A numerical code for the evaluation of external straylight signal from Solar System main bodies (straylight_SS_v1e0.f)

Carlo Burigana
INAF-IASF Bologna Internal Report 517/2008
March 2008

Summary – This reports describes the numerical code (**straylight_SS_v1e0.f**) for the computation of the external straylight contamination on Planck data from Solar System objects, as provided to the LFI DPC to be integrated in the Level S (simulation) (LS) pipeline. The code assumes to ingest body positions from other LS codes and then, given a suitable input set including parameters and database such as the information on considered beam and frequency, computes the straylight signal. In the current version it can be used i) at any frequency for computing the signal from Sun, Earth, and Moon; ii) at 30, 44, 70, 100 GHz for all the other Solar System main bodies with reasonable - i.e. based on observation results – assumptions; iii) formally, also at higher frequencies where it could be easily updated assigning more reasonable - i.e. based on observations or on reasonable models -temperatures to Solar System main bodies at higher Planck frequencies.

1. Introduction

The external straylight contamination on Planck data from Solar System objects is a potential source of systematic effect. In the context of current Planck simulation activity it is important to appropriately model and possibly remove this effect.

In this report, I describe the main properties of a numerical code (**straylight_SS_v1e0.f**) dedicated to the computation of the external straylight contamination on Planck data from the main Solar System bodies, as provided to the LFI DPC to be integrated in the Level S (simulation) (LS) pipeline. The code assumes to ingest body positions from the results of other LS codes and then, given a suitable input set including parameters and database such as the information on considered beam and frequency, computes the straylight signal. In the current version it can be used i) at any frequency for computing the signal from Sun, Earth, and Moon; ii) at 30, 44, 70, 100 GHz for all the other Solar System main bodies with reasonable - i.e. based on observation results - assumptions; iii) formally, also at higher frequencies where it could be easily updated assigning more reasonable - i.e. based on observations or on reasonable models - temperatures to Solar System main bodies at higher Planck frequencies.

This code has been written in Fortran (F90) by Carlo Burigana @ INAF-IASF Bologna, Italy, during 2007-2008 and it has taken advantage from the work done by Burigana et al. (2000a) aimed at simulating the Sun, Earth, and Moon straylight signal in the context of a home-made simulation pipeline. Here, the main routines are rewritten to be easily integrated in the current Planck Level S (simulation) pipeline.

The reader can refer also to Burigana et al. (2000a,b) and to Burigana et al. (2001) for a discussion of the fundamental concepts and a first presentation of the main results of this analysis and to a forthcoming report by Galeotta & Burigana (2008) for further descriptions of the code version after the integration into LFI simulation pipeline infrastructure.

2. Main routines

The routines of the code can be divided into categories:

A) routines that need to be called only a single time at the beginning to select the bodies of interest, set the relevant values and read input data;

B) routines that need to be called for each sample of interest since their results obviously depend on the geometry of spacecraft and bodies at each considered instant.

2.1 Routines called only a single time

2.1.1 Subroutine list_sol_syst_read

Read the input parameters for choosing the considered Solar System objects numbered according to:
0: Planck
1: Sun
2: Mercury
3: Venus
etc …
10: Pluto
11: Moon

The vector of flags list_sol_syst controls the routine output:
0 no output for that body;
1 only coordinates (originally, barycentric);
2 coordinates and velocity components (originally, barycentric).

In reality, no particular choice for reference system is needed in this version but it is obviously crucial that all coordinates (both of Solar System bodies and of spacecraft) are in the same reference. Velocities will be not used here, so it is possible to select only 1 or 0 respectively if one is interested or not in that object.

The block
common/object_used/list_sol_syst
gives this choice to the other routines where it is relevant.

2.1.2 Subroutine prepri_temperatures(freq_ghz)

It simply calls the subsequent subroutine, given the considered frequency expressed in GHz.

2.1.3 Subroutine intrinsic_Ta_mK_read(freq_ghz)

It gives the intrinsic antenna temperature of Solar System objects in mK at the considered frequency expressed in GHz.

