GCMprocpy

A TIE-GCM / WACCM-X post-processing tool

Nikhil Rao

nikhilr@ucar.edu

Jun 2025

This material is based upon work supported by the National Center for Atmospheric Research, which is a major faciligy sponsored by the National Science Foundation under Cooperative Agreement No. 1852977.

What is GCMprocpy?

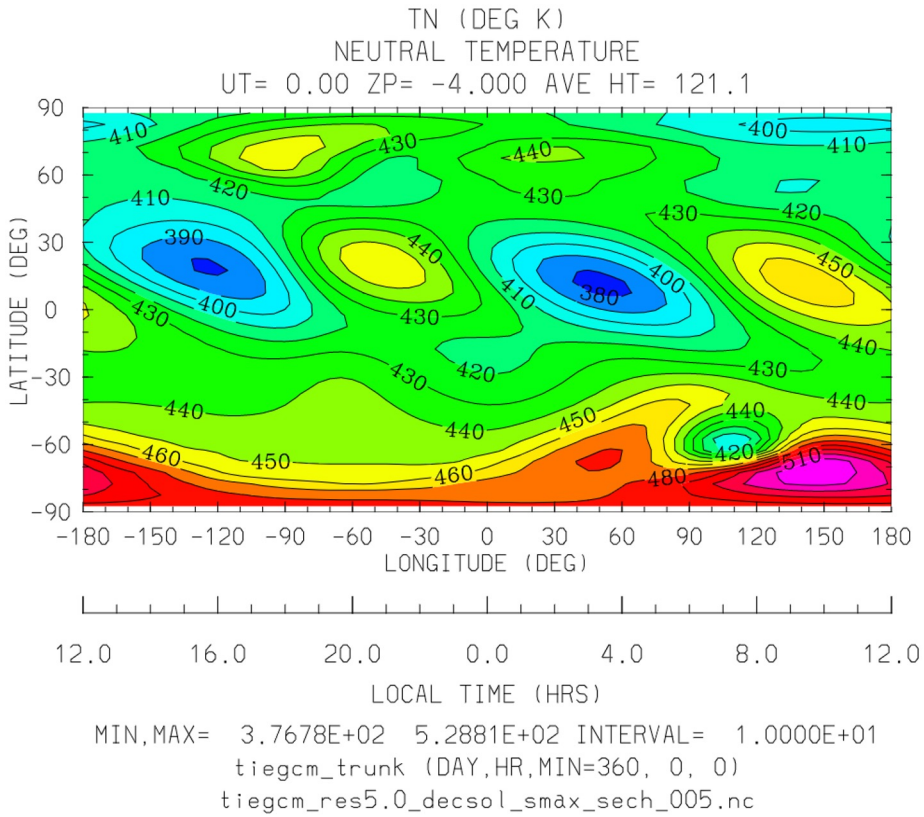
- A Python package designed for post-processing and visualization of outputs from the TIE-GCM and WACCM-X models.
- Facilitates data analysis and plot generation.

TIE-GCM

WACCM -X

Motivation Behind GCMprocpy

- Legacy Tools:
  - tiegcmidl
  - tiegcmprocf90
- Challenges with Legacy Tools:
  - Steep learning curve for new users
  - Limited flexibility and integration with modern data analysis workflows
- Why GCMprocpy?
  - Offers a more user-friendly and versatile platform for researchers
  - Leverages Python's extensive scientific libraries for enhanced data analysis.

