the scattering angle is in the correct range. @param angle : The angle to check. @return The checked angle. @raise ValueError if not $0 \leq \text{angle} \leq 180$

`SimEx.Parameters.PlasmaXRTSCalculatorParameters.checkAndSetSourceSpectrum(source_spectrum)`
Utility to check sanity of given input for the source spectrum identifier.

@param source_spectrum : The source spectrum identifier to check.
type : str @return : The checked identifier. @raise : TypeError or ValueError if input is not valid.

`SimEx.Parameters.PlasmaXRTSCalculatorParameters.checkAndSetSourceSpectrumFWHM(fwhm)`
Utility to check sanity of given input for the source spectrum full width at half maximum (fwhm).

@param source_spectrum : The value to check.
type : float @return : The checked value. @raise : TypeError or ValueError if input is not valid.

SimEx.Parameters.SingFELPhotonDiffractorParameters module

module SingFELPhotonDiffractorParameters Module that holds the SingFELPhotonDiffractorParameters class.

class `SimEx.Parameters.SingFELPhotonDiffractorParameters.SingFELPhotonDiffractorParameters`

Bases: `SimEx.Parameters.AbstractPhotonDiffractorParameters`
`AbstractPhotonDiffractorParameters`

Class SingFELPhotonDiffractorParameters Class representing parameters for the SingFELPhotonDiffractor calculator.

Parameters

- **calculate_Compton** (*bool*, *default False*) – Whether to calculate incoherent (Compton) scattering.
- **slice_interval** (*int*, *default 100*) – Length of time slice interval to extract from each trajectory.
- **number_of_slices** (*int*, *default 1*) – Number of time slices to read from each trajectory.
- **pmi_start_ID** (*int*, *default 1*) – Identifier for the first pmi trajectory to read in.
- **pmi_stop_ID** (*int*, *default 1*) – Identifier for the last pmi trajectory to read in.

property calculate_Compton
Query for the 'calculate_Compton' parameter.

property number_of_slices
Query for the 'number_of_slices' parameter.

property pmi_start_ID
Query for the 'pmi_start_ID' parameter.

property pmi_stop_ID
Query for the 'pmi_stop_ID' parameter.

property slice_interval
Query for the 'slice_interval' parameter.

SimEx.Parameters.WavePropagatorParameters module

module WavePropagatorParameters Module that holds the WavePropagatorParameters class.

class SimEx.Parameters.WavePropagatorParameters.**WavePropagatorParameters** (*use_opmd=None, beamline=None, **kwargs*)

Bases: *SimEx.Parameters.AbstractCalculatorParameters*
AbstractCalculatorParameters

Class WavePropagatorParameters Class representing parameters for the WavePropagator.

Parameters

- **use_opmd** (*bool, default False*) – Whether to use the openPMD output format.
- **beamline** (*class or module that defines a function*) – The WPG beamline to use in the propagation.

get_beamline, which in turn returns a WPG.Beamline instance.

property beamline
Query for the 'beamline' parameter.

property use_opmd
Query for the 'use_opmd' parameter.

SimEx.Parameters.XCSITPhotonDetectorParameters module

module XCSITPhotonDetectorParameters Hosts the XCSITPhotonDetectorParameters class.

class SimEx.Parameters.XCSITPhotonDetectorParameters.**XCSITPhotonDetectorParameters** (*detector_type, plasma_section, plasma_simulation, point_simulation, patterns=Non*)

Bases: *SimEx.Parameters.AbstractCalculatorParameters*
AbstractCalculatorParameters

Class XCSITPhotonDetectorParameters Encapsulates all parameters for the XCSITPhotonDetector class.

Parameters

- **detector_type** (*str*) – The detector type to simulate (“pnCCD” | “LPD” | “AGIPD” | “AGIPDSPB”).
- **plasma_search_flag** (*str*) – Flag for the plasma search method (“BLANK”).
- **plasma_simulation_flag** (*str*) – Flag for the plasma simulation method (“BLANKPLASMA”).
- **point_simulation_method** (*str*) – Method for the charge point simulation (“FULL” | “FANO” | “LUT” | “BINNING”).
- **patterns** (*(str || int) or iterable over these types.*) – Which patterns to feed into the detector simulation. Default: Use all patterns.

Example patterns patterns=0 # use the first pattern.

Example patterns patterns=range(10) # use the first 10 patterns

Example patterns patterns=[‘0000001’,‘0001001’] # user patterns with Ids ‘0000001’ and ‘0001001’.

property detector_type
:return string containing the detector name

property patterns

Returns The patterns to use in the detector simulation.

property plasma_search_flag
:return string, the plasma search method

property plasma_simulation_flag
:return string, the plasma simulation method

property point_simulation_method
:return string, the charge simulation method

7.11 Data format documentation

7.11.1 FEL source calculations (FAST)

File format: hdf5 (extension .h5)

Data set	Description	Physical unit
/data/arrEhor	horizontal electric field component	$\sqrt{W/mm^2}$
/data/arrEver	vertical electric field component	$\sqrt{W/mm^2}$
/params/Mesh/nSlices	number of time slices	1
/params/Mesh/nx	number of grid points in horizontal dimension (x)	1
/params/Mesh/ny	number of grid points in vertical dimension (y)	s
/params/Mesh/sliceMax	time corresponding to last slice	s
/params/Mesh/sliceMin	time corresponding to first slice	s
/params/Mesh/xMax	x coordinate of last grid point in horizontal dimension	m
/params/Mesh/xMin	x coordinate of first grid point in horizontal dimension	m
/params/Mesh/yMax	y coordinate of last grid point in vertical dimension	m
/params/Mesh/yMin	y coordinate of first grid point in vertical dimension	m
/params/Mesh/zCoord	z coordinate of the wavefront	m
/params/Rx	instantaneous horizontal wavefront radius	m

continues on next page

Table 1 – continued from previous page

Data set	Description	Physical unit
/params/Ry	instantaneous vertical wavefront radius	m
/params/dRx	error in Rx	m
/params/dRy	error on Ry	m
/params/nval	data type of field values, 2 for complex	1
/params/photonEnergy	central photon energy (mean of spectrum)	eV
/params/wDomain	time or frequency domain	
/params/wEFieldUnit	Electric field unit	
/params/wFloatType	field numerical type	
/params/wSpace	direct (r-space) or reciprocal (q-space)	m
/params/xCentre	x coordinate of wavefront center	m
/params/yCentre	y coordinate of wavefront center	
/history/parent/info/data_description	data documentation	
/history/parent/info/package_version	code version	
/history/parent/misc/FAST2XY.DAT	FAST calculation data	
/history/parent/misc/angular_distribution	angular distribution of source pulse	1/rad
/history/parent/misc/spot_size	fwhm spot size	m
/history/parent/misc/gain_curve	gain curve	1
/history/parent/misc/nzc	Undulator length (point on gain curve)	
/history/parent/misc/temporal_struct	Temporal pulse structure (on-axis projection)	

7.11.2 Coherent wavefront propagation (WPG/SRW)

File format: hdf5 (extension .h5)

Data set	Description	Physical unit
/data/arrEhor	horizontal electric field component	$\sqrt{W/mm^2}$
/data/arrEver	vertical electric field component	$\sqrt{W/mm^2}$
/params/Mesh/nSlices	number of time slices	1
/params/Mesh/nx	number of grid points in horizontal dimension (x)	1
/params/Mesh/ny	number of grid points in vertical dimension (y)	s
/params/Mesh/sliceMax	time corresponding to last slice	s
/params/Mesh/sliceMin	time corresponding to first slice	s
/params/Mesh/xMax	x coordinate of last grid point in horizontal dimension	m
/params/Mesh/xMin	x coordinate of first grid point in horizontal dimension	m
/params/Mesh/yMax	y coordinate of last grid point in vertical dimension	m
/params/Mesh/yMin	y coordinate of first grid point in vertical dimension	m
/params/Mesh/zCoord	z coordinate of the wavefront	m
/params/Rx	instantaneous horizontal wavefront radius	m
/params/Ry	instantaneous vertical wavefront radius	m
/params/dRx	error in Rx	m
/params/dRy	error on Ry	m
/params/nval	data type of field values, 2 for complex	1
/params/photonEnergy	central photon energy (mean of spectrum)	eV
/params/wDomain	time or frequency domain	
/params/wEFieldUnit	Electric field unit	
/params/wFloatType	field numerical type	
/params/wSpace	direct (r-space) or reciprocal (q-space)	m
/params/xCentre	x coordinate of wavefront center	m
/params/yCentre	y coordinate of wavefront center	
/history/parent	Previous module simulation data	
/misc/xFWHM	Horizontal full width at half maximum (intensity distribution)	m
/misc/yFWHM	Vertical full width at half maximum (intensity distribution)	m

