PLANETOCOSMICS Software User Manual

Title:	**PLANETOCOSMICS Software User Manual**
Project reference:	
Prepared by:	Dr. Laurent Desorgher
Signature:	
Date:	
Authorised by:	
Signature:	
Issued by: Sector/location/telephone/fax:	Physikalisches Institut University of Bern Sidlerstrasse 5 CH 3012 Bern Switzerland Tel: ++41 31 631 4053 Fax: ++41 31 631 4404

Abstract

This document provides all the necessary information to allow the user to install and operate the PLANETOCOSMICS code. A general description of the software is given, followed by an extensive coverage of the commands available. Some tutorial examples provided with the code are described at the end.

Record of changes

Issue	Date	Detail of changes
0.1	2005 -06-14	First issue to reviewers in draft form

Table of contents

Table of Contents

1 Introduction

1.1 Contractual
This document has been issued by the Physikalisches Institut of the University of Bern to ESA/ESTEC under contract .

1.2 Purpose of the Document
This document is the Software User Manual (SUM) for the PLANETOCOSMICS Geant4 application.

1.3 Scope of the Software
The PLANETOCOSMICS software is only intended for use with the Geant4 code for Monte Carlo, high-energy particle transport.

1.4 Definitions, acronyms and abbreviations
ASCII American Standard Code for Information Interchange

AIDA Abstract Interface for Data Analysis

CERN Conseil Européen pour la Recherche Nucléaire

DGRF Definitive Geomagnetic Reference Field

ESA European Space Agency

GEANT4 C++ toolkit for Monte Carlo simulation of high-energy, fundamental particle transport, developed by an international collaboration led by CERN.

GUI Graphical User Interface

IAGA International Association of Geomagnetism and Aeronomy

IGRF International Geomagnetic Reference Field

NAIF Navigation and Ancillary Information Facility

OO Object-Oriented

UI User Interface

UML Unified Modeling Language

UR User Requirement

URD User Requirement Document

JAS Java Analysis Studio

1.5 References

[1] The CERN Geant4 Collaboration provides a significant amount of information at the web site: http://wwwinfo.cern.ch/asd/geant4/geant4.html. From this Web page, access can be obtained to the following User Documentation and web page:

[2] Geant4 User Guide for Application Developers: http://wwwinfo.cern.ch/asd/geant4/geant4_public/G4UsersDocuments/UsersGuides/ForApplicationDeveloper/html/index.html

[3] Geant4 User Guide for Toolkit Developers: http://wwwinfo.cern.ch/asd/geant4/geant4_public/G4UsersDocuments/UsersGuides/ForToolkitDeveloper/html/index.html

[4] The Geant4 Physics Reference manual: http://wwwinfo.cern.ch/asd/geant4/geant4_public/G4UsersDocuments/UsersGuides/PhysicsReferenceManual/html/index.html

[5] Web site for educated guess physics list for Geant4 Hadronic physics: http://cmsdoc.cern.ch/~hpw/GHAD/HomePage/

[6] The Software Reference manual provides information on the public methods to the Geant4 classes: http://geant4.kek.jp/../cgi-bin/G4GenDoc.csh?flag=1

[7] Information on AIDA can be found on http://aida.freehep.org

[8] The ROOT analysis package can be downloaded from http://root.cern.ch

[9] The ANAPHE web site: http://anaphe.web.cern.ch/anaphe/

[10] JAIDA: http://java.freehep.org/jaida/index.html

[11] AIDAJNI : http://java.freehep.org/aidajni/index.html

[12] Russell, C. T., Geophysical Coordinate Transformations, *Cosmic Electrodyn.,* **2**, 184, 1971

[13] Hapgood, M. A., Space Physics Coordinate Transformations: a User Guide, *Planet. Space Sci.*, **40**, No 5, 711-717, 1992

[14] Fränz, M. and D. Harper, Heliospheric coordinate system, *Planet. Space Sci.*, **50**, 217-233, 2002

[15] SPICE Information System, http://naif.jpl.nasa.gov/naif/

[16] Simon J. L., P. Bretagon, J. Chapront, M. Chapront-Touzé, G. Francou, and J. Laskar, Numerical expressions for precession formulae and mean elements for the Moon and the planets, *Astron. Astrophys.*, **282**, 663-683, 1994

[17] Astronomical Almanac, Astronomical Almanac for the Year 2001, US & H:M. Nautical Almanac Offices, Washington & London, 2000

[18] Hedin A. E., Extension of the MSIS thermosphere model into the middle and lower atmosphere, *J. Geophys. Res.*, **96**, A2, p. 1159-1172, 1991.

[19] Picone J. M., A. E. Hedin, D. P. Drob, and A. C. Aikin, NRLMSISE-00 empirical model of the atmosphere: Statistical comparisons and scientific issues, *J. Geophys. Res.,* **107**, A12, 1468, doi:10.1029/2002JA009430, 2002.

[20] Justus, C. G., and D. L. Johnson, Mars Global Reference Atmospheric Model 2001 Version (Mars-GRAM 2001): Users Guide, George C. Marshall Space Flight Center Marshall Space Flight Center, AL 35812, National Aeronautics and Space Administration Washington, DC, 20546-0001, 2001, http://trs.nis.nasa.gov/archive/00000549/

[21] Haberle, R. M., J. B. Pollack, J. R. Barnes, et al., Mars atmospheric dynamics as simulated by the NASA Ames General Circulation Model, 1. The zonal-mean circulation, *J.Geophys.Res,* **98**, E2, 3125-3148, 1993

[22] Bougher, S. W., R. G. Roble, E. C. Ridley, et al., The Mars thermosphere: 2. General circulation with coupled dynamics and composition, *J. Geophys. Res.*, **95**, B9, 14811-14827,1990

[23] The Martian Climate Database, http://www-mars.lmd.jussieu.fr/

[24] Forget F., F. Hourdin, R. Fournier, C. Hourdin, O. Talagrand, M. Collins, S. R. Lewis, P. L. Read, and J.-P. Huot, Improved general circulation models of the Martian atmosphere from the surface to above 80 km, *J.Geophys.Res.*, **104**, 24,155-24,176, 1999.

[25] Langel R. A., Main Field in Geomagnetism, 249-512, vol I, ed. J.A. Jacobs, Academic Press, London, 1987.

[26] Fraser-Smith, A. C., Centered and Eccentric Geomagnetic Dipoles and Their Poles, 1600-1985, *Rev. Geophys.*, **25**, 1-16, 1987.

[27] IGRF model web site of the International Association of Geomagnetism and Aeronomy (IAGA): http://www.ngdc.noaa.gov/IAGA/vmod/igrf.html

[28] Wolf, R. A., Magnetospheric configuration, *in Introduction to Space Physics, ed. Kivelson M. G. and Russell C. T., Cambridge Univ. Press*, p. 288-329.

[29] The different Tsyganenko models (89, 96 and 2001) can be downloaded as FORTRAN code form the url http://nssdc.gsfc.nasa.gov/space/model/magnetos/data-based/modeling.html

[30] Tsyganenko, N. A., Global quantitative models of the geomagnetic field in the cislunar magnetosphere for different disturbance levels, *Planet. Space Science,* **35**, 1347, 1987

[31] Tsyganenko, N. A., A magnetospheric magnetic field model with a warped tail current sheet, *Planet. Space Sci.,* **37**, 5, 1989

[32] Tsyganenko, N. A., Modeling the Earth's magnetospheric magnetic field confined within a realistic magnetopause, *J. Geophys. Res.,* **100**, 5599, 1995

[33] Tsyganenko, N. A., Effects of the solar wind conditions on the global magnetospheric configuration as deduced from data-based field models, *Eur. Space Agency Spec. Publ., ESA SP-389,* 181*,* 1996

[34] Tsyganenko, N. A. , A model of the near magnetosphere with a dawn-dusk asymmetry, 1. Mathematical structure, *J. Geophys. Res.,***107***,* No A8, 10.1029/2001JA000219

[35] Tsyganenko, N. A. , A model of the near magnetosphere with a dawn-dusk asymmetry, 2. Parameterization and fitting to observations, *J. Geophys. Res.,***107***,* No A8, 10.1029/2001JA000220

[36] Acuña, M. H., et al., Magnetic field of Mars: Summary of results from the aerobraking and mapping orbits, *J. Geophys. Res.,* **106***,* 23,403-23,417, 2001

[37] Connerney, J. E. P., M. H. Acuña, P. Wasilewski and G. Kletetschka, The global magnetic field of Mars and implications for crustal evolution, *Geophys. Res. Let.,* **28***,* 4015-4018, 2001

[38] Purucker M., D. Ravat, H. Frey, C. Voorhies, M. Acuña , An altitude-normalized magnetic map of Mars and its interpretation, *Geophys. Res. Let,* **27***,*2449-2452, 2000

[39] Cain, J.C., B. B. Ferguson, and D. Mozzoni, An n=90 internal potential function of the Martian crustzal magnetic field, *J. Geophys. Res.,* **108,** E2*,* 5008,doi:10.1019/2000JE001487, 2003

[40] Arkhani-Hamed, J., A 50 degree spherical harmonic model of the magnetic field of Mars, *J. Geophys. Res.,* **106***,* 23-291-23,316, 2001

[41] Mitchell, D. L., et al., Probing Mars' crustal magnetic field and ionosphere with the MGS electron reflectometer, *J. Geophys. Res.*, **106**, 23,419, 2001

[42] Lepping, R. P., N. F., Ness, and K. W. Behannon, Summary of Mariner 10 magnetic field and trajectory data for Mercury I and III encounters, NASA-GSFC, TM-80600, Greenbelt, MD, 1979.

[43] Connerney J. E. P., and N. F. Ness, Mercury's magnetic field and interior, *in Mercury*, editors F. Vilas, C. R. Chapman and M. S. Matthews, The university of Arizona Press, Tucson, 494 1988.

[44] Russell C. T., D. N. Baker, and J. A. Slavin, The magnetosphere of Mercury, *in Mercury*, editors F. Vilas, C. R. Chapman and M. S. Matthews, The university of Arizona Press, Tucson, 494 1988.

[45] Lei F., User manual for the Geant4 general particle source http://reat.space.qinetiq.com/gps/new_gps_sum_files/gps_sum.htm

[46] Gleeson, L. J. and W. I. Axford, Solar Modulation of Galactic Cosmic Rays, *Astrophys. J., 154,* 1011, 1968.

[47] Garcia-Munoz, M., G. M. Mason, and J.A. Simpson, "The Annomalous ^{4}He component in the Cosmic Ray Spectra of ≤ 50 MeV per Z nucleon During 1972-1974, *Astrophys. J., 202,* 265, 1975.

[48] Cooke, D. J., Humble, J. E, Shea, M. A., Smart, D. F., Lund, N., Rasmussen, I. L., Byrnak, B., Goret, P., and Petrou, N.: 1991, On cosmic-ray cutoff terminology, *Il Nuovo Cimento, 14C,* 213-234, 1991

[49] 'Runge-Kutta Method', *in 'Numerical Recipes in C++, The Art of Scientific Computing',* Press W. H., Teukolsky S. A., Flannery B. P., Vetterling W. T., 2nd –ed, Cambridge University Press, 2001, p. 715

[50] Boyce, J. M., The Smithsonian Book of Mars, Smithsonian Institution Press Washington and London, 2002

[51] Goettel, K. A., Present bounds on the bulk composition of Mercury: Implications for planetary formation processes, *in Mercury*, editors F. Vilas, C. R. Chapman and M. S. Matthews, The university of Arizona Press, Tucson, 1988.

2 Short description of the code

The PLANETOCOSMICS Geant4 application allows to compute the hadronic and electromagnetic interactions of cosmic rays with the Earth, Mars and Mercury environment. For each planet it is possible to take into account the presence of the planetary magnetic field, atmosphere and soil. Following the planet that is considered different magnetic field models and atmospheric models are available. The code has been developed such that the addition of new models should be rather simple.

The main applications of the code are :

- The computation of flux of particles resulting from the interaction of cosmic rays with a planet environment at user defined altitudes, and/or atmospheric depths.
- The computation of the propagation of charged particles in the planet magnetosphere.
- The computation of cutoff rigidity (mainly for the Earth) at given position on a planet and for different direction of incidence.
- The visualisation of magnetic field lines, and the trajectories of primary and secondary particles in the planet environment.

3 Installation of the code

The PLANETOCOSMICS software is a Geant4 application. For this reason before installing the code the user should install the Geant4 toolkit [1-4]. The files needed to install PLANETOCOSMICS are in the tar file PLANETOCOSMICS.tar.gz. From the directory where you want to install the source code of PLANETOCOSMICS, you should type "*tar –xvzf tarfilepath/PLANETOCOSMICS.tar.gz*" where *tarfilepath* define the directory where the file PLANETOCOSMICS.tar.gz is located. It creates in the current directory the directory ./planetocosmics. In this directory you should find the PLANETOCOSMICS.cc main file, the configuration shell script "configure.sh", the files "makefile_aida" and "makefile_root", and different directories ./planetocosmics/src, ./planetocosmics/include, ./planetocosmics/lib, ./planetocosmics/data, ./planetocosmics/examples, ./planetocosmics/fortran, ./planetocosmics/doc, and ./planetocosmics/mars. The interface files and source files of the code are contained in the ./planetocosmics/include and ./planetocosmics/src directories, respectively. The ./planetocosmics/lib directory contains different additional libraries needed to run the code. The ./planetocosmics/data directory contains different tables needed for the different magnetic field models implemented in the code. The ./planetocosmics/examples directory contained different g4 macro files representing tutorial examples, and some additional files needed to run these examples. This user guide of the code is contained in the directory ./planetocosmics/doc.

We have developed two different versions of PLANETOCOSMICS that differ in the type of libraries that have been used for histogramming purpose. In the AIDA version of PLANETOCOSMICS the analysis part of the code has been developed in compliance with the AIDA3.0.0 interface [7]. In the ROOT version the analysis part is using the ROOT package for storage of the results [8].

Before compiling and installing the AIDA version of the code, the user should have installed an AIDA compliant library. Usually such library is provided with an aida-config script. If this is not the case the user should modify the makefile_aida to link the code with his selected AIDA compliant library and to declare the directory where the AIDA interface files are on the system. For testing the AIDA version of PLANETOCOSMICS we have used a static version of the ANAPHE library. The ANAPHE library is a C++ library that was developed by the CERN IT division for replacing the FORTRAN CERNLIB [9]. Another possibility would be to use the AIDAJNI library that allows to link a C++ code compliant to AIDA with a Java AIDA implementation [10,11]. The A01 advance example of the Geant4.6.x release illustrates how to link a Geant4 code compliant to AIDA with the JAIDA (Java AIDA) package by using AIDAJNI.

Before compiling and linking the ROOT version of the code, the user should have installed the ROOT package and set the ROOTSYS environment variable to the directory where it is installed [8].

The PLANETOCOSMICS code makes use of the Geant4 Packaging library. This library is not compiled during the standard installation of Geant4. For the Geant4.7.x versions the Packaging library is compiled by typing make from the directory G4INSTALL/physics_lists/hadronic/Packaging where G4INSTALL defines the directory where Geant4 is installed. For the Geant4.6.x versions it is compiled by typing make from the G4INSTALL/hadronic_lists/lists/Packaging directory. Providing that all external libraries and packages have been correctly installed, the PLANETOCOSMICS code is compiled by doing the following :

1. From the directory where PLANETOCOSMICS is installed type either "./configure.sh aida " for an installation of the AIDA version or "./configure.sh root" for an installation of the ROOT version.

2. If the ./configure.sh script has been successfully executed, from the same directory type "make".

3. Add the command "source PLANETOCOSMICS_dir/setupPLANETOCOSMICS.sh" to the bash startup file (most probably $HOME/.bashrc) where PLANETOCOSMICS_dir should be replaced by the directory where PLANETOCOSMICS is installed.

4 Execution of the code

The code is executed by typing PLANETOCOSMICS <*planet_name* [*macrofile*]> . The *planet_name* argument should be one of the following candidates: Earth, Mars, or Mercury. If you do not provide a macrofile argument you will use PLANETOCOSMICS in the G4 interactive mode. When providing a macrofile argument, the commands given in the macrofile are executed in batch mode. You can interact with PLANETOCOSMICS by using standard Geant4 UI commands and the additional commands provided in the directory /PLANETOCOSMICS. These additional commands are described in the rest of this manual.

5 Some definitions

In this chapter we define different notions used trough all this document.

5.1 Definition of rigidity

The motion of a charged particle trough a magnetic field is described by the Lorentz equation of motion

$$d\vec{p}/dt = q\vec{v}\times\vec{B} \quad (5.1)$$

where $\vec{p}$, q, $\vec{V}$ and $\vec{B}$ represent the particle momentum, the particle charge, the particle velocity vector, and the magnetic field respectively. This equation of motion conserves p the magnitude of the momentum, and therefore the energy of the particle. After some transformations the equation of motion becomes

$$d\vec{I}_v/ds = \frac{q}{p}\vec{I}_v\times\vec{B} \quad (5.2)$$

where $\vec{I}_v$ represents the velocity direction and s is the path length along the particle trajectory. The rigidity of a particle is defined by $\frac{pc}{q}$ where c represents the velocity of light. Equation 8 shows that for the same initial position and direction, charged particles with the same rigidity and with charges of the same sign have identical trajectories. For this reason it is more convenient to characterise the trajectory of cosmic rays in function of their rigidity and not of their energy. The rigidity is an energy divided by a charge, in cosmic ray physics it is generally expressed in GV or MV.

5.2 Definition of fluxes

To avoid any confusion we specify in this section which definitions of fluxes are considering all over this document.

The direction of observation is specified from the vertical direction by the zenith and azimuth angles (θ, ϕ) .

The differential directional intensity or the differential directional flux $j(\theta, \phi, E)$ is defined as the number of particles within the energy range *[E,E+dE]*, crossing per unit time *dt,* an element of surface *dA* perpendicular to the direction of observation, within a solid angle element $d\Omega$ centered on the direction of observation

$$j(\theta, \phi, E) = \frac{dN}{dA\, dt\, d\Omega\, dE} [cm^{-2}\, s^{-1}\, sr^{-1}\, MeV^{-1}] \quad (5.3)$$

The integral directional flux $J_{E>E1}(\theta, \phi)$ represents the number of particles with energy greater than E_1, crossings per unit time *dt,* an element of surface *dA* perpendicular to the direction of observation, within a solid angle element $d\Omega$ centered on the direction of observation

$$J_{E \geq E1}(\theta, \phi) = \frac{dN_{E \geq E1}}{dA\, dt\, d\Omega} = \int_{E \geq E_1} j(\theta, \phi, E) dE \quad [cm^{-2}\, s^{-1}\, sr^{-1}] \quad (5.4)$$

In general an integral flux corresponds to the integration over a given energy range of a differential flux.

The omnidirectional differential flux $j^{4\pi}(E)$ is defined as the number of particles within the energy range *[E,E+dE]*, crossings per unit time *dt,* a sphere with a crossing surface of unit area *dA*. From the equation (4) we obtain

$$j^{4\pi}(E) = \int_0^{2\pi} \int_0^{\pi} j(\theta, \phi, E) \sin\theta\, d\theta\, d\phi = \int_0^{4\pi} j(\Omega, E)\, d\Omega \quad [cm^{-2}\, s^{-1}\, MeV^{-1}] \quad (5.5)$$

In the case of an isotropic flux $j(\theta, \phi, E) = j(E)$ and therefore $j^{4\pi}(E) = 4\pi\, j(E)$.

The omnidirectional integral flux

$$J^{4\pi}_{E \geq E_1} = \int_{E \geq E_1} j^{4\pi}(E) dE \quad [cm^{-2} s^{-1}] \quad (5.6)$$

represents the number of particles with kinetic energy $>E_1$, crossings per unit time *dt,* a sphere with a crossing surface of 1 cm^{-2}. In the case of an isotropic flux $J^{4\pi}_{E \geq E_1} = 4\pi J_{E \geq E_1}$

The downward flux $J^{down}_{E_1-E_2}$ and the upward flux $J^{up}_{E_1-E_2}$ represent the number of particles in the energy range *[E₁ ,E₂]* crossing an horizontal surface of unit area, per unit time, downward and upward respectively . From the equation (5.6) we get

$$J^{down}_{E_1-E_2}=\int_0^{\pi/2}\int_0^{2\pi}\int_{E_1}^{E2} j(\theta,\phi,E)\cos(\theta)\sin(\theta)\,d\theta\,d\phi\,dE \quad [cm^{-2}s^{-1}] \quad (5.7)$$

and

$$J^{up}_{E_1-E_2}=\int_{\pi/2}^{\pi}\int_0^{2\pi}\int_{E_1}^{E2} j(\theta,\phi,E)\cos(\theta)\sin(\theta)\,d\theta\,d\phi\,dE \quad [cm^{-2}s^{-1}] \quad (5.8)$$

respectively.

A fluence is defined by a flux integrated over time.

6 Space coordinate systems

Different Earth's space coordinate systems have been defined and are used in solar-terrestrial physics since the early year of space era [12-13]. These space coordinate systems have been extended to other solar planets [14]. In this section we define the coordinate systems that can be used in PLANETOCOSMICS and present their equivalent to the Earth coordinate systems. An extended review on Planetary coordinate system has been written by Fränz and Harper [14]. This paper has been the principal source for our implementation of transformation of coordinate system in PLANETOCOSMICS.

The body fixed planetocentric (PLA) system associated to a specific planet is fixed with the rotation of this planet. The z-axis represents the rotation axis of the planet. The xz-plane contains the planet prime meridian (equivalent of the Earth's Greenwich meridian). The y-axis close the system. The longitude is measured eastward from the prime meridian. For the Earth the PLA system is equivalent to the geographic geocentric coordinates system (GEO).

The planetographic (PLAG) coordinates are used for cartographic purpose and define the position of a point by its planetographic altitude, longitude and latitude. In this system a surface of reference, generally an ellipsoid, define the mean surface of the planet. The vertical is the line passing trough the point that is perpendicular to the surface of reference. The altitude is defined by the distance from the point to the surface of reference along the vertical . The latitude is the angle between the vertical and the equatorial plane of the planet. The planetocentric and planetographic longitudes are equivalent. The GEODETIC coordinates represent the PLAG coordinates for the Earth.

The planetocentric solar orbital system (PSO) has its x-axis pointing from the planet center to the sun and the y-axis is chosen to be in the orbital plane of the planet pointing toward the dusk. The z-axis is perpendicular to the orbital plane. For the Earth the orbital plane is the ecliptic and the system therefore is called the geocentric solar ecliptic system (GSE).

The planetocentric solar equatorial system (PSEQ) has its x-axis pointing from the center of the planet to the Sun. The y axis is parallel to the Sun equatorial plane pointing toward dusk. The planet sun direction being not necessarily parallel to the equatorial plane the z-axis is not parallel to the Sun rotation axis. The Earth's equivalent system for PSEQ is the geocentric solar equatorial system (GSEQ).

For planets as the Earth, Mercury or Jupiter that have a global magnetic field, different coordinate systems oriented in function of the magnetic field configuration are defined. In these systems the global magnetic field of the planet is modeled by a magnetic dipole centered on the planet and with an axis tilted with the planet rotation axis.

We define the planetocentric magnetic coordinate system PMAG as the system fixed to the magnetic dipole of the planet. The z axis is parallel to the dipole axis pointing toward the planet north pole. The xz-plane contains the rotation axis of the planet and the y-axis closes the system.

The planetocentric solar magnetospheric system (PSM) has its x-axis that coincides with the planet-Sun direction. The xz-plane contains the planet magnetic dipole axis, so that the y-axis is perpendicular to this axis. The z-axis point toward the northern magnetic pole. For the Earth this system is called the geocentric solar magnetospheric system (GSM).

The planetocentric solar magnetic system (PSMAG) has the same y-axis and xz-plane than the PSM system but its z-axis coincides with the planet dipole axis. The difference between this system and the PSM system is a rotation about the y-axis. For the Earth this system is called the solar magnetic (SM) system.

To compute the transformation from one coordinate system to another one it is needed to know the position, the rotation axis and the prime meridian of the planets, as well as the the Sun rotation axis , in an inertial coordinate system. In PLANETOCOSMICS two different methods can be used to compute these information. In the first method the SPICE toolkit is used [15]. This toolkit developed by NAIF allows to compute the planet ephemerides and orientations in function of time and for different system of coordinates. This library uses the results of precise numerical integration of planet orbits. In the second method approximate ephemerides are computed analytically by using classical keplerian theory with approximate orbital elements computed following the work of Simon et al [14, 16]. The planet orientation and prime meridian are taken from the Astronomical Almanach [17]. The user can select the method that should be used.

6.1 Command directory /PLANETOCOS/SPACECOORDINATE

6.1.1 /PLANETOCOS/SPACECOORDINATE/UseSpice

Format: /PLANETOCOS/SPACECOORDINATE/UseSpice <*aBool*>

Argument: Boolean *aBool*

Function: If *aBool* is true the Spice library is used to compute space coordinate transformation. If *aBool* is false the method based on classical keplerian theory is used.

7 Geometry

In PLANETOCOSMICS the geometry is either flat or spherical. The planet environment is modeled by a planet core that absorbs all particles, recovered by successive flat or spherical layers that represent the soil, the atmosphere and the space surrounding the planet. In the rest of this document the space surrounding the planet will be called the magnetosphere region. The soil, the atmosphere and the magnetosphere region are divided into sub-layers. In the case of the flat geometry the planet core is represented by a flat layer with a thickness defined by the user, while in the case of the spherical geometry it is is modeled by a sphere of radius equal to the mean radius of the planet plus the altitude of the bottom of the atmosphere minus the thickness of the soil, such that the top of the soil is at the altitude of the atmosphere bottom. Both atmosphere and soil can be neglected if needed.

7.1 Detection levels

The user defines the altitudes and/or atmospheric depths at which it will be possible to detect the flux of particles during the simulation. From this definition a list of detection altitude is established and arranged in order of decreasing altitude (highest altitude first, lowest altitude last). To establish this vector the user defined detection depths are converted to altitudes by using the depth vs altitude profiles provided by the atmospheric model (see 7.2) . If the atmosphere is not considered the detection levels specified by atmospheric depths will not be taken into account. Detection levels can lie in the magnetosphere region, at the top of the atmosphere, in the atmosphere or on the ground. The geometry is computed such that the detection levels corresponds to layer boundaries.

7.2 The atmosphere

The atmosphere is divided by homogeneous sub-layers in order to model the variation of the atmosphere density with the altitude. The density and composition of the layers are computed from the atmospheric model selected by the user such that: i) the atmospheric depths at the altitudes of layer boundaries are equivalent to the depths given by the atmospheric model at these altitudes; ii) the composition and the densities of the layers represent the mean composition and densities given by the atmospheric model in the layers. The thickness and the upper and lower altitudes of each atmospheric layer are computed by respecting the following criteria: i) the detection altitudes defined by the user correspond to layer boundaries; ii) the thickness of all layers are lower than a user defined upper limit and higher than a user defined lower limit; iii) when all the preceding criteria are respected a layer contains a fixed percentage of total atmospheric depth that is defined by the user .

