n-sphere Documentation

Release unknown

Wai-Shing Luk

Dec 24, 2020

Contents

This is the documentation of **n-sphere**.

Note: This is the main page of your project's Sphinx documentation. It is formatted in reStructuredText. Add additional pages by creating rst-files in `docs` and adding them to the toctree below. Use then references in order to link them from this page, e.g. *Contributors* and *Changelog*.

It is also possible to refer to the documentation of other Python packages with the Python domain syntax. By default you can reference the documentation of Sphinx, Python, NumPy, SciPy, matplotlib, Pandas, Scikit-Learn. You can add more by extending the `intersphinx_mapping` in your Sphinx's `conf.py`.

The pretty useful extension autodoc is activated by default and lets you include documentation from docstrings. Docstrings can be written in Google style (recommended!), NumPy style and classical style.

CHAPTER 1

Contents

1.1 License

1.2 Contributors

- Wai-Shing Luk <luk036@gmail.com>

1.3 Changelog

1.3.1 Version 0.1

- Feature A added

- FIX: nasty bug #1729 fixed
- add your changes here!

1.4 n_sphere

1.4.1 n_sphere package

Submodules

n_sphere.discrep_2 module

n_sphere.discrep_2.**discrep_2**(*K*, *X*)
dispersion measure

Parameters

- **K** (*[type]*) – [description]
- **X** (*[type]*) – [description]

Returns dispersion

Return type float

n_sphere.halton_n module

class n_sphere.halton_n.**halton**(*b*)
Bases: object

Generate base-b Halton sequence

Parameters

- **n** (*int*) – [description]
- **b** (*[int]*) – sequence base, integer exceeding 1

Returns base-b low discrepancy sequence

Return type ([float])

class n_sphere.halton_n.**halton_n**(*n*, *b*)
Bases: object

Generate base-b Halton sequence

Parameters

- **n** (*int*) – [description]
- **b** (*[int]*) – sequence base, integer exceeding 1

Returns base-b low discrepancy sequence

Return type ([float])

n_sphere.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 lines in the [options.entry_points] section in setup.cfg:

console_scripts = fibonacci = n_sphere.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!

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

Parameters **n** (*int*) – integer

Returns n-th Fibonacci number

Return type int

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

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

n_sphere.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

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

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

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

n_sphere.sphere module

class n_sphere.sphere.**circle**(*base=2*)
Bases: object

Generate Circle Halton sequence 0,..,k

Parameters **k** (*int*) – maximum sequence index, non-negative integer

Keyword Arguments **base** (*int*) – [description] (default: {2})

Returns base-b low discrepancy sequence

Return type ([float])

class n_sphere.sphere.**sphere**(*b*)
Bases: object

Generate Sphere Halton sequence 0,..,k

Parameters **k** (*int*) – maximum sequence index, non-negative integer

Keyword Arguments **b** (*[int]*) – sequence base, integer exceeding 1

Returns base-b low discrepancy sequence

Return type ([float])

n_sphere.sphere3 module

class n_sphere.sphere3.**sphere3**(*b*)
Bases: object

Generate Sphere-3 Halton sequence

Parameters **k** (*int*) – maximum sequence index, non-negative integer

Keyword Arguments **b** (*[int]*) – sequence base, integer exceeding 1

Returns base-b low discrepancy sequence

Return type ([float])

class n_sphere.sphere3.**sphere3_hopf**(*b*)
Bases: object

sphere3_hopf Halton sequence INPUTS : k - maximum sequence index, non-negative integer

b - sequence base, integer exceeding 1

n_sphere.sphere_n module

class n_sphere.sphere_n.**cylin_n**(*n*, *b*)
Bases: object

Generate using cylindrical coordinate method

Parameters

- **k** (*int*) – maximum sequence index, non-negative integer
- **n** (*int*) – [description]
- **b** (*[int]*) – sequence base, integer exceeding 1

Returns base-b low discrepancy sequence

Return type ([float])

n_sphere.sphere_n.**int_sin_power**(*n*, *x*)
Evaluate integral sin^n(x) dx

Parameters

- **n** (*int*) – power
- **x** (*float*) – [description]

Returns [description]

Return type float

class n_sphere.sphere_n.**sphere_n**(*n*, *b*)
Bases: object

Generate Sphere-3 Halton sequence

Parameters **k** (*int*) – maximum sequence index, non-negative integer

Keyword Arguments **b** (*[int]*) – sequence base, integer exceeding 1

Returns base-b low discrepancy sequence

Return type ([float])

n_sphere.vdcorput module

n_sphere.vdcorput.**vdc**(*n*, *base=2*)
[summary]

Parameters **n** (*int*) – number

Keyword Arguments **base** (*int*) – [description] (default: {2})

Returns [description]

Return type int

class n_sphere.vdcorput.**vdcorput**(*base=2*)
Bases: object

n_sphere.vdcorput.**vdcorput_co**(*k*, *base=2*)
[summary]

Parameters **k** (*int*) – number of points

Keyword Arguments **base** (*int*) – [description] (default: {2})

Returns [description]

Return type int

Module contents

CHAPTER 2

Indices and tables

- genindex
- modindex
- search

Python Module Index

n

Index