Compilation and Usage of the Planck Simulation Modules

Document: PL-COM-MPA-MA-SIM007

Revision: unknown; April 8, 2013

Abstract: This document summarises how the modules for Planck data simulation can be compiled and used.

Prepared by: Martin Reinecke, Klaus Dolag

Authorised by: no one

Contents

1 Simulation Code

1.1 Obtaining the code

The simulation packages reside in the IDIS CVS respository at ESTEC. If you do not know how to get there, contact Martin Bremer (`mbremer@rssd.esa.int`) for `<username>` and the related password.

Note: It is not recommended to copy a CVS file tree from someone else, because CVS remembers the user who did the original checkout in the `*/CVS/Root` files. That means that the user who copied the files will not be able to update them or commit changes.

To do an initial checkout of the LevelS sources you should issue the following commands:

```
setenv CVSROOT :pserver:<username>@cvs.rssd.esa.int:/services/repositories/PLANCK_CVS
(or export CVSROOT=:pserver:<username>@cvs.rssd.esa.int:/services/repositories/PLANCK_CVS)
cvs login
cvs co -P planck/LevelS
```

Please note that the local directory `planck/LevelS` should not exist before the checkout.

1.2 Updating your version of the code

To update your copy of the source code, enter the directory `planck/LevelS` and issue the command

```
cvs -q update -Pd
```

1.3 Compilation

In order to compile the code you will need following programs:

- an up-to-date version of `GNU make`. Parallel compilation is supported for `GNU make` 3.79 and later.
- a `C` compiler supporting the C90 standard.
- a `C++` compiler supporting the C++98 standard; this includes `g++` 3.0 and above and the compilers of SGI and IBM.
- a Fortran 95 compiler with support for the Fortran 2003 module `iso_c_binding`. This includes `GNU gfortran` 4.3 and above as well as Intel `ifort` 10.0 and above.

To compile the code, follow the instructions in `planck/LevelS/README.compilation`.

Note: in order to make use of hybrid C++/Fortran modules, it is required that the employed C++ and Fortran compilers come from the same "vendor", like `g++/gfortran` or `icpc/ifort`.

2 Test Pipeline

- To run the simulation, you will need the following items:
  - Template maps and beam patterns.
    In order to keep code and data separate, there are no template maps, spectra or beam patterns included in the package. An archive file containing all the data necessary for the test pipeline can be downloaded from MPA's Planck web pages at
    http://planck.MPA-Garching.MPG.DE/SimData/test/testdata.tar.gz.
  - `xv`
    The image viewer `xv` is used only for displaying some results. It is thus not strictly required. If you prefer another image viewer, you can enter its name when `SetupSimulation` asks for it.
- Run the setup script `planck/LevelS/SetupSimulation`. It will ask for the path containing the template data and module binaries, the name of your image viewer and several other simulation-specific parameters which are explained in chapter 3.
- The script creates the parameter files necessary for running the test pipeline and tells you how to start the simulation.

  If everything worked fine, the image viewer will show you the results of your test run. You can compare them with our sample run available on our web pages at
  http://planck.MPA-Garching.MPG.DE/SimData/index_test.html.
- Be aware that the `Pipeline` script only offers a way for testing your self-compiled simulation packages and is thus only a very specialized simulation. If you want to produce *real* data, you need to set the parameters of each step carefully according to the requirements of the simulation you want to run.

3 Simulation parameters

- `NSIDE`

  The resolution of the output maps produced by the pipeline (i.e. the `HEALPix NSIDE` parameter). It has to be a power of 2 and should lie between 16 and 512.

  The sky is tesselated into $12N_{\mathrm{side}}^2$ pixels. This implies that the approximate side length of a HEALPix pixel is

  $$\delta\theta \approx \sqrt{\frac{3}{\pi}}\,\frac{3600}{N_{\mathrm{side}}} \text{ arc minutes} .$$

  This gives a clue for the choice of N_{side} required by your simulation.

- `LMAX`

  This quantity determines how accurately the "synthetic skies" are calculated, which are scanned by the detector later on. The `LMAX` parameter should be twice to three times the value chosen for `NSIDE`, up to a maximum of 2900. Note that the foreground maps distributed in `testdata.tar.gz` only contain a_{lm} up to $l_{\mathrm{max}} = 1024$, so that a higher l will not further improve the quality of the foreground signal.

- Detector name

  Depending on this, all characteristic quantities like detector frequency, sampling frequency, bandwidth, noise amplitudes, orientation relative to the optical axis etc. are determined from the focal plane database. If a simulation with special parameters is required, an additional detector with the desired parameters has to be added by hand to the database.

- First and last pointing period of the mission

  These parameters influence the sky coverage of the simulation and also the size of the output files. If unsure, leave them at the default of -1, which means that all pointing periods available in the input PPL file will be used.

- Nominal vs. realistic pointing

  If realistic pointing is chosen, detailed pointing information with synthesised pointing errors is created for each pointing period, and the sky is scanned 60 times instead of once per period. This will increase the overall running time by roughly a factor of 60 and also require more storage space for intermediate data.

4 The Simulation Package

4.1 Libraries

4.1.1 cfitsio

The cfitsio library can be obtained from http://heasarc.gsfc.nasa.gov/docs/software/fitsio/fitsio.html. LevelS currently uses version 3.33.

4.2 Optional external packages

4.2.1 fv

You may want to install the interactive FITS file viewer `fv`, which can be obtained from HEASARC. It is very convenient for looking at the data in FITS files, plotting them or making images from them. It can also be used for editing header information.

4.3 Module descriptions

Note to the authors of the simulation modules
We have tried our best in extracting the overviews and the descriptions of the parameters from the documentation provided in your packages. If there are errors in this text, or if you would like to give a more detailed description of your modules or their parameters, please send any comments to Martin Reinecke (`martin@mpa-garching.mpg.de`), who will be happy to add them to this document.

Parameter file format
The modules read parameters from ASCII files. Each parameter is defined in a separate line of the form:

```
[name] = [value]
```

Lines starting with a # sign are ignored and can thus be used as comment lines.

4.3.1 CAMB

(A. Lewis & A. Challinor)
http://camb.info

```
camb [parameter file]

example parameter file:

#Parameters for CAMB

#What to do
get_scalar_cls = T
get_vector_cls = F
get_tensor_cls = F
get_transfer   = F

#if do_lensing then scalar_output_file contains additional columns of l^4 C_l^pp and l^3 C_l^pT
```

```
#where p is the projected potential. Output lensed CMB Cls (without tensors) are in lensed_output
do_lensing     = F

# 0: linear, 1: non-linear matter power (HALOFIT), 2: non-linear CMB lensing (HALOFIT)
do_nonlinear = 0

#Maximum multipole and k*eta.
#  Note that C_ls near l_max are inaccurate (about 5%), go to 50 more than you need
#  Lensed power spectra are computed to l_max_scalar-250 where accurate at %-level
#  For high accuracy lensed spectra set l_max_scalar = (l you need) + 500
#  To get accurate lensed BB need to have l_max_scalar>2000, k_eta_max_scalar > 10000
#  Otherwise k_eta_max_scalar=2*l_max_scalar usually suffices
l_max_scalar      = 5000
k_eta_max_scalar  = 5500

#  Tensor settings should be less than or equal to the above
l_max_tensor      = 5000
k_eta_max_tensor  = 5500

#Main cosmological parameters, neutrino masses are assumed degenerate
# If use_phyical set phyiscal densities in baryone, CDM and neutrinos + Omega_k
use_physical   = F
#ombh2          = 0.022
#omch2          = 0.12
#omnuh2         = 0
#omk            = 0
hubble         = 50
#effective equation of state parameter for dark energy, assumed constant
w              = -1
#constant comoving sound speed of the dark energy (1=quintessence)
cs2_lam        = 0

#if use_physical = F set parameters as here
omega_baryon   = 0.05
omega_cdm      = 0.95
omega_lambda   = 0.0
omega_neutrino = 0.0

#massless_neutrinos is the effective number (for QED + non-instantaneous decoupling)
temp_cmb           = 2.726
helium_fraction    = 0.24
massless_neutrinos = 3.04
massive_neutrinos  = 0

#Neutrino mass splittings
nu_mass_eigenstates = 1
#nu_mass_degeneracies = 0 sets nu_mass_degeneracies = massive_neutrinos
#otherwise should be an array
#e.g. for 3 neutrinos with 2 non-degenerate eigenstates, nu_mass_degeneracies = 2 1
nu_mass_degeneracies = 0
```

```
#Fraction of total omega_nu h^2 accounted for by each eigenstate, eg. 0.5 0.5
nu_mass_fractions = 1

#Initial power spectrum, amplitude, spectral index and running. Pivot k in Mpc^-1.
initial_power_num         = 1
pivot_scalar              = 0.05
pivot_tensor              = 0.05
scalar_amp(1)             = 1
scalar_spectral_index(1)  = 1
scalar_nrun(1)            = 0
tensor_spectral_index(1)  = 1
#ratio is that of the initial tens/scal power spectrum amplitudes
initial_ratio(1)          = 1
#note vector modes use the scalar settings above


#Reionization, ignored unless reionization = T, re_redshift measures where x_e=0.5
reionization         = F

re_use_optical_depth = F
re_optical_depth     = 0.00
#If re_use_optical_depth = F then use following, otherwise ignored
re_redshift          = 50
#width of reionization transition. CMBFAST model was similar to re_delta_redshift~0.5.
re_delta_redshift    = 1.5
#re_ionization_frac=-1 sets to become fully ionized using YHe to get helium contribution
#Otherwise x_e varies from 0 to re_ionization_frac
re_ionization_frac   = 0.2


#RECFAST 1.5 recombination parameters;
RECFAST_fudge = 1.14
RECFAST_fudge_He = 0.86
RECFAST_Heswitch = 6
RECFAST_Hswitch  = T

#Initial scalar perturbation mode (adiabatic=1, CDM iso=2, Baryon iso=3,
# neutrino density iso =4, neutrino velocity iso = 5)
initial_condition   = 1
#If above is zero, use modes in the following (totally correlated) proportions
#Note: we assume all modes have the same initial power spectrum
initial_vector = -1 0 0 0 0

#For vector modes: 0 for regular (neutrino vorticity mode), 1 for magnetic
vector_mode = 0

#Normalization
COBE_normalize = T
##CMB_outputscale scales the output Cls
#To get MuK^2 set realistic initial amplitude (e.g. scalar_amp(1) = 2.3e-9 above) and
```

```
#otherwise for dimensionless transfer functions set scalar_amp(1)=1 and use
#CMB_outputscale = 1
CMB_outputscale = 1

#Transfer function settings, transfer_kmax=0.5 is enough for sigma_8
#transfer_k_per_logint=0 sets sensible non-even sampling;
#transfer_k_per_logint=5 samples fixed spacing in log-k
#transfer_interp_matterpower =T produces matter power in regular interpolated grid in log k;
# use transfer_interp_matterpower =F to output calculated values (e.g. for later interpolation)
transfer_high_precision = F
transfer_kmax           = 2
transfer_k_per_logint   = 5
transfer_num_redshifts  = 1
#transfer_interp_matterpower = T
transfer_redshift(1)    = 0
transfer_filename(1)    = transfer_out.dat
#Matter power spectrum output against k/h in units of h^-3 Mpc^3
transfer_matterpower(1) = matterpower.dat


FITS_filename = _OUTPATH_/cl.fits

##Optional parameters to control the computation speed,accuracy and feedback

#If feedback_level > 0 print out useful information computed about the model
feedback_level = 1

# 1: curved correlation function, 2: flat correlation function, 3: inaccurate harmonic method
lensing_method = 1
accurate_BB = F


#massive_nu_approx: 0 - integrate distribution function
#                   1 - switch to series in velocity weight once non-relativistic
#                   2 - use fast approximate scheme (CMB only- accurate for light neutrinos)
#                   3 - intelligently use the best accurate method
massive_nu_approx = 1

#Whether you are bothered about polarization.
accurate_polarization   = T

#Whether you are bothered about percent accuracy on EE from reionization
accurate_reionization   = F

#whether or not to include neutrinos in the tensor evolution equations
do_tensor_neutrinos     = F

#Whether to turn off small-scale late time radiation hierarchies (save time,v. accurate)
do_late_rad_truncation   = T
```

```
#Computation parameters
#if number_of_threads=0 assigned automatically
number_of_threads       = 0

#Default scalar accuracy is about 0.3% (except lensed BB).
#For 0.1%-level try accuracy_boost=2, l_accuracy_boost=2.

#Increase accuracy_boost to decrease time steps, use more k values,  etc.
#Decrease to speed up at cost of worse accuracy. Suggest 0.8 to 3.
accuracy_boost          = 1

#Larger to keep more terms in the hierarchy evolution.
l_accuracy_boost        = 1

#Increase to use more C_l values for interpolation.
#Increasing a bit will improve the polarization accuracy at l up to 200 -
#interpolation errors may be up to 3%
#Decrease to speed up non-flat models a bit
l_sample_boost          = 1
```

4.3.2 HEALPix

(K. Górski, E. Hivon, B. Wandelt, A.J. Banday, M. Bartelmann, M. Reinecke)

This is a software package for HEALPix (http://healpix.sourceforge.net), the Hierarchical, Equal Area, and iso-Latitude Pixelisation of the sphere, which was originally devised in early 1997 by Krzysztof M. Górski at the Theoretical Astrophysics Center in Copenhagen. Early development of the HEALPix concept and initial implementation was made in the spring of 1997 in collaborative work of K. Górski with Eric F. Hivon, currently at Caltech. Benjamin D. Wandelt, currently at Princeton, has contributed critically to the further development of HEALPix and related mathematical methods.

