tommaso grassi / Untitled project

compute_qabs

Clone

compute Qabs and Qext from rifractive index or dielectric

master Files Filter files

/

Name	Size	Last commit	Message
data		2022-05-23	data updated to 2022 paper
README.md	2.43 KB	2019-02-11	README.md edited online with Bitbu...
bhcoat.py	4.58 KB	2018-09-21	PY: handles coating
bhmie.py	5.75 KB	2021-08-30	python 3 updated
clean	527 B	2021-08-30	python 3 updated
dust.py	7.84 KB	2021-08-30	python 3 updated
material.py	14.78 KB	2021-08-30	python 3 updated
optical.py	10.12 KB	2021-08-30	python 3 updated
qabsmana...	12.19 KB	2021-08-30	python 3 updated
test_01.py	337 B	2019-02-11	PY: simplified test 01
test_02.py	101 B	2019-02-11	PY: import fix in test 2
test_03.py	870 B	2019-02-11	PY: test03 compute coated grains si...
test_04.py	4.1 KB	2019-02-13	PY: constant m fixed
test_05.py	3.91 KB	2022-05-23	test_05 updated to paper 2022
test_06.py	3.05 KB	2022-05-23	added script for paper 2022
utility.py	721 B	2018-11-05	PY: better argument name

README.md

README

This code computes Qabs, Qsca, and Qext from the refractive index or the dielectric constant using Mie's theory.
It also compute opacity for coated materials.

Getting started

Basic usage to load a material and compute its opacity for a grain size distribution is (see `test_01.py` file)

```
from qabsmanager import QabsManager

# create object
q = QabsManager()
```

```
core = q.load_material( data/eps_carb_P93.dat , labels=[ wlen ,  real_m ,  im_m ])

# compute opacity for core material
core.compute_kappa()

# print some info to screen
q.report()

# save opacity to plot
core.plot_kappa("kappa_01.png")
```

Different file formats

To load data from files with different formats, for example a file with wavelength, real and imaginary part of dielectric:

```
from qabsmanager import Qabs_utils

# create object
q = Qabs_utils()

# load data from file with given format
q.load_eps("eps_CO.dat", labels=["wlen", "real_eps1", "im_eps"])
```

The file can be both space- or tab-separated.

labs are
`wlen` : wavelength in micron
`real_eps1` : real part of dialectric - 1
`real_eps` : real part of dialectric
`im_eps` : imaginary part of dielectric
`real_m1` : real part of refractive index - 1
`real_m` : real part of refractive index
`im_m` : imaginary part of refractive index

Composite material

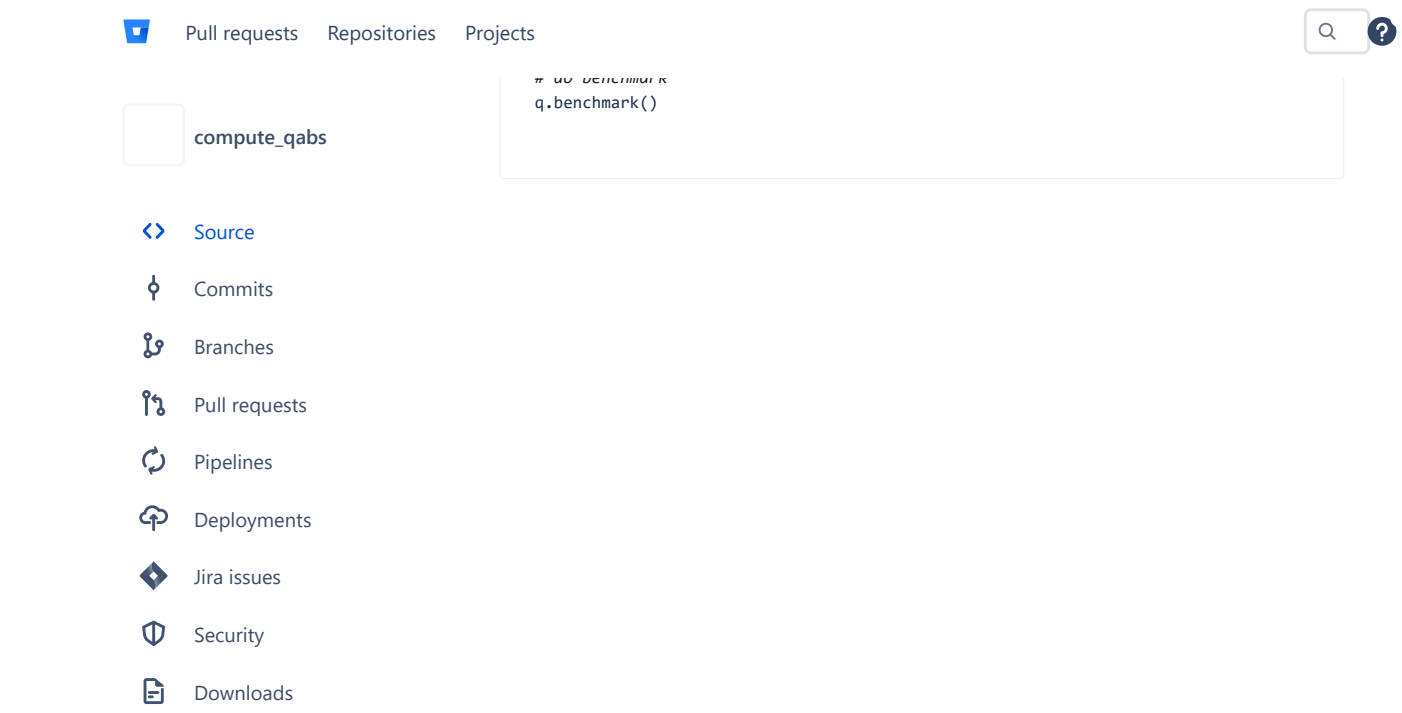
It is possible to use coated materials by loading their optical properties (see `test03.py`).

```
from qabsmanager import QabsManager

# create object
q = QabsManager()

q.clear_plots()

# load core data from file
core = q.load_material("data/eps_carb_P93.dat", labels=["wlen", "real_m", "im_m"])

# load ice data from file
mantle = q.load_material("data/eps_H93.dat", labels=["wlen", "real_m", "im_m"])

# create new material from core and mantle
composite = q.make_optical([core, mantle])

# set size ratio silicon/ice material
composite.dust.aratio = 0.15

# compute opacity of the composite material
composite.compute_kappa()

# plot opacity
composite.add_plot_kappa("kappa_03.png", postfix="coated", xlim=(5e1, 1e3), ylim=(1
```

Benchmark

The code has been benchmarked against the Original Astronomical Silicate (Draine & Lee 1984; Laor & Draine 1993) link.
See `test_02.py`

```
from qabsmanager import Qabs_utils
```


Pull requests
Repositories
Projects
# do benchmark
q.benchmark()
compute_qabs
Source
Commits
Branches
Pull requests
Pipelines
Deployments
Jira issues
Security
Downloads