Publication Year	2008
Acceptance in OA@INAF	2023-02-23T09:10:09Z
Title	Solar System Mobile Objects Predictions for LS and PMOB
Authors	MARIS, Michele; Dick, Jason
Handle	http://hdl.handle.net/20.500.12386/33783
Number	PL-LFI-OAT-TN-057

TITLE: # Solar System Mobile Objects Predictions for LS and PMOB

DOC. TYPE: TECHNICAL NOTE

PROJECT REF.: PL-LFI-OAT-TN–057 **PAGE:** I of IV, 10

ISSUE/REV.: 0.4 **Ref.:**

Issued by	**Michele Maris** **Jason Dick**	**Date:** 18 December 2008 **Signature:**
Agreed by	**A. ZACCHEI** **LFI DPC Manager**	**Date:** **Signature:**
Approved by	**R.C. BUTLER** **LFI Program Manager**	**Date:** **Signature:**
Approved by	**N. MANDOLESI** **LFI Principal Investigator**	**Date:** **Signature:**

DISTRIBUTION LIST

Recipient	Company / Institute	E-mail address	Sent

CHANGE RECORD

Issue	Date	Sheet	Description of Change	Release
0	2008 Dec 18		Starting of the document	0
0	2009 Feb 16		Equation of motion, official number assigned.	3
0	2009 Feb 20		Fits files described, STEP3 sw released.	4
1			First Release	0

Table of Contents

TITLE: I

DISTRIBUTION LIST ii

CHANGE RECORD iii

1 SCOPE 1

1.1 LIMITS OF APPLICABILITY 1

2 APPLICABLE/REFERENCE DOCUMENTS 2

2.1 APPLICABLE DOCUMENTS 2

2.2 REFERENCE DOCUMENTS 2

2.3 ACRONYMS LIST 4

3 ALGORITHMS 5

3.1 CONVENTIONS, REFERENCE FRAMES, TRANSFORMATION EQUATIONS 6

3.1.1 OBSERVER ECLIPTICAL REFERENCE FRAME 7

3.1.2 SPIN REFERENCE FRAME 7

3.1.3 LINE OF SIGHT REFERENCE FRAME 8

3.1.4 BEAM REFERENCE FRAME 9

3.2 THE OSCULATING REFERENCE FRAME AND THE EQUATION OF MOTION 9

3.3 VISIBILITY CONDITIONS FOR A MOB WITHIN PLANCK F.O.V. AND BEAMS 16

3.3.1 INTERSECTION OF A SCAN CIRCLE WITH THE OSCULATING X,Y PLANE 16

3.3.2 OBJECT INSIDE A CIRCLE 17

3.3.3 OBJECT INSIDE A BAND 18

3.4 COMPRESSING THE EQUATION OF MOTION 21

3.5 FROM TABLES TO PREDICTIONS 22

4 PMOB: Requirements and Architecture 23

4.1 REQUIREMENTS 23

4.2 ARCHITECTURE 23

4.3 LIBRARIES 27

4.4 STRUCTURE OF A MOBS FITS FILE 28

4.4.1 LIMITATIONS TO THE ISSUE 1.0 OF THE MOBS FITS TABLES 30

5 Usage of the Horizon Scanner 31

6 The Dynamical Compressor 34

7 The LS Interface.....35

9 Interfaces.....37

1 SCOPE

During the life-time of the Planck mission a large number of Solar System objects will enter the Planck FOV or main spilloever at epochs which will be function of the Launch data, the effective Planck orbit, and the scanning strategy [RD-1]. In [AD-1] and [AD-2] various options have been illustrated to predict the amount of contamination by these bodies in Planck TODs. These documents emphasizes the need to provide the DPC of a software tool able to predict which samples are contaminated by a Mobile Object. The purpose of this document is to describes the PMOB library whose purpose is to give to the DPC these new capabilities.

PMOB is a package for a massive simulation of Solar System mobile objects in the Planck mission.

The purposes of PMOB are:

1. to provide the LS as a mean to introduce accurate simulations of mobile objects in simulated Planck TODs.
2. to provide a flagging tool to flag the samples contaminated by a mobile object.
3. to provide a planning tool under which to plan the observation of potentially interesting objects.

The most important problem for this application: the reconstruction of the position of objects entering the Planck f.ov. reducing as most as possible the needed information is analyzed in Sect.3.

Sect.4 is devoted to the general architecture of the code which is divided in three modules:

1. Horizons scanner
2. Equation of motion compressor
3. LS module

described respectively in Sect.s 5, 6 and 7.

1.1 LIMITS OF APPLICABILITY

This document refers to Issue 1.0. of the software.

2 APPLICABLE/REFERENCE DOCUMENTS

2.1 APPLICABLE DOCUMENTS

[AD-1] Planck LFI – On Moving Objects Solar System Calculations
M.Maris, C.Burigana
PL-LFI-OAT-TN-034 , Ver. 1.0, 18 Mar 2006

[AD-2] Tracking Solar System Objects in Planck Data
M.Maris, C.Burigana
PL-LFI-OAT-TN-038 , Ver. 1.0, 28 Oct 2006

[AD-3] *PLANCK Parameter Definition Document*
R.Ansari, et al.
2007-03-12, Ver. 0.8.1, PL-COM-IAS-SD-L2.02.005

[AD-4] *Planck/LFI: Main Beam Locations and Polarization Alignment for the LFI Baseline FPU*
M. Sandri, F. Villa
July 2001, Issue 1.0, PL-LFI-PST-TN-027

2.2 REFERENCE DOCUMENTS

[RD-1] *Asteroid detection at millimetric wavelengths with the PLANCK survey*
G. Cremonese, F. Marzari, C. Burigana, M. Maris
New Astronomy, 2002, 7, 483

[RD-2] Horizons Manual
Release 2005 January04
ftp://ssd.jpl.nasa.gov/pub/ssd/Horizons_doc.pdf

[RD-3] *On the Transits of Solar System Objects in the Forthcoming PLANCK Mission: Data Flagging and Coeval Multifrequency Observations*
M. Maris, C. Burigana
2009 Feb 03, arXiv:0902.0468

[RD-4] *PLANCK Telescope Design Specification*
August 2000, SCI-PT-RS-07024

[RD-5] *LFI Polarization: Parameters Definition and Instrument Sensitivity*
December 2001, Issue 1.0 , PL-LFI-PST-TN-031

2.3 ACRONYMS LIST

B-RF	Beam Ref.F.
CSV	Comma Separated Values
DBMS	Data Base Management System
DPC	Data Processing Center
FD	Flight Dynamics
FM	Flight Model
FOV	Field of View
FS	Flight Simulator
JPL	Jet Propulsion Laboratory
KBO	Kuipert Belt Objects
L2	Second Lagrangian Point of the Sun-Earth System
LOS	Line of Sight
LOS-RF	LOS Ref.F.
LS	Level S
MBA	Main Belt Asteroids
OBT	On Board Time
OSC-RF	Osculating Ref.F.
Ref.F.	Reference Frame
S-RF	Spin Ref.F.
SN	Signal/Noise ratio
SS	Solar System
SSO(s)	Solar System Object(s)
TBC	To Be Confirmed
TBD	To Be Defined
TOD	Time Ordered Data
TOD	Time Ordered Data
TOI	Time Ordered Information
MOB	Moving Object
MOBs	Plural of MOB

3 ALGORITHMS

This section contains a description of those algorithms used by the package which needs a more deep explanation than the one contained in the comments within the program.