7.11.3 Photon-Matter interaction (XMDYN)

File format: hdf5 (extension .h5)

Data set	Description	Physical unit
/data/snp*/ff	2D array of form factors. Rows: Atomic number, Columns: q	1
/data/snp*/halfQ	1D array of half-period resolutions corresponding to column index in form factor arrays	1/Å
/data/snp*/Nph	number of photons in the calculation	1
/data/snp*/r	2D array of atomic position vectors. Rows: Atom index, columns: x,y,z coordinates	m
/data/snp*/T	1D array of unique ID per atomic number	
/data/snp*/Z	1D array of atom types per atomic position vector \$r\$	
/data/snp*/xyz	1D array of indices of ff for each atom in Z	
/data/snp*/Sq_halfQ	half period resolution q space spanned by Sq_bound, Sq_free	1/Å
/data/snp*/Sq_bound	1D array for Compton scattering from bound electrons	1
/data/snp*/Sq_free	1D array for Compton scattering from free electrons	1
/history/parent	Parent data	
/info/package_version	backengine code version	
/info/contact	Support contact for backengine code	
/info/data_description	Data documentation	
/info/method_description	Method documentation	
/version	hdf5 format version	

7.11.4 Diffraction (singFEL)

File format: hdf5 (extension .h5)

Data set	Description	Physical unit
/data/diffr	2D array. Photon number per pixel	
/data/data	2D array. Integrated photon number per pixel, poisson noise added.	
/data/angle	Euler angles applied to atomic positions	
/history/parent	Details of parent data	
/info/package_version	Backengine code version	
/info/contact	Support contact	
/info/data_description	Data documentation	
/info/method_description	Method documentation	
/params/geom/detectorDist	Detector distance	
/params/geom/pixelWidth	Pixel width (x)	
/params/geom/pixelHeight	Pixel height (y)	
/params/geom/mask	Masked pixels	
/params/beam/photonEnergy	Central photon energy	
/params/beam/photons	Number of photons in beam	
/params/beam/focusArea	Beam focus area	
/params/info	Input parameter for backengine code	

PYTHON MODULE INDEX

S

SimEx.Analysis.AbstractAnalysis, 26
SimEx.Analysis.DiffractionAnalysis, 27
SimEx.Analysis.XFELPhotonAnalysis, 30
SimEx.Analysis.XMDYNPhotonMatterAnalysis, 31
SimEx.Calculators.AbstractBaseCalculator, 32
SimEx.Calculators.AbstractPhotonAnalyzer, 34
SimEx.Calculators.AbstractPhotonDetector, 34
SimEx.Calculators.AbstractPhotonDiffraction, 35
SimEx.Calculators.AbstractPhotonInteractor, 35
SimEx.Calculators.AbstractPhotonPropagator, 36
SimEx.Calculators.AbstractPhotonSource, 37
SimEx.Calculators.ComptonScatteringCalculator, 37
SimEx.Calculators.CrystFELPhotonDiffraction, 38
SimEx.Calculators.DMPhasing, 39
SimEx.Calculators.EMCCaseGenerator, 40
SimEx.Calculators.EMCOrientation, 42
SimEx.Calculators.EstherPhotonMatterInteractor, 42
SimEx.Calculators.FEFFPhotonMatterInteractor, 43
SimEx.Calculators.GenesisPhotonSource, 44
SimEx.Calculators.IdealPhotonDetector, 44
SimEx.Calculators.PlasmaXRTSCalculator, 45
SimEx.Calculators.S2EReconstruction, 46
SimEx.Calculators.SingFELPhotonDiffraction, 46
SimEx.Calculators.WavePropagator, 48
SimEx.Calculators.XCSITPhotonDetector, 47
SimEx.Calculators.XFELPhotonSource, 48
SimEx.Calculators.XMDYNDemoPhotonMatterInteractor, 49
SimEx.Calculators.XMDYNPhotonMatterInteractor, 50
SimEx.Parameters.AbstractCalculatorParameters, 52
SimEx.Parameters.AbstractPhotonDiffractionParameters, 52
SimEx.Parameters.CrystFELPhotonDiffractionParameters, 53
SimEx.Parameters.DetectorGeometry, 56
SimEx.Parameters.DMPhasingParameters, 55
SimEx.Parameters.EMCOrientationParameters, 58
SimEx.Parameters.EstherPhotonMatterInteractorParameters, 59
SimEx.Parameters.FEFFPhotonMatterInteractorParameters, 63
SimEx.Parameters.PhotonBeamParameters, 64
SimEx.Parameters.PhotonMatterInteractorParameters, 65
SimEx.Parameters.PlasmaXRTSCalculatorParameters, 65
SimEx.Parameters.SingFELPhotonDiffractionParameters, 70
SimEx.Parameters.WavePropagatorParameters, 71
SimEx.Parameters.XCSITPhotonDetectorParameters, 71

INDEX

A

- ablator () (*SimEx.Parameters.EstherPhotonMatterInteractorParameters* property), 61
- ablator_thickness () (*SimEx.Parameters.EstherPhotonMatterInteractorParameters* property), 61
- AbstractAnalysis (class in *SimEx.Analysis.AbstractAnalysis*), 26
- AbstractBaseCalculator (class in *SimEx.Calculators.AbstractBaseCalculator*), 32
- AbstractCalculatorParameters (class in *SimEx.Parameters.AbstractCalculatorParameters*), 52
- AbstractPhotonAnalyzer (class in *SimEx.Calculators.AbstractPhotonAnalyzer*), 34
- AbstractPhotonDetector (class in *SimEx.Calculators.AbstractPhotonDetector*), 34
- AbstractPhotonDiffractor (class in *SimEx.Calculators.AbstractPhotonDiffractor*), 35
- AbstractPhotonDiffractorParameters (class in *SimEx.Parameters.AbstractPhotonDiffractorParameters*), 52
- AbstractPhotonInteractor (class in *SimEx.Calculators.AbstractPhotonInteractor*), 35
- AbstractPhotonPropagator (class in *SimEx.Calculators.AbstractPhotonPropagator*), 36
- AbstractPhotonSource (class in *SimEx.Calculators.AbstractPhotonSource*), 37
- amplitude_reduction_factor () (*SimEx.Parameters.FEFFPhotonMatterInteractorParameters* property), 63
- animate () (*SimEx.Analysis.XFELPhotonAnalysis.XFELPhotonAnalysis* method), 30
- animate () (*SimEx.Analysis.XMDYNPhotonMatterAnalysis.XMDYNPhotonMatterAnalysis* method), 31
- animatePatterns () (*SimEx.Analysis.DiffractionAnalysis.DiffractionAnalysis* method), 27
- atoms () (*SimEx.Parameters.FEFFPhotonMatterInteractorParameters.FEFFPhotonMatterInteractorParameters* property), 63
- averaging_start () (*SimEx.Parameters.DMPhasingParameters.DMPhasingParameters* property), 56
- azimuthalIntegration () (in module *SimEx.Analysis.DiffractionAnalysis*), 29

B

- backengine () (*SimEx.Calculators.AbstractBaseCalculator.AbstractBaseCalculator* method), 32
- backengine () (*SimEx.Calculators.ComptonScatteringCalculator.ComptonScatteringCalculator* method), 38
- backengine () (*SimEx.Calculators.CrystFELPhotonDiffractor.CrystFELPhotonDiffractor* method), 38
- backengine () (*SimEx.Calculators.DMPhasing.DMPhasing* method), 39
- backengine () (*SimEx.Calculators.EMCOrientation.EMCOrientation* method), 42
- backengine () (*SimEx.Calculators.EstherPhotonMatterInteractor.EstherPhotonMatterInteractor* method), 43
- backengine () (*SimEx.Calculators.FEFFPhotonMatterInteractor.FEFFPhotonMatterInteractor* method), 43
- backengine () (*SimEx.Calculators.GenesisPhotonSource.GenesisPhotonSource* method), 44
- backengine () (*SimEx.Calculators.IdealPhotonDetector.IdealPhotonDetector* method), 44
- backengine () (*SimEx.Calculators.PlasmaXRTSCalculator.PlasmaXRTSCalculator* method), 45
- backengine () (*SimEx.Calculators.S2EReconstruction.S2EReconstruction* method), 46
- backengine () (*SimEx.Calculators.SingFELPhotonDiffractor.SingFELPhotonDiffractor* method), 47
- backengine () (*SimEx.Calculators.WavePropagator.WavePropagator* method), 48
- backengine () (*SimEx.Calculators.XCSITPhotonDetector.XCSITPhotonDetector* method), 47
- backengine () (*SimEx.Calculators.XFELPhotonSource.XFELPhotonSource* method), 48

