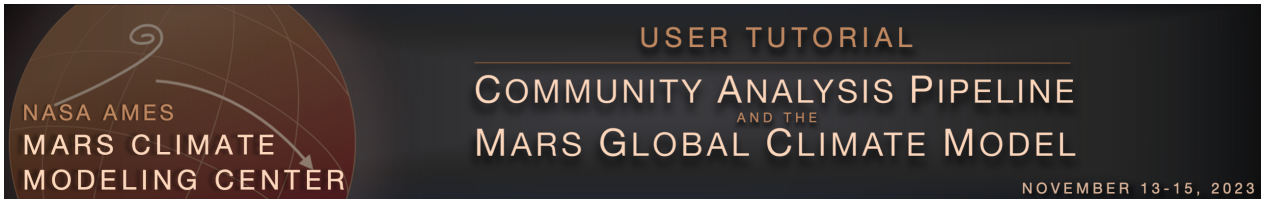
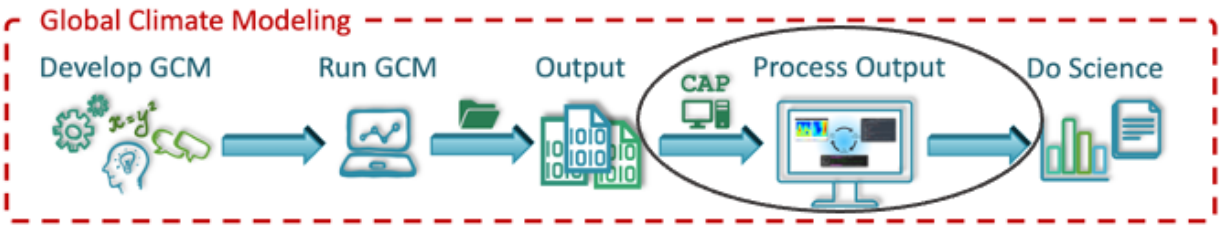


Advanced Capabilities (CAP)