One of the most complex task is reconstruction of the motion of SS MOBs in the Planck f.o.v.. In this case we want to determine which objects enters the Planck f.o.v. or a given beam, at each pointing period. We need also to know how long the object stay in the f.o.v. or the beam, and what is the angular distance from each beam.

The simplest solution would be to store the positions of the all of the MOBs computed at the time of observation. Despite simple this is not practical given

I.) there is not a priory a perfect knowledge of the epoch in which each sample will be taken and of its exact pointing direction.

II.) II) the position may change significantly during a pointing period, especially with respect to the HFI beams and/or when fast objects, as comets are considered, so a large number of points for each object will have to be stored.

Best is then

1. to determine for each pointing period what are the objects entering the Planck FOV;
2. find a suitable approximation for the equation of motion of that objects during the pointing period;
3. store the parameters of the equation of motion;
4. use that parameters to recompute when needed the accurate position of the MOBs with respect to the PLANCK beams.

Given pointing periods will last for at most 1 hour, it is possible to keep constant the heliocentric distance as well as the distance from Planck of the object of interest. So it is sufficient to parametrize the equation of motion in therms of angular motion of the object on the sky sphere.

After recalling the various conventions and reference frames used in solving this problem (Sect.3.2) we will analyse a suitable approximation to the equation of motion in Sect.3.3, the problem of determining whether a MOB enters at least partially a Planck beam or f.o.v. in Sect.3.4, the problem of compressing the equation of motion in Sect.3.5.

3.1 CONVENTIONS, REFERENCE FRAMES, TRANSFORMATION EQUATIONS

In this section we will take the following conventions:

1. $\min(a_1,a_2,a_3,\cdots,a_n)$ and $\max(a_1,a_2,a_3,\cdots,a_n)$ are the smallest and the greatest element of the not ordered list of *n* elements $(a_1,a_2,a_3,\cdots,a_n)$.
2. $\boldsymbol{V}$ denotes a vector of components (V_x,V_y,V_z) or, with an alternative notation, $(V^{(x)},V^{(y)},V^{(z)})$.
3. $\hat{\boldsymbol{V}}$ denotes the versor of vector $\boldsymbol{V}$
4. $\hat{\boldsymbol{V}}=\mathrm{versor}(\boldsymbol{V})$ is the operator converting a vector in its versor
5. $\boldsymbol{A}^t$ denotes transpose of $\boldsymbol{A}$
6. $\boldsymbol{A}\times\boldsymbol{B}$ denotes an external product
7. $\boldsymbol{A}\cdot\boldsymbol{B}$ denotes an internal product
8. $\mathrm{atan2}(y,x)$ is the usual FORTRAN arctan(y,x) function returning the angle of the (x,y) vector in the range [-180°,+180°].
9. $\boldsymbol{M}_{b,a}$ represents the transformation matrix from a Ref.F. *a* to *b*. The inverse transforms from *b* to *a* is $\boldsymbol{M}_{a,b}$.
10. $R_x(\varphi),R_y(\varphi),R_z(\varphi)$ are rotations of an angle φ about the X, Y or Z axis respectivelly. The corresponding rotation matrices are

$$\boldsymbol{R}_x(\varphi)=\begin{bmatrix}1 & 0 & 0\\ 0 & \cos\varphi & -\sin\varphi\\ 0 & \sin\varphi & \cos\varphi\end{bmatrix}$$

$$\boldsymbol{R}_y(\varphi)=\begin{bmatrix}\cos\varphi & 0 & -\sin\varphi\\ 0 & 1 & 0\\ \sin\varphi & 0 & \cos\varphi\end{bmatrix}$$

$$\boldsymbol{R}_z(\varphi)=\begin{bmatrix}\cos\varphi & -\sin\varphi & 0\\ \sin\varphi & \cos\varphi & 0\\ 0 & 0 & 1\end{bmatrix}$$

The program uses different reference frames which are described in the remainder of the section with the relevant transform equations.

3.1.1 OBSERVER ECLIPTICAL REFERENCE FRAME

The Observer Ecliptical Reference Frame (OE-RF) is a Ref.F. whose origin is the Observer (in this case PLANCK) with X, Y and Z axis aligned with the Heliocentric Ecliptical Reference Frame.

3.1.2 SPIN REFERENCE FRAME

The Spin Reference Frame (S-RF) is the Ref.F. whose origin is the Observer and whose X axis is the Spin axis[1], which is always about opposed to the Sun. The Y axis is parallel to the ecliptic (is on the ecliptic if the spin axis is on the ecliptic) and it is oriented along the direction of Spin drift. The Z axis is normal to the X,Y plane and oriented toward the North Ecliptic Pole (NEP).

If $\hat{\boldsymbol{S}}$ denotes the spin axis its ecliptical longitude and latitude is denoted as (l_{spin}, b_{spin}) .

The conversion matrix from S-RF to OE-RF is

$$\boldsymbol{M}_{ecl,spin} = \boldsymbol{R}_{\boldsymbol{z}}(-l_{spin})\,\boldsymbol{R}_{\boldsymbol{y}}(-b_{spin}) \quad ;$$

its inverse is

$$\boldsymbol{M}_{spin,ecl} = \boldsymbol{R}_{\boldsymbol{y}}(b_{spin})\,\boldsymbol{R}_{\boldsymbol{z}}(l_{spin}) \quad .$$

Versors of the $\hat{\boldsymbol{X}}, \hat{\boldsymbol{Y}}, \hat{\boldsymbol{Z}}$ axis in the OE-RF are

$$\boldsymbol{X}_s = \begin{pmatrix} \cos l_{spin} \cos b_{spin} \\ \sin l_{spin} \cos b_{spin} \\ \sin b_{spin} \end{pmatrix}, \boldsymbol{Y}_s = \begin{pmatrix} -\sin l_{spin} \\ \cos l_{spin} \\ 0 \end{pmatrix}, \boldsymbol{Z}_s = \begin{pmatrix} -\cos l_{spin} \sin b_{spin} \\ -\sin l_{spin} \sin b_{spin} \\ \cos b_{spin} \end{pmatrix}$$

A sample on a scan circle circle is defined by the coordinates (ϕ,β) as defined in Fig.3.2.1.

[1] In some other applications within the Planck pipeline the spin axis is taken as a definition of *Z*.

