HELIOSARES (ANR-09-BLAN-223)

Output format and data description

EGM – Exospheric Global Model

Title:*	Output format and data description – EGM/Exospheric Gloabl Model
Date:*	July 07th, 2017
Editor:*	F. Leblanc
Contributors:*	JY. Chaufray, R. Modolo
Distribution:*	*HELIOSARES project + MAVEN team*
Level:*	*low level*
Roles:*	

Version History

Version	Date	Released by	Detail
0.0	24/03/2014	F. Leblanc	draft
0.1	31/03/2014	R. Modolo	Add species and inner boundary information
0.2	07/07/2017	R. Modolo	Clarification on altitude grid
1.0	28/08/2017	R. Modolo	Consolidated version

Content

1-Data accessibility and data description

1.1- Data description

Information about the model is discussed in several papers:

- For the sputtering components:

Leblanc, F. and R.E Johnson, Sputtering of the martian atmosphere by solar wind pick-up ions, *Planet. Space Sci*., 49, 645-656, 2001.

- For the photo-chemical components:

Chaufray, J.Y., Modolo R., Leblanc F., G. Chanteur, R.E. Johnson and J.G. Luhmann, Mars Solar Wind interaction: formation of the Martian corona and atmospheric loss to space, *Journal of Geophysical Research*, Volume 112, Issue E9, CiteID E09009, doi: 10.1029/2007JE002915, 2007.

- For the dissociative components, thermal components and role of the atmosphere on the exosphere

Yagi M., Leblanc F., Chaufray J.Y., F. Galindo-Gonzalez and R. Modolo, Mars exosphere: thermal and non-thermal components: seasonal and local variations, Icarus, 221 (2), 682-693, doi: 10.1016/j.icarus.2012.07.022, 2012.

Limitations :

The sputtering component does take into account only a one-species atmosphere and calculates only the oxygen component produced by sputtering. The calculation of the photo-chemical component does take into account only the recombinative dissociation of the O_2^+ ion. As a consequence, only the O non-thermal component is probably described.

Coordinate system:

Computation and simulations output are provided in MSO latitude and longitude. Table 1 provide information on the grid structure and their related variables

Variable name	Information	Quantities using this variable
theta_low **theta_up**	Co-latitude values along theta (npt_lat)	density_O…, VOx…,TempO
phi_low **phi_up**	Longitude values along phi (npt_lon)	
altitude_low **altitude_up**	Altitude values along altitude (npt_alt)	

Table 1: Variables related to the grid structure

Most of the quantities (density, Vx, Temp, …) provide 4D information (exospheric component, Longitude, Latitude, altitude). The first dimension refers to the four components of the exospheric (1 = Sum of all components, 2 = RD component only, 3 = Sputtering component only, 4 = thermal component only). The following dimensions refer to a combination of latitude, longitude and altitude information.

The different species included in the exospheric files are listed in table 2

Species	Distribution
H	Thermal
CO2	Thermal
O	Thermal + non-thermal (DR + sputtering ?)

Table 2: Species included in the exospheric file

Simulations start at altitude 200km. Data points below this altitude should not be considered. The inner boundary condition is given by the GCM model where the density (for each species) and temperature are provided (for the same Ls value and solar activity)

The simulation includes planetary rotation.

2.- Output format

Archived outputs files are saved in the "netCDF" format (http://www.unidata.ucar.edu/software/netcdf/).

This format is well documented and largely adopted in numerical and data archives.

General information concerning each file can be accessed through netcdf functionalities. For instance in a terminal window the command "ncdump –h Exo3D_ThNonTh_Case1_Ls270_SolMean_150314.nc" returns information concerning the "Exo3D_XXXX.nc" file (assuming that the netcdf package has been installed on the local machine). It is composed of three parts: 1- dimensions of variables of the files, 2- the name of the variables, their data type and dimension, 3- the attribute of the file.

To have access to the value of a specific variable, the command "ncdump –v variable_name Exo3D_ThNonTh_Case1_Ls270_SolMean_150314.nc" return the requested information.

Ex : ncdump –v Ls Exo3D_ThNonTh_Case1_Ls270_SolMean_150314.nc
>>Ls = 270;

In front of each line we have indicated information about the variable. Only a subset of the header is provided in table 2.