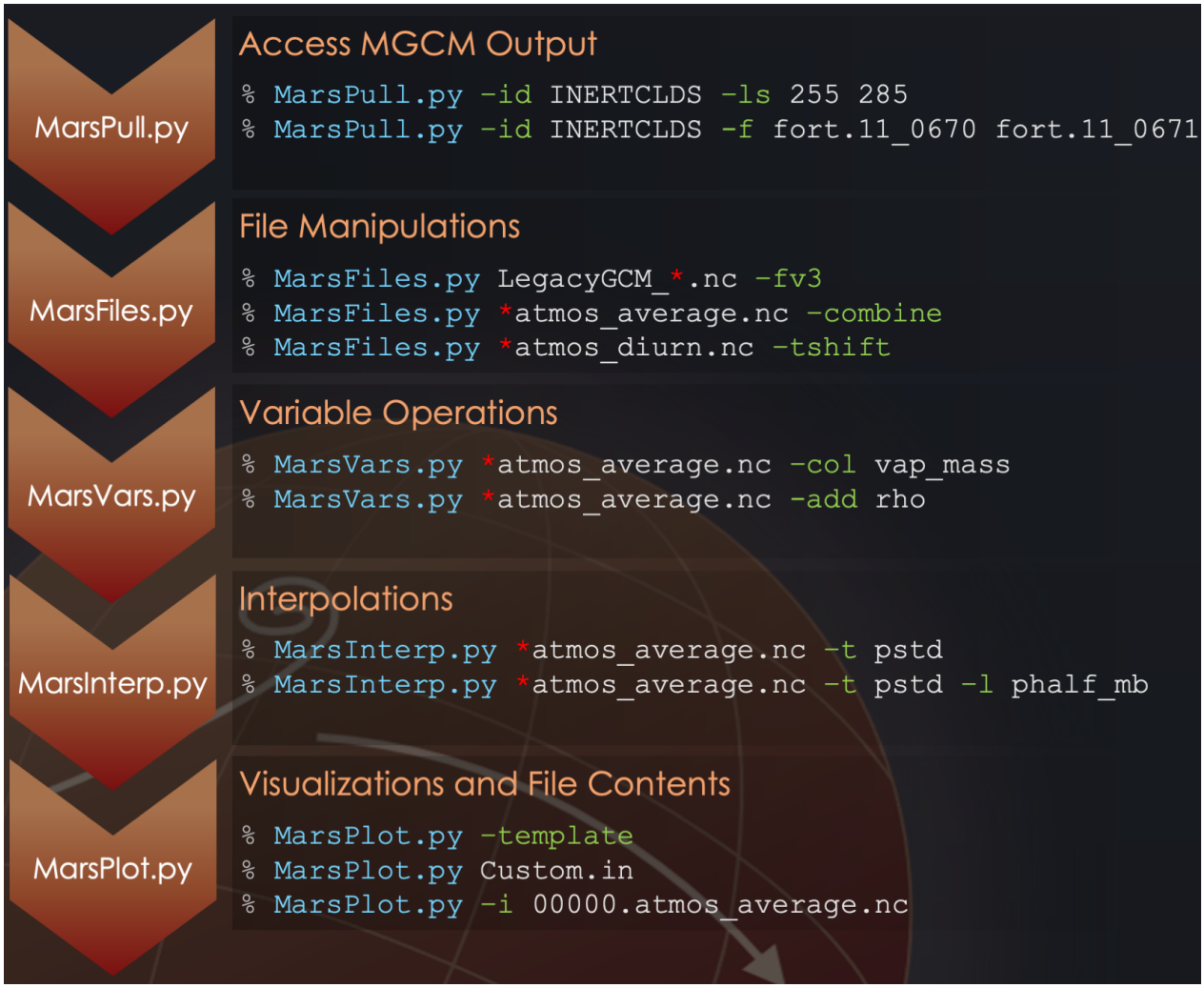
This section gives an overview of advanced capabilities not discussed in the lecture. There are three broad types of capability we can discuss:

1. Specific operations, included in the `main` branch
2. Operations involving proprietary data/modeling products
3. Newest capabilities being developed

This discussion is intended to bring awareness of current and planned development.

Contact the MCMC with specific feedback or question (POC alexandre.m.kling@nasa.gov

