



# Snakes on a Spaceship Survival of the Python

1. Tutorial: Building documentation - Angeline G. Burrell
2. aurora-asi-lib – Mike Shumko
3. pysat 3.0 – Russell Stoneback  
HIME – Doğa Ozturk
4. sami2py – Jeff Klenzing
5. IGRF – Leslie Lamarche and Ashley Smith
6. Community Discussion – You!



CEDAR 2021 - 21 June 2021

[http://cedarweb.vsp.ucar.edu/wiki/index.php/2021\\_Workshop:Python\\_for\\_Space\\_Science](http://cedarweb.vsp.ucar.edu/wiki/index.php/2021_Workshop:Python_for_Space_Science)



# Building Documentation

Angeline G. Burrell

U.S. Naval Research Laboratory  
Space Science Division

CEDAR 2021 - 21 June 2021



# Documenting your package

- Include docstrings
- Choose a documentation builder and language
- Include useful information and examples
- Community documentation

# Docstrings

- Provides the bones of your documentation
- PEP-257 defines Python docstring conventions
- Keeping a consistent style will improve legibility
  - Balance html rendering with ASCII legibility
  - numpydoc standard is common: <https://numpydoc.readthedocs.io/en/latest/format.html>
  - Linters can identify non-uniform areas (pydocstyle, flake8-docstrings)

```
$ > ocbpy.ocb_correction.elliptical?
```

```
53 def elliptical(mlt, instrument='ampere', method='median'):
54     """ Return the results of an elliptical correction to the data boundary
55
56     Parameters
57     -----
58     mlt : (float or array-like)
59         Magnetic local time in hours
60     instrument : (str)
61         Data set's instrument name (default='ampere')
62     method : (str)
63         Method used to calculate the elliptical correction, accepts
64         'median' or 'gaussian'. (default='median')
65
66     Returns
67     -----
68     r_corr : (float or array-like)
69         Radius correction in degrees at this MLT
70
71     References
72     -----
73     Burrell, A. G. et al.: AMPERE Polar Cap Boundaries, Ann. Geophys., 38,
74     481-490, doi:10.5194/angeo-38-481-2020, 2020.
75
76     """
```

# Building Documentation

- Create a home for documentation files
- Choose a language and generator
  - reStructuredText (reST; file.rst)
    - plain text markup syntax
    - Used by python docstrings, docutils, PyPi
    - Cheat sheet: <https://docutils.sourceforge.io/docs/user/rst/quickref.html>
  - Markdown (file.md)
    - Text-to-HTML conversion tool
    - Used by GitHub, PyPi, accepted by sphinx
    - Cheat sheet: <https://www.markdownguide.org/cheat-sheet/>
  - Sphinx
    - Python package that helps generate and maintain documentation using reST
    - Outputs HTML, LaTeX, ePub, Texinfo, manual pages, and plain text
    - Tests docstrings and web links

aburrell / ocbpy

<> Code Issues 2 Pull requests Actions Projects 2 Wiki Security Insights

main ocbpy / docs /

Go to file Add file

aburrell REL: update release number 24ef7b4 on Nov 17, 2020 History

..		
examples	DOC: boundary plotting example	7 months ago
figures	DOC: vector example	12 months ago
Makefile	Sphinx documentation	4 years ago
citing.rst	REL: update citation	10 months ago
conf.py	REL: update release number	7 months ago
contributing.rst	DOC: added contributing guidelines	10 months ago
example.rst	DOC: example index update	10 months ago
example_dmsp_track.png	DMSP example	4 years ago
index.rst	DOC: updated index	10 months ago
make.bat	Sphinx documentation	4 years ago
ocb_boundary_correction.rst	DOC: boundary correction	13 months ago
ocb_datasets.rst	DOC: data set update	2 years ago
ocb_gridding.rst	Sphinx hotfix	4 years ago
overview.rst	DOC: moved logo	12 months ago
requirements.txt	BUG: logging requirements	2 years ago
supported_datasets.rst	DOC: dataset update	2 years ago

- Overview
- Installation
- Citation guidelines
- Examples
- Change log
- Guide for developers
- Description of package contents

## ocbpy

### Navigation

[Overview](#)

[Citation Guide](#)

[OCB Datasets](#)

[OCB Gridding](#)

[OCB Boundary Correction](#)

[Supported Data Sets](#)

[Examples](#)

- [Initialise an OCBoundary object](#)
- [Retrieve a good OCB record](#)
- [Convert between AACGM and OCB coordinates](#)
- [Loading DMSP SSJ boundary files](#)
- [Load a general data file \(DMSP SSIES\)](#)
- [Grid and scale vector data](#)

[Contributing](#)

### Quick search

 

## Grid and scale vector data

Many space science observations, such as ion drifts, are vectors. The `ocbpy.OCB_scaling.VectorData` class ensures that the vector location, direction, and magnitude are gridded and scaled appropriately.