- `syn_alm_cxx`: Given a power spectrum, this program produces a Gaussian realisation of this power spectrum in the form of spherical harmonic coefficients. These coefficients can be further processed by `alm2map_cxx`.

  ```
  syn_alm_cxx [parameter file]

  Parameters read by syn_alm_cxx:

  nlmax (integer):
    maximum order of l

  nmmax (integer):
    maximum order of m (must not be larger than nlmax, default=nlmax)

  infile (string):
    input file containing the CMB power spectrum

  outfile (string):
    output file name for the calculated a_lm
  ```

```
rand_seed (integer):
  random-number seed

fwhm_arcmin (real):
  FWHM (in arcmin) of a Gaussian beam, which is used to smooth the
  resulting sky a_lm (default=0)

polarisation (bool):
  if false, only the intensity a_lm are generated,
  if true, T, G and C a_lm are generated

double_precision (bool, default=false):
  if false, the a_lm are created in single precision,
  otherwise in double precision.
```

- `alm2map_cxx`: This program can be used to create HEALPix maps (temperature only or temperature and polarisation) from $a_{\ell m}$ coefficients. Total operation count scales as $O(N_{\rm pix}^{3/2}\log N_{\rm pix})$ with a prefactor dependent on the limiting spherical harmonics order $\ell_{\rm max}$ of the actual problem. The map resolution, Gaussian beam FWHM, and random seed for the simulation can be selected by the user.

```
alm2map_cxx [parameter file]

Parameters read by alm2map_cxx:

nlmax (integer):
  maximum order of l

nmmax (integer):
  maximum order of m (must not be larger than nlmax, default=nlmax)

infile (string):
  input file containing the a_lm

outfile (string):
  output file name for the Healpix map(s)

nside (integer):
  nside parameter for the output map(s)

polarisation (bool):
  if false, only the intensity map is generated,
  if true, maps for I, Q and U are generated

fwhm_arcmin (double, default=0):
  FWHM in arc minutes of a Gaussian beam, which is used to smooth
  the a_lm

windowfile (string, default=""):
  if supplied, the pixel window function from this file will be applied
```

```
  to the a_lm

double_precision (bool, default=false):
  if false, a_lm and maps are read/written in single precision,
  otherwise in double precision.
```

- `anafast_cxx`: This program performs harmonic analysis of the HEALPix maps up to a user specified maximum spherical harmonic order ℓ_{max}. Scalar, or scalar and tensor, spherical harmonic coefficients are evaluated from the map(s) if the input provides, respectively, only the temperature, or temperature and polarisation maps. The total operation count scales as $O(\sqrt{N_{\rm pix}} l_{\rm max}^2)$ with a prefactor depending on ℓ_{max}.

  `anafast_cxx` reads a file containing the map(s) and produces a file containing the temperature power spectrum C_l^T and, if requested, also the polarisation power spectra C_l^E, C_l^B and $C_l^{T\times E}$. The a_{lm} coefficients computed during the execution also can be written to a file if requested.

  `anafast_cxx` executes an approximate, discrete point-set quadrature on a sphere sampled at the HEALPix pixel centers. Spherical harmonic transforms are computed using recurrence relations for Legendre polynomials on co-latitude, θ, and Fast Fourier Transforms on longitude, ϕ.

  `anafast_cxx` permits two execution options which allow a significant improvement of accuracy of the approximate quadrature performed by this facility:

  - An improved analysis using the provided ring weights, which correct the quadrature on latitude, and/or
  - An iterative scheme using in succession several backward and forward harmonic transforms of the maps.

```
anafast_cxx [parameter file]

Parameters read by anafast_cxx:

nlmax (integer):
  maximum order of l

nmmax (integer):
  maximum order of m (must not be larger than nlmax, default=nlmax)

infile (string):
  input file containing the Healpix map

outfile (string, default=""):
  output file name for power spectrum; if empty, no spectrum is written

outfile_alms (string, default=""):
  output file name for the a_lm; if empty, no a_lm are written

polarisation (bool):
  if false, only the intensity a_lm are generated,
  if true, a_lm for T, G and C component are generated
```

```
ringweights (string, default=""):
  if supplied, ring weights will be read from this file

iter_order (integer, default=0)
  number of iterations for the analysis (0: standard analysis)

double_precision (bool, default=false):
  if false, maps and a_lm are read/written in single precision,
  otherwise in double precision.

if (polarisation==true && outfile!="")
  full_powerspectrum (bool, default=false):
    if true, write a 6-column power spectrum;
    if false, write a 4-column power spectrum.
endif
```

- `map2tga`: This facility provides a means to generate a TGA image from an input HEALPix sky map. It is intended to allow some primitive visualisation for those with limited or no access to IDL. It is also useful for image generation in a pipeline environment. `map2tga` can read parameters from the command line or from a file.

```
Usage:
  map2tga <init object>

or:
  map2tga <input file> <output file> [-sig <int>] [-pal <int>]
    [-xsz <int>] [-bar] [-log] [-asinh] [-lon <float>] [-lat <float>]
    [-mul <float>] [-add <float>] [-min <float>] [-max <float>]
    [-res <float>] [-title <string>] [-flippal] [-gnomonic]
    [-interpol] [-equalize] [-viewer <viewer>]


Parameters read by map2tga:

infile (string):
  input file containing the Healpix map

outfile (string):
  output TGA file

sig (string, default="I_Stokes"):
  column name of the requested Healpix map

pal (integer, default=4):
  number of the color palette

flippal (bool, default=false):
  whether the palette should be flipped

xsz (integer, default=1024):
```

```
  number of image pixels in x direction

bar (logical, default=false):
  whether a color bar should be added to the image

log (logical, default=false):
  whether the logarithm of the map values should be displayed

equalize (logical, default=false):
  whether histogram equalisation should be performed

asinh (logical, default=false):
  whether the hyperbolic arcsine of the map values should be displayed

lon (double, default=0):
  the longitude (in degrees) of the image center

lat (double, default=0):
  the latitude (in degrees) of the image center

mul (double, default=1):
  scale factor applied to the data

add (double, default=0):
  offset added to the data (before multiplication)

min (double, optional):
  if specified, this value is used as minimum of the color scale

max (double, optional):
  if specified, this value is used as maximum of the color scale

res (double, default=1):
  only for gnomonic projection: the size (in arcmin) of an image pixel

title (string, optional):
  if specified, this value is used as the image title

viewer (string, optional):
  if specified, this executable is used to show the resulting image

pro (string, default="mollw"):
  if this is equal to "gno", gnomonic projection is used, else Mollweide

interpol (bool, default=false):
  false: no interpolation
  true : bilinear interpolation
```

- `rotalm_cxx`: Converts a_{lm} between different coordinate systems.

```
Usage: rotalm_cxx <infile> <outfile> <itransform> <pol>
or   : rotalm_cxx <infile> <outfile> <psi> <theta> <phi> <pol>

itransform: 1: Equatorial (2000) -> Galactic   (2000)
            2: Galactic   (2000) -> Equatorial (2000)
            3: Equatorial (2000) -> Ecliptic   (2000)
            4: Ecliptic   (2000) -> Equatorial (2000)
            5: Ecliptic   (2000) -> Galactic   (2000)
            6: Galactic   (2000) -> Ecliptic   (2000)
            7: Equatorial (1950) -> Galactic   (1950)
            8: Galactic   (1950) -> Equatorial (1950)
            9: Equatorial (1950) -> Ecliptic   (1950)
           10: Ecliptic   (1950) -> Equatorial (1950)
           11: Ecliptic   (1950) -> Galactic   (1950)
           12: Galactic   (1950) -> Ecliptic   (1950)

psi, theta, phi: Euler angles (in degrees)

pol: T or F
```

4.3.3 Almmixer

(Martin Reinecke, MPA)

This module reads CMB temperature fluctuation a_{lm} and various foreground a_{lm}, calculates the frequency-dependent intensities of all contributions and convolves them with the response function W_ν of a given detector. The resulting combined set of a_{lm} is written in units of antenna temperature.

The antenna temperature T_A is defined as follows:

$$T_\mathrm{A} = \int \frac{dT_\mathrm{A}(\nu)}{d\nu} d\nu = \int \frac{W_\nu c^2}{2\nu^2 k} \frac{dF(\nu)}{d\nu} d\nu \tag{1}$$

This quantity is calculated approximately by subdividing the frequency band of the detector into many sub-bands, computing the contribution of the nth sub-band using the formula

$$T_{\mathrm{A}n} = \frac{W_{\nu n} c^2}{2\nu_n^2 k} \frac{dF(\nu_n)}{d\nu} \Delta\nu_n, \tag{2}$$

where ν_n is the central frequency of the sub-band, and $\Delta\nu_n$ is its width, and summing over all $T_{\mathrm{A}n}$.

So far, `almmixer` knows about the following radiation sources:

- CMB (from `synfast`):
  Thermodynamic temperature fluctuations in Kelvin. This information is used to determine the spectral intensity at the required frequencies.
- Galactic dust emission:
  Input is given at 100 μm in MJy/sr and extrapolated to the actual frequency using the spectrum shown in the left panel of Fig. 1. The dust temperature can be adjusted.
- Galactic dust emission with a simplified 2-component model:
  Input is given at 100 μm in MJy/sr and extrapolated to the actual frequency using a 2-component model suggested by C. Baccigalupi.

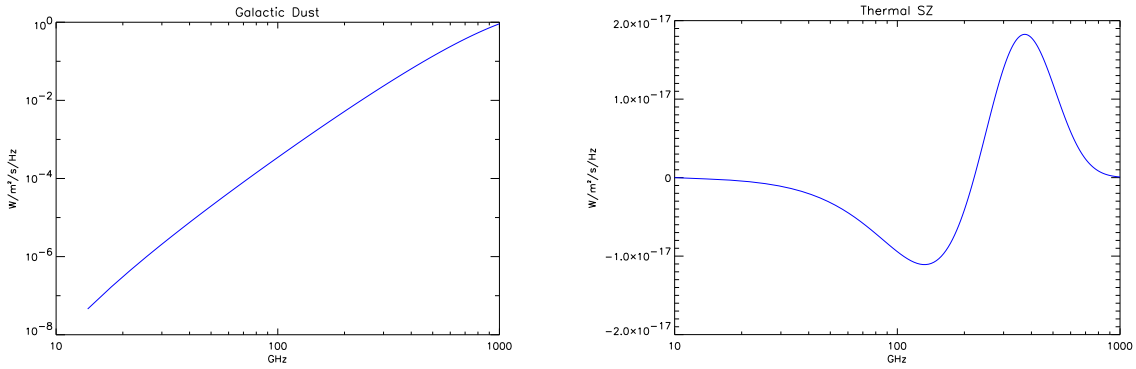

Figure 1: Spectrum for galactic dust emission (left, assumed dust temperature is 18K) and for the thermal SZ effect (right).