For Sun, Earth, and Moon it starts from the intrinsic temperature and it converts their temperatures (assumed to be respectively 6000, 300, and 250 K) from thermodynamic to antenna temperatures.

Estimates/approximated measures for Mercury and Venus are based on Table 2 of Greve et al. (1994). For Mercury large phase variability effect exist, so the average of minimum and maximum temperature is used here. For Venus the average of the values found at 43 GHz and at 31.4 GHz is used here since they are well consistent within errors.

For Pluto, an estimate is provided by the IRAM measure at mm wavelengths, consistent with the IRAS 60 and 100 micron results extrapolated deep into surface reported in McFadden et al. (2007).
For external planets, it has been possible in some cases to include the WMAP 5yr results (2008). For Jupiter, linear interpolation (or extrapolation in the case of the 100 GHz channel) of WMAP 5yr results have applied at 30, 44, 70, 100 GHz. For Saturn and Mars, WMAP <W> 5yr results have been properly used at 100 GHz and also, less properly, at other channels as rough estimates.
All these choices can be easily changed and possibly updated in the presence of better estimates. Note that in principle, particularly for inner bodies, the intrinsic antenna temperature depends on the considered instant, since we are interested here in the body temperature under very different geometrical configurations. Fortunately, the straylight signal expected by these bodies is very low and a very accurate computation including this effect is not so interested in practice in this context.

2.1.4 Subroutine read_case1_beams(name_file_beam,minimum_db_belowpeak)

This is the routine that reads far (full - 4π) beams simulated by GRASP and re-organize this information in a matrix in polar coordinates (expressed in rad) that can be easily interpolated to give the beam response at any desired direction. It assumes a given structure for the beam file (name_file_beam), appropriate to LFI beams as delivered by Sandri (2003). A step of 2 deg in both polar coordinates is assumed here. In addition it sets to minimum_db_belowpeak the response in the regions where no contribution are obtained with the assumed optical treatment. minimum_db_belowpeak could be set, for example, to the requirement adopted at the considered frequency channel for straylight rejection.
Note that the beam response is here re-normalized at the beam centre (it is set 1 at the beam centre).
Note that commented dimension & common blocks refer to different beam tabulation sizes & conventions during LFI optical optimization studies (long history!): possibly, change/update the current definitions according to different beam tabulations. Note also that the value of the integer*4 parameter i360 depends on the beam grid. For the current beam 4π tabulation we have

```
integer*4 i360
parameter (i360=90).
```

2.2 Routines called at each sample of interest

2.2.1 Subroutine obj_planck_frame(vec_pos, unitvec_planck_obj, dist_planck_obj, angR_planck_obj, solidang_planck_obj, Ta_mK_planck_obj_R1)

Given the vectors (vec_pos), in Cartesian coordinates, of the Planck (object 0) position and Solar System object (objects from 1 to 11) positions computes:
the unit vector of the direction Planck → object (unitvec_planck_obj);
the distance Planck-object (dist_planck_obj);
the angle of the object radius seen by Planck in rad (solidang_planck_obj);
the solid angle of the object seen by Planck in sr (solidang_planck_obj);
the antenna temperature (in mK) of the object at the Planck distance, without taking into account the beam pattern (Ta_mK_planck_obj_R1).
The observed antenna temperature will obtained by multiplying Ta_mK_planck_obj_R1 by the antenna response at the considered direction and by dividing the result by the integrated antenna

response. The temperature is computed only for required objects and is set to 0 for non required objects.

In principle, not particular requirement on assumed coordinate system and distance units is needed. In practice for interface with LS ecliptic coordinates work better. Also, the object radius is here in AU, so object positions/distances should be given in AU, or a proper transformation factor should be included.

Note that, according to the details of the celestial mechanics code used for producing vec_pos, the code outputs could include (or not) the delay in propagation of light, the light aberration and relativistic corrections (Maris & Burigana 2006). Obviously, according to the adopted approximation, input vectors vec_pos need to be computed at the time corresponding to that of the considered data sample.