The user can define the following parameters for the building of the atmosphere

- Maximum and minimum allowed thickness of a layer
- The altitude of the top of the atmosphere and of the ground
- The maximum percent of atmospheric depth contained in one atmospheric layer
- The model that defines the atmospheric composition and density in function of altitude.

Following the planet considered the user can select different type of atmospheric models.

7.2.1 Atmospheric TABLE model

For all planets the atmospheric TABLE model allows the user to define the atmospheric composition in an ASCII table. An example of such a table is given below :

```
\comments

Year : 1990      Day of year : 8                                     Latitude : 0.00

atomic number densities vs altitude

\definition
\type_of_composition{number_of_particles}
\mass_density_unit{g/cm3}
\number_density_unit{1/cm3}
\mass_density_unit{g/cm3}
\number_density_unit{1/cm3}
\altitude_unit{km}
\pressure_unit{hPa}

\data

altitude   pressure    Oxygen          N      Ar

120.000 0.1840986E-04 0.1509628E+12 0.5652739E+12 0.1160063E+10 119.900
0.1857324E-04 0.1528846E+12 0.5729153E+12 0.1180412E+10 119.800 0.1873881E-
04 0.1548424E+12 0.5806996E+12 0.1201213E+10 119.700 0.1890662E-04
0.1568369E+12 0.5886303E+12 0.1222478E+10 119.600 0.1907669E-04
0.1588686E+12 0.5967090E+12 0.1244215E+10 119.500 0.1924907E-04
0.1609382E+12 0.6049390E+12 0.1266436E+10 119.400 0.1942381E-04
0.1630466E+12 0.6133234E+12 0.1289155E+10 119.300 0.1960095E-04
0.1651944E+12 0.6218651E+12 0.1312381E+10 119.200 0.1978054E-04
0.1673827E+12 0.6305679E+12 0.1336131E+10 119.100 0.1996262E-04
0.1696119E+12 0.6394336E+12 0.1360412E+10 119.000 0.2014723E-04
0.1718829E+12 0.6484658E+12 0.1385239E+10
```

This table is divided by three successive sections that are started by the labels \comments, \definition and \data respectively. The first and second sections can be omitted. If both of these sections are omitted the label \data can be omitted for starting the last section. In all the tables empty lines are not considered. All the information given in the \comments section is purely informative and will not be considered by the code. In the \definition section the units used for defining the altitude, the number density, the mass density and the pressure are given by the parameter *unit_value* in the strings \altitude_unit{*unit_value*}, \number_density{*unit_value*}, \mass_density{*unit_value*}, and \pressure_unit{*unit_value*} respectively. The different units that can be used to define the altitude are *km* or *m*. The different units available for defining the number density are $1/cm^3$, $\#/cm^3$, $/cm^3$, $1/m^3$, $\#/m^3$ or $/m^3$. The different units available for defining the mass density are g/cm^3, mg/cm^3 and kg/m^3. The different units available for defining the pressure are *hPa*, *Pa*, *bar* and *atm.* When no units are specified the following default units are selected: *km* for altitude, *hPa* for pressure, g/cm^3 for mass density and $1/cm^3$ for number density.

The parameter *para_type* in \type_of_composition{*para_type*} specifies if the table defines the mass composition (*para_type ="mass_composition")* of the atmosphere or the particle density composition of the atmosphere (*para_type ="number_of_particles").* When the type of composition is not specified the particle density composition is considered.

In the \data section the atmospheric composition in function of altitude is provided in a column table. The first line of the table defines the information's contained in the different columns. The information given in the columns can be the altitude, pressure, density, temperature and concentration of the different elements that compose the atmosphere. The altitude column and at least one concentration column should be present while the pressure, mass density and temperature columns can be omitted. The temperature should be given in *K*. The concentration of the atmosphere in a specific element is given in a column named by the chemical symbol corresponding to this atom or molecule (O, O2, CO2, ..) or in the case of an atom by its name (Oxygen, Nitrogen, Argon,..). If the type of composition selected by the user is "*nb_of_particles*" the concentration is given in number of particles per unit volume. If the type of composition is "*mass_composition*" the concentration is given in weight per volume. The density column represents the mass density. If the density column is absent the total mass density is computed from the concentration columns. If the mass density is provided the concentrations are used to compute the percentage in mass or in number of particle (depending on the selected type of composition) of the different elements of the atmosphere. If the pressure is not provided it is computed in function of altitude from the mass density such that the pressure force is opposite to the gravity force. If the temperature is not provided it is computed from the perfect gas law $p=nkT$ where p, n, k and T represent the pressure, the number density, the Boatsman's constant and the temperature respectively.

7.2.2 Earth's atmospheric models

Two empirical atmospheric models specific to the Earth case are available in PLANETOCOSMICS: the MSISE90 model and its upgrade version NRLMSISE2000. Both models provide temperature, density and concentration profiles vs altitude from the ground to the exobase (~450-500 km) as function of geographic latitude, longitude, UT, $F_{10.7}$ index (10.7 cm solar radio flux used as solar UV proxy), $F_{10.7\,A}$ index (3month average of $F_{10.7}$) and the geomagnetic index Ap. The dependence of the model on $F_{10.7}$, $F_{10.7\,A}$, and Ap is negligible below 80 km. For a complete description of these models we refer to [18,19].

7.2.3 Martian atmospheric model

The Martian atmosphere is ~10-20 g/cm2 thick, which is very thin compared to the ~ 1030 g/cm2 of the Earth's atmosphere. It is mainly composed of 95.7 % of C02, 2.7 % of N2 and 1.6 % of Ar. Two models of the Martian atmosphere are available today for general use: the NASA/MSFC Mars Global Reference Atmospheric Model (Mars-Gram2001) and the ESA sponsored Mars Climate database [20, 23].

The Mars-GRAM 2001 model is an engineering Mars atmospheric model based on input data tables computed with the NASA Ames Mars General Circulation Model (MGCM) for altitude lower than 80 km and based on the University of Arizona Mars Thermospheric General Circulation Model (MTGCM) above 80 km [20-23]. It uses Mars topographic information from the Mars Orbiter Laser Altimeter (MOLA) instrument onboard of the Mars Global Surveyor. The program can compute the atmospheric pressure, density, and temperature vs altitude profiles in function of position and Martian season. Dust storm component can be taken into account if needed. Information on how to get the Mars-Gram model is described in the preface of the Mars-GRAM 2001 User guide that can be downloaded from the url http://trs.nis.nasa.gov/archive/00000549/ [20]. The Mars-Gram model can be obtained only on request and once obtained can not be further distributed to third parties. For this reason we could not incorporate it as an extra libraries in the PLANETOCOSMICS code. As an alternative solution we have developed the python script "marsgram_to_atmotable.py" that translates the output from the MarsGram2001 code into an atmospheric composition table that can be used in PLANETOCOSMICS as input of the TABLE atmospheric model (see section 6.2.1). This script is located in the directory "planetocosmics/MarsGramToPlanetocosmics". Before using the script the user should copy in this directory the binary code marsgram.out representing the executable of the MarsGram code. Information on how to compile this binary file is provided in the MarsGram user guide [20]. The user should also set the parameter DATADIR and GCMDIR in the file marsgram_template.nml, as explained in the section 3.2 of the MarsGram user guide. To produce from the MarsGram code a PLANETOCOSMICS atmospheric table corresponding to a given time and position on Mars, the user should type from the directory planetocosmics/mars/MarsGramToPlanetocosmics:

./marsgram_to_atmotable.py *Year Month Day Hour Minute Second Lat Long OutputFileName*

In this command the parameters *Year, Month, Day, Hour, Minute* and *Second* define the date, the parameters *Lat* and *Long* define the Martian latitude and longitude and the parameter *OutputFileName* defines the name of the file in which the resulting atmospheric table will be written. The atmospheric table obtained by this way represents a Mars daily averaged atmosphere at the selected position, with default scenario for the dust contribution. The output table *OutputFileName* is read in by using the UI command "/PLANETOCOS/GEOMETRY/ReadAtmosphereCompositionTable" (see section 7.4.15). Different tables obtained with the script marsgram_to_atmotable.py" are located in the directory "./planetocosmics/data/mars/marsgram_atmo_table".

The Mars Climate Database(MCD) sponsored by ESA/ESTEC is a climate database of atmospheric parameters compiled from simulation results of the General Circulation Model of the Martian atmosphere developed by Forget et al. [23-24]. The MCD contains simulated data (temperature, wind density, pressure, ...) stored on a 5x5 longitude-.latitude grid from the surface up to ~120 km. Five dust scenarios can be considered. The MCD is available freely on the WWW or on CD roms. A Geant4 model of the Mars radiation environment using a Mars atmosphere model based on the MCD is currently being developed under an ESA contract. To avoid any replication of work we have not implemented the use of MCD in PLANETOCOSMICS yet. However in the aim of completeness we plan to implement an interface between PLANETOCOMICS and MCD in the future.

7.2.4 Mercury atmospheric model

With an estimated particle density at the surface of Mercury of 10^2-10^4 cm^{-3}, the Hermenean atmosphere is very close to the vacuum and can be neglected. For this reason no atmospheric model specific to Mercury has been implemented in PLANETOCOSMICS.

7.3 Soil model

If needed the soil of the planet can be taken into account. The soil is divided into superposed homogeneous flat layers (flat geometry) or concentric spherical layers (spherical geometry) representing the first layers of the solid planet. The user defines the number of soil layers and the composition, density, and thickness of each layer. The composition of a layer is determined by setting the concentration of the different molecules and/or atoms contained in the layer. The molecules and atoms contained in the layer are specified by their chemical symbol. By default the soil is not considered.

7.4 User Interactive commands for defining the geometry

The different parameters defining the simulation geometry can be defined by using the UI commands contain in the directory /PLANETOCOS/GEOMETRY and /PLANETOCOS/SOIL.

7.4.1 /PLANETOCOS/GEOMETRY/SetAtmosphereTop

Format: /PLANETOCOS/GEOMETRY/SetAtmosphereTop <*altitude length_unit*>

Arguments: G4double *altitude*, G4String *length_unit/PLANETOCOS/GEOMETRY/ReadAtmosphereCompositionTable*

Function: Set the altitude of the top of the atmosphere.

Candidates: Candidates for *length_unit* are km and m.

7.4.2 /PLANETOCOS/GEOMETRY/SetGroundAltitude

Format: /PLANETOCOS/GEOMETRY/SetGroundAltitude <*altitude length_unit*>

Arguments: G4double *altitude*, G4String *unit_length*

Function: Set the altitude of the ground.

Candidates: Candidates for *length_unit* are km and m.

7.4.3 /PLANETOCOS/GEOMETRY/SetHeightOfWorldAboveAtmosphere

Format: /PLANETOCOS/GEOMETRY/SetHeightOfWorldAboveAtmosphere <*height length_unit*>

Arguments: G4double *height*, G4String *unit_length*

Function: Set the distance between the top boundary of the world volume and the top of the atmosphere.

Candidates: Candidates for *length_unit* are km, m, and R_p (mean radius of the planet).

7.4.4 /PLANETOCOS/GEOMETRY/SetPlanetCoreThickness

Format: /PLANETOCOS/GEOMETRY/SetPlanetCoreThickness < *thickness length_unit*>

Arguments: G4double *thickness*, G4String *unit_length*

Function: Set the thickness of the layer representing the planet core, in the case of a flat geometry.

Candidates: Candidates for *length_unit* are km and m.

7.4.5 /PLANETOCOS/GEOMETRY/verbose

Format: /PLANETOCOS/GEOMETRY/verbose *n*

Arguments: G4int *n*

Function: If *n*>0 the density, the top and bottom altitudes and depths of each layer forming the geometry are printed during the construction of the geometry.

7.4.6 /PLANETOCOS/GEOMETRY/SetMaxLayerThickness

Format: /PLANETOCOS/GEOMETRY/SetMaxLayerThickness <*lmax length_unit*>

Arguments: G4double *lmax*, G4String *length_unit*

Function: Set the maximum allowed thickness for an atmospheric layer.

Candidates: Candidates for *length_unit* are km and m.

7.4.7 /PLANETOCOS/GEOMETRY/SetMinLayerThickness

Format: /PLANETOCOS/GEOMETRY/SetMinLayerThickness <*lmin length_unit*>

Arguments: G4double lmin, G4String length_unit

Function: Set the minimum thickness of the atmospheric layers.

Candidates: Candidates for *length_unit* are km and m.

7.4.8 /PLANETOCOS/GEOMETRY/SetLayerLength

Format: /PLANETOCOS/GEOMETRY/SetLayerLength <*layer_length length_unit*>

Arguments: G4double *lmax*, G4String *length_unit*

Function: Set the length of all the layers in the case of a flat geometry

Candidates: Candidates for *length_unit* are km and m.

7.4.9 /PLANETOCOS/GEOMETRY/SetPercentOfDepth

Format: /PLANETOCOS/GEOMETRY/SetPercentOfDepth <*percent*>

Arguments: G4double *percent*

Function: Set the percentage of total depth contained in an atmospheric layer

7.4.10 /PLANETOCOS/GEOMETRY/SetType

Format: /PLANETOCOS/GEOMETRY/SetType <*type* >

Arguments: G4String *type*

Candidates: flat, spherical, FLAT, SPHERICAL

Function: Defines the type of the geometry.

7.4.11 /PLANETOCOS/GEOMETRY/RemoveAllDetectors

Format: /PLANETOCOS/GEOMETRY/RemoveAllDetectors

Arguments: none

Function: Reset the vector of altitudes and depths where flux of atmospheric shower particles can be detected.

7.4.12 /PLANETOCOS/GEOMETRY/DetectorAtAltitude

Format: /PLANETOCOS/GEOMETRY/DetectorAtAltitude <*altitude length_unit*>

Arguments: G4double *altitude*, G4String *length_unit*

Function: Add the given altitude in the vector of altitudes where flux of shower particles can be detected during the simulations.

Candidates: Candidates for *length_unit* are km and m.

7.4.13 /PLANETOCOS/GEOMETRY/DetectorAtDepth

Format: /PLANETOCOS/GEOMETRY/DetectorAtDepth <*depth depth_unit*>

Arguments: G4double *depth*, G4String *depth_unit (g/cm2,g/m2, kg/cm2, kg/m2)*

Function: Add the given depth in the vector of atmospheric depth where flux of atmospheric shower particles can be detected.

Remark: If no atmosphere is considered, no detection layer will be built at the user defined depth.

7.4.14 /PLANETOCOS/GEOMETRY/SetAtmosphereModel

Format: /PLANETOCOS/GEOMETRY/SetAtmosphereModel G4String model

Arguments: G4String model

Function: Select the model that defines the atmospheric composition and density in function of altitude.

Candidates : for all planets: TABLE (default)
for the Earth: MSISE90, NRLMSISE00

7.4.15 /PLANETOCOS/GEOMETRY/ReadAtmosphereCompositionTable

Format: /PLANETOCOS/GEOMETRY/ReadAtmosphereCompositionTable <*file_name*>

Arguments: G4String *file_name*

Function: Read in the given file the table that defines the composition of the atmosphere in function of altitude. To use this table as definition of the atmospheric composition the *TABLE* model should be selected with the *SetAtmosphereModel* command.

7.4.16 /PLANETOCOS/GEOMETRY/SetConsiderAtmosphere

Format: /PLANETOCOS/GEOMETRY/SetConsiderAtmosphere <*aBool*>

Arguments: Boolean *aBool*

Function: If *aBool* is true (false) the atmosphere is (not) taken into account.

7.4.17 /PLANETOCOS/GEOMETRY/SetReferenceDate

Format: /PLANETOCOS/GEOMETRY/SetReferenceDate
<*year month day* [*hour* [*min* [*sec*]]]>

Arguments: G4int *year, month, day, hour minute second*

Function: Select the reference date for the the MSISE90 and NRLMSISE00 Earth's atmospheric models.

Restriction: Only available for the Earth.

7.4.18 /PLANETOCOS/GEOMETRY/SetAp

Format: /PLANETOCOS/GEOMETRY/SetAp *Ap*

Arguments: G4double *Ap*

Function: Set the Ap geomagnetic index parameter for the MSISE90 and NRLMSISE00 atmospheric models.

Restriction: Only available for the Earth.

7.4.19 /PLANETOCOS/GEOMETRY/SetF107

Format: /PLANETOCOS/GEOMETRY/SetF107 *F107*

Arguments: G4double *F107*

Function: Set the *F107* solar flux parameter for the MSISE90 and NRLMSISE00 atmospheric models

Restriction: Only available for the Earth.

7.4.20 /PLANETOCOS/GEOMETRY/SetF107A

Format: /PLANETOCOS/GEOMETRY/SetF107A *F107A*

Arguments: G4double *F107A*

Function: Set the *F107A* solar flux parameter for the MSISE90 and NRLMSISE00 atmospheric models

Restriction: Only available for the Earth.

7.4.21 /PLANETOCOS/GEOMETRY/SetReferencePosition

Format: /PLANETOCOS/GEOMETRY/SetReferencePosition <*longitude latitude*>

Arguments: G4double *longitude, latitude*

Function: Set the geodetic longitude and latitude defining the position of reference for the MSISE90 and NRLMSISE00 atmospheric models.

Restriction: Only available for the Earth.

7.4.22 /PLANETOCOS/SOIL/AddMonoElementLayer

Format: /PLANETOCOS/GEOMETRY/AddMonoElementLayer
<*chemical_symbol density density_unit depth_or_thickness depth_or_length_unit*>

Arguments: double d*ensity, depth_or_thickness*
String *chemical_symbol, density_unit, depth_or_length_unit*

Function: Add a soil layer made of one element and specify its density, depth (or thickness) and the chemical symbol of the element. The layer is added at the bottom of the atmosphere , below the already defined soil layers. If *depth_or_length_unit* corresponds to a unit of length the parameter d*epth_or_thickness* defines the thickness of the layer. If *depth_or_length_unit* corresponds to a unit of depth the parameter d*epth_or_thickness* defines the depth of the layer.

Candidates: *density_unit :* g/cm3, mg/cm3, and kg/m3.
depth_or_length_unit: g/cm2, g/m2, kg/m2, kg/cm2, cm, dm, km, and m.

7.4.23 /PLANETOCOS/SOIL/AddLayer

Format: /PLANETOCOS/SOIL/AddLayer
<*nb_el density density_unit depth_or_thickness depth_or_thickness_unit*>

Arguments: double *density, depth_or_thickness*
integer *nb_el*
*S*tring *length_unit, depth_or_thickness_unit*

Function: Add a new soil layer made of *nb_el* elements and defines the density and thickness or depth of the layer. The layer is added at the bottom of the atmosphere, below the already defined soil layers. If *depth_or_length_unit* corresponds to a unit of length the parameter d*epth_or_thickness* defines the thickness of the layer. If *depth_or_length_unit* corresponds to a unit of depth the parameter *depth_or_thickness* defines the depth of the layer. After this command, each element of the layer should be defined by the command "/PLANETOCOS/*SOIL/AddElementToLayer"*.

Candidates: *density_unit :* g/cm3, mg/cm3, and kg/m3.
depth_or_length_unit: g/cm2, g/m2, kg/m2, kg/cm2, cm, dm, km, and m.

7.4.24 /PLANETOCOS/SOIL/AddElementToLayer

Format: /PLANETOCOS/SOIL/AddElementToLayer <*chemical_symbol concentration*>

Arguments: String *chemical_symbol*
double *concentration*

Function: Define the concentration and chemical symbol of an element in the layer that is being defined.

7.4.25 /PLANETOCOS/SOIL/ResetLayers

Format: /PLANETOCOS/ResetLayers

Function: Remove all the soil layers.

8 Electromagnetic and hadronic physics models

Different electromagnetic and hadronic physics models can be selected in PLANETOCOSMICS. For electromagnetic physics the Geant4 standard and the Geant4 low energy models are available. By default the standard model is considered. For description of these models we refer to the Geant4 physics reference manual [4].

For the hadronic interaction of hadrons with matter different hadronic physics lists are available in PLANETOCOSMICS. These lists are built in the same way than the G4 user case physics lists by using physics model constructor [5]. The different physics lists available in the code are described in the next table. Except for some energy range of validity of the different models contained in a list, these lists are the same than their corresponding user case physics list that can be found in [5] or in the geant4 source code under the path $G4INSTALLATION/physics_list/.

Hadronic physics list	***Description***
LHEP	This list uses the low energy LEP and high energy HEP parameterized models for elastic and inelastic scattering.

Hadronic physics list	*Description*
LHEP_BIC	For inelastic scattering of protons, neutrons, pions and kaons below 10 GeV it uses the binary intranuclear cascade models. Otherwise it is equivalent to the LHEP list.
LHEP_BIC_HP	For elastic and inelastic scattering of neutron with energy < 20 MeV it uses the Geant4 HPNeutron model based on ENDF database. Otherwise it is equivalent to LHEP_BIC.
LHEP_PRECO	For inelastic scattering of nucleons below 150 MeV it makes use of the Geant4 precompound model. Otherwise it is equivalent to LHEP.
LHEP_PRECO_HP	For elastic and inelastic scattering of neutron with energy < 20 MeV it uses the Geant4 HPNeutron model based on ENDF database. Otherwise it is equivalent to LHEP_PRECO.
QGSP	It uses theory driven models for the reactions of energetic pions, kaons, and nucleons with energy > 15GeV. It employs quark gluon string model for the 'punch-through' interactions of the projectile with a nucleus while a pre-equilibrium decay model with an extensive evaporation phase is used to model the behavior of the nucleus 'after the punch'. Otherwise it is equivalent to the LHEP physics list.
QGSP_BIC	For inelastic scattering of protons, neutrons, pions and kaons below 10 GeV it uses the binary intranuclear cascade models. Otherwise it is equivalent to the QGSP list.
QGSP_BIC_HP	For elastic and inelastic scattering of neutron with energy < 20 MeV it uses the Geant4 HPNeutron model based on ENDF database. Otherwise it is equivalent to the QGSP_BIC list.
QGSP_HP	For elastic and inelastic scattering of neutron with energy < 20 MeV it uses the Geant4 HPNeutron model based on ENDF database. Otherwise it is equivalent to the QGSP list.

If needed the electromagnetic hadronic interactions of high energy gamma, electrons and positrons with nuclei can be considered.

For the hadronic interaction of light ions with nucleus an extension of the binary intranuclear cascade (BIC) model is also available in PLANETOCOSMICS. The validity energy range of this model is ~ 0-5 GeV/nuc.

8.1 User interface commands /PLANETOCOS/PHYSICS

The commands described below allows to select the electromagnetic and hadronic physic models that will be used for the simulation.

8.1.1 /PLANETOCOS/PHYSICS/SelectTypeOfEMPhysics

Format: /PLANETOCOS/PHYSICS/SelectTypeOfEMPhysics <*EMmodel*>

Argument: String *EMmodel*

Function: Defines the model of electromagnetic physics that will be used for the simulation. Possible models are the standard electromagnetic model (*EMmodel = "STANDARD")* or the low energy electromagnetic model (*EMmodel="LOWENERGY")*. By default the standard model is taken into account. If *EMmodel* is set to NONE no electromagnetic physics is considered.

Default: *STANDARD*

8.1.2 /PLANETOCOS/PHYSICS/SelectTypeOfHadronicPhysics

Format: /PLANETOCOS/PHYSICS/SelectTypeOfHadronicPhysics <*HADROlist*>

Argument: String *HADROlist*

Function: Defines the hadronic physics list that will be used for the simulation. The parameter *HADROmodel* defines the selected type of hadronic list. Possible choices are *: NOHADRONIC, LHEP, LHEP_PRECO*, *LHEP_PRECO_HP*, *LHEP_HP*, *LHEP_BIC*, *LHEP_BIC_HP*, *QGSP, QGSP_HP, QGSP_BIC,* and *QGSP_BIC_HP*. By default the *QGSP_BIC_HP* model is taken into account. If is *HADROlist* is set to NOHADRONIC the hadronic interactions are not considered.

8.1.3 /PLANETOCOS/PHYSICS/SelectTypeOfIonHadronicPhysics

Format: /PLANETOCOS/PHYSICS/SelectTypeOfHadronicPhysics <*model*>

Argument: String model

Function: Defines the light ion hadronic physics model will be used for the simulation.

Candidates: *LE (low elastic), BIC (binary intranuclear cascade + LE) , NONE*

8.1.4 /PLANETOCOS/PHYSICS/ConsiderElectromagneticNuclearPhysics

Format: /PLANETOCOS/PHYSICS/ConsiderElectromagneticNuclearPhysics <*aBool*>

Argument: bool *aBool*

Function: The electromagnetic nuclear interaction of high energetic gamma, e-, and e+ with nuclei is taken into account if the parameter *aBool* is true and if an electromagnetic model is selected for the simulation. By default this physics is not considered.

9 Initialisation of the geometry and the physics

Directly after the selection of the parameters defining the geometry and the physics that will be used for the simulation, the user should initialise the code by using the UI command "/PLANETOCOSMICS/Initialise" described below. Other simulation parameters should be defined after this command.

9.1 /PLANETOCOS/Initialise

Format: /PLANETOCOS/Initialise

Function: Initialise the geometry and the physics list.

10 Magnetic field models

The effect of the planetary and magnetospheric magnetic field on the particle trajectories can be taken into account. When the field is switch on, it is considered for both the atmosphere and the magnetosphere region. For reason of computing time limitation the effect of the magnetic field is never considered in the soil. Following the planet and the type of geometry different magnetic field models can be considered. The magnetic field is the sum of two contributions: i)the internal field coming from source inside the planet (global or crustal field); ii) the external field that gives the magnetospheric contribution. The external field models are always specific to a given planet.

10.1 Computation of the magnetic field in the case of a flat geometry

Planetary magnetic field models are global models that provide the magnetic field components over the entire planet in function of PLA coordinates. In the case of the flat geometry the simulation is limited to a local region of the planet where the local coordinate system does not correspond to the PLA coordinate system. A 2D projection of this situation is illustrated in Figure1.

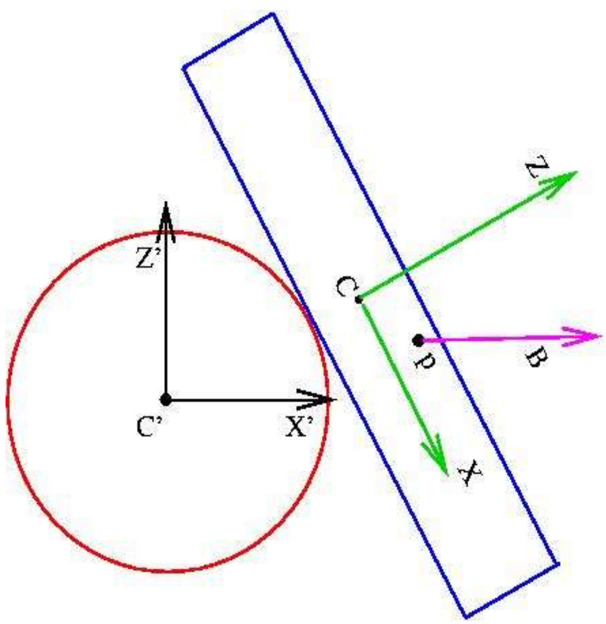

Figure 1**Illustration of the coordinate transformation needed when computing the magnetic field in the case of a flat geometry.**

The planet surface is represented in red while the blue box delimits the simulation region for the flat case. The position C and C' represents the center of the simulation box and the center of the planet, respectively. The simulation coordinate system is defined by the x, y(not shown) and z axes. The x', y', and z' axes define the planetocentric coordinate system. For computing the magnetic field in the flat geometry, from a model that gives the PLA component of the magnetic field at a given position expressed in term of PLA coordinates, the following has to be done:

- Compute the planetocentric coordinates of the position from its components given in the local coordinate system. This is obtained after a rotation and the translation of the position vector p. The translation is given by the the vector CC'.
- Compute the magnetic field in the PLA coordinate system
- Rotate back the magnetic field vector from the planetocentric coordinate system to the simulation coordinate system.

