
nexoclom

Release 2.0.15

Matthew Burger

Jan 13, 2020

CONTENTS

1	Input	3
2	LOSResult	5
3	Output	7
4	configure_model	9
5	nexoclom Outline	11
6	Input Files	15
	Python Module Index	17
	Index	19

This is the documentation for nexoclom. Monte Carlo model of neutral clouds and exospheres

INPUT

class nexoclom.**Input** (*infile*)

Bases: `object` (<https://docs.python.org/3/library/functions.html#object>)

Read the input options from a file.

Parameters

infile Plain text file containing model input parameters. See `inputfiles` for a description of the input file format.

Class Attributes

- `geometry`
- `surface_interaction`
- `forces`
- `spatialdist`
- `speeddist`
- `angulardist`
- `options`

Methods Summary

<code>search()</code>	Search the database for previous model runs with the same inputs.
<code>produce_image(format_[, filenames, overwrite])</code>	
<code>run(npackets[, packs_per_it, overwrite, ...])</code>	Run the nexoclom model with the current inputs.

Methods Documentation

search ()

Search the database for previous model runs with the same inputs. See `searchtolerances` for tolerances used in searches.

Parameters

No parameters.

Returns

- A list of filenames corresponding to the inputs.

- Number of packets contained in those saved outputs.
- Total modeled source rate.

produce_image (*format_*, *filenames=None*, *overwrite=False*)

run (*npackets*, *packs_per_it=None*, *overwrite=False*, *compress=True*)

Run the nexoclom model with the current inputs.

Parameters

npackets Number of packets to simulate

packs_per_it Maximum number of packets to run at one time. Default = 1e5 in constant step-size mode; 1e6 in adaptive step-size mode.

overwrite Erase any files matching the current inputs that exist. Default = False

compress Remove packets with frac=0 from the outputs to reduce file size. Default = True

Outputs

Nothing is returned, but model runs are saved and cataloged.

LOSRESULT

class nexoclom.**LOSResult** (*inputs, data, quantity, dphi=<Quantity 3. deg>, filenames=None, overwrite=False, **kwargs*)

Bases: nexoclom.ModelResults.ModelResult

Determine column or emission along lines of sight. This assumes the model has already been run.

inputs An Inputs object

data A Pandas DataFrame object with information on the lines of sight.

quantity Quantity to calculate: 'column', 'radiance', 'density'

dphi Angular size of the view cone. Default = 3 deg.

filenames A filename or list of filenames to use. Default = None is to find all files created for the inputs.

overwrite If True, deletes any images that have already been computed. Default = False

Methods Summary

`create_model(data, outfile, **kwargs)`

`restore(data, fname)`

`save(data, fname, radiance, packets)`

Methods Documentation

create_model (*data, outfile, **kwargs*)

restore (*data, fname*)

save (*data, fname, radiance, packets*)

OUTPUT

class nexoclom.**Output** (*inputs, npackets, compress=True, logger=None*)

Bases: `object` (<https://docs.python.org/3/library/functions.html#object>)

Determine and store packet trajectories.

Parameters

inputs An Input object with the run parameters.

npackets Number of packets to run.

compress Remove packets with frac=0 from the outputs to reduce file size. Default = True

Class Attributes

x0, y0, z0

f0

vx0, vy0, vz0

phi0, lat0, lon0

time, x, y, z, vx, vy, vz index, npackets, totalsource

inputs The inputs used for the simulation

logfile Path to file with output log

compress Whether output is compressed.

unit Basic length unit used. Equal to radius of central planet.

GM GM_planet in units of R_planet/s**2

aplanet Distance of planet from the Sun in AU

vrplanet Radial velocity of planet relative to the Sun in R_planet/s

radpres Radiation pressure object containing acceleration as function of velocity in units of R_planet/s**2 and R_planet/s

Methods Summary

<code>Output.determine_filename</code>	
<code>Output.driver</code>	
<code>restore(filename)</code>	
<code>save()</code>	Add output to database and save as a pickle.
<code>Output.source_distribution</code>	
<code>Output.stream_driver</code>	

Methods Documentation

classmethod `restore` (*filename*)

save ()

Add output to database and save as a pickle.

CONFIGURE_MODEL

Create and read configuration file, create necessary database tables.

`nexoclom.configure_model.configfile()`

Configure external resources used in the model.

The following parameters can be saved in the file `$HOME/.nexoclom`.

- `savepath` = <path where output files are saved>
- `datapath` = <path where MESSENGER data is kept>
- `database` = <name of the postgresql database to use> (*optional*)
- `port` = <port for postgresSQL server to use> (*optional*)

If `savepath` and `datapath` are not present, user is prompted to enter them.

`nexoclom.configure_model.configure_model()`

Ensure the database and configuration file are set up for nexoclom.

Parameters

No parameters.

Returns

No output.

`nexoclom.configure_model.verify_output_tables()`

Create the database tables used by nexoclom to save output.

NEXOCLOM OUTLINE

- `configure_model.py`
 - Functions
 - * `configfile()`
 - * `set_up_output_tables()`
 - * `configure_model()`
 - Classes
- `Input.py`
 - Functions
 - Classes
 - * `Geometry`
 - * `StickingInfo`
 - * `Forces`
 - * `SpatialDist`
 - * `SpeedDist`
 - * `AngularDist`
 - * `Options`
- `input_classes.py`
 - Functions
 - Classes
- `modeldriver.py`
 - Functions
 - * `modeldriver()`
 - * `delete_files()`
 - Classes
- `Output.py`
 - Functions
 - Classes

- * Output
- satellite_initial_positions.py
 - Functions
 - Classes
- LossInfo.py
 - Functions
 - Classes
 - * LossInfo
- State.py
 - Functions
 - Classes
- bouncepackets.py
 - Functions
 - Classes
- rk5.py
 - Functions
 - Classes
- source_distribution.py
 - Functions
 - Classes
- xyz_to_magcoord.py
 - Functions
 - Classes
- ModelResults.py
 - Functions
 - Classes
 - * ModelResult
- LOSResult.py
 - Functions
 - Classes
 - * LOSResult(ModelResult)
- produce_image.py
 - Functions
 - Classes
 - * ModelImage(ModelResult)
- IDLout.py

- Functions
 - Classes
 - * IDLout
- `__init__.py`
 - Functions
 - Classes

INPUT FILES

The inputs for a model run are defined in an plain text file with lines in the form:

```
category.parameter = setting
```

A description of everything that can go into an input file is at [nexoclom/inputfiles](#).

PYTHON MODULE INDEX

n

`nexoclom.configure_model`, 9

INDEX

C

`configfile()` (in module *nexoclom.configure_model*), 9
`configure_model()` (in module *nexoclom.configure_model*), 9
`create_model()` (*nexoclom.LOSResult* method), 5

I

Input (class in *nexoclom*), 3

L

LOSResult (class in *nexoclom*), 5

N

nexoclom.configure_model (module), 9

O

Output (class in *nexoclom*), 7

P

`produce_image()` (*nexoclom.Input* method), 4

R

`restore()` (*nexoclom.LOSResult* method), 5
`restore()` (*nexoclom.Output* class method), 8
`run()` (*nexoclom.Input* method), 4

S

`save()` (*nexoclom.LOSResult* method), 5
`save()` (*nexoclom.Output* method), 8
`search()` (*nexoclom.Input* method), 3

V

`verify_output_tables()` (in module *nexoclom.configure_model*), 9