dimensions: npt_alt = 86 ;	Number of altitude bins

npt_lat = 30 ;	Number of latitude bins
npt_lon = 60 ;	Number of longitude bins
ncomp = 4 ;	Number of exospheric components
dim_string = 1 ;	
dim_int = 1 ;	
dim_real = 1 ;	
variables:	
int npt_alt(dim_int) ;	
int npt_lat(dim_int) ;	
int npt_lon(dim_int) ;	
int ncomp(dim_int) ;	
double altitude_low(npt_alt) ;	Lower altitude of the each cell
double altitude_up(npt_alt) ;	Upper altitude of each cell
double theta_low(npt_lat) ;	Lower co-latitude of each cell
double theta_up(npt_lat) ;	Upper co-latitude of each cell
double phi_low(npt_lon) ;	Lower longitude of each cell
double phi_up(npt_lon) ;	Upper longitude of each cell
double density_O(ncomp, npt_lon, npt_lat, npt_alt) ;	Average O density in each cell (ncomp exospheric component)
double density_H(ncomp, npt_lon, npt_lat, npt_alt) ;	Average H density in each cell (ncomp exospheric component)
double density_H2(ncomp, npt_lon, npt_lat, npt_alt) ;	Average H2density in each cell (ncomp exospheric component)
double density_CO2(ncomp, npt_lon, npt_lat, npt_alt) ;	Average CO2 density in each cell (ncomp exospheric component)
double VOx(ncomp, npt_lon, npt_lat, npt_alt) ;	Average O Vx in each cell (ncomp exospheric component)
double VOy(ncomp, npt_lon, npt_lat, npt_alt) ;	Average O Vy in each cell (ncomp exospheric component)
double VOz(ncomp, npt_lon, npt_lat, npt_alt) ;	Average O Vz in each cell (ncomp exospheric component)
double VHx(ncomp, npt_lon, npt_lat, npt_alt) ;	Average H Vx in each cell (ncomp exospheric component)
double VHy(ncomp, npt_lon, npt_lat, npt_alt) ;	Average HVy in each cell (ncomp exospheric component)
double VHz(ncomp, npt_lon, npt_lat, npt_alt) ;	Average H Vz in each cell (ncomp exospheric component)
double VH2x(ncomp, npt_lon, npt_lat, npt_alt) ;	Average H2 Vx in each cell (ncomp exospheric component)
double VH2y(ncomp, npt_lon, npt_lat, npt_alt) ;	Average H2 Vy in each cell (ncomp exospheric component)
double VH2z(ncomp, npt_lon, npt_lat, npt_alt) ;	Average H2 Vz in each cell (ncomp exospheric component)
double VCO2x(ncomp, npt_lon, npt_lat, npt_alt) ;	Average CO2 Vx in each cell (ncomp exospheric component)
double VCO2y(ncomp, npt_lon, npt_lat, npt_alt) ;	Average CO2 Vy in each cell (ncomp exospheric component)
double VCO2z(ncomp, npt_lon, npt_lat, npt_alt) ;	Average CO2 Vz in each cell (ncomp exospheric component)
double TempO(ncomp, npt_lon, npt_lat, npt_alt) ;	Average O Temperature in each cell (ncomp exospheric component)
double TempH(ncomp, npt_lon, npt_lat, npt_alt) ;	Average H Temperature in each cell (ncomp exospheric component)
double TempH2(ncomp, npt_lon, npt_lat, npt_alt) ;	Average H2 Temperature in each cell (ncomp exospheric component)
double TempCO2(ncomp, npt_lon, npt_lat, npt_alt) ;	Average CO2 Temperature in each cell (ncomp exospheric component)
char Solar_Activity(dim_string) ;	Solar activity
double LS(dim_int) ;	Solar longitude
char Object(dim_string) ;	Planet name
double Radius(dim_real) ;	Radius of the planet
double Nb_of_species(dim_int) ;	Number of species
char Name_of_species_1(dim_string) ;	Name of species 1
char Name_of_species_2(dim_string) ;	Name of species 2
char Name_of_species_3(dim_string) ;	Name of species 3
char Name_of_species_4(dim_string) ;	Name of species 4
char Frame(dim_string) ;	Type of reference frame
char Date_of_writing(dim_string) ;	Date of writing
char Input_files_O_RD(dim_string) ;	Source file for recombinative dissociation exosphere (O)
char Input_files_O_SP(dim_string) ;	Source file for sputtering exosphere (O)
char Input_files_O_Th(dim_string) ;	Source file for thermal component (O)
char Input_files_H_Th(dim_string) ;	Source file for thermal component (H)
char Input_files_H2_Th(dim_string) ;	Source file for thermal component (H2)
char Input_files_CO2_Th(dim_string) ;	Source file for thermal component (CO2)
char Authors(dim_string) ;	Authors name
char Reference(dim_string) ;	Reference papers
char Comments(dim_string) ;	Comments
char Components(dim_string) ;	Components

3.- IDL reader

An example of IDL reader is shown below. It reads and extracts only part of variables stored in the file. However it is relatively easy to extend it to complementary data set recorded.

```
fileID = ncdf_open(filename, /nowrite)  ; Open the ncdf file
structure = ncdf_inquire(fileID)        ; Read the global structure
nb_data = structure.nvars

for i=0,nb_data-1 do begin          ;Loop on variable
 vardata = ncdf_varinq(fileID,i)    ;Information on each variable
```

```
 varname = vardata.name               ;Name of the variable
 NCDF_DIMINQ, fileID, vardata.dim(0), Name, dim
 varid = ncdf_varid(fileID, varname)
 ncdf_varget,fileID,varid,data0
 if (i eq 0)   then npt_alt      = data0
 if (i eq 1)   then npt_lat      = data0
 if (i eq 2)   then npt_lon      = data0
 if (i eq 3)   then ncomp        = data0
 if (i eq 4)   then altitude0    = data0
 if (i eq 5)   then altitude1    = data0
 if (i eq 6)   then theta0       = data0
 if (i eq 7)   then theta1       = data0
 if (i eq 10)  then den0         = data0
…

endfor
```

4.- IDL visualization example

Different examples of data visualization are proposed in order to cross check that the file has been correctly read and data have been correctly displayed.

