sigma0 Documentation

Release 0.0.1

TU Wien

May 01, 2020

Contents

1 Contents **3**
- 1.1 License . . . 3
- 1.2 Contributors . . . 3
- 1.3 Changelog . . . 4
- 1.4 sigma0 . . . 4

2 Indices and tables **7**

Python Module Index **9**

This is the documentation of **sigma0**.

CHAPTER 1

Contents

1.1 License

1.2 Contributors

- Sebastian Hahn <sebastian.hahn@geo.tuwien.ac.at>
- Raphael Quast <raphael.quast@geo.tuwien.ac.at>

1.3 Changelog

1.3.1 Version 0.0.1

- First release
- Update pyscaffold v3.2.3

1.4 sigma0

1.4.1 sigma0 package

Submodules

sigma0.oh_model module

sigma0.oh_model.**gammah**(*eps*, *theta*)
Incoherent reflectivity for H-pol.

Parameters

- **eps** (*complex number*) – Complex dielectric constant.
- **theta** (*float*) – Angle in degrees.

Returns **gammah** – Incoherent reflectivity for H-pol.

Return type float

sigma0.oh_model.**gammav**(*eps*, *theta*)
Incoherent reflectivity for V-pol.

Parameters

- **eps** (*complex number*) – Complex dielectric constant.
- **theta** (*float*) – Angle in degrees.

Returns **gammav** – Incoherent reflectivity for V-pol.

Return type float

sigma0.oh_model.**sigma0_bare**(*theta*, *eps_low*, *f_rms*, *f*, *eps_top=1*)
Oh et.al. (1992) surface backscatter calculations

This functions calculations surface backscatter using the Oh et al. (1992) surface model.

References Oh et al., 1992, An empirical model and an inversion technique for rader scattering from bare soil surfaces. IEEE Trans. Geos. Rem., 30, pp. 370-380

Parameters

- **theta** (*float*) – incidence angle in degrees
- **eps_low** (*complex number*) – complex permittivity of lower medium
- **f** (*float*) – frequency in hertz
- **f_rms** (*float*) – fractional rms height. rms_height in meters divided by wavelength

- **eps_top** (*complex number, optional*) – complex permittivity of upper(incoming) medium

Returns **sigma0** – Surface backscatter.

Return type float

sigma0.skeleton module

This is a skeleton file that can serve as a starting point for a Python console script. To run this script uncomment the following line in the entry_points section in setup.py:

[console_scripts] fibonacci = sigma0.skeleton:run

Then run *python setup.py install* which will install the command *fibonacci* inside your current environment. Besides console scripts, the header (i.e. until _logger...) of this file can also be used as template for Python modules.

Note: This skeleton file can be safely removed if not needed!

sigma0.skeleton.**fib**(*n*)
Fibonacci example function

Parameters **n** (*int*) – integer

Returns n-th Fibonacci number

Return type int

sigma0.skeleton.**main**(*args*)
Main entry point allowing external calls

Parameters **args** (*[str]*) – command line parameter list

sigma0.skeleton.**parse_args**(*args*)
Parse command line parameters

Parameters **args** (*[str]*) – command line parameters as list of strings

Returns command line parameters namespace

Return type argparse.Namespace

sigma0.skeleton.**run**()
Entry point for console_scripts

sigma0.skeleton.**setup_logging**(*loglevel*)
Setup basic logging

Parameters **loglevel** (*int*) – minimum loglevel for emitting messages

Module contents

CHAPTER 2

Indices and tables

- genindex
- modindex
- search

Python Module Index

S

sigma0, 5
sigma0.oh_model, 4
sigma0.skeleton, 5

Index

F

fib() (in module sigma0.skeleton), 5

G

gammah() (in module sigma0.oh_model), 4
gammav() (in module sigma0.oh_model), 4

M

main() (in module sigma0.skeleton), 5

P

parse_args() (in module sigma0.skeleton), 5

R

run() (in module sigma0.skeleton), 5

S

setup_logging() (in module sigma0.skeleton), 5
sigma0 (module), 5
sigma0.oh_model (module), 4
sigma0.skeleton (module), 5
sigma0_bare() (in module sigma0.oh_model), 4