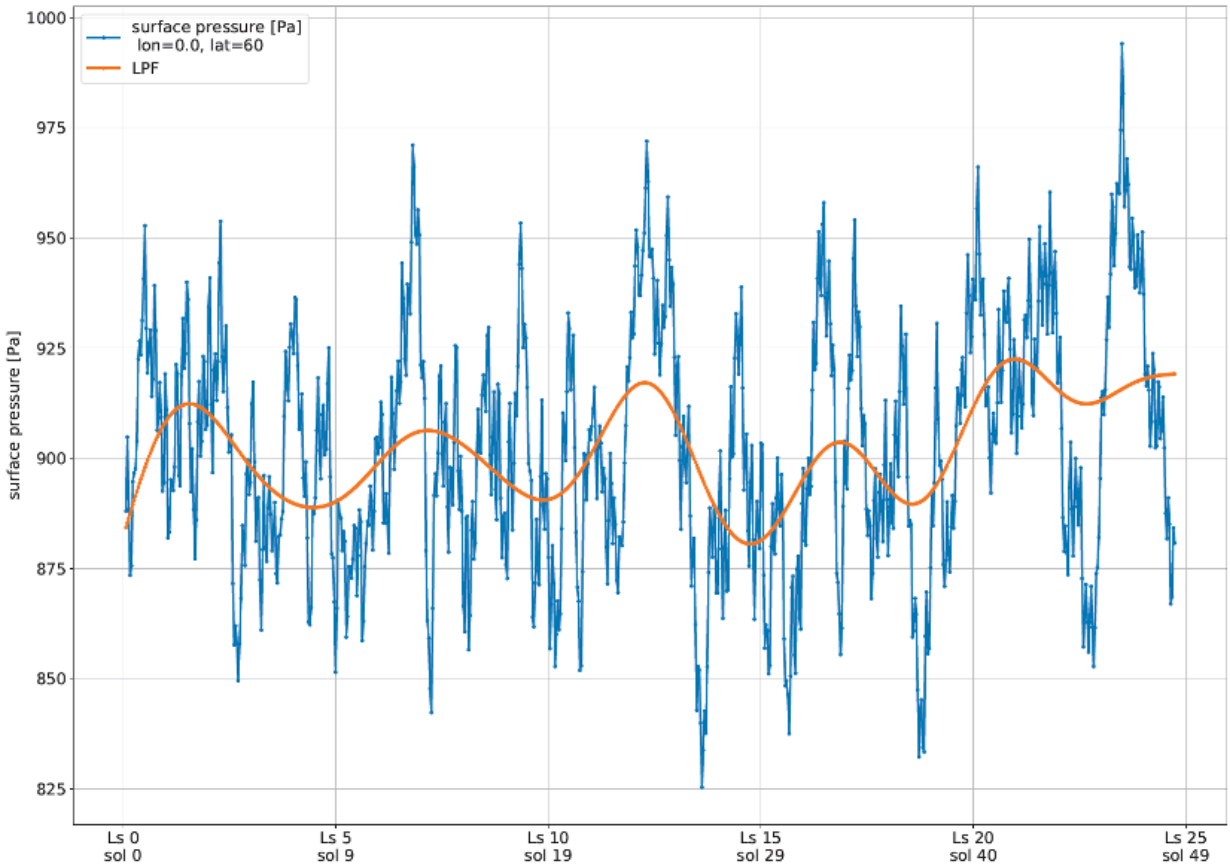
This is a recap of the CAP executables:



Time filtering

CAP supports **low-pass**, **high-pass** and **band-pass** time filtering of the 3D fields. Those are generated as additional files.

```
(AmesCAP)>$ MarsFiles.py 00668.atmos_daily.nc --low_pass_filter  10   #sols max cut-off
(AmesCAP)>$ MarsFiles.py 00668.atmos_daily.nc --high_pass_filter 5    #sols min cut-off
(AmesCAP)>$ MarsFiles.py 00668.atmos_daily.nc --bad_pass_filter  5 20 #sols min  and max cut-off
```



> Example of a low pass filter applied to the surface pressure field.

Tidal analysis

The tidal analysis works out of the diurnal composite (*atmos_diurn*) files and allows to isolate the diurnal, semi-diurnal etc… phase and amplitude of any surface or 3D fields.

```
(AmesCAP)>$ MarsFiles.py *.atmos_diurn.nc -tidal 6 #number of harmonics
```

Running the command above, will extract the **phase** and **amplitude** as individual arrays. The individual harmonics requested (N=1 is diurnal, N=2 is semi-diurnal, etc…), are stores along the `time_of_day_N` axis:

For example, the surface pressure `ps` will reads as:

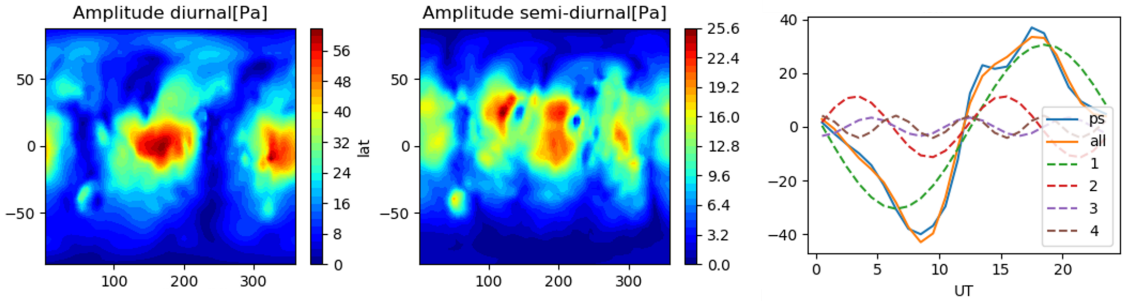
```
ps_amp          : ('time', 'time_of_day_6', 'lat', 'lon')= (133, 6, 48, 96), tidal amplitude for surface pressure  [Pa]
ps_phas         : ('time', 'time_of_day_6', 'lat', 'lon')= (133, 6, 48, 96), tidal phase for surface pressure  [hr]
```

