Hybrid simulation outputs

[Data format and Data description]
[LATMOS SMDB]

Title:*	Hybrid diagnostics and data volume
Date:*	29 November 2012
Editor:*	R. Modolo
Contributors:*	S. Hess
Distribution:*	*project*
Level:*	*low level*
Roles:*	
User Requirement:	
Related Workpackage(s):	WP3-WP2

EC Project Nr.: **262863**

Version History

Version	Date	Released by	Detail
0.0	16/04/2012	R. Modolo	draft
0.1	31/05/2012	R. Modolo	
0.2	29/11/2012	R. Modolo	Update netCDF + add VOTable description
1.0	28/08/2017	R. Modolo	Consolidated version

Table of Contents

Preface 2

1- Simulation Outputs 3

1.1 Data format 4

1.2 Data description: 4

1.2.1 Grid and coordinate system 4

1.2.2 Simulation unit and Normalization 5

1.2.3 Simulation variables 6

2. Specific IMPEx diagnostic 10

2.1 “Cuts” 10

2.1.1 Data format 11

2.1.2 Data description 11

Annex 1 : Parametrization 13

Preface

Simulations performed at LATMOS are based on a parallel multi-species hybrid simulations model developed in the frame of the ANR HELIOSARES project (ANR-09-BLAN-223). Simulations are performed on local computational platform (http://ciclad-web.ipsl.jussieu.fr/accueil/) with private access. Currently, the computational domain accessible for this LATMOS simulation model is limited to 5 servers (32-cores server) with 4Gb memory per core. These computational ressources have been financed by ANR HELIOSARES (Agence National de la recherche), CNRS-INSU (Centre National de la Recherche Scientifique – Institut National des Sciences de l’Univers) and CNES (Centre National d’Etude Spatial). The simulation uses a 3D cartesian and uniform grid.

1- Simulation Outputs

Global simulation outputs provide a 3D view of the state of the simulation and quasi-stationary solutions are archived. The IMPEx user will have access to these different files.

The set of files (and physical parameters associated) which are generated for a typical Martian simulation run :

- "Magw_XXXX.nc" : 3D magnetic field components (Bx,By,Bz)
- "Elew_XXXX.nc" : 3D electric field components (Ex,Ey,Ez)
- "Thew_XXXX.nc" : 3D electron number density (Dne), 3D bulk velocity components (Ux,Uy,Uz), Electron temperature (Te)
- "Read_moment_species_XXXX.dat" : ascii file containing ion species name include in the simulation
- "Hsw_XXXX.nc" : Solar wind proton diagnostic : 3D density, bulk speed and temperature (Dn, Ux,Uy,Uz,T)
- "Hesw_XXXX.nc" : Solar wind alpha particle diagnostic : 3D density, bulk speed and temperature (Dn, Ux,Uy,Uz,T)
- "Hpl_XXXX.nc" : Planetary proton diagnostic : 3D density, bulk speed and temperature (Dn, Ux,Uy,Uz,T)
- "Opl_XXXX.nc" : Planetary O+ diagnostic : 3D density, bulk speed and temperature (Dn, Ux,Uy,Uz,T)
- "O2pl_XXXX.nc" : Planetary O2+ diagnostic : 3D density, bulk speed and temperature (Dn, Ux,Uy,Uz,T)
- "CO2pl_XXXX.nc" : Planetary CO2+ diagnostic : 3D density, bulk speed and temperature (Dn, Ux,Uy,Uz,T)
- "Atmw_XXXX.nc" : 3D neutral distribution (Dn for each neutral species)

These archived files correspond to functional services required for the SMDBs. The corresponding UR are: UR-1.4.1.1, UR-1.4.1.2 and UR-1.4.1.2.1.
Concerning ion species (H+pl, O+pl, O2+pl and CO2+pl), information are provided without discrimination with respect to the source of production (photoproduction, electron impact, charge exchange or ionospheric chemistry).

Just to remind the IMPEx user about the size of each global output (3D) data files, table 1 summarizes this information.