backendengine() (SimEx.Calculators.XMDYNDemoPhotonMatterInteractorParameters),
 method), 50 68
 backendengine() (SimEx.Calculators.XMDYNPhotonMatterInteractorParameters),
 method), 51 68
 bad_bit_mask() (SimEx.Parameters.DetectorGeometry.DetectorParameters),
 property), 57
 badregion_flag() (SimEx.Parameters.DetectorGeometry.DetectorParameters),
 property), 57 68
 band_gap() (SimEx.Parameters.PlasmaXRTSCalculatorParameters),
 property), 67
 beam_diameter_fwhm() (SimEx.Parameters.PlasmaXRTSCalculatorParameters),
 property), 64 68
 beam_parameters() (SimEx.Parameters.PlasmaXRTSCalculatorParameters),
 property), 53 68
 beam_parameters() (SimEx.Parameters.PlasmaXRTSCalculatorParameters),
 property), 65 68
 beamline() (SimEx.Parameters.WavePropagatorParameters),
 property), 71
 beamstop() (SimEx.Parameters.EMCOrientationParameters),
 property), 59 69
C
 calculate_Compton() (SimEx.Parameters.PhotonMatterInteractorParameters),
 property), 65 62
 calculate_Compton() (SimEx.Parameters.SingFELPhotonDiffractionParameters),
 property), 70 62
 calculate_displacement() (in module SimEx.Analysis.XMDYNPhotonMatterAnalysis),
 31 62
 calculate_fwhm() (in module SimEx.Analysis.XFELPhotonAnalysis),
 30 62
 calculate_ion_charge() (in module SimEx.Analysis.XMDYNPhotonMatterAnalysis),
 32 62
 checkAndSetAblator() (in module SimEx.Parameters.EstherPhotonMatterInteractorParameters),
 61 62
 checkAndSetAblatorThickness() (in module SimEx.Parameters.EstherPhotonMatterInteractorParameters),
 62 62
 checkAndSetBandGap() (in module SimEx.Parameters.PlasmaXRTSCalculatorParameters),
 68 62
 checkAndSetBaseCalculator() (in module SimEx.Analysis.AbstractAnalysis),
 27 69
 checkAndSetBaseCalculator() (in module SimEx.Calculators.AbstractBaseCalculator),
 33 69
 checkAndSetDebyeTemperature() (in module SimEx.Parameters.PlasmaXRTSCalculatorParameters),
 69 69
 checkAndSetDensityAndCharge() (in module SimEx.Parameters.PlasmaXRTSCalculatorParameters),
 68 68
 checkAndSetIonTemperature() (in module SimEx.Parameters.PlasmaXRTSCalculatorParameters),
 69 69
 checkAndSetLaserIntensity() (in module SimEx.Parameters.EstherPhotonMatterInteractorParameters),
 62 62
 checkAndSetLaserPulse() (in module SimEx.Parameters.EstherPhotonMatterInteractorParameters),
 62 62
 checkAndSetLaserPulseDuration() (in module SimEx.Parameters.EstherPhotonMatterInteractorParameters),
 62 62
 checkAndSetLaserWavelength() (in module SimEx.Parameters.EstherPhotonMatterInteractorParameters),
 62 62
 checkAndSetLayer1() (in module SimEx.Parameters.EstherPhotonMatterInteractorParameters),
 62 62
 checkAndSetLayer1Thickness() (in module SimEx.Parameters.EstherPhotonMatterInteractorParameters),
 62 62
 checkAndSetLayer2() (in module SimEx.Parameters.EstherPhotonMatterInteractorParameters),
 62 62
 checkAndSetLayer2Thickness() (in module SimEx.Parameters.EstherPhotonMatterInteractorParameters),
 62 62
 checkAndSetLFC() (in module SimEx.Parameters.PlasmaXRTSCalculatorParameters),
 69 69
 checkAndSetModelIPL() (in module SimEx.Parameters.PlasmaXRTSCalculatorParameters),
 69 69
 checkAndSetModelMix() (in module SimEx.Parameters.PlasmaXRTSCalculatorParameters),
 69 69

SimEx.Parameters.PlasmaXRTSCalculatorParameters), 69

checkAndSetModelSbf() (in module *checkAndSetScatteringAngle()* (in module *SimEx.Parameters.PlasmaXRTSCalculatorParameters*), 69

checkAndSetModelSee() (in module *checkAndSetSourceSpectrum()* (in module *SimEx.Parameters.PlasmaXRTSCalculatorParameters*), 69

checkAndSetModelSii() (in module *checkAndSetSourceSpectrumFWHM()* (in module *SimEx.Parameters.PlasmaXRTSCalculatorParameters*), 69

checkAndSetNumberOfLayers() (in module *checkAndSetWindow()* (in module *SimEx.Parameters.EstherPhotonMatterInteractorParameters*), 62

checkAndSetParameters() (in module *checkAndSetWindowThickness()* (in module *SimEx.Calculators.AbstractBaseCalculator*), 33

checkAndSetParameters() (in module *checkConsistency()* (in module *SimEx.Calculators.ComptonScatteringCalculator*), 38

checkAndSetParameters() (in module *cluster_two_means()* (in module *SimEx.Calculators.PlasmaXRTSCalculator*), 45

checkAndSetPhotonAnalyzer() (in module *ComptonScatteringCalculator* (class in *SimEx.Calculators.AbstractPhotonAnalyzer*), 34

checkAndSetPhotonDetector() (in module *computeNTasks()* (in module *SimEx.Calculators.AbstractPhotonDetector*), 34

checkAndSetPhotonDiffraction() (in module *computeNTasks()* (in module *SimEx.Calculators.AbstractPhotonDiffraction*), 35

checkAndSetPhotonEnergy() (in module *computeNTasks()* (in module *SimEx.Parameters.PlasmaXRTSCalculatorParameters*), 69

checkAndSetPhotonInteractor() (in module *computeNTasks()* (in module *SimEx.Calculators.AbstractPhotonInteractor*), 35

checkAndSetPhotonPropagator() (in module *computeNTasks()* (in module *SimEx.Calculators.AbstractPhotonPropagator*), 36

checkAndSetPhotonSource() (in module *computeNTasks()* (in module *SimEx.Calculators.AbstractPhotonSource*), 37

checkAndSetRunTime() (in module *corners()* (in module *SimEx.Parameters.EstherPhotonMatterInteractorParameters*), 62

checkAndSetSample() (in module *cpus_per_task()* (in module *SimEx.Parameters.EstherPhotonMatterInteractorParameters*), 62

checkAndSetSampleThickness() (in module *createTestScatteringGeometry()* (in module *SimEx.Parameters.EstherPhotonMatterInteractorParameters*), 62

checkAndSetSbfNorm() (in module *crystal_size_max()* (in module *SimEx.Parameters.EstherPhotonMatterInteractorParameters*), 62

checkAndSetScatteringAngle() (in module *crystal_size_min()* (in module *SimEx.Parameters.PlasmaXRTSCalculatorParameters*), 69

checkAndSetSourceSpectrum() (in module *CrystFELPhotonDiffraction* (class in *SimEx.Parameters.PlasmaXRTSCalculatorParameters*), 69

checkAndSetSourceSpectrumFWHM() (in module *CrystFELPhotonDiffractionParameters* (class in *SimEx.Parameters.PlasmaXRTSCalculatorParameters*), 70

checkAndSetWindow() (in module *CrystFELPhotonDiffractionParameters* (class in *SimEx.Parameters.EstherPhotonMatterInteractorParameters*), 62

checkAndSetWindowThickness() (in module *CrystFELPhotonDiffractionParameters* (class in *SimEx.Parameters.EstherPhotonMatterInteractorParameters*), 62

checkConsistency() (in module *CrystFELPhotonDiffractionParameters* (class in *SimEx.Parameters.EstherPhotonMatterInteractorParameters*), 61

cluster_two_means() (in module *CrystFELPhotonDiffractionParameters* (class in *SimEx.Calculators.EMCCaseGenerator*), 41

