flipchem:
An ion density calculator via Python wrapped C and Fortran

Ashton S. Reimer

Virtual CEDAR Workshop

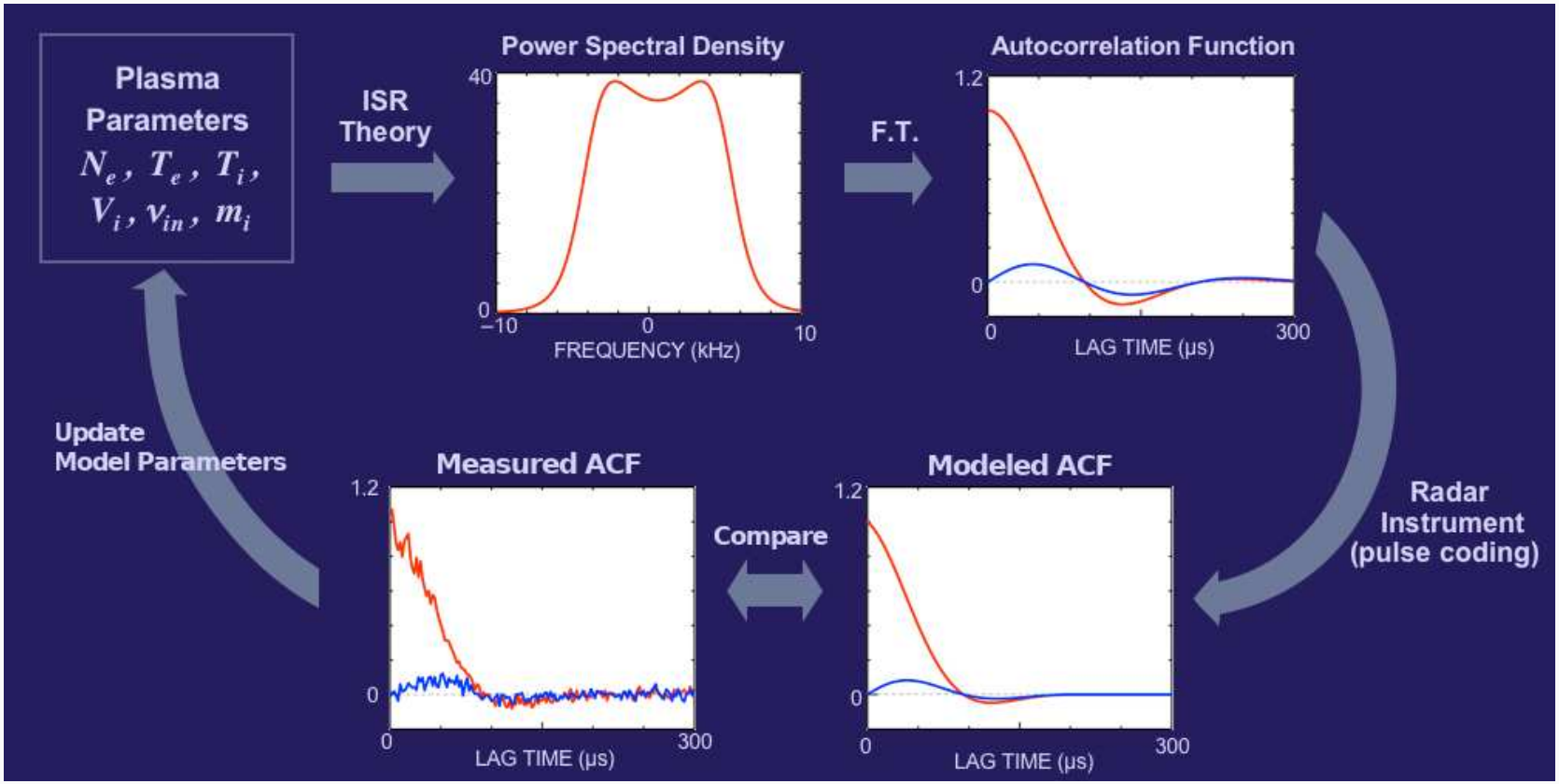
24/6/2021