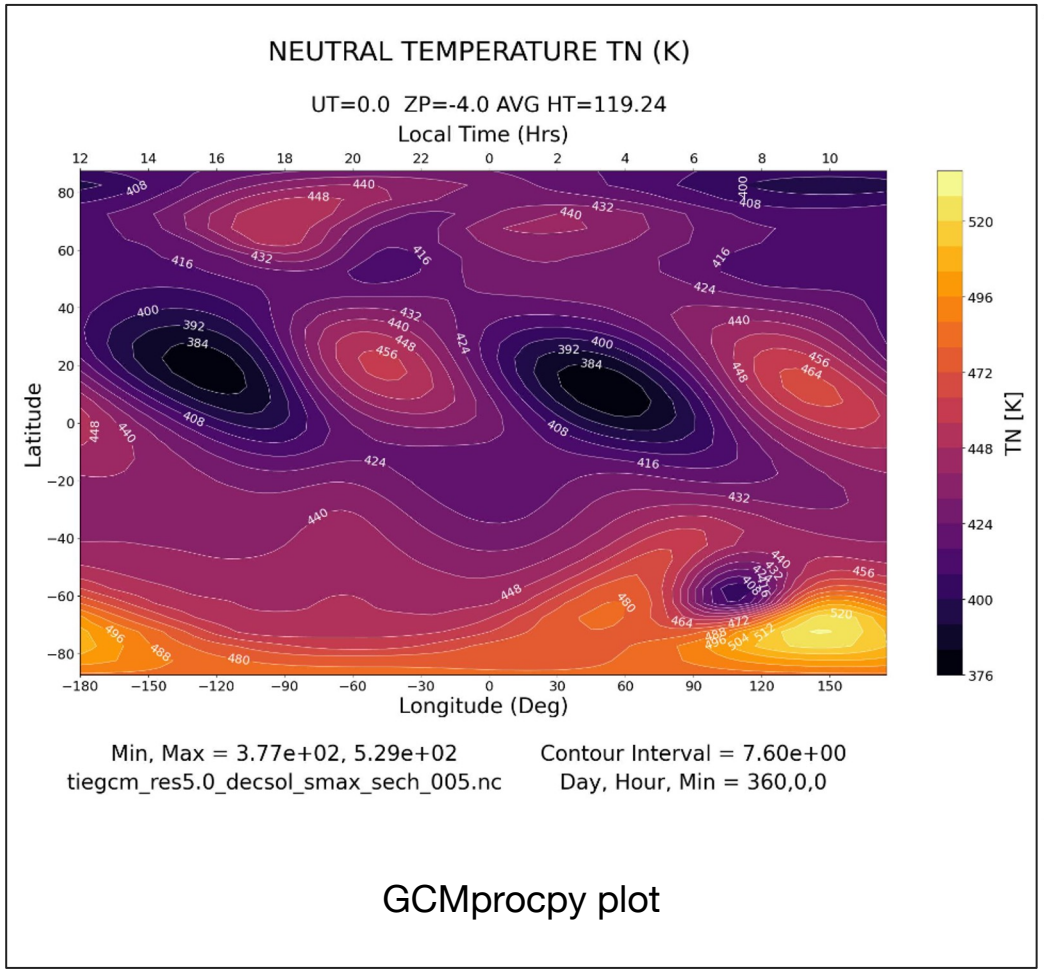


tgcmprocf90 plot

Evolution of GCMprocpy

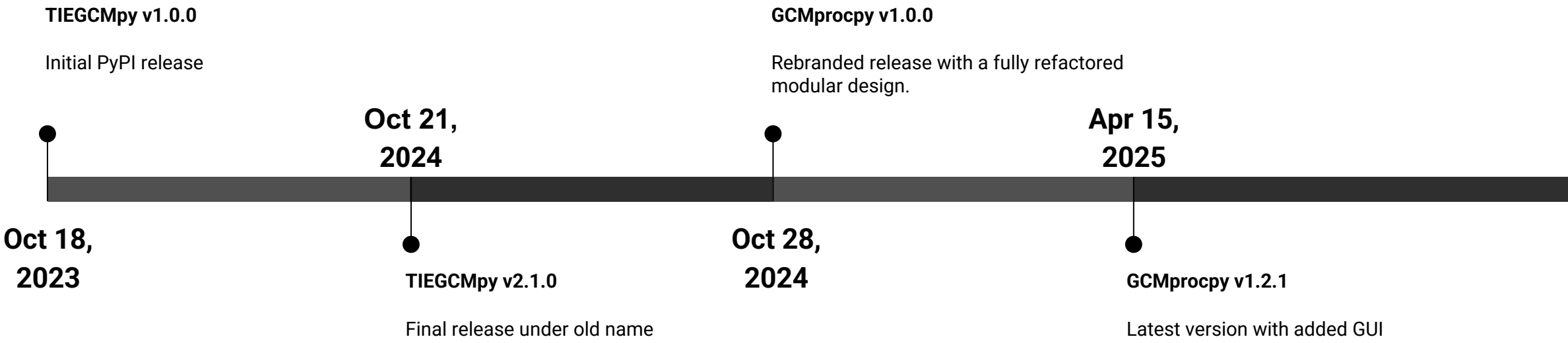
- Origins as TIEGCMpy:
  - Initially developed as TIEGCMpy, a Python 3 tool for post-processing and plotting TIE-GCM outputs.

- Transition to GCMprocpy:
  - Renamed and restructured to GCMprocpy to reflect broader capabilities, including support for multiple models like WACCM-X.
  - Adopted a modular design with separate components for different functionalities.
  - Added a Graphical User Interface (GUI) for users preferring point-and-click over scripting.