ComptonScatteringCalculator (class in *CrystFELPhotonDiffractionParameters* (class in *SimEx.Calculators.ComptonScatteringCalculator*), 37

computeNTasks() (in module *CrystFELPhotonDiffractionParameters* (class in *SimEx.Calculators.CrystFELPhotonDiffractionParameters*), 38

computeNTasks() (in module *CrystFELPhotonDiffractionParameters* (class in *SimEx.Calculators.EMCOrientation.EMCOrientationParameters*), 42

computeNTasks() (in module *CrystFELPhotonDiffractionParameters* (class in *SimEx.Calculators.SingFELPhotonDiffractionParameters*), 47

computeNTasks() (in module *CrystFELPhotonDiffractionParameters* (class in *SimEx.Calculators.WavePropagator.WavePropagatorParameters*), 48

corners() (in module *CrystFELPhotonDiffractionParameters* (class in *SimEx.Parameters.DetectorGeometry.DetectorPanel*), 57

cpus_per_task() (in module *CrystFELPhotonDiffractionParameters* (class in *SimEx.Parameters.AbstractCalculatorParameters*), 52

createTestScatteringGeometry() (in module *CrystFELPhotonDiffractionParameters* (class in *SimEx.Calculators.EMCCaseGenerator.EMCCaseGenerator*), 40

crystal_size_max() (in module *CrystFELPhotonDiffractionParameters* (class in *SimEx.Parameters.CrystFELPhotonDiffractionParameters*), 55

crystal_size_min() (in module *CrystFELPhotonDiffractionParameters* (class in *SimEx.Parameters.CrystFELPhotonDiffractionParameters*), 55

CrystFELPhotonDiffraction (class in *CrystFELPhotonDiffractionParameters* (class in *SimEx.Calculators.CrystFELPhotonDiffraction*), 38

CrystFELPhotonDiffractionParameters (class in *CrystFELPhotonDiffractionParameters* (class in *SimEx.Parameters.CrystFELPhotonDiffractionParameters*), 53

D

diffractTestCase() (SimEx.Calculators.EMCCaseGenerator.EMCCaseGenerator method), 40
 data() (SimEx.Calculators.ComptonScatteringCalculator.ComptonScatteringCalculator property), 38
 data() (SimEx.Calculators.CrystFELPhotonDiffractor.CrystFELPhotonDiffractor property), 38
 data() (SimEx.Calculators.DMPhasing.DMPhasing property), 39
 data() (SimEx.Calculators.EMCOrientation.EMCOrientation property), 42
 data() (SimEx.Calculators.EstherPhotonMatterInteractor.EstherPhotonMatterInteractor property), 43
 data() (SimEx.Calculators.FEFFPhotonMatterInteractor.FEFFPhotonMatterInteractor property), 43
 data() (SimEx.Calculators.GenesisPhotonSource.GenesisPhotonSource property), 44
 data() (SimEx.Calculators.IdealPhotonDetector.IdealPhotonDetector property), 45
 data() (SimEx.Calculators.PlasmaXRTSCalculator.PlasmaXRTSCalculator property), 45
 data() (SimEx.Calculators.S2EReconstruction.S2EReconstruction property), 46
 data() (SimEx.Calculators.SingFELPhotonDiffractor.SingFELPhotonDiffractor property), 47
 data() (SimEx.Calculators.WavePropagator.WavePropagator property), 48
 data() (SimEx.Calculators.XFELPhotonSource.XFELPhotonSource property), 48
 data() (SimEx.Calculators.XMDYNDemoPhotonMatterInteractor.XMDYNDemoPhotonMatterInteractor property), 50
 data() (SimEx.Calculators.XMDYNPhotonMatterInteractor.XMDYNPhotonMatterInteractor property), 51
 debye_temperature() (SimEx.Parameters.PlasmaXRTSCalculatorParameters.PlasmaXRTSCalculatorParameters property), 67
 delta_time() (SimEx.Parameters.EstherPhotonMatterInteractorParameters.EstherPhotonMatterInteractorParameters property), 61
 detailed_output() (SimEx.Parameters.EMCOrientationParameters.EMCOrientationParameters property), 59
 detector_geometry() (SimEx.Parameters.AbstractPhotonDiffractorParameters.AbstractPhotonDiffractorParameters property), 53
 detector_type() (SimEx.Parameters.XCSITPhotonDetectorParameters.XCSITPhotonDetectorParameters property), 72
 DetectorGeometry (class in SimEx.Parameters.DetectorGeometry), 56
 detectorGeometryFromFile() (in module SimEx.Parameters.DetectorGeometry), 58
 DetectorPanel (class in SimEx.Parameters.DetectorGeometry), 56
 DiffractionAnalysis (class in SimEx.Analysis.DiffractionAnalysis), 27
 diffractionParameters() (in module SimEx.Analysis.DiffractionAnalysis), 29
 diffractTestCase() (SimEx.Calculators.EMCCaseGenerator.EMCCaseGenerator method), 40
 distance_offset() (SimEx.Parameters.DetectorGeometry.DetectorPanel property), 57
 divergence() (SimEx.Parameters.PhotonBeamParameters.PhotonBeamParameters property), 64
 DMPhasing (class in SimEx.Calculators.DMPhasing), 39
 DMPhasingParameters (class in SimEx.Parameters.DMPhasingParameters), 55
 dumpLoader() (SimEx.Analysis.AbstractAnalysis.AbstractAnalysis class method), 26
 dumpLoader() (SimEx.Calculators.AbstractBaseCalculator.AbstractBaseCalculator class method), 32
 dumpToFile() (SimEx.Analysis.AbstractAnalysis.AbstractAnalysis method), 26
 dumpToFile() (SimEx.Calculators.AbstractBaseCalculator.AbstractBaseCalculator method), 33
 E
 electron_density() (SimEx.Parameters.PlasmaXRTSCalculatorParameters.PlasmaXRTSCalculatorParameters property), 67
 electron_temperature() (SimEx.Parameters.PlasmaXRTSCalculatorParameters.PlasmaXRTSCalculatorParameters property), 67
 elements() (SimEx.Analysis.XMDYNPhotonMatterAnalysis.XMDYNPhotonMatterAnalysis method), 31
 elements() (SimEx.Parameters.PlasmaXRTSCalculatorParameters.PlasmaXRTSCalculatorParameters property), 67
 EMCCaseGenerator (class in SimEx.Calculators.EMCCaseGenerator), 40
 EMCOrientation (class in SimEx.Calculators.EMCOrientation), 42
 EMCOrientationParameters (class in SimEx.Parameters.EMCOrientationParameters), 58
 energy_range() (SimEx.Parameters.PlasmaXRTSCalculatorParameters.PlasmaXRTSCalculatorParameters property), 67
 energy_response() (SimEx.Parameters.DetectorGeometry.DetectorPanel property), 57
 EstherPhotonMatterInteractor (class in