In PLANETOCOSMICS the user has to specify the latitude and longitude of the center of the simulation box (position C) in the planetary coordinate system of his choice. If these coordinates are given in the planetographic (PLAG) coordinate system the z axis of the local coordinate system coincides with the planetographic vertical. In the other cases it corresponds to the planetocentric vertical.

10.2 General internal field models

In this section we present the different internal models available for all the planets.

10.2.1 Spherical Harmonic GAUSS model

The internal planetary fields are generally modeled as the gradient of a scalar potential given by

$$V(r,\theta,\phi)=R_p\sum_{n=1}^{n_{max}}(R_p/r)^{(n+1)}\sum_{m=0}^{n}P_n^m(\cos\theta)(g_n^m\cos m\phi+h_n^m\sin m\phi) \quad (10.1)$$

where R_p is the planet radius, r, ϕ, and θ are the spherical planetocentric coordinates, P_n^m is the Schmidt normalized Legendre polynomial of degree n and order m and g_n^m , h_n^m are the Gauss spherical harmonic coefficients [25]. The Gauss coefficients are derived from observations. This model is implemented in PLANETOCOMICS for all planets and is referred as the GAUSS model. The user can define the Gauss spherical harmonic coefficients trough an ASCII table as the one presented below

$1 \quad 0 \quad g_1^0$
$1 \quad 1 \quad g_1^1 \quad h_1^1$
$2 \quad 0 \quad g_2^0 \qquad 2$
$1 \quad g_2^1 \quad h_2^1 \qquad 2 \quad 2 \quad g_2^2$
$h_2^2 \qquad ... \quad ... \quad ...$
$... \qquad n \quad m \quad h_n^m \quad g_n^m$

where the first two columns defines the degree and order of the coefficient g and h that are defined in the third and fourth column. For coefficients of order 0 the fourth column is empty as the h coefficients do not exist at zero order.

10.2.2 Dipole field

For a multitude of application a satisfactory approximation of the internal field is obtained by modeling the internal magnetic field as a magnetic dipole. This model is implemented in PLANETOCOSMICS and is referred as the DIPOLE model. The user can set directly the dipole magnetic moment B_0

(expressed as the strength of the field at the magnetic equator at the planet surface), the dipole axis and the position of the center of the dipole center in PLA coordinates, or compute it from the first and second degree Gauss spherical harmonic coefficients providing that they have been defined. The Gauss spherical harmonic model limited to the first degree corresponds to a magnetic dipole centered on the planet center, with the magnetic dipole moment given by

$$B_0=\sqrt{(g_1^0)^2+(g_1^1)^2+(h_1^1)^2} \quad (10.2)$$

and with the planetocentric spherical coordinates θ_{dip} and ϕ_{dip} of the planet dipole given by

$$\cos\theta_{dip}=g_1^0/B_0 \quad \text{and} \quad \tan\phi_{dip}=h_1^1/g_1^1 \quad (10.3)$$

A good approximation of the planetocentric Cartesian coordinate $(x_{off}, y_{off}, z_{off})$ of the center of the eccentric planet dipole has been derived in the literature as a function of the Gauss coefficients of degree 1 and 2. This expression has been implemented in the PLANETOCOSMICS. The interested reader will find it in reference [26].

10.2.3 HOMOGENEOUS Internal field

In PLANETOCOSMICS the user can select an homogeneous field as an internal field. The x, y and z components of the field can be defined separately or can be set equal to the components of the field given by a selected internal field model plus a selected external field model, at a selected position.

10.2.4 INTERPOL field

In this model, only valid for the flat geometry, the magnetic field at any position is obtained by a linear interpolation of pre-computed magnetic field vectors at the nodes of a Cartesian grid covering all the simulation box. The user can define the number of cells of the grid in the x, y and z directions. The magnetic field vectors at the grid nodes are either computed from the user defined magnetic field model or read from an ASCII file obtained from a previous calculation. This model has been developed to decrease drastically the computing time in the case of the Mars crustal magnetic field. The use of it is illustrated in the example .

10.3 Earth's geomagnetic field

Three different internal field models specific to the Earth are available: IGRF, IGRFTiltedDipole, IGRFEccentricTiltedDipole.

10.3.1 IGRF model

The IGRF model is a Gauss spherical harmonic model. It represents the most precise model of the geomagnetic field [25,27]. The Gauss coefficients are derived from magnetic field measurements from geomagnetic stations, ship-towed magnetometers and satellites. Every five years, the International Association of Geomagnetism and Aeronomy (IAGA) issues a new set of Gauss coefficients defining the new IGRF model [27]. The precedent IGRF model becomes then a Definitive Geomagnetic Reference Field (DGRF). In the IGRF model, the Gauss coefficients for a given period are obtained by interpolating and extrapolating the different DGRF/IGRF parameters released every five years. An

ASCII table defining the DGRCF/IGRF table from 1900 till now and obtained from the url http://www.ngdc.noaa.gov/IAGA/vmod/igrf.html , is distributed with the PLANETOCOSMICS code and is read when the Earth planet is selected. The user can define the date of reference for the IGRF model.

10.3.2 IGRFTiltedDipole

In this model the geomagnetic field is represented by a geomagnetic dipole centered on the center of the Earth, with the dipole momentum and axis deduced from the IGRF Gauss coefficients by using the equations 10.2 and 10.3.

10.3.3 IGRFEccentricTiltedDipole

In this model the geomagnetic field is represented by an Geomagnetic dipole shifted from the center of the Earth with the dipole momentum and axis deduced from the IGRF Gauss coefficients by using the equations 10.2 and 10.3. The center of the dipole is computed form the IGRF Gauss coefficients of degree 1 and 2 as given in [27].

10.4 Earth's magnetospheric field

The main sources for the Earth's external magnetospheric magnetic field are the ring current, the Chapman-Ferraro currents on the magnetopause, the tail current sheet, and the field aligned Birkeland current systems I and II [28]. WE have implemented three different models of the magnetospheric magnetic field : the Tsyganenko89c, Tsyganenko96, and Tsyganenko2001 models [29-35]. These models are available as FORTRAN code from the url http://nssdc.gsfc.nasa.gov/space/model/-magnetos/data-based/modeling.html . To use these models in PLANETOCOSMICS we have build a library that contains the binary files obtained by compiling the FORTRAN Tsyganenko codes. In the next paragraphs we provide a brief overview of the Tsyganenko models, for a more precise description we refer to the publications of Tsyganenko [30-35].

In all the Tsyganenko models the external magnetospheric field is influenced by the geomagnetic field that is considered as a geomagnetic dipole. The dipole tilt angle *PS* that represents the angle of the geomagnetic dipole axis with the GSM z-axis, is an important parameter of all the Tsyganenko models as it influences the shape of the magnetosheet . The magnetosheet is centered on the geomagnetic equatorial plane close to the Earth, and becomes slowly parallel to the GSM equatorial plane as it moves away form the Earth.

The Tsyganenko89 model provides seven different states of the magnetosphere corresponding to different levels of geomagnetic activity [30-31]. The *Iopt* parameter is an integer defining the different state of the magnetosphere. The correspondence between *Iopt* and the *Kp* index is given in table1. The Tsyganenko89 model has been derived from satellite measurements at distance from the Earth's lower than 70 R_E. Its domain of validity is therefore limited to this region of space.

Iopt	1	2	3	4	5	6	7
Kp	0,0+	1-,1,1+	2-,2,2+	3-,3,3+	4-,4,4+	5-,5,5+	>6-

Table 1 Correspondence between Iopt and Kp index for the Tsyganenko89 model

The Tsyganenko89 model does not provide a modeling of the continuous variation of the structure of the magnetosphere as a function of geomagnetic indices like *Dst* and of solar wind parameters. Such modeling is for example important when considering the evolution of the magnetosphere during a magnetic storm, or during the compression of the magnetosphere by an increase of the solar wind pressure. The Tsyganenko96 model introduced such a dependance [32-33]. In this model the external magnetospheric field is produced by different systems of modular currents with shape and strength depending on the dipole tilt angle *PS*, on the solar wind dynamic pressure P_{dyn},, on the *Dst* index, and on the y and z GSM components of the interplanetary magnetic field (IMF) B_y and B_z . The solar dynamic pressure is given by $P_{dyn}=nV^2$ where *n* and *V* represents the solar wind density and velocity, respectively. The contributions from the ring current, the field aligned currents and the magnetosheet currents are confined into a specific model of the magnetopause. The magnetopause is represented by a semi ellipsoid in the front, continued in the far tail ($x_{gsm} \leq -60R_E$) by a cylindrical surface. The axis of the magnetopause is parallel to the GSM equatorial plane. The size of the magnetopause decreases, when P_{dyn} increases. The strength of the ring current is a function of the *Dst* index with a correction depending on P_{dyn} to take into account the contribution of the magnetopause currents on *Dst.* The ring current is axisymetric and no partial ring current is considered. The amplitude of the magnetosheet currents depends on P_{dyn}, B_y, and B_z. Both the shape of the ring current and magnetosheet are dependent on the dipole tilt angle *PS*. The model considers also the interconnection of the IMF with the magnetospheric magnetic field. It produces a component of the field perpendicular to the magnetopause and therefore opens the magnetospheric configuration.

The Tsyganenko 2001 model is based on the same principles than the Tsyganenko96 model but brought important improvements compare to it [34-35]. The model is not only parameterised by *PS* , P_{dyn},, *Dst,* B_y and B_z., but also by two additional parameters, G_1 and G_2. These new parameters influenced the magnetotail components. The G_1 parameter was derived to quantify the energy transfer from the solar wind into the magnetosphere over the last hour. It is a function of the solar wind velocity *V* given in km/s, of the IMF transverse component $B_\perp = \sqrt{B_y^2 + B_z^2}$ given in nT, and of the IMF clock angle $\theta = \arctan(B_y/B_z)$:

$$G_1 = \sum_{i=1}^{12} V_i h(B_\perp) \sin^3 \frac{\theta_i}{2} \quad (4)$$

where the different i^th values represent 5 min average values that cover the last hour, and

$$h(B_\perp) = (B_\perp/4)^2 / (1 + B_\perp/4) \quad (5).$$

The parameter G_2 quantifies the strength of the sun-ward convection electric field over the last hour of observation, and controls the tailward/earthward shift of the magnetotail current system. It is defined by the function

$$G_2 = a\sum_{i=1}^{12} V_i B_s \quad (6)$$

where B_s represents the southward component of the IMF in nT ($B_s = |B_z|$ for $B_z < 0$ and $B_s = 0$ for $B_z >$ 0). The constant $a = 0.005$ was introduced to keep the parameter G_2 within the range $0 \le G_2 \le 10$, for commonly observed solar wind parameter values. Compare to the Tsyganenko96 model, the ring current in the Tsyganenko2001 model contains not only an axisymmetric component but also a contribution from the partial ring current that formed a closed current system with the field aligned currents. The Birkeland currents vary in response to the interplanetary conditions. The magnetopause is specified by an empirical model with the size varying with the solar wind pressure as in the Tsyganenko96 model but with a shape that is no more axisymmetric and depends on the Earth's dipole tilt angle. The user should be aware that the Tsyganenko 2001 model is a near Earth magnetosphere model. The dependence of the model on its parameters was derived from space measurements with x_{gsm}>-15 R_E . Therefore, at a position with xgsm<-15 R_E especially in the far tail, the output of the model should be considered with caution

10.5 Mars crustal magnetic field

The Magnetometer/Electron Reflectometer (MAG/ER) instrument onboard of the Mars Global Surveyor (MGS) has provided for the first time a global coverage of the magnetic field on Mars [36-37]. From these observations it has been confirmed that Mars lacks a global magnetic field at present. But the most significant result of the MAG/ER investigation is certainly the discovery of intensive magnetized regions of crustal origin. A summary of the results obtained with the MAG/ER experiment is presented in the Figure 2 that has been extracted from the paper of Conerney et al. [37]. The panels a, b, and c are contour colored maps representing the variation with position of the Martian magnetic field components B_r, B_θ, and B_ϕ at ~400 km altitude. The panel d represents a relief map of the surface of Mars with bluish colors for low altitudes and reddish colors for high latitudes. On this panel the contour of equal *Br* magnitude are reproduced in white and black colours for positive and negative values respectively. From these maps we can see that strong crustal magnetic fields are observed in the high lands while the northern low lands do not show any significant crustal magnetic field. In some regions, a magnetic field with a magnitude as high as 1600 nT was detected at 100 km. It represents a crustal field roughly two order of magnitude higher than the maximal crustal field observed on Earth. The fact that the high land regions are recovered by many craters while it is not the case for the lowland regions show that the low lands are younger lands that were probably resulting from a global geological process that took place after the meteoritic bombarding phase of the planet. Regions of strong crustal magnetic field are clearly observed in the high lands, while nearly no field is observed over the low lands. It is believed that the actual crustal magnetic field is an old signature of an ancient global magnetic field on Mars and that the dynamo process on Mars stopped before the geological process that led to the formation of the low lands and removed the crustal field from it.

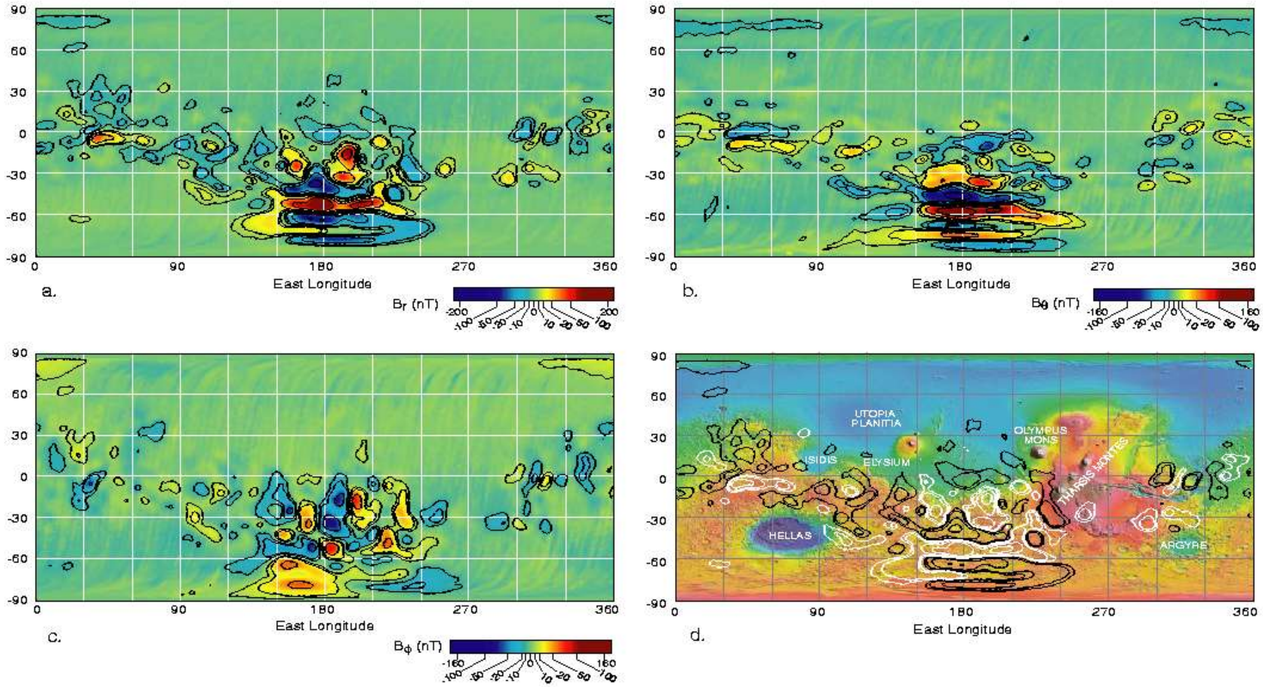

Figure 2 **Maps of the magnetic field of Mars at ~400 km as observed by the MAG/ER experiment onboard of the Mars Global Surveyor. For more details see text. This Figure has been taken from Conerney et al., 2001 [37].**

From the data of the MAG/ER experiment different Mars crustal magnetic field models have been developed by Purucker et al., Cain et al, and Arkahni-Ahmed [38-40]. The first aim of these models was to provide a global view of the Mars crustal field as a complement of information for studying the geological processes that could have taken place on Mars. The altitude coverage of MGS is ~100-400 km, therefore the precision of the crustal field models based on these data is limited below 100 km. For a more precise model of the crustal field on the Mars surface measurements at lower altitude would be needed. In PLANETOCOMICS we have implemented the models of Purucker et al, and Cain et al. [38-39]. The model of Arkhani Ahmed is not available and therefore could not been implemented [40].

Due to the lack of a global field, Mars is surrounded by a very faint magnetosphere resulting from the interaction of the solar wind with the planet ionosphere and in some regions with the crustal magnetic field [41]. Although the Mars Global surveyor magnetic has allowed to study some characteristic of the Mars magnetosphere, no models of this field has been released yet.

10.5.1 PURUCKER model

Purucker et al. have developed a Mars crustal magnetic field model in order to provide a magnetic map of Mars at ~200 km [38]. In this model the magnetic field is represented by 11500 vertical oriented dipole, distributed all over the surface of Mars with an average spacing of 111km. The dipole positions and magnitudes have been determined such that the model reproduces the best possible the radial magnetic field components observed by MAG/ER over an altitude range of 80-200 km, during the aerobraking phase and the science phasing orbit of MGS . A dipole contributes to the total magnetic field at a given position only if the distance between this position and its center is shorter than 1500 km.

The dipole magnitudes and positions as well as some FORTRAN programs to compute the field at 200 km can be obtained from the url http://denali.gsfc.nasa.gov/research/purucker/mars_mag.html .

We have implemented this model in PLANETOCOSMICS. However we should mention that this model is of limited use in PLANETOCOSMICS as it should not be used to compute the crustal magnetic field very close to the Martian surface where the magnetic dipole centers are located. Indeed the dipole approximation of a magnetic source is no more valid very close to the source where the magnetic field should be roughly equal to the magnetization of the source and not increase as $1/r^3$.

10.5.2 CAIN50 and CAIN90 models

Cain et al. have developed two spherical harmonic models of degree 50 and 90 (see section 3.11). The coefficients of both models can be obtained from the url http://geomag.gfdi.fsu.edu/mars/index.html. These models are implemented in PLANETOCOSMICS under the name CAIN50 and CAIN90 respectively [39]. The Gauss coefficients of the CAIN90 model have been deduced to best fit a combination of low altitude and high altitude data of MAG/ER onboard MGS. The low altitude data were taken during the aerobraking phase (altitude >~107 km) and the science phase orbit (altitude >~170 km) , while the high altitude data where taken during the mapping phase orbit (altitude ~400 km). This model represents probably the most precise model of the Mars crustal field today. The Gauss coefficient for the CAIN50 model have been deduced from data taken only during the mapping phase orbit. The CAIN50 model is less precise than the CAIN90 model at low altitude but is less time consuming by about a factor 4.

10.6 Mercury's magnetic field

The Mariner 10 mission is the only one that has allowed to measure the magnetic field in the near space of the planet Mercury [42-44]. The first magnetic field measurements were taken during the first Mercury flyby by Mariner 10 in March 1974 when the spacecraft approaches the planet to a minimum altitude of ~ 705 km. An additional set of magnetic field observations were taken one year later during the third Mercury Flyby of Mariner 10 when the spacecraft approach the planet to a minimum altitude of ~327 km. Although both data-set represent only 40 minutes of observations in a rather limited area of the Mercurian near space, it could be shown that Mercury contains a global internal field roughly 7.10^{-4} as strong as the Earth's dipole as well as a miniature Earth' like magnetosphere. Based on this short data set a multitude of different analytical models of the Mercurian magnetic field have been developed where both internal and external contribution are taken into account[43]. All these models differ significantly from each other and are limited in precision due to the inevitable limitations of the Mariner data-set. The estimates of the dipole magnitude in these models can vary from 100 to 300 $nT\text{-}Rm^{-3}$ following that external and non dipole internal source are considered or not. Therefore we did not find worthy to implement any complex models of the Mercury magnetosphere before future magnetic field observations of Mercury. By default the magnetic field of Mercury is represented in PLANETOCOSMICS by a magnetic dipole centered on the center of Mercury, with a 300 $nT\text{-}Rm^{-3}$ magnetic moment and with an axis parallel to the planet rotation axis.

10.7 Magnetopause model

When no magnetospheric magnetic field model is selected the magnetopause is modeled by a sphere with a radius that can be specified by the user. By default the radius is fix to 25 planet radius. In the case of the Earth different magnetopause models are associated to the different Tsyganenko magnetospheric magnetic field models. When one of the magnetospheric model is selected its respective magnetopause model is selected. If a user wants to select another magnetopause model he should do it after the selection of the magnetospheric model. It is also possible to remove the magnetopause.

10.8 Time dependence of the magnetic field models

Some magnetic field model as the Earth's IGRF and Tsyganenko models are time dependent. The time at which the magnetic field is calculated is defined by a time t after a given start date of reference $date_{ref}$. The user can fix both t and $date_{ref}$. Every time a new $date_{ref}$ parameter or a new time t are selected the time dependent parameters of the different magnetic field models available are computed. In the case of the Earth the geomagnetic dipole momentum B_0, and axis are recomputed according to the new IGRF parameters by using equations 2 and 3, while the shift of the geomagnetic dipole is set to (0,0,0) .

For the Tsyganenko96 and 2001 models it is possible to define the dependence of $Pdyn$, Dst, B_y, B_z, G_1 and G_2 on time trough an ASCII file. The format of this file is as follow:

	Year	Month	Day	hour	minute	second	
	nlines						
time	Pdyn	Dst	By	Bz	G1	G2	
	....	....	......	.....	.....	....	.....

The first line defines the date of reference $date_{ref}$. The second line defines the number of times at which the parameters are given in the rest of the file. Each of the following lines define the values of $Pdyn$, Dst, B_y, B_z, G_1 and G_2 at different times given in second after $date_{ref,}$. The solar wind pressure $Pdyn$ is given in nanopascals (nPa), the Dst index, B_y, and B_z components of IMF are given in nT wile the parameters G_1 and G_2 are dimensionless. The time should be given chronologically. The values of the Tsyganenko parameters and their corresponding times are stored into vectors. When the user select a given time t for computing the magnetic field, the $Pdyn$, Dst, B_y, B_z, G_1 and G_2 parameters values are interpolated/extrapolated from these vectors. If the vectors are empty the values of the Tsyganenko parameters are static values that can be defined by the *SetPdyn, SetDst, SetBy, SetBz, SetG1 and SetG2* commands (See below).

10.9 User interactive commands for selecting the magnetic field model

By using UI commands of the directory /PLANETOCOS/BFIELD/ the user select the internal and external magnetic field model.

10.9.1 /PLANETOCOS/BFIELD/SetTimeOfB

Format: /PLANETOCOS/BFIELD/SetTimeOfB *time unit*

Arguments: Double *time*
G4String *unit*.

Function: Defines the time after the reference date, at which the magnetic field is computed (see section 9.8).

Default: 0.

Errors: An error is flagged if the units are not time units defined in the unit table.

10.9.2 /PLANETOCOS/BFIELD/SetStartDate

Format: /PLANETOCOS/BFIELD/SetSartDate < *year month day [hour min sec]*>

Arguments: Int *year, month, day, hour, min* and *sec*

Function: Set the reference date for the magnetic field models in universal time (see section 9.8)

Default: 2000 1 1 0 0 0

Errors: An error is flagged if *month, day, hour, min,* and *sec* are not integral values in the ranges [1-12], [1-31], [0-23], [0-59], and [0-59] respectively.

10.9.3 /PLANETOCOS/BFIELD/SetNmaxForGAUSS

Format: /PLANETOCOS/BFIELD/SetNmaxForGauss <n_{max}>

Arguments: Integer n_{max}

Function: Set the maximum degree n_{max} for the GAUSS model (see Equation 1).

Default: For the IGRF model n_{max}=10 before 2000 and n_{max}=13 . When the Gauss coefficients are read in an ASCII table n_{max} is set by default to the highest degree of the Gauss coefficients.

10.9.4 /PLANETOCOS/BFIELD/SetInternalFieldModel

Format: /PLANETOCOS/BFIELD/SetInternalFieldModel <*Model*>

Arguments: G4String *Model*

Function: Set the internal field model .

Candidate: All planets: DIPOLE, GAUSS, FLATGRID (only for flat geometry).

Earth: IGRF, IGRFTiltedDipole, IGRFEccentricTiltedDipole

Mars: CAIN90, CAIN 50, PURUCKER

Default: Earth: IGRF at 2000/1/1

Mars: CAIN90

Mercury: Non tilted centered dipole with B_0= 300 nT

10.9.5 /PLANETOCOS/BFIELD/SetExternalFieldModel

Format: /PLANETOCOS/BFIELD/SetExternalFieldModel <*Model*>

Arguments: G4String *Model*

Function: Set the external magnetospheric magnetic field model .

Candidates: All planets: NOFIELD

Earth: TSY89, TSY96 and TSY2001.

Default: NOFIELD

Remark: For Mars and Mercury no magnetospheric magnetic field models are provided.

10.9.6 /PLANETOCOS/BFIELD/SetMagnetopauseModel

Format: /PLANETOCOS/SetMagnetopauseModel *model*>

Arguments: G4string *model*.

Function: Defines the magnetopause model.

Candidates: All planets: SPHERE, NOMAGNETOPAUSE

Earth: TSY89, TSY2001, TSY96

Default: Mercury and Mars: SPHERE

Earth: Depend on the selected magnetospheric model

Remark: Only available for the spherical geometry

10.9.7 /PLANETOCOS/BFIELD/SetDipoleB0

Format: /PLANETOCOS/SetDipoleB0 <B_0 *unit*>

Arguments: Double B_0
G4string *unit*.

Function: Set the planet dipole momentum B_0.

Default: Earth: By default B_0 is derived from the IGRF Gauss coefficients.

Mercury: 300 nT

Mars: 0

Errors: An error is flagged if *unit* is not one of the following units: nT, T, kG, G, nanotesla, tesla, kilogauss, gauss.

10.9.8 /PLANETOCOS/BFIELD/SetDipoleAxis

Format: /PLANETOCOS/BFIELD/SetDipoleAxis < θ Φ *angle_unit*>

Arguments: Double θ Φ
String *angle_unit*.

Function: Defines the direction of the planet dipole axis in planetocentric spherical coordinates.

Default: Earth: By default the direction of the dipole axis is computed from IGRF Gauss coefficients.

Mars and Mercury: (0,0)

Errors: An error is flagged if *angle_unit* is not one of the following units: degree, deg, rad or radian.