The `--reconstruct` flag may be used to resynthetize the signal based on the selected number of harmonics.

```
ps_N1           : ('time', 'time_of_day_24', 'lat', 'lon')= (132, 24, 48, 96), harmonic N=1 for surface pressure  [Pa]
ps_N2           : ('time', 'time_of_day_24', 'lat', 'lon')= (132, 24, 48, 96), harmonic N=2 for surface pressure  [Pa]
ps_N3           : ('time', 'time_of_day_24', 'lat', 'lon')= (132, 24, 48, 96), harmonic N=3 for surface pressure  [Pa]
ps_N4           : ('time', 'time_of_day_24', 'lat', 'lon')= (132, 24, 48, 96), harmonic N=4 for surface pressure  [Pa]
```

> In this case, summing the `ps_N1+ps_N2+ ... etc` harmonics will reconstruct the original amplitude of the `ps` variable, minus the residual.

Finally the `--normalize` flag provides the amplitude as a percentage of the daily mean value instead of the absolute amplitude.



> Note that the reconstructed fields are Universal Time (UT) but can be time-shifted to allow plotting as a function of the local time.

File reggridding and interpolation

CAP provides a reggridding tool that allows to compare a simulation with either an other simulation or an observational dataset that is not on the same grid (e.g. the time sampling, vertical grid or horizontal grid can be different). This is done by running the following interpolations on the 3D fields:

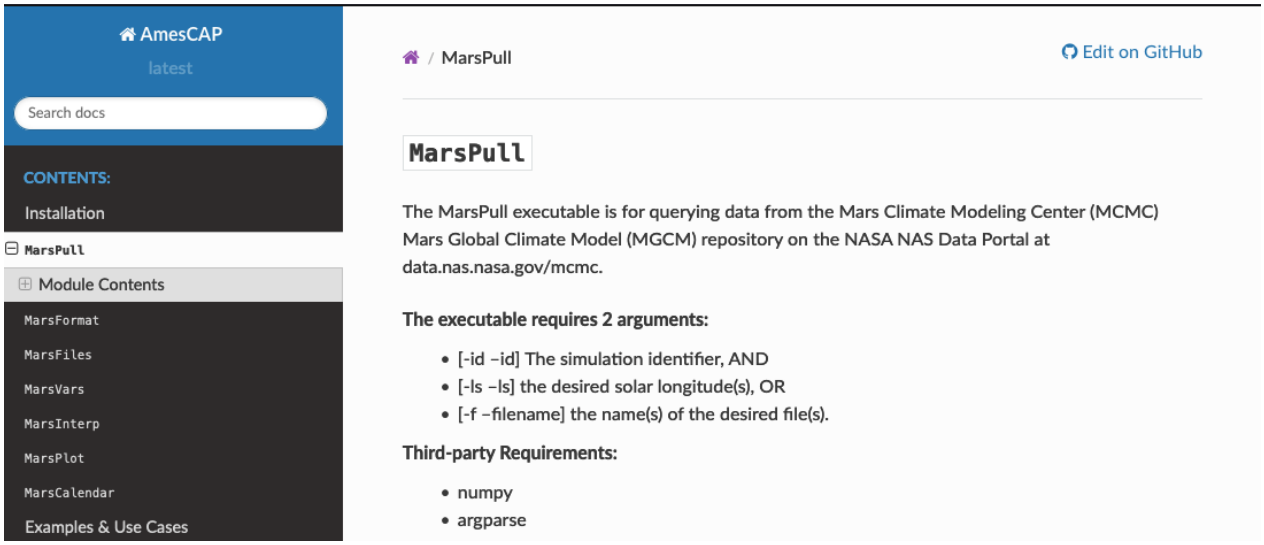
- linear time interpolation
- linear (cyclical) time of day interpolation
- logarithmic (pressure) or linear (altitude) interpolation
- unstructured, nearest neighboor (inversed weighted) 2D lat/lon interplation

```
(AmesCAP)>$ MarsFiles.py 00668.atmos.average_pstd.nc -rs simu2/00668.atmos_average_pstd.nc
```