SimEx.Calculators.EstherPhotonMatterInteractor), 50
 42 f_hdf5_simple_read() (in module
 EstherPhotonMatterInteractorParameters *SimEx.Calculators.XMDYNDemoPhotonMatterInteractor*),
 (class in *SimEx.Parameters.EstherPhotonMatterInteractorParameters*),
 59 f_init_random() (*SimEx.Calculators.XMDYNDemoPhotonMatterInteractor*)
 expectedData() (*SimEx.Calculators.AbstractBaseCalculator.AbstractBaseCalculator*
 method), 33 f_load_pulse() (*SimEx.Calculators.XMDYNDemoPhotonMatterInteractor*)
 expectedData() (*SimEx.Calculators.AbstractPhotonPropagator.AbstractPhotonPropagator*
 method), 36 f_load_sample() (*SimEx.Calculators.XMDYNDemoPhotonMatterInteractor*)
 expectedData() (*SimEx.Calculators.AbstractPhotonSource.AbstractPhotonSource*
 method), 37 f_load_snp_content()
 expectedData() (*SimEx.Calculators.ComptonScatteringCalculator.ComptonScatteringCalculator*)
 method), 38 f_load_snp_xxx() (*SimEx.Calculators.XMDYNDemoPhotonMatterInteractor.PMIDemoPhotonMatterInteractor*)
 expectedData() (*SimEx.Calculators.CrystFELPhotonDiffraction.CrystFELPhotonDiffraction*
 method), 38 (class in *SimEx.Calculators.XMDYNDemoPhotonMatterInteractor.PMIDemoPhotonMatterInteractor*)
 expectedData() (*SimEx.Calculators.DMPhasing.DMPhasing* method), 49
 method), 39 f_md_step() (in module
 expectedData() (*SimEx.Calculators.EMCOrientation.EMCOrientation* method), 49
 method), 42 f_rotate_sample()
 expectedData() (*SimEx.Calculators.EstherPhotonMatterInteractor.EstherPhotonMatterInteractor*),
 method), 43 50
 expectedData() (*SimEx.Calculators.FEFFPhotonMatterInteractor.FEFFPhotonMatterInteractor*)
 method), 43 (class in *SimEx.Calculators.XMDYNDemoPhotonMatterInteractor.PMIDemoPhotonMatterInteractor*)
 method), 49
 expectedData() (*SimEx.Calculators.IdealPhotonDetectors.IdealPhotonDetectors_help*) (in module
 method), 45 *SimEx.Calculators.XMDYNDemoPhotonMatterInteractor*),
 expectedData() (*SimEx.Calculators.PlasmaXRTSCalculator.PlasmaXRTSCalculator*
 method), 45 50
 expectedData() (*SimEx.Calculators.S2EReconstruction.S2EReconstruction*)
 method), 46 (class in *SimEx.Calculators.XMDYNDemoPhotonMatterInteractor.PMIDemoPhotonMatterInteractor*)
 method), 49
 expectedData() (*SimEx.Calculators.SingFELPhotonDiffraction.SingFELPhotonDiffraction*)
 method), 47 (class in *SimEx.Calculators.XMDYNDemoPhotonMatterInteractor.PMIDemoPhotonMatterInteractor*)
 method), 49
 expectedData() (*SimEx.Calculators.XCSITPhotonDetectors.XCSITPhotonDetectors*)
 method), 47 (class in *SimEx.Calculators.XMDYNDemoPhotonMatterInteractor.PMIDemoPhotonMatterInteractor*)
 method), 49
 expectedData() (*SimEx.Calculators.XMDYNDemoPhotonMatterInteractor.XMDYNDemoPhotonMatterInteractor*)
 method), 50 (class in *SimEx.Calculators.XMDYNDemoPhotonMatterInteractor.PMIDemoPhotonMatterInteractor*)
 method), 49
 expectedData() (*SimEx.Calculators.XMDYNDemoPhotonMatterInteractor.XMDYNDemoPhotonMatterInteractor*)
 method), 51 (class in *SimEx.Calculators.XMDYNDemoPhotonMatterInteractor.PMIDemoPhotonMatterInteractor*)
 method), 49
 extractDate() (in module f_system_setup() (*SimEx.Calculators.XMDYNDemoPhotonMatterInteractor.PMIDemoPhotonMatterInteractor*)
SimEx.Calculators.PlasmaXRTSCalculator), method), 49
 45 f_time_evolution()
 (*SimEx.Calculators.XMDYNDemoPhotonMatterInteractor.PMIDemoPhotonMatterInteractor*)
 method), 49
F
 f_dbase_setup() (*SimEx.Calculators.XMDYNDemoPhotonMatterInteractor.PMIDemoPhotonMatterInteractor*)
 method), 49 (class in *SimEx.Parameters.DetectorGeometry.DetectorParameters*)
 property), 57
 f_dbase_zq2id() (*SimEx.Calculators.XMDYNDemoPhotonMatterInteractor.FEFFPhotonMatterInteractor*)
 method), 49 (class in *SimEx.Calculators.FEFFPhotonMatterInteractor.PMIDemoPhotonMatterInteractor*)
 method), 49
 f_eval_disp() (in module 43
SimEx.Calculators.XMDYNDemoPhotonMatterInteractor)
 method), 49
 50 (class in *SimEx.Parameters.FEFFPhotonMatterInteractorParameters*)
 method), 49
 f_eval_numE() (in module 63
SimEx.Calculators.XMDYNDemoPhotonMatterInteractor)
 method), 49
 50 (class in *SimEx.Calculators.ComptonScatteringCalculator*)
 method), 49
 f_h5_out2in() (in module 38
SimEx.Calculators.XMDYNDemoPhotonMatterInteractor)
 method), 49
 50 (class in *SimEx.Parameters.FEFFPhotonMatterInteractorParameters*)
 method), 49

method), 63

finalized() (*SimEx.Parameters.FEFFPhotonMatterInteractorParameters.FEFFPhotonMatterInteractorParameters* property), 63

find_two_means() (in module *SimEx.Calculators.EMCCaseGenerator*), 41

forced_mpi_command() (*SimEx.Parameters.AbstractCalculatorParameters.AbstractCalculatorParameters* property), 52

G

GenesisPhotonSource (class in *SimEx.Calculators.GenesisPhotonSource*), 44

getChargeData() (*SimEx.Calculators.XCSITPhotonDetector.XCSITPhotonDetector* method), 47

getInteractionData() (*SimEx.Calculators.XCSITPhotonDetector.XCSITPhotonDetector* method), 47

getPhotonData() (*SimEx.Calculators.XCSITPhotonDetector.XCSITPhotonDetector* method), 47

good_bit_mask() (*SimEx.Parameters.DetectorGeometry.DetectorParameters.DetectorParameters* property), 58

gpus_per_task() (*SimEx.Parameters.AbstractCalculatorParameters.AbstractCalculatorParameters* property), 52

H

h5_out2in() (in module *SimEx.Calculators.XMDYNPhotonMatterInteractor*), 52

I

IdealPhotonDetector (class in *SimEx.Calculators.IdealPhotonDetector*), 44

initial_number_of_quaternions() (*SimEx.Parameters.EMCOrientationParameters.EMCOrientationParameters* property), 59

input_path() (*SimEx.Analysis.AbstractAnalysis.AbstractAnalysis* property), 26

input_path() (*SimEx.Analysis.XMDYNPhotonMatterAnalysis.XMDYNPhotonMatterAnalysis* property), 31

input_path() (*SimEx.Calculators.AbstractBaseCalculator.AbstractBaseCalculator* property), 33

input_path() (*SimEx.Calculators.XCSITPhotonDetector.XCSITPhotonDetector* property), 47

intensities_file() (*SimEx.Parameters.CrystFELPhotonDiffractorParameters.CrystFELPhotonDiffractorParameters* property), 55

intensity() (*SimEx.Analysis.XFELPhotonAnalysis.XFELPhotonAnalysis* property), 30

ion_charge() (*SimEx.Parameters.PlasmaXRTSCalculatorParameters.PlasmaXRTSCalculatorParameters* property), 67

ion_temperature() (*SimEx.Parameters.PlasmaXRTSCalculatorParameters.PlasmaXRTSCalculatorParameters* property), 67

L

laser_intensity() (*SimEx.Parameters.EstherPhotonMatterInteractorParameters.EstherPhotonMatterInteractorParameters* property), 61

laser_pulse() (*SimEx.Parameters.EstherPhotonMatterInteractorParameters.EstherPhotonMatterInteractorParameters* property), 61

laser_pulse_duration() (*SimEx.Parameters.EstherPhotonMatterInteractorParameters.EstherPhotonMatterInteractorParameters* property), 61

laser_wavelength() (*SimEx.Parameters.EstherPhotonMatterInteractorParameters.EstherPhotonMatterInteractorParameters* property), 61

layer1() (*SimEx.Parameters.EstherPhotonMatterInteractorParameters.EstherPhotonMatterInteractorParameters* property), 61

layer1_thickness() (*SimEx.Parameters.EstherPhotonMatterInteractorParameters.EstherPhotonMatterInteractorParameters* property), 61

leash() (*SimEx.Parameters.DMPhasingParameters.DMPhasingParameters* property), 56

lfc() (*SimEx.Parameters.PlasmaXRTSCalculatorParameters.PlasmaXRTSCalculatorParameters* property), 67

load_ff_database() (in module *SimEx.Calculators.XMDYNDemoPhotonMatterInteractor*), 50

load_intensities() (in module *SimEx.Calculators.EMCCaseGenerator*), 41

load_sample() (in module *SimEx.Analysis.XMDYNPhotonMatterAnalysis*), 32

load_snapshot() (*SimEx.Analysis.XMDYNPhotonMatterAnalysis.XMDYNPhotonMatterAnalysis* method), 31

load_snp_from_dir() (*SimEx.Calculators.XMDYNPhotonMatterInteractor.XMDYNPhotonMatterInteractor* method), 51

load_trajectory() (*SimEx.Analysis.XMDYNPhotonMatterAnalysis.XMDYNPhotonMatterAnalysis* method), 31