	Low Resolution	Medium Resolution
"Magw_XXXX.nc"	61Mo	175Mo
"Elew_XXXX.nc"	61Mo	175Mo
"Velw_XXXX.nc"	61Mo	175Mo
"Dne_XXXX.nc"	20Mo	92 Mo

“Hsw_XXXX.nc”	101Mo	460 Mo
“Hesw_XXXX.nc”	101Mo	460 Mo
“Hpl_XXXX.nc”	101Mo	460 Mo
“Opl_XXXX.nc”	101Mo	460 Mo
“O2pl_XXXX.nc”	101Mo	460 Mo
“CO2pl_XXXX.nc”	101Mo	460 Mo
“Atmw_XXXX.nc”	122Mo	645 Mo
“p3_XXXX.nc”	~ 1.7 Go	~ 10.5 Go
Total	~2.6Go	~14.5Go

Table 1 : data volume of 3D simulation runs for low and medium resolution. Global outputs.

1.1 Data 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.

1.2 Data description:

1.2.1 Grid and coordinate system

The simulation uses a 3D uniform Cartesian grid with the following coordinate system called “simulation coordinate system”.

- For Mars’ simulation: The x-axis is aligned with the solar wind direction (+Vsw), ie from the Sun toward the planet. The z-axis is perpendicular to Mars’ orbital plane and the y-axis completes the right handed system.
- For Titan’s simulation: The x-axis coincides with the ideal magnetospheric plasma flow direction, the z–axis is perpendicular to Titan ecliptic plane and the y-axis completes the right handed system.

One has to note that the origin of the coordinate system is the bottom right corner of the simulation box as illustrated on figure 1. The position of the planet is indicated in the variable ‘s_centr’ (cf 1.2.3).

Mainly two other coordinate systems will be used at LATMOS SMDB :

- The MSO coordinate system. This coordinate is used for Mars It is centered on the planet the x-axis points from the Planet toward the Sun. The y-axis points in the opposite direction of the orbital velocity vector and the z-axis completes the right-handed system.
- The TIIS coordinate system. This coordinate system is used for Titan (orbservation/simulation). The x-axis follows the ideal magnetospheric flow direction, the y-axis points toward Saturn and the z-axis completes the right-handed system.

Figure 1 shows the different system for Mars.

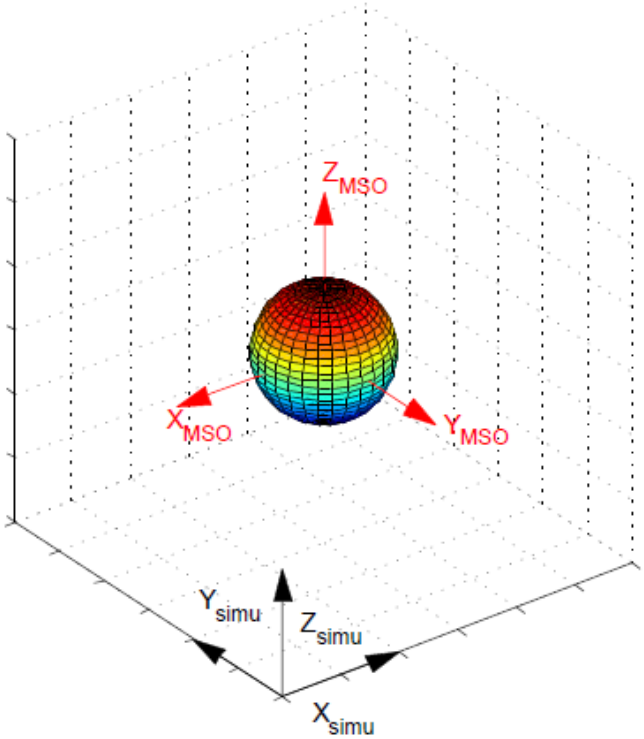

Figure 1 : Coordinate systems used for LATMOS SMDb. Example given for a Martian simulation.

1.2.2 Simulation unit and Normalization

The hybrid model solves a set of equation which has been nondimensionalized. Physical units are therefore removed and all quantities/variables do not have units. Each basic quantity has been parametrized with a combination of characteristical values.
The basic quantities are:
- time *t0*
- length *x0*
- speed *v0*
- Density *n0*
- Magnetic field *B0*
- Mass *m0*
- charge *q0*

These parameters are constants. To transform from simulation values to physical values, one has to applied the following method
Z_simulation*Z_0 => Z_physical
With Z_simulation the simulated quantity, Z_0 the characteristical parameter (or combination of characteristical parameters). The most used transformations are :

Density	n_phys = n_simu**n0*
Magnetic field	B_phys = B_simu**B0*
Velocity	V_phys = V_simu**V0*
Time	t_phys = t_simu**t0*
Length	x_phys = x_simu**x0*

Electric field E_phys = E_simu*v0*B0

…

A more detail information concerning parametrization and typical simulation parameter is provided in Annex 1.