10.9.9 /PLANETOCOS/BFIELD/SetDipoleCenter

Format: /PLANETOCOS/BFIELD/SetDipoleCenter < C_x C_y C_z *length_unit* >

Arguments: Double C_x, C_y, C_z String *length_unit*

Function: Sets the center of the planet dipole in planetocentric coordinates.

Default: (0,0,0).

Errors: An error is flagged if *length_unit* is not one of the unit of length contained in the unit table.

10.9.10/PLANETOCOS/BFIELD/BfieldVsPositions

Format: /PLANETOCOS/BFIELD/BfieldVsPositions <*coorsys* alt_0 d_{alt} n_{alt} *length_unit* lat_0 d_{lat} n_{lat} $long_0$ d_{long} n_{long} *angle_unit output_file*>

Arguments: Double alt_0, d_{alt}, lat_0, d_{lat}, $long_0$, d_{long} String *coorsys*, *length_unit*, *angle_unit*, *output_file* Integer n_{alt}, n_{lat}, n_{long}

Function: Computes the magnetic field at the different nodes of a 3D grid. On this grid the node *(i,j,k)* is located at the altitudes alt_i, latitude lat_j and longitude $long_k$ where

$$
\begin{aligned}
alt_i &= alt_0 + i \cdot d_{alt} & i &= 0 \cdots n_{alt} - 1 \\
lat_j &= lat_0 + j \cdot d_{lat} & j &= 0 \cdots n_{lat} - 1 \\
long_k &= long_0 + k \cdot d_{long} & k &= 0 \cdots n_{long} - 1
\end{aligned}
$$

The parameter *coorsys* represents the coordinate system in which the nodes positions are defined and in which the magnetic field components are computed . The computed magnetic field X, Y and Z components are printed in the file *output_file.*

Remark: This command can only be used in the case of a spherical geometry.

10.9.11/PLANETOCOS/BFIELD/BfieldVsXYZ

Format: /PLANETOCOS/BFIELD/BfieldVsXYZ
< alt_0 d_{alt} n_{alt} x_0 d_x n_x y_0 d_y n_y *length_unit output_file*>

Arguments: Double alt_0, d_{alt}, x_0, d_x, y_0, d_y String *length_unit*,*output_file* Integer n_{alt}, n_x, n_y

Function: Computes the magnetic field at the different nodes of a 3D grid. On this grid the node *(i,j,k)* is located at the altitude alt_i, x coordinate x_j and y-coordinate y_k defined by

$$
\begin{aligned}
alt_i &= alt_0 + i \cdot d_{lat} & i &= 0 \cdots n_{alt} - 1 \\
x_j &= x_0 + j \cdot d_x & j &= 0 \cdots n_x - 1 \\
y_k &= y_0 + k \cdot d_y & k &= 0 \cdots n_y - 1
\end{aligned}
$$

The computed magnetic field X, Y and Z components are printed in the file *output_file*.

Remark: This command can only be used in the case of a flat geometry

10.9.12/PLANETOCOS/BFIELD/SetHomogeneousField

Format: /PLANETOCOS/BFIELD/SetHomogeneousField < B_x B_y B_z *unit* >

Arguments: Double B_x, B_y, B_z String *unit*

Function: Select an homogeneous magnetic field of Cartesian components defined by the vector (B_x, B_y, B_z).

10.9.13/PLANETOCOS/BFIELD/SetHomogeneousFieldFromSelectedModels

Format: /PLANETOCOS/BFIELD/SetHomogeneousFieldFromIGRF
<alt length_unit lat long coord_system int_model ext_model>

Argument: G4double *altitude, latitude, longitude*
G4String *angle_unit, length_unit, coord_system, int_model, ext_model*

Function: Fixes the homogeneous magnetic field to the total field (internal +external) at the position defined by the altitude *alt,* latitude *lat,* and longitude *long* in the coordinate system *coord_system.* The internal and external field models used to compute the homogeneous field are specified by the parameters *int_model* and *ext_model* respectively. If the parameter *ext_model* is not given, no external model is considered. In the case of the flat geometry it is also considered that the *lat* and *long* parameters defines the orientation of the vertical direction that correspond to the z direction of the local coordinate system . Therefore after being computed from the user selected models the homogeneous field are converted from planetocentric coordinate system to local coordinate system as explained in section 9.1

10.9.14/PLANETOCOS/BFIELD/SetWorldCenterPositionForFlatGeometry

Format: /PLANETOCOS/BFIELD/SetWorldCenterPositionForFlatGeometry
< lat long coord_system>

Argument: G4double *lat, long*
G4String *angle_unit, length_unit, coord_system*

Function: Set the latitude and longitude in degree of the center of the world box in the case of the flat geometry. This has to be specified to compute the orientation of the magnetic field in the local coordinate system (see section 9.1).

10.9.15/PLANETOCOS/BFIELD/SwitchOn

Format: /PLANETOCOS/BFIELD/SwitchOn

Function: Switch on the magnetic field.

10.9.16/PLANETOCOS/BFIELD/SwitchOff

Format: /PLANETOCOS/BFIELD/SwitchOff

Function: Switch off the magnetic field.

10.9.17/PLANETOCOS/BFIELD/ComputeInterpolMatrixForFlatGeometry

Format: /PLANETOCOS/BFIELD/ComputeInterpolMatrixForFlatGeometry < n_x n_y n_z *output_file* >

Arguments: Integer n_x, n_y, n_z String *output_file*

Function: Compute the magnetic field at different nodes of a Cartesian grid recovering all the simulation box from the ground to the top of the magnetosphere region. The parameters n_x n_y n_z specify the number of cells that the grid has in the X, Y and Z directions respectively. The computed magnetic field components represent the matrix of magnetic field component from which the magnetic field is interpolated in the case of the FLATGRID model. These magnetic field component are also registered in the file *output_file* such that for a later simulation they have just to be read rather than computed.

Remark: For time computing reason and memory reason it should be avoid to have n_x, n_y, n_z>100.

10.9.18/PLANETOCOS/BFIELD/ReadInterpolMatrixForFlatGeometry

Format: /PLANETOCOS/BFIELD/ReadInterpolMatrixForFlatGeometry < in*put_file* >

Arguments: Integer n_x, n_y, n_z String *output_file*

Function: Read from the file *input_file* the magnetic field component matrix from which the magnetic field is interpolated in the case of the FLATGRID model. This matrix can also be computed by using the command *ComputeInterpolMatrixForFlatGeometry*.

10.9.19/PLANETOCOS/BFIELD/SetIopt

Format: /PLANETOCOS/BFIELD /SetIopt <*Iopt*>

Arguments: Int Iopt.

Function: Sets the Iopt parameter for the Tsyganenko 89 model. The correspondence between Iopt and Kp is given in table 1.

Default: Iopt =1

Errors: An error is flagged if Iopt ≤0 or ≥ 8.

Remark: This command is available only if the planet Earth is considered.

10.9.20/PLANETOCOS/BFIELD/SetPdyn

Format: /PLANETOCOS/BFIELD <P_{dyn}>

Arguments: Double P_{dyn} .

Function: Set the solar wind dynamic pressure in nPa. This parameter is used by the Tsyganenko 96 and 2001 models.

Default: 2.

Error: An error is flagged if $P_{dyn} \leq 0$

Remark: This command is available only if the planet Earth is considered.

10.9.21/PLANETOCOS/BFIELD/SetDst

Format: /PLANETOCOS/BFIELD/SetDst <*Dst bfield_unit*>

Arguments: Double *Dst*
G4string *bfield_unit*.

Function: Set the *Dst* index used in the Tsyganenko96 and 2001 models.

Default: 0. nT

Errors: An error is flagged if *bfield_unit* is anything else than nanotesla, tesla, gauss, kilogauss, nT, T, G or kG.

Remark: This command is available only if the planet Earth is considered.

10.9.22/PLANETOCOS/BFIELD/SetImfBy

Format: /PLANETOCOS/BFIELD/SetImfBy <B_y *bfield_unit*>

Arguments: Double B_y
G4String *bfield_unit*.

Function: Sets the GSM y component of the

Errors: An error is flagged if *bfield_unit* is anything else than nanotesla, tesla, guauss, kilogauss, nT, T, G or kG.

Remark: This command is available only if the planet Earth is considered.

10.9.23/PLANETOCOS/BFIELD/SetImfBz

Format: /PLANETOCOS/BFIELD/SetImfBz <B_z *bfield_unit*>

Arguments: Double B_z
G4String *bfield_unit*.

Function: Sets the GSM z component of the interplanetary magnetic field. It represents an input parameter of the Tsyganenko96 and 2001 models.

Default: 1 nT.

Errors: An error is flagged if *bfield_unit* is anything else than nanotesla, tesla, guauss, kilogauss, nT, T, G or kG.

Remark: This command is available only if the planet Earth is considered.

10.9.24/PLANETOCOS/BFIELD/SetG1

Format: /PLANETOCOS/BFIELD/SetImfBy <G_1 >

Arguments: Double G_1

Function: Sets the G_1 parameter used in the Tsyganenko 2001 model.

Default: 1.

Remark: This command is available only if the planet Earth is considered.

10.9.25/PLANETOCOS/BFIELD/SetG2

Format: /PLANETOCOS/BFIELD/SetImfBy <G_2>

Arguments: Double G_2.

Function: Sets the G_2 parameter used in the Tsyganenko 2001 model.

Default: 0.

Remark: This command is available only if the planet Earth is considered.

10.9.26/PLANETOCOS/BFIELD/ReadTSYParametersVsTime

Format: /PLANETOCOS/BFIELD/ReadTSYParametersVsTime <*file_name*>

Arguments: G4String file_name

Function: Read an ASCII file that defines the time variation of the parameters P_{dyn} , Dst, B_y, B_z, G_1 and G_2 used in the Tsyganenko 96 and 2001 models. The format of the file is described in section 3.1.

Remark: This command is available only if the planet Earth is considered.

10.9.27/PLANETOCOS/BFIELD/PrintTSYParameters

Format: /PLANETOCOS/BFIELD/PrintTSYParameters

Arguments: None

Function: Prints the actual values of the parameters P_{dyn} , Dst, B_y, B_z, G_1 and G_2 used in the Tsyganenko 89, 96 and 2001 models.

Remark: This command is available only if the planet Earth is considered.

11 Primary source definition

For generating primary particle the PLANETOCOSMICS code makes use of the G4GeneralParticleSource class of the Geant4 toolkit. It allows the user to select different type of position distribution, angular distribution, and spectra for the primary source by using the /gps Geant4 built-in UI commands. For a full description of the /gps commands we refer to the G4GeneralParticleSource user manual[45].

Although the /gps offers the possibility to define many types of particle source, it does not cover all the user requirements of the PLANETOCOSMICS code, as for example the ability to define the position of the particle source in a Space coordinate systems or the ability to select of a galactic proton spectrum. For this reason we add to implement additional methods to extend the capability of the PLANETOCOSMICS code in term of generation of particle source.

11.1 Definition of primary particle position and direction in different Space Coordinate systems

By using the G4GeneralParticleSource the particle source is either represented by a point source or by an extended source centered on a user selected position. When the Spherical geometry is selected, it is possible in PLANETOCOSMICS to define this position either by giving its Cartesian coordinates or its altitude, latitude and longitude, in the different Space coordinate systems available for the planet considered. In the case of the Flat geometry the central position of the source can be specified by its altitude and xy-coordinates.

For a monodirectional source the direction of incidence can be specified by Cartesian coordinates or by azimuth and zenith angles in the different coordinate systems available for the planet considered. When the direction is given by the azimuth and zenith angles, the zenith=0 direction defines the vertical downward direction at the central position of the source. In the case of the Spherical geometry this vertical direction is function of the user selected space coordinate system. For a planetocentric system (PLA, PSO,..) it represents the direction passing trough the source position and the center of the planet while in the case of the PLAG system it represents the direction perpendicular to the planet surface of reference and passing trough the position of the source.

11.2 Galactic cosmic ray models and user defined spectra

The galactic cosmic ray flux in the interplanetary space at the top of the atmosphere is the results of the modulation of the interstellar galactic flux by diffusive processes in the heliosphere and the propagation of the modulated flux trough the planet magnetosphere. In PLANETOCOSMICS we have implemented a model for the modulated galactic cosmic ray flux of protons and alpha at Earth, based on the work of Gleeson and Axford and Garcia-Munoz [46-47]. In this model the differential flux of galactic cosmic rays at 1AU j_{1AU} is considered as isotropic and is related to the galactic cosmic ray flux in the local interstellar medium j_{LIS} by the relation

$$\frac{j_{1AU}(E_{kin})}{E^2-m_0^2}=\frac{j_{LIS}(E_{kin}+|z|e\Phi)}{(E+|z|e\Phi)^2-m_0^2} \qquad (11.1)$$

where Φ, E, m_0 and E_{kin} represent the modulation parameter, the total energy, the rest energy and the kinetic energy of the particle respectively. The modulation parameter Φ is function of the solar activity and is given in MV unit. The quantity $|z|e\ \Phi$ represents the energy loss of charged particles during their journey through the heliosphere before reaching 1AU. The range of Φ extends from 400 MV at solar minimum to 1000 MV at maximum with a solar cycle mean value of 550 MV.

The local interstellar omnidirectional flux of proton is given by

$$j_{LIS}(E_{kin})=1.244\ 10^6\ (E_{kin}+780.\exp(-2.05\ 10^{-4}\ E_{kin}))^{-2.65}\,cm^{-2}\ s^{-1}\ MeV^{-1} \quad (11.2)$$

while the local interstellar flux of alpha particle is given by

$$j_{LIS}(E_{kin})=2.262\ 10^5\ (E_{knuc}+660.\exp(-1.4\ 10^{-4}\ E_{knuc}))^{-2.77}\,cm^{-2}\ s^{-1}\ nuc\ MeV^{-1} \quad (11.3)$$

where E_{knuc} represents the kinetic energy per nucleon. Although this galactic flux presented here refers to the galactic cosmic ray flux at the Earth, this model can be also used at different planets by considering another modulation parameter Φ.

We do not have implemented in the code any models of solar energetic particles as it is considerably varying from solar eruptive event to solar eruptive event, and from planet to planet . However the code allows the user to define its solar or galactic cosmic ray spectrum through an ASCII file. An example of such file is given below

```
\definition
\energy_unit{GeV}
\flux_unit{#/m2/s/sr/GeV}
\particle{proton}
\interpolation{Log}
\data
```

```
0.199752268195 1775.91797534
0.232583452412 1825.90066655
0.27140000425 1903.52468575
0.316007403992 1903.52468575
```

This file is divided by three successive sections that are started by the labels \comments, \definition and \data respectively. The first and second sections started by \comments and \definition can be omitted. If both of these sections are omitted the label \data can be omitted for starting the last section. In all the tables white lines are not considered. All the information given in the \comments section is purely informative and will not be considered by the code.

In the \definition section the parameter *name* in the string \particle{*name*} defines the type of particle for which the incident flux is given in the table. The parameter *unit_value* in the string \energy_unit {*unit_value*} represents the units used to define the energy and the flux. The different units that can be used to define the energy are : *eV*, *keV*, *keV/nuc, MeV*, *MeV/nuc, GeV, GeV/nuc, TeV, TeV/nuc, PeV* and *PeV/nuc*. The parameter *unit_value* in the string \flux_unit{*unit_value*} represents the unit used to define the flux. The flux given is considered as an omnidirectional differential flux if the flux unit is contained in the following list : *1/m2/s/MeV, #/m2/s/MeV, 1/cm2/s/MeV, #/cm2/s/MeV, 1/m2/s/GeV, #/m2/s/GeV, 1/cm2/s/GeV* and *#/cm2/s/GeV.* The flux is considered as an directional differential flux if the flux unit is one of the following : *1/m2/s/sr/MeV, #/m2/s/sr/MeV, 1/cm2/s/sr/MeV, #/cm2/s/sr/MeV, 1/m2/s/sr/GeV, #/m2/s/sr/GeV, 1/cm2/s/sr/GeV* and *#/cm2/s/sr/GeV.* The flux is considered as an directional integral flux if the flux unit is one of the following : *1/m2/s/sr, #/m2/s/sr, 1/cm2/s/sr, #/cm2/s/sr, 1/m2/s/sr, #/m2/s/sr, 1/cm2/s/sr* and *#/cm2/s/sr.* The flux is considered as an omnidirectional integral flux if the flux unit is one of the following : *1/m2/s, #/m2/s, 1/cm2/s, #/cm2/s, 1/m2/s, #/m2/s, 1/cm2/s* and *#/cm2/s.* The primary flux is considered as isotropic such that a directional flux is considered as equivalent to an omnidirectional flux divided by a factor 4π.

In the section \data the flux is given in function of increasing energy in a 2 or 3 column table. A differential flux is defined by two columns where the first one gives the energy and the second one defines the flux itself. For an integral flux the integrated flux is given per energy bin in a three columns table. The first and second columns represent the lower and upper limit of the energy bins with the condition that the upper limit of a bin corresponds to the lower limit of the next energy bin. The third column represents the flux integrated over each energy bin. The energy and flux information read from the table is sent to the G4GeneralParticleSource class that is responsible for the generation of the flux in the primary generator action of the PLANETOCOSMICS code. This class makes use of different schemes for the interpolation of the flux at any given energy [45]. The interpolation scheme that should be used can be defined in the section \definition by the parameter *type* in the string \interpolation{*type*}. Possible interpolation schemes are: log, lin, exp and spline. By default a spline interpolation scheme is considered.

When one of the primary flux models described above is selected, a one-point source of primaries is considered and located at the top of the atmosphere, and the angular distribution is defined as a cosine law . This selection models the penetration of an isotropic galactic flux at the top of the atmosphere.

The user can als0 define a cutoff rigidity that is either constant or function of the direction of incidence. The cutoff rigidities in function of direction of incidence are read in a file corresponding to the output of the PLANETOCOSMICS command /PLANETOCOS/MAGNETIC/CutoffVsDirection (see 14.2.9).

11.3 User interactive commands for selecting the particle source

For defining the particle source the user can use the /gps commands built in the Geant4 toolkit and the commands contained in the directory /PLANETOCOSMICS/SOURCE.

11.3.1 /PLANETOCOS/SOURCE/SetPositionVector

Format: /PLANETOCOS/SOURCE/SetPositionVector <*coorsys X Y Z length_unit*>

Arguments: G4String *coorsys*, *unit*
Double *X, Y,Z* .

Function: Set the Cartesian coordinates of the central position of the source in the coordinate system defined by *coorsys*.

Remark: Valid for the spherical geometry.

11.3.2 /PLANETOCOS/SOURCE/SetPosition

Format: /PLANETOCOS/SOURCE/SetPosition
<*coorsys altitude length_unit latitude longitude angle_unit*>

Arguments: G4String *coorsys*, *length_unit*, *angle_unit*
G4double *altitude, latitude, longitude*

Function: Define the central position of the source by its altitude, latitude and longitude in the space coordinate system defined by *coorsys*.

Remark: Valid for the spherical geometry.

11.3.3 /PLANETOCOS/SOURCE/SetDirectionVector

Format: /PLANETOCOS/SOURCE/SetDirectionVector <*coorsys X Y Z* >

Arguments: G4String *coorsys*
G4double *X, Y,Z* .

Function: Define the start direction of a particle in the system of coordinate defined by *coorsys*. The vector (*X*,*Y*,*Z*) does not need to be normalised.

Remark: Valid for the spherical geometry.

11.3.4 /PLANETOCOS/SOURCE/SetDirection

Format: /PLANETOCOS/SOURCE/SetDirection
<coorsys zen azim angle_unit>

Arguments: G4String *coorsys*, *angle_unit*
G4double *zen*, *azim*

Function: The start or incoming (for backward tracking) direction of a particle is set antiparallel to a direction of observation defined by the zenith angle *zen*, and the azimuth angle *azim*, given in the coordinate system defined by *coorsys.* As the vertical direction (zenith = 0.) depends on the position, this command should be invoked after the definition of the position.

Remark: Valid for the spherical geometry.

11.3.5 /PLANETOCOS/SOURCE/SetPositionForFlatGeometry

Format: /PLANETOCOS/SOURCE/SetPositionForFlatGeometry *<X Y Alt length_unit>*

Arguments: G4String length*_unit*
Double *X, Y, Alt* .

Function: Set the Cartesian coordinate of the central position of the particle source in the case of the flat geometry. The parameter *Alt* defines the altitude of this position.

Remark: Valid for the flat geometry.

11.3.6 /PLANETOCOS/SOURCE/SetDirectionVectorForFlatGeometry

Format: /PLANETOCOS/SOURCE/SetDirectionVector *<X Y Z >*

Arguments: G4double *X, Y,Z* .

Function: Define the start direction of a primary particle. The vector (*X*,*Y*,*Z*) does not need to be normalised.

Remark: Valid for the flat geometry.

11.3.7 /PLANETOCOS/SOURCE/SetDirectionForFlatGeometry

Format: /PLANETOCOS/SOURCE/SetDirectionForFlatGeometry
<zen azim angle_unit>

Arguments: G4String *angle_unit*
G4double *zen*, *azim*

Function: The start or incoming (for backward tracking) direction of a particle is set antiparallel to a direction of observation defined by the zenith angle *zen*, and the azimuth angle *azim.* The Z-axis represents the vertical direction (zenith = 0).

Remark: Valid for the flat geometry.

11.3.8 /PLANETOCOS/SOURCE/SelectSolMinGalacticFlux

Format: /PLANETOCOS/SOURCE/SelectSolMinGalacticFlux
<particle_name Emin Emax [unit]>

Argument: G4String *particle_name, unit*
G4double *Emin, Emax*

Function: The flux of primaries is set to a galactic flux of protons or alpha at Earth at minimum solar activity according to the model described in section 2.5 with a modulation parameter of 400 MV. The type of primary defined by the parameter *particle_name* is either proton or alpha. The minimum and maximum energies of the spectrum are defined by *Emin* and *Emax* respectively. The unit parameter specified in which unit *Emin* and *Emax* are defined. Possible units are *keV/nuc*, *MeV/nuc*, *GeV/nuc*, *TeV/nuc* or *PeV/nuc*. If omitted unit is set to *GeV/nuc*. The angular distribution is set to a cosinus law and the source is fixed to a point source at the top of the atmosphere.

11.3.9 /PLANETOCOS/SOURCE/SelectSolMaxGalacticFlux

Format: /PLANETOCOS/SOURCE/SelectSolMaxGalacticFlux
<particle_name E_{min} E_{max} [unit]>

Argument: G4String *particle_name*, *unit*
G4double E_{min}, E_{max}

Function: The flux of primaries is set to a galactic flux of protons or alpha at Earth at maximum solar activity according to the model described in section 2.3 with a modulation parameter of 1000 MV. The type of primary defined by the parameter *particle_name* is either proton or alpha. The minimum and maximum energies of the spectrum are defined by *Emin* and *Emax* respectively. The unit parameter specified in which unit *Emin* and *Emax* are defined. Possible units are *keV/nuc*, *MeV/nuc*, *GeV/nuc*, *TeV/nuc* or *PeV/nuc*. If omitted unit is set to *GeV/nuc*. The angular distribution is set to a cosinus law and the source is fixed to a point source at the top of the atmosphere.

11.3.10/PLANETOCOS/SOURCE/SelectMeanGalacticFlux

Format: /PLANETOCOS/SOURCE/SelectSolMaxGalacticFlux
<*particle_name* E_{min} E_{max} *[unit]*>

Argument: G4String *particle_name*, *unit*
G4double E_{min}, E_{max}

Function: The flux of primaries is set to a galactic flux of protons or alpha at Earth at mean solar activity. It corresponds to the model described in section 2.5 where the modulation parameter is set to 500 MV. The type of primary defined by the parameter *particle_name* is either proton or alpha. The minimum and maximum energies of the spectrum are defined by E_{min} and E_{max} respectively. The *unit* parameter specified in which unit E_{min} and E_{max} are defined. Possible units are keV/nuc, MeV/nuc, GeV/nuc, TeV/nuc or PeV/nuc. If omitted *unit* is set to GeV/nuc. The angular distribution is set to a cosinus law and the source is fixed to a point source at the top of the atmosphere.

11.3.11/PLANETOCOS/SOURCE/SelectModulatedGalacticFlux

Format: /PLANETOCOS/SOURCE/SelectModulatedGalacticFlux
<*particle_name mod* E_{min} E_{max} *[unit]*>

Argument: G4String *particle_name*, *unit*
G4double mod, E_{min}, E_{max}

Function: The flux of primaries is set to a galactic flux of protons or alpha considering the model defined in the section 2.3 where the modulation parameter is given by *mod* in MV. The type of primary defined by the parameter *particle_name* is either proton or alpha. The minimum and maximum energies of the spectrum are defined by E_{min} and E_{max} respectively. The *unit* parameter specified in which unit E_{min} and E_{max} are defined. Possible units are keV/nuc, MeV/nuc, GeV/nuc, TeV/nuc or PeV/nuc. If omitted *unit* is set to GeV/nuc. The angular distribution is set to a cosinus law and the source is fixed to a point source at the top of the atmosphere.

11.3.12/PLANETOCOS/SOURCE/ReadPrimaryFlux

Format: /PLANETOCOS/SOURCE/ReadPrimaryFlux <*file_name*>

Argument: String *file_name*

Function: Read the differential or integrated spectrum and the type of particle of the primary flux in the table contained in the file *file_name.* The format of this table is described in section 2.5.

11.3.13/PLANETOCOS/SOURCE/SetRigidity

Format: /PLANETOCOS/SOURCE/SetRigidity *r rigidity_unit*

Arguments: Double *r*
String *rigidity_unit*

Function: Set the rigidity of the primary particle.

Error: An error is flagged if *rigidity_unit* is anything else than GV, V, kV, MV gigavolt, volt, kilovolt, or megavolt.

11.3.14/PLANETOCOS/SOURCE/ConsiderCutoff

Format: /PLANETOCOS/SOURCE/ConsiderCutoff <*aBool*>

Argument: Boolean *aBool*

Function: If *aBool* is true (false) the rigidity cutoff is taken (not taken) into account when generating primaries.

11.3.15/PLANETOCOS/SOURCE/SetCutoffRigidityValue

Format: /PLANETOCOS/SetCutoffRigidityValue <*cutoff_value unit*>

Argument: G4double *cutoff_value*, G4String *unit*

Function: The cutoff rigidity is considered as constant for all direction, and is set to *cutoff_value*. The *unit* parameter should be MV, GV, kV, or V.

11.3.16/PLANETOCOS/SOURCE/ReadCutoffVsDirection

Format: /PLANETOCOS/SOURCE/ReadCutoffVsDirection <*file_name*>

Argument: G4String *file_name*

Function: Reads the file that defines the variation of the cutoff rigidity in function of direction of incidence. This file should be formatted like an output file of MAGNETOCOSMICS.

11.3.17/PLANETOCOS/SOURCE/verbose

Format: /PLANETOCOS/SOURCE/verbose *n*

Arguments: G4int *n*

Function: For testing purpose if *n*>0 the particle name, rigidity, energy, and position are printed.

11.3.18/PLANETOCOS/SOURCE/BfieldAtPrimaryPosition

Format: /PLANETOCOS/SOURCE/BfieldAtPrimaryPosition

Arguments: none

Function: Print the components of the magnetic field in different coordinate systems.