GCMprocpy plot

GCMprocpy Timeline`



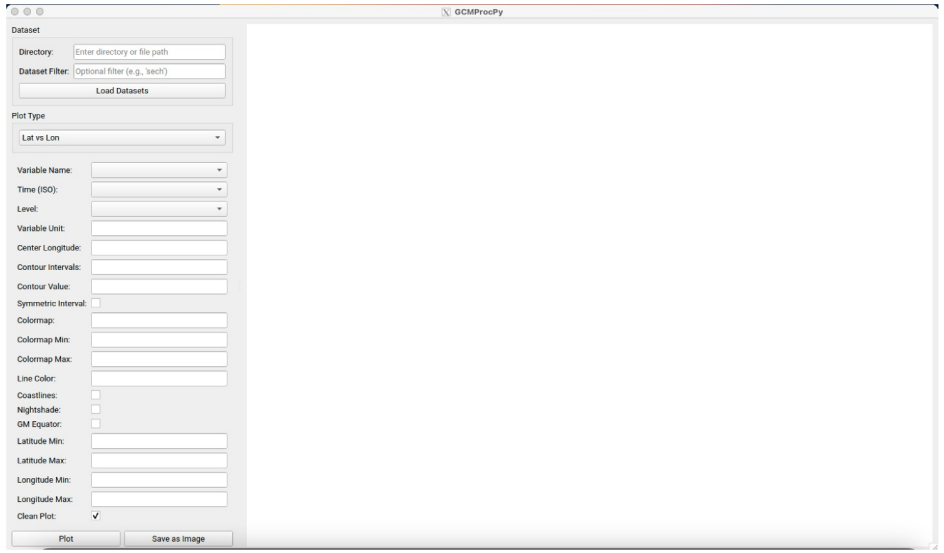
GCMprocpy Features

- Usage Methods
  - Graphical User Interface (GUI) ->
  - Python Function calls (API)

```
datasets = gy.load_datasets(directory, dataset_filter)
variable_name = 'TN'
value_of_mtime = [360, 0, 0, 0]
pressure_level = 4.0
unit_of_variable = 'K'
intervals = 20
plot = gy.plt_lat_lon(datasets, variable_name, mtime=value_of_mtime,
level=pressure_level, variable_unit=unit_of_variable,
contour_intervals=intervals)
```

  - Command line Interface (CLI)

```
gcmprocpy --plot_type lat_lon --directory
/glade/work/nikhilr/work/test_db/2.0/tiegcm_res5.0_decsol_smax/hist -
-dataset_filter sech --output_format jpeg var TN -zp -4.00 -mtime 360 0
0
```