- Galactic synchrotron emission:
  Input is given at 408 MHz in MJy/sr and extrapolated to the actual frequency using a power law with an exponent of -0.75 between 408 MHz and 22 GHz, and an exponent of -1.25 above 22 GHz.
- Galactic free-free emission:
  Input is the map of galactic H_α-emission published by Finkbeiner (2003). The intensity is calculated according to the model of Valls-Gabaud (1998).
- thermal SZ effect:
  The Compton-y parameter is given. `Skymixer2` multiplies it with the thermal SZ spectrum shown in the right panel of Fig. 1.
- kinetic SZ effect:
  The Compton-w parameter is given. `Skymixer2` multiplies it with the kinetic SZ spectrum.
- Emission by rotating CO molecules:
  This foreground component only contributes to the HFI channels. The input map was published by Dame et al. (1996,2001).
- arbitrary a_{lm} in MJy/sr at the detector frequency:
  No spectral extrapolation is performed.

```
almmixer [parameter file]

Parameters read by almmixer:

focalplane_db (string):
  input file containing the detector database

detector_id (string):
  name of the detector

amx_nmaps (integer):
  number of maps to mix
```

```
amx_output_map (string):
  output file name for the mixed a_lm

amx_lmax (integer):
  maximum l of the output a_lm

amx_mmax (integer):
  maximum m of the output a_lm

amx_polar (bool):
  if true, polarised a_lm are created. Note that a_lm without polarisation
  information can still be used as input; they will only be mixed into the
  intensity a_lm set.

amx_response_type (string):
  shape of the detector response function
    DELTA : delta function at the detector frequency, with a magnitude of the
            detector bandwidth
    TOPHAT: tophat function from the minimum to the maximum frequency of the
            detector
    GAUSS : Gauss function centered at the detector frequency, with a sigma
            of the detector bandwidth

amx_monopole (bool, default=false):
  whether the CMB monopole should be included when a CMB map is present.

do x=1,amx_nmaps
  amx_mapname[x] (string):
    input alm file for component [x]

  amx_maptype[x] (string):
    type of the component [x]
      CMB       : polarised CMB fluctuation map (as created by syn_alm)
      CMB_UNPOL : unpolarised CMB fluctuation map (as created by syn_alm)
      DUST      : galactic dust emission
      DUST2     : galactic dust emission with a two-component model suggested
                  by C. Baccigalupi
      SYNCHRO   : galactic synchrotron emission
      SZ        : thermal Sunyaev-Zeldovich effect
      SZKIN     : kinetic Sunyaev-Zeldovich effect
      FREEFREE  : free-free emission
      CO        : emission of rotating CO
      MJY       : unpolarised map is given in MJy/sr at the detector frequency;
                  no spectral extrapolation is performed
      MJY_POL   : polarised map is given in MJy/sr at the detector frequency;
                  no spectral extrapolation is performed
      KRJ       : unpolarised map is given in K_RJ;
                  will be added unchanged to the output map
      KRJ_POL   : polarised map is given in K_RJ;
                  will be added unchanged to the output map
```

```
  if (amx_maptype[x]==DUST)
    amx_dust_temp[x] (double):
      temperature (in K) for the galactic dust emission.
      A commonly used value is 18K
  endif

  if (amx_maptype[x]==CO)
    amx_co_temp[x] (double):
      temperature (in K) of the rotating CO molecules.
      A commonly used value is 20K.
  endif
end do
```

4.3.4 skymixer3

```
skymixer3 [parameter file]

WARNING: This is still incomplete and preliminary!

Parameters read by skymixer3:

nside (integer):
  NSIDE parameter of the output map; the templates will be up/degraded
  accordingly.

output_map (string):
  output file name for the mixed Healpix map

response_type (string):
  shape of the detector response function
    DELTA  : delta function at the detector frequency, with a magnitude of the
             detector bandwidth
    TOPHAT : tophat function from the minimum to the maximum frequency of the
             detector
    GAUSS  : Gauss function centered at the detector frequency, with a sigma
             of the detector bandwidth
    TABULAR: Tabulated response function

if (response_type==TABULAR)
  response_table (string):
    input file containing the tabulated detector frequency response
else
  focalplane_db (string):
    input FITS file containing the detector database
  detector_id (string):
    name of the detector
endif
```

```
calc_CMB (bool, default=false):
  should a CMB component be added?

if (calc_CMB)
  CMB_LS_map (string):
    input map containing the CMB
  CMB_LS_monopole (bool, default=false):
    should the CMB monopole be included?
endif

analogously for the component types
  CMB, CMB_PRSM, DUST_PRSM, SYNCHRO_PRSM, SZ_PRSM, RADIO_PRSM, INFRARED_PRSM,
  FREEFREE_PRSM, SZ, SZKIN, FREEFREE, and CO


n_tabular(integer, default=0):
  number of tabular components

for i=1,n_tabular
  tabular_[i]_polarised (bool, default=true)
    true if the input maps for this component are polarised, else false

  tabular_[i]_nmaps (integer):
    number of maps for this tabular component

  for j=1,tabular_[i]_nmaps
    tabular_[i]_map_[j] (string):
      name of the map [j] for component [i]; the map is expected to have the
      unit MJy/sr
    if (tabular_[i]_nmaps>1)
      tabular_[i]_freq_[j] (double):
        frequency (in Hz) of the map [j] for component [i]
    endif
  end
end
```

4.3.5 Beam

Polarised Beam Analysis (Mark Ashdown, CPAC)

This package contains tools for reading beam data from Grasp files (in a limited set of ”cut” and ”grid” sub-formats), converting polarised beam data into Stokes parameters, calculating effective beams including cross-polar leakage in the detectors and extracting the spherical harmonic coefficients from the Stokes parameters.

The sources of the programs described below are commented, as are the modules on which they are based. They are programmed in an object-based style; data is stored in Fortran 90 types.

- `grasp2stokes`: Reads beam amplitudes from a Grasp file and converts them to Stokes parameters. This module can read three particular sub-formats of Grasp files, called for our purposes ”grd_hfi”, ”grd_lfi” and ”cut”.

grd_hfi: grid format file containing beam data on a polar theta-phi grid. Used by HFI for main beam simulations. This type of input will be converted into Stokes parameters on a polar grid.

grd_lfi: grid format file containing beam data on a square u-v grid. Used by LFI for main beam simulations. This type of input will be converted into Stokes parameters on a square grid.

cut: cut format file containing beam data on constant-phi cuts. Usually contains full-sky data. This type of input will be converted into Stokes parameters on a polar grid.

```
grasp2stokes [parameter file]

Parameters read by grasp2stokes:

grasp_file (string):
  Input Grasp (text) file.

grasp_format (string):
  Specifies sub-format of Grasp file.  Must be one of:
    - "grd_square";
    - "grd_polar";
    - "cut".

grasp_copol (string):
  Specifes co-polar basis direction. Must be "x" or "y".

grasp_norm (string):
  Normalisation convention of the input Grasp beam.  Must be "unity",
  "four_pi" or "eight_pi".  The value of this parameter is used to
  re-normalise the output Stokes parameters correctly for internal
  pipeline purposes.

if (grasp_format == "grd_square")
  stokes_file_square (string):
    Output beam object in square format.
else
  stokes_file_polar (string):
    Output beam object in polar format.
endif
```

- `crosspol`: Combines two perfectly polarised beams (as produced by grasp2stokes) to create an effective beam with detector cross-polar leakage included.

```
crosspol [parameter file]

Parameters read by crosspol:

focalplane_db (string):
  name of file containing focal plane database

detector_id (string):
  detector ID
```

```
Either:
  co_file_polar (string):
    Input co-polar beam object.

  cross_file_polar (string, default=''):
    Input cross-polar beam object.

  eff_file_polar (string):
    Output effective beam object.
Or:
  co_file_square (string):
    Input co-polar beam object.

  cross_file_square (string, default=''):
    Input cross-polar beam object.

  eff_file_square (string):
    Output effective beam object.
End

angle (double, default=0):
  Angle between coordinate systems of co- and
  cross-polar beams
```

- `beam2alm`: Reads beam Stokes parameters and then calculates their spherical harmonic coefficients.

  This program can take its input from one or two objects:

  1. Main beam at high resolution (the ”north polar cap”);
  2. Full-sky beam at lower resolution.

  Both are optional, but there must be at least one input object!

  The main beam may be a polar or square beam object. If it is a square beam object, it will be interpolated onto a polar grid before extracting the multipoles. The full-sky beam must be in a polar beam object.

  If there is both a main beam and a full sky beam then the full sky beam will be interpolated in the theta direction so that it has the same theta resolution as the full sky beam. This is to give better extraction of the multipoles.

```
beam2alm [parameter file]

Parameters read by beam2alm:

Either:
  beam_main_file_polar (string):
    Object containing a polar main beam.
Or:
  beam_main_file_square (string):
    Object containing a square main beam.
```

```
  beam_nphi (integer):
    Number of phi positions used to interpolate main beam

  beam_ntheta (integer):
    Number of theta positions used to interpolate main beam
Or:
  <nothing>
End

beam_full_file (string):
  Object containing full-sky beam. If left empty, there is no full-sky beam.

beam_lmax (integer):
  maximum l for analysis

beam_mmax (integer):
  maximum m for analysis

beam_alm_file (string):
  Output object for beam multipoles
```

- `gaussbeampol`: Calculates the multipoles of an linearly-polarised elliptical Gaussian beam with following properties:
  - beam points in z-direction (towards ”north pole”).
  - beam ellipse is described by three parameters:
    1. mean FWHM, defined as fwhm = sqrt(fwhm_max*fwhm_min);
    2. ellipticity, defined as fwhm_max/fwhm_min;
    3. orientation angle. Major axis is oriented at angle psi_ell to the x-axis;
    4. beam is polarised along direction at angle psi_pol to the x-axis.

```
gaussbeampol [parameter file]

Parameters read by gaussbeampol:

focalplane_db (string):
  name of file containing focal plane database

detector_id (string):
  detector ID

beam_lmax (integer, default=1024):
  maximum l for output

beam_mmax (integer, default=2):
  maximum m for output

beam_nstokes (integer, default=3):
```

```
    number of Stokes components (1, 3 or 4)

  beam_elliptic (logical, default=.true.):
    if .true., an elliptic beam is simulated, otherwise a circular beam

  beam_alm_file (string):
    output file containing a_lm of the generated beam
```

4.3.6 Total convolution

The mathematics behind total convolution is described in Wandelt & Górski, "Fast convolution on the sphere", Phys. Rev. D 63, 123002 (2001), and in Challinor et al., "All-sky convolution for polarimetry experiments", Phys. Rev. D 62, 123002 (2000). Briefly, the algorithm performs full-sky convolutions of beam and sky for all possible beam pointings, and this is repeated for all possible rotations of the beam about its symmetry axis. The output is written into one separate file for each beam rotation. The phrase "all possible" refers to the resolution of beam and sky. If $\ell_{\rm max}$ is the maximum multipole order supplied, angles on the sphere need to be discretised into $2\ell_{\rm max}+1$ steps. The output per beam rotation is then written into arrays of size $(2\ell_{\rm max}+1)\times(\ell_{\rm max}+1+{\rm padding})$. Likewise, if $m_{\rm max}$ is the maximum multipole order describing the azimuthal asymmetry of the beam, there are $2m_{\rm max}+1$ possible beam rotations.

```
conviqt_v3 [parameter file]

Parameters read by conviqt_v3:

fwhm_deconv (real, default=0):
  deconvolution FWHM in arc minutes.
  If the input map is smoothed, set fwhm_deconv to the FWHM of the smoothing
  beam, else set to 0.

polarisation (bool, default=true):
  whether the G and C components are also convolved and added to T

conv_lmax (integer):
  maximum l fo the convolution

lmax_out (integer, default=conv_lmax):
  used to determine the resolution of the generated ring set. It will have
  2*lmax_out+1 pixels in phi-direction.

nphi (integer, default=2*lmax_out+1):
  used to determine the resolution of the generated ring set. It will have
  nphi pixels in phi-direction.

ntheta (integer, default=lmax_out+2):
  used to determine the theta resolution of the generated ring set. It will have
  ntheta rings in theta-direction (including theta=0 and theta=pi).
  (NB: additional rings will be automatically computed for interpolation
  purposes.)
```

```
beam_alm (string):
  input file containing the beam a_lm

beammmax (integer):
  maximum m of the beam

sky_alm (string):
  input file containing the sky a_lm

ringset (string):
  output file name for the ring set
```

4.3.7 simmission4

(F. v. Leeuwen, D. Mortlock, M. Reinecke)

This program writes out a file containing location and pointing information for the Planck satellite.

The program prepares pointing information for the Planck satellite according to a scanning strategy, dynamical parameters of the satellite and various noise estimates.

The output consists of a file containing one record for each pointing period, from which it is possible to reconstruct the pointing of the satellite as a function of time over the pointing period to an accuracy level of a few arcsec.

There is also an option to produce additional information, containing the results of a numerical integration over the satellite attitude. This produces pointing records at a requested frequency with an accuracy of well below one arcsec. The latter option can also be used to study the effects of internal torques acting on the satellite (such as due to moving liquids). Such effects would have to be included in the calculations of the torques acting on the satellite.