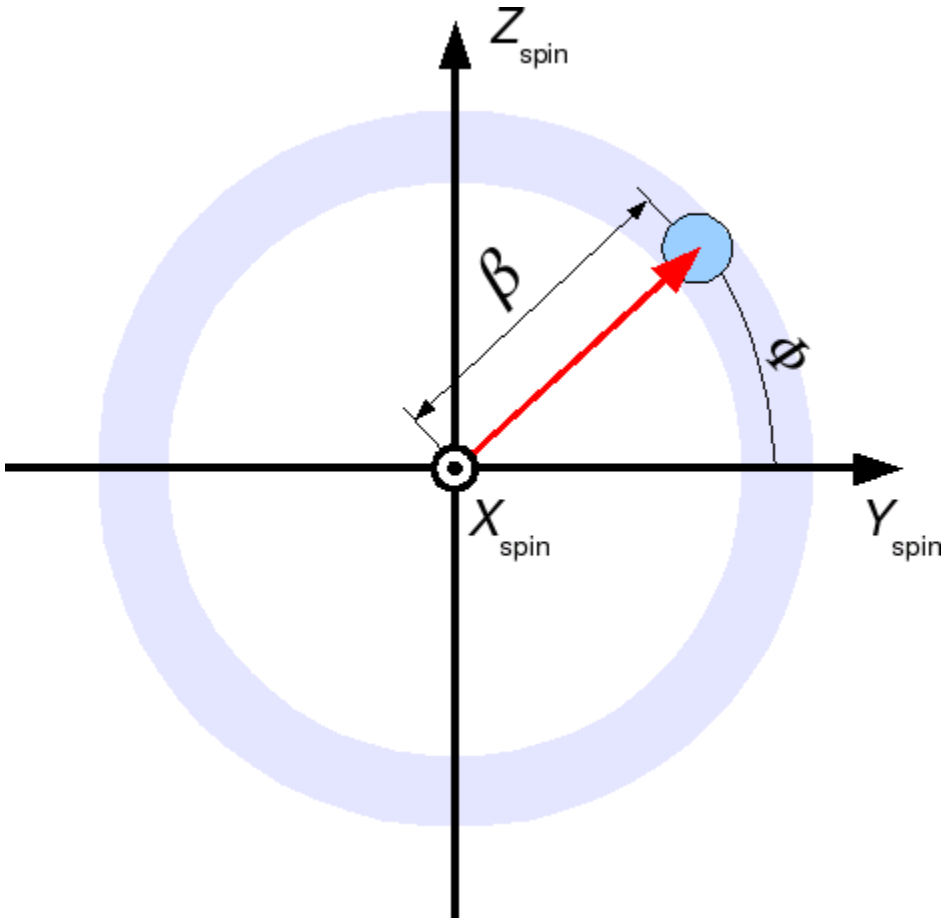

Fig. 3.2.1 The Spin Reference Frame, in the (Y,Z) projection. The gray band is the scan-circle. The blue spot a sample on the scan circle.

3.1.3 LINE OF SIGHT REFERENCE FRAME

The Line of Sight (LOS) is the direction toward which the telescope optical axis is directed.

The Line of Sight Reference Frame (LOS-RF) is the Ref.F. whose Z axis is the telescope pointing direction. The X axis is the axis belongs at the plane defined by Z and the Spin axis. The Y axis is defined as

$$\hat{\boldsymbol{Y}}_{los} = \hat{\boldsymbol{Z}}_{los} \times \hat{\boldsymbol{X}}_{los}$$

The LOS Coordinates in the S-RF are (β, ϕ_{los}) where

$$\beta = \arccos\left(\hat{\boldsymbol{S}} \cdot \hat{\boldsymbol{Z}}_{los}\right)$$

is the angle between the spin and the pointing direction, ϕ_{los} is the phase of rotation with respect to the $\hat{\boldsymbol{X}}_{spin}$ axis.

3.1.4 BEAM REFERENCE FRAME

The Beam Reference Frame (B-RF) is the Ref.F. with Z axis the direction toward which the beam is pointing.

In the LOS-RF the beam has coordinates $(\varphi_{beam}, \theta_{beam})$ which defines in the LOS-RF the longitude and colatitude of the beam pointing direction $\hat{\boldsymbol{Z}}_{beam}$.

A third coordinate ψ_{beam} defines the orientation of the polarization ellipse of the beam in the B-RF.

The passage from the B-RF to the LOS-RF is [AD-4]

$$\boldsymbol{M}_{los,beam} = \boldsymbol{R}_z(90 - \varphi_{beam})\boldsymbol{R}_x(\theta_{beam})\boldsymbol{R}_z(-\varphi_{beam}) \quad .$$

3.2 THE OSCULATING REFERENCE FRAME AND THE EQUATION OF MOTION

Objects in the sky are represented by status vectors $\hat{\boldsymbol{O}}_{obj}$ defining the direction in which the object is seen, if the object moves it changes its status in time. In general, having to do with short intervals of time, $[t_0, t_1]$ it is possible to approximate the equation of motion with an osculating trajectory which allows interpolation of $\hat{\boldsymbol{O}}_{obj}$ at $t_0 \le t \le t_1$. Here the osculating trajectory is

assumed to be an arch major circle drawn about a rotation axis $\hat{\boldsymbol{R}}_{obj}$. Both $\hat{\boldsymbol{R}}_{obj}$ and the rate of rotation ω_{obj} are assumed to be constant along the time interval $[t_0, t_1]$.

Denoting with $\hat{\boldsymbol{O}}_{obj,0}$ and $\hat{\boldsymbol{O}}_{obj,1}$ the positions of the the object at times t_0 and t_1 , the arch of major circle drawn from $\hat{\boldsymbol{O}}_{obj,0}$ to $\hat{\boldsymbol{O}}_{obj,1}$ covers an angle $\Delta\lambda_{obj}$ defined as

$$\cos\Delta\lambda_{obj}=\hat{\boldsymbol{O}}_{obj,1}\cdot\hat{\boldsymbol{O}}_{obj,0} \quad ,$$

the intermediated positions are computed by keeping the same rotation axis and supposing the angle $\Delta\lambda_{obj}$ to vary linearly in time.

The osculating rotation axis $\hat{\boldsymbol{R}}_{osc,obj}$ and the spin axis $\hat{\boldsymbol{S}}$ defines an Osculating Reference Frame (OSC-RF) with Z axis given by $\hat{\boldsymbol{R}}_{osc,obj}$, X axis in the plane defined by $\hat{\boldsymbol{R}}_{osc,obj}, \hat{\boldsymbol{S}}$, and Y axis complementar at the two. Hence the definition of the OSC-RF is

$$\hat{\boldsymbol{R}}_{osc,obj}=\mathrm{versor}(\hat{\boldsymbol{O}}_{obj,0}\times\hat{\boldsymbol{O}}_{obj,1}) \quad ,$$
$$\hat{\boldsymbol{Y}}_{osc,obj}=\mathrm{versor}(\hat{\boldsymbol{R}}_{osc,obj}\times\hat{\boldsymbol{S}}_{spin}) \quad ,$$
$$\hat{\boldsymbol{X}}_{osc,obj}=\hat{\boldsymbol{Y}}_{osc,obj}\times\hat{\boldsymbol{R}}_{osc,obj} \quad ,$$
$$\|\hat{\boldsymbol{S}}_{obj,0}\times\hat{\boldsymbol{S}}_{obj,1}\|=\sqrt{1-\cos^2\Delta\lambda_{obj}} \quad ,$$

with the convention that if the object is not moving or if $\hat{\boldsymbol{R}}_{osc,obj}\times\hat{\boldsymbol{S}}_{spin}=0$, it is assumed the OSC-RF to coincide with the S-RF.