Work in progress

Auto documentation

We are working on auto-documentation (documentation that sits with the CAP software) to provide extensive, searchable and up-to date descriptions of all CAP functionalities, as well as a reference paper (Kling et al. in prep.)



In the meantime, **ALL** functionalities available through the command-line in select branches are detailed in the `--help` (`-h` for short) sections of the CAP executables.

```
(amesGCM3)>$ MarsPlot.py -h
> usage: MarsPlot.py [-h] [-i INSPECT_FILE] [-d DATE [DATE ...]] [--template]
>                    [-do DO] [-sy] [-o {pdf,eps,png}] [-vert] [-dir DIRECTORY]
>                    [--debug]
>                    [custom_file]
```

We recognize it is not possible to develop command-line tools that satisfy all of the users' applications. Therefore, most of the operations available through the command-line are also accessible as **standalone Python libraries**:

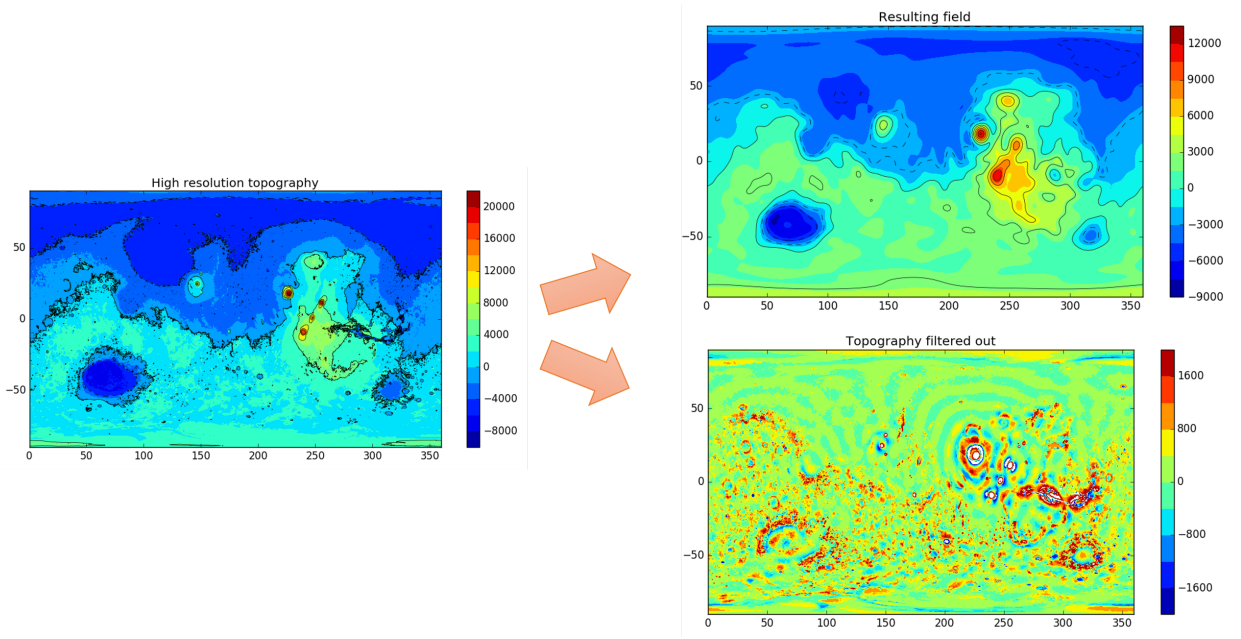
- `amecap/FV3_utils.py` (geospatial and meteorology functions)
- `amecap/Script_utils.py` (input/outputs)
- `amecap/Spectral_utils.py` (filtering libraries).