M

mask() (*SimEx.Analysis.DiffractionAnalysis.DiffractionAnalysis* property), 27

mesh() (*SimEx.Parameters.PlasmaXRTSCalculatorParameters.PlasmaXRTSCalculatorParameters* property), 40

mesh_plane_and_support() (*SimEx.Calculators.EMCCaseGenerator.EMCCaseGenerator* method), 40

mask () (<i>SimEx.Parameters.DetectorGeometry.DetectorPanel</i> <i>property</i>), 58	SimEx.Calculators.EstherPhotonMatterInteractor, 42
mask_nans () (in <i>module</i> <i>SimEx.Analysis.XFELPhotonAnalysis</i>), 30	SimEx.Calculators.FEFFFPhotonMatterInteractor, 43
mass_density () (<i>SimEx.Parameters.PlasmaXRTSCalculatorParameters</i> <i>property</i>), 68	<i>SimEx.Parameters.PlasmaXRTSCalculatorParameters</i> Source, 44
max_number_of_iterations () (<i>SimEx.Parameters.EMCOrientationParameters.EMCOrientationParameters</i> <i>property</i>), 59	SimEx.Calculators.IdealPhotonDetector, 44 SimEx.Calculators.PlasmaXRTSCalculator, 45
max_number_of_quaternions () (<i>SimEx.Parameters.EMCOrientationParameters.EMCOrientationParameters</i> <i>property</i>), 59	SimEx.Calculators.S2EReconstruction, 46
min_error () (<i>SimEx.Parameters.EMCOrientationParameters.EMCOrientationParameters</i> <i>property</i>), 59	SimEx.Calculators.SingFELPhotonDiffraction, 46
model_IPL () (<i>SimEx.Parameters.PlasmaXRTSCalculatorParameters</i> <i>property</i>), 68	SimEx.Calculators.PlasmaXRTSCalculatorParameters, 48 SimEx.Calculators.XCSITPhotonDetector, 43
model_Mix () (<i>SimEx.Parameters.PlasmaXRTSCalculatorParameters</i> <i>property</i>), 68	<i>SimEx.Parameters.PlasmaXRTSCalculatorParameters</i> 43 SimEx.Calculators.XFELPhotonSource, 48
model_Sbf () (<i>SimEx.Parameters.PlasmaXRTSCalculatorParameters</i> <i>property</i>), 68	<i>SimEx.Parameters.PlasmaXRTSCalculatorParameters</i> 48 SimEx.Calculators.XMDYNDemoPhotonMatterInteract 49
model_See () (<i>SimEx.Parameters.PlasmaXRTSCalculatorParameters</i> <i>property</i>), 68	<i>SimEx.Parameters.PlasmaXRTSCalculatorParameters</i> 49 SimEx.Calculators.XMDYNPhotonMatterInteractor, 50
model_Sii () (<i>SimEx.Parameters.PlasmaXRTSCalculatorParameters</i> <i>property</i>), 68	<i>SimEx.Parameters.PlasmaXRTSCalculatorParameters</i> 50 SimEx.Parameters.AbstractCalculatorParameters, 52
module	
SimEx.Analysis.AbstractAnalysis, 26	SimEx.Parameters.AbstractPhotonDiffractionParamete
SimEx.Analysis.DiffractionAnalysis, 27	52 SimEx.Parameters.CrystFELPhotonDiffractionParamete
SimEx.Analysis.XFELPhotonAnalysis, 30	53 SimEx.Parameters.DetectorGeometry, 56
SimEx.Analysis.XMDYNPhotonMatterAnalysis, 31	56 SimEx.Parameters.DMPHasingParameters, 55
SimEx.Calculators.AbstractBaseCalculator, 32	55 SimEx.Parameters.EMCOrientationParameters, 58
SimEx.Calculators.AbstractPhotonAnalyzer, 34	58 SimEx.Parameters.EstherPhotonMatterInteractorPa
SimEx.Calculators.AbstractPhotonDetector, 34	59 SimEx.Parameters.FEFFFPhotonMatterInteractorPara
SimEx.Calculators.AbstractPhotonDiffraction, 35	63 SimEx.Parameters.PhotonBeamParameters, 64
SimEx.Calculators.AbstractPhotonInteractor, 35	64 SimEx.Parameters.PhotonMatterInteractorParamete
SimEx.Calculators.AbstractPhotonPropagator, 36	65 SimEx.Parameters.PlasmaXRTSCalculatorParameters
SimEx.Calculators.AbstractPhotonSource, 37	65 SimEx.Parameters.SingFELPhotonDiffractionParamete
SimEx.Calculators.ComptonScatteringCalculat 37	70 SimEx.Parameters.WavePropagatorParameters, 71
SimEx.Calculators.CrystFELPhotonDiffraction, 38	71 SimEx.Parameters.XCSITPhotonDetectorParameters, 71
SimEx.Calculators.DMPHasing, 39	71
SimEx.Calculators.EMCCaseGenerator, 40	
SimEx.Calculators.EMCOrientation, 42	
	N
nodes_per_task () (<i>SimEx.Parameters.AbstractCalculatorParameters</i> .	

property), 52

npattern() (*SimEx.Analysis.DiffractionAnalysis.DiffractionAnalysis* property), 28

number_of_background_photons() (*SimEx.Parameters.CrystFELPhotonDiffractorParameters.CrystFELPhotonDiffractorParameters* property), 55

number_of_diffraction_patterns() (*SimEx.Parameters.AbstractPhotonDiffractorParameters.AbstractPhotonDiffractorParameters* property), 53

number_of_iterations() (*SimEx.Parameters.DMPhasingParameters.DMPhasingParameters* property), 56

number_of_layers() (*SimEx.Parameters.EstherPhotonMatterInteractorParameters.EstherPhotonMatterInteractorParameters* property), 61

number_of_shrink_cycles() (*SimEx.Parameters.DMPhasingParameters.DMPhasingParameters* property), 56

number_of_slices() (*SimEx.Parameters.SingFELPhotonDiffractorParameters.SingFELPhotonDiffractorParameters* property), 71

number_of_snapshots() (*SimEx.Analysis.XMDYNPhotonMatterAnalysis.XMDYNPhotonMatterAnalysis* method), 31

number_of_trajectories() (*SimEx.Parameters.PhotonMatterInteractorParameters.PhotonMatterInteractorParameters* property), 65

number_of_trials() (*SimEx.Parameters.DMPhasingParameters.DMPhasingParameters* property), 56

numpyPattern() (*SimEx.Analysis.DiffractionAnalysis.DiffractionAnalysis* method), 28

numpyTotalPower() (*SimEx.Analysis.XFELPhotonAnalysis.XFELPhotonAnalysis* method), 30

O

output_path() (*SimEx.Calculators.AbstractBaseCalculator.AbstractBaseCalculator* property), 33

output_path() (*SimEx.Calculators.XCSITPhotonDetector.XCSITPhotonDetector* property), 47

P

panels() (*SimEx.Parameters.DetectorGeometry.DetectorGeometry* property), 56

parameters() (*SimEx.Analysis.DiffractionAnalysis.DiffractionAnalysis* property), 28

parameters() (*SimEx.Calculators.AbstractBaseCalculator.AbstractBaseCalculator* property), 33

parameters() (*SimEx.Calculators.XMDYNPhotonMatterInteractor.XMDYNPhotonMatterInteractor* property), 51

path_to_executable() (*SimEx.Calculators.FEFFFPhotonMatterInteractor.FEFFFPhotonMatterInteractor* property), 43

pattern_indices() (*SimEx.Analysis.DiffractionAnalysis.DiffractionAnalysis* property), 28

patternGenerator() (*SimEx.Analysis.DiffractionAnalysis.DiffractionAnalysis* method), 28

patterns() (*SimEx.Parameters.XCSITPhotonDetectorParameters.XCSITPhotonDetectorParameters* property), 72

patterns_iterator() (*SimEx.Analysis.DiffractionAnalysis.DiffractionAnalysis* property), 28

photon_energy() (*SimEx.Parameters.PhotonBeamParameters.PhotonBeamParameters* property), 64

photon_energy_relative_bandwidth() (*SimEx.Parameters.PhotonBeamParameters.PhotonBeamParameters* property), 64

photon_energy_spectrum_type() (*SimEx.Parameters.PhotonBeamParameters.PhotonBeamParameters* property), 64

photon_response() (*SimEx.Analysis.DetectorGeometry.DetectorPanel* property), 58

PhotonBeamParameters (class in *SimEx.Parameters.PhotonBeamParameters*), 64

PhotonMatterInteractorParameters (class in *SimEx.Parameters.PhotonMatterInteractorParameters*), 65

physics() (*SimEx.Analysis.DiffractionAnalysis.DiffractionAnalysis* method), 28

pixel_size() (*SimEx.Parameters.DetectorGeometry.DetectorPanel* property), 58

placePixel() (*SimEx.Calculators.EMCCaseGenerator.EMCCaseGenerator* method), 40

plasma_search_flag() (*SimEx.Parameters.XCSITPhotonDetectorParameters.XCSITPhotonDetectorParameters* property), 72