12 Cut in range

An important simulation parameter in Geant4 is the cut-in-range parameter. In order to limit the computing time during a G4 simulation secondary e-, e+ and gamma are tracked only if their range in the material where they are produced is higher than the cut-in-range. This cut-in-range parameter can be defined globally or by region. By using cut by regions we have implemented in the code the possibility for the user to select a cut-in-range that is constant in depth (but not in length) for all atmospheric layers. It has the effect that e-, e+ and gamma are not produced below an energy threshold that is constant for all atmospheric layers. The user can also select a cut in range for a given layer (atmosphere or soil) of the geometry by specifying its name. By using Geant4 built-in interactive commands, it is also possible to define a global cut in range. For more information concerning cut in range we refer to the sections 5.4 and 5.5 of the Geant4 User Guide for application developer [2].

12.1 User interface commands /PLANETOCOS/CUT

The commands described below allow to define a cut in range given in depth for all the atmospheric layers, or to define the cut in range for selected layers in the soil or the atmosphere.

12.1.1 /PLANETOCOS/CUT/SetCutInDepthUnitForAllAtmosphericLayers

Format: /PLANETOCOS/PHYSICS/SetCutInDepthUnitForAllAtmosphericLayers <*depth unit_depth*>

Argument: G4double *depth,* G4String *unit_depth*

Function: Defines a cut in range for all atmospheric layers that is constant in depth. An error message is printed if *depth_unit* is not among [g/cm2, kg/cm2, g/m2 , kg/m2].

12.1.2 /PLANETOCOS/CUT/SetCutInDepthUnitForAGivenVolume

Format: /PLANETOCOS/PHYSICS/SetCutInDepthForAllLayers <*volume_name depth depth_unit*>

Argument: G4double *depth,* G4String *depth_unit*

Function: Set the cut in range of to the layer defined by the name *volume_name*. The cut in range is given as a depth. An error message is printed if *depth_unit* is not among [g/cm2, kg/cm2, g/m2 , kg/m2].

12.1.3 /PLANETOCOS/PHYSICS/SetCutInLengthUnitForAGivenVolume

Format: /PLANETOCOS/PHYSICS/SetCutInRangeForAGivenVolume
<volume_name length unit_length>

Argument: G4double *length,* G4String *unit_length*

Function: Defines the cut in range of the layer defined by the name *volume_name*. The cut in range is given as a length.

13 Stopping conditions

During a run primary and secondary particles are tracked till one of several user selected stopping conditions have been fulfilled. These different stopping conditions are presented in the next table.

Stopping Condition	***Description***	***Geometry restriction***
Outside magnetopause	If this condition is selected by the user all tracked particles are stopped outside the magnetopause. Following the planet type different magnetopause models are available. All magnetopause models can be scaled by a user defined scaling factor. By default this factor is 1.	Only valid for spherical geometry.
Maximum number of turn around a planet	A particle is stopped to be tracked after it has accomplished a user defined number of turn around the planet rotation axis.	Only valid for spherical geometry.
Stop altitude for particle moving downward	A particle moving downward is stopped when reaching an altitude specified be the user. The user can specified is this condition should be also applied for primary particles.	Valid for both types of geometry.

Stopping Condition	*Description*	*Geometry restriction*
Stop altitude for particle moving upward	A particle moving upward is stopped when reaching a stopping altitude specified by the user. The user can specified if this condition should be applied also for primary particles.	Valid for both types of geometry.
Maximum trajectory length	A particle is stopped when its trajectory length is higher that a user defined maximum length.	Valid for both types of geometry.
Maximum trajectory duration	A particle is stopped when the trajectory duration is higher that user defined maximum duration.	Valid for both types of geometry.
Stopping energy	Particles are stopped when they reach a user defined minimum energy. The stopping energy is given in function of particle type.	Valid for both type of geometry.

13.1 User interactive command for controlling the stopping conditions

The /PLANETOCOS/STOPCONDITION commands allows the user to control the stopping conditions available in PLANETOCOSMICS.

13.1.1 /PLANETOCOS/STOPCONDITION/SetMaxTrackLength

Format : /PLANETOCOS/STOPCONDITION/SetMaxTrackLength *<MaxTrackLength length_unit>*

Argument: G4String *length_unit* G4double *MaxTrackLength*

Function: Defines the maximum allowed track length. When a particle track reaches this limit the particle is stopped.

13.1.2 /PLANETOCOS/STOPCONDITION/SetMaxTrackDuration

Format : /PLANETOCOS/STOPCONDITION/SetMaxTrackDuration *<MaxTrackDuration time_unit>*

Argument: G4String *time_unit*
G4double *MaxTrackDuration*

Function: Defines the maximum allowed track duration in global time (not run time).

13.1.3 /PLANETOCOS/STOPCONDITION/SetStoppingEnergy

Format: /PLANETOCOS/STOPCONDITION/SetStoppingEnergy <*particle_name Ekin Eunit*>

Argument: G4String *particle_name, Eunit*
Double *Ekin*

Function: Defines the kinetic energy below which the particle of type *particle_name* should be stopped to be tracked.

13.1.4 /PLANETOCOS/STOPCONDITION/DesactivateStopEnergyCondition

Format: /PLANETOCOS/STOPCONDITION/DesactivateStopEnergyCondition *particle_name*

Argument: G4String *particle_name*

Function: Desactivate the stopping energy condition for the selected particle.

13.1.5 /PLANETOCOS/STOPCONDITION/StopAtMagnetopause

Format: /PLANETOCOS/STOPCONDITION/StopAtMagnetopause <*aBool*>

Argument: Boolean *aBool*

Function: If true (false) all particles are (not) stopped outside the magnetopause.

Remark: Only valid in the case of the spherical geometry.

13.1.6 /PLANETOCOS/STOPCONDITION/SetMagnetopauseScaling

Format: /PLANETOCOS/STOPCONDITION/SetMagnetopauseScaling <*scaling_factor*>

Argument: Double *scaling_factor*

Function: Scale the magnetopause.

Remark: Only valid in the case of the spherical geometry.

13.1.7 /PLANETOCOS/STOPCONDITION/SetMaxNbOfTurnsAroundThePlanet

Format: /PLANETOCOS/STOPCONDITION/SetMaxNbOfTurnsAroundThePlanet
<max_nb_turn>

Argument: Double *max_nb_turn*

Function: Set the maximum number of turns that a particle can make around the rotation axis of the planet before it will be stopped.

Remark: Only valid in the case of the spherical geometry.

13.1.8 /PLANETOCOS/STOPCONDITION/SetStopAltitudeForUpwardFlux

Format: /PLANETOCOS/STOPCONDITION/SetStopAltitudeForUpwardFlux
<alt length_unit>

Argument: Double *alt*
String *length_unit*

Function: Set the altitude above which upward moving particles are stopped.

13.1.9 /PLANETOCOS/STOPCONDITION/SetStopAltitudeForDownwardFlux

Format: /PLANETOCOS/STOPCONDITION/SetStopAltitudeForDownwardFlux
<alt length_unit>

Argument: Double *alt* String
length_unit

Function: Set the altitude above which downward moving particles are stopped.

13.1.10/PLANETOCOS/STOPCONDITION/StopAlsoUpwardPrimary

Format: /PLANETOCOS/STOPCONDITION/StopAlsoUpwardPrimary
<aBool>

Argument: Boolean *aBool*

Function: If true (false) the primary particles are also (not) stopped when crossing upward the stop altitude defined by the command *SetStopAlsoUpwardPrimary* .

13.1.11/PLANETOCOS/STOPCONDITION/StopAlsoDownwardPrimary

Format: /PLANETOCOS/STOPCONDITION/StopAlsoDownwardPrimary <*aBool*>

Argument: Boolean *aBool*

Function: If true (false) the primary particles are also stopped when crossing downward the stop altitude defined by the command *SetStopAlsoDownwardPrimary* .

14 Type of applications

The PLANETOCOSMICS can be used either to compute and analysis the propagation of charged particles in the planet magnetic field or to compute and analysis the flux of particles resulting from the interaction of cosmic rays with the planet atmosphere and soil, with or without considering the presence of the magnetic field.

14.1 Simulation of hadronic and electromagnetic interactions

In the first simulation mode the hadronic and electromagnetic interactions of primary and secondary cosmic rays with the planet atmosphere and soil are computed. If needed the effect of the magnetic field on the particle trajectory can be taken into account. The user can select different models of hadronic and electromagnetic physics (see section 8). During this simulation mode the following information can be detected:

- The flux of any type of primary and secondary particles at user defined altitudes
- The energy deposited by atmospheric cosmic rays showers in the atmosphere
- The production of cosmogneic nuclides in the atmosphere
- The flux of quasi trapped particles.

This kind of simulations are started as a usual Geant4 run, by using the command /run/beamOn.

14.2 Study of the propagation of charged particles in the planet magnetic field

The second simulation mode can be used for studying the propagation of charged particles in the planet magnetic field. In this mode particles are tracked either backward or forward in time in the magnetic field without considering the hadronic and electromagnetic interactions with the atmosphere and the soil.

For the analysis of cosmic ray measurements and for space radiation environment studies its is important to quantify the so called cutoff rigidity that represents the lower rigidity limit above which cosmic rays can cross a planet magnetosphere and reach a specific position from a specific observational direction. It is also important to determine the asymptotic direction of a cosmic ray, that represents its direction before entering into the magnetosphere.

The method uses to compute the cutoff rigidity and the asymptotic direction of incidence in the case of the Earth is illustrated in Figure3, a complete description on asymptotic direction computation method and cosmic ray cutoff terminology is provided in [48] . The trajectories of cosmic rays with different rigidities, arriving at the same observing position and from the same direction of incidence are computed backward in time. The red curves labeled by 1, 2, 3 ,4 and 5 represent the trajectories of positive charged particles with 20, 15, 10, 5 and 4.5 GV rigidities respectively. In this case all the trajectories are initiated in the vertical direction from the same observing position. Particles at high rigidity (trajectory 1,2,3) have small trajectory bending before escaping the Earth's magnetosphere. The particle with 5 GV rigidity has a more important bending but can still escape the Earth's magnetosphere. The trajectory labeled by 5 is making several complex loops before reaching another point on the Earth’s surface illustrating that for this specific rigidity a cosmic ray can not reach the selected Earth’s position, from the vertical direction. Such trajectory is said forbidden while trajectories of particles escaping the Earth magnetosphere are called allowed trajectories. The direction at the last position of an allowed trajectory represents the asymptotic direction of incidence of the particle corresponding to this trajectory. The black arrow on Figure3 represents the asymptotic direction for the trajectory 3.

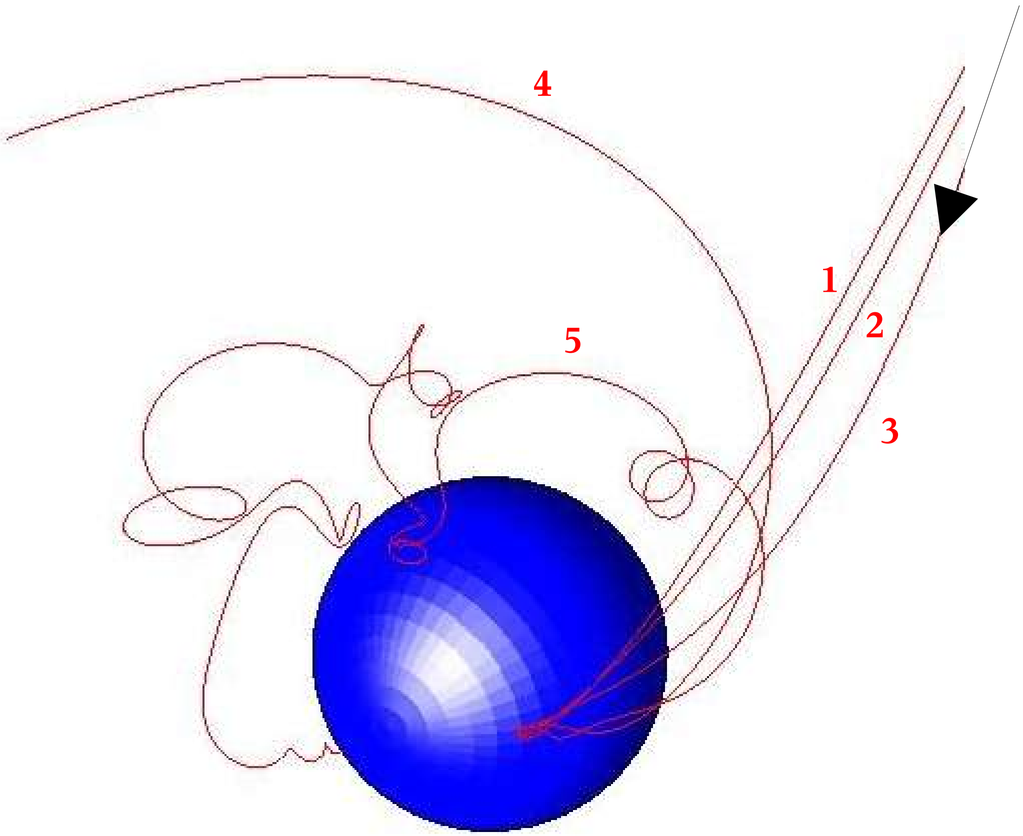

Figure 3 Illustration of the backward trajectory method used for determining if a cosmic ray with a given rigidity can reach a given position from a given direction of incidence (see text). The Earth is represented by the blue sphere. The red lines 1, 2, 3, 4 and 5, represent computed backward trajectories of positive ions with rigidity of 20, 15, 10, 5 and 4.5 GV respectively.

In general backward trajectories are computed for several rigidities spanning a large range of values with a constant rigidity interval ΔR (usually 0.01 GV for the Earth case). Results of such a computation are illustrated in Figure4. In this plot a filter value of 0 and 1 is associated to the rigidities corresponding to the forbidden and allowed trajectories respectively. Three distinct regions are observed. A high rigidity region where all trajectories are allowed, a low rigidity region where all trajectories are forbidden and an intermediate region called the penumbra where bands of allowed trajectories are separated by band of forbidden ones. The rigidity of the last allowed computed trajectory before the first forbidden one is called the upper cutoff rigidity R_U. The rigidity of the last allowed trajectory, below which all trajectories are forbidden is called the lower cut-off rigidity R_L. An effective cutoff rigidity R_C characterising the structure of the penumbra is defined by

$$R_C = R_U - n_{allowed} \Delta R \quad (12.9)$$

where $n_{allowed}$ represents the number of allowed trajectories encountered in the penumbra.

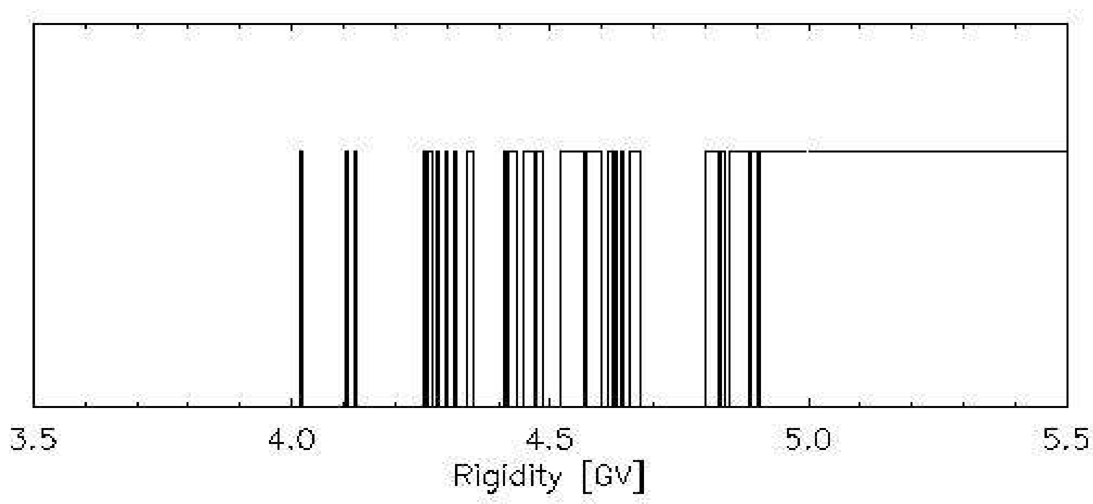

Figure 4Results of backward trajectories computation used for computing cutoff rigidities. See detail in text.

Although the notions of asymptotic directions and cutoff rigidity have been defined in the case of the Earth, it can be extended to any planet containing a magnetic field. In PLANETOCOSMICS it is possible to compute cutoff-rigidities and asymptotic directions for all planets but only in the case of the Spherical geometry.

When computing asymptotic directions and cutoff rigidity with PLANETOCOSMICS, backward trajectories are computed at different rigidities values contained in a decreasing vector that can be defined by using the commands “/PLANETOCOS/MAGNETIC/AddValuesToRigidityVector”. To determine correctly the cutoff rigidities R_U, R_L and R_C it is necessary that the rigidity vector covers at least all the penumbra region. Moreover over the penumbra the values in the rigidity vector should decrease by a constant interval Δ*R*. In standard rigidity cutoff computation program for the Earth's case Δ*R* is generally set to 0.01 *GV*.

The output of the computation of asymptotic direction of incidence is contained in an ASCII table structured as below:

Rigidity	Filter	Asympt. Lat.	Asympt. Long.	Position Xpla	Ypla	Zpla	
20.00	1	69.33	95.90	-0.53	8.81	24.42	19.99
	1	69.33	95.90	-0.53	8.81	24.42	19.98
	1	69.32	95.89	-0.53	8.81	24.42	19.97
	1	69.32	95.89	-0.53	8.82	24.42	
...	...	...	...	...	...	...	
0.12	1	-21.60	153.18	-8.09	24.02	0.69	
0.11	1	-22.27	163.95	-9.37	23.19	-0.57	
0.10	1	-18.22	-177.20	-12.60	21.62	-2.42	
0.09	1	6.89	-135.51	-21.13	13.15	-3.17	
0.08	-1	-33.09	-121.42	-11.85	-6.06	-4.89	
0.07	-1	10.88	-15.62	-8.91	-5.70	-6.46	
0.06	-1	-15.26	-39.38	-14.54	-0.57	-3.36	
0.05	-1	81.43	47.38	-5.08	2.21	-4.17	

0.04	-1	79.62	-38.08	-8.89	7.04	-2.77
0.03	-1	-20.47	-72.00	-5.23	9.00	-3.31
0.02	-1	-28.69	-74.79	-2.59	9.97	-3.22
0.01	0	62.48	-99.77	-0.10	0.48	-0.87
Rl 0.09	Rc 0.09	Ru 0.09				

The first column represents the rigidity of the particle in GV. In the second column a filter value is set to -1, 0 or 1 if the backward trajectory has reached the user defined maximal trajectory length, has reach the minimum allowed altitude given in PLAG or has reached the magnetopause, respectively. In the third and fourth columns the asymptotic direction is given by the asymptotic latitude and longitudes. It defines the direction of the trajectory when it crosses the magnetopause. For trajectories that do not cross the magnetopause it represents their direction at the last position of the trajectory. The PLA coordinates of the last position of the backward trajectories are given in the last columns. The R_U, R_L and R_C cut-off rigidities deduced from the computation results are written on the last line.

14.3 Command directory /PLANETOCOS/MAGNETIC/

Commands contained in this directory allows to control the second simulation mode of PLANETOCOSMICS (see section 14.2).

14.3.1 /PLANETOCOS/MAGNETIC/AddValuesToRigidityVector

Format: /PLANETOCOS/MAGNETIC/AddValuesToRigidityVector <rig_0 d_{rig} n_{rig}>

Arguments: G4double rig_0, d_{rig}
G4int n_{rig}

Function: Add the vector [rig_0,..., $rig_0+i\ .d_{rig}$, ..., $rig_0 + (n-1)\ .d_{rig}$] to the rigidity vector used for computing asymptotic direction of incidence and .

Error: As the vector should be decreasing, an error is flagged if $d_{rig} \geq 0$, and if rig_0 is not smaller than the last defined value of the rigidity vector.

14.3.2 /PLANETOCOS/MAGNETIC/SetDefaultRigidityVector

Format: /PLANETOCOS/MAGNETIC/SetDefaultRigidityVector

Arguments: none

Function: Set the rigidity vector to the default vector.

14.3.3 /PLANETOCOS/MAGNETIC/ResetRigidityVector

Format: /PLANETOCOS/MAGNETIC/ResetRigidityVector

Arguments: none

Function : Reset the rigidity vector

14.3.4 /PLANETOCOS/MAGNETIC/SetStopAltitude

Format: /PLANETOCOS/MAGNETIC/SetStopAltitude <*alt length_unit*>

Arguments: Double *alt*
String *length_unit*

Function : Defines the PLAG altitude below which particle should be stopped

14.3.5 /PLANETOCOS/MAGNETIC/ComputeTrajectories

Format: /PLANETOCOS/MAGNETIC/ComputeTrajectories <*nb_particles*>

Argument: Integer *nb_particles*

Function: Compute the trajectory of *nb_particles* particles in the planet magnetic field without considering the hadronic and electromagnetic interaction of the particles with the planet atmosphere and soil. The particles are emanating from the particle source defined by user trough the /gps and /PLANETOCOSMICS/SOURCE commands.

14.3.6 /PLANETOCOS/MAGNETIC/ComputeReverseTrajectories

Format: /PLANETOCOS/MAGNETIC/ComputeReverseTrajectories <*nb_particles*>

Argument: Integer *nb_particles*

Function: Compute the trajectory of *nb_particles* particles backward in time in the planet magnetic field without considering the hadronic and electromagnetic interaction of the particles with the planet atmosphere and soil. The particles are emanating from the particle source defined by user trough the /gps and /PLANETOCOSMICS/SOURCE commands.

14.3.7 /PLANETOCOS/MAGNETIC/ComputeAsymptoticDirections

Format: /PLANETOCOS/MAGNETIC/ComputeAsymptoticDirections <*file_name*>

Arguments: G4String *file_name*

Function: Compute the asymptotic directions, rigidity filter values and cutoff rigidities for the user selected position, and direction of incidence and for a selected type of particle. The different rigidities at which particle trajectory are computed are defined in a rigidity vector defined by using the UI commands *AddValuesToRigidityVector, ResetRigidityVector* and *SetDefaultRigidityVector*. The position and direction of incidence are defined by using UI commands from the /gps and "/PLANETOCOS/SOURCE" directories. The asymptotic directions and rigidity filter values in function of rigidities as well as cutoff rigidities are stored in the ASCII file *file_name*.

14.3.8 /PLANETOCOS/MAGNETIC/RCutoffVsPosition

Format: /PLANETOCOS/MAGNETIC/RCutoffVsPosition <*coorsys altitude length_unit*
lat_0 d_{lat} n_{lat} $long_0$ d_{long} n_{long}
zen azim angle_unit
output_file>

Arguments: G4String *coorsys*, *length_unit*, *angle_unit*, *output_file*.
G4double *altitude*, lat_0, d_{lat}, $long_0$, d_{long} , *zen*, *azim* G4int n_{lat}, n_{long}

Function: Computes the cutoff rigidities in function of latitude and longitude, at a given observing altitude, and for a given observing direction of incidence defined by the zenith and azimuth angle. The string coorsys defines the space coordinate system in which position and direction are expressed. The following latitudes lat_i and longitudes $long_i$ are considered:

$$\begin{aligned} lat_i &= lat_0 + i \cdot d_{lat} \qquad & i &= 0 \cdots n_{lat} - 1 \\ long_j &= long_0 + j \cdot d_{long} \qquad & j &= 0 \cdots n_{long} - 1 \end{aligned}$$

The results are printed in the file *output_file.*

14.3.9 /PLANETOCOS/MAGNETIC/RCutoffVsDirection

Format: /PLANETOCOS/MAGNETIC/RCutoffVsDirection
<*coorsys* zen_0 d_{zen} n_{zen} $azim_0$ d_{azim} n_{azim} *output_file*>

Arguments: G4String *coorsys, output_file*.
G4double zen_0, d_{zen}, $azim_0$, d_{dazim}
G4int n_{zen}, n_{azim}

Function: Compute the cutoff rigidities for different directions of incidence, at the same observing position. The string *coorsys* defines the space coordinate system in which directions are expressed. The position of observation should have been defined previously by UI commands from /gps or /PLANETOCOS/SOURCE commands. The directions of incidence are defined by their zenith angle zen_i and azimuth angle $azim_j$ where

$$zen_i = zen_0 + i \cdot d_{zen} \qquad i = 0 \cdots n_{zen} - 1$$
$$azim_j = azim_0 + j \cdot d_{azim} \qquad j = 0 \cdots n_{azim} - 1$$

The results are saved in the ASCII file *output_file*.

14.3.10/PLANETOCOS/MAGNETIC/RCutoffVsTime

Format: /PLANETOCOS/MAGNETIC/RCutoffVsTime < t_0 d_t n_t *time_unit output_file*>

Arguments: G4String *time_unit output_file*.
G4double t_0, d_t
G4int n_t

Function: Compute the cutoff rigidities for the same direction and position of observation but at different time t_i where

$$t_i = t_0 + i \cdot d_t \quad i = 0 \cdots n_t - 1$$

The observing position and direction of incidence should have been defined previously by UI commands from the /PLANETOCOS/SOURCE directory. The results are saved in the ASCII file output_file.

15 Analysis and histogramming

During all simulations, except for the computation of cutoff rigidity and asymptotic directions of incidence, it is possible to compute: i) the flux of primary particles, ii) the flux of any kind of particles at the user defined altitudes and or depth; iii) the energy deposited by the cosmic ray showers in the atmosphere in function of altitude and or depth; iv) the production rate of cosmogenic nuclides over the entire atmosphere; v) different information on quasi trapped particles. All these results are registered in form of 1D and 2D histograms.

We have developed two different versions of PLANETOCOSMICS that differs in the type of package that have been used for histogramming. In the AIDA version of PLANETOCOSMICS the analysis part of the code has been developed in compliance with the AIDA3.0.0 interface [7]. In the ROOT version the analysis part is using the ROOT package for storage of the results[8]. The way to install both versions is explained in the section 3 . In both version the histograms containing the simulation results are organised into an histogram tree in the same way than a file system. In this tree an histogram is associated to a path that is represented by a directory name + the histogram label that should be an integer. Histograms can be created and added to the tree interactively. In the following we call the histogram tree contained in the computer memory during a simulation as the permanent tree to distinguish it from histogram tree saved into files. The permanent tree is divided into five directories /PRIMARY, /FLUX, /EDEP /COSMONUC and /QUASITRAPPED corresponding to the different kind of information that can be detected during a run.

15.1 Primary flux detection

For testing if the primary source is defined as expected the following information on primary particle flux can be registered:

- The energy, cos(θ) and ϕdistributions by using 1D histograms
- The cos(θ) vs energy distribution by using 2D histograms
- The cos(θ) vs ϕdistribution by using 2D histograms

In a primary flux histogram a primary particle is always counted with a weight *w*=1. The primary flux histograms are found in the histogram tree under the directory /PRIMARY. The different UI commands to create primary flux histograms are contained in the command directory "/PLANETOCOS/ANALYSIS/PRIMARY".