Center for Geospace Studies, SRI International

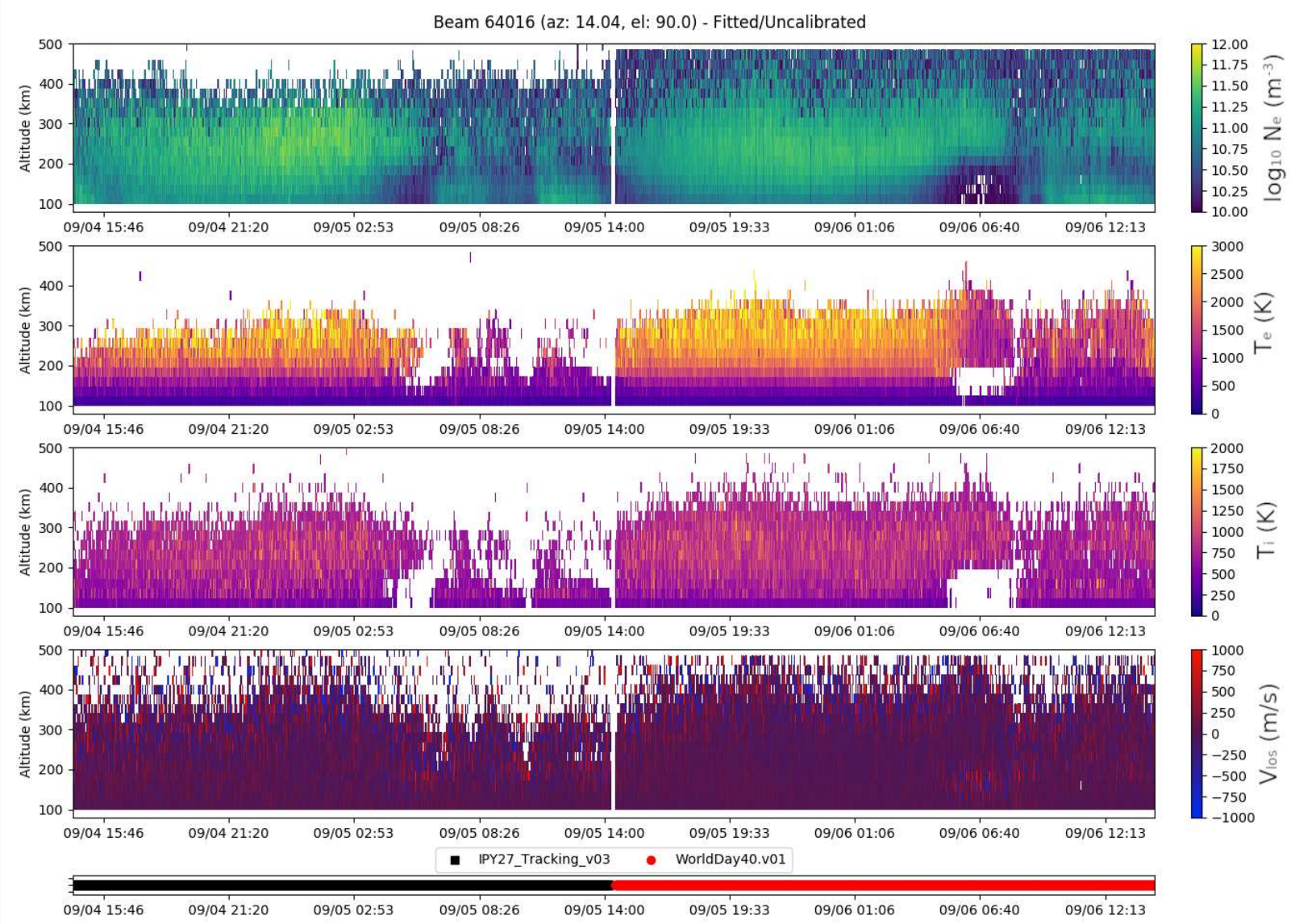
Motivation: Incoherent Scatter Radar

Motivation: Extracting Ionospheric Parameters

Inverse Problem Solution Procedure:

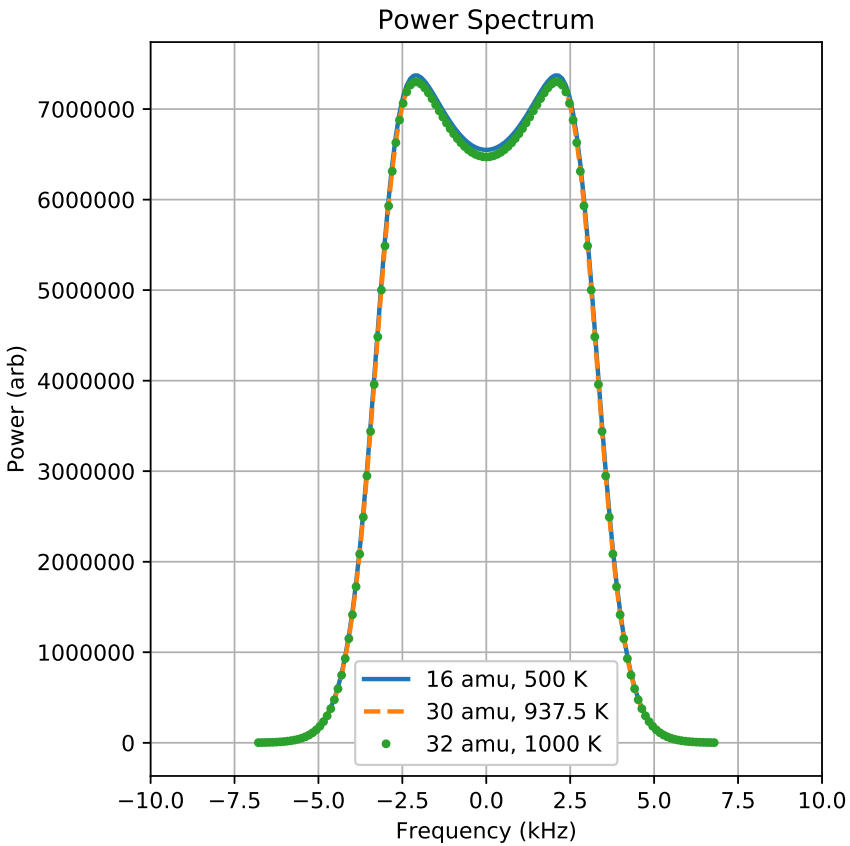


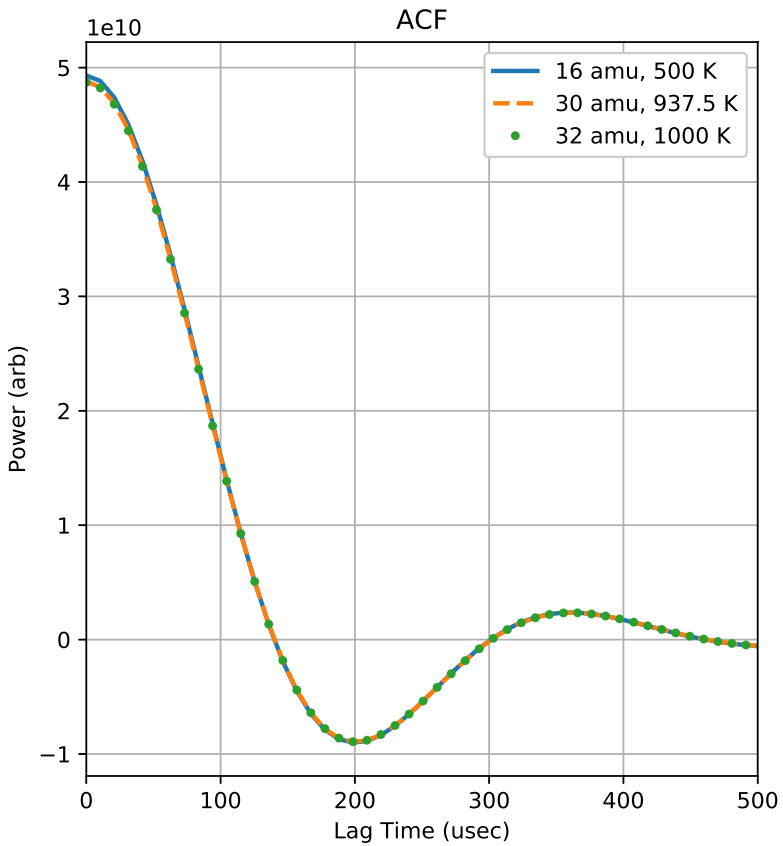
Motivation: Density, Temperature, and Velocity



Motivation: Ion Temperature/Ion Mass Ambiguity

Ion-acoustic peaks depend on $\sqrt{T_i/m_i}$:

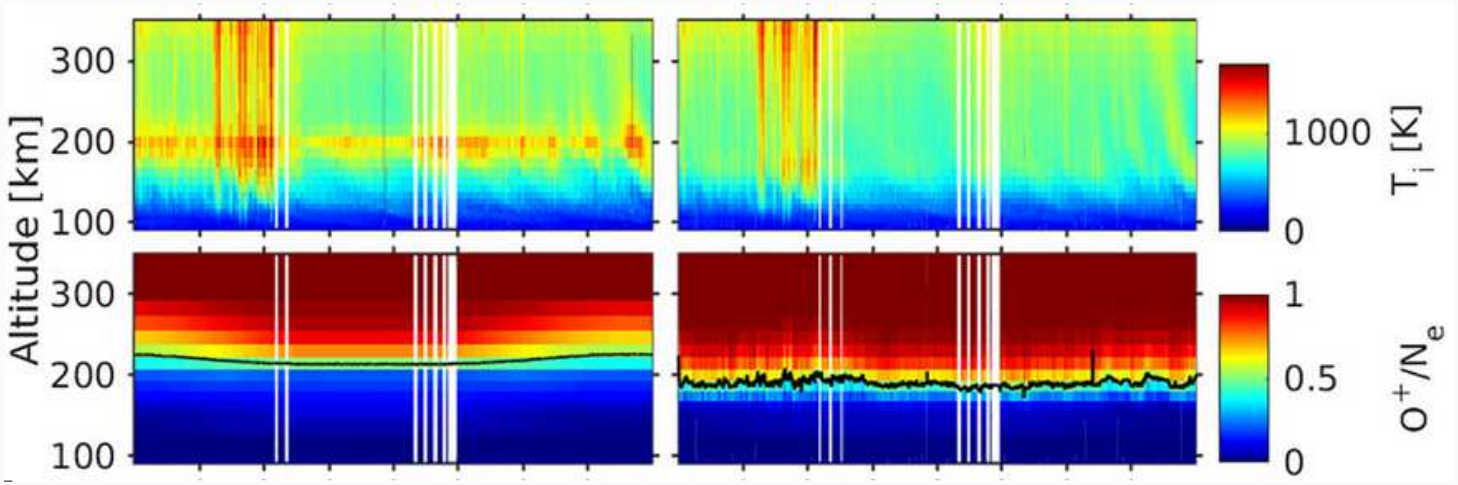




Motivation: Anomalous Ion Temperature

Punchline: If the ion composition is incorrectly specified, the ion temperature is incorrect!

e.g. IRI vs. BAFIM:



Adapted from Figure 2 of Virtanen et al. (2021):
doi:10.1029/2020JA028700

flipchem: What is it?

flipchem: An **easy to use** ion density calculator in Python.

Calculating ion density requires:

- Ion Density Calculator:
  - Richards et al. (2010): `doi:10.1029/2009RS004332`
  - solves for $O^+(^2P)$, $O^+(^2D)$, N_2^+, O^{2+}, NO^+, and N^+ using chemical equilibrium
- NRLMSISe-00:
  - Picone et al. (2002): `doi:10.1029/2002JA009430`
- Geophysical parameter access:
  - `https://amisr.com/geophys_params/`
  - Fuses Penticton F10.7 and Potsdam KP-AP files
  - Reproduces NOAA files (ceased 14 May 2018)

flipchem simplifies obtaining ion densities.