Those allow users to customize CAP functions for their own applications.

Spatial filtering

CAP supports a decomposition into spherical harmonics that allows to spatially **low-pass**, **high-pass** and **band-pass** filter the fields by their total wavenumber. The filtering is based on Wieczorek and Meschede, (2018)

```
(AmesCAP)>$ MarsFiles.py 00668.atmos_average.nc --lpk 24 # max wavenumber for low-pass
(AmesCAP)>$ MarsFiles.py 00668.atmos_average.nc --hpk 48 # min wavenumber for high-pass
(AmesCAP)>$ MarsFiles.py 00668.atmos_average.nc --bpk 24 48 # min and max wavenumber for band-pass
```



> Low and high pass filtering by total zonal wavenumbers applied to the topography

Multi-model comparison

We are actively developping cross-model compatibility for CAP. At the moment, the **Legacy GCM**, **openMars** and **MarsWRF** outputs can be ingested with CAP using the `MarsFile` (for the *Legacy GCM*) and the `MarsFormat` (for *openMars* and *MarsWRF*) executables. The executables currently performs model-specific operations (e.g. destaggering of the horizontal grids, creation of a new vertical grid) that are needed to make the output of other models fully usable with CAP.

```
(AmesCAP)>$ MarsFiles.py fort.11  --fv3 fixed average
(AmesCAP)>$ MarsFormat.py file.nc --openmars daily
(AmesCAP)>$ MarsFormat.py file.nc --marswrf daily
```

At the moment, the executables rename the dimensions and variables in the file to ensure compatibility with CAP (e.g. a temperature variable `T` might be renamed `temp`) but future releases will preserve the **native** variables' names.