GCMprocpy GUI

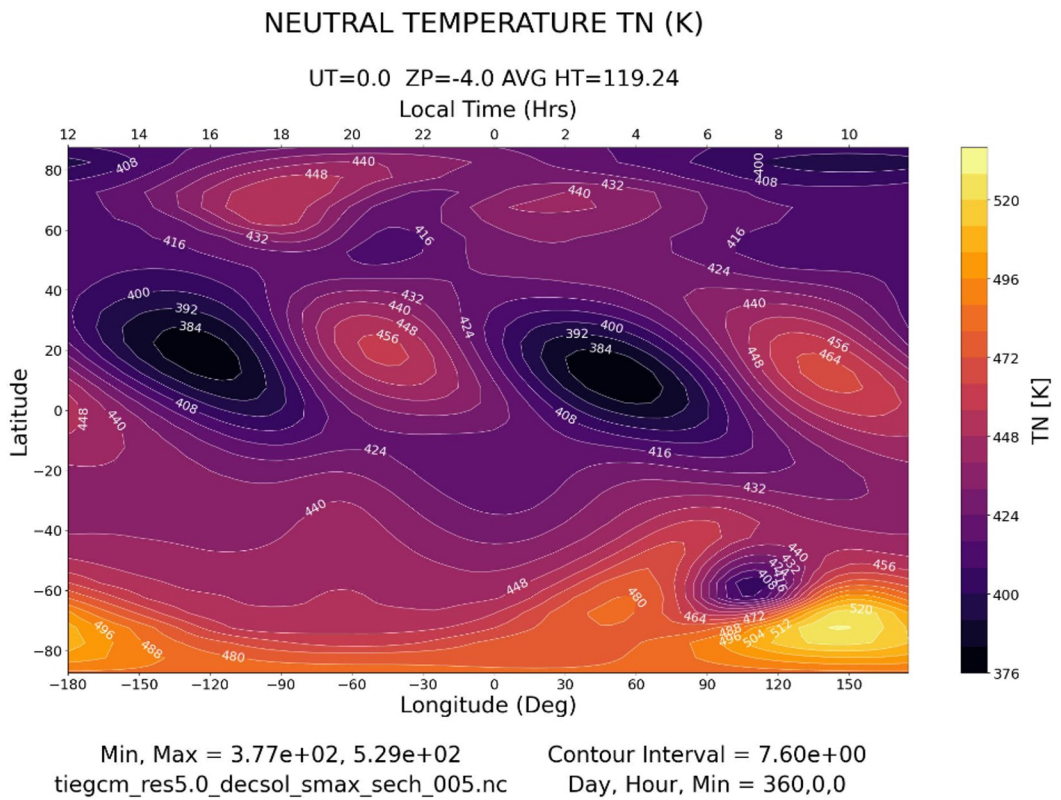
GCMprocpy Features

- Types of functions
  - Data exploration
    - Inspect variables, metadata, and file structures
  - Data manipulation
    - Subset, merge, and transform model outputs
  - Data arrays generation
    - Create structured arrays
  - Emissions calculation
    - Compute emissions
  - Plotting routines
    - Create static/interactive images data visualization
  - Movie routines
    - Create time-lapse animations for dynamic data visualization

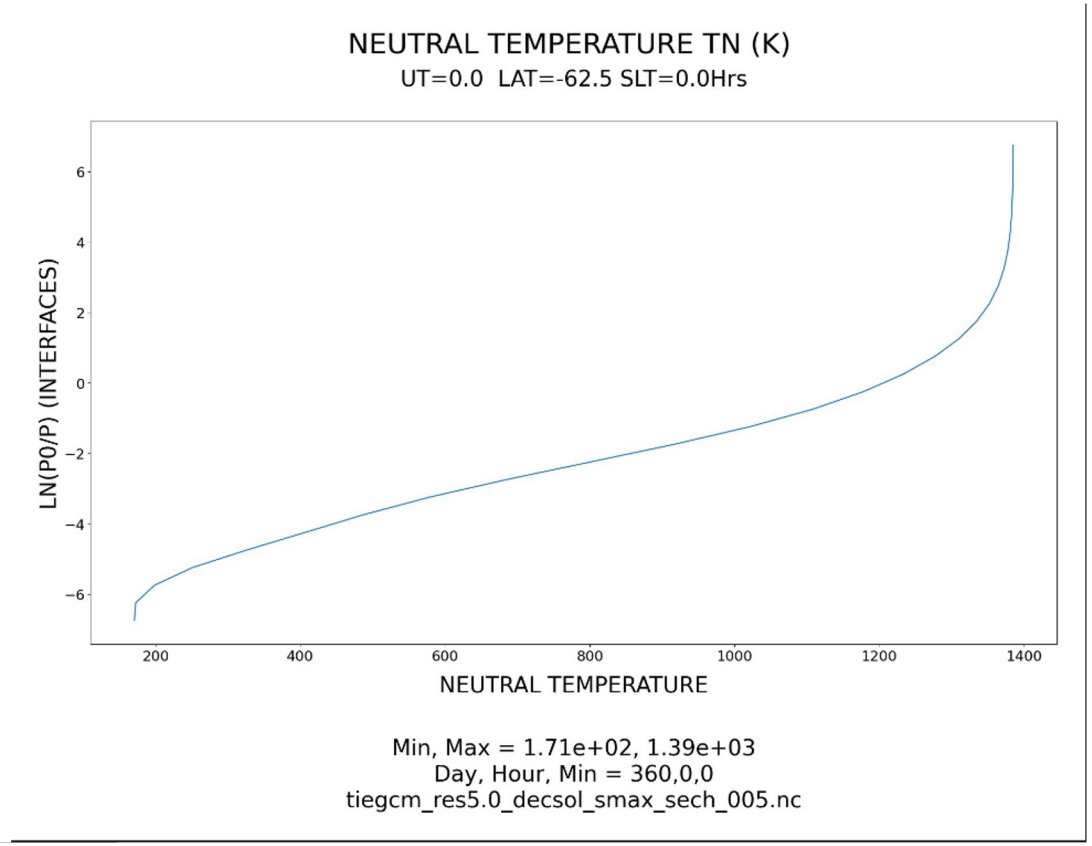
```
[ ]  # List of timestamps in the smax dataset
     gy.time_list(mareq_smax_ds)[0:5]

     [numpy.datetime64('2002-03-21T01:00:00.000000000'),
      numpy.datetime64('2002-03-21T02:00:00.000000000'),
      numpy.datetime64('2002-03-21T03:00:00.000000000'),
      numpy.datetime64('2002-03-21T04:00:00.000000000'),
      numpy.datetime64('2002-03-21T05:00:00.000000000')]
```