PlasmaXRTSCalculator (class in *SimEx.Calculators.PlasmaXRTSCalculator*), 45

PlasmaXRTSCalculatorParameters (class in *SimEx.Parameters.PlasmaXRTSCalculatorParameters*), 65

plot_charge() (*SimEx.Analysis.XMDYNPhotonMatterAnalysis.XMDYNPhotonMatterAnalysis* method), 31

plot_displacement() (*SimEx.Analysis.XMDYNPhotonMatterAnalysis.XMDYNPhotonMatterAnalysis* method), 31

plot_energies() (*SimEx.Analysis.XMDYNPhotonMatterAnalysis.XMDYNPhotonMatterAnalysis* method), 31

method), 31

plotImage () (in module *SimEx.Analysis.DiffractionAnalysis*), 29

plotIntensityMap () (in module *SimEx.Analysis.XFELPhotonAnalysis.XFELPhotonAnalysis*), 30

plotOnAxisPowerDensity () (in module *SimEx.Analysis.XFELPhotonAnalysis.XFELPhotonAnalysis*), 30

plotPattern () (in module *SimEx.Analysis.DiffractionAnalysis.DiffractionAnalysis*), 28

plotRadialProjection () (in module *SimEx.Analysis.DiffractionAnalysis*), 29

plotRadialProjection () (in module *SimEx.Analysis.DiffractionAnalysis.DiffractionAnalysis*), 28

plotResolutionRings () (in module *SimEx.Analysis.DiffractionAnalysis*), 29

plotTotalPower () (in module *SimEx.Analysis.XFELPhotonAnalysis.XFELPhotonAnalysis*), 30

pmi_start_ID () (in module *SimEx.Parameters.SingFELPhotonDiffractionParameters.SingFELPhotonDiffractionParameters*), 71

pmi_stop_ID () (in module *SimEx.Parameters.SingFELPhotonDiffractionParameters.SingFELPhotonDiffractionParameters*), 71

PMIDemo (class in *SimEx.Calculators.XMDYNDemoPhotonMatterInteractor*), 49

point_simulation_method () (in module *SimEx.Parameters.XCSITPhotonDetectorParameters.XCSITPhotonDetectorParameters*), 72

poissonize () (in module *SimEx.Analysis.DiffractionAnalysis.DiffractionAnalysis*), 28

poissonize () (in module *SimEx.Parameters.CrystFELPhotonDiffractionParameters.CrystFELPhotonDiffractionParameters*), 55

potentials () (in module *SimEx.Parameters.FEFFPhotonMatterInteractorParameters.FEFFPhotonMatterInteractorParameters*), 63

powder () (in module *SimEx.Parameters.CrystFELPhotonDiffractionParameters.CrystFELPhotonDiffractionParameters*), 55

propToBeamParameters () (in module *SimEx.Parameters.PhotonBeamParameters*), 64

providedData () (in module *SimEx.Calculators.AbstractBaseCalculator.AbstractBaseCalculator*), 33

providedData () (in module *SimEx.Calculators.AbstractPhotonPropagator.AbstractPhotonPropagator*), 36

providedData () (in module *SimEx.Calculators.AbstractPhotonSource.AbstractPhotonSource*), 37

providedData () (in module *SimEx.Calculators.ComptonScatteringCalculator.ComptonScatteringCalculator*), 38

providedData () (in module *SimEx.Calculators.CrystFELPhotonDiffraction.CrystFELPhotonDiffraction*), 38

providedData () (in module *SimEx.Calculators.DMPhasing.DMPhasing*), 39

providedData () (in module *SimEx.Calculators.EMCOrientation.EMCOrientation*), 42

providedData () (in module *SimEx.Calculators.EstherPhotonMatterInteractor.EstherPhotonMatterInteractor*), 43

providedData () (in module *SimEx.Calculators.FEFFPhotonMatterInteractor.FEFFPhotonMatterInteractor*), 43

providedData () (in module *SimEx.Calculators.IdealPhotonDetector.IdealPhotonDetector*), 45

providedData () (in module *SimEx.Calculators.PlasmaXRTSCalculator.PlasmaXRTSCalculator*), 45

providedData () (in module *SimEx.Calculators.S2EReconstruction.S2EReconstruction*), 46

providedData () (in module *SimEx.Calculators.SingFELPhotonDiffraction.SingFELPhotonDiffraction*), 47

providedData () (in module *SimEx.Calculators.XCSITPhotonDetector.XCSITPhotonDetector*), 47

providedData () (in module *SimEx.Calculators.XMDYNDemoPhotonMatterInteractor.XMDYNDemoPhotonMatterInteractor*), 50

providedData () (in module *SimEx.Calculators.XMDYNPhotonMatterInteractor.XMDYNPhotonMatterInteractor*), 51

providedData () (in module *SimEx.Parameters.PhotonBeamParameters.PhotonBeamParameters*), 64

Q

pmi_start_ID () (in module *SimEx.Parameters.SingFELPhotonDiffractionParameters.SingFELPhotonDiffractionParameters*), 71

pmi_stop_ID () (in module *SimEx.Parameters.SingFELPhotonDiffractionParameters.SingFELPhotonDiffractionParameters*), 71

PMIDemo (class in *SimEx.Calculators.XMDYNDemoPhotonMatterInteractor*), 49

R

ranges () (in module *SimEx.Parameters.DetectorGeometry.DetectorPanel.DetectorPanel*), 38

read_h5_dataset () (in module *SimEx.Analysis.XMDYNPhotonMatterAnalysis*), 32

readGeomFromDetectorFile () (in module *SimEx.Calculators.EMCCaseGenerator.EMCCaseGenerator*), 40

readGeomFromPhotonData () (in module *SimEx.Calculators.EMCCaseGenerator.EMCCaseGenerator*), 40

root_path () (in module *SimEx.Calculators.XMDYNPhotonMatterInteractor.XMDYNPhotonMatterInteractor*), 51

rotation () (in module *SimEx.Parameters.PhotonMatterInteractorParameters.PhotonMatterInteractorParameters*), 68

run_files_path () (in module *SimEx.Calculators.EMCOrientation.EMCOrientation*), 42

run_time () (in module *SimEx.Parameters.EstherPhotonMatterInteractorParameters.EstherPhotonMatterInteractorParameters*), 61

runFromCLI () (in module *SimEx.Analysis.AbstractAnalysis.AbstractAnalysis*), 26

runFromCLI () (in module *SimEx.Calculators.AbstractBaseCalculator.AbstractBaseCalculator*), 35

s2e_gen_randrot_quat () (in module *SimEx.Calculators.XMDYNDemoPhotonMatterInteractor*), 50

s2e_rand_orient() (in module saveH5_geom() (SimEx.Calculators.CrystFELPhotonDiffractor.CrystFELPhotonDiffractor), method), 39

50 Sbf_norm() (SimEx.Parameters.PlasmaXRTSCalculatorParameters.PlasmaXRTSCalculatorParameters), 67

S2EReconstruction (class in property), 67

SimEx.Calculators.S2EReconstruction), 46 scattering_angle() (SimEx.Parameters.AbstractPhotonDiffractorParameters.AbstractPhotonDiffractorParameters), 68

sample() (SimEx.Parameters.AbstractPhotonDiffractorParameters.AbstractPhotonDiffractorParameters), 53

sample() (SimEx.Parameters.EstherPhotonMatterInteractorParameters.EstherPhotonMatterInteractorParameters), 61

sample_path() (SimEx.Analysis.XMDYNPhotonMatterAnalysis.XMDYNPhotonMatterAnalysis), 31

sample_path() (SimEx.Calculators.XMDYNPhotonMatterInteractor.XMDYNPhotonMatterInteractor), 51

sample_thickness() (SimEx.Parameters.EstherPhotonMatterInteractorParameters.EstherPhotonMatterInteractorParameters), 61

saturation_adu() (SimEx.Parameters.DetectorGeometry.DetectorGeometry), 29

saturation_map() (SimEx.Parameters.DetectorGeometry.DetectorGeometry), 58

saveH5() (SimEx.Calculators.AbstractBaseCalculator.AbstractBaseCalculator), 33

saveH5() (SimEx.Calculators.ComptonScatteringCalculator.ComptonScatteringCalculator), 38

saveH5() (SimEx.Calculators.CrystFELPhotonDiffractor.CrystFELPhotonDiffractor), 39

saveH5() (SimEx.Calculators.DMPhasing.DMPhasing), 39

saveH5() (SimEx.Calculators.EMCOrientation.EMCOrientation), 42

saveH5() (SimEx.Calculators.EstherPhotonMatterInteractor.EstherPhotonMatterInteractor), 43

saveH5() (SimEx.Calculators.FEFFPhotonMatterInteractor.FEFFPhotonMatterInteractor), 44