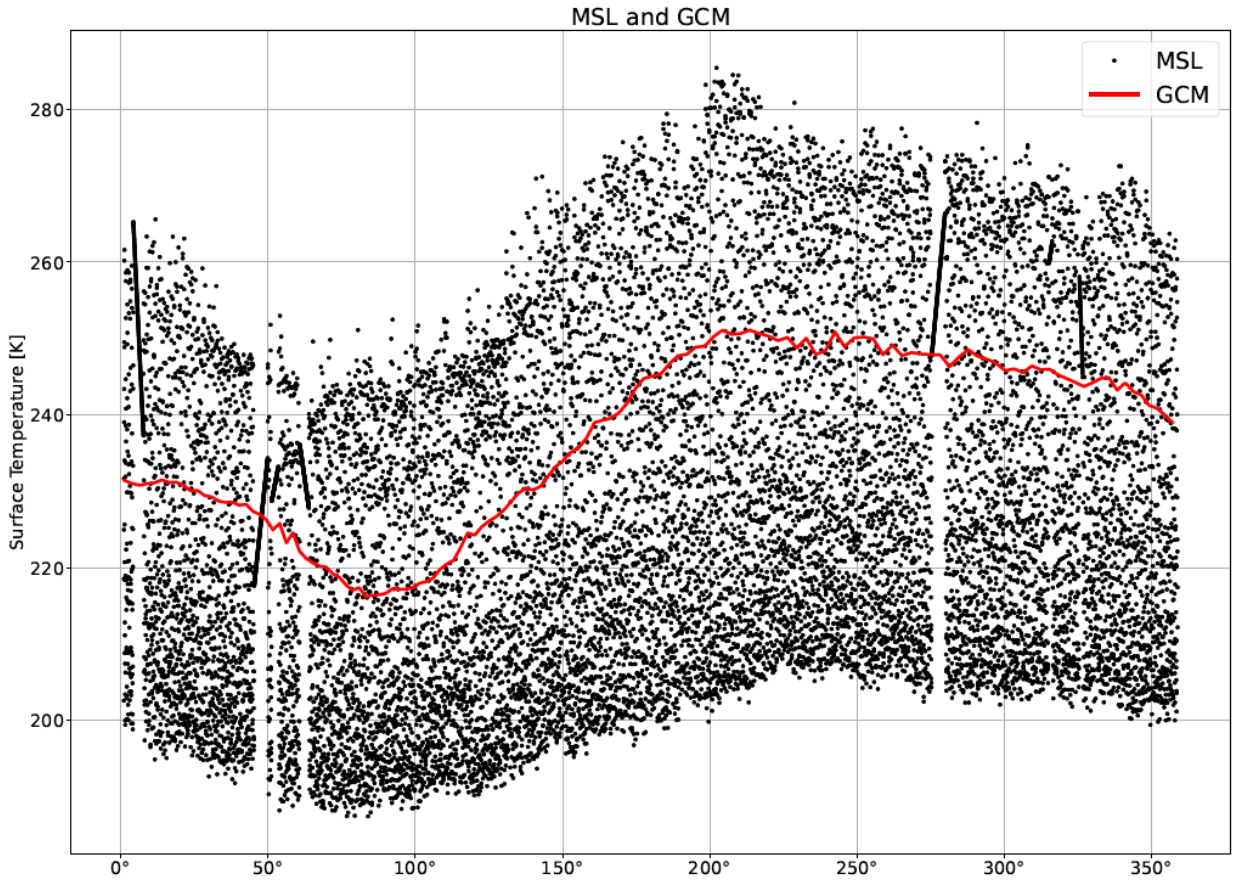
We are working to extend the compatibility of CAP with the eMARS, PCM (formally LMD) and the Mars Climate Database (MCD), (others?) modeling products.

Ingesting observations, proprietary data

This is an example of how proprietary station-data (Viking, Perseverance etc…) can be formatted to be ingested in CAP. Here, the file is analogous to a *daily* file (instantaneous time serie).

```
(AmesCAP)>$  MarsPlot.py -i station_data.nc
==================DIMENSIONS==========================
['lon', 'lat', 'time', 'pfull', 'scalar_axis']
[...]]
====================CONTENT=========================
time           : ('time',)= (15936,), time  [days since 0000-00-00 00:00:00]
pfull          : ('pfull',)= (1,), ref full pressure level  [Pa]
lat            : ('lat',)= (1,), latitude  [degrees_N]
lon            : ('lon',)= (1,), longitude  [degrees_E]
```

```
areo           : ('time', 'scalar_axis')= (15936, 1), solar longitude  [degree]
ps             : ('time', 'lat', 'lon')= (15936, 1, 1), surface pressure  [Pa]
```