Listing timestamps in the dataset

Plot Types

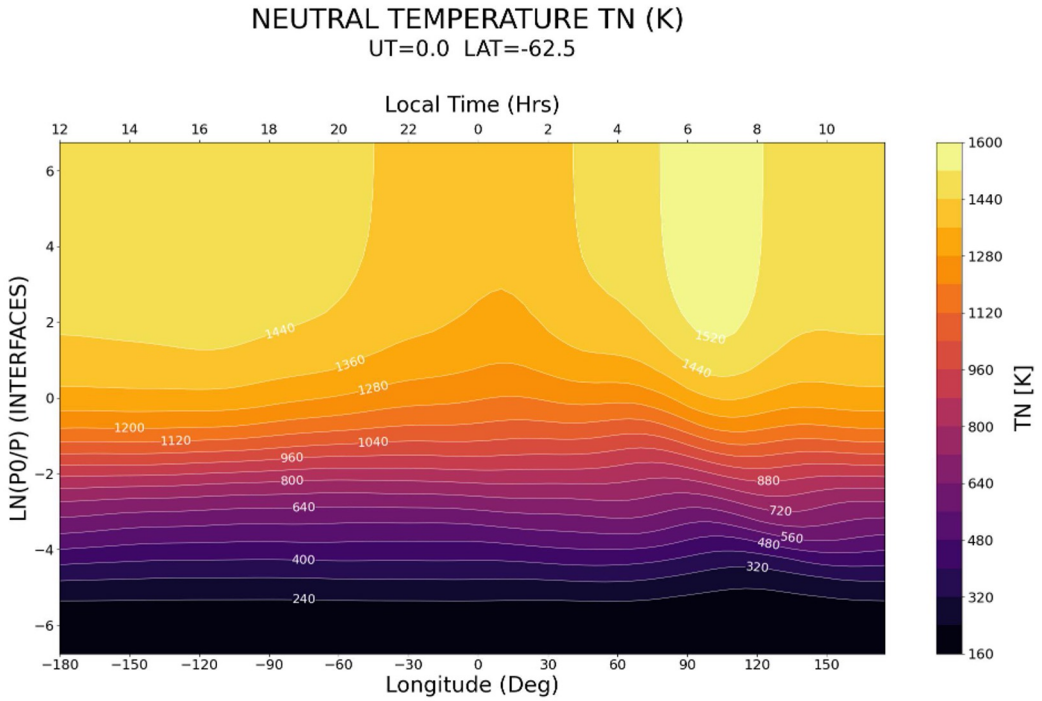


Latitude vs Longitude



Pressure Level vs Variable

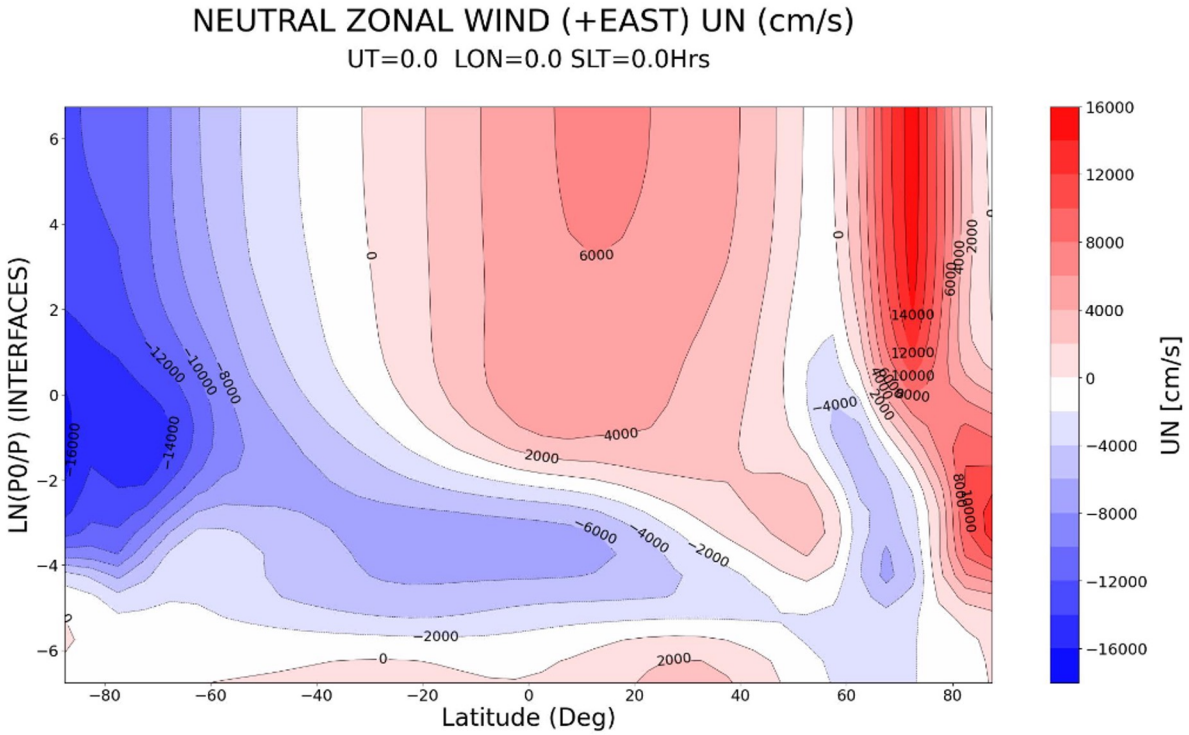
Plot Types



Min, Max = 1.66e+02, 1.58e+03
tiegcm_res5.0_decsol_smax_sech_005.nc

Contour Interval = 7.08e+01
Day, Hour, Min = 360,0,0

Pressure Level vs Longitude

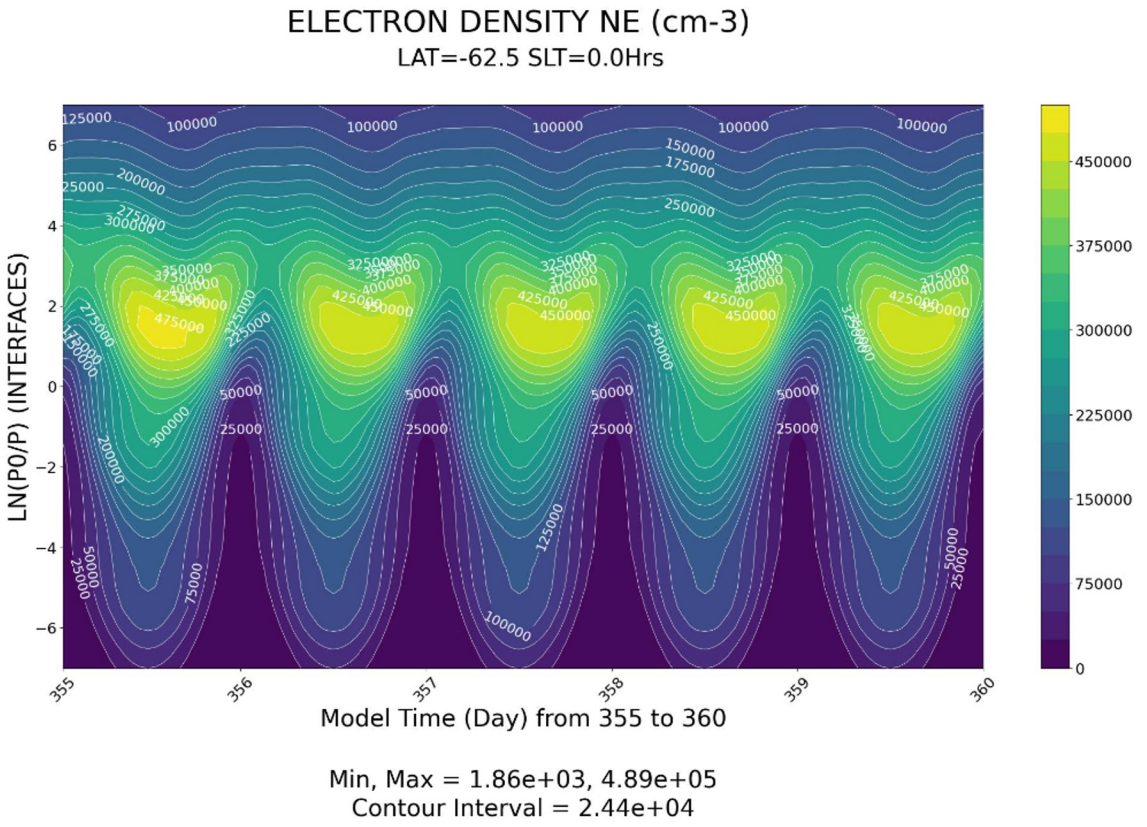


Min, Max = -1.66e+04, 1.53e+04
