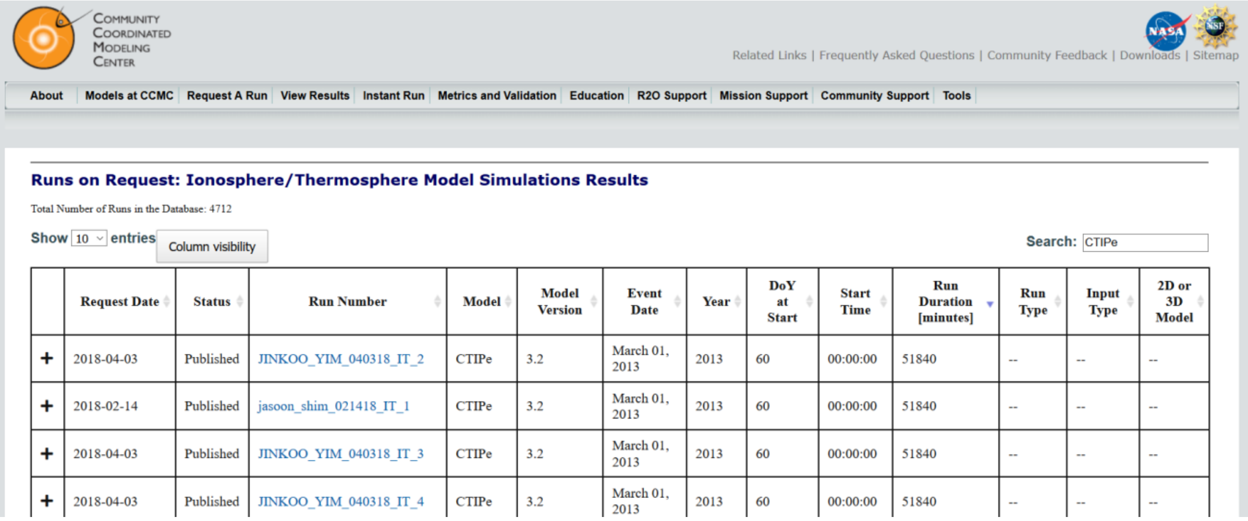
Instructions for calling CTIPe data from fortran (slow method – via file output)
Rebecca Ringuette, 28 Apr 2021

The instructions for calling CTIPe data from fortran are outlined below. Instructions and notes are indicated by the bulleted points, while commands are given with the dash-style bullets. In the current state, the python scripts are specifically designed for the Geodyn software and the CTIPe model, but can be extended to other models and even more easily to other software calls, depending upon the requirements. These directions enable a slow interface between fortran and python through a result file, which is projected to take roughly four hours to execute for a 2-week arc (15 min cadence). A faster interface is under investigation via ForPy. If there are issues with these instructions, please contact me at Rebecca.ringuette@nasa.gov.

1. Create conda environment with kamodo installed. A miniconda environment will likely be fine, but I haven’t tested it.
    - cd (to dir where you want things to run)
    - cp (full path)/Kamodo-Master .
    - conda create -n FortranKamodo python=3.7.9
    - conda install -n FortranKamodo -c conda-forge plotly sympy scipy pytest pandas hydra-core requests ipython
    - conda activate FortranKamodo
    - pip install python-forge
    - pip install netCDF4
    - pip install ./Kamodo-Master
    - conda deactivate (when done)
    - Use the Kamodo-Master dir I send you to avoid possible issues with new versions
    - Test your kamodo installation by executing the following commands with the environment activated:
    - ipython
    - import kamodo
    - If no error occurs, your installation is good. You may exit ipython (exit()) and continue. Otherwise, let me know if you need help.
2. Get desired data from ccmc website for CTIPe model.

- https://ccmc.gsfc.nasa.gov/ungrouped/IT/IT_db.php



Runs on Request: Ionosphere/Thermosphere Model Simulations Results

Total Number of Runs in the Database: 4712

Search: CTIPe

	Request Date	Status	Run Number	Model	Model Version	Event Date	Year	DoY at Start	Start Time	Run Duration [minutes]	Run Type	Input Type	2D or 3D Model
+	2018-04-03	Published	JINKOO_YIM_040318_IT_2	CTIPe	3.2	March 01, 2013	2013	60	00:00:00	51840	--	--	--
+	2018-02-14	Published	jasoon_shim_021418_IT_1	CTIPe	3.2	March 01, 2013	2013	60	00:00:00	51840	--	--	--
+	2018-04-03	Published	JINKOO_YIM_040318_IT_3	CTIPe	3.2	March 01, 2013	2013	60	00:00:00	51840	--	--	--
+	2018-04-03	Published	JINKOO_YIM_040318_IT_4	CTIPe	3.2	March 01, 2013	2013	60	00:00:00	51840	--	--	--

- On the page linked above, I searched for CTIPe and then sorted by run duration in decreasing order. If you follow the link, it sends you to a page for the run. However, I don’t see how to retrieve the data files for the run.
- Select one that is long enough (at least 2 weeks long) and work with Lutz to get the data. Send the data to me as well for comparison and testing.

3. Prepare the data for use
    - With the conda environment activated and within ipython, paste in and execute the following commands:
    - – from kamodo.readers import ctipe_data_wrapper as DW
    - – import glob
    - – file_dir = ‘*file_dir*’ #copy over the directory name where the data files are stored
    - – files = glob.glob(file_dir+’*plot-density.nc’)
    - – for f in files: DW(f)
    - – exit()
    - This step speeds up the data processing in the kamodo reader. Don’t worry about the other file types from the CTIPe output. The data wrapper function takes care of it.
4. Check satellite flythrough functionality
    - With the conda environment activated, execute the following command in the conda command prompt. *File_dir* is the directory where the data is located, rho is the variable name, and ilev means the variable depends on the pressure level for the CTIPe model. The numbers are the satellite timestamp in UTC since Jan 1, 1970 (*sat_time*), the satellite altitude above the ground (*sat_height*, in km), and the satellite latitude and longitude (*sat_lat* and *sat_lon*).
    - – python C:\Users\rringuet\Kamodo_WinDev1\Kamodo-master\kamodo\readers\CTIPe_wrapper_fortrancmd.py *file_dir* rho ilev 1426637500.0 400.0 -25.0 10.0

- This program finds the value of the density and the density derivative at the time and location specified for the data files in the given directory. The two values are printed in a simple file called results.txt in the same directory as the data files, with the two values separated by a comma. The density, rho (kg/m^3), is given first, and rho_dz is given second, both printed to the 15th decimal place.
- Note: The results.txt file is overwritten each time the program is run.
- If the results.txt file is produced without error, then you can continue. Otherwise, let me know if there are issues.

5. Add the necessary lines to the GeoDyn code
    - You will need to execute a command line call from within Geodyn. I believe the name is EXECUTE_COMMAND_LINE (https://gcc.gnu.org/onlinedocs/gfortran/EXECUTE_005fCOMMAND_005fLINE.html)
    - The command to execute is below. *File_dir* and *sat_time* are as described before. *Sat_height* is the altitude of the satellite above the ground (in km), and *sat_lat* and *sat_lon* are the latitude and longitude of the satellite in degrees.

    -
      ```
      python C:\Users\rringuet\Kamodo_WinDev1\Kamodo-master\kamodo\readers\CTIPe_wrapper_fortrancmd.py file_dir rho ilev sat_time sat_height sat_lat sat_lon
      ```
    - After the command is run, you will need a few commands in fortran to read in the two data values from the results.txt file.
    - Note: You must run the Geodyn code with the conda environment activated for the python code to run correctly.

If you have any questions, please let me know. Good luck!