Components of these vectors are denoted as:

$$\hat{\boldsymbol{R}}_{osc,obj}=\begin{pmatrix}\hat{R}^{(x)}_{osc,obj}\\ \hat{R}^{(y)}_{osc,obj}\\ \hat{R}^{(z)}_{osc,obj}\end{pmatrix} \quad , \quad \hat{\boldsymbol{X}}_{osc,obj}=\begin{pmatrix}\hat{X}^{(x)}_{osc,obj}\\ \hat{X}^{(y)}_{osc,obj}\\ \hat{X}^{(z)}_{osc,obj}\end{pmatrix} \quad , \quad \hat{\boldsymbol{Y}}_{osc,obj}=\begin{pmatrix}\hat{Y}^{(x)}_{osc,obj}\\ \hat{Y}^{(y)}_{osc,obj}\\ \hat{Y}^{(z)}_{osc,obj}\end{pmatrix} \quad .$$

The matrix representing the rotation from the OSC-RF to the S-RF is a column matrix

$$\boldsymbol{M}_{spin,osc}=\left(\hat{\boldsymbol{X}}_{osc,obj},\hat{\boldsymbol{Y}}_{osc,obj},\hat{\boldsymbol{R}}_{osc,obj}\right) \quad .$$

Its inverse is the raw matrix

$$\boldsymbol{M}_{osc,spin}=\begin{pmatrix}\widehat{\boldsymbol{X}}^{t}_{osc,obj}\\ \widehat{\boldsymbol{Y}}^{t}_{osc,obj}\\ \widehat{\boldsymbol{R}}^{t}_{osc,obj}\end{pmatrix} .$$

In general the OSC-RF is determined starting from object positions in Spin Ref.F., in this case $\widehat{\boldsymbol{S}}_{spin}=(1,0,0)^{t}$ and so

$$\widehat{\boldsymbol{X}}_{osc,obj}=\begin{pmatrix}(\widehat{R}^{(y)}_{osc,obj})^{2}+(\widehat{R}^{(z)}_{osc,obj})^{2}\\ -\widehat{R}^{(x)}_{osc,obj}\widehat{R}^{(y)}_{osc,obj}\\ -\widehat{R}^{(x)}_{osc,obj}\widehat{R}^{(z)}_{osc,obj}\end{pmatrix} , \quad \widehat{\boldsymbol{Y}}_{osc,obj}=\begin{pmatrix}0\\ \widehat{R}^{(z)}_{osc,obj}\\ -\widehat{R}^{(y)}_{osc,obj}\end{pmatrix} .$$

The position of an object in the OSC-RF is defined by the object osculating longitude and latitude $(\lambda_{obj},\Theta_{obj})$ taken with respect to the X, Y

1. $\widehat{\boldsymbol{O}}_{obj}=\widehat{\boldsymbol{X}}_{osc,obj}$ then $\lambda_{obj}=0,\Theta_{obj}=0$
2. $\lambda_{obj}<0$ if $y_{obj}<0$ else $\lambda_{obj}\geq 0$
3. $\Theta_{obj}<0$ if $z_{obj}<0$ else $\Theta_{obj}\geq 0$.

It is important to note that:

1. always $\Theta_{obj}\equiv 0$ so λ_{obj} is the only parameter needed to define the object position having $\widehat{\boldsymbol{O}}_{obj}=(\cos\lambda_{obj},\sin\lambda_{obj},0)^{t}$,
2. whatever is the motion of the object in the original Ref.F. always $\lambda_{obj,0}\leq\lambda_{obj,1}$.

At last the equation of motion of an object in the OSC-RF hence is

$$\omega_{obj}=\frac{\lambda_{obj,1}-\lambda_{obj,0}}{t_1-t_0}$$
$$\lambda_{obj}(t)=\omega_{obj}(t-t_0)+\lambda_{obj,0},$$
$$\hat{\boldsymbol{O}}_{osc,obj}(t)=\left(\cos\lambda_{obj}(t),\sin\lambda_{obj}(t),0\right)^t$$

Which could be converted to the Spin Ref.F. and then to the Ecliptic Ref. F. to obtain.

$$\hat{\boldsymbol{O}}_{obj}(t)=\boldsymbol{M}_{spin,osc}\hat{\boldsymbol{O}}_{osc,obj}(t),$$

$$\hat{\boldsymbol{O}}_{ecl,obj}(t)=\boldsymbol{M}_{ecl,osc}\hat{\boldsymbol{O}}_{osc,obj}(t);$$

where

$$\boldsymbol{M}_{ecl,osc}=\boldsymbol{M}_{ecl,spin}\boldsymbol{M}_{spin,osc}$$

At last to recover the time dependent phase of the object with respect to the scan circle is

$$\phi_{obj}(t)=\mathrm{atan2}\left(\hat{O}^{(y)}_{obj},\hat{O}^{(z)}_{obj}\right)$$

while the time dependent angular distance from the spin axis is

$$\beta_{obj}(t)=\arccos\left(\cos\Theta_{spin}\cos\left(\omega_{obj}(t-t_0)+\lambda_0\right)\right)$$

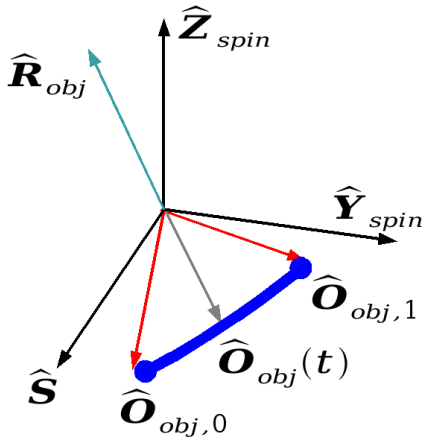

Fig.3.2. The osculating reference frame and the Spin Reference frame. Red are the initial and final positions of the object, gray the position at time *t*, turquoise the rotation axis position.