2.2.2 Subroutine signal_straylight(theta_beam,phi_beam,psi_beam, ant_pattern_integral_max1_sr, FWHM_arcmin, unitvec_planck_obj, Ta_mK_planck_obj_R1, Ta_mK_planck_obj)

Given:
the unit vector of the direction Planck → object (unitvec_planck_obj);
the beam direction and orientation (theta_beam,phi_beam,psi_beam);
the full beam solid angle (given by ant_pattern_integral_max1_sr or by FWHM_arcmin**);**
the antenna temperature of the object at the Planck distance without taking into account the beam pattern (Ta_mK_planck_obj_R1);
the subroutine computes:
the observed antenna temperature, obtained by multiplying Ta_mK_planck_obj_R1 by the antenna response at the considered direction and by dividing the result by the integrated antenna response.

The temperature is computed only for required objects and is set to 0 for not required objects.

For the full beam integrated antenna response it uses the parameter
ant_pattern_integral_max1_sr
expressd in sr, if it is known by integrating the beam tabulations (jointly using main, intermediate, and far - 4π – beam tabulation)
otherwise it uses the parameter
FWHM_arcmin
which can be used to provide a reasonable estimates of ant_pattern_integral_max1_sr (with < few % accuracy, given typical sidelobe contributions).

The signal is computed only for required objects and is set to 0 for non required objects.

In principle, not particular requirement on assumed coordinate system and distance units is needed. In practice for interface with LS ecliptic coordinates work better. Also, the object radius is here in AU, so object positions/distances should be given in AU, or a proper transformation factor should be included.

The beam direction and orientation at the considered sample is given (by LS) in terms of:
theta_beam = colatitude of the beam centre direction
phi_beam = longitude of the beam centre direction
psi_beam = "inclination" of the beam u (or x, positively oriented) axis, i.e. angle counted anti-clock-wisely as seen externally to the sphere from the meridian identified by the longitude of the beam centre direction oriented from South to North and to the beam u (positively oriented) axis.

All unit vectors in the adopted reference frame.

2.2.3 Subroutine versors_beam(theta_beam,phi_beam,psi_beam, uorx_beam,vory_beam,z_beam)

Given the angles theta_beam, phi_beam, psi_beam defined in the above subroutine (assumed in rad) it computes the corresponding unit vectors in the same reference frame. See Burigana (2008) for a detailed definition of these angles and unit vectors and the related transformations rules.

This subroutine is used by the above subroutine signal_straylight.

Input:

theta_beam = colatitude of the beam centre direction
phi_beam = longitude of the beam centre direction
psi_beam = "inclination" of the beam u (or x, positively oriented) axis, i.e. angle counted anti-clock-wisely as seen externally to the sphere from the meridian identified by the longitude of the beam centre direction oriented from South to North and to the beam u (positively oriented) axis.

Output:

uorx_beam = unit vector of the u (or x) beam direction;
vory_beam = unit vector of the v (or y) beam direction;
z_beam = unit vector of the beam centre direction.

These vectors are used to compute the object direction in the beam frame.

All unit vectors in the adopted reference frame.

2.2.4 Subroutine bilinear(x,y,z,stat_agreebeam)

Given the geometry at the considered sample interpolate over the beam tabulation to evaluate the beam response (z) at the direction (x=colatitude, y=longitude in the beam frame) of the considered object.

stat_agreebeam is a subroutine check parameter that has to be initially set to 0. In the presence of (possible) problems with beam matrix sizes it is set to 1 and an error (ATTENTION) message is printed.

Bilinear interpolation is used since it is stable in the presence of possible oscillations of tabulated data.

Note that commented dimension & common blocks refer to different beam tabulation sizes & conventions during LFI optical optimization studies (long history!): possibly, change/update the current definitions according to different beam tabulations.