Note: The samples produced by `simmission4` are always evaluated at the *center* of a time interval, not at its beginning! This means that the first sample of a simulation starting at $t = 0$ with a sampling rate of 1Hz will be taken at $t = 0.5$s.

```
simmission4 [parameter file]

example parameter file:

# ****************************************************************************
# simmission4
# ****************************************************************************

##############################################################################
# EPHEMERIS DATA

# object containing the HORIZON ephemerides of the Sun, where Planck is
# the observer.
# This object must be of DDL type "ephemeris.LS_ephemeris"
ephemerides = /path/to/file

##############################################################################
# TECHNICAL PARAMETERS
```

```
# Seeds for the random number generator
randseed_1 = 1234
randseed_2 = 5678

################################################################################
# SATELLITE PARAMETERS

# How the satellite dynamics are to be modelled (``ideal'' or ``analytical'').
dynamics = analytical

# Satellite centre-of-gravity (in m, relative to the centre of the Solar
# panel, in coordinates parallel to the satellite reference system, with
# the z-displacement ``into'' the satellite or away from the Sun, the
# x-displacement in the direction of the focal plane and the y-displacement
# perpendicular to this).
pos_cog_sat_x = 0.03
pos_cog_sat_y = 0
pos_cog_sat_z = 0.93

# Satellite inertia tensor (in kg m^2 in the satellite reference system
# defined above). The off-diagonal elements I_xz = I_zx and I_yz = I_zy
# are not zero in this reference frame, unlike the inertial reference
# frame.
i_sat_xx = 2236.5
i_sat_yy = 2276.7
i_sat_zz = 2650.3
i_sat_xy = -98
i_sat_xz = -0.1
i_sat_yz = 0.7

# Radius of the solar panel (in m).
r_panel_sat = 2

# Specular reflection coefficient of the satellite solar panel.
specref_panel_sat = 0.17

# Diffuse reflection coefficient of the satellite solar panel.
diffrel_panel_sat = 0.1

################################################################################
# POINTING PARAMETERS

# The z-axis pointing mode, the default option being ``ideal'' (i.e.,
# the satellite's z-axis is always ideal), the other option being
# ``gaussian'' (i.e., there is a Gaussian error added to the nominal
# pointing of mean error sigma_zaxis_x_pointing (in arcmin) about the
# x-axis and sigma_zaxis_y_pointing (in arcmin) about the y-axis but
# with a maximum value of delta_zaxis_max_x_pointing (in arcmin) about
# the x-axis and delta_zaxis_max_y_pointing (in arcmin) about the
```

```
# y-axis)
mode_zaxis_pointing = gaussian

# Mean error on the satellite pointing (in arcmin).
sigma_zaxis_x_pointing = 0.2
sigma_zaxis_y_pointing = 0.2

# Maximum error on the satellite pointing (in arcmin).
delta_zaxis_max_x_pointing = 1.0
delta_zaxis_max_y_pointing = 1.0

# The mode with which the initial scan phase of a pointing period is
# determined, the default being ‘‘ideal’’ (in which case the satellite can
# be thought of as happily rotating continuously about its z-axis, the
# repointings notwithstanding) or ‘‘random’’ (in which case the scan
# phase is, of course, random).
mode_phase_pointing = random

# The rotation rate error mode, ‘‘ideal’’ again being that the nominal
# rotation rate is held to rigorously and ‘‘gaussian’’ results in
# Gaussian errors on the rotation rate (of mean error
# sigma_rate_rot_pointing and maximum error delta_rate_rot_max_pointing).
mode_rate_rot_pointing = gaussian

# Mean error on the satellite rotation rate [in deg s^(-1)].
sigma_rate_rot_pointing = 0.0006

# Maximum error on the satellite rotation rate [in deg s^(-1)].
delta_rate_rot_max_pointing = 0.0012

##############################################################################
# SCAN STRATEGY PARAMETERS

# File name of mission file.
# This must be an ASCII file.
scanfile = data/mission_test.dat

# Format of the mission file.
# This can be either "ppl" (default), "appls", or "plan".
scandataformat = ppl

# First pointing period to be processed.
# If unset or set to -1, use the first period in the mission file.
first_period=-1

# Last pointing period to be processed.
# If unset or set to -1, use the last period in the mission file.
last_period=-1

# Rotation period of the satellite (in min).
```

```
period_rot_scan = 1.0

##############################################################################
# SIMULATION OUTPUTS

# Filename for the mission output file containing all pointings.
file_mission = satpoint_mission

# Period between output times (in s)
period_sm_mission = 1

# Whether to perform a full mission simulation (i.e., whether to produce
# pointing files and loop over the samples in a pointing at all).
simulatepointings = N
```

4.3.8 Focalplane

- `focalplane`: converts the detector database from text format to the LevelS format.

  ```
  focalplane [parameter file]

  Parameters read by focalplane:

  focalplane (string):
    input ASCII file containing the detector database

  focalplane_db (string):
    output file containing the detector database

  theta_b (double):
    bore sight angle in degrees (also known as kappa)
  ```

4.3.9 OOF

(K. Górski, B. Wandelt, C. Burigana, D. Maino, S. Plaszczynski, M. Reinecke)
This module allows the simple generation of processes with power spectrum

$$P(f) = NET_{\rm RJ}^2 \left[1 + \left(\frac{f_{\rm knee}^{\alpha}}{f^{\alpha} + f_{\rm min}^{\alpha}}\right)\right] ,$$

where $0 \leq \alpha \leq 2$. In other words, the spectrum is white below $f_{\rm min}$ and above $f_{\rm knee}$, and has a slope of $-\alpha$ in between. The parameters $NET_{\rm RJ}$, α, $f_{\rm knee}$ and $f_{\rm min}$ are taken from the focal plane database.

The code contains three different implementations of the noise generator:

- One algorithm is based on the superposition of a number of component processes, each of which is generated by evolving a set of simple stochastic differential equations (SDE).
- A fast algorithm for the special case $\alpha = 2$, which uses a digital filter to obtain the desired spectral shape, was developed by S. Plaszczynski.

- A still experimental algorithm nakes use of SDEs that are fed with one random number per sample, instead of one random number per SDE and per sample. Since the generation of Gaussian random numbers is the most time-consuming part of the code, this results in a significant speedup.
- A fast algorithm for $1 <= \alpha <= 2$, which uses a set of digital filters to obtain the desired spectral shape, was developed by S. Plaszczynski. This algorithm produces noise with a power spectrum of

$$P(f) = NET_{\mathrm{RJ}}^2 \left[\frac{f^2 + f_{\mathrm{knee}}^2}{f^2 + f_{\mathrm{min}}^2} \right]^{\alpha/2}.$$

The code is not available as a stand-alone module; it is integrated into `multimod` instead.

4.3.10 Multimod

(R. Hell, A. Mennella, K. Górski, B. Wandelt, C. Burigana, D. Maino, M. Bartelmann, K. Dolag, L. Mendes, I. Grivell, R. Mann, M. Reinecke)

`Multimod` is a C++ module which includes functionality of many modules running after the `simmission` step (i.e. everything dealing with time-ordered data). Currently it includes the functionality of `detpoint`, `interpol`, `dipole`, `oofnoise`, `sampler`, `countMap` and `makeMap`. The combination of various modules greatly reduces the amount of temporary data I/O.

```
multimod [parameter file]

Parameters read by multimod:

focalplane_db (string):
  input file containing the detector database

detector_id (string):
  name of the detector

nominal_pointing (bool):
  Determines whether nominal (idealized) pointing is used.

bypass_sampler (bool, default=false):
  if true, the effect of the sampling electronics is not simulated

if (bypass_sampler==false)
  sampler_timeshift (real, default=0):
    number of samples (measured at the nominal detector sampling frequency),
    by which the sampled output signal will be shifted.
    (For LFI, a value of 0.5 will compensate the time lag caused by the
    integration mechanism.)
endif

satinfo_type (string, default="SIMMISSION"):
  can have the values "SIMMISSION", "LFI", and "HFI"

satinfo_ephemeris (string):
  object containing the ephemeris of the Sun relative to Planck
```

```
if (satinfo_type==SIMMISSION)
  sat_info (string):
    Name of the input file (produced by simmission) containing the satellite
    orientation. NOTE: the satellite position will be ignored!
else if (satinfo_type==LFI)
  satinfo_input (string):
    Name of the input object containing the full satellite pointing information
  satinfo_sampleinfo (string):
    object containing information about the time of the first sample and number
    of samples for each pointing period
else if (satinfo_type==HFI)
  satinfo_quaternions (string):
    Name of the input object containing satellite orientation quaternions.
  satinfo_ctr (string):
    Name of the input object containing the central time reference.
  satinfo_indexobject (string):
    Name of the input object containing start and end indices of the pointing
    periods
endif

first_pointing (int):
  number of the first pointing period to calculate. Default is 1,
  which is also used when "-1" is given.

last_pointing (int):
  number of the last pointing period to calculate. Default is the last
  period in the satellite information file, which is also used when "-1"
  is given.

oversampling_factor (double):
  the ideal sky samples are taken at a rate of
  "oversampling_factor*f_samp". If "bypass_sampler==true", this factor must be
  exactly 1.

cnt_file (string, default=""):
  if not empty, coverage information for the detector pointings
  is written to a file with this name.

detpt_file (string, default=""):
  if not empty, detector pointings are written to a file with this name.

detpt_aberration (bool, default=false):
  if true, detector pointings are corrected for aberration caused by
  Planck's motion around the Solar System Barycenter

if (detpt_file!=""):
  single_precision_detpt (bool, false by default):
    determines whether detector pointings are written in single or double
    precision
```

```
endif

detpt_wcorr (bool, default=false):
  if true, wobble correction is enabled

if (detpt_wcorr=true):
  wcorr_tilt_angles (string):
    object containing the tilt angles for each pointing period
endif

detpt_ptcor (bool, default=false):
  if true, PTCOR is enabled

if (detpt_ptcor=true):
  ptcor_file (string):
    CSV-formatted file containing times and the respective pointing corrections
endif

map_file (string, default=""):
  if not empty, the Healpix map containing the binned TOD is written
  to a file with this name.

if (cnt_file!="" || map_file!="")
  nside (int):
    nside parameter of the Healpix maps that are output by multimod.
endif

tod_file (string, default=""):
  if not empty, the TOD is written to a file with this name.

if (tod_file!="" || map_file!="")
  calibrate_signal (bool):
    if true, the TOD output is multiplied by 2/(1+epsilon), where epsilon is the
    amount of cross-polar leakage.
endif

quaternions_file (string, default=""):
  if not empty, the quaternions describing the satellite orientation are
  written to a file with this name.

timestamp_file (string, default=""):
  if not empty, a timestamp for every sample of the TOD will be
  written to a file with this name.

index_file (string, default=""):
  if not empty, the indices of the first and last sample of every pointing
  period are written to a file with this name.

repointing_flag_file (string, default=""):
  if not empty, a flag timestream will be written to this object, which is true
```

```
  during repointing manoeuvres, and fals otherwise. This only works if
  nominal_pointing is false.

if (repointing_flag_file != "")
  repointing_rand_seed (integer, default=4711):
    seed for the random number generator used to determine the lengths of
    the repointing manoeuvres.
endif

source_mixed (bool, false by default):
  determines whether the ring sets calculated by the totalconvolver are
  used as input signal.

if (source_mixed)
  ringset (string):
    File containing the input ringset (output of the total convolver).
    The data in this file must be given in K(antenna).

  output_type (string,default = SIGNAL):
    I     : extract the I signal from the ring files
    Q     : extract the Q Stokes parameter from the ring files
    U     : extract the U Stokes parameter from the ring files
    SIGNAL: extract the signal actually seen by the detector

    NOTE: The modes I, Q, and U only produce useful output data for
    ringsets produced with axisymmetric beams!

  interpol_order (int, default=1):
    the order of polynomial interpolation. This parameter must be positive,
    odd, and smaller than 20.

  interpol_galactic (bool, default=false):
    if true, the ringset is expected to be in Galactic coordinates, else in
    Ecliptic coordinates.
    The multimod output will always be in Ecliptic coordinates.
endif

source_mixed2 (bool, false by default):
  determines whether Healpix maps (which are smoothed with a circular beam
  profile) are used as input.
  NOTE: This facility has been added for special-purpose simulations only,
        and should not be used in general!

if (source_mixed2)
  interpol2_map (string):
    name of the input file containing the Healpix map.
    The data in this file must be given in K(antenna).

  interpol2_polarisation (bool, false by default):
    if false, only the intensity map is read, else I, Q and U components
```

```
    are read.

  interpol2_galactic (bool, default=false):
    if true, the map is expected to be in Galactic coordinates, else in
    Ecliptic coordinates.
    The multimod output will always be in Ecliptic coordinates.

  interpol2_output_type (string,default = SIGNAL):
    I     : return the I map value
    Q     : return the Q map value
    U     : return the U map value
    SIGNAL: extract the signal actually seen by the detector
endif

source_dipole (bool, false by default):
  determines whether the CMB dipole is used as input signal.

source_fsldp (bool, false by default):
  determines whether the dipole pickup of the far side lobes is used as
  input signal.

if (source_dipole||source_fsldp)
  dipole_thermotemp (bool, false by default):
    true : calculate thermodynamic temperature
    false: calculate antenna temperature

  dipole_speed (string,default="TOTAL")
    motion used to calculate the dipole
      SOLSYS   : use only the speed of the solar system relative to the CMB
      SATELLITE: use only the speed of the satellite relative to the sun
      TOTAL    : use the speed of the satellite relative to the CMB

  dipole_type (int, 2 by default):
    type of dipole component to calculate:
      1: total relativistic Doppler effect (monopole plus relativistic Doppler
         effect / "dipole anisotropy")
      2: relativistic Doppler effect ("dipole anisotropy")
      3: total non-relativistic Doppler effect (pure dipole plus monopole)
      4: non-relativistic Doppler effect (pure dipole)
      5: quadrupole component of relativistic Doppler effect
      6: higher components of relativistic Doppler effect
      7: relativistic component of the Doppler effect
         (difference between relativistic and non-relativistic Doppler effect)

  dip_norm (real, 1 by default):
    factor applied to the computed dipole signal (handy to emulate
    bandpass effects, for example)
endif

source_zle (bool, false by default):
```

```
  determines whether the zle is used as input signal.

if (source_zle)
  zle_components(string):
    integer list containing the desired components (possible choices are
    1,2,11,12,13,20 and 30)
  zle_ephemeris(string):
    object containing the ephemeris of the Sun and the Solar system barycenter
    relative to Planck
endif

source_oof (bool, false by default):
  determines whether the 1/f-noise is used as input signal.

if (source_oof)
  oof_rand_seed (integer):
    random number generator seed for the 1/f-noise

  oof_mode (string, default=BOTH):
    WHITE: simulate only white noise
    OOF  : simulate only pure 1/f noise
    BOTH : both of the above

  oof_method (string, default=CLASSIC):
    CLASSIC  : multiple SDEs, multiple random numers per sample
    NEWNOISE : multiple SDEs, one random number per sample (experimental!)
    OOF2NOISE: 1/f^2 generator by S. Plaszczynski
               (works only if the slope is -2 and oof_mode==BOTH)
    OOFANOISE: fast 1/f^alpha generator by S. Plaszczynski
               (works only if oof_mode==BOTH)

  oof_stationary_periods (integer, default=-1):
    number of pointing periods after which the noise generator is reset
    (i.e. the correlation is destroyed)
    If this is <=0, no resets take place.

  if (nominal_pointing)
    oof_num_real (integer, default=60):
      number of OOF realisations per pointing period with nominal pointing
  endif
endif

source_pntsrc (bool, false by default):
  determines whether the point source convolver is used as input signal.

if (source_pntsrc)
  pntsrc_polarisation (bool, default=false):
    should polarised intensity be calculated?

  pntsrc_file (string, default=""):
```

```
    if not empty, input file containing the point source catalog

  planet_file (string):
    if not empty, object containing the planet ephemerides

  beam_file_type (integer):
    type of the beam:
      2: Gaussian beam with fwhm given in the detector database
      3: Elliptic Gauss beam with parameters given in the detector database
      4: beam on an equidistant (theta,phi) grid, as produced by crosspol
         or alm2grid
      5: beam on an equidistant Cartesian grid (LFI main beams), as produced
         by crosspol

  if (beam_file_type==[4,5])
    beam_file (string):
      input file containing the beam
  endif

  beam_radius_max (real):
    cutoff radius of the beam (in degrees)

  psrchits_file (string, default=""):
    if not empty, output file containing a list of encountered point sources

  variable_pntsrc_factor (real, default=1):
    factor to be applied to the signal of variable point sources
endif
```