1.2.3 Simulation variables

General information concerning each file can be accessed through netcdf functionalities. For instance in a terminal window the command "ncdump –h Magw_28_11_12_t00000.nc" returns information concerning the "Magw_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.
In front of each line we have indicated the meaning and color coded in green if this information might be useful for the IMPEx user or red if the information is a "technical" information. Some of these information are already mentioned in the METADATA (cf DataModelDefinition-0.9.doc).

"Magw_XXXX.nc":

```
$ ncdump -h Magw_28_11_12_t00000.nc
netcdf Magw_28_11_12_t00000 {
dimensions:
        dim_scalar = 1 ;                        Dimension of a scalar
        bidimensional = 2 ;                     Dimension of a vector
        space_dimension = 3 ;                   Dimension of a 3D matrix
        mpi_ndims = 2 ;                         Number of dimension for MPI's domain
        mpi_nb_voisin = 8 ;                     Number of neighbours for each process
        size_x = 202 ;                          Max grid points in the X-direction
        size_y = 386 ;                          Max grid points in the Y-direction
        size_z = 386 ;                          Max grid points in the Z-direction
        particles coords = 11 ;
        nptot = 140970426 ;                     # of particles in the simulation.
        number of diag. = 200 ;                 Max number of diagnostic time
        dim_1 = 1 ;                             Scalar dimension
        ns = 2 ;                                # of incident ion species
        nfl = 1000 ;                            Variable of injected distrib. Function
        dimname = 20 ;                          Max # character for the sim. Name
        dimplanet = 20 ;                        # of derived parameters for the planet
        dim_3 = 3 ;                             Space dimension
variables:
        float gstep(space_dimension) ;          Spatial step dx=gstep(1),dy=gstep(2)…
        int iter(dim_scalar) ;                  # of time steps performed
        float dt_t(bidimensional) ;             Time step value and time value
        int nptot(dim_scalar) ;                 # of particles in the simulation
        int ns(dim_1) ;                         # of incident ion species
        int nfl(dim_1) ;                        Related to distribution function
        char planetname(dimplanet) ;            Planet name : 'Mars','Titan'…
        char speciesname(ns, dimname) ;         Ion species name
        float phys_density(dim_1) ;             Reference density (in m-3)
        float phys_mag(dim_1) ;                 Reference magnetic field (in T)
        float phys_speed(dim_1) ;               Reference speed (in km/s)
        float phys_length(dim_1) ;              Reference length (in m)
        float phys_time(dim_1) ;                Reference time (in s)
        float absorpt_len(dim_1) ;
        float s_centr(dim_3) ;                  Center position of the obstacle
        float r_planet(dim_1) ;                 Radius of the planet
        float r_lim(dim_1) ;                    Radius of the obstacle
        float r_exo(dim_1) ;                    Radius of the exobase level
        float r_iono(dim_1) ;                   Radius of ionospheric chemistry calc.
        float v_planet(dim_3) ;                 Velocity of the planet
```

```
        int np(ns) ;                                    #of particle per cell (at init.)
        int ng(ns) ;                                    Start particle index (at ini)
        int n1(ns) ;                                    Stop particle index (at init)
        int n2(ns) ;
        float betae(dim_1) ;                            Betae of the plasma
        float betas(ns) ;                               Ion betas for undisturbed SW plasma
        float rvth(ns) ;                                Ratio betw. therm. speed(H+ and He++)
        float vth1(ns) ;                                Thermal speed (parallel)
        float vth2(ns) ;                                Thermal speed (perp)
        float rmds(ns) ;                                Total Macropart. mass in 1 cell (H+)
        float qms(ns) ;                                 Charge / mass
        float sq(ns) ;                                  Macroparticle charge (in SW)
        float sm(ns) ;                                  Macroparticle mass (in SW)
        float vxs(ns) ;                                 X- Bulk speed (in SW)
        float vys(ns) ;                                 Y-Bulk speed (in SW)
        float vzs(ns) ;                                 Z-bulk speed (in SW)
        float flux_tot(ns) ;                            Numerical SW particle input flux
        float percent(ns) ;                             Ratio between H+ and He++
        float ratios_charges(ns) ;                      Charge ratio
        float ratios_masses(ns) ;                       Mass ratio
        float THesTH(ns) ;                              Temperature ratio
        float vtHesvtH(ns) ;                            Thermal speed ratio
        float prob_extract(ns) ;                        Probabibility of extractaion
        float vplus(ns, nfl) ;                          Related to the injected distr. Funct.
        float s_min(space_dimension) ;                  Min values of the box size
        float s_max(space_dimension) ;                  Max values of the box size
        float Bfield_x(size_z, size_y, size_x)          Bx values :  magnetic field  (*)
;
        float Bfield_y(size_z, size_y, size_x)          By values : magnetic field   (*)
;
        float Bfield_z(size_z, size_y, size_x)          Bz values : magnetic field   (*)
;

// global attributes:
                :file_format = "NetCDF" ;
                :file_format_version = 1.4f ;
                :Conventions =
"http://www.unidata.ucar.edu/software/netcdf/"
;
                :Program = "Quiet Plasma" ;
                :Title = "Field" ;
                :Date = "28_11_12" ;
}
```

