
ne2001

Release 0.0.1

Apr 25, 2017

Contents

1	Overview	1
1.1	Development	1
2	Installation	3
2.1	Dependencies	3
3	Code	5
3.1	NE2001	5
4	Parameters	7
4.1	v0.1	7
5	Usage	11
5.1	Overview	11
5.2	Simple Calculations	11
6	Scripts	13
6.1	ne2001_lb	13
7	Reference	15
7.1	ne2001	15
8	Contributing	17
8.1	Bug reports	17
8.2	Documentation improvements	17
8.3	Feature requests and feedback	17
8.4	Development	18
9	Authors	19
10	Changelog	21
10.1	0.0.1 (‘’)	21
10.2	0.0.2 (‘Not released yet’)	21
11	Indices and tables	23
	Python Module Index	25

CHAPTER 1

Overview

docs	
tests	

Python implementation of Cordes-Lazio's NE2001 Galactic Free Electron Density Model. BEWARE: The code as implemented in FORTRAN differs in several ways from the 2003 posting (astro-ph/0301598: <http://adsabs.harvard.edu/abs/2003astro.ph..1598C>). See the docs for details.

- Free software: BSD license

Development

To run the all tests run:

<code>tox</code>

Note, to combine the coverage data from all the tox environments run:

Windows	<pre>set PYTEST_ADDOPTS=--cov-append tox</pre>
Other	<pre>PYTEST_ADDOPTS=--cov-append tox</pre>

At the command line:

```
pip install ne2001
```

Dependencies

ne2001 depends on the following list of Python packages.

We recommend that you use [Anaconda](#) to install and/or update these packages.

- [python](#) versions 2.7, or 3.4 or later
- [numpy](#) version 1.11 or later
- [astropy](#) version 1.3 or later
- [scipy](#) version 0.17 or later

If you are using Anaconda, you can check the presence of these packages with:

```
conda list "^python|numpy|astropy|scipy"
```


This document describes the code.

NE2001

Overall, the current implementation is a nearly direct port of the FORTRAN code created and kindly provide by Cordes & Lazio ([astro-ph/3101598](#))

It is important to appreciate, however, that the code is not fully consistent with the detailed description provided in the publication. These are the key differences:

- The electron density near the Galactic Center is a constant value
- In NE2001, the thin disk assumes a height of 1.8kpc instead of the A2 parameter
- In NE2001, the Galactic parameters are substantially different than those in the posting

In addition, we have made a few modifications of our own:

- In the original NE2001 code for clumps, $q_2 \leq 5$ is used instead of $q \leq 5$.

CHAPTER 4

Parameters

Components of the Galactic ISM are parameterized in terms of size, central electron density, etc.

Currently, the code reads the parameters from the JSON file `data/ne2001_params.json`. You can modify this directly and future code will allow easier manipulation.

v0.1

Here is the JSON file:

```
{
  "version": "v0.1 by JXP on 05March2017",
  "galactic_center": {
    "F": 60000.0,
    "center": [
      -0.01,
      0.0,
      -0.02
    ],
    "e_density": 10.0,
    "height": 0.026,
    "radius": 0.145
  },
  "ldr": {
    "F": 0.1,
    "center": [
      1.36,
      8.06,
      0.0
    ],
    "e_density": 0.012,
    "ellipsoid": [
      1.5,
      0.75,
```

```
        0.5
      ],
      "theta": -0.4223696789826278
    },
    "lhb": {
      "F": 0.01,
      "center": [
        0.01,
        8.45,
        0.17
      ],
      "cylinder": [
        0.085,
        0.1,
        0.33
      ],
      "e_density": 0.005,
      "theta": 0.2617993877991494
    },
    "loop_in": {
      "F": 0.2,
      "center": [
        -0.045,
        8.4,
        0.07
      ],
      "e_density": 0.0125,
      "radius": 0.12
    },
    "loop_out": {
      "F": 0.01,
      "center": [
        -0.045,
        8.4,
        0.07
      ],
      "e_density": 0.0125,
      "radius": 0.18
    },
    "lsb": {
      "F": 0.01,
      "center": [
        -0.75,
        9.0,
        -0.05
      ],
      "e_density": 0.016,
      "ellipsoid": [
        1.05,
        0.425,
        0.325
      ],
      "theta": 2.426007660272118
    },
    "spiral_arms": {
      "e_density": 1.0,
      "F": 1.0
    }
  },
```

```
"thick_disk": {  
  "F": 0.18,  
  "e_density": 0.034020618556701035,  
  "height": 0.97,  
  "radius": 17.5  
},  
"thin_disk": {  
  "F": 120,  
  "e_density": 0.08,  
  "height": 0.15,  
  "radius": 3.8  
}  
}
```