tiegcm_res5.0_decsol_smax_sech_005.nc

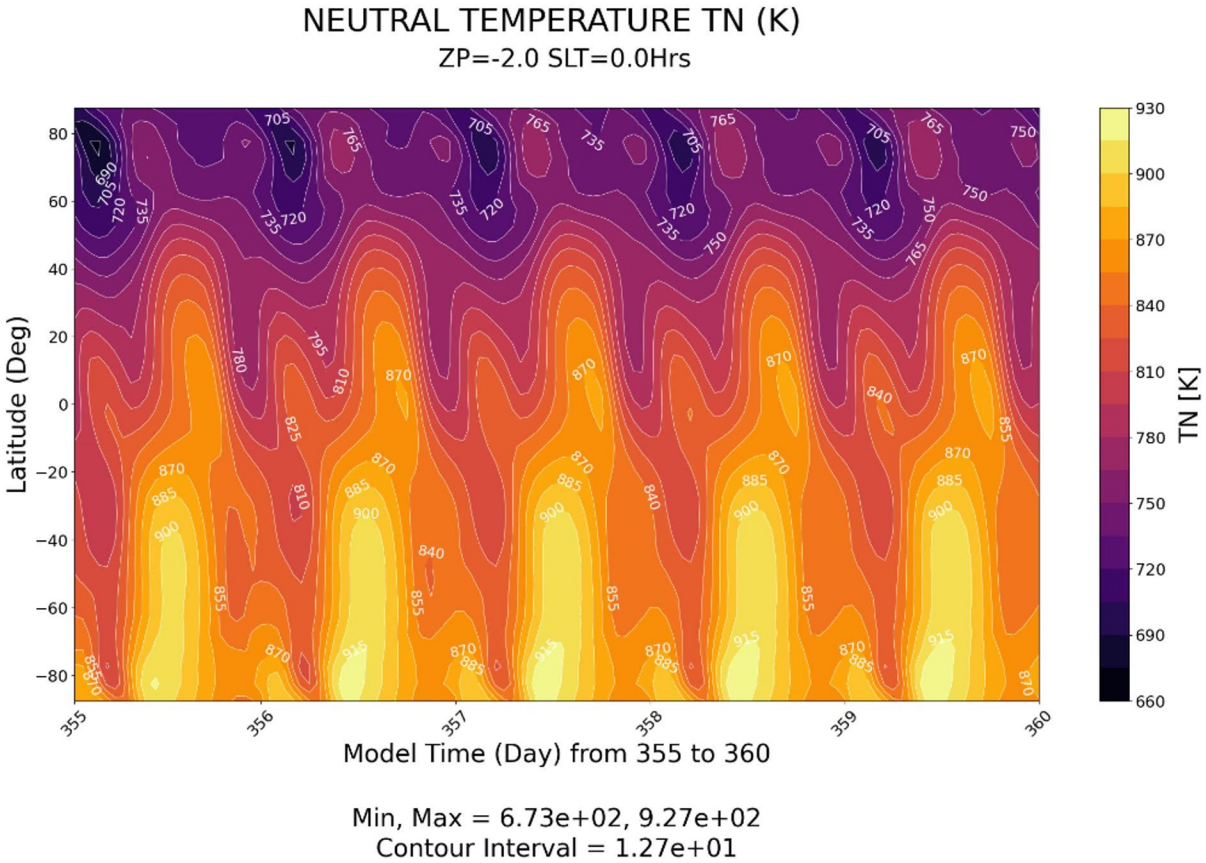
Contour Interval = 1.60e+03
Day, Hour, Min = 360,0,0

Pressure Level vs Latitude

Plot Types



Pressure Level vs Time



Latitude vs Time

GCMprocpy Features

Automated Documentation

- Uses Autodoc, a Sphinx extension that automatically generates documentation from Python docstrings.
- It reduces manual effort and ensures that documentation stays in sync with the codebase
- GCMprocpy uses Google Style docstrings

```
[docs]
def dim_list(datasets):
    """
    Retrieves a sorted list of unique dimension names across all datasets.

    Args:
        datasets (list of tuples): A list of tuples, where each tuple contains an xarray datase

    Returns:
        list: A sorted list of unique dimension names across all datasets.
    """

    unique_dims = set()
```