(*) The dataset dimensions as viewed by “ncdump” should be in the reverse order from the variable dimensions in fortran. The recorded values have the dimension (size_x,size_y,size_z). This is not a mistake. The “ncdump” utility uses the C-language convention for displaying array dimension. That means that when an array have more than one dimension and the programming language is fortran, the dimensions are listed in the reverse order of the way the array would be used in IDL, Matlab, …
More information concerning the array section access for Fortran are given at : http://www.unidata.ucar.edu/software/netcdf/docs/netcdf/Fortran-Section-Access.html

A documentation of the “ncdump” command is available at : http://www.unidata.ucar.edu/software/netcdf/docs/ncdump-man-1.html

A similar structure has been adopted for all netcdf files. In the following only the differences to this example are provided and explained:

“Elew_XXXX;nc”

$ ncdump -h Elew_28_11_12_t00000.nc netcdf Elew_28_11_12_t00000 {… float Efield_x(size_z, size_y, size_x) ; float Efield_y(size_z, size_y, size_x) ; float Efield_z(size_z, size_y, size_x) ; …	Ex values: electric field Ey values: electric field Ez values: electric field

"Thew_XXXX.nc"

$ ncdump -h Denw_28_11_12_t00000.nc
netcdf Denw_28_11_12_t00000 {
…
float Dne(size_z, size_y, size_x) ;
…
	Electron number density values

"Hsw_XXXX.nc","Hesw_XXXX.nc","Hpl_XXXX.nc","Opl_XXXX.nc","O2pl_XXXX.nc", "CO2_XXXX.nc"

$ ncdump -h Hesw_28_11_12_t00000.nc
netcdf Hesw_28_11_12_t00000 {
…
int npm(dim_scalar) ;
int nb_ion_species(dim_scalar) ;
float Density(size_z, size_y, size_x)
;
float Vx(size_z, size_y, size_x) ;
float Vy(size_z, size_y, size_x) ;
float Vz(size_z, size_y, size_x) ;
float Temperature(size_z, size_y,
size_x) ;
…
	Max number of particles
	# of different ion species
	Number density of the ion species"
	
	Vx bulk speed of the ion species
	Vx bulk speed of the ion species
	Vx bulk speed of the ion species
	Temperature of the ion species

"Atmw_XXXX.nc"

$ ncdump -h Atmw_28_11_12_t00000.nc
netcdf Atmw_28_11_12_t00000 {
…
float Den_CO2+(size_z, size_y, size_x)
;
float Den_O+(size_z, size_y, size_x) ;
float Den_H(size_z, size_y, size_x) ;
float Den_O2+(size_z, size_y, size_x)
;
float Den_O(size_z, size_y, size_x) ;
float Den_CO2(size_z, size_y, size_x)
;
	CO2+ density
	
	O+ density
	Neutral H density
	O2+ density
	
	Neutral O density
	Neutral CO2 density

"p3_MMM_XXXX.nc"