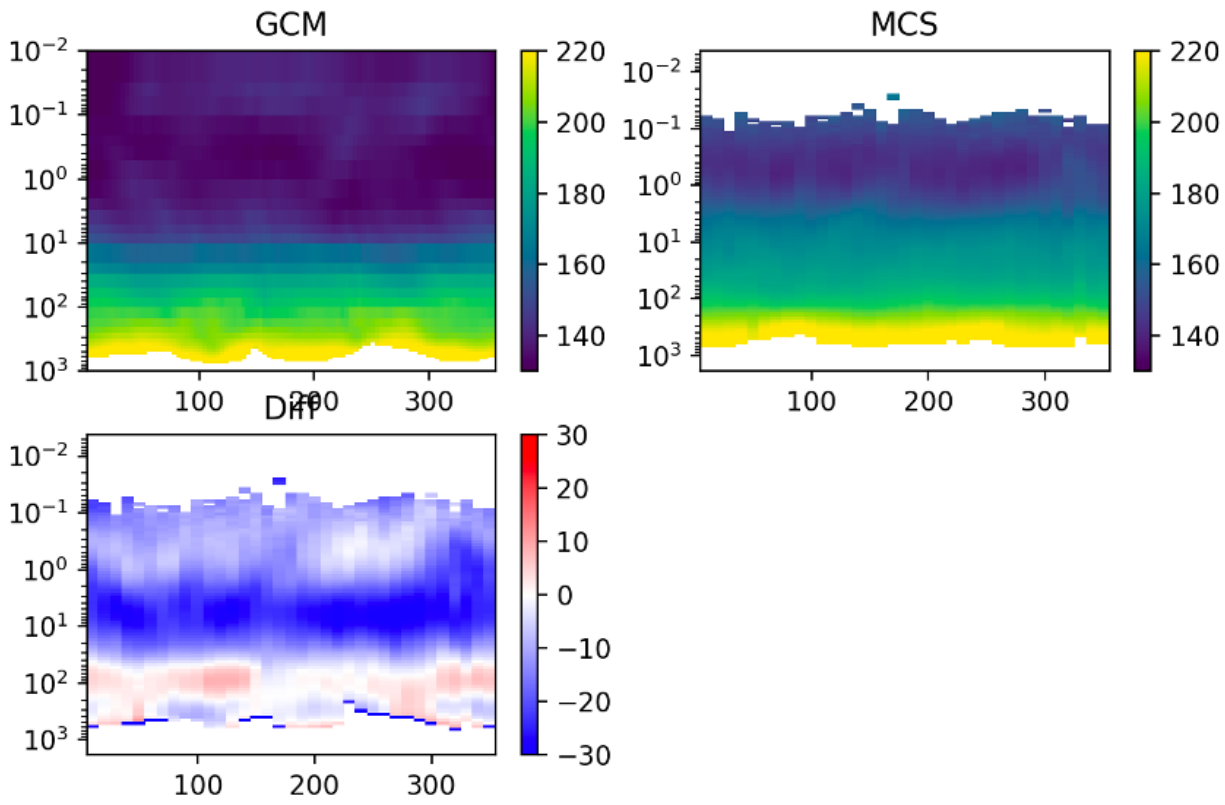


> Station data and MGCM output plotted in CAP.

This is an example of how orbital datasets (MCS etc…) can be formatted to be ingested in CAP. Here the dataset is formatted as an *atmos_diurn* file (diurnal composite with two am/pm times).

```
(AmesCAP)>$  MarsPlot.py -i my_obs_MCS.nc
==================DIMENSIONS==========================
['lon', 'lat', 'pstd', 'time', 'time_of_day_2']
[...]
===================CONTENT==========================
time_of_day_2  : ('time_of_day_2',)= (2,), reference time of day  [hours since 0000-00-00 00:00:00]
pstd           : ('pstd',)= (105,), standard pressure  [Pa]
lon            : ('lon',)= (64,), longitude  [degrees_E]
time           : ('time',)= (54,), time  [days since 0000-00-00 00:00:00]
areo           : ('time',)= (54,), solar longitude  [degree]
lat            : ('lat',)= (48,), latitude  [degrees_N]
temp           : ('time', 'time_of_day_2', 'pstd', 'lat', 'lon')= (54, 2, 105, 48, 64), Temperature  [K]
[...]
```

After an observational dataset is reformatted to be used in CAP, one can use MarsFile's reggriding utility (`--regrid_source`), to reggrid the GCM output onto the observational dataset' grid. This is useful for model/observation comparisons.



> GCM (left) and Mars Climate Sounder (right) are put on the same grid to allow for difference plots (bottom)

If you have an existing (or future) publicly available observational dataset (e.g. Xenodo) and would like to make it compatible with CAP, please reach out.

Community feedback

We would like to hear feedback and suggestions about existing and future capabilities you would find impactful for your work.