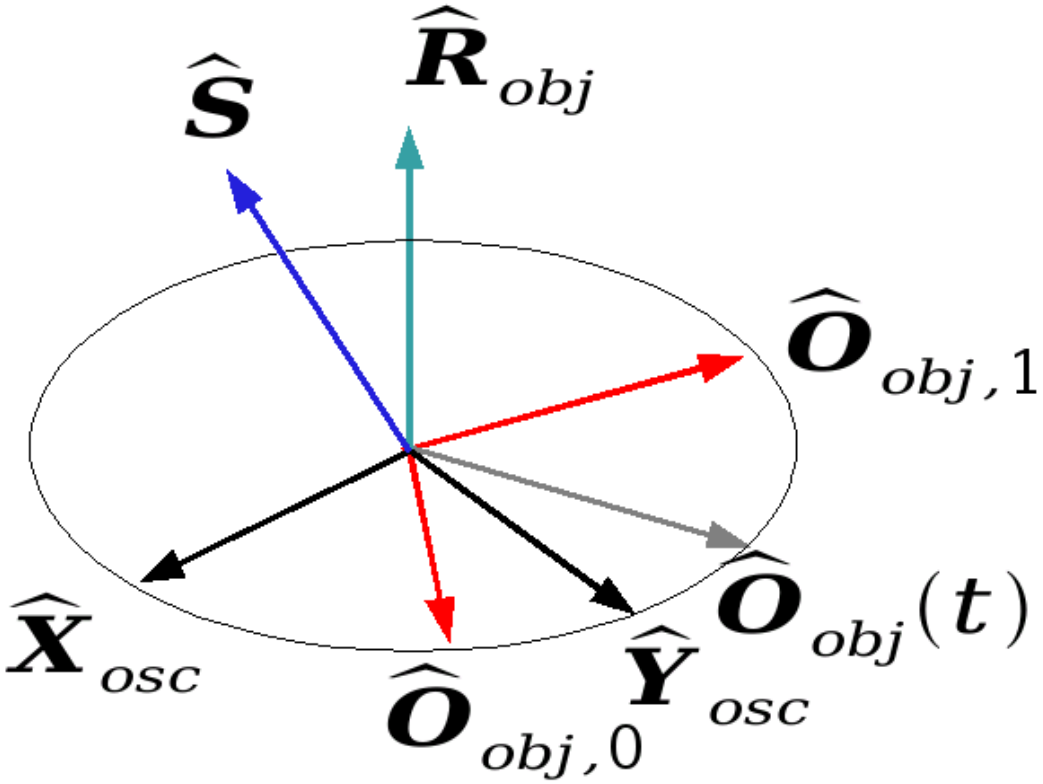

Fig.3.2b. The osculating reference frame. Red are the initial and final positions of the object, gray the position at time *t*, turquoise the rotation axis, blue the spin axis.

3.3 VISIBILITY CONDITIONS FOR A MOB WITHIN PLANCK F.O.V. AND BEAMS

We are interested in determining whether a moving object at a given time enters or not the PLANCK f.o.v. or a given PLANCK beam, or equivalently what is the distance of a beam from an object. Depending on the accuracy this could be approximated determining whether the object enters or not the ring drawn by the PLANCK f.o.v. during the scanning for a given pointing period. Whether the object enters the ring drawn by a particular beam, or whether the object enters a particular sample taken by at a particular time by a given beam. In most cases this is equivalent to solve the problem whether or not the object in its motion crosses a band defined by an inner and an outer ring of angular radii β_{inner} , β_{outer} and drawn around the spin axis. Given in the OSC-RF all the objects has Z=0, the first condition to be fulfilled is that the circle crosses the X,Y, plane.

3.3.1 INTERSECTION OF A SCAN CIRCLE WITH THE OSCULATING X,Y PLANE

We want then to define the intersections of a circle of angular radius β drawn about the spin axis with the X,Y plane of the OSC-RF. In this Ref.F. the spin axis is

$$\hat{\boldsymbol{S}}_{spin,osc}=\begin{pmatrix}\hat{X}_{osc,x,obj}\\0\\\hat{R}_{osc,x,obj}\end{pmatrix},$$

it has null longitude and latitude in the OSC-RF defined by

$$\cos\Theta_{spin}=\hat{S}^{(x)}_{spin,osc}\ .$$

The equation for a circle of radius β drawn around the spin axis is

$$\hat{\boldsymbol{R}}_{\beta}(\zeta)=\boldsymbol{R}_{y}(\Theta_{spin})\begin{pmatrix}\cos\beta\\\sin\beta\sin\zeta\\-\sin\beta\cos\zeta\end{pmatrix}$$

it will intersect the X,Y plane if and only if

$$\hat{S}^{(x)}_{spin,osc} \geq \cos\beta$$

having one intersection if $\hat{S}^{(x)}_{spin,osc} = \cos\beta$ or two in the other cases.

The two intersections are symmetric about the X axis with longitudes $\pm\lambda_{osc,I}$ and

$$\lambda_{osc,I} = \arccos\left(\frac{\cos\beta}{\hat{S}^{(x)}_{spin,osc}}\right) .$$

Given always λ_{osc} increases with time, we will label the intersection with negative longitude the *past intersection*, and the intersection with positive longitude the *future intersection*. The corresponding vectors are

$$\hat{\boldsymbol{I}}_{past} = \left(\cos\lambda_{osc,I}, -\sin\lambda_{osc,I}, 0\right)^t ,$$

$$\hat{\boldsymbol{I}}_{future} = \left(\cos\lambda_{osc,I}, \sin\lambda_{osc,I}, 0\right)^t .$$

After applying the transformation matrix from the OSC-RF to the S-RF

$$\hat{\boldsymbol{I}}^{spin}_{past(future)} = \widehat{\boldsymbol{M}}_{spin,osc}\,\hat{\boldsymbol{I}}_{past(future)} ,$$

the phases of the intersections on the scan circle are defined as

$$\phi_{past(future)} = \mathrm{atan2}\left(\hat{I}^{spin,(z)}_{past(future)}, \hat{I}^{spin,(y)}_{past(future)}\right) .$$

3.3.2 OBJECT INSIDE A CIRCLE

Assuming the circle intersects the X,Y plane, the equation of motion in the OSC-RF allows to determine whether in the time interval $[t_0, t_1]$ the object is inside the circle at least for part of the time taking in account that always $\lambda_{osc,0} \leq \lambda_{osc,1}$.

The condition to have the object INSIDE the circle for at least some time is that both conditions

NOT($\lambda_{osc,1} < -\lambda_{osc,I}$ AND $\lambda_{osc,I} < \lambda_{osc,0}$).

In particular if

$$\lambda_{osc,0} \leq -\lambda_{osc,I} \leq \lambda_{osc,1}$$

the object will cross the *past* intersection and it will enter the circle at time

$$t_{past} = t_0 - \frac{\lambda_{osc,I} + \lambda_{osc,0}}{\omega_{osc}} \quad .$$

If $\lambda_{osc,0} \leq \lambda_{osc,I} \leq \lambda_{osc,1}$ the object will cross the *future* intersection and it will go out from the circle at time

$$t_{future} = t_0 + \frac{\lambda_{osc,I} - \lambda_{osc,0}}{\omega_{osc}} \quad .$$

If $-\lambda_{osc,I} < \lambda_{osc,0} \leq \lambda_{osc,1} < +\lambda_{osc,I}$ the object will stay all the time within the circle.

The interval of time in which the object will be inside the circle, the so called residence time, is

$$\tau_\beta = \min(t_{future}, t_1) - \max(t_{past}, t_0)$$

or zero otherwise.

3.3.3 OBJECT INSIDE A BAND

In the case of a band defined by two circles of radii $\beta_{\text{inner}} < \beta_{\text{outer}}$ things are a bit more complex.