"p3" files store the information about the particles.There are as much "p3" files for 1 diagnostic time that the number of process used to performed the simulation. Each "p3_MMM" files stores only information concerning particles which are in the sub-

domain managed by the “MMM” processus. Particles position in such files refer however to the position in the global domain.

```
$ ncdump -h p3_000_28_11_12_t00000.nc
netcdf p3_000_28_11_12_t00000 {
dimensions:
        dim_scalar = 1 ;                        Dimension of a scalar
        bidimensional = 2 ;                     Dimension of a vector
        space_dimension = 3 ;                   Dimension of a 3D matrix
        mpi_ndims = 2 ;                         Number of dimension for MPI’s domain
        mpi_nb_voisin = 8 ;                     Number of neighbours for each process
        size_x = 202 ;                          Max grid points in the X-direction
        size_y = 69 ;                           Max grid points in the Y-direction
        size_z = 69 ;                           Max grid points in the Z-direction
        particles coords = 11 ;                 # of information for 1 particle
        nptot = 3591200 ;                       # of particles in the simulation.
        number of diag. = 20000 ;               Max number of diagnostic time
        dim_1 = 1 ;                             Scalar dimension
        ns = 2 ;                                # of incident ion species
        nfl = 1000 ;                            Variable of injected distrib. Function
        dimname = 20 ;                          Max # character for the sim. Name
        dimplanet = 20 ;                        # of derived parameters for the planet
        dim_3 = 3 ;                             Space dimension
variables:
        float gstep(space_dimension) ;          Spatial step dx=gstep(1),dy=gstep(2)…
        int iter(dim_scalar) ;                  # of time steps performed
        float dt_t(bidimensional) ;             Time step value and time value
        int nptot(dim_scalar) ;                 # of particles in the simulation
        int nproc(dim_scalar) ;                 # of processor used
        int mpiinfo_me(dim_scalar) ;            Rank of the processor
        int mpiinfo_dims(mpi_ndims) ;           Dimension of the decomposition domain
        int mpiinfo_coord(mpi_ndims) ;          Coord of the proc in decompos. domain
        int mpiinfo_voisin(mpi_nb_voisin) ;     Neighbor process
        int ns(dim_1) ;                         # of incident ion species
        int nfl(dim_1) ;                        Related to distribution function
        char planetname(dimplanet) ;            Planet name : ‘Mars’,’Titan’…
        char speciesname(ns, dimname) ;         Ion species name
        float phys_density(dim_1) ;             Reference density (in m-3)
        float phys_mag(dim_1) ;                 Reference magnetic field (in T)
        float phys_speed(dim_1) ;               Reference speed (in km/s)
        float phys_length(dim_1) ;              Reference length (in m)
        float phys_time(dim_1) ;                Reference time (in s)
        float absorpt_len(dim_1) ;
        float s_centr(dim_3) ;                  Center position of the obstacle
        float r_planet(dim_1) ;                 Radius of the planet
        float r_lim(dim_1) ;                    Radius of the obstacle
        float r_exo(dim_1) ;                    Radius of the exobase level
        float r_iono(dim_1) ;                   Radius of ionospheric chemistry calc.
        float v_planet(dim_3) ;                 Velocity of the planet
        int np(ns) ;
        int ng(ns) ;                            #of particle per cell (at init.)
        int n1(ns) ;                            Start particle index (at ini)
        int n2(ns) ;                            Stop particle index (at init)
        float betae(dim_1) ;                    Betae of the plasma
        float betas(ns) ;                       Ion betas for undisturbed SW plasma
        float rvth(ns) ;                        Ratio betw. therm. speed(H+ and He++)
        float vth1(ns) ;                        Thermal speed (parallel)
        float vth2(ns) ;                        Thermal speed (perp)
        float rmds(ns) ;                        Total Macropart. mass in 1 cell (H+)
        float qms(ns) ;                         Charge / mass
        float sq(ns) ;                          Macroparticle charge (in SW)
        float sm(ns) ;                          Macroparticle mass (in SW)
        float vxs(ns) ;                         X- Bulk speed (in SW)
        float vys(ns) ;                         Y-Bulk speed (in SW)
        float vzs(ns) ;                         Z-bulk speed (in SW)
        float flux_tot(ns) ;                    Numerical SW particle input flux
        float percent(ns) ;                     Ratio between H+ and He++
        float ratios_charges(ns) ;              Charge ratio
        float ratios_masses(ns) ;               Mass ratio
        float THesTH(ns) ;                      Temperature ratio
        float vtHesvtH(ns) ;                    Thermal speed ratio
```