3. Code usage

This set of subroutines can be used in two ways into LS: i) integrating them directly into the LS pipeline to produce signal streams that could be directly added to the other ones; ii) through a proper interface with LS outputs, i.e. reading the outputs from LS relevant for this code and separately producing signal streams to be stored and that could be then added to the other ones. The option ii) is strongly suggested, since it allows to check results, update the code, and manage their outputs in an easier, safer, and practical way to include possible complications (e.g. body intrinsic brightness variability, modifications on beam assumptions, etc.).

The routines need to be called in two phases:

A) calling routines that need to be called only a single time at the beginning to select the bodies of interest, set the relevant values and read input data;

B) calling routines that need to be called for each sample of interest since their results obviously depend on the geometry of spacecraft and bodies at each considered instant, in a loop over the considered samples.

The following call cascade is suggested.

Phase A)

1a. Call routine that reads the (far) 4pi beam by GRASP8 and sets to minimum_db_belowpeak the response in the regions where no contribution are obtained with the assumed treatment.
The input parameter minimum_db_belowpeak need to be previously, for example to –100.
The name of the beam (4π beam) specifies the considered horn and receiver.

call read_case1_beams(name_file_beam,minimum_db_belowpeak)

2a. Choice of the Solar System bodies considered for external straylight simulation.
Remember:
0: Planck - 1: Sun - 2: Mercury - etc - 10: Pluto - 11: Moon
It is based on a list (vector) of 11 parameters
0 means that the body is not considered; 1 means that the body is considered.

call list_sol_syst_read

3a. Call the routine that gives body intrinsic temperature according to selected frequency and existing data.
It requires to specify the input parameter:
freq_ghz
which gives the considered frequency in GHz.

call prepri_temperatures(freq_ghz)

This subroutine will

call intrinsic_Ta_mK_read(freq_ghz)

Possible output:

it could be very useful to print a single time the intrinsic brightness temperature adopted for each body of interest in the current simulation.

Phase B)

Loop over the considered samples.

1b. The code assumes to acquire vec_pos from a database (or from other codes vec_pos) that gives the positions of the various bodies in the adopted reference frame in Cartesian coordinates (coordinate 1 corresponds to x, coordinate 2 corresponds to y, coordinate 3 coresponds to z), as derived from LS.
AU are used here since body radii are in AU in subroutine obj_planck_frame.

The dimension is vec_pos(1:3,0:(nbody-1)). The adopted reference frame is arbitrary, but it need to be obviously the same for all bodies (including Planck). If polar coordinates are used, they need to be converted to Cartesian ones.

2b. Call the routine that computes the body antenna temperature in mK as seen at the maximum beam response - set to 1 - (and other useful informations) given the relevant geometry

```
call obj_planck_frame(vec_pos,
&                         unitvec_planck_obj,
&                         dist_planck_obj,
&                         angR_planck_obj,
&                         solidang_planck_obj,
&                         Ta_mK_planck_obj_R1)
```

3b. The code assumes to acquire beam direction & orientation
theta_beam, phi_beam, psi_beam
from a database or from other codes.
In particular these quantities can be obtained from the output of LS pipeline for each considered beam/receiver at the adopted sampling rate.

4b. The code assumes to have 4π beam solid angle computed assuming to be 1 the response at maximum (ant_pattern_integral_max1_sr).
Alternatively, for simplicity, it can use the parameter FWHM_arcmin.
It uses ant_pattern_integral_max1_sr if it not 0;
if ant_pattern_integral_max1_sr is seto to 0 then it uses FWHM_arcmin
to estimate ant_pattern_integral_max1_sr.

5b. Call to routine that compute the body antenna temperature in mK (and other useful info)
as seen for the given beam response given the relevant geometry and full beam solid angle.

```
call signal_straylight(theta_beam,phi_beam,psi_beam,
&                        ant_pattern_integral_max1_sr,
&                        FWHM_arcmin,
&                        unitvec_planck_obj,
&                        Ta_mK_planck_obj_R1,
&                        Ta_mK_planck_obj)
```

Ta_mK_planck_obj is the code main result and it need to be stored. It is a vector of 11 elements (as the potentially considered Solar System main bodies) sample by sample, i.e. within each loop. Considering Nsamples, it will produces for example a table of 11 columns and Nsamples rows (plus possibly an additional column with the sum of all the considered straylight signals).