15.1.1 /PLANETOCOS/ANALYSIS/PRIMARY/CosZenVsEnergy

Format: /PLANETOCOS/ANALYSIS/CosZenVsEnergyHisto
<*particle_name label nE E_{min} E_{max} Eunit scale_type ncos cos1 cos2*>

Arguments: G4int *nE, ncos;* G4double *E_{min}, E_{max}, cos1, cos2*
G4String *particle_name, label, Eunit, type*

Function: Create a 2D histogram that register the flux of primaries for a given type of particle in function of particle kinetic energy and of cosine of the zenith angle of the particle direction. The particle type is defined by the *particle_name* parameter. If *particle_name* is set to “all” all type of primaries are registered. The histogram is identified by the *label* parameter in the histogram tree. The energy axis is divided into *nE* bins covering the energy range [E_{min}, E_{max}]. The parameter *scale_type* defines the type of scale used for the energy axis, that can be linear or logarithmic (*scale_type* = *lin* or *log)*. The cosine zenith axis is linear and divided by *ncos* bins covering the range [*cos1,cos2*].

Candidates: *Eunit* (MeV, keV, eV, GeV)

15.1.2 /PLANETOCOS/ANALYSIS/PRIMARY/CosZenVsAzimuthHisto

Format: /PLANETOCOS/ANALYSIS/CosZenVsAzimuthHisto
<*particle_name label nAz Az1 Az2 ncos cos1 cos2*>

Arguments: G4int *nAz, ncos*
G4double *Az1, Az2, cos1, cos2*
G4String *particle_name, label*

Function: Create a 2D histogram that register the flux of primaries for a given type of particle in function of the azimuth angle and of the cosine of the zenith angle of the particle direction. The particle type is defined by the *particle_name* parameter. If *particle_name* is set to “all” all type of primaries are registered. The histogram is identified by the *label* parameter in the histogram tree. The azimuth axis is linear and divided into *nAz* bins covering the range [Az1*, Az2*] . The different UI commands to create primary flux histograms are contained in the command directory /PLANETOCOS/ANALYSIS/PRIMARY. The cosine zenith axis is linear and divided by *ncos* bins covering the range [*cos1,cos2*].

15.1.3 /PLANETOCOS/ANALYSIS/PRIMARY/FluxHisto

Format: /PLANETOCOS/ANALYSIS/FluxHisto
<*particle_name label E* E_{min} E_{max} *Eunit scale_type unit*>

Arguments: G4int *nE;* G4double E_{min}, E_{max}
G4String *particle_name, label, Eunit, scale_type*

Function: Create a 1D histogram that register the flux of primaries for a given type of particle in function of kinetic energy. The particle type is defined by the *particle_name* parameter. If *particle_name* is set to “all” all type of primaries are registered. The histogram is identified by the *label* parameter in the histogram tree. The energy axis is divided into *nE* bins covering the energy range [E_{min}, E_{max}]. The parameter *scale_type* defines the type of scale used for the energy axis, that can be linear or logarithmic.

Candidates: *scale_type* (*lin, linear, Linear, Lin, LINEAR, LIN, log, Log, LOG)* *Eunit (*MeV, eV, keV, GeV)

15.1.4 /PLANETOCOS/ANALYSIS/PRIMARY/CosZenithHisto

Format: /PLANETOCOS/ANALYSIS/CosZenithHisto
<*particle_name label nCos cos1 cos2*>

Arguments: G4double *cos1, cos2*
G4int *nCos*
G4String *particle_name, label*

Function: Create a 1D histogram that register the flux of primaries for a given type of particle in function of the cosine of the zenith angle of the particle direction. The particle type is defined by the *particle_name* parameter. If *particle_name* is set to “all” all type of primaries are registered. The histogram is identified by the *label* parameter in the histogram tree. The cosine zenith axis is linear and divided by *ncos* bins covering the range *[cos1,cos2*].

15.1.5 /PLANETOCOS/ANALYSIS/PRIMARY/AzimuthHisto

Format: /PLANETOCOS/ANALYSIS/PRIMARY/AzimuthHisto
<*particle_name label nAz az1 az2*>

Arguments: G4double *az1, az2*
G4int *nAz*
G4String *particle_name, label*

Function: Create a 1D histogram that registers the flux of primaries for a given type of particle in function of the azimuth angle of the particle direction. The particle type is defined by the *particle_name* argument. If *particle_name* is set to “all” all type of primaries are registered. The histogram is identified by the *label* parameter in the histogram tree. The azimuth axis is linear and divided into *nAz* bins covering the range [a*z1, az2*] .

15.2 Secondary flux detection

It is possible to detect the flux of particles at user defined altitude and or atmospheric depth. During the definition of the geometry the user should define at which altitudes and/or atmospheric depth the flux of particles can be detected. From this definition a list of detecting altitudes and atmospheric depths is established and arranged in order of decreasing altitude (highest altitude first, lowest altitude last). By using UI commands the user can define which kind of flux information of which type of particle and at which detection altitude should be detected. The following information on secondary particle flux can be registered:

- The energy distribution of downward and upward flux by using 1D histograms.
- The cos(θ), and ϕdistribution by using 1D histograms
- The cos(θ) vs energy distribution by using 2D histograms
- The upward and downward flux of particle in function either of latitude and longitude in the case of spherical geometry or in function of x and y position in the case of flat geometry, by using 2D Histograms.

In the case of spherical geometry, for the detection of particle flux in function of energy and/or direction (first three cases in the list above) the user can specify the latitude and longitude region covered by the next created histograms.

For the detection of particle flux in function of position the flux represents an integrated flux over an energy range specified by the user.

The 1D and 2D histograms corresponding to the fluxes of a specific type of particle at the n^{th} detection altitude are registered in the permanent tree in the directory /FLUX/DET*n*/*particle_name* where *n* represents the number of the detection boundary and *particle_name* defined the type of the particle considered. In a secondary flux histogram a particle is registered with either a weight *w*=1 or *w*=1/cos *(*θ*)*. The level of a bin in an flux histogram with *w*=1 is proportional to the flux of particles that cross a flat horizontal area, within the same energy and solid angle ranges that those covered by the bin, while in the case of *w=1/cos(*θ*)* the bin level is proportional to the flux of the particles that cross a small sphere that has a cross section area equal to the flat horizontal area.

The different UI commands to create secondary flux histograms are contained in the command directory /PLANETOCOS/ANALYSIS/SECONDARY.

15.2.1 /PLANETOCOS/ANALYSIS/SECONDARY/SelectDetector

Format: /PLANETOCOS/ANALYSIS/SECONDARY/SelectDetector <*nb*>

Arguments: G4int *nb*

Function: Add the nb^{th} detection altitude to the list of altitudes to which the next created flux histograms will be associated.

15.2.2 /PLANETOCOS/ANALYSIS/SECONDARY/UnselectDetector

Format: /PLANETOCOS/ANALYSIS/SECONDARY/UnselectDetector <*nb*>

Arguments: G4int *nb*

Function: Remove the nb^{th} detection altitude from the list of altitudes to which the next created flux histograms will be associated

15.2.3 /PLANETOCOS/ANALYSIS/SECONDARY/SelectAllDetectors

Format: /PLANETOCOS/ANALYSIS/SECONDARY/SelectAllDetectors

Arguments: none

Function: By using this command the next created flux histograms will be associated to all detection altitudes.

15.2.4 /PLANETOCOS/ANALYSIS/SECONDARY/UnselectAllDetectors

Format: /PLANETOCOS/ANALYSIS/SECONDARY/UnselectAllDetectors

Arguments: none

Function: Clear the list of selected detection altitudes.

15.2.5 /PLANETOCOS/ANALYSIS/SECONDARY/DownwardFluxHisto

Format: /PLANETOCOS/ANA

Analysis and histogrammingLYSIS/SECONDARY/DownwardFluxHisto
< *particle_name label nE E_{min} E_{max} Eunit scale_type*>

Arguments: G4int *nE*; G4double *E_{min}, E_{max};*
G4String *particle_name, Eunit, scale_type, label*

Function: Create 1D histograms that register the downward flux of a given type of particle at the user selected detection altitudes in function of particle kinetic energy. The particle type is defined by the *particle_name* parameter. The histogram is identified by the *label* parameter in the histogram tree. The energy axis is divided into *nE* bins covering the energy range [E_{min}, E_{max}]. The parameter *scale_type* defines the type of scale used for the energy axis, that can be linear or logarithmic (*scale_type* = [*lin, linear, Linear, Lin, LINEAR, LIN, log, Log, LOG]*

Candidates: *scale_type* [*lin, linear, Linear, Lin, LINEAR, LIN, log, Log, LOG*] *Eunit* [MeV, eV, keV, GeV]

15.2.6 /PLANETOCOS/ANALYSIS/SECONDARY/UpwardFluxHisto

Format: /PLANETOCOS/ANALYSIS/SECONDARY/UpwardFluxHisto
<*particle_name label E E_{min} E_{max} Eunit scale_type unit*>

Arguments: G4int *nE*; G4double *E_{min}, E_{max};*
G4String *particle_name, Eunit, scale_type, label*

Function: Create 1D histograms that register the upward flux of a given type of particle at the user selected detection altitudes in function of particle kinetic energy. The particle type is defined by the *particle_name* parameter. The histogram is identified by the *label* parameter in the histogram tree. The energy axis is divided into *nE* bins covering the energy range [E_{min}, E_{max}]. The parameter *scale_type* defines the type of scale used fo the energy axis, that can be linear or logarithmic (*scale_type* = [*lin, linear, Linear, Lin, LINEAR, LIN, log, Log, LOG]*

Candidates: *scale_type* (*lin, linear, Linear, Lin, LINEAR, LIN, log, Log, LOG)* *Eunit (*MeV, eV, keV, GeV)

15.2.7 /PLANETOCOS/ANALYSIS/SECONDARY/CosZenithVsEkinHisto

Format: /PLANETOCOS/ANALYSIS/SECONDARY/CosZenithVsEkinFluxHisto
<particle_name label nE E_{min} E_{max} *Eunit scale_type ncos cos1 cos2>*

Arguments: integer *nE*, *ncos*;
double E_{min}, E_{max}, *cos1, cos2*
String *particle_name, label, Eunit, type*

Function: Create 2D histograms that register the flux of a given type of particle at the user selected detection altitudes in function of kinetic energy and of the cosinus of the zenith angle of the particle direction. The particle type is defined by the *particle_name* parameter. The histogram is identified by the *label* parameter in the histogram tree. The energy axis is divided into *nE* bins covering the energy range [E_{min}, E_{max}] . The parameter *scale_type* defines the type of scale used fo the energy axis, that can be linear or logarithmic (*scale_type = lin* or *log)*. The *cos_zenith* axis is linear and divided by *ncos* bins covering the range [*cos1,cos2*].

Candidates: *Eunit* (MeV, keV, eV, GeV)

15.2.8 /PLANETOCOS/ANALYSIS/SECONDARY/CosZenithHisto

Format: /PLANETOCOS/ANALYSIS/SECONDARY/CosZenithHisto
<particle_name label nCos cos1 cos2>

Arguments: G4double *cos1, cos2*
G4int *nCos*
G4String *particle_name, label*

Function: Create 1D histograms that register the flux of a given type of particle at the selected detection altitudes in function of the cosinus of the zenith angle of the particle direction. The particle type is defined by the *particle_name* parameter. The histogram is identified by the *label* parameter in the histogram tree. The cosinus zenith axis is linear and divided into *ncos* bins covering the range [*cos1,cos2*].

15.2.9 /PLANETOCOS/ANALYSIS/SECONDARY/AzimuthHisto

Format: /PLANETOCOS/ANALYSIS/SECONDARY/AzimuthHisto
<particle_name label nAz az1 az2>

Arguments: G4double *az1, az2*
G4int *nAz*
G4String *particle_name, label*

Function: Create 1D histogram that register the flux of a given type of particle at the selected detection altitudes in function of the azimuth angle of the particle direction. The particle type is defined by the *particle_name* parameter. The histogram is identified by the *label* parameter in the histogram tree. The azimuth axis is linear and divided into *nAz* bins covering the range [*Az1, Az2*] .

15.2.10/PLANETOCOS/ANALYSIS/SECONDARY/DownwardIntegralFluxVsPosHisto

Format: /PLANETOCOS/ANALYSIS/SECONDARY/DownwardIntegralFluxVsPosHisto
< *particle_name label* n_{lon} lon_{min} lon_{max} n_{lat} lat_{min} lat_{max} >

Arguments: G4int n_{lon}, n_{lat}; G4double lon_{min}, lon_{max}, lat_{min}, lat_{max}
G4String *particle_name, label*

Function: Create a 2D histogram that register the downward integral flux of a given type of particle at the user selected detection altitudes in function of longitude ([-180,180]) and latitude for the case of the spherical geometry. The integral flux is integrated over the energy range defined by the commands. The particle type is defined by the *particle_name* parameter. The histogram is identified by the *label* parameter in the histogram tree. The longitude axis is divided into n_{lon} bins covering the range [lon_{min}, lon_{max}]. The longitude axis is divided into n_{lon} bins covering the range [lat_{min}, lat_{max}].

15.2.11/PLANETOCOS/ANALYSIS/SECONDARY/UpwardIntegralFluxVsPosHisto

Format: /PLANETOCOS/ANALYSIS/SECONDARY/UpwardIntegralFluxVsPosHisto
< *particle_name label* n_{lon} lon_{min} lon_{max} n_{lat} lat_{min} lat_{max} >

Arguments: G4int n_{lon}, n_{lat}; G4double lon_{min}, lon_{max}, lat_{min}, lat_{max}
G4String *particle_name, label*

Function: Create a 2D histogram that register the upward integral flux of a given type of particle at the user selected detection altitudes in function of longitude ([-180,180]) and latitude for the case of the spherical geometry. The flux represents is integrated over the energy range defined by the commands. The particle type is defined by the *particle_name* parameter. The histogram is identified by the *label* parameter in the histogram tree. The longitude axis is divided into n_{lon} bins covering the range [lon_{min}, lon_{max}]. The longitude axis is divided into n_{lon} bins covering the range [lat_{min}, lat_{max}].

15.2.12/PLANETOCOS/ANALYSIS/SECONDARY/DownwardIntegralFluxVsFlatPosHisto

Format: /PLANETOCOS/ANALYSIS/SECONDARY/DownwardIntegralFluxVsFlatPosHisto
< *particle_name label* n_x n_y >

Arguments: G4int n_x, n_x;
G4String *particle_name, label*

Function: Create a 2D histogram that register the downward integral flux of a given type of particle at the user selected detection altitudes in function of X and Y position for the case of the flat geometry. The flux is integrated over the energy range defined by the commands. The particle type is defined by the *particle_name* parameter. The histogram is identified by the *label* parameter in the histogram tree. The X and Y axis are divided into n_x and n_y bins and cover all the X and X extension of the simulation box.

15.2.13/PLANETOCOS/ANALYSIS/SECONDARY/UpwardIntegralkFluxVsFlatPosHisto

Format: /PLANETOCOS/ANALYSIS/SECONDARY/UpwardIntegralFluxVsFlatPosHisto
< *particle_name label* n_x n_y >

Arguments: G4int n_x, n_x;
G4String *particle_name, label*

Function: Create a 2D histogram that register the upward integral flux of a given type of particle at the user selected detection altitudes in function of X and Y position for the case of the flat geometry. The flux is integrated over the energy range defined by the commands. The particle type is defined by the *particle_name* parameter. The histogram is identified by the *label* parameter in the histogram tree. The X and Y axis are divided into n_x and n_y bins and cover all the X and X extension of the simulation box.

15.2.14/PLANETOCOS/ANALYSIS/SECONDARY/SetIntegralFluxEkinRange

Format: /PLANETOCOS/ANALYSIS/SECONDARY/SetIntegralFluxEkinRange
$<E_{min}\ E_{max}\ Eunit>$

Arguments: *G4int* E_{min}, E_{max};
G4String Eunit

Function: Defines the energy range over which the flux will be integrated for the next created flux vs position histograms.

15.2.15/PLANETOCOS/ANALYSIS/SECONDARY/SetLatitudeLongitudeLimits

Format: /PLANETOCOS/ANALYSIS/SECONDARY/SetLatitudeLongitudeLimits < lat_{min} lat_{max} lon_{min} lon_{max} >

Arguments: Double lat_{min}, lat_{max}, lon_{min}, lon_{max}

Function: Defines the latitude and longitude ([-180,180]) coverage of the next created secondary flux histograms.

15.2.16 /PLANETOCOS/ ANALYSIS/SECONDARY/SetTypeOfWeight

Format: /PLANETOCOS/ANALYSIS/SECONDARY/SetTypeOfWeight *type_of_weight*

Arguments: string *type_of_weight*

Function: Select the type of weight for the next created histograms used for registering flux of secondary particles . If the parameter *type_of_weight* is “INVERSE_COSTH “the weight of a detected particle in these histograms will be 1/cos(th) where th is the angle between the particle momentum and the vertical direction while if the parameter *type_of_weight* is “ONE” the weight of a detected particle is set to 1.

Candidates: INVERSE_COSTH, ONE

15.3 Energy deposited in the atmosphere

It is possible to detect the energy deposited by cosmic ray showers in the atmosphere in function of altitude and/or atmospheric depths. This information can be used for computing the ionisation rate of the atmosphere by cosmic rays.

The different UI commands to create secondary flux histograms are contained in the command directory /PLANETOCOS/ANALYSIS/EDEP.

15.3.1 /PLANETOCOS/ANALYSIS/EDEP/EdepVsAltitudeHisto

Format: /PLANETOCOS/ANALYSIS/EdepVsAltitudeHisto < *label nbins*>

Arguments: integer *nbins*
String *label*

Function: Create the histogram to register the energy deposited in the atmosphere in function of altitude. The altitude axis is divided into *nbins* bins from the bottom of the atmosphere to the top of the atmosphere. The histogram is identified by the *label* parameter in the histogram tree.

15.3.2 /PLANETOCOS/ANALYSIS/EDEP/EDEP/EdepVsDepthHisto

Format: /PLANETOCOS/ANALYSIS/EdepVsDepthHisto < *label nbins*>

Arguments: integer *nbins*
String *label*

Function: Create an histogram to register the energy deposited in function of atmospheric depth. The depth axis is divided into *nbins* bins from the top of the atmosphere (depth 0) to the bottom of the atmosphere. The histogram is identified by the *label* parameter in the histogram tree.

15.4 Registering of quasi trapped particles

Some secondary charged particles and backward scattering primary particles can escape the planet soil and atmosphere and travel for a while in the magnetophere before it interacts again with the planet atmosphere and/or soil or escape the magnetosphere . These so called quasi trapped particle population can be studied in PLANETOCOSMICS, by registering in 2D histograms the following kind of information :

- The number of turns that a particle makes around the planet rotation axis during its life vs the particle starting kinetic energy
- The number of turns that a particle makes around the planet rotation axis during its live vs the particle life time
- The number of times a particle crosses the planet equatorial plane during its live vs the particle starting kinetic energy
- The number of times a particle crosses the planet equatorial plane during its live vs the particle life time
- The number of turns that a particle makes around the planet rotation axis during its life vs the nb of times it crosses the planet equatorial plane
- The life time of a particle vs its starting kinetic energy

The different UI commands to create histograms for registering quasi trapped particles are contained in the command directory /PLANETOCOS/ANALYSIS/QUASITRAPPED.

15.4.1 /PLANETOCOS/ANALYSIS/QUASITRAPPED/NbOfPlanetTurnVsStartEkinHisto

Format: /PLANETOCOS/ANALYSIS/QUASITRAPPED/NbOfPlanetTurnVsStartEkinHisto
<*particle_name label nE E_{min} E_{max} Eunit scale_type $_{nbinturn}$ max_nb_turn* >

Arguments: integer *nE*, *$nbin_{turn}$*
double *E_{min}, E_{max}, max_nb_turn* String
particle_name, label, Eunit, scale_type

Function: Create an histogram to register the number of turns that particles make around the planet rotation axis during their life vs their starting kinetic energy. The particle type is defined by the *particle_name* parameter. The histogram is identified by the *label* parameter in the histogram tree. The starting energy axis is divided into *nE* bins covering the energy range [E_{min}, E_{max}]. The parameter *scale_type* defines the type of scale used for the energy axis, that can be linear or logarithmic (*scale_type* = [*lin, linear, Linear, Lin, LINEAR, LIN, log, Log, LOG]*. The axis representing the number of turns around the planet is divided into $nbin_{turn}$ covering the range [0, *max_nb_turn*].

15.4.2 /PLANETOCOS/ANALYSIS/QUASITRAPPED/NbOfPlanetTurnVsLifeTimeHisto

Format: /PLANETOCOS/ANALYSIS/QUASITRAPPED/NbOfPlanetTurnVsLifeTimeHisto
<particle_name label
$nbin_{time}$ t_{min} t_{max} *time_unit scale_type* $nbin_{tiurn}$ *max_nb_turn* >

Arguments: integer $nbin_{time}$, $nbin_{turn}$
double t_{min}, t_{max}, *max_nb_turn* String
particle_name, label, time_unit, , scale_type

Function: Create an histogram to register the number of turns that particles make around the planet rotation axis during their life vs their life time. The particle type is defined by the *particle_name* parameter. The histogram is identified by the *label* parameter in the histogram tree. The life time axis is divided into $nbin_{time}$ bins covering the range [t_{min}, t_{max}]. The parameter *scale_type* defines the type of scale used for the life time axis, that can be linear or logarithmic (*scale_type* = [*lin, linear, Linear, Lin, LINEAR, LIN, log, Log, LOG]*. The axis representing the number of turns around the planet is divided into $nbin_{turn}$ covering the range [0, *max_nb_turn*].

15.4.3 /PLANETOCOS/ANALYSIS/QUASITRAPPED/NbOfEquatorCrossingVsStartEkinHisto

Format: /PLANETOCOS/ANALYSIS/QUASITRAPPED/NbOfEquatorCrossingVsStartEkinHisto
<particle_name label nE E_{min} E_{max} *Eunit scale_type max_nb_cross* >

Arguments: integer *nE*, *max_nb_cross*
double E_{min}, E_{max}, String
particle_name, label, Eunit, scale_type

Function: Create an histogram to register the number of times particles cross the equatorial plane during their life vs their starting kinetic energy. The particle type is defined by the *particle_name* parameter. The histogram is identified by the *label* parameter in the histogram tree. The starting energy axis is divided into *nE* bins covering the energy range [E_{min}, E_{max}]. The parameter *scale_type* defines the type of scale used for the energy axis, that can be linear or logarithmic (*scale_type* = [*lin, linear, Linear, Lin, LINEAR, LIN, log, Log, LOG]*. The axis representing the number of turns around the planet is divided into *max_nb_cross*+1 bins covering the range [-0.5, *max_nb_cross*+0.5].

15.4.4 /PLANETOCOS/ANALYSIS/QUASITRAPPED/NbOfEquatorCrossingVsLifeTimeHisto

Format: /PLANETOCOS/ANALYSIS/QUASITRAPPED/NbOfEquatorCrossingVsLifeTimeHisto
<*particle_name label* $nbin_{time}$ t_{min} t_{max} *time_unit scale_type max_nb_cross* >

Arguments: integer $nbin_{time}$, *max_nb_cross*
double t_{min}, t_{max} String
particle_name, label, time_unit, scale_type

Function: Create an histogram to register the number of times that particles cross the planet equatorial plane during their life vs their life time. The particle type is defined by the *particle_name* parameter. The histogram is identified by the *label* parameter in the histogram tree. The life time axis is divided into $nbin_{time}$ bins covering the range [t_{min}, t_{max}]. The parameter *scale_type* defines the type of scale used for the life time axis, that can be linear or logarithmic (*scale_type* = [*lin, linear, Linear, Lin, LINEAR, LIN, log, Log, LOG]*. The axis representing the number of turns around the planet is divided into *max_nb_cross*+1 bins covering the range [-0.5, *max_nb_cross*+0.5].

15.4.5 /PLANETOCOS/ANALYSIS/QUASITRAPPED/NbOfEquatorCrossingVsNbOfPlanetTurnHisto

Format:
PLANETOCOS/ANALYSIS/QUASITRAPPED/NbEquatorCrossingVsNbOfPlanetTurnHisto
<*particle_name label* $nbin_{tiurn}$ *max_nb_turn max_nb_cross* >

Arguments: integer $nbin_{turn}$ *max_nb_cross*
double *max_nb_turn* String
particle_name, label

Function: Create an histogram to register the number of times that particles cross the planet equatorial plane vs the number of times they turn around the planet. The particle type is defined by the *particle_name* parameter. The histogram is identified by the *label* parameter in the histogram tree. The axis representing the number of turns around the planet is divided into $nbin_{turn}$ covering the range [0, *max_nb_turn*]. The axis representing the number of turns around the planet is divided into *max_nb_cross*+1 bins covering the range [-0.5, *max_nb_cross*+0.5].

15.4.6 /PLANETOCOS/ANALYSIS/QUASITRAPPED/LifeTimeVsStartEkinHisto

Format: /PLANETOCOS/ANALYSIS/QUASITRAPPED/LifeTimeVsStartEkinHisto
<*particle_name label*
nE E_{min} E_{max} *Eunit E_scale_type*
$nbin_{time}$ t_{min} t_{max} *time_unit time_scale_type* >

Arguments: integer nE, $nbin_{time}$
double E_{min}, E_{max}, t_{min}, t_{max} String
particle_name, label, Eunit, E_scale_type, time_unit, time_scale_type

Function: Create an histogram to register that particle lifetime vs the particle starting energy. The particle type is defined by the *particle_name* parameter. The histogram is identified by the *label* parameter in the histogram tree. The starting energy axis is divided into *nE* bins covering the energy range [E_{min}, E_{max}]. The parameter *E_scale_type* defines the type of scale used for the energy axis, that can be linear or logarithmic ([*lin, linear, Linear, Lin, LINEAR, LIN, log, Log, LOG]*. The life time axis is divided into $nbin_{time}$ bins covering the range [t_{min}, t_{max}]. The parameter *time_scale_type* defines the type of scale used for the life time axis, that can be linear or logarithmic.

.

15.5 Cosmogenic nuclides

All nuclides produced by the inelastic hadronic interaction of cosmic rays with the planet atmosphere are registered in function of their neutron number N and charge number Z in a 2D histograms located in the histogram tree in the /COSMONUC directory.

15.6 Normalisation

In PLANETOCOSMICS the results of the simulation can be saved without normalisation , with a normalisation to the flux of primary particle or with a normalisation per primary particle.

For normalising the results to the primary spectrum in the case of a flat geometry, the histograms have to be scaled by a normalisation factor

$$F_{norm} = J_{prim}^{down} / N_{prim}$$

where $J_{primary}^{down}$ represents the downward flux of primaries entering the atmosphere and N_{prim} represents the number of primaries that have been considered during the simulations. In the case of the spherical geometry the normalisation factor becomes

$$F_{norm} = \frac{R_{prim}^2 J_{prim}^{down}}{R_{det}^2 N_{prim}}$$

where R_{prim} and R_{det} represent the distance from the Earth's center to the primary source and to the detector respectively. If the directional differential flux of primary particles $j_{primary}(\theta, \phi, E)$ is known , we have

$$J_{primary}^{down} = \int_0^{\pi/2} \int_0^{2\pi} \int_{E_{min}}^{E_{max}} j_{primary}(\theta, \phi, E) \cos(\theta) \sin(\theta) d\theta d\phi dE \quad [cm^{-2} s^{-1}]$$

that becomes in the case of an isotropic flux

$$J_{primary}^{down} = \pi \int_{E_{min}}^{E_{max}} j_{primary}(E) dE \quad [cm^{-2} s^{-1}]$$

This type of normalisation can be applied in PLANETOCOSMICS if the user select one of the primary cosmic ray model defined in section 5. If he defines another type of source by using the /gps commands, he should normalised the results himself.