4.3.11 Tools

(M. Bartelmann, MPA; M. Reinecke, MPA)

This package contains several tools to convert or modify different types of data or provide some input data for other modules.

- `HPXconvert_cxx`: Convert a Healpix map between different coordinate systems. It is used to convert foreground maps like the galactic synchrotron map. Both unpolarised and polarised maps are supported.

  `HPXconvert_cxx` operates in real space only.

  ```
  Usage: HPXconvert_cxx <infile> <outfile> <itransform> <pol>
  Transform 1: Equatorial (2000) -> Galactic   (2000)
            2: Galactic   (2000) -> Equatorial (2000)
            3: Equatorial (2000) -> Ecliptic   (2000)
            4: Ecliptic   (2000) -> Equatorial (2000)
            5: Ecliptic   (2000) -> Galactic   (2000)
            6: Galactic   (2000) -> Ecliptic   (2000)
            7: Equatorial (1950) -> Galactic   (1950)
            8: Galactic   (1950) -> Equatorial (1950)
  ```

```
 9: Equatorial (1950) -> Ecliptic   (1950)
10: Ecliptic   (1950) -> Equatorial (1950)
11: Ecliptic   (1950) -> Galactic   (1950)
12: Galactic   (1950) -> Ecliptic   (1950)

pol: T or F
```

- `rotmap_cxx`: Convert a Healpix map between different coordinate systems. It is used to convert foreground maps like the galactic synchrotron map. Both unpolarised and polarised maps are supported. The rotation algorithm used in this code is different from `HPXconvert_cxx` and works as follows:
  - extract a_{lm} from the input map, using an iterative scheme with 3 iterations, up to $l_{\max} = 3N_{\mathrm{side}}$
  - perform the rotation in a_{lm} space
  - convert the rotated a_{lm} back to a HEALPix map

  This code is rather resource-consuming, but much more accurate than `HPXconvert_cxx`.

```
Usage: rotmap_cxx <infile> <outfile> <itransform> <pol>
Transform 1: Equatorial (2000) -> Galactic   (2000)
          2: Galactic   (2000) -> Equatorial (2000)
          3: Equatorial (2000) -> Ecliptic   (2000)
          4: Ecliptic   (2000) -> Equatorial (2000)
          5: Ecliptic   (2000) -> Galactic   (2000)
          6: Galactic   (2000) -> Ecliptic   (2000)
          7: Equatorial (1950) -> Galactic   (1950)
          8: Galactic   (1950) -> Equatorial (1950)
          9: Equatorial (1950) -> Ecliptic   (1950)
         10: Ecliptic   (1950) -> Equatorial (1950)
         11: Ecliptic   (1950) -> Galactic   (1950)
         12: Galactic   (1950) -> Ecliptic   (1950)

pol: T or F
```

- `rotalm_cxx`: Converts a_{lm} between different coordinate systems.

```
Usage: rotalm_cxx <infile> <outfile> <itransform> <pol>
Transform 1: Equatorial (2000) -> Galactic   (2000)
          2: Galactic   (2000) -> Equatorial (2000)
          3: Equatorial (2000) -> Ecliptic   (2000)
          4: Ecliptic   (2000) -> Equatorial (2000)
          5: Ecliptic   (2000) -> Galactic   (2000)
          6: Galactic   (2000) -> Ecliptic   (2000)
          7: Equatorial (1950) -> Galactic   (1950)
          8: Galactic   (1950) -> Equatorial (1950)
          9: Equatorial (1950) -> Ecliptic   (1950)
         10: Ecliptic   (1950) -> Equatorial (1950)
         11: Ecliptic   (1950) -> Galactic   (1950)
         12: Galactic   (1950) -> Ecliptic   (1950)

pol: T or F
```

- `fpdbhelper`: Command-line tool to retrieve data from the focal plane database. Mainly intended for automatic parameter file generation.

```
Usage: fpdbhelper <database> <detector> <quantity>

<database>: name of the focalplane data base file

<detector>: name of the required detector

<quantity>: determines the output of the program
  fwhm_arcmin   : FWHM in arc minutes
  freq_GHz      : central frequency in GHz
                  (rounded to the nearest integer)
  lmax          : a hint at a suitable lmax parameter for this detector
  ringres       : number of samples taken per minute
                  (rounded to the nearest integer)
  f_samp        : sampling frequency (in Hz)
  alpha         : slope of the noise spectrum (as a positive number)
  f_knee        : knee frequency (in Hz)
  f_min         : minimum noise frequency (in Hz)
  net_rj        : noise equivalent antenna temperature (in K/sqrt(Hz))
  tau_bol       : bolometer time constant (in s)
  tau_int       : integration time for each sample (in s)
  ellipticity   : (max FWHM)/(min FWHM)
  theta_uv_deg  : the theta_uv angle (in degrees)
  phi_uv_deg    : the phi_uv angle (in degrees)
  psi_uv_deg    : the psi_uv angle (in degrees)
  psi_pol_deg   : the psi_pol angle (in degrees)
  psi_ell_deg   : the psi_ell angle (in degrees)
```

- `makeMap`: Makes a map out of sampled detector pointings and `TOD`.

```
makeMap [TOD] [DETPNT] [baseoutname] [RES]
```

- `addTOI`: (Davide Maino, Martin Reinecke) Coadd a number of different `TOI`.

```
usage: addTOI <init object>
or:    addTOI <file1> <factor1> [<file2> <factor2>] [...] [output file]

Parameters read by addTOI:

toi[x] (string):
  input file name for TOI stream [x]

factor[x] (double):
  multiplication factor for TOI stream [x]

In the descriptions above, [x] is an integer without leading zeroes.
For n components, each component must have a unique number from the
range [1;n].
```

```
outfile (string):
  output file name for coadded TOI stream
```

- addMaps: (Davide Maino, Martin Reinecke) Coadd a number of different Healpix maps.

```
usage: addMaps <init object>
or:    addMaps <file1> <factor1> [<file2> <factor2>] [...] [output file]

Parameters read by addMaps:

map[x] (string):
  input file name for map [x]

factor[x] (double):
  multiplication factor for map [x]

In the descriptions above, [x] is an integer without leading zeroes.
For n components, each component must have a unique number from the
range [1;n].

outfile (string):
  output file name for coadded map
```

- beamsampler: (Martin Reinecke) Smear a set of beam a_{lm} to simulate the effect of the detector electronics.

```
Parameters read by beamsampler:

focalplane_db (string):
  input file containing the detector database

detector_id (string):
  name of the detector

beam_in (string):
  input object containing the beam a_lm

beam_out (string):
  output object containing the smeared beam a_lm

n_integ (int, dafault=5):
  number of samples to use for time constant integration
  (only relevant for HFI detectors)

sat_rpm (double, default=1):
  number of satellite revolutions per minute
```

```
mmax_out (int):
  maximum m quantum number for the output beam
```

- `sat2quat`: (Martin Reinecke) Converts the detailed satellite pointings produced by `simmission` to quaternions.

  ```
  Parameters read by sat2quat:

  infile (string):
    input object containing the satellite pointing information.
    This object should have been created by simmission with
    "simulatepointings = Y".

  outfile (string):
    output object containing the quaternions that correspond to the detailed
    satellite pointings.
  ```

- `pointing_errors`: (Martin Reinecke) Adds random perturbations to satellite pointing quaternions.

  ```
  Parameters read by pointing_errors:

  infile (string):
    input object containing satellite pointing quaternions.

  outfile (string):
    output object containing the perturbed quaternions.

  rand_seed (integer):
    random-number seed

  sigma (double):
    RMS sigma of the perturbation (in radians)
  ```

- `gapmaker`: (Martin Reinecke) Introduces gaps into time-ordered information produced by `multimod`; this can be done either by cutting out data (for LFI) or flagging data as non-existent (for HFI).

  ```
  Parameters read by gapmaker:

  focalplane_db (string):
    input file containing the detector database

  detector_id (string):
    name of the detector

  sat_info (string):
    Name of the input file containing the satellite information.
  ```

```
first_pointing (int):
  number of the first pointing period to calculate. Default is 1,
  which is also used when "-1" is given.

last_pointing (int):
  number of the last pointing period to calculate. Default is the last
  period in the satellite information file, which is also used when "-1"
  is given.

gapfile (string):
  input object containing the gap information

flagfile (string, default=""):
  if not empty, the flag information is written to this object

timestamp_in (string, default=""):
  if not empty, contains the input object with timestamp data

if (timestamp_in!="")
  timestamp_out (string):
    output object containing the compacted timestamp data
endif

toi_in (string, default=""):
  if not empty, contains the input object with TOI data

if (toi_in!="")
  toi_out (string):
    output object containing the compacted TOI data
endif

detpt_in (string, default=""):
  if not empty, contains the input object with detector pointings

if (detpt_in!="")
  detpt_out (string):
    output object containing the compacted detector pointings
endif

quaternions_in (string, default=""):
  if not empty, contains the input object with quaternions

if (quaternions_in!="")
  quaternions_out (string):
    output object containing the compacted quaternions
endif
```

- `mult_alm`: (Martin Reinecke) Allows convolution and deconvolution of a_{lm} with Gaussian beams and/or pixel window functions.

```
Parameters read by mult_alm:

infile (string):
  input file containing the a_lm

outfile (string):
  output file name for the calculated a_lm

fwhm_arcmin_in (real, default=0):
  FWHM (in arcmin) of a Gaussian beam, which will be _removed_ from the
  input a_lm

fwhm_arcmin_out (real, default=0):
  FWHM (in arcmin) of a Gaussian beam, which will be used to smoothe the
  output a_lm

pixwin_in (string, default=""):
  if supplied, the pixel window function from this file will be _removed_
  from the input a_lm

pixwin_out (string, default=""):
  if supplied, the pixel window function from this file will be applied
  to the output a_lm

cl_in (string, default=""):
  if supplied, the power spectrum from this file will be _removed_
  from the input a_lm
  NOTE: currently only supported for unpolarised a_lm

cl_out (string, default=""):
  if supplied, the power pectrum from this file will be applied
  to the output a_lm
  NOTE: currently only supported for unpolarised a_lm

polarisation (bool):
  if false, only the intensity a_lm are generated,
  if true, T, G and C a_lm are generated

double_precision (bool, default=false):
  if false, the a_lm are read/written in single precision,
  otherwise in double precision.
```

4.3.12 LFI-specific modules

- `ls2lfitoi` (D. Maino, UniMI):

  `ls2lfitoi` is a C++ module which produces un-differenced data (i.e. sky and load signals) according to LFI DDL.

  ```
  ls2lfitoi [parameter file]
  ```

```
Parameters read by ls2lfitoi:

therm_sky_file (string, default="")
 if not empty Sky thermal fluctuations TOD is read

therm_load_file (string, default="")
 if not empty Load thermal fluctuations TOD is read

fspike_file (string, default="")
 if not empty frequency spikes TOD is read

time_file (string, default="")
 if not empty toi.LS_TimeTOD object with time stamp is read

telescope_file (string, default="")
  if not empty toi.LS_toi with fluctuations of telescope temperature
  sampled at the right detector sampling frequency are read

tref_temp (double, default=4.5):
 reference load radiometric temperature (in K). In the current
 implementation it is assumed to be constant. Usually around 4.5 K
 this is the zero level for the load fluctuations. Be sure to
 input here the proper zero level if it is different.

t_telescope (double, default=42):
  reference telescope temperature in K.

emissivity (double, default=1):
  emissivity of the telescope.

has_monopole (bool):
   boolean variable to determines whether input simulated signal
   toi contains the CMB monopole

detector_id (string):
  detector identifier (e.g. LFI-28S)

tsky (string):
  toi as output from multimod (expected toi.LS_toi) with ONLY signal

tsky_noise (string):
  toi with pure white noise with the expected value of instrument sensitivity

tref_noise (string):
  another independent toi with pure white noise with the
  expected value of instrument sensitivity

oof_noise (string):
  toi with pure 1/f noise with properties expected in the
  differenced (i.e. T_sky - R * T_load) data
```

```
oof_noise_tp (string):
  toi with pure 1/f noise with properties expected in the
  Total Power (un-differenced) data. Usually knee-frequency
  is around few x 10 Hz

file_toi (string):
  output toi.science.LFI_Data with correct data structure

focalplane_db (string):
  location of the instrument database

verbosity (bool, default=false):
  enable verbose messages to the console?
```

The way in which Sky and Load data are constructed has been subject of discussion within the LFI-Demo group. Here is the algorithm as it is implemented now.