Other useful code outputs (to be possible stored with analogous structures) can be:

unitvec_planck_obj, i.e. the unit vector of the direction Planck → object (attention: it is a set of unit vectors, in a Cartesian frame, x,y,z components)
dist_planck_obj, i.e. the distance Planck-object (typically in UA)
angR_planck_obj, i.e. the angle of the object radius seen by Planck (typically in rad)
solidang_planck_obj, i.e. the solid angle of the object seen by Planck (typically in sr)
Ta_mK_planck_obj_R1, i.e. the antenna temperature (in mK) of the object at the Planck distance, without taking into account the beam pattern (the observed antenna temperature will obtained by

multiplying Ta_mK_planck_obj_R1 by the antenna response at the considered direction and by dividing the result by the integrated antenna response).

4. Dimensions and type of variables and parameters and common blocks

The dimensions and types of the variables and parameters that are needed by a program that calls these subroutines are here reported, divided into categories: INPUT (sample independent), INPUT (sample dependent), and OUTPUT (sample dependent).

INPUT (sample independent)

```
real*8 freq_ghz,ant_pattern_integral_max1_sr,FWHM_arcmin,minimum_db_belowpeak
character*128 name_file_beam
```

INPUT (sample dependent)

```
integer*4 nbody
parameter (nbody=12)
real*8 vec_pos(1:3,0:(nbody-1))

real*8 theta_beam,phi_beam,psi_beam
```

OUTPUT (sample dependent)

```
real*8 unitvec_planck_obj(1:3,1:(nbody-1)),dist_planck_obj(1:(nbody-1))
real*8 angR_planck_obj(1:(nbody-1)),solidang_planck_obj(1:(nbody-1))
real*8 Ta_mK_planck_obj_R1(1:(nbody-1)),Ta_mK_planck_obj(1:(nbody-1))
```

4.1 Dimensions and type of variables and parameters of subroutines

The dimensions and types of the variables and parameters of the various subroutines that have not been described above (since they are typically not called by a main program but only by subroutines themselves) are here reported.

In the **subroutine bilinear(x,y,z,stat_agreebeam)**

INPUT (sample dependent)

```
real*8 x,y
```

OUTPUT (sample dependent)

```
real*8 z
```

INPUT/OUTPUT (sample dependent)

```
integer*4 stat_agreebeam
```

In the **subroutine subroutine versors_beam(theta_beam,phi_beam,psi_beam,**
& uorx_beam,vory_beam,z_beam)

OUTPUT (sample dependent)

real*8 uorx_beam(1:3),vory_beam(1:3),z_beam(1:3)

4.2 Common blocks

The dimensions and types of the variables and parameters appearing in common blocks in some subroutines are here reported.

integer*4 nbody
parameter (nbody=12)
integer*4 list_sol_syst(1:(nbody-1))
common/object_used/list_sol_syst

integer*4 nbody
parameter (nbody=12)
real*8 intrin_Ta_mK(1:(nbody-1))
common/temp_obj/intrin_Ta_mK

integer*4 nth,nphi
parameter (nth=91,nphi=181)
real*8 th(0:nth-1),phi(0:nphi-1),db(0:nphi-1,0:nth-1)
common/case1_beamshape/th,phi,db

integer*4 nth_use,nphi_use
common/warning_dim_beam/nth_use,nphi_use

5. Computational time

The required computational time should be not an issue.
For example, to simulate 1 hour (~ spin period) of data at about 10' sampling (30 GHz) (i.e. 2160×60=129600 samples) on my laptop (CPU Intel Pentium M 753, 1.2 GHz), assuming some simple geometries it requires:
about 2 seconds to give results for Sun, Earth & Moon together;
about 5 seconds to give results for all Solar System bodies.
Maybe, a more realistic CPU estimate could be increased by same (≈ unit) factors. Anyway, CPU time should to be very short and certainly not an issue, given Planck simulation pipeline facilities.

