Guide for how to read the MACDA & EMARS datasets

First, you have to be familiar with NetCDF files and how to read them. During my research, we used the Panoply application. Moreover, here's an analysis on the MACDA & EMARS data variables

MACDA

1. float co2ice(time=360, lat=36, lon=72);
standard_name = "surface_frozen_carbon_dioxide_amount";
long_name = "Carbon dioxide ice";
units = "kg m-2";
comment = "This variable refers to the mass per square meter of carbon dioxide deposited as ice on the ground at polar latitudes, after having condensed from the atmosphere.";

2. float coldust(time=360, lat=36, lon=72);
:standard_name = "atmosphere_optical_thickness_due_to_dust_dry_aerosol";
:long_name = "Total dust optical thickness";
:units = "1";
:comment = "This variable refers to the fraction of radiation (at average visible wavelengths of 600 nm) that would be removed from a beam during its path through the entire atmosphere, from the surface to outer space, by scattering/absorption due to airborne mineral dust. The radiative flux would be reduced by a factor exp(-optical_thickness) on traversing the path.";

3. float lat(lat=36);
:standard_name = "latitude";
:long_name = "latitude";
:units = "degrees_north";
:axis = "Y";

4. float lev(lev=25);
:standard_name = "atmosphere_sigma_coordinate";
:long_name = "model sigma levels";
:units = "1";
:axis = "Z";
:comment = "Sigma levels are non-dimensional terrain-following vertical levels, with values between 1 (at the ground) and 0 at infinite distance from the ground. The sigma value at model grid points is defined as the ratio between atmospheric pressure and

surface pressure. Atmospheric pressure at each model level and grid point can therefore be calculated by using the formula p(i,j,k)=psurf(i,j)*lev(k) where i,j,k are indices of longitude, latitude and level, p is the atmospheric pressure, psurf is the surface pressure and lev is the sigma value.";

5. float lon(lon=72);
 :standard_name = "longitude";
 :long_name = "longitude";
 :units = "degrees_east";
 :axis = "X";

6. float Ls(time=360);
 :standard_name = "solar_longitude";
 :long_name = "Solar longitude";
 :units = "degrees";
 :comment = "The solar longitude Ls (also sometimes referred to as \'areocentric longitude\') is the Mars-Sun angle described by the orbit of Mars and measured from the northern hemisphere spring equinox where Ls=0 degrees. Ls=90 degrees thus corresponds to summer solstice, just as Ls=180 degrees marks the autumn equinox and Ls=270 degrees the winter solstice (all relative to the northern hemisphere)";

7. short MY(time=360);
 :long_name = "Martian year";
 :units = "1";
 :comment1 = "Please be aware that \"Martian year\" is not a standard CF variable, and can be calculated from the time and solar_longituyde variables. We explicitly report it here for easiness of use of the dataset.";
 :comment2 = "Martian years are counted following the convention proposed by Clancy et al. (J. Geophys Res. 105, 9553-9571, 2000). The first Martian year begins at spring equinox of the year during which the 1956 Martian planet-encircling dust storm was observed, hence starting a new period of \"careful and unified\" (namely, properly scientific) observations on Mars. This dataset starts in MY=24, which is the first year of available MGS/TES mapping-phase observations (corresponding to May 1999) and ends in MY=27 (August 2004).";

8. float psurf(time=360, lat=36, lon=72);
 :standard_name = "surface_air_pressure";
 :long_name = "Surface pressure";
 :units = "Pa";

9. float temp(time=360, lev=25, lat=36, lon=72);
:standard_name = "air_temperature";
:long_name = "Atmospheric Temperature";
:units = "K";

10. double time(time=360);
:standard_name = "time";
:long_name = "time";
:units = "days since 0000-00-0 00:00:00";
:calendar = "none";
:axis = "T";
:comment1 = "Please be aware that formally the time unit should be \'sols since 0.0\', but in the present release we use the \'days since 0000-00-0 00:00:00\' unit while the CF community decides on the introduction of sol as a new CF variable.";
:comment2 = "Time is referred to the Martian mean solar day, which is called sol and has 88775.244 s. The time origin in the dataset (sol=0.0) corresponds to midnight at the Martian Prime Meridian at solar longitude Ls=0 degrees (northern spring equinox) in Martian year MY=24, which is the first year of available MGS/TES mapping-phase observations. Sols are divided in 24 Martian hours, and output fields are provided each 2 hours, beginning at 2 a.m. in sol 301 (first available sol in the dataset). The Martian (sidereal) year from one spring equinox to the next is 668.5921 sols (~668.6 sols), but we count sols continuously in the dataset. One can therefore calculate the corresponding Martian year of an output field by using the solar longitude variable, which resets to zero every new Martian year at spring equinox. The entire dataset spans four Martian years, 24 through 27, although MY 24 and 27 are not complete. The conversion between solar longitude/Martian year and Earth date (and viceversa) can be operated using the converters available at the Mars Climate Database Website (http://www-mars.lmd.jussieu.fr/mars/time/martian_time.html), which also includes more detailed information about the time on Mars.";

11. float tsurf(time=360, lat=36, lon=72);
:standard_name = "surface_temperature";
:long_name = "Surface temperature";
:units = "K";

12. float uwind(time=360, lev=25, lat=36, lon=72);
:standard_name = "eastward_wind";
:long_name = "Zonal wind component";

:units = "m s-1";

13. float vwind(time=360, lev=25, lat=36, lon=72);
:standard_name = "northward_wind";
:long_name = "Meridional wind component";
:units = "m s-1";

Notes:
+ The entire dataset spans four Martian years, 24 through 27, although MY 24 and 27 are not complete. (please check comment 2 of "MY" and "time" variables).
+ read comment 1 for the "time" variable. In comment 2: t = 88775.244 s and The Martian (sidereal) year from one spring equinox to the next is 668.6.