float prob_extract(ns) ; float vplus(ns, nfl) ; int nxyzm(dim_scalar) ; int ncm_tot(space_dimension) ; int nc_tot(space_dimension) ; int ncm(space_dimension) ; float s_min(space_dimension) ; float s_max(space_dimension) ; int npm(dim_scalar) ; float particule_x(nptot) ; float particule_y(nptot) ; float particule_z(nptot) ; float particule_vx(nptot) ; float particule_vy(nptot) ; float particule_vz(nptot) ; float particule_mass(nptot) ; float particule_char(nptot) ; float particule_exc(nptot) ; int particule_orig(nptot) ; int particule_dir(nptot) ; // global attributes: :file_format = "NetCDF" ; :file_format_version = 1.4f ; :Conventions = "http://www.unidata.ucar.edu/software/netcdf/" ; :Program = "Quiet Plasma" ; :Title = "Particles" ; :Date = "28_11_12" ;	Probabibility of extractaion Related to the injected distr. Funct. Xcell number*Ycell number*Zcell number Global # grid point in X,Y and Z dir Global # cell in X, Y and Z dir Local (subdomain) # cell Min values of the box size Max values of the box size Max # of particles that can tracked X position of the particle Y position of the particle Z position of the particle Vx velocity comp. of the particle Vy velocity comp. of the particle Vz velocity comp. of the particle Numerical mass of the particle Numerical charge of the particle Charge exchange tag for the particle Origin of the particle Direction indicating the part goes (MPI)

2. Specific IMPEx diagnostic

The data format and the data description of the following diagnostic are opened for discussion. Daignostic files are sometimes written in simulation unit, sometimes in physical unit, sometimes in MSO coordinate system sometimes in the simulation coordinate system. At the end we expect that all data information will be provide in physical units in the MSO coordinate system for Mars and TIIS for Titan.
The VOTable standard has been one of the standard selected as IMPEX diagnostic standard and format. The following part describes 1D and 2D VOtable files archived at LATMOS. Conventions used for the VOTable are the ones defined by IVOA (http://www.ivoa.net/Documents/VOTable/).

2.1 "Cuts"

This diagnostic has been implemented and tested. It generates three VOTable-like files for a given quantity (eg magnetic field) in a 3 planes (XY, XZ and YZ) at a given value/quotation in the third direction. The diagnostic is implemented for Magnetic field, Electric field and bulk speed.

The ascii files generated are:

- "Magw_XXXX.nc" ⇒ "Bfield_XY_XXXX.xml", "Bfield_XZ_XXXX.xml", "Bfield_YZ_XXXX.dat"
- "Elew_XXXX.nc" ⇒ "Efield_XY_XXXX.xml", "Efield_XZ_XXXX.xml", "Efield_YZ_XXXX.xml"

- “Thew_XXXX.nc” ⇒ “Thew_XY_XXXX.xml”, “Thew_XZ_XXXX.xml”, “Thew_YZ_XXXX.xml”
- ….

2.1.1 Data format

As mentioned above data in this diagnostic are compliant with the VOTable standard defined by IVOA.

2.1.2 Data description

An example of a file structure for a given cut is below (ex: Vbulk_XY_XXXX.xml).
Two parts can be identified.
The header defining the structure of the dataset

```
<?xml version="1.0"?>
<VOTABLE version="1.2" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.ivoa.net/xml/VOTable/v1.2" xmlns:stc="http://www.ivoa.net/xml/STC/v1.30" >
<RESSOURCE name="IMPEx 2D cut">
<TABLE name="results">
<FIELDref ref="col1"/>
<FIELDref ref="col2"/>
<FIELDref ref="col3"/>
</GROUP>
<DESCRIPTION> Velocity components </DESCRIPTION>
<FIELD          name="X"          ID="col1"          ucd="pos.cartesian.x"          ref="MSO          "
utype="stc:AstroCoords.Position3D.Value3.C1" datatype="float" width="9">
<FIELD          name="Y"          ID="col2"          ucd="pos.cartesian.y"          ref="MSO          "
utype="stc:AstroCoords.Position3D.Value3.C2" datatype="float" width="9">
<FIELD          name="Z"          ID="col3"          ucd="pos.cartesian.z"          ref="MSO          "
utype="stc:AstroCoords.Position3D.Value3.C3" datatype="float" width="9">
<FIELD  name="Utot"  ID="col4"  ucd="phys.veloc"  ref="MSO  "  utype=""  datatype="float"  width="12"
unit="km.s-1">
<FIELD  name="Ux"  ID="col5"  ucd="phys.veloc"  ref="MSO  "  utype=""  datatype="float"  width="12"
unit="km.s-1">
<FIELD  name="Uy"  ID="col6"  ucd="phys.veloc"  ref="MSO  "  utype=""  datatype="float"  width="12"
unit="km.s-1">
<FIELD  name="Uz"  ID="col7"  ucd="phys.veloc"  ref="MSO  "  utype=""  datatype="float"  width="12"
unit="km.s-1">
<DATA>
<TABLEDATA>
```