5. Conclusion

This reports describes the numerical code designed at the computation of the external straylight contamination on Planck data from Solar System objects, as provided to the LFI DPC to be integrated in the Level S (simulation) (LS) pipeline.

The code assumes to ingest body positions from other LS codes and then, given a suitable input set including parameters and database such as the information on considered beam and frequency, computes the straylight signal.

After a brief introduction to the problem, all the main code aspects are presented:
i) the code subroutines, divided into categories, those that need to be called only a single time at the beginning to select the bodies of interest, set the relevant values and read input data, and those that need to be called for each sample of interest since their results obviously depend on the geometry of spacecraft and bodies at each considered instant;
ii) the code usage, with a suitable sequence of calls to produce scientific results and the description of code input and output;
iii) the dimensions and types of the variables and parameters that are needed by a program that calls these subroutines, by subroutines, and appearing in common blocks;
iv) the computational time needed by the code.
This version of the code (**straylight_SS_v1e0.f**) has been successfully tested on various platforms (my laptop, the spike facility at INAF-IASF Bologna and the grid001 facility at INAF-OATs). Future improvements/updates can be easily included. In particular, more accurate body intrinsic temperatures can be simply updated in the code, when available.

Acknowledgements

I warmly acknowledge useful conversations with F. Perrotta, M. Reinecke, M. Sandri, and A. Zacchei on various conventions, input/output and software requirements relevant for this code. It is a pleasure to thank P. Natoli, N. Vittorio, N. Mandolesi, and M. Bersanelli for our collaboration on this field and M. Maris for our collaboration on Solar System studies with Planck. The kind hospitality of the LFI DPC group at INAF-OATs during the finalization of this report is acknowledged. This work has been supported by the ASI contract Planck LFI Activity of Phase E2.

References

Burigana C., Natoli P., Vittorio N., Mandolesi N., Bersanelli M., *Straylight Contamination from Internal Solar System Bodies in PLANCK/LFI Observations*, (2000a), Internal Report ITeSRE 272/2000, April.

Burigana C., Natoli P., Vittorio N., Mandolesi N., Bersanelli M., *A Preliminary Analysis of In-Flight Main Beam Reconstruction Through External Planets for PLANCK/LFI*, (2000b), Internal Report ITeSRE 273/2000, April.

Burigana C., Natoli P., Vittorio N., Mandolesi N., Bersanelli M., *In-flight main beam reconstruction for PLANCK/LFI*, (2002), Experimental Astronomy, 12/2, 87, 2001.

Galeotta S., Burigana C., *Straylight from Solar System main bodies: integration of the numerical code into LFI simulation pipeline*, (2008), Technical Note PL-LFI-OAT-TN-049, issues/rev. 1.0 (in preparation).

Burigana C., *On the definition of beam pointing direction and orientation in Planck simulation pipeline*, (2008), INAF-IASF Bologna Internal Report 518/2008, April (in preparation).

Greve A., Steppe H., Graham D., Schalinski C. J., *Disk brightness temperature of the planets at 43 GHz (and 43 GHz flux densities of some continuum sources)*, (1994), A&A, 286, 654.

Hill R.S., et al., *Five-Year Wilkinson Microwave Anisotropy Probe (WMAP)Observations: Beam Maps and Window Functions*, (2008), astro-ph/0803.0570, submitted to Astrophysical Journal Supplement Series.

Maris M., Burigana C., *Planck LFI – On Moving Objects Solar System Calculations*, (2006), Technical Note PL-LFI-OAT-TN-034, issues/rev. 1.0.

McFadden L.-A., Weissman P.R., Johnson T.V., (Editors), *Encyclopedia of the Solar System*, (2007), Second Edition, Elsevier.

Sandri M., *LFI Beams Delivery: Format Specifications*, (2003), Technical Note PL-LFI-PST-TN-044, issues/rev. 1.0.