When the normalisation per primary particle is considered the histograms are scaled by a normalisation factor

$$F_{norm} = 1 / N_{prim}$$

where N_{prim} represents the number of primaries that have been processed during the simulations.

15.7 Output format when using the ROOT version of PLANETOCOSMICS

If the ROOT version of the PLANETOCOSMICS code is used, the results of the simulation can be saved into a ROOT file or an ASCII file.

In the ROOT file the entire histogram tree is saved under a directory specified by the user. By this way several trees can be saved on the same ROOT file under different directories. Histograms can be saved without normalisation, with normalisation to the primary flux or with normalisation per one incident particle (see section 15.6). The general, x and y titles of an histogram define the type of information contained in this histogram.

In the ASCII file the first lines defined some general information on the simulation (type of geometry, type of normalisation, nb of events processed). The rest of the file is divided into successive sections corresponding to the different histograms contained in the tree. In these sections the 1D histograms are written as 5-column tables where the 1st ,2nd ,3rd ,4th and 5th columns represent the lower limit, upper limit, mean value, height and error of the histogram bins respectively. The 2D histograms are written as 6-columns tables where the 1st, 2nd, 3rd, 4th, 5th, and 6th columns represent the x lower limit, x upper limit, y lower limit, y upper limit, height and error of the histogram bins, respectively. At the top of each section the path of the corresponding histogram in the tree is written followed by information's defining the histogram title, the x and y titles, and the factor of normalisation.

15.8 Output format when using the AIDA version of PLANETOCOSMICS

All the histograms contained in the permanent tree can be saved into a file in one of the different formats supported by the AIDA compliant library that is used for registering the histogram. In the following we will call these files AIDA files. In the case of the ANAPHE static version of PLANETOCOSMICS that has been linked with a static version of the ANAPHE library., two different AIDA output format can be considered: “xml” and “hbook”.

In an AIDA file the entire histogram tree is saved under a directory specified by the user. By this way several trees can be saved on the same AIDA file under different directories. Histograms are saved without normalisation, with normalisation to the primary flux or with normalisation per one incident particle (see section 15.6). Different annotations are added to each histogram in order to define its path in the tree, the histogram title, the x-axis and y-axis variable, the type of the normalisation applied to the histograms(“NONE”, “TO_PRIMARY_FLUX”. “PER_PRIMARY”), and the factor by which the histogram has been scaled for purpose of normalisation.

Histograms contained into a subtree of the permanent tree or in a subtree registered in an AIDA file can be saved into an ASCII file. This ASCII file is divided into successive sections corresponding to the different histograms. In these sections the 1D histograms a written as 5-column tables where the 1st ,2nd , 3rd ,4th and 5th columns represent the lower limit, upper limit, mean value, height and error of the histogram bins respectively. The 2D histograms are written as 6-columns tables where the 1st, 2nd, 3rd, 4th, 5th, and 6th columns represent the x lower limit, x upper limit, y lower limit, y upper limit, height and error of the histogram bins, respectively. At the top of each section the path of the corresponding histogram is written followed by information's defining the histogram title, the title of the x and y (only for 2D) axis, the physical meaning of the histogram height. If the subtree to be registered is taken from the permanent tree the nb of events, the type of normalisation, and the factor of normalisation are written at the beginning of the ASCII file. If the subtree is read from an AIDA file, this information is written in the different histogram section.

15.9 Command directory /PLANETOCOS/ANALYSIS

In this section we describe the different UI commands contains in the directory "/PLANETOCOS/ANALYSIS "

15.9.1 /PLANETOCOS/ANALYSIS/ResetHistograms

Format: /PLANETOCOS/ANALYSIS/ResetHistograms

Argument: none

Function: Reset all the histograms of the histogram tree.

15.9.2 /PLANETOCOS/ANALYSIS/ResetTree

Format: /PLANETOCOS/ANALYSIS/ResetTree

Argument: none

Function: Remove all the histograms from the histogram tree.

15.9.3 /PLANETOCOS/ANALYSIS/SaveTree

Format: /PLANETOCOS/ANALYSIS/SaveTree
<file_name file_type dir normalisation_type>

Argument: G4String *file_name*, *file_format, dir, normalisation_type*

Function: Save the histograms tree into a file with or without normalisation. The parameter *file_name* defines the name of the file. The format of the file is specified by the argument *file_format.* In the ROOT version of PLANETOCOSMICS possible file formats are : root, ascii. In the case of the AIDA version of the code the different file formats that can be selected depends on the library realizing the AIDA interface. In the case of ANAPHE the following format can be selected: xml, hbook and ascii. Except for the ascii the histogram tree will be copied in the directory specified by the *dir* argument. The argument *normalisation_type* defines the type of normalisation. If it is set to NONE or if the argument is not given no normalisation is considered. If it is set to TO_PRIMARY_FLUX, the histograms are normalised to the flux of primary particles. If *normalisation_type* is set to PER_PRIMARY the results are normalised to the number of primaries processed during the simulation.

15.9.4 /PLANETOCOS/ANALYSIS/WriteTreeInASCIIFile

Format: /PLANETOCOS/ANALYSIS/WriteTreeInASCIIFile
<file_name dir [normalisation_type tree_file file_type]>

Argument: G4String ascii_file, dir, tree_file, file_type

Function: Write all the histograms of an histogram subtree specified by the path *dir* into an ascii file *file_name.* The histogram subtree is taken either from the permanent histogram tree where the simulation results are registered during the execution of PLANETOCOSMICS, or from a user defined file if the argument *tree_file* and *file_type,* that define the name and type of the file respectively, are specified. In the first case the normalisation of the histogram is considered only in the case of writing histograms from the permanent tree and only if the *normalisation_type* argument is specified. If this argument is set to TO_PRIMARY_FLUX or to PER_PRIMARY the histograms are normalised to the flux of primary particles or to an incident flux of one primary particle per second per cm^2 respectively. Otherwise no normalisation is considered.

Remarks: This command is only available for the AIDA version of the code.

15.9.5 /PLANETOCOS/ANALYSIS/SetSecuritySave

Format: /PLANETOCOS/ANALYSIS/SetSecuritySave <*aBool* >

Argument: G4bool *aBool*

Function: If *aBool* is true the histogram tree is saved periodically after a serie of user specified fix number of events (see SetSecurityNbOfEvents), in a user specified tree file (see SetSecurityFile) .

15.9.6 /PLANETOCOS/ANALYSIS/SetSecurityFile

Format: /PLANETOCOS/ANALYSIS/SetSecurityFile <*file_name file_type*>

Argument: G4String *file_name*, *file_type*

Function: Defines the file where a security copy of the histogram tree will be periodically saved. The arguments *file_name* and *type* represents the name and type of the security file. In this file the tree is registered under the directory /security.

15.9.7 /PLANETOCOS/ANALYSIS/SetSecurityNbOfEvents

Format: /PLANETOCOS/ANALYSIS/SetSecurityNbOfEvents nb

Argument : G4int nb

Function: Defines the number of events after which a security copy of the histogram tree should be saved.

16 Visualisation

In PLANETOCOSMICS it is possible to draw primary and secondary particle trajectories and magnetic field lines in different user defined colours. If specified by the user, particle trajectories and magnetic field lines computed during different simulations are stored in a vector of curves. If needed the different step positions of the trajectory are also stored in a vector of step positions. These vectors can be drawn at any time in a user selected system of coordinates. When plotted the step points are represented by full circle of a user selected size. At any time it is possible to clear the vector of trajectories and step positions.

16.1 Command directory /PLANETOCOS/DRAW/

The commands contained in the directory /PLANETOCOS/DRAW/ allows to define the visualisation parameters of particle trajectories and magnetic field lines, as well as to trace magnetic field lines from one or several user defined positions.

16.1.1 /PLANETOCOS/DRAW/TraceBlineAtPositions

Format: /PLANETOCOS/DRAW/TraceBlineAtPositions
<*coorsys alt length_unit* $lat_0\ d_{lat}\ n_{lat}\ long_0\ d_{long}\ n_{long}$ *angle_unit*>

Arguments: Double alt, lat_0, d_{lat}, $long_0$, d_{long}
Integer n_{lat}, n_{long}
String *coorsys, length_unit, angle_unit*

Function: Trace the magnetic field lines going trough different positions. The positions are located at the same altitude *alt,* but at the different latitudes lat_i and longitude $long_j$ given by

$$lat_i = lat_0 + i \cdot d_{lat} \qquad i = 0 \cdots n_{lat} - 1$$
$$long_j = long_0 + j \cdot d_{long} \qquad j = 0 \cdots n_{long} - 1$$

The parameter *coorsys* defines the system of coordinate in which these positions are defined.

Remark: This command is only available for the spherical geometry.

16.1.2 /PLANETOCOS/DRAW/TraceBlineAtXYAltPos

Format: /PLANETOCOS/DRAW/TraceBlineAtXYAltPos < $alt\ x_0\ d_x\ n_x\ y_0\ d_y\ n_y$ *length_unit* >

Arguments: Double alt, x_0, d_x, y_0, d_y
String *length_unit*,*output_file*
Integer n_x, n_y

Function: Trace the magnetic field lines going trough the different nodes of a 2D grid. On this grid the node *(i,j)* is located at the altitude *alt*, x-coordinate x_i and y-coordinate y_j defined by

$$x_i = x_0 + i \cdot d_x \qquad i = 0 \cdots n_x - 1$$
$$y_j = y_0 + j \cdot d_y \qquad j = 0 \cdots n_y - 1$$

Remark: This command can only be used in the case of a flat geometry.

16.1.3 /PLANETOCOS/DRAW/SetColourForBline.

Format: /PLANETOCOS/DRAW/SetColourForBline. <*red green blue*>

Arguments: G4double *red, green, blue*

Function: Defines the colour for visualising the next computed magnetic field lines. The color is defined by a RGB code.

Default: (1,0,0) corresponding to red.

Errors: The values of the color index should be in the range [0.,1].

16.1.4 /PLANETOCOS/DRAW/SetCoordinateSystem

Format: /PLANETOCOS/DRAW/SetCoordinateSystem <*coorsys*>

Arguments: G4String *coorsys*

Function: Define the coordinate system that will be used for drawing trajectories and magnetic field lines.

16.1.5 /PLANETOCOS/DRAW/DrawTrajectory

Format: /PLANETOCOS/DrawTrajectory < *aBool*>

Arguments: G4bool *aBool*

Function: If *aBool is* true the next computed particle trajectories and magnetic field lines are added to the vector of curves that are visualised later on by using the command "/PLANETOCOS/DRAW/Show".

16.1.6 /PLANETOCOS/DRAW/DrawPoints

Format: /PLANETOCOS/DrawPoints < *aBool*>

Arguments: G4bool *aBool*

Function: If *aBool* is true the step points of particle trajectories and magnetic field lines computed during the next runs are added to the vector of positions that can be visualised later on by using the command "/PLANETOCOS/DRAW/Show".

16.1.7 /PLANETOCOS/DRAW/SetPointSize

Format: /PLANETOCOS/DRAW/SetPointSize

Arguments: G4double *size*

Function: Set the size of the circle used for visualising the step points of the next computed particle trajectory and magnetic field lines.

Default: 1

16.1.8 /PLANETOCOS/DRAW/Show

Format: /PLANETOCOS/DRAW/Show

Function: Visualise the trajectories and the step positions that have been registered during the last runs.

16.1.9 /PLANETOCOS/DRAW/Reset

Format: /PLANETOCOS/DRAW/Reset

Function: Clear the vectors of curves and step positions that had been registered before for purpose of visualisation.

16.1.10/PLANETOCOS/DRAWING/SetParticleTrajectoryColour

Format: /PLANETOCOS/DRAWING/SetParticleTrajectoryColour
<particle_name red green blue>

Argument: G4String *particle_name*
G4double *red, green, blue*

Function: If a visualisation driver is defined and the storing of particle trajectory(see 15.1.5 and 15.1.6) is selected, the trajectories of the particles of type *particle_name* are stored for later visualisation. At visualisation these trajectories are plotted in the colour specified by the RGB code defined by the parameters *red, green* and *blue* that should be in the range [0,1].

16.1.11 /PLANETOCOS/DRAWING/DoNotDrawParticleTrajectory

Format: /PLANETOCOS/DRAWING/DoNotDrawParticleTrajectory <*particle_name*>

Argument: G4String particle_name

Function: The particle with name *particle_name* is removed from the list of particles for which trajectories should be registered for a later visualisation .

17 Numerical integration

For computing the trajectory of a charged particle trough a magnetic field, the Lorentz equation of motion is integrated numerically. For tracing the magnetic field lines the Lorentz equation is replaced by a differential equation, that defines the motion parallel to the magnetic field.

In the Geant4 toolkit, the trajectory or track of a particle is divided in tracking steps. For non linear motion trough magnetic and electric fields these tracking steps are divided in smaller steps called chord. The equation of motion is integrated over these chords. After the integration of a chord it is checked if the particle did not cross a boundary between two different regions of the geometry. If this is the case the intersection with a boundary is determined and the tracking step is stopped at this boundary. To avoid the non-detection of boundaries an upper limit is given to the length of chord. This limits is defined by the parameter *MaxChord* and can be changed according to the application.

For the numerical integration, the chord is divided into small integration steps. The motion of the particle over a small integration step is done by a G4Stepper object. Different type of G4Stepper objects are available corresponding to different integration algorithms. In PLANETOCOSMICS the user can choose between the following stepper methods: Euler implicit, Euler explicit, 3rd order RungeKutta , 4th order RungeKutta, and KashKarper. For more information on these algorithms we refer to the Numerical recipes and to the Geant4 documentation[3,49]. After each integration step an estimate of the numerical relative error is computed, if this error is bigger than the maximum accepted relative error ξ, a smaller integration step is chosen and the integration restart from precedent step. When the relative error is significantly smaller than ξ, the step size is increased.

17.1 Command directory /PLANETOCOS/INTEGRATION

The commands in this directory allows to fix the parameters of the numerical algorithm used to integrate the lorentz equation of motion and the magnetic field line differential equation.

17.1.1 /PLANETOCOS/INTEGRATION/SetPrecision

Format: /PLANETOCOS/INTEGRATION/SetPrecision <*epsilon*>

Arguments: G4double *epsilon*

Function: Set the relative precision for the numerical integration of the Lorentz equation of motion.

Default: 10^{-5}

17.1.2 /PLANETOCOS/INTEGRATION/SetMaxChord

Format: /PLANETOCOS/INTEGRATION/SetMaxChord <*max_chord length_unit*>

Arguments: G4double *max_chord*
G4String *length_unit*

Function: Set the maximum length allowed for a chord in the numerical integration algorithm used in the Geant4 toolkit.

Default: 10^{-2} Rplanet

Candidates: *length_unit: km, m, rplanet, Rplanet*

17.1.3 /PLANETOCOS/INTEGRATION/SetDeltaIntersection

Format: /PLANETOCOS/INTEGRATION/SetDeltaIntersection
<*delta_intersection length_unit*>

Arguments: G4double *delta_intersection*,
G4String *length_unit*

Function: Defines the precision for detection of crossing boundary.

Default : 10^{-3} Rplanet

Candidates: *length_unit: km, m, rplanet, Re*

17.1.4 /PLANETOCOS/INTEGRATION/ResetIntegrationParameters

Format: /PLANETOCOS/INTEGRATION/ResetIntegrationParameter

Arguments: None

Function: Reset the integration parameters to their default values.

17.1.5 /PLANETOCOS/INTEGRATION/SetStepperModel

Format: /PLANETOCOS/INTEGRATION/SetStepperModel <*stepper_model*>

Argument: G4String *stepper_model*

Function: Set the stepper integration model used for the numerical integration of the Lorentz motion equation in Geant4.

Candidates: ExplicitEuler, ImplicitEuler, SimpleRunge, ClassicalRK4, RKG3_Stepper, and CashKarpRKF45.

18 Step length

In the tracking algorithm of Geant4 a particle track is is divided into tracking steps. For charged particle in a magnetic field, the Lorentz equation of motion is integrated over a track step. A track step is therefore divided in smaller integration steps. Integration steps and tracking steps are therefore not the same. In PLANETOCOSMICS it is possible to fix the maximum length of a tracking stepin the atmosphere and in the magnetosphere. For visualisation purpose, the trajectory is divided by straight lines representing the different tracking step. For this reason to obtain a rather smooth drawing of a particle trajectory you should select rather small maximum step length. Too small steps length limit the length of integration step and decrease the computing performance. Therefore if you are not considering visualisation you should not slect too small maximum step length.

18.1 Command directory /PLANETOCOS/USERLIMIT

The commands in this directory allow to fix the maximum length of tracking steps in the atmosphere and magnetosphere regions.

18.1.1 /PLANETOCOS/USERLIMIT/SetAtmoMaxStepLength

Format: /PLANETOCOS/USERLIMIT/SetAtmoMaxStepLength <*max_step length_unit*>

Arguments: Double *max_step*
String *length_unit*

Function: Set an upper limit to the length of the tracking step in the atmosphere.

18.1.2 /PLANETOCOS/USERLIMIT/SetMagnetoMaxStepLength

Format: /PLANETOCOS/USERLIMIT/SetMagnetoMaxStepLength <*max_step length_unit*>

Arguments: Double *max_step*
String *length_unit*

Function: Set an upper limit to the length of the tracking step in the magnetosphere.

19 Limiting the duration of the simulation

It is possible to fix an upper limit for the total duration of the execution of the code, for the duration of runs, and for the duration of events.

19.1 Command directory /PLANETOCOS/DURATION

19.1.1 /PLANETOCOS/DURATION/SetMaxTotalDuration

Format: /PLANETOCOS/DURATION/SetMaxTotalDuration <*max_duration time_unit*>

Argument: Double *max_duration*
String *time_unit*

Function: Set the maximum duration after which the execution of the program will be interrupted. This time can be given either in second, minute, hour, or day.

19.1.2 /PLANETOCOS/DURATION/SetMaxRunDuration

Format: /PLANETOCOS/DURATION/SetMaxRunDuration <*max_duration time_unit*>

Argument: Double *max_duration*
String *time_unit*

Function: Set the maximum duration after which a run will be interrupted. This time can be given either in second, minute, hour, or day.

19.1.3 /PLANETOCOS/DURATION/SetMaxEventDuration

Format: /PLANETOCOS/DURATION/SetMaxRunDuration <*max_duration time_unit*>

Argument: Double *max_duration*
String *time_unit*

Function: Set the maximum duration after which an event will be interrupted. This time can be given either in second, minute, hour, or day.

20 Random seed

We have defined a procedure to set randomly the random seed used by the generator of random numbers in Geant4 by fixing this seed in function of the time of the computer clock.

20.1 Command directory /PLANETOCOS/RANDOM

The command of this directory allows to fix the time after which a run will be aborted.

20.1.1 /ATMOCOSMICS/RANDOM/FixTheSeedRandomly

Format: /ATMOCOSMICS/RANDOM/FixTheSeedRandomly

Argument: none

Function: The random seed is computed randomly by using the computer clock.

20.1.2 /ATMOCOSMICS/RANDOM/SetRandomSeedAtRunStart

Format: /ATMOCOSMICS/RANDOM/SetRandomSeedAtRunStart < *aBool*>

Argument: G4bool *aBool*

Function: If *aBool* is true (false) the random seed will be recomputed randomly by using the computer clock at the beginning of each run.

21 Default simulation parameters

21.1 Geometry

By default the spherical geometry is selected. The outer limit of the space around the planet is located at 60 planet radius above the atmosphere. No detection altitude and depth are considered.

21.2 Earth's Atmosphere and Soil

By default the composition of the Earth's atmosphere is the one given by the NRMLSISE00 model at 0^o latitude and 0^o longitude at 12:00 on January 1^{st}, 2000. The altitude of the ground is set to 0 km. The altitude of the top of the atmosphere is set to 100 km. The maximum thickness of an atmospheric layer is 5 km. Each layer contains 5% of the total atmospheric depth.

By default the Earth's soil is modeled by a 10 m thick layer of SiO_2 with 1.7 g/cm^3 density.

21.3 Mars' Atmosphere and Soil

By default the composition of the Mars atmosphere is read from the table "./planetocosmics/data/mars/marsgram_atmo_tables/marsgram_atmo_table_0N180E.txt". This table represents the daily averaged atmosphere of Mars at 45^o north latitude and 180^o east longitude, on January$1^{st,}$ 2000. It was obtained from the MarsGram2001 model by running the python script "marsgram_to_atmotable.py" (see section 7.2.3). The maximum thickness of an atmospheric layer is 5 km. Each layer contains 5% of total atmospheric depth.

By default the Mars' soil is modeled by a 10 m thick layer with 1.7 g/cm^3 density. The composition of this layer is taken as given in the next table with the abundance's normalised to 100%. It corresponds to the Mars' soil composition derived from measurements on the Pathfinder-Sojourner mission [50].

Material	***Na2O***	***MgO***	***Al2O3***	***SiO2***	***SO3***	***K2O***	***CaO***	***TiO2***	***Fe2O3***
Abundance [%]	1,5	7,7	8,1	46,8	6	0,2	6,2	1,1	18,8

21.4 Mercury's Atmosphere and Soil

By default no atmosphere is considered for the planet Mercury. The Mercury's soil is modeled by a 10 m thick layer with 1.3 g/cm^3 density. The composition of this layer is taken as given in the next table with the abundance's normalised to 100%. This composition is based on the preferred model proposed by Goettel [51].

Material	***SiO2***	***MgO***	***Al2O3***	***CaO***	***FeO***	***Na2O***	***TiO2***
Abundance [%]	45	35	7	7	5	0,7	0,3

21.5 Magnetic field

By default the Earth magnetic field is represented by the IGRF model at 12:00 UT on 1^{st} January 2000, without an external field. The Mars magnetic field is represented by the CAIN90 crustal field model.

The Mercury's magnetic field is represented by a magnetic dipole with a 300 nT/rm^3 magnetic moment, and an axis that coincides with the planet rotation axis. Note that by default the magnetic field is switched off. It can be switched on by the command “/PLANETOCOS/BFIELD/SwitchOn”.

21.6 Electromagnetic and hadronic physics model

By default the standard electromagnetic and the QGSP_BIC_HP hadronic physics models are selected. For the light ion hadronic physics the BIC model is selected. No electromagnetic nuclear physics is considered.

22 Examples

The directory ./planetocosmics/examples contains different *g4mac macrofiles corresponding to different simulation examples of PLANETOCOSMICS. The other files corresponds to additional input files needed to run these examples as well as pictures illustrating some output of these examples. If you are using the VRML2 driver you should scale the *wrl file produced by PLANETOCOSMICS. This is done by typing the command "scale_vrmlfile.sh *file1 file2"* where *file1* is the name of the VRML2 file produced by PLANETOCOSMICS and *file2* is the output file that you will look at with your vrml viewer.

22.1 Example#1 for Mercury

This example is executed by typing "PLANETOCOSMICS Mercury mercury_example.1g4mac". It simulates the interaction of 10 GeV protons with Mercury. The default composition of the soil is considered. The magnetic field is modeled by a magnetic dipole with 300 nT/rm^3 magnetic moment, and an axis that coincides with the planet rotation axis. The flux of secondary particles are detected at 100 km and 400 km altitudes and the results are saved in different histograms. Figure 5 represents the visualisation of the interaction of ten primaries with Mercury. The trajectories of the primary particles are drawn in white. These particles interact directly with the soil. The yellow, green and red lines represent the trajectories of secondary protons, e- and e+ that are produced by the interaction of the primaries with the soil, and that escape from the soil in the magnetosphere and are deflected by the planet magnetic field. Some of these e- and e+ are precipitating on the soil after a shot journey in the magnetosphere while more energetic ones escape from the magnetosphere.

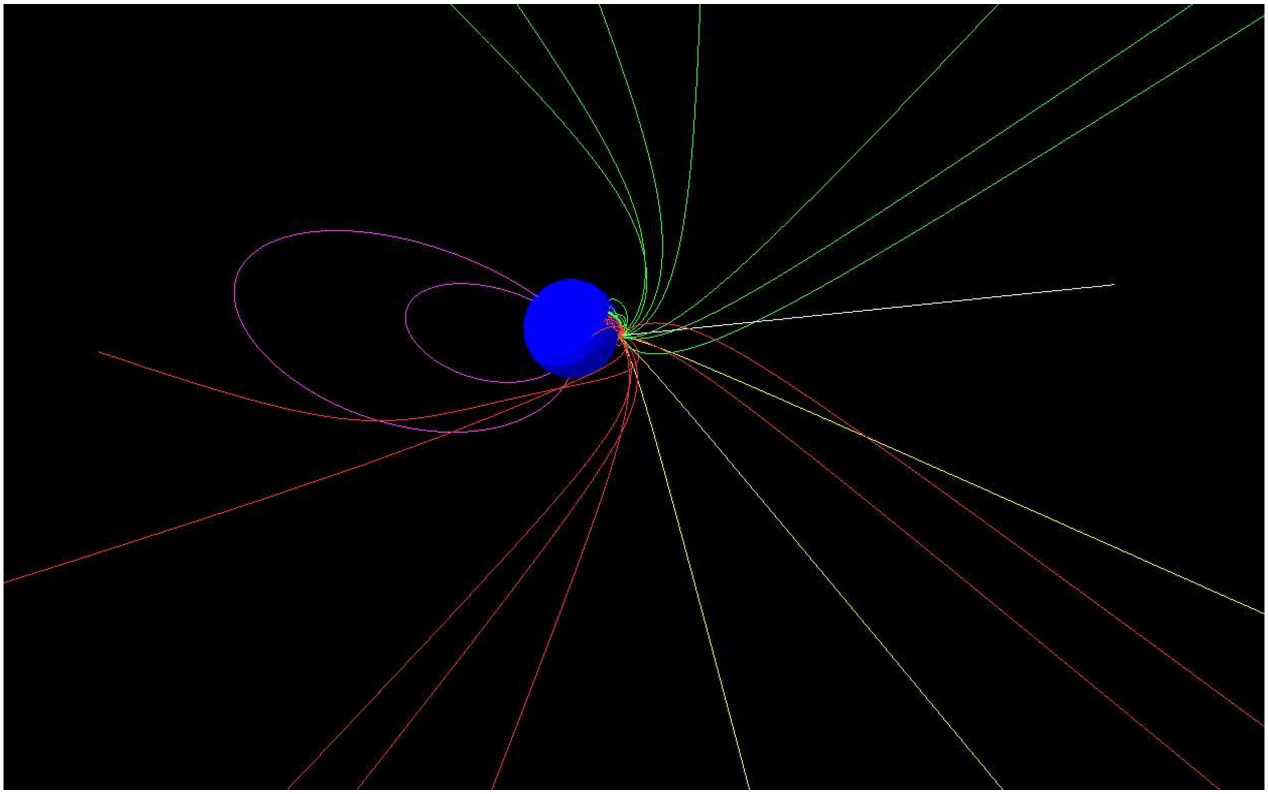

Figure 5 Illustration of the interaction of 10 GeV protons with the soil of Mercury. The white line represents the tracks of incoming primary protons. The yellow, green and red lines represent the track of secondary protons, e- and e+ that escape from the soil in the magnetosphere and are deflected by the planet magnetic field. Some e- and e+ are precipitating on the soil after a shot journey along a magnetic field line while more energetic ones can escape from the magnetosphere. The violet curves represent two magnetic field lines of the Mercury's dipole.