EMARS

1. float ak(phalf=29);
:long_name = "pressure part of the hybrid coordinate";
:units = "pascal";
:cell_methods = "time: point";

2. float bk(phalf=29);
:long_name = "vertical coordinate sigma value";
:units = "none";
:cell_methods = "time: point";

3. double earth_day(time=1464);
:long_name = "Earth day of the month";
:units = "Earth day";

4. double earth_hour(time=1464);
:long_name = "Earth hour of the day";
:units = "Earth hour";

5. double earth_minute(time=1464);
:long_name = "Earth minute of the hour";
:units = "Earth minute";

6. double earth_month(time=1464);
:long_name = "Earth month of the year";
:units = "Earth month";

```
7. double earth_second(time=1464);
 :long_name = "Earth second and fractional second of the minute";
 :units = "Earth second";

8. double earth_year(time=1464);
 :long_name = "Earth year AD";
 :units = "Earth year";

9. double emars_sol(time=1464);
 :long_name = "sols after MY 22 perihelion";
 :units = "Martian sol";

10. double lat(lat=36);
 :long_name = "latitude";
 :units = "degree_N";

11. double latu(latu=36);
 :long_name = "latitude";
 :units = "degree_N";

12. double lon(lon=60);
 :long_name = "longitude";
 :units = "degree_E";

13. double lonv(lonv=60);
 :long_name = "longitude";
 :units = "degree_E";

14. double Ls(time=1464);
 :long_name = "areocentric longitude";
 :units = "deg";

15. double macda_sol(time=1464);
 :long_name = "sols after the start of MY 24";
 :units = "Martian year";

16. double mars_hour(time=1464);
 :long_name = "hour of the Martian day";
 :units = "Martian hour";
```

```
17. double mars_soy(time=1464);
 :long_name = "sols after the last Martian vernal equinox";
 :units = "Martian sol";

18. double MY(time=1464);
 :long_name = "Mars Year";
 :units = "Martian year";

19. double pfull(pfull=28);
 :long_name = "ref full pressure level";
 :units = "mb";

20. double phalf(phalf=29);
 :long_name = "ref half pressure level";
 :units = "mb";
 :cartesian_axis = "Z";
 :positive = "down";

21. double ps(time=1464, lat=36, lon=60);
 :long_name = "surface pressure";
 :units = "pascal";

22. double Surface_geopotential(lat=36, lon=60);
 :long_name = "surface geopotential height";
 :units = "m^2/s/s";

23. double T(time=1464, pfull=28, lat=36, lon=60);
 :long_name = "Temperature";
 :units = "K";

24. double time(time=1464);
 :long_name = "number of hours since start of file";
 :units = "Martian hour";

25. double U(time=1464, pfull=28, latu=36, lon=60);
 :long_name = "zonal wind";
 :units = "m/s";

26. double V(time=1464, pfull=28, lat=36, lonv=60);
```

```
 :long_name = "meridional wind";
 :units = "m/s";
```

Notes:
+ for FOR 'BACK' AND 'MEMB', I will only show the things that are included in it specifically, and not everything because there are many similarities between both the 'anal, mean' then the 'back, memb'. Anything that is included in 'back, memb' but not the other one will be mentioned in these notes InshaAllah.

FOR 'BACK, MEMB':

```
float cld(time=1464, pfull=28, lat=36, lon=60);
 :long_name = "water ice mass mixing ratio";
 :units = "kg/kg";
 :missing_value = -1.0E10f; // float
 :cell_methods = "time: point";

float cldcol(time=1464, lat=36, lon=60);
 :long_name = "column water ice";
 :units = "kg/m/m";
 :missing_value = -1.0E10f; // float
 :cell_methods = "time: point";

float dod(time=1464, lat=36, lon=60);
 :long_name = "column dust visible opacity";
 :units = "unitless";
 :missing_value = -1.0E10f; // float
 :cell_methods = "time: point";

float frost(time=1464, lat=36, lon=60);
 :long_name = "Surface water ice";
 :units = "kg/m/m";
 :missing_value = -1.0E10f; // float
 :cell_methods = "time: point";

float h(time=1464, phalf=29, lat=36, lon=60);
 :long_name = "height";
 :units = "m";
 :missing_value = -1.0E10f; // float
 :cell_methods = "time: point";
```

```
float hrad(time=1464, pfull=28, lat=36, lon=60);
 :long_name = "Radiative heating rate";
 :units = "K/s";
 :missing_value = -1.0E10f; // float
 :cell_methods = "time: point";

double lat(lat=36);
 :long_name = "latitude";
 :units = "degrees_N";
 :cartesian_axis = "Y";
 :edges = "latb";

float lheat(time=1464, pfull=28, lat=36, lon=60);
 :long_name = " CO2 latent heating";
 :units = "deg_K/sec";
 :missing_value = -1.0E10f; // float
 :cell_methods = "time: point";

double lon(lon=60);
 :long_name = "longitude";
 :units = "degrees_E";
 :cartesian_axis = "X";
 :edges = "lonb";

double macda_sol(time=1464);
 :long_name = "sols after the start of MY 24";
 :units = "Martian year";
```

There are more variables! This dataset is huge