First of all one defined the "target" value of the gain modulation factor R parameter as

$$R_{\rm target} = \frac{\langle T_{\rm sky}\rangle + T_{\rm noise}}{T_{\rm load} + T_{\rm noise}}\,. \tag{3}$$

Once this is done one constructs the signal from Sky and Load with the following expressions:

$$V_{\rm sky} = {\rm signalTOI} + T_{0,CMB} + \frac{\rm wn1}{\sqrt{2}} + {\rm ooftp} + {\rm oof}/2 + T_{\rm noise} \tag{4}$$

$$V_{\rm load} = T_{\rm load} + \frac{\rm wn2}{\sqrt{2}} + \frac{{\rm ooftp} - {\rm oof}/2}{R_{\rm target}} + T_{\rm noise} \tag{5}$$

One would like to derive the value of the R parameter in order to construct the differenced data usually taking the ratio of the mean values of $V_{\rm sky}$ and $V_{\rm load}$ on suitable time lag and then build the quantity $V_{\rm sky} - R_{\rm estimated} V_{\rm load}$. With the actual recipe this quantity contains the correct value of the signal, the correct amount of white noise and only the differenced part of $1/f$ noise since the total power part is suppressed by a factor proportional to the difference between the target and the estimated value of R.

- `quantum` (Michele Maris, OAT, and Davide Maino, UniMI):

  `quantum` is a C++ module which simulates the on-board processing of PType 2 data and the subsequent decoding and conversion to TOI in the DPC L1. In particular it simulates the activity of PType 2 processing from PType 1 data. The code accepts data in input of toi.science.Data type specified in the LFI DDL (and built-in into the delivered DDL with LevelS from ESTEC CVS). It is able to reproduce the creation of two sets of differenced data with two different and known value of the gain modulation factors (gmf1 and gmf2). This is done in order to satisfy the constraints related to transmission band-width from S/C to ground station.

```
quantum [parameter file]

Parameters read by quantum:

fnamein (string):
  input toi.science.LFI_Data (un-differenced data) in ADU to be quantized
```

```
fnameout (string):
  output toi.science.LFI_Data (un-differenced data) in ADU after quantization
  and reconstruction has been performed

n_aver (int, default=1):
  number of average of samples. If data come from LevelS or from TMH
  (LFI-L1) use default value

offset (float):
  offset to be applied to differenced data (with gmf1 and gmf2)

quant (float):
  quantization factor (in units of sigma of white noise)

gmf1 (float):
  On-board value of gain modulation factor to produce differenced
  data to satisfy band-width constraints

gmf2 (float):
  The other on-board value of gain modulation factor to
  produce differenced data.
```

- `ahf2satpt` (Martin Reinecke, MPA):

  This module reads a set of AHF objects with the DDL type `toi.attitude.HighFrequency`, extracts the satellite pointing information from these and writes it into a single new object of type `sat.LS_satpoint_real`. This object contains a list of all input satellite quaternions and their time stamps, as well as a list of name, start time, end time, index of the first quaternion and total number of quaternions for every pointing period found in the input AHFs.

```
ahf2satpt [parameter file]

Parameters read by ahf2satpt:

input_list (string):
  name of the input text file containing a list of AHF object names to read

outfile (string):
  name of the output object (DDL type "sat.LS_satpoint_real")

outfile_wobble (string, default=""):
  if not empty, name of the output object containing tilt angle information
  (DDL type "sat.LS_tiltAngles")

first_period_number (int, default=1):
  number used for the first pointing period in the output file
  (this can be used to synchronize numbers with LFI)
```

A LevelS dataflow

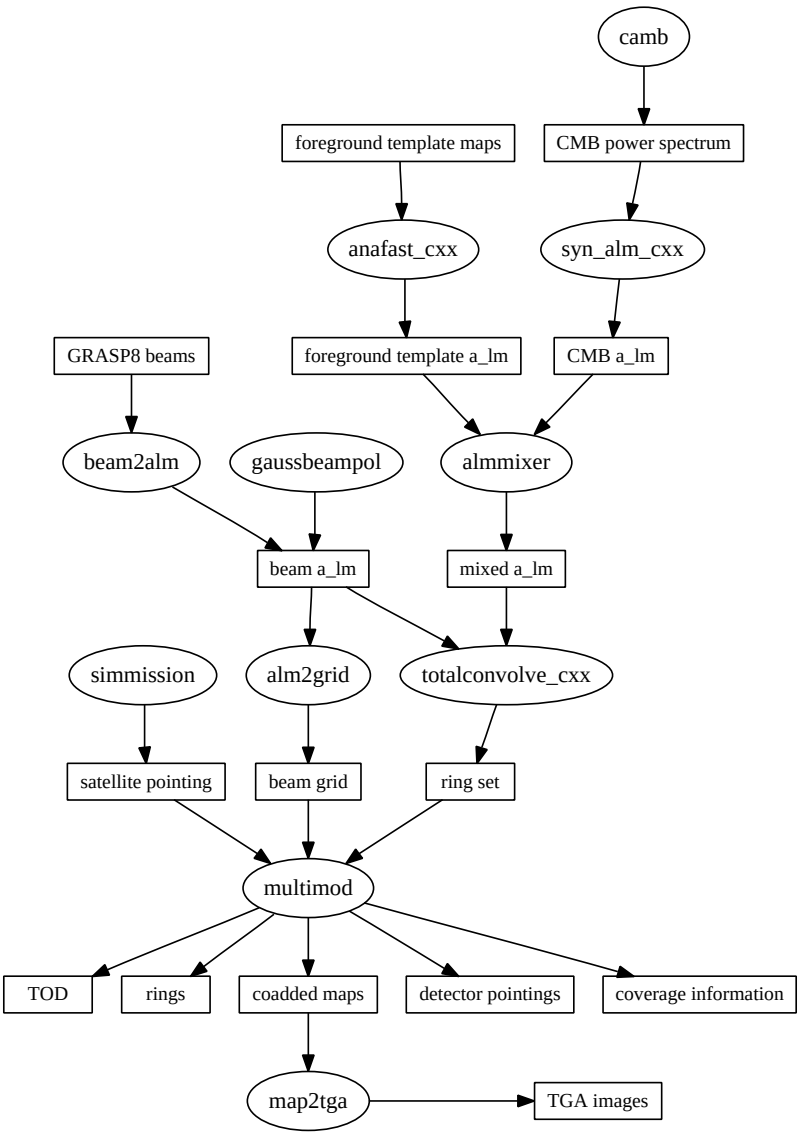

Figure 2: Schematic dataflow of a typical *Planck* simulation pipeline. Rectangular components denote data products, whereas elliptic shapes represent modules.