Overview

To use ne2001 in a project:

```
import ne2001
```

Simple Calculations

Here is an example for calculating the DM along a Galactic sightline to a distance of 20kpc:

```
from ne2001 import io as ne_io
PARAMS = ne_io.read_params()
ne = density.ElectronDensity(**PARAMS)
l, b = 0., 90.
DM = ne.DM(l, b, 20.)
```


This document describes the scripts packaged with ne2001.

ne2001_lb

Calculate quantities along an input Galactic line-of-sight. Main inputs are longitude and latitude. Here is the usage:

```
ne2001_lb -h
usage: ne2001_lb [-h] [-d D] l b

Calculate quantities along a Galactic sightline v0.1

positional arguments:
  l          Galactic longitude (deg)
  b          Galactic latitude (deg)

optional arguments:
  -h, --help  show this help message and exit
  -d D        Distance (kpc)
```

Currently, this script calculates the DM along the sightline to a default distance of 100 kpc.

CHAPTER 7

Reference

ne2001

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

Bug reports

When [reporting a bug](#) please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

Documentation improvements

ne2001 could always use more documentation, whether as part of the official ne2001 docs, in docstrings, or even on the web in blog posts, articles, and such.

Feature requests and feedback

The best way to send feedback is to file an issue at <https://github.com/benbaror/ne2001/issues>.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that code contributions are welcome :)

Development

To set up *ne2001* for local development:

1. Fork *ne2001* (look for the “Fork” button).
2. Clone your fork locally:

```
git clone git@github.com:your_name_here/ne2001.git
```

3. Create a branch for local development:

```
git checkout -b name-of-your-bugfix-or-feature
```

Now you can make your changes locally.

4. When you’re done making changes, run all the checks, doc builder and spell checker with *tox* one command:

```
tox
```

5. Commit your changes and push your branch to GitHub:

```
git add .  
git commit -m "Your detailed description of your changes."  
git push origin name-of-your-bugfix-or-feature
```

6. Submit a pull request through the GitHub website.

Pull Request Guidelines

If you need some code review or feedback while you’re developing the code just make the pull request.

For merging, you should:

1. Include passing tests (run *tox*)¹.
2. Update documentation when there’s new API, functionality etc.
3. Add a note to *CHANGELOG.rst* about the changes.
4. Add yourself to *AUTHORS.rst*.

Tips

To run a subset of tests:

```
tox -e envname -- py.test -k test_myfeature
```

To run all the test environments in *parallel* (you need to `pip install detox`):

```
detox
```

¹ If you don’t have all the necessary python versions available locally you can rely on Travis - it will run the tests for each change you add in the pull request.
It will be slower though ...

CHAPTER 9

Authors

- Ben Bar-Or -
- 10. Xavier Prochaska -

CHAPTER 10

Changelog

0.0.1 (‘’)

- First release on PyPI.

0.0.2 (‘Not released yet’)

- Add simple_lb script
- Add units

CHAPTER 11

Indices and tables

- `genindex`
- `modindex`
- `search`

n

ne2001, [15](#)

N

ne2001 (module), [15](#)