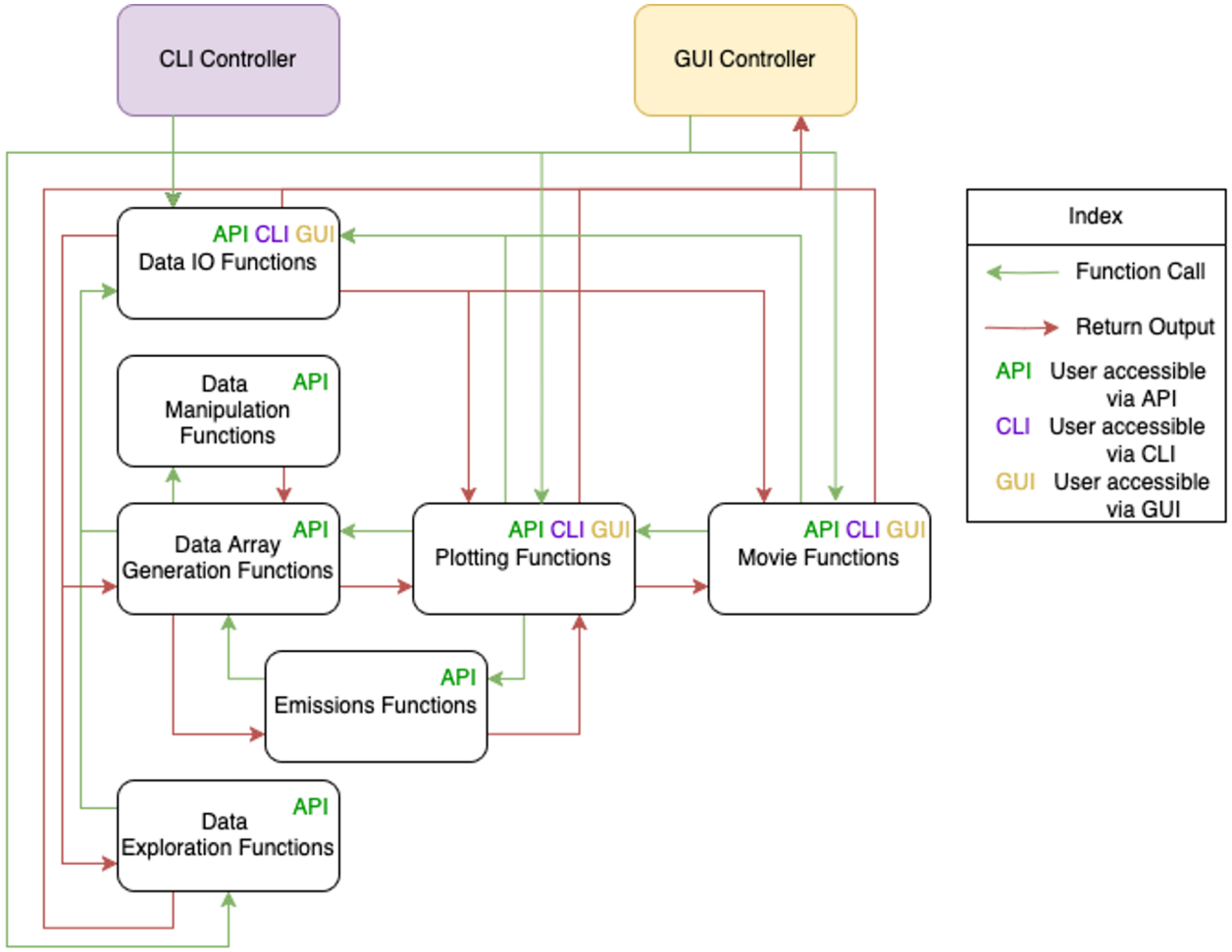
gcmprocpy.data_parse.dim_list(*datasets*) [source]

Retrieves a sorted list of unique dimension names across all datasets.

Parameters:	**datasets** (*list of tuples*) – A list of tuples, where each tuple contains an xarray dataset and its filename.
Returns:	A sorted list of unique dimension names across all datasets.
Return type:	list

Docstring to documentation on RTD

GCMprocpy Stucture`



GCMprocpy Future Development

- Building a robust testing framework
- Enabling movie routines on GUI
- Adding polar plots to plotting routines
- Adding additional emissions calculations
- Performance optimization to data array creation

GCMprocpy Links`

Google Collab Tutorial

Github

https://github.com/NCAR/gcmprocpy/

RTD Documentation

https://gcmprocpy.readthedocs.io/en/latest/index.html

Thank you

Google Collab Tutorial

Github
https://github.com/NCAR/gcmprocpy/

RTD Documentation
https://gcmprocpy.readthedocs.io/en/latest/index.html