B Detector Database

For reference, we reproduce here the detector database:

```
# Planck Level-S Focalplane Database
# ==================================
#
# Compiled by Mark Ashdown (MAJA) <maja1@mrao.cam.ac.uk>
#
# 2009/03/27
#
  Version = 6.1
#
# Based on version 1.0 compiled by Bob Mann (RGM) <rgm@roe.ac.uk>,
# in turn based on an original module by Ian Grivell (IJGR).
#
# with information from
#
# Marco Bersanelli (MB) <marco@mi.iasf.cnr.it>
# Daniele Mennella (DM) <daniele@mi.iasf.cnr.it>
# Jean-Michel Lamarre (JML) <lamarre@ias.fr>
# Michel Piat (MP) <piat@ias.fr>
# Carlo Burigana (CB) <burigana@bo.iasf.cnr.it>
# Vladimir Yurchenko (VY) <v.yurchenko@may.ie>
# Simon Prunet (SP) <prunet@iap.fr>
#
# References
# ----------
#
# [1] "Planck/LFI: Main Beam Locations and Polarization Alignment for
#     the LFI Baseline FPU" (PL-LFI-PST-TN-027, v1.0), M. Sandri and
#     F. Villa, 2001/07. Communicated to MAJA by MB.
#
# [2] "HFI Instrument data for simulations" (version MIG 3.1), JML on
#     behalf of the Instrument Working Group, 2003/07/31. Communicated
#     to MAJA by JML.
#
# [3] CB email to RGM, 2000/05/05.
#
# [4] MP email to RGM, 2000/05/09.
#
# [5] MB email to MAJA, 2003/07/24.
#
# [6] DM email to MAJA, 2003/07/24.
#
# [7] "Planck-LFI On-board data handling and compression", LFI instrument
#     team, 1999/06/04.
#
# [8] HFI Proposal, Chapter 3.
#
# [9] "Gaussian Fitting Parameters of the ESA Planck HFI beams",
#     V. Yurchenko, J. A. Murphy, J.-M. Lamarre and J. Brossard,
#     preprint of paper to appear in Int. J. Infrared and Millimetre
#     Waves (vol. 25, no. 5, 2004/04). Communicated to MAJA by VY.
#
# [10] "LFI Main beams at 30 GHz" (PL-LFI-PST-TN-040, v2.0), M. Sandri
#      and F. Villa, 2005/02.
#
# [11] "LFI Main beams at 44 GHz" (PL-LFI-PST-TN-061, v1.0), M. Sandri
#      and F. Villa, 2005/02.
#
# [12] "LFI Main beams at 70 GHz" (PL-LFI-PST-TN-062, v1.0), M. Sandri
#      and F. Villa, 2005/02.
#
# [13] "Optical Effects (LFI)", F. Villa, presentation at the Planck
#      HFI/LFI Consortia Meeting, Garching 26th-28th January, 2005
#      (available in Livelink).
#
# [14] "A brief summary of HFI noise sensitivities: from NEPs to NETs",
#      S. Prunet, 2005/01. Communicated to MAJA by SP.
#
# [15] P. Natoli email to T. Poutanen, 2005/04/01. Communicated to MAJA
#      by T. Poutanen.
#
# Columns
# -------
#
# (1) detector ID
#
#   LFI: detector IDs of the form LFI-xxS and LFI-xxM, where xx is the horn
#        number in [1]. 100 GHz channel (horns 2-17) was removed.
#
#   HFI: detector IDs of the form xxx-yy where xxx is the band name and
#        yy the horn number in that band. PSBs have a or b appended to the
#        horn number (for example, 100-1a) [9].
#
# (2) phi_uv   / degrees
# (3) theta_uv / degrees
# (4) psi_uv   / degrees
#
#   These angles describe the position of the detectors in the focal plane.
#   They give the rotation of the beam pattern from a fiducial orientation
#   (forward beam direction (z-axis) pointing along the telescope line of
#   sight, with y-axis aligned with the nominal scan direction) to their
#   positions in the focal plane [1, 10, 11, 12]. HFI values from [9],
#   converted to this coordinate system by MAJA.
#
# (5) psi_pol / degrees (angle of polarisation measurement)
# (6) epsilon (degree of cross-polar leakage)
```

```
#
#   The polarisation angle psi_pol is defined with the beam in the fiducial
#   orientation described above, that is, before rotation onto the detector
#   position [9, 13]. The degree of cross-polar leakage would be 0.0 for an
#   idealised polarised detector and 1.0 for an idealised unpolarised
#   detector.
#
# (7) nu_cen / Hz (central frequency)
# (8) nu_min / Hz (minimum frequency)
# (9) nu_max / Hz (maximum frequency)
#
#   Top hat spectral response assumed for all detectors, with
#   delta_nu/nu_cen = 0.2 for LFI [3], and delta_nu/nu_cen =
#   0.33 for HFI [4].
#
# (10) f_knee / Hz (knee frequency)
#
#   LFI: 0.05 Hz for 30 GHz channel [15]. Adopted for all LFI channels
#        until further information is available.
#
#   HFI: 0.03 Hz for all channels, worst case [2].
#
# (11) alpha (power law index for low-frequency noise)
#
#   LFI: alpha = 1.7 for 30 GHz channel [15]. Adopted for all LFI
#        channels until further information is available.
#
#   HFI: alpha = 2.0 [2].
#
# (12) f_min / Hz (cut-off frequency below which there are no
#                     correlations in the noise)
#
#   LFI and HFI: arbitrarily set to 1.15e-5 Hz (approximately 1/(1 day))
#
# (13) f_samp / Hz (frequency at which science data samples are produced)
#
#   LFI: 32.5 Hz at 30 GHz, 45.0 Hz at 44 GHz and 76.8 Hz at 70 GHz [6].
#
#   HFI: 200 Hz for all channels [2].
#
# (14) tau_bol / s (time constant for detector response)
#
#   LFI: 0.0 - instantaneous response for LFI detectors.
#
#   HFI: 7.8e-3 s for HFI 100GHz, 5.8e-3 s for 143GHz, 4.4e-3 s for
#        217-857GHz [2].
#
# (15) tau_int / s (integration time for each output sample)
#
#   LFI: tau_integ = 1/f_samp [3]. Changed to reflect new sampling
#        frequencies by MAJA in version 2.0.
#
#   HFI: tau_integ = 1/f_samp [2].
#
# (16) n_read (number of "fast samples" per output sampling period)
#
#   LFI: 8 [7].
#
#   HFI: 32 [8]. Need to check this, because [2] has this as N/A.
#
# (17) fwhm_beam / arcmin (mean FWHM of beam)
# (18) ellipticity (max FWHM / min FWHM)
# (19) psi_ell / degrees (ellipse orientation)
#
#   Mean FWHM of the beam is calculated as the geometric mean of maximum
#   and minimum FWHMs, fwhm_beam = sqrt(fwhm_max*fwhm_min). This is so
#   that a circular beam with FWHM = fwhm_beam has the same area as the
#   elliptical beam.
#
#   psi_ell is the orientation of the beam major axis, defined in the
#   fiducial orientatation described above, that is, before rotating by
#   (phi_uv, theta_uv, psi_uv) onto its position in the focal plane.
#
#   LFI: Values from [10, 11, 12].
#
#   HFI: Values from [9].
#
# (20) NET_RJ / K sqrt(s) (Noise equivalent temperature in Raleigh-Jeans units)
#
#   Unified noise measure introduced in version 3 of this database. To get the
#   RMS of the noise per sample, this should be multiplied by sqrt(f_samp):
#
#   sigma = NET_RJ * sqrt(f_samp)
#
#   LFI: Antenna temperatures of (9.87, 14.18, 22.75) K at
#        (30, 44, 70) GHz [6]. This is the sum of the system antenna
#        temperature and the CMB antenna temperature (2.73K). Converted
#        to NET_RJ by MAJA using formula:
#
#        NET_RJ = T_ann * sqrt(2/bandwidth)
#
#   HFI: NEPs from [2] converted to NET_RJ using formula given in [14]
#        (without factor of 1+P):
#
#        NET_RJ = NEP * (1+P) / (2*sqrt(2)*k*n*bandwidth)
#
#        where P = 0, 1 for unpolarised and polarised detectors
#        respectively, k is the Boltzmann constant and n is the number
#        of modes (1 for the monomode channels 100-353 GHz, 4 for 545 GHz
#        and 8 for 857 GHz).
```

```
#
# (21) SLDP_X  X component of sidelobe dipole vector, (dimensionless)
# (22) SLDP_Y  Y component of sidelobe dipole vector, (dimensionless)
# (23) SLDP_Z  Z component of sidelobe dipole vector, (dimensionless)
#
#      The reference frame for the sidelobe dipole vector is the one defined
#      by the three angles theta_uv, phi_uv, psi_uv for each detector. [16]
#      LFI values from [16], based on 4 pi beams calculated by M. Sandri
#      HFI values temporarily set to zero.
#
100-1a   -141.5330  1.8994  -90.0878 112.5000 0.0000 100.0e9  83.3e9 116.7e9 0.03 2.0 1.15e-5 200.0 7.80e-3  5.0e-3 32  9.6491  1.1582   11.2067 8.800e-5  0.000000   0.000000   0.000000
100-1b   -141.5104  1.8998  -90.9968  22.5000 0.0000 100.0e9  83.3e9 116.7e9 0.03 2.0 1.15e-5 200.0 7.80e-3  5.0e-3 32  9.6432  1.1912   15.0427 8.800e-5  0.000000   0.000000   0.000000
100-2a   -167.7324  1.8003  -91.1133 135.0000 0.0000 100.0e9  83.3e9 116.7e9 0.03 2.0 1.15e-5 200.0 7.80e-3  5.0e-3 32  9.6551  1.1544    0.6475 8.800e-5  0.000000   0.000000   0.000000
100-2b   -167.7427  1.8010  -90.3994  45.0000 0.0000 100.0e9  83.3e9 116.7e9 0.03 2.0 1.15e-5 200.0 7.80e-3  5.0e-3 32  9.6262  1.1490    5.9453 8.800e-5  0.000000   0.000000   0.000000
100-3a    167.7396  1.8010  -90.3452  90.0000 0.0000 100.0e9  83.3e9 116.7e9 0.03 2.0 1.15e-5 200.0 7.80e-3  5.0e-3 32  9.6313  1.1281    0.0000 8.800e-5  0.000000   0.000000   0.000000
100-3b    167.7378  1.8011  -89.6008   0.0000 0.0000 100.0e9  83.3e9 116.7e9 0.03 2.0 1.15e-5 200.0 7.80e-3  5.0e-3 32  9.6431  1.1617   -3.0440 8.800e-5  0.000000   0.000000   0.000000
100-4a   141.5259  1.8996  -89.5992  67.5000 0.0000 100.0e9  83.3e9 116.7e9 0.03 2.0 1.15e-5 200.0 7.80e-3  5.0e-3 32  9.6651  1.1533  -11.6036 8.800e-5  0.000000   0.000000   0.000000
100-4b   141.5104  1.8998  -89.0031 157.5000 0.0000 100.0e9  83.3e9 116.7e9 0.03 2.0 1.15e-5 200.0 7.80e-3  5.0e-3 32  9.6438  1.1912  -15.0650 8.800e-5  0.000000   0.000000   0.000000
143-1a   -48.6975  1.8152  -90.8900 135.0000 0.0000 143.0e9 119.2e9 166.8e9 0.03 2.0 1.15e-5 200.0 5.80e-3  5.0e-3 32  7.0508  1.0585   48.9319 5.329e-5  0.000000   0.000000   0.000000
143-1b   -48.6981  1.8150  -89.2503  45.0000 0.0000 143.0e9 119.2e9 166.8e9 0.03 2.0 1.15e-5 200.0 5.80e-3  5.0e-3 32  7.0667  1.1089   46.2298 5.329e-5  0.000000   0.000000   0.000000
143-2a   -24.5836  1.3465  -89.1450 135.0000 0.0000 143.0e9 119.2e9 166.8e9 0.03 2.0 1.15e-5 200.0 5.80e-3  5.0e-3 32  6.9862  1.0241   68.7113 5.329e-5  0.000000   0.000000   0.000000
143-2b   -24.5788  1.3474  -89.5491  45.0000 0.0000 143.0e9 119.2e9 166.8e9 0.03 2.0 1.15e-5 200.0 5.80e-3  5.0e-3 32  6.9695  1.0692   56.2022 5.329e-5  0.000000   0.000000   0.000000
143-3a    25.0383  1.3237  -89.4459  90.0000 0.0000 143.0e9 119.2e9 166.8e9 0.03 2.0 1.15e-5 200.0 5.80e-3  5.0e-3 32  6.9869  1.0533  -68.9327 5.329e-5  0.000000   0.000000   0.000000
143-3b    25.0484  1.3243  -90.4489   0.0000 0.0000 143.0e9 119.2e9 166.8e9 0.03 2.0 1.15e-5 200.0 5.80e-3  5.0e-3 32  6.9772  1.0308  -38.0312 5.329e-5  0.000000   0.000000   0.000000
143-4a    49.5255  1.8839  -90.3711  90.0000 0.0000 143.0e9 119.2e9 166.8e9 0.03 2.0 1.15e-5 200.0 5.80e-3  5.0e-3 32  7.0756  1.0931  -56.4746 5.329e-5  0.000000   0.000000   0.000000
143-4b    49.5465  1.8841  -90.7711   0.0000 0.0000 143.0e9 119.2e9 166.8e9 0.03 2.0 1.15e-5 200.0 5.80e-3  5.0e-3 32  7.0892  1.0759  -38.9115 5.329e-5  0.000000   0.000000   0.000000
143-5    -33.2329  2.0742  -91.4539   0.0000 1.0000 143.0e9 119.2e9 166.8e9 0.03 2.0 1.15e-5 200.0 5.80e-3  5.0e-3 32  7.3108  1.1664   60.1933 3.370e-5  0.000000   0.000000   0.000000