flipchem: Example Usage

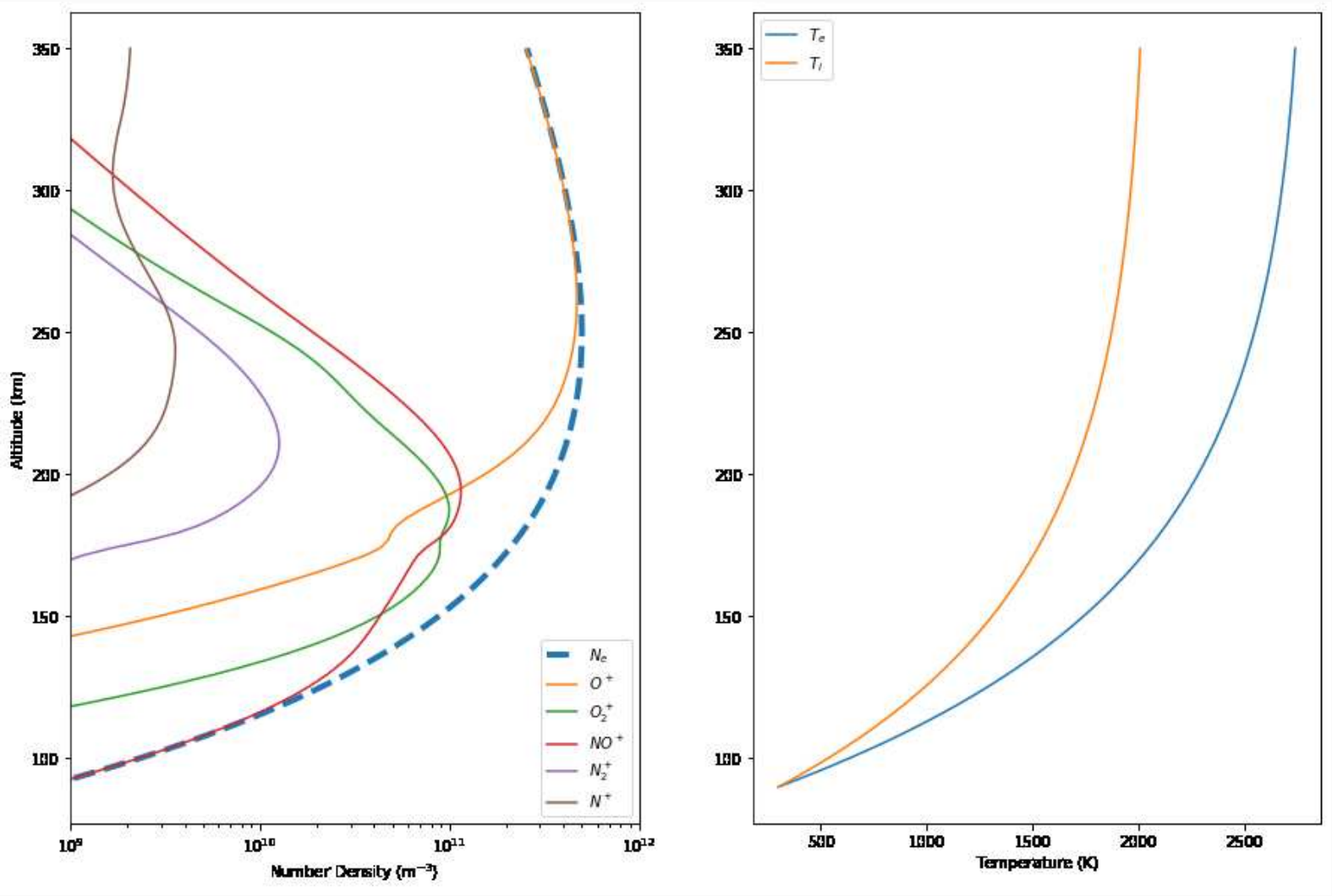
Using flipchem in 4 lines of code:

```
import flipchem

from datetime import datetime

fc = flipchem.Flipchem(datetime(2017,1,4,18))

outputs = fc.get_point(glat,glon,alt,ne,te,ti)
```