saveH5() (SimEx.Calculators.GenesisPhotonSource.GenesisPhotonSource), 44

saveH5() (SimEx.Calculators.IdealPhotonDetector.IdealPhotonDetector), 45

saveH5() (SimEx.Calculators.PlasmaXRTSCalculator.PlasmaXRTSCalculator), 45

saveH5() (SimEx.Calculators.S2EReconstruction.S2EReconstruction), 46

saveH5() (SimEx.Calculators.SingFELPhotonDiffractor.SingFELPhotonDiffractor), 47

saveH5() (SimEx.Calculators.WavePropagator.WavePropagator), 48

saveH5() (SimEx.Calculators.XCSITPhotonDetector.XCSITPhotonDetector), 48

saveH5() (SimEx.Calculators.XFELPhotonSource.XFELPhotonSource), 49

saveH5() (SimEx.Calculators.XMDYNDemoPhotonMatterInteractor.XMDYNDemoPhotonMatterInteractor), 50

saveH5() (SimEx.Calculators.XMDYNPhotonMatterInteractor.XMDYNPhotonMatterInteractor), 52

show_support() (in module SimEx.Calculators.DMPhasing), 39

showDensity() (SimEx.Calculators.EMCCaseGenerator.EMCCaseGenerator), 40

showDetector() (SimEx.Calculators.EMCCaseGenerator.EMCCaseGenerator), 41

showLogIntensity() (SimEx.Calculators.EMCCaseGenerator.EMCCaseGenerator), 41

showLogIntensitySlices() (SimEx.Calculators.EMCCaseGenerator.EMCCaseGenerator), 41

SimEx.Analysis.AbstractAnalysis

SimEx.Analysis.DiffractionAnalysis

SimEx.Analysis.XFELPhotonAnalysis

SimEx.Analysis.XMDYNPhotonMatterAnalysis

SimEx.Calculators.AbstractBaseCalculator

SimEx.Calculators.AbstractPhotonAnalyzer

SimEx.Calculators.AbstractPhotonDetector

SimEx.Calculators.AbstractPhotonDiffractor

SimEx.Calculators.AbstractPhotonInteractor

SimEx.Calculators.AbstractPhotonPropagator

SimEx.Calculators.AbstractPhotonSource

SimEx.Calculators.ComptonScatteringCalculator

SimEx.Calculators.CrystFELPhotonDiffractor

module, 38
 SimEx.Calculators.DMPHasing
 module, 39
 SimEx.Calculators.EMCCaseGenerator
 module, 40
 SimEx.Calculators.EMCOrientation
 module, 42
 SimEx.Calculators.EstherPhotonMatterInteractor 46
 module, 42
 SimEx.Calculators.FEFFPhotonMatterInteractor
 module, 43
 SimEx.Calculators.GenesisPhotonSource
 module, 44
 SimEx.Calculators.IdealPhotonDetector
 module, 44
 SimEx.Calculators.PlasmaXRTSCalculator
 module, 45
 SimEx.Calculators.S2EReconstruction
 module, 46
 SimEx.Calculators.SingFELPhotonDiffractor
 module, 46
 SimEx.Calculators.WavePropagator
 module, 48
 SimEx.Calculators.XCSITPhotonDetector
 module, 47
 SimEx.Calculators.XFELPhotonSource
 module, 48
 SimEx.Calculators.XMDYNDemoPhotonMatterInteractor
 module, 49
 SimEx.Calculators.XMDYNPhotonMatterInteractor
 module, 50
 SimEx.Parameters.AbstractCalculatorParameters
 module, 52
 SimEx.Parameters.AbstractPhotonDiffractorParameters
 module, 52
 SimEx.Parameters.CrystFELPhotonDiffractorParameters
 module, 53
 SimEx.Parameters.DetectorGeometry
 module, 56
 SimEx.Parameters.DMPHasingParameters
 module, 55
 SimEx.Parameters.EMCOrientationParameters
 module, 58
 SimEx.Parameters.EstherPhotonMatterInteractorParameters
 module, 59
 SimEx.Parameters.FEFFPhotonMatterInteractorParameters
 module, 63
 SimEx.Parameters.PhotonBeamParameters
 module, 64
 SimEx.Parameters.PhotonMatterInteractorParameters
 module, 65
 SimEx.Parameters.PlasmaXRTSCalculatorParameters
 module, 65
 SimEx.Parameters.SingFELPhotonDiffractorParameters
 module, 70
 SimEx.Parameters.WavePropagatorParameters
 module, 71
 SimEx.Parameters.XCSITPhotonDetectorParameters
 module, 71
 SingFELPhotonDiffractor (class in
 SimEx.Calculators.SingFELPhotonDiffractor),
 46
 SingFELPhotonDiffractorParameters (class
 in *SimEx.Parameters.SingFELPhotonDiffractorParameters*),
 70
 slice_interval() (*SimEx.Parameters.SingFELPhotonDiffractorParameters*
 property), 71
 slow_scan_xyz() (*SimEx.Parameters.DetectorGeometry.DetectorPanel*
 property), 58
 snapshot_indices()
 (*SimEx.Analysis.XMDYNPhotonMatterAnalysis.XMDYNPhotonMatterAnalysis*
 property), 31
 solidAngles() (*SimEx.Analysis.DiffractionAnalysis.DiffractionAnalysis*
 property), 29
 source_spectrum()
 (*SimEx.Parameters.PlasmaXRTSCalculatorParameters.PlasmaXRTSCalculatorParameters*
 property), 68
 source_spectrum_fwhm()
 (*SimEx.Parameters.PlasmaXRTSCalculatorParameters.PlasmaXRTSCalculatorParameters*
 property), 68
 static_data() (*SimEx.Calculators.PlasmaXRTSCalculator.PlasmaXRTSCalculator*
 property), 45
 statistics() (*SimEx.Analysis.DiffractionAnalysis.DiffractionAnalysis*
 method), 29
 support_from_autocorr() (in module
 SimEx.Calculators.EMCCaseGenerator),
 41
 support_from_singles()
 (*SimEx.Parameters.CrystFELPhotonDiffractorParameters.CrystFELPhotonDiffractorParameters*
 property), 55

T

tmp_files_path() (*SimEx.Calculators.EMCOrientation.EMCOrientation*
 property), 42
 totalNPattern() (in module
 SimEx.Analysis.DiffractionAnalysis), 29

U

uniform_rotation()
 (*SimEx.Parameters.AbstractPhotonDiffractorParameters.AbstractPhotonDiffractorParameters*
 property), 53
 use_opmd() (*SimEx.Parameters.WavePropagatorParameters.WavePropagatorParameters*
 property), 71

W

wavefront() (*SimEx.Analysis.XFELPhotonAnalysis.XFELPhotonAnalysis*
 property), 30

WavePropagator (class in 41
SimEx.Calculators.WavePropagator), 48

WavePropagatorParameters (class in
SimEx.Parameters.WavePropagatorParameters),
 71

window() (*SimEx.Parameters.EstherPhotonMatterInteractorParameters.EstherPhotonMatterInteractorParameters*
property), 61

window_thickness() (*SimEx.Parameters.EstherPhotonMatterInteractorParameters.EstherPhotonMatterInteractorParameters*
property), 61

working_directory() (*SimEx.Calculators.FEFFPhotonMatterInteractor.FEFFPhotonMatterInteractor*
property), 44

writeAllOutputToFile() (*SimEx.Calculators.EMCCaseGenerator.EMCCaseGenerator*
method), 41

writeDensityToFile() (*SimEx.Calculators.EMCCaseGenerator.EMCCaseGenerator*
method), 41

writeDetectorToFile() (*SimEx.Calculators.EMCCaseGenerator.EMCCaseGenerator*
method), 41

writeSparsePhotonFile() (*SimEx.Calculators.EMCCaseGenerator.EMCCaseGenerator*
method), 41

writeSupportToFile() (*SimEx.Calculators.EMCCaseGenerator.EMCCaseGenerator*
method), 41

X

XCSITPhotonDetector (class in
SimEx.Calculators.XCSITPhotonDetector),
 47

XCSITPhotonDetectorParameters (class in
SimEx.Parameters.XCSITPhotonDetectorParameters),
 71

XFELPhotonAnalysis (class in
SimEx.Analysis.XFELPhotonAnalysis), 30

XFELPhotonSource (class in
SimEx.Calculators.XFELPhotonSource),
 48

XMDYNDemoPhotonMatterInteractor (class in
SimEx.Calculators.XMDYNDemoPhotonMatterInteractor),
 49

XMDYNPhotonMatterAnalysis (class in
SimEx.Analysis.XMDYNPhotonMatterAnalysis),
 31

XMDYNPhotonMatterInteractor (class in
SimEx.Calculators.XMDYNPhotonMatterInteractor),
 50

Z

zero_neg() (in module
SimEx.Calculators.EMCCaseGenerator),