143-6     -9.5895  1.7855  -89.0723   0.0000 1.0000 143.0e9 119.2e9 166.8e9 0.03 2.0 1.15e-5 200.0 5.80e-3  5.0e-3 32  7.2464  1.1107   81.2338 3.370e-5  0.000000   0.000000   0.000000
143-7      9.7212  1.7611  -89.6811   0.0000 1.0000 143.0e9 119.2e9 166.8e9 0.03 2.0 1.15e-5 200.0 5.80e-3  5.0e-3 32  7.2368  1.1113  -78.6064 3.370e-5  0.000000   0.000000   0.000000
143-8     32.8515  2.0941  -90.1783   0.0000 1.0000 143.0e9 119.2e9 166.8e9 0.03 2.0 1.15e-5 200.0 5.80e-3  5.0e-3 32  7.3742  1.1496  -60.4363 3.370e-5  0.000000   0.000000   0.000000
217-1    -134.7711  1.3990  -89.5414   0.0000 1.0000 217.0e9 180.8e9 253.2e9 0.03 2.0 1.15e-5 200.0 4.40e-3  5.0e-3 32  4.7263  1.1280   12.8744 3.041e-5  0.000000   0.000000   0.000000
217-2    -161.8816  1.0093  -90.7004   0.0000 1.0000 217.0e9 180.8e9 253.2e9 0.03 2.0 1.15e-5 200.0 4.40e-3  5.0e-3 32  4.7055  1.1184    4.0827 3.041e-5  0.000000   0.000000   0.000000
217-3    162.3210  1.0331  -90.2807   0.0000 1.0000 217.0e9 180.8e9 253.2e9 0.03 2.0 1.15e-5 200.0 4.40e-3  5.0e-3 32  4.7011  1.1302   -4.3018 3.041e-5  0.000000   0.000000   0.000000
217-4    134.0357  1.3811  -89.6330   0.0000 1.0000 217.0e9 180.8e9 253.2e9 0.03 2.0 1.15e-5 200.0 4.40e-3  5.0e-3 32  4.7247  1.1482  -13.3840 3.041e-5  0.000000   0.000000   0.000000
217-5a   -111.6191  1.3139  -89.7511 135.0000 0.0000 217.0e9 180.8e9 253.2e9 0.03 2.0 1.15e-5 200.0 4.40e-3  5.0e-3 32  4.7269  1.1114   21.0541 4.307e-5  0.000000   0.000000   0.000000
217-5b   -111.6166  1.3137  -90.5394  45.0000 0.0000 217.0e9 180.8e9 253.2e9 0.03 2.0 1.15e-5 200.0 4.40e-3  5.0e-3 32  4.7370  1.1165   19.4575 4.307e-5  0.000000   0.000000   0.000000
217-6a   -130.2269  0.7089  -90.4651 135.0000 0.0000 217.0e9 180.8e9 253.2e9 0.03 2.0 1.15e-5 200.0 4.40e-3  5.0e-3 32  4.7192  1.0955   11.4913 4.307e-5  0.000000   0.000000   0.000000
217-6b   -130.2185  0.7091  -90.3907  45.0000 0.0000 217.0e9 180.8e9 253.2e9 0.03 2.0 1.15e-5 200.0 4.40e-3  5.0e-3 32  4.7148  1.0904    7.9065 4.307e-5  0.000000   0.000000   0.000000
217-7a    131.7249  0.7255  -90.3246  90.0000 0.0000 217.0e9 180.8e9 253.2e9 0.03 2.0 1.15e-5 200.0 4.40e-3  5.0e-3 32  4.7123  1.0966   -8.8825 4.307e-5  0.000000   0.000000   0.000000
217-7b    131.7354  0.7255  -89.5987   0.0000 0.0000 217.0e9 180.8e9 253.2e9 0.03 2.0 1.15e-5 200.0 4.40e-3  5.0e-3 32  4.7116  1.0789   -8.8843 4.307e-5  0.000000   0.000000   0.000000
217-8a    110.5930  1.3045  -89.5638  90.0000 0.0000 217.0e9 180.8e9 253.2e9 0.03 2.0 1.15e-5 200.0 4.40e-3  5.0e-3 32  4.7438  1.1353  -19.9173 4.307e-5  0.000000   0.000000   0.000000
217-8b    110.5930  1.3045  -89.4804   0.0000 0.0000 217.0e9 180.8e9 253.2e9 0.03 2.0 1.15e-5 200.0 4.40e-3  5.0e-3 32  4.7442  1.1228  -21.1648 4.307e-5  0.000000   0.000000   0.000000
353-1     -89.4780  2.0552  -90.2047   0.0000 1.0000 353.0e9 294.2e9 411.8e9 0.03 2.0 1.15e-5 200.0 4.40e-3  5.0e-3 32  4.5781  1.2648   20.0845 2.124e-5  0.000000   0.000000   0.000000
353-2     -88.1479  1.4147  -90.0769   0.0000 1.0000 353.0e9 294.2e9 411.8e9 0.03 2.0 1.15e-5 200.0 4.40e-3  5.0e-3 32  4.5025  1.1617   20.6846 2.124e-5  0.000000   0.000000   0.000000
353-3a    -88.5553  0.8214  -90.0507 135.0000 0.0000 353.0e9 294.2e9 411.8e9 0.03 2.0 1.15e-5 200.0 4.40e-3  5.0e-3 32  4.4711  1.1746   27.5055 3.002e-5  0.000000   0.000000   0.000000
353-3b    -88.5417  0.8214  -90.0505  45.0000 0.0000 353.0e9 294.2e9 411.8e9 0.03 2.0 1.15e-5 200.0 4.40e-3  5.0e-3 32  4.4776  1.0968    0.8876 3.002e-5  0.000000   0.000000   0.000000
353-4a    -77.6040  0.2140  -89.9728 135.0000 0.0000 353.0e9 294.2e9 411.8e9 0.03 2.0 1.15e-5 200.0 4.40e-3  5.0e-3 32  4.4909  1.1012   27.1548 3.002e-5  0.000000   0.000000   0.000000
353-4b    -77.5837  0.2158  -89.9734  45.0000 0.0000 353.0e9 294.2e9 411.8e9 0.03 2.0 1.15e-5 200.0 4.40e-3  5.0e-3 32  4.4934  1.0795  -21.7530 3.002e-5  0.000000   0.000000   0.000000
353-5a    86.6818  0.3680  -89.9467  90.0000 0.0000 353.0e9 294.2e9 411.8e9 0.03 2.0 1.15e-5 200.0 4.40e-3  5.0e-3 32  4.4923  1.1367   -4.8728 3.002e-5  0.000000   0.000000   0.000000
353-5b    86.6965  0.3674  -89.9889   0.0000 0.0000 353.0e9 294.2e9 411.8e9 0.03 2.0 1.15e-5 200.0 4.40e-3  5.0e-3 32  4.4831  1.0259  -45.3774 3.002e-5  0.000000   0.000000   0.000000
353-6a    87.2541  0.9625  -89.9620  90.0000 0.0000 353.0e9 294.2e9 411.8e9 0.03 2.0 1.15e-5 200.0 4.40e-3  5.0e-3 32  4.4549  1.1777  -14.6236 3.002e-5  0.000000   0.000000   0.000000
353-6b    87.2719  0.9625  -89.9617   0.0000 0.0000 353.0e9 294.2e9 411.8e9 0.03 2.0 1.15e-5 200.0 4.40e-3  5.0e-3 32  4.5360  1.0828  -29.3738 3.002e-5  0.000000   0.000000   0.000000
353-7     89.2129  1.5183  -89.8905   0.0000 1.0000 353.0e9 294.2e9 411.8e9 0.03 2.0 1.15e-5 200.0 4.40e-3  5.0e-3 32  4.5274  1.1678  -19.7471 2.124e-5  0.000000   0.000000   0.000000
353-8     88.7671  2.0555  -89.8680   0.0000 1.0000 353.0e9 294.2e9 411.8e9 0.03 2.0 1.15e-5 200.0 4.40e-3  5.0e-3 32  4.5704  1.2515  -22.1910 2.124e-5  0.000000   0.000000   0.000000
545-1     -75.3744  2.1308  -90.7539   0.0000 1.0000 545.0e9 454.3e9 635.8e9 0.03 2.0 1.15e-5 200.0 4.40e-3  5.0e-3 32  4.2239  1.1954   33.5781 1.095e-5  0.000000   0.000000   0.000000
545-2     -68.1972  1.5297  -89.5933   0.0000 1.0000 545.0e9 454.3e9 635.8e9 0.03 2.0 1.15e-5 200.0 4.40e-3  5.0e-3 32  3.8514  1.2464   30.0005 1.095e-5  0.000000   0.000000   0.000000
545-3     70.4151  1.6152  -89.3423   0.0000 1.0000 545.0e9 454.3e9 635.8e9 0.03 2.0 1.15e-5 200.0 4.40e-3  5.0e-3 32  3.9373  1.1067  -32.3890 1.095e-5  0.000000   0.000000   0.000000
545-4     74.7297  2.1372  -90.3269   0.0000 1.0000 545.0e9 454.3e9 635.8e9 0.03 2.0 1.15e-5 200.0 4.40e-3  5.0e-3 32  4.2413  1.1605  -37.8301 1.095e-5  0.000000   0.000000   0.000000
857-1     -57.2537  0.9982  -90.6050   0.0000 1.0000 857.0e9 714.2e9 999.9e9 0.03 2.0 1.15e-5 200.0 4.40e-3  5.0e-3 32  4.3246  1.0982   58.3461 5.080e-6  0.000000   0.000000   0.000000
857-2     -21.9841  0.6085  -89.5498   0.0000 1.0000 857.0e9 714.2e9 999.9e9 0.03 2.0 1.15e-5 200.0 4.40e-3  5.0e-3 32  4.2908  1.0132   86.0246 5.080e-6  0.000000   0.000000   0.000000
857-3     32.9765  0.6438  -89.7657   0.0000 1.0000 857.0e9 714.2e9 999.9e9 0.03 2.0 1.15e-5 200.0 4.40e-3  5.0e-3 32  4.2869  1.0514  -71.9764 5.080e-6  0.000000   0.000000   0.000000
857-4     59.5786  1.1150  -90.2153   0.0000 1.0000 857.0e9 714.2e9 999.9e9 0.03 2.0 1.15e-5 200.0 4.40e-3  5.0e-3 32  4.3449  1.0936  -58.0198 5.080e-6  0.000000   0.000000   0.000000
LFI-18S  -131.8147  3.2975   22.3000  -0.1    0.0000  70.0e9  63.0e9  77.0e9 0.05 1.7 1.15e-5  76.8 0.0e0   13.0e-3  8 13.0328  1.2601   86.6400 2.719e-4  0.002813  -0.000797  -0.002583
LFI-18M  -131.8147  3.2975   22.3000 -89.8    0.0000  70.0e9  63.0e9  77.0e9 0.05 1.7 1.15e-5  76.8 0.0e0   13.0e-3  8 12.9916  1.2184   86.0800 2.719e-4  0.002209  -0.000695  -0.002208
LFI-19S  -150.8570  3.1750   22.4000   0.0    0.0000  70.0e9  63.0e9  77.0e9 0.05 1.7 1.15e-5  76.8 0.0e0   13.0e-3  8 12.7053  1.2509   79.4200 2.719e-4  0.003770  -0.000964  -0.003233
LFI-19M  -150.8570  3.1750   22.4000 -90.0    0.0000  70.0e9  63.0e9  77.0e9 0.05 1.7 1.15e-5  76.8 0.0e0   13.0e-3  8 12.6494  1.2162   77.9400 2.719e-4  0.002988  -0.001035  -0.002814
LFI-20S  -168.4438  3.1649   22.4000   0.0    0.0000  70.0e9  63.0e9  77.0e9 0.05 1.7 1.15e-5  76.8 0.0e0   13.0e-3  8 12.4795  1.2460   73.0500 2.719e-4  0.003653  -0.001322  -0.003373
LFI-20M  -168.4438  3.1649   22.4000  89.9    0.0000  70.0e9  63.0e9  77.0e9 0.05 1.7 1.15e-5  76.8 0.0e0   13.0e-3  8 12.4273  1.2224   71.2800 2.719e-4  0.003344  -0.001270  -0.003222
LFI-21S   168.4438  3.1649  -22.4000   0.0    0.0000  70.0e9  63.0e9  77.0e9 0.05 1.7 1.15e-5  76.8 0.0e0   13.0e-3  8 12.4795  1.2460  106.9500 2.719e-4  0.003639   0.001319  -0.003361
LFI-21M   168.4438  3.1649  -22.4000 -89.9    0.0000  70.0e9  63.0e9  77.0e9 0.05 1.7 1.15e-5  76.8 0.0e0   13.0e-3  8 12.4273  1.2224  108.7200 2.719e-4  0.003324   0.001268  -0.003207
LFI-22S   150.8570  3.1747  -22.4000   0.0    0.0000  70.0e9  63.0e9  77.0e9 0.05 1.7 1.15e-5  76.8 0.0e0   13.0e-3  8 12.7053  1.2509  100.5800 2.719e-4  0.003770   0.000964  -0.003233
LFI-22M   150.8570  3.1747  -22.4000  90.0    0.0000  70.0e9  63.0e9  77.0e9 0.05 1.7 1.15e-5  76.8 0.0e0   13.0e-3  8 12.6494  1.2162  102.0600 2.719e-4  0.002988   0.001035  -0.002814
LFI-23S   131.8147  3.2975  -22.3000   0.1    0.0000  70.0e9  63.0e9  77.0e9 0.05 1.7 1.15e-5  76.8 0.0e0   13.0e-3  8 13.0328  1.2601   93.3600 2.719e-4  0.002813   0.000797  -0.002583
LFI-23M   131.8147  3.2975  -22.3000  89.8    0.0000  70.0e9  63.0e9  77.0e9 0.05 1.7 1.15e-5  76.8 0.0e0   13.0e-3  8 12.9916  1.2184   93.9200 2.719e-4  0.002209   0.000695  -0.002208
LFI-24S   180.0000  4.0536    0.0000   0.0    0.0000  44.0e9  39.6e9  48.4e9 0.05 1.7 1.15e-5  45.0 0.0e0   22.2e-3  8 22.4336  1.3114   90.0000 2.138e-4  0.000733  -0.000001  -0.000601
LFI-24M   180.0000  4.0536    0.0000  90.0    0.0000  44.0e9  39.6e9  48.4e9 0.05 1.7 1.15e-5  45.0 0.0e0   22.2e-3  8 22.5347  1.3704   90.0000 2.138e-4  0.000980  -0.000001  -0.000842
LFI-25S    61.1350  5.0186 -113.5000   0.5    0.0000  44.0e9  39.6e9  48.4e9 0.05 1.7 1.15e-5  45.0 0.0e0   22.2e-3  8 29.6208  1.2053  113.0400 2.138e-4 -0.000213   0.000387  -0.000352
LFI-25M    61.1350  5.0186 -113.5000  89.7    0.0000  44.0e9  39.6e9  48.4e9 0.05 1.7 1.15e-5  45.0 0.0e0   22.2e-3  8 28.7201  1.2508  109.7000 2.138e-4 -0.000238   0.000597  -0.000517
LFI-26S   -61.1350  5.0186  113.5000  -0.5    0.0000  44.0e9  39.6e9  48.4e9 0.05 1.7 1.15e-5  45.0 0.0e0   22.2e-3  8 29.6208  1.2053   66.9600 2.138e-4 -0.000213  -0.000387  -0.000352
LFI-26M   -61.1350  5.0186  113.5000 -89.7    0.0000  44.0e9  39.6e9  48.4e9 0.05 1.7 1.15e-5  45.0 0.0e0   22.2e-3  8 28.7201  1.2508   70.3000 2.138e-4 -0.000238  -0.000597  -0.000517
LFI-27S   153.6074  4.3466  -22.5000   0.2    0.0000  30.0e9  27.0e9  33.0e9 0.05 1.7 1.15e-5  32.5 0.0e0   30.8e-3  8 32.2352  1.3562  101.6800 1.802e-4  0.003400   0.001420  -0.002925
LFI-27M   153.6074  4.3466  -22.5000  89.9    0.0000  30.0e9  27.0e9  33.0e9 0.05 1.7 1.15e-5  32.5 0.0e0   30.8e-3  8 32.1377  1.3929  100.8900 1.802e-4  0.003445   0.001436  -0.003090
LFI-28S  -153.6074  4.3466   22.5000  -0.2    0.0000  30.0e9  27.0e9  33.0e9 0.05 1.7 1.15e-5  32.5 0.0e0   30.8e-3  8 32.2352  1.3562   78.3200 1.802e-4  0.003403  -0.001421  -0.002928
LFI-28M  -153.6074  4.3466   22.5000 -89.9    0.0000  30.0e9  27.0e9  33.0e9 0.05 1.7 1.15e-5  32.5 0.0e0   30.8e-3  8 32.1377  1.3929   79.1100 1.802e-4  0.003441  -0.001435  -0.003086
```