Information required to read and load such file are provided in the header. For instance it indicates the resource name “IMPEx 2D cuts”, a summary description of the file (“Velocity components”) and the meaning of each columns.
The first column of the dataset is refer to the ID “col1” which has the name “X” and provide x abscissa of the data in a Cartesian coordinate system. These data will have a “float” attribute. The unified content descriptor (ucd) and unified type (utype) have already been defined by IVOA (http://cdsweb.u-strasbg.fr/UCD/ucd1p-words.txt and http://www.ivoa.net/Documents/Notes/VOTableSTC/20100618/NOTE-VOTableSTC-2.0-20100618.pdf).

The core of the file correspond to the dataset :

```
<TR>
<TD>    3.084</TD> <TD>    4.965</TD> <TD>   -0.008</TD> <TD>     488.408</TD> <TD>    -488.347</TD> <TD>       3.791</TD> <TD>       6.743</TD>
</TR>
<TR>
<TD>    3.084</TD> <TD>    4.941</TD> <TD>   -0.008</TD> <TD>     491.716</TD> <TD>    -491.594</TD> <TD>       1.649</TD> <TD>      10.823</TD>
</TR>
```

```
<TR>
<TD>     3.084</TD> <TD>     4.916</TD> <TD>    -0.008</TD> <TD>      494.299</TD> <TD>     -494.240</TD> <TD>       -0.819</TD> <TD>         7.580</TD>
</TR>
<TR>
<TD>     3.084</TD> <TD>     4.892</TD> <TD>    -0.008</TD> <TD>      491.964</TD> <TD>     -491.923</TD> <TD>        3.178</TD> <TD>         5.519</TD>
</TR>
...
```

Annex 1 : Parametrization

Incident Plasma : Typical Parameters

Solar Wind / Magnetospheric Plasma

Few plasma parameters definition (in reference to ion specie 'i' impigned in a magnetic field B_0) :
Ion intertial length

$$\frac{c}{\omega_{pi}} = c\sqrt{\frac{\epsilon_0 m_i}{e^2 n_i}}$$

ion cyclotron frequency

$$\Omega_i = \frac{q_i B_0}{m_i}$$

thermal speed

$$v_{thi} = \sqrt{\frac{2k_B T_i}{m_i}}$$

Alfvèn speed of specie 'i'

$$V_A = \frac{B_0}{\mu_0 m_i n_i}$$

Sound speed

$$C_s = \sqrt{\frac{3T_e}{m_i}}$$

Ion specie 'i' plasma beta :

$$\beta_i = \frac{n_i k_B T_i}{\frac{B_0^2}{2\mu_0}}$$

Alfvèn Mach number

$$M_A = \frac{V_{incplasma}}{V_A}$$

Sonic Mach number

$$M_S = \frac{V_{incplasma}}{C_s}$$

Magnetosonic Mach number

$$M_{MS} = \frac{M_A M_s}{\sqrt{M_A^2 + M_s^2}}$$

Mars

Typical solar wind plasama parameters at the Martian orbit :
The main ion specie is H^+. Simulation values have been normalized with respect to H^+ parameters.
IMF :$B_{IMF} = (1.6, 2.5, 0.)nT$
Bulk speed : $V_{sw} = (400, 0, 0)km/s$
Electron temperature : $T_e = 3.10^5 K$.

H^+	He^{++}
m_{H^+}	$4m_{H^+}$
q_{H^+}	$2q_{H^+}$
$n_{H^+} = 2.5$ cm^{-3}	$n_{He^{++}} = 0.05 n_{H^+}$
$T_{H^+} = 5.10^4$ K	$T_{He^{++}} = 4T_{H^+}$

The derived quantites are listed in table 0.1.

EC Project Nr.: **262863**