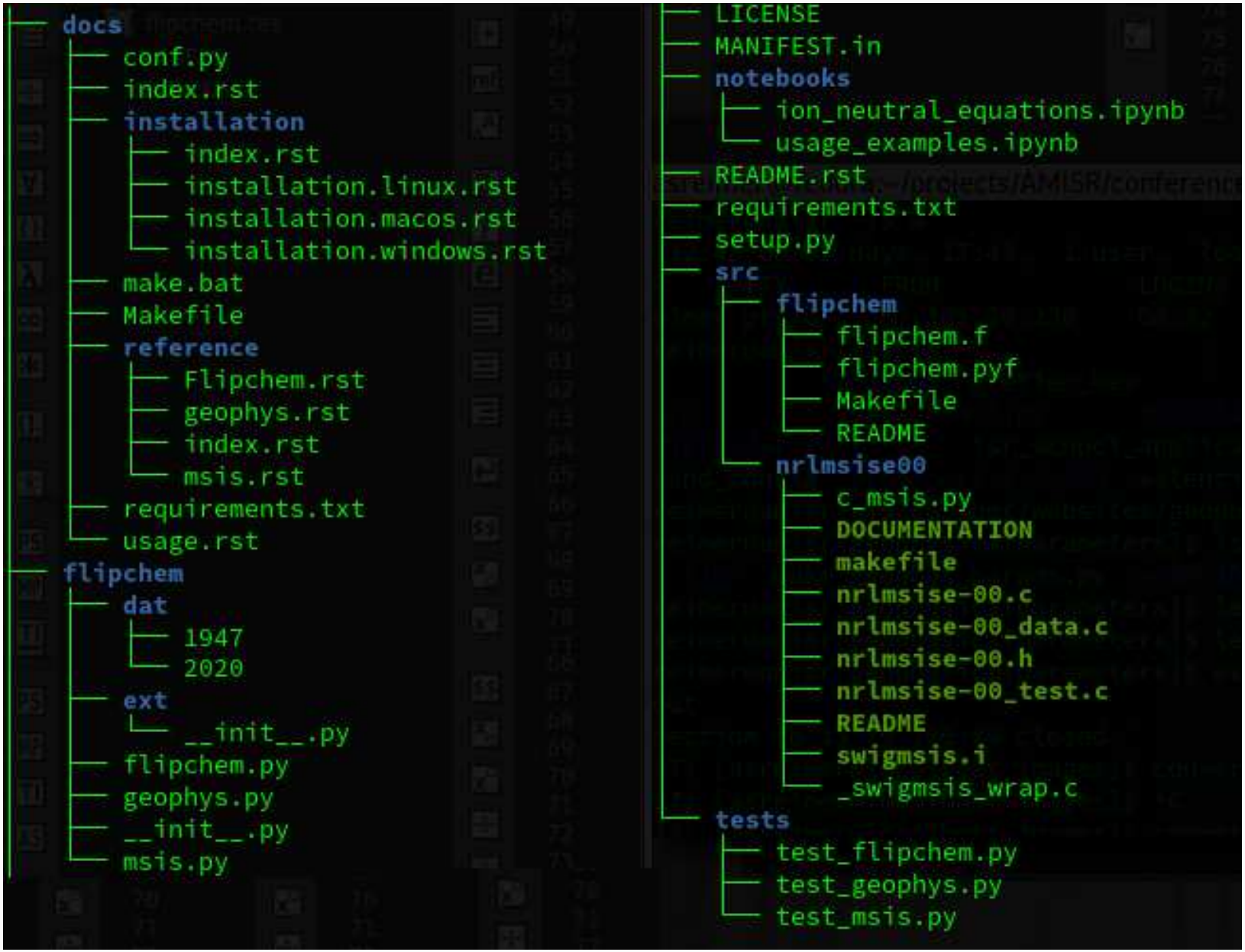
Output includes densities for: $O^+(^2P)$, $O^+(^2D)$, N_2^+, O^{2+}, NO^+, and N^+

flipchem: Example Usage

Altitude profiles:



flipchem: Package Structure



flipchem: Module Implementations

- Geophysical parameter manager (Pure Python)
- SWIG wrapper for NRLMSISe-00 (C)
  - Simplified Wrapper and Interface Generator
  - `swig -python -o _swigmsis_wrap.c swigmsis.i`
- `f2py` wrapper for Ion Density Calculator (fortran)
  - Mixed syntax Fortran (f77 and f90)
  - Manually removed inline comments
  - `f2py -m _f_flipchem -h flipchem.pyf flipchem.f`

Development Decision/Notes

- Fortran flags `-finit-local-zero`
- SWIG - not as easy to use as f2py
  -
```
def _call_library(self,yr,doy,hrUT,alt,glat,glong,ap_array,f107a,f107,ap=-1):

    # input structure
    _msis.nrlmsise_input_year_set(self._inputs,int(yr))
    _msis.nrlmsise_input_doy_set(self._inputs,int(doy))
    _msis.nrlmsise_input_sec_set(self._inputs,_msis.PyFloat_AsDouble(hrUT * 3600))
```
- Naming: preface extensions with “_”
  -
```
from flipchem.ext import chemion as _chemion
from flipchem.ext import getltsza as _getltsza
```
- Docstring convention: Numpy
- Readthedocs: RST and sphinx
  - `sphinx.ext.napoleon`, `sphinx.ext.autodoc`, `sphinx.ext.viewcode`

Current Project Status

Nuts and Bolts:

- Release v2020.2.2: `doi:10.5281/zenodo.3837037`
- Cross platform support: Windows, macOS, Linux
- Readthedocs: `https://flipchem.readthedocs.io`
- GitHub: `https://github.com/amisr/flipchem`
- C.I. via Travis

Usage:

- Testing with ISR fitting software
- Providing collision frequency estimates

What's Next?

Ongoing/Future Work:

- Testing with ISR fitting software
- Incorporate MSIS 2.0
- Migrate C.I. from Travis to GitHub Actions
- Article in JOSS (The Journal of Open Source Software)
  `https://joss.theoj.org/`
- PyPi?