Either:

1. there are no intersection at all if
$$\hat{S}^{(x)}_{spin,osc} < \cos\beta_{\text{outer}}$$
2. there are potentially intersection with the outer circle but not with the inner

$$\hat{S}^{(x)}_{spin,osc} \geq \cos\beta_{\text{outer}} \quad \text{and} \quad \hat{S}^{(x)}_{spin,osc} < \cos\beta_{\text{inner}}$$

3. there are potentially intersection with both circles if

$$\hat{S}^{(x)}_{spin,osc} \geq \cos\beta_{\text{outer}} \quad \text{and} \quad \hat{S}^{(x)}_{spin,osc} \geq \cos\beta_{\text{inner}}$$

The simplest Case 2 reduces at the intersection with a single circle previously discussed. The only change is that now that at the *past* and *future* attributes we will add the *outer* attribute having as an example $t_{\text{outer},future} \geq t_{\text{outer},past}$.

Case 3 is more complex, It is evident as if $-\lambda_{osc,inner} < \lambda_{osc,0} \leq \lambda_{osc,1} < +\lambda_{osc,outer}$ the object will never be in the band. So to have the object in the band or crossing it at same time at least one of the two conditions

$$\text{NOT}(\ \lambda_{osc,1} < -\lambda_{osc,outer} \ \text{OR} \ -\lambda_{osc,inner} < \lambda_{osc,0}\ )$$

$$\text{NOT}(\ \lambda_{osc,1} < \lambda_{osc,inner} \ \text{OR} \ \lambda_{osc,outer} < \lambda_{osc,0}\ )$$

shall be satisfied.

There are also two sub cases to be considered

Case 3a: IF NOT($\lambda_{osc,1} < -\lambda_{osc,outer}$ **OR** $-\lambda_{osc,inner} < \lambda_{osc,0}$ **)**
then the object will reside some time in the *past* side of the band, in particular

1. if $-\lambda_{osc,outer} \leq \lambda_{osc,0} \leq \lambda_{osc,1} \leq -\lambda_{osc,inner}$ the object will stay all the time in the *past* band
2. if $\lambda_{osc,0} \leq -\lambda_{osc,outer} \leq \lambda_{osc,1}$ the object will cross the *outer* band at

$$t_{\text{outer},past} = t_0 - \frac{\lambda_{osc,outer} + \lambda_{osc,0}}{\omega_{osc}}$$

3. $\lambda_{osc,0} \leq -\lambda_{osc,inner} \leq \lambda_{osc,1}$ the object will cross the *inner* band exiting at

$$t_{\text{inner},past} = t_0 - \frac{\lambda_{osc,inner} + \lambda_{osc,0}}{\omega_{osc}}$$

the residence time in the *past* side will be

$$\tau_{\beta,past} = \min(t_{\text{inner},past}, t_1) - \max(t_{\text{outer},past}, t_0)$$

or zero otherwise.

Case 3b: if NOT($\lambda_{osc,1} < \lambda_{osc,inner}$ **OR** $\lambda_{osc,outer} < \lambda_{osc,0}$ **)**
then the object will reside some time in the *future* side of the band, in particular

4. if $\lambda_{osc,inner} \le \lambda_{osc,0} \le \lambda_{osc,1} \le \lambda_{osc,outer}$ the object will stay all the time in the *future* band
5. $\lambda_{osc,0} \le \lambda_{osc,inner} \le \lambda_{osc,1}$ the object will cross the *inner* band at

$$t_{\text{inner},future} = t_0 + \frac{\lambda_{osc,inner} - \lambda_{osc,0}}{\omega_{osc}}$$

6. if $\lambda_{osc,0} \le \lambda_{osc,outer} \le \lambda_{osc,1}$ the object will cross the *outer* band at

$$t_{\text{outer},future} = t_0 + \frac{\lambda_{osc,outer} - \lambda_{osc,0}}{\omega_{osc}}$$

the residence time in the *future* side will be

$$\tau_{\beta,future} = \min(t_{\text{outer},future}, t_1) - \max(t_{\text{inner},future}, t_0)$$

or zero otherwise.

The total residence time will be

$$\tau_\beta = \tau_{\beta,future} + \tau_{\beta,past} \quad .$$

3.4 COMPRESSING THE EQUATION OF MOTION

In order to reconstuct the position of MOBs entering the f.o.v. at a given pointing period it is sufficient to store:

1. the dynamical parameters: $\lambda_{obj,0}$, ω_{obj} and t_0 ;
2. the $\boldsymbol{M}_{ecl,osc}$ or the $\boldsymbol{M}_{spin,osc}$ matrices (depending whether the ecliptical or spin coordinates are needed);
3. the crossing times $(t_{\text{outer},past}, t_{\text{outer},future})$ or the $(t_{\text{outer},past}, t_{\text{inner},past})$ and/or $(t_{\text{inner},future}, t_{\text{outer},future})$.

Given the $\boldsymbol{M}_{ecl,osc}$ or the $\boldsymbol{M}_{spin,osc}$ matrices are 3x3 matrices a factor of 3 of memory space could be saved by storing the corresponding Euler angles defined as

$$\boldsymbol{M}_{spin,osc} = \boldsymbol{R}_z(\varphi_{spin,osc})\boldsymbol{R}_x(\theta_{spin,osc})\boldsymbol{R}_z(\psi_{spin,osc}) \quad ,$$

$$\boldsymbol{M}_{ecl,osc} = \boldsymbol{R}_z(\varphi_{ecl,osc})\boldsymbol{R}_x(\theta_{ecl,osc})\boldsymbol{R}_z(\psi_{ecl,osc}) \quad .$$

but at the cost of a greater computational load.

The angles could be extracted as follow:

1. given

$$\hat{\boldsymbol{Z}} = \boldsymbol{M}\begin{pmatrix}0\\0\\1\end{pmatrix} \quad ;$$

2. then

$$\theta = \arccos(\hat{Z}_z) \quad ;$$

$$\varphi = \text{atan2}(\hat{Z}_y, \hat{Z}_x) \quad ;$$

putting $\varphi = 0$ if $\hat{Z}_x = \hat{Z}_y = 0$.

3. Given

$$\hat{\boldsymbol{X}} = \boldsymbol{R}_x(-\theta)\,\boldsymbol{R}_z(-\varphi)\,\boldsymbol{M}\begin{pmatrix}1\\0\\0\end{pmatrix} \quad ;$$

4. then

$$\psi = \mathrm{atan2}(\hat{X}_y, \hat{X}_x) \quad ;$$

putting $\psi = 0$ if $\hat{X}_x = \hat{X}_y = 0$.

3.5 FROM TABLES TO PREDICTIONS

4 PMOB: REQUIREMENTS AND ARCHITECTURE

4.1 REQUIREMENTS

The software shall be able to predict with astrometric accuracy of 1 arcsec, the instantaneous position of a SSO in the Planck spacecraft reference frame all over the mission.

The software shall be able to predict with an accuracy compatible with the current knowledge the level of contamination introduced by a given body.

The software shall be able to generate alarms for samples contaminated by SSO above a given threshold level.

The threshold level above which an alarm has to be issued is 1 μK.

4.2 ARCHITECTURE

The software is based on the usage of the Horizons email service to determine the position of a set of *relevant* SSO with respect to Planck along the mission.