The example presented here uses SuperDARN data. The example file, `20010214.0100.00.pgr.grd` may be obtained by fitting and then gridding the rawacf file, available from any of the SuperDARN mirrors. FitACF v3.0 was used to create this file. See the [Radar Software Toolkit](#) for more information.

The SuperDARN data may be read in python using `pydarn`. To load this file (or any other grid file), use the following commands.

```
import datetime as dt
import numpy as np
import matplotlib as mpl
import matplotlib.pyplot as plt

import aacgm2
import ocbpy
import pydarn

filename = '20010214.0100.00.pgr.grd'
sd_read = pydarn.SDarnRead(filename)
grd_data = sd_read.read_grid()
print(len(grd_data))

13
```

If you used the same file, there will be 13 grid records. Next, load the OCBs for the northern hemisphere (PGR is a Canadian radar) and the period of time available within this file.

```
stime = dt.datetime(grd_data[0]['start.year'], grd_data[0]['start.mo',
                    grd_data[0]['start.day'], grd_data[0]['start.hou',
                    grd_data[0]['start.minute'],
                    int(np.floor(grd_data[0]['start.second'])))
etime = dt.datetime(grd_data[-1]['start.year'], grd_data[-1]['start.h',
                    grd_data[-1]['start.day'], grd_data[-1]['start.h',
                    grd_data[-1]['start.minute'],
                    int(np.floor(grd_data[-1]['start.second'])))
ocb = ocbpy.OCBboundary.OCBboundary(stime=stime, etime=etime)
print(ocb)
```

# Community Documentation

- Project Description
  - Brief description of package, makes project more searchable
- README
  - First (last?) place people look
- Code of Conduct
  - Norms, rules, and responsibilities of project contributors
  - Guides behaviour and requires transparent enforcement
  - Contributor Covenant: <https://www.contributor-covenant.org>
- Contributing Guidelines
  - Information needed for people to provide useful contributions
- License
  - Required by most institutions, prevents misuse
  - Check out Morin et al., (2012) and <https://choosealicense.com>
- Issue and Pull Request Templates
  - Makes questions easier to answer and debugging simpler
  - Remind contributors to performed all the required steps

## Description

Please include a summary of the change and which issue is fixed. Please also include relevant motivation and context. List any dependencies that are required for this change. Please see `CONTRIBUTING.rst` for more guidelines.

Fixes # (issue)

## Type of change

Please delete options that are not relevant.

- Bug fix (non-breaking change which fixes an issue)
- New feature (non-breaking change which adds functionality)
- Breaking change (fix or feature that would cause existing functionality to not work as expected)
- This change requires a documentation update

## How Has This Been Tested?

Please describe the tests that you ran to verify your changes. Provide instructions so we can reproduce. Please also list any relevant details for your test configuration

```
import ocby
# Do a thing
```

## Test Configuration

- Operating system: (Hal)
- Version number: (Python 1.0)
- Any details about your local setup that are relevant: (e.g., pysat version)

## Checklist:

- Make sure you are merging into the `develop` (not `main`) branch
- My commits are formatted appropriately (following the SciPy/NumPy style)
- My code follows the style guidelines of this project
- I have performed a self-review of my own code
- I have commented my code, particularly in hard-to-understand areas
- I have made corresponding changes to the documentation
- My changes generate no new warnings
- I have added tests that prove my fix is effective or that my feature works
- New and existing unit tests pass locally with my changes
- Any dependent changes have been merged and published in downstream modules
- Add a note to `ChangeLog.rst`, summarising the changes
- Add yourself to `AUTHORS.rst` and `.zenodo.json`

# Acknowledgements

Dr. Burrell is supported by the Chief of Naval Research



[angeline.burrell@nrl.navy.mil](mailto:angeline.burrell@nrl.navy.mil)



- `.zenodo.json`
  - Improve release DOI by supplying Zenodo with supplementary information
  - Useful keywords: “keywords”, “creators”, “license”, “notes”, “references”
- Sphinx documentation check commands
  - `$> sphinx-build -E -b html <doc directory> <target directory>`
  - `$> sphinx-build -b linkcheck <doc directory> <target directory>`