22.2 Example#1 for Mars

This example is executed by typing “PLANETOCOSMICS Mars mars_example.1g4mac”. It illustrates how to select and use different Mars' crustal magnetic field models in the case of a flat geometry. No atmosphere is considered in this example. The simulation box is 500 km x 500 km wide, 150 km high, and located at 48° south latitude and 174° east longitude. It corresponds to one of the most magnetised region of Mars. The visualisation output of this example is plotted in Figure 6. The yellow and green lines represent the magnetic field line configuration obtained with the CAIN90 and FLATGRID internal field models respectively. The violet lines represent the motion of normal incident 10 MeV e- in the Martian crustal field. In the case of the FLATGRID model a 50x50x50 grid is considered. The magnetic field components at the grid nodes are read in the file FlatGridCain90.txt and correspond to the magnetic field components calculated previously with the CAIN90 model for the same central position (48 S, 174 E). If the user want to recompute this file, he should remove the first character of the line #47 of the macrofile earth_example1.g4mac The fact that the green magnetic field lines are close to the yellow ones illustrates that the FLATGRID model gives a good approximation of the CAIN90 model. The user will notice by running this example that the FLATGRID model is much more rapid than the CAIN90 model.

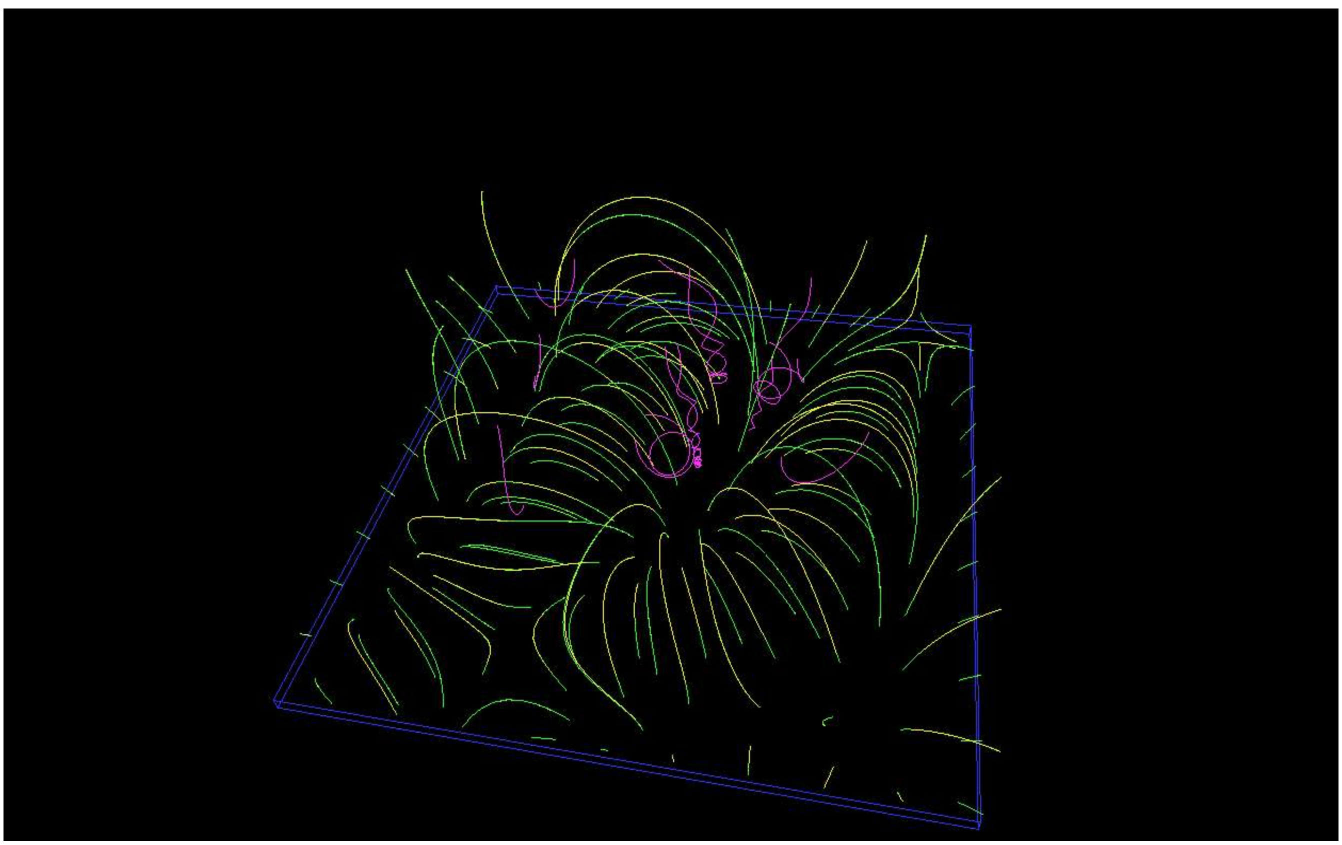

Figure 6Configuration of the Martian crustal field at 48° south latitude and 174° east longitude. The yellow and green lines represent the magnetic field lines obtained with the CAIN90 and FLATGRID models respectively. The violet lines represent the motion of 10 MeV normal incident electrons in the Martian crustal field.

22.3 Example#2 for Mars

This example is executed by typing "PLANETOCOSMICS Mars mars_example.2g4mac". It simulates the interaction of 5 GeV protons with the Mars' atmosphere and soil. The magnetic field is not considered. The structure of the atmosphere is read from the table "marsgram_atmo_table_45N180E.txt". This table represents the daily averaged atmosphere of Mars at 45^o north latitude and 180^o east longitude, on January1st 2000. It was obtained from the MarsGram2001 model by running the python script "marsgram_to_atmotable.py" (see section 7.2.3). The soil is represented by a 100 m thick layer of SiO2 with a density of 1.7g/cm3 on the top of a 50 m thick water layer. The flux of protons, e-, e+, mu-, mu+ , gamma and neutrons at different altitudes and depths are registered without normalisation. Figure 7 represents the visualisation output of the interactions of 10 primaries with Mars. The red, white, blue and green lines represent the trajectories of protons, neutron , gammas and e-. The primary protons are interacting mostly with the soil. Some secondaries produced during these interactions escape the soil upward and interact with the atmosphere.

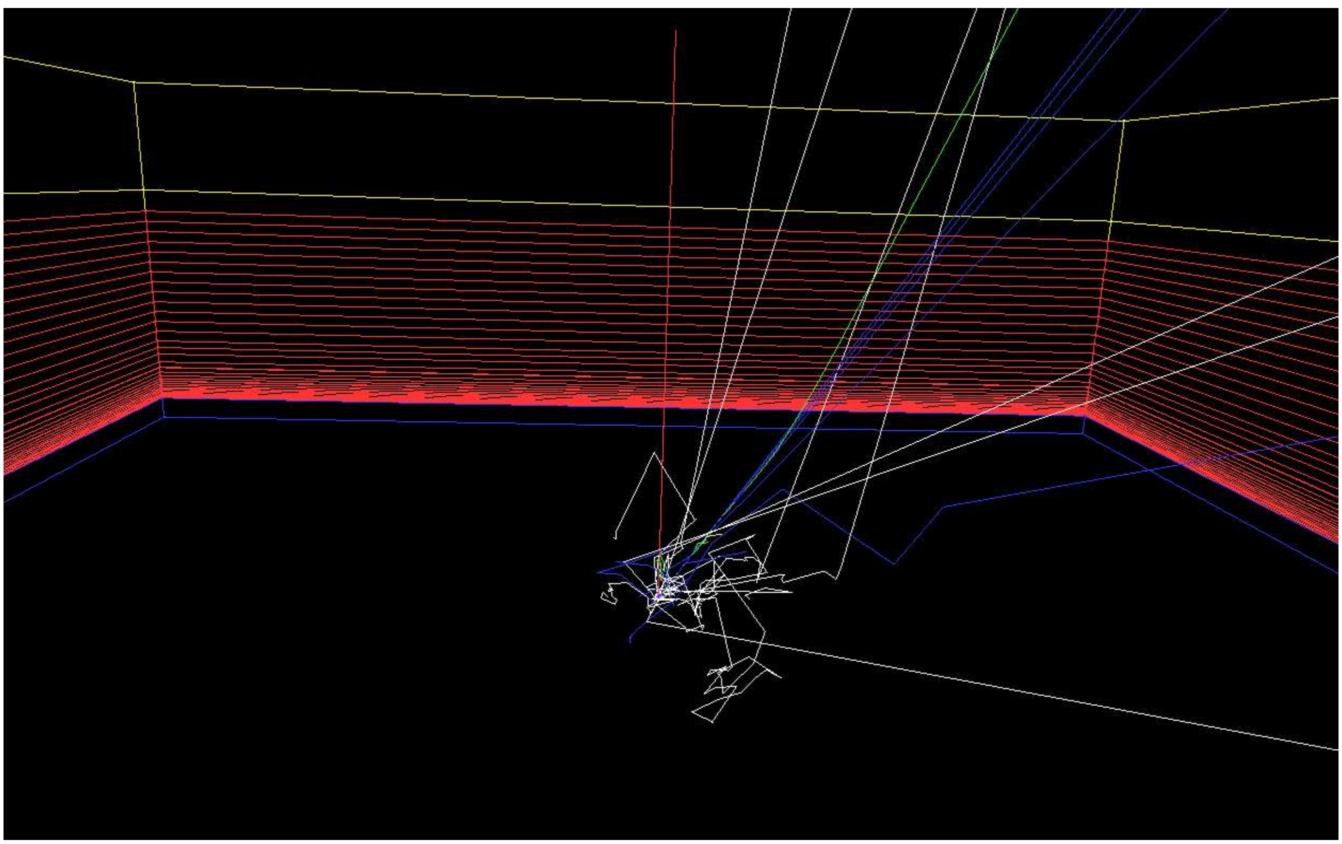

Figure 7 Interactions of 10 GeV protons with the Mars' soil and atmosphere. The red, white, blue and green lines represent the trajectories of protons, neutron , gammas and e-. The primary protons are interacting mostly with the soil. Some secondaries produced during these interactions escape the soil upward and interact with the atmosphere.

22.4 Example#1 for the Earth

This example is executed by typing "PLANETOCOSMICS Earth earth_example.1g4mac". It allows to visualise the magnetic field line configuration obtained with different models of the geomagnetic field and of the Earth's magnetospheric magnetic field. Figure 8 shows the last image obtained from this example. It represents the Earth's magnetosphere as given by the IGRF geomagnetic field model plus the Tsyganenko 2001 external field model.

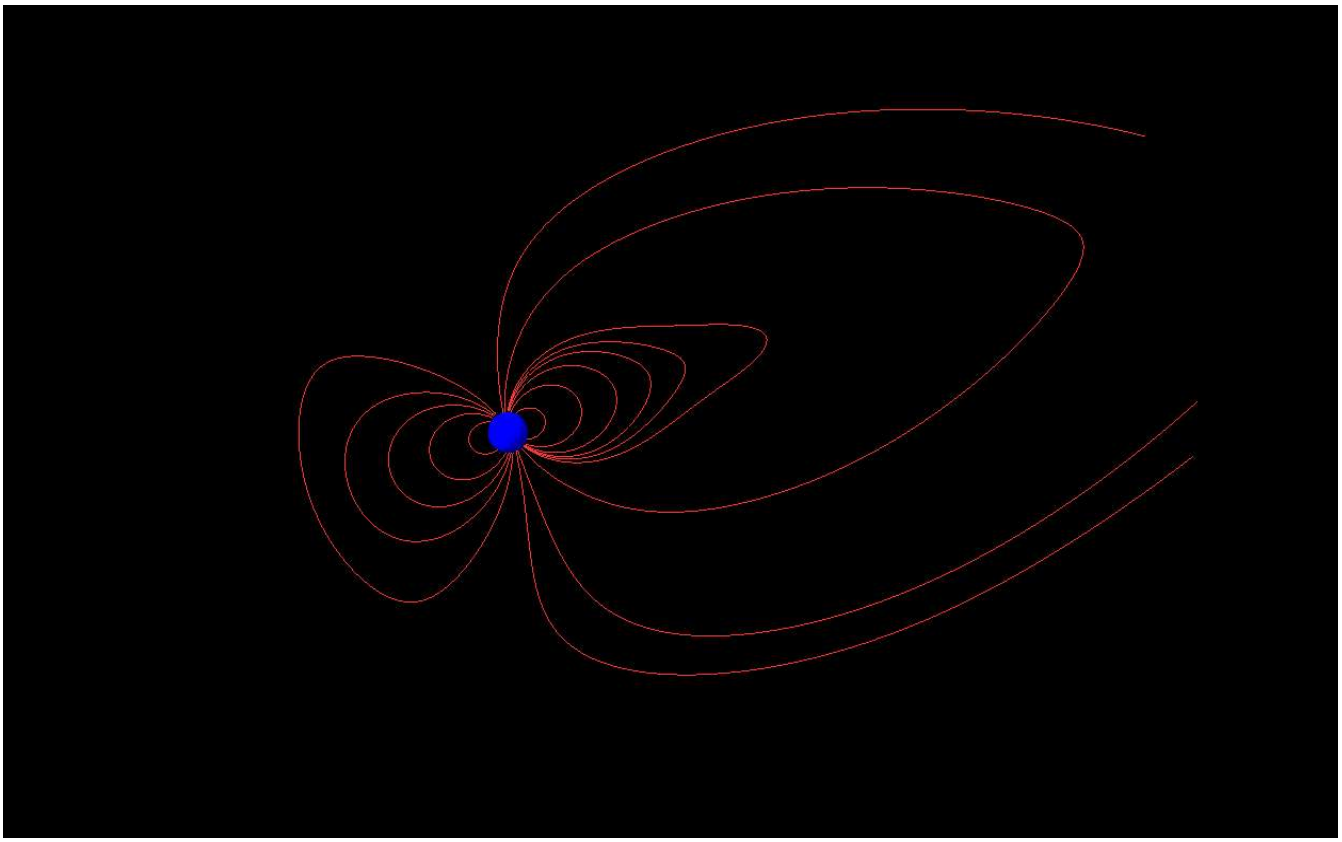

Figure 8Configuration of the Earth's magnetosphere as given by the IGRF model plus the Tsyganenko 2001 model.

22.5 Example#2 for the Earth

This example is executed by typing "PLANETOCOSMICS Earth earth_example.2g4mac". It allows to compute the cutoff rigidities at different positions on January 1^{st} ,1990. For this simulation the geomagnetic field is described by the IGRF model while no external field model is considered.

22.6 Example#3 for the Earth

This example is executed by typing “PLANETOCOSMICS Earth earth_example3.g4mac”. It allows to draw the trajectories of some charged particles in different models of the Earth magnetosphere. Figure 8 is the first image obtained from this example. It represents the motion of a 100 MeV proton in the geomagnetic field. The green and red curves are obtained by tracing the trajectory of the proton forward and backward respectively, from the same starting position. The white line represents a magnetic field line. Figure 9 is the last image obtained from this example. It represents the trajectories of 5 GeV protons computed backward from the position 20 km altitude, 46.55^{o} north latitude and 7.98^{0} east longitude. The red line is obtained by selecting the IGRFTiltedEccentric dipole as the internal field and without considering an external field. The blue line is obtained by selecting the IGRF model as the internal field and without considering an external field. The green line is obtained by selecting the IGRF model as the internal field plus the Tsyganenko89 model as the external field.

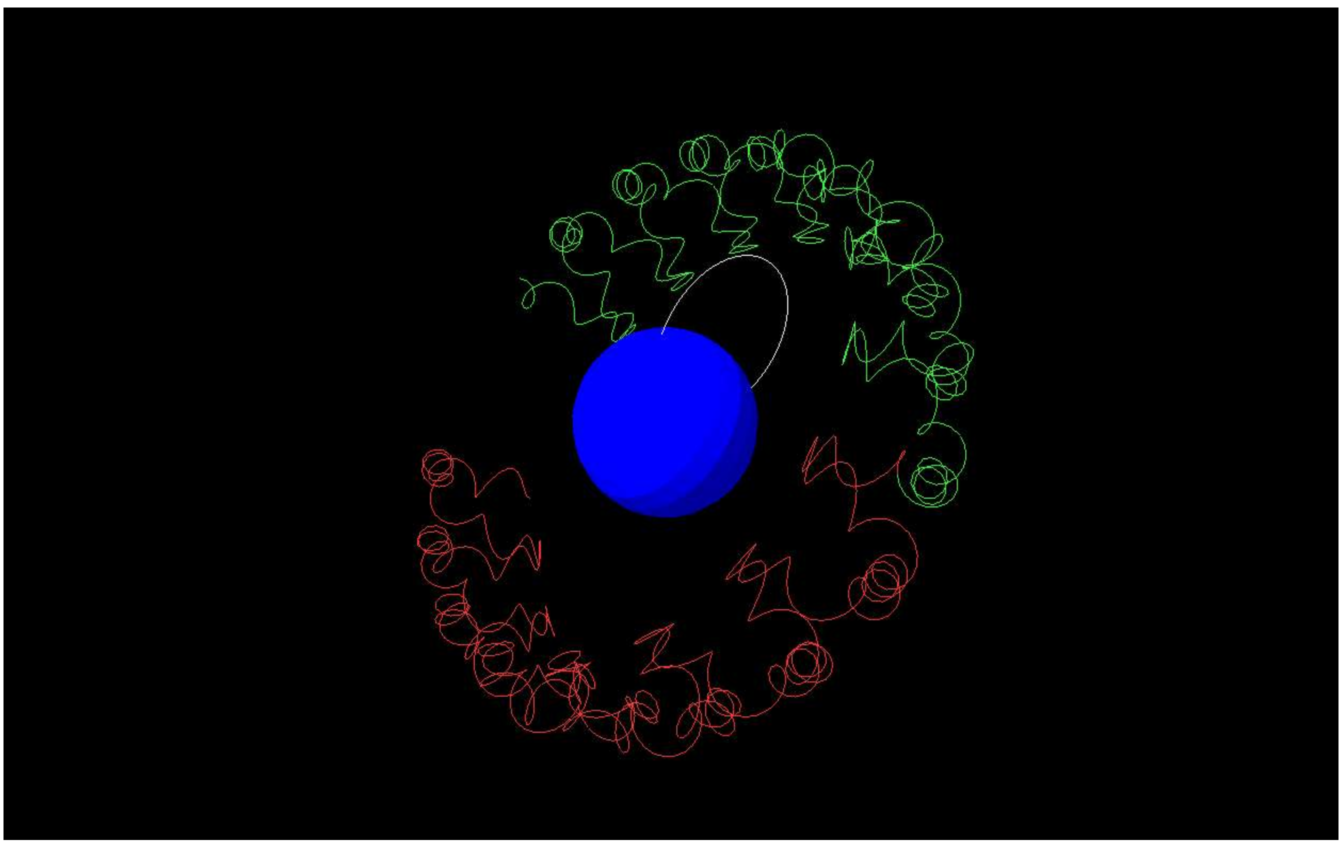

Figure 9Motion of a 100 MeV protons in the geomagnetic field. The green and red curves are obtained by tracing the trajectory of the proton forward and backward respectively, from the same starting position.

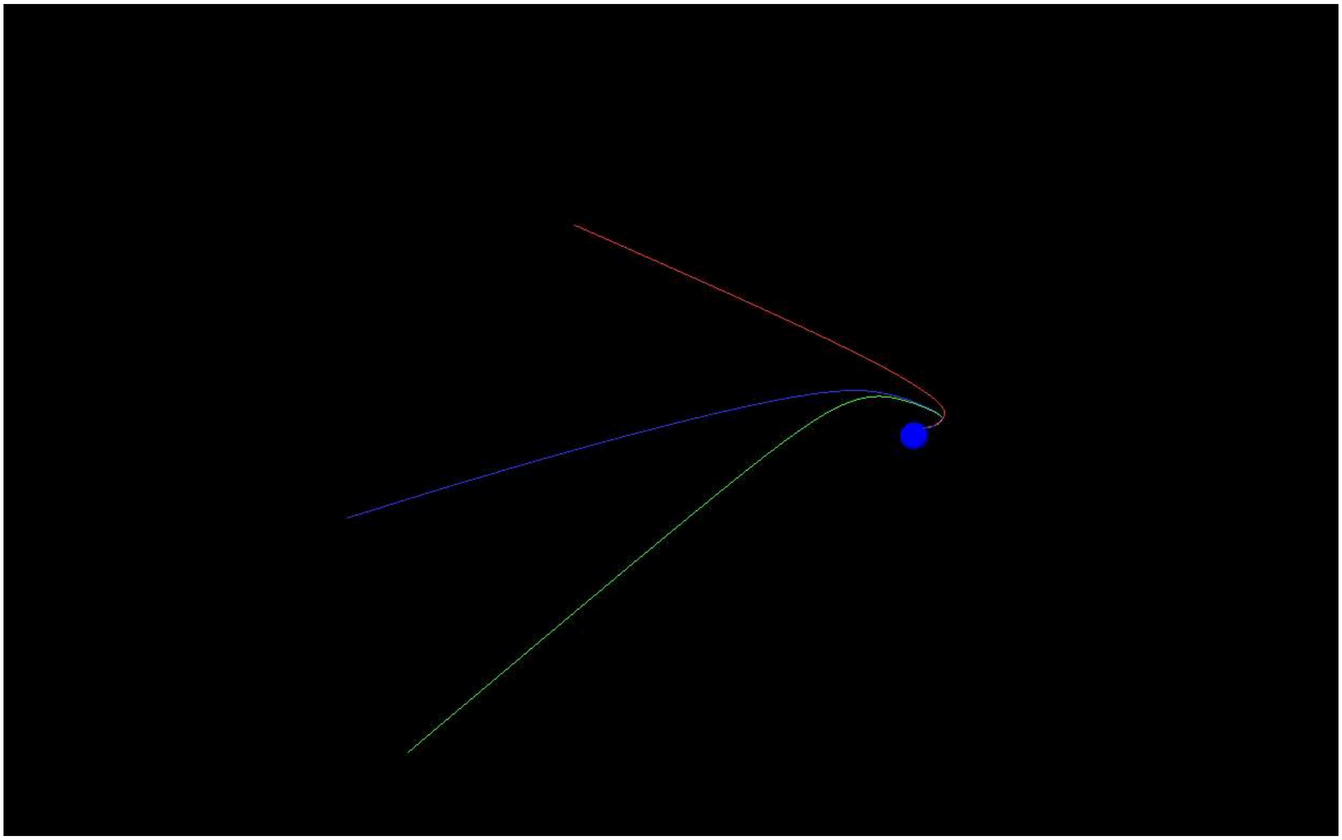

Figure 10Trajectories of 5 GeV protons computed backward from the position 20 km altitude, 46.55º north latitude and 7.98⁰ east longitude. The red line is obtained by selecting the IGRFTiltedEccentric dipole as the internal field and without considering an external field model. The blue line is obtained by selecting the IGRF model as the internal field and without considering an external field model. The green line is obtained by selecting the IGRF model as the internal field plus the Tsyganenko89 model as the external field.

22.7 Example#4 for the Earth

This example is executed by typing “PLANETOCOSMICS Earth earth_example4.g4mac”. It allows to compute the energy deposited by 100 MeV normal incident protons in the Earth atmosphere in function of altitude and depth. The histogram plotted in Figure 11 represents the energy deposited vs altitude computed during this simulation.

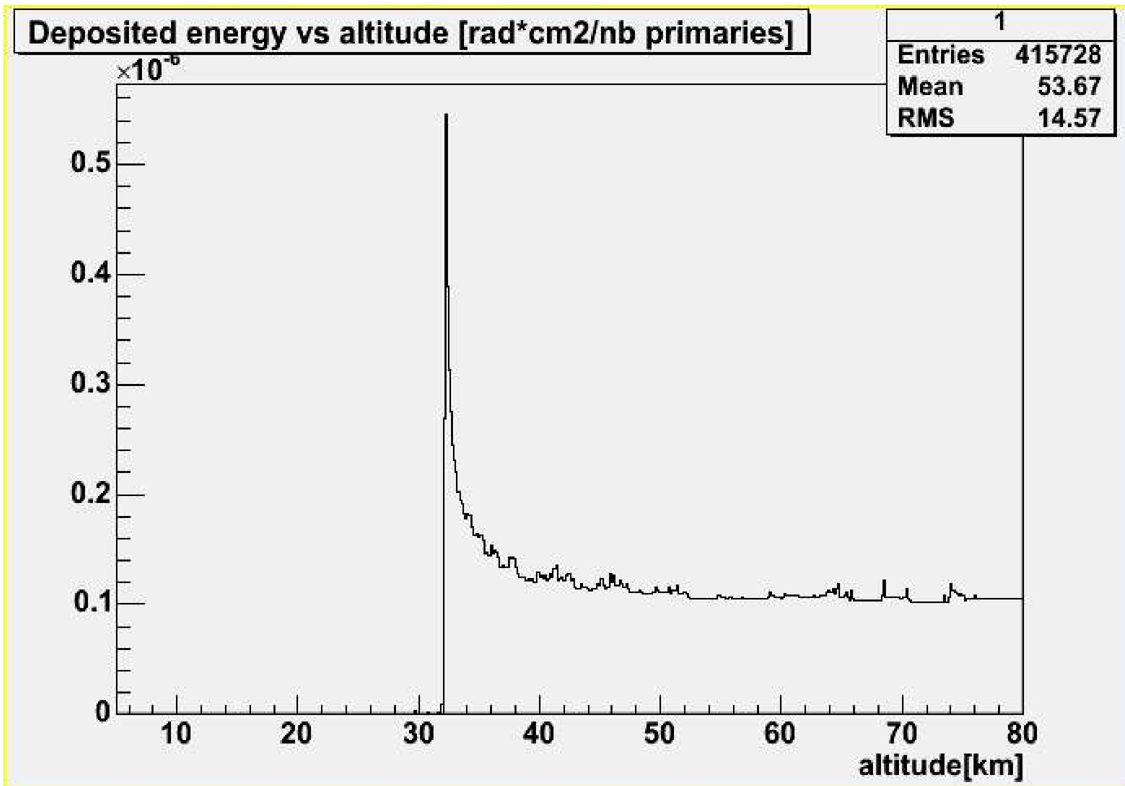

Figure 11Energy deposited by 100 MeV normal incident protons in the atmosphere in function of altitude.

22.8 Example#5 for the Earth

This example is executed by typing “PLANETOCOSMICS Earth earth_example.1g4mac”. It simulates the interaction of galactic cosmic ray protons at solar maximum with the Earth's atmosphere. The upward and downward fluxes of proton, neutron, gamma, positron and electron are detected at different atmospheric depths and altitudes.