The software assumes Horizons already holds the Planck orbit and that the Planck orbit will be updated on a regular base as soon as a new orbit is designed.

The software uses the Horizons services to generate a set of precomputed tables for a given orbit, a list of objects and a time interval of interest.

The tables are stored within a dedicated database.

The tables has to be computed only once after an orbit is defined and ingested.

The software is splitted in two packages which has to be used independently:

1. Horizons Scanner
2. MOBs Tables Compiler
3. Observation Predictor

The Horizons Scanner takes a selected list of objects considered relevant, a time interval and time step, and calls *Horizons* to generate the related table of ephemerides, which is subsequently ingested in the data base.

The compiler of MOBs tables extracts the list of objects which enters the Planck FOV for a given scanning strategy by using the output of the Horizons scanner. Each MOBs table contains the information to handle all the objects entering the Planck FOV both from the dynamical and photometrical point of view.

The Observation Predictor uses the database, the pointing list for a given scanning strategy and a thermal model to estimate their emissivity, to determine which objects enter each beam (main beams or main spillover) and estimate whether they are or not relevant.

The general structure of the whole package is illustrated in Fig.1.

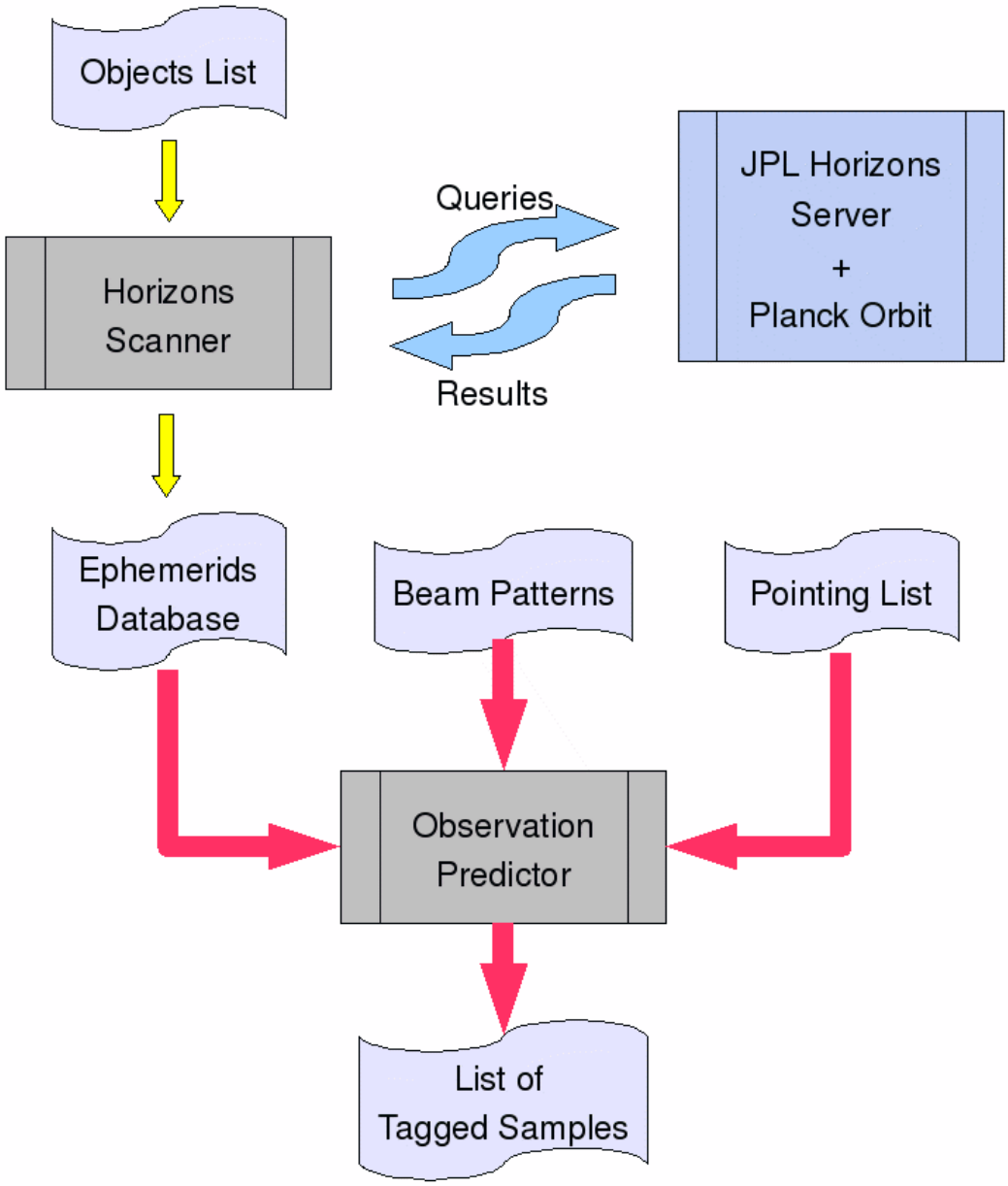
Objects List
Horizons Scanner
Queries
Results
JPL Horizons Server + Planck Orbit
Ephemerids Database
Beam Patterns
Pointing List
Observation Predictor
List of Tagged Samples

Fig. 4.1: Scheme of the package

4.3 LIBRARIES

PMOBs is based on a collection of libraries.

The most important are:

Library	Purpose	Designed for Step	Notes
horizons_scan.pro	Creates an Horizons Database	1	
planck_spin_axis.pro	Simulates a scanning strategy	1, 2	even step 3, to be replaced by PPT file
horizons_edb.pro	handle an Horizons Database	1, 2	
asteroid_photometry.pro	Tant calculations	2	even step 3
lens_plot.pro	Monitoring and Display	2	even step 3
hits.pro	handle crossing of focal plane by objects	2	
fp_database.pro	handle focal plane data base	1, 2	even step 3
mobs_fits.pro	Fits I/O of MOBs tables	2,3	
mobs_motion.pro	Interpolated motion of MOBs from MOBs tables	2,3	
vector_arrays.pro	matrix arrays	all	support library
rot.pro	matrix rotations	all	support library

4.4 STRUCTURE OF A MOBS FITS FILE

A MOBs FITS file contains the parameters for the equation of the motion and photometry information for the objects entering the f.o.v. in a range of spin periods.

A MOBs FITS file is formatted as a binary table FITS file having two extensions:

Extension 1: INDEXING_TABLE
Extension 2: OBJECTS_TABLE

The OBJECTS_TABLE contains the equation of motion and photometry for objects entering the f.o.v. with one row of the table for each spin period and each object in the spin period.

The INDEXING_TABLE contains a table listing, for each spin period, what are the relevant rows in the OBJECTS_TABLE.

The INDEXING_TABLE is provided as a mean to speed up the search through the OBJECTS_TABLE and as a mean to pass all the information which varies with the spin period but does not depend on the object, as an example: the ecliptical coordinates of the spin in the given spin period.

