

pyLTR

pyLTR: Python Tools for the LTR

- [pyLTR: Python Tools for the LTR](#)
- [Setup](#)
 - [Requirements](#)
 - [Install](#)
 - [Notes](#)
- [Usage](#)
 - [solarWind.py](#)
 - [Visualization](#)
 - [MIX output](#)
 - [LFM output](#)
 - [Advanced](#)
- [BXZERO, BY_COEF, BZ_COEF](#)

pyLTR is a repository of Python modules & packages useful to the CISM community. This includes:

- Solar Wind file processing
- Standard visualizations used for testing & validation
- Simple data processing



pyLTR ships with LTR version 2.2.0 and newer

After [downloading LTR](#), look in misc/pyLTR

Setup

Requirements

- Python >= 2.4
- Matplotlib >= 0.99.1.1
- NumPy >= 1.4.0
- SciPy >= 0.7.2
- pyHDF >= 0.8.2
- Nose >= 0.11.3 (optional)
- Fortran compiler (optional)

These can be easily installed in several ways:

- [Enthought Python Distribution \(EPD\)](#). There are a few versions of EPD floating around. You need to make sure to obtain one which includes pyhdf. As of v7.1, pyhdf is no longer included with the free "trial" version. However, pyhdf is available in the academic version of 7.3 (registration required). From the [EduDownload page](#), navigate to [request an academic license](#) (*.edu e-mail address required).
- Package manager (yum, apt, macports etc.).
 - macports:

```
sudo port install py27-nose py27-scipy py27-numpy py27-matplotlib py27-ipython
```

- [Python Package Index \(PyPi\)](#) via [easy_install](#). [Click here for instructions](#), or try this command:

```
easy_install ipython matplotlib numpy scipy nose
```

Note: You will need a Fortran compiler and umfpack installed. Find [Mac OSX instructions here](#).

- Build & install from source

pyLTR has been successfully tested on Intel Macs running OSX 10.5 (Leopard) and 10.6 (Snow Leopard), 10.7 (Lion), 32 & 64 bit Linux environments, and IBM AIX systems.

Install

Once the prerequisites are installed:

1. Execute 'install.sh'
2. Set up your environment in one of two ways:
 - a. Append /path/to/pyLTR (*not* /path/to/pyLTR/pyLTR) to \$PYTHONPATH:

```
export PYTHONPATH=/path/to/LTR-para/misc/pyLTR:$PYTHONPATH
```

b. Setup a soft link

```
ln -s /path/to/LTR-para/misc/pyLTR/pyLTR /path/to/lib/python2.x/site-packages/pyLTR
```

Notes

- EPD includes its own Python-2.6 Make sure you modify `install.sh` to point to the correct python. Strange things may happen if you attempt to install with Python-2.5 when the requirements were built using Python-2.6.
- the academic version of EPD is available for 32-bit and 64-bit. When using the 32-bit version, `install.sh` may complain about missing libraries unless you set the `CFLAGS` environment variable to use 32-bit libraries (i.e. for GCC, `export CFLAGS="-m32"`).

Although not required, it's a good idea to install Nose & run `test.sh` to make sure everything has been installed properly (note: Nose should be included with the Enthought Python Distribution).

Usage

pyLTR ships with a variety of convenience scripts in the "scripts" subdirectory. Learn what a script does by passing the "--about" command-line argument, like so:

Execute with the `--help` flag for usage information. For example:

```
LTR-para/misc/pyLTR/scripts/solarWind.py --about
```

Execute with the `--help` flag for usage information. For example:

```
.LTR-para/misc/pyLTR/scripts/solarWind.py --help
```

solarWind.py

Generates TIEGCM (NetCDF) or LFM (SW-SM-DAT) solar wind file. Can convert from a variety of data sources (eg. CCMC, Enlil, LFM, MAS, OMNI).

Sample usage:

```
solarWind.py --filename=OMNI_HRO_1MIN_27372.txt -o TIEGCM
```

When output is LFM (ie. flag `-o LFM`), this will generate coefficients for a linear Bx fit; save the SW-SM-DAT file; and generate a [Python Pickle file](#) storing the data for further analysis.

Visualization

See the "Usage" section above to learn how to use some of the following visualization tools:

MIX output

MIX output can be visualized with a variety of tools:

- 2d Potential plots of polar caps
 - `mixEfieldSummary.py`: 1-panel plot of electric field
 - `mixlonSummary.py`: 6-panel plot of MIX potential, energy, flux, conductances and field-aligned current
- Time series
 - `mixTimeSeries.py` plots hemispheric power, cross polar cap potential and current density

LFM output

Magnetosphere:

- `dst.py`: Disturbance Storm Time (DST) index

Ionosphere:

- `lfmionlonSummary.py`:

Advanced

Want to write your own custom scripts using the pyLTR developers API? Take a look at source code of files from both the "examples" and "scripts" directories.

BXZERO, BY_COEF, BZ_COEF

The solar wind processor computes a linear regression fit to B_x and writes the coefficients `BXZERO`, `BYCOEF`, `BZCOEF`:



Unknown macro: 'latex'

Write these numbers down and use them as input to the corresponding [MakeltSo](#) prompt.



Note on `BXZERO`

`BXZERO` is currently unsupported due to a bug within the LFM. If prompted, set `BXZERO=0.0`. This will be fixed in a future release of the code (refs [#143](#)).