The file *Exo3D_ThNonTh_Case1_Ls270_SolMean_150314.nc* has been used to create the following figures.

The following script allows selecting a given set of points in order to plot a 2D section of the 3D density file. Figure 1 provides an example of a plot of the density in the equatorial plane.

We first reorganize the data.

```
npt0   = long(npt_alt * npt_lat * npt_lon)
Nexo   = fltarr(npt0,nb_species,ncomp)      ;nb_species if the total number
of species, ncomp the total number of exospheric components
Xexo   = fltarr(npt0,nb_species,ncomp,2)
Yexo   = fltarr(npt0,nb_species,ncomp,2)
Zexo   = fltarr(npt0,nb_species,ncomp,2)
r      = fltarr(npt0,nb_species,ncomp,2)
phi    = fltarr(npt0,nb_species,ncomp,2)
tet    = fltarr(npt0,nb_species,ncomp,2)
npt    = lonarr(nb_species,ncomp)
for ispecies = 0,nb_species-1 do begin
 if (ispecies eq 0) then density = den0
…
 for icomp = 0,ncomp-1 do begin
  ipt = long(0)
  for ialt = 0,npt_alt-1 do begin
   for ilat = 0,npt_lat-1 do begin
    for ilon = 0,npt_lon-1 do begin
      Nexo[ipt,ispecies,icomp]     = density[ialt,ilat,ilon,icomp]
      r[ipt,ispecies,icomp,0]      = altitude0[ialt]
      phi[ipt,ispecies,icomp,0]    = phi0[ilon]
      tet[ipt,ispecies,icomp,0]    = theta0[ilat]
      r[ipt,ispecies,icomp,1]      = altitude1[ialt]
      phi[ipt,ispecies,icomp,1]    = phi1[ilon]
      tet[ipt,ispecies,icomp,1]    = theta1[ilat]
      ipt = ipt + 1
    endfor
```

```
    endfor
   endfor
   npt[ispecies,icomp] = ipt
  endfor
 endfor
```

We then select the points close to the equatorial plane

```
 good =
where((abs((tet[0:npt[ispecies,icomp],1,ispecies,icomp,0]+tet[0:npt[ispecies
,icomp]-1,ispecies,icomp,1])*0.5 - !Pi*0.5) lt !Pi/(npt_lat + 0.),nplane)
  rN          = fltarr(nplane,2)
  AngN        = fltarr(nplane,2)
  dataN       = fltarr(nplane)
  rN[0:nplane-1,0:1]    = r[good,ispecies,icomp,0:1]
  AngN[0:nplane-1,0:1] = phi[good,ispecies,icomp,0:1]
  dataN[0:nplane-1]     = Nexo[good,ispecies,icomp]
```

We then plot the set of points (rN, AngN, dataN).

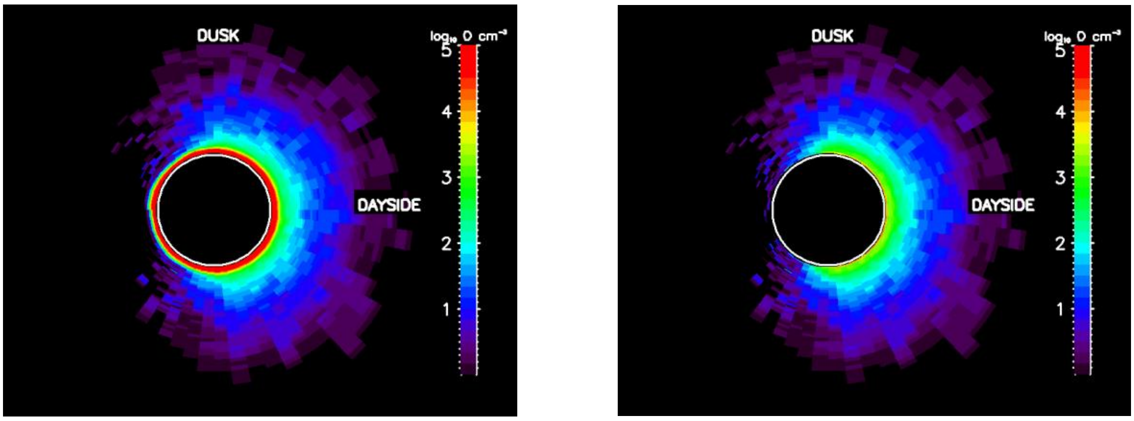

Figure 1: Density equatorial distribution of the O exospheric component. Left panel: total density, right panel: recombinative dissociation component.