In addition the following rules shall be strictly respected:

1. In both tables ROWS are numbered from 0.
2. I_SPIN is repeated in both tables.
3. I_SPIN shall be kept in growing order in the INDEXING_TABLE.
4. Repeated values of I_SPIN are not allowed in the INDEXING_TABLE
5. For a given I_SPIN repeated values of OBJ_ID in the OBJECTS_TABLE are not allowed.
6. All the rows in the OBJECTS_TABLE having the same I_SPIN shall be strictly consecutive. No rows with different values of I_SPIN or negative I_SPIN shall be interposed.
7. The OBJ_ID is the code to identify the object and corresponds to the JPL code for a Solar System Object.
8. I_SPIN and OBJ_ID shall be positive numbers.

9. Negative values for I_SPIN and OBJ_ID are not valid values. The programs reading these tables when finding I_SPIN < 0 and/or OBJ_ID < 0 shall consider them not valid values and skip them. Negative values are used as a way to reserve space in tables.

So as an example, let be to have two spin periods: 1000 and 1001 with 4 objects in the former and 5 in the latter. The INDEXING_TABLE will be as follow:

INDEXING_TABLE

	I_SPIN	OTHER COLUMNS	FIRST_ROW	LAST_ROW	OTHER COLUMNS
0	1000	...	0	3	...
1	1001	...	4	8	...
2	-1	...	-1	-1	...

The corresponding OBJECTS_TABLE will appear as follow:

OBJECTS_TABLE

	I_SPIN	OTHER COLUMNS	OBJ_ID	OTHER COLUMNS
0	1000	...	10	...
1	1000	...	211	...
2	1000	...	10012	...
3	1000	...	2213	...
4	1001	...	210	...
5	1001	...	122	...
6	1001	...	223	...
7	1001	...	444	...
8	1001	...	333	...
9	-1	...	-1	...

Note the usage of negative I_SPIN and OBJ_ID to denote not-used lines.

4.4.1 LIMITATIONS TO THE ISSUE 1.0 OF THE MOBS FITS TABLES

In the current (1.0) release, due to limitations in the IDL part it is not possible to add more than 1 spin period per file.

So the program will generate 1 fits file for spin period.

5 USAGE OF THE HORIZON SCANNER

The Horizons scanner needs:

IDL 6.0 or higher
ASTROLIB
perl 5.0 or higher
a dedicated mail account on a mail server accessible via POP3 and SMTP

It is IMPORTANT to recall that the Anti-Spam of the e-mail account shall be switched off or it shall be instructed at accept any email from the Horizons mail server:

horizons@ssd.jpl.nasa.gov

The Horizons scanner communicates with Horizons through the Horizons EMAIL interfaces [RD-2].

The Horizons Scanner is implemented as an IDL 7.0 code. It needs Perl 5.0 or higher to handle email interfaces, and it is designed to work under LINUX [2].

The details of each routine are reported in the automaticaly generated documentation.

The user has to provide a project description file.

I.e. a file where the following informations are provided:

1. A name for the project.
2. A directory where to store the files produced by the simulation.
3. A time interval and time step.
4. Configuration parameters for a POP3 and SMTP servero used by the POP3 and SMTP client embedded in the software.
5. A valid account on the email client used to send and receive the eamils.

We strongly suggests to define a dedicated account for this application.
While executing the application the account could be not accessed by any other mail service or client able to delete and / or modify the email content. In addition due to limitations in the

[2]The operativity under other Oss, such as: Windows or MAC/OS is not guaranteed and it is not supported at all.

Horizons software it is not possible to operate concurrently on the same mail server with different projects.

In addition the user has to provide a list of MAJOR BODIES and a list of SMALL BODIES which has to be scanned.

The default lists are stored in the StandardLists directories within the software distribution and are automatically loaded at the time of creation of a new project.

After edited the project file, idl plus the atrolib support has to be started, the horizons_scan.pro has to be compiled (at least twice to resolve inner compilation conflicts) and the project description file has to be runned.

IDL will generate a directory containing the project file as all the related files and the data base of tabels produced by Horizons.

Tables are genrerated running

IDL> horizons_scan,*PROJECT_PATH, KIND_OF_OBJECT*',/new

An example of project description is reported in the project_dump.pro example file.
In that case the project directory will be named: *DUMP*. Then typing

IDL> horizons_scan,'DUMP', 'MAJOR',/new

the tables of MAJOR BODIES (planets) will be generated and stored into DUMP/TABLES. While typing

IDL> horizons_scan,'DUMP', 'SMALL',/new

the minor tables for the minor bodies will be generated and stored into DUMP/TABLES.

Note that many days could be required to generate a complete scan. If the progrram is stopped for some reason. it is possible to restart it by typing

IDL> horizons_scan,'DUMP', 'MAJOR'

or

IDL> horizons_scan,'DUMP', 'SMALL',

respectively.

6 THE DYNAMICAL COMPRESSOR

The observation predictor

7 THE LS INTERFACE

8

9 INTERFACES

Shall we interface to the LS?

Better to use multimod? Which coordinates uses? Ecliptical or spin? Ask Martin Reinecker

Shall we interface to the MapMaker?

File format

Three classes of objects
Planets

Asteroids

Comets

Fits file object list

Header:

Bmin = lowest ecliptical longitude
Bmax = highest ecliptical longitude

Indexing block:
spin_period, first_row,last_row,ut0,ut1, Bmin, Bmax, ell1, ell2, ell3, ell4

Body block:
For each spin period a bloc of lines one for object
spin_period,leading_or_trailing_flag,obj_id,obj_name,lambda0,omega,t0,tend,R_ecl_phi,R_ecl_theta,R_ecl_psi,R_spin_phi,R_spin_theta,R_spin_psi
,TantP030,TantP044,TantP070,TantP100,TantP143,TantP217,TantP353,TantP545,TantP857

Global bounds To select the samples are defined by Bmin, Bmax, ell1, ell2, ell3, ell4
(ell1,ell2) ell1 < ell2 trailing side limits
(ell3,ell4) ell3 < ell4 leading side limits

An algorithm to filter relevant objects *I.e. to compute single object bounds in the spin period*
Compute starting, ending positions of the object

take these positions after shifting a safety boundary = once or twice the FWHM of the beam
PARAMETER:
NBEAMS_BORDER = 1 or 2 is the number of FWHM to define the safety region

Algorithm:
possibility one

Compute ***O***0, ***O***1 then accept the object if
O0x-NF <= *Bx* <= *O0x+NF*
O0y-NF <= *By* <= *O0y+NF*

possibility two: better

B . Oaverage >= CosTh

the following are computed while loading the file
CosTh = cos(NBEAMS_BORDER*FWHM+Displacement)
Oaverage = (**O**0+ **O**1)/2
Displacement = Omega * (t1-t0)

O(t) = **M**(phi,theta,psi) (cos(displacement(t)),sin(displacement(t)),0)
displacement = lambda_0 + omega*(t-t0)

An algorithm to compute instantaneous position and distance *D* from a given sample *S* after selection

An algorithm to compute average position and distance *D* from a given sample *S*

Compute the istantaneous flux

Order 1 is: TantFreq = TantFreq0