HEALPix Fortran Facility User Guidelines
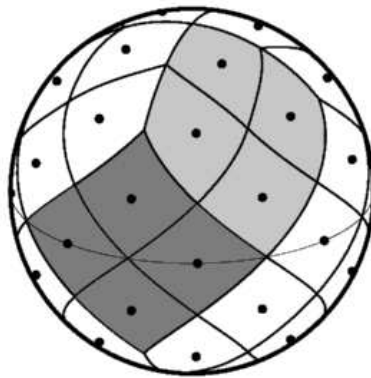
Abstract:	This document describes the **HEALPix** Fortran stand-alone facilities

TABLE OF CONTENTS

Using the HEALPix Fortran 90 facilities

Default File Names and Directories: for some applications, the **HEALPix** facilities require some precalculated input files describing the pixel window function and quadrature ring weight (shipped as `Healpix/data/pixel_window_n????.fits` and `Healpix/data/weight_ring_n?????.fits` respectively).

By default, files with the very same name will be looked for into: the current directory (.), the parent directory (..), ./data, ../data, $HEALPIX and $HEALPIX/data where $HEALPIX is a system variable defined as the full path to the **HEALPix** package (see the installation documentation). The user has the possibility to change both the name of those files and their location.

Double/Single Precision mode: several facilities offer the option of switching at run time the precision of the internal variables and arrays and of the I/O data from single to double precision floating point reals. The following points should be noted:

- Facilities running in double precision mode can read indifferently single and double precision data files (and the same is true for single precision facilities). On the other hand, a double (resp. single) precision facility will only output double (single) precision files.

- Since the internal calculations sensitive to numerical round-off error (like the spherical harmonics recurrence) are *always* performed in double precision, switching to double precision mode

    – will have a limited impact on the output accuracy if the input file contains only single precision data,

    – is recommended if the input file contains double precision data, and the precision of the output is critical

    – will not alter the execution speed, but it will almost double the memory consumption of the facility,

    – will obviously double the size of the output file(s).

Changes between release 1.2 and 2.0

- faster implementation of a_{lm} related facilities, generalization of OpenMP parallelization, and availability of MPI parallelized routines (see mpi_* routines in **Fortran90 Subroutines Overview** document).

- introduction of `alteralm` facility to modify and/or rotate the spherical harmonics coefficients a_{lm} and greater flexibility for constraining a_{lm} in `synfast`

- single and double precision implementation of most facilities (see **Input and Output Precision** page 3)

ALTERALM

Location in HEALPix directory tree: src/f90/alteralm/alteralm.f90

This program can be used to modify a set of a_{lm} spherical harmonics coefficients, as those extracted by anafast or simulated by synfast, before they are used as constraints on a synfast run. Currently the alterations possible are

- rotation (using Wigner matrices) of the a_{lm} from the input coordinate system to any other standard astrophysical coordinate system. The resulting a_{lm} can be used with e.g. synfast to generate a map in the new coordinate system.

- removal of the pixel and beam window functions of the input a_{lm} (corresponding to the pixel size and beam shape of the map from which they were extracted) and implementation of an arbitrary pixel and beam window function.

$$a_{\ell m}^{\mathrm{OUT}} = a_{\ell m}^{\mathrm{IN}} \frac{B^{\mathrm{OUT}}(\ell) P^{\mathrm{OUT}}(\ell)}{B^{\mathrm{IN}}(\ell) P^{\mathrm{IN}}(\ell)}, \quad (1)$$

where $P(\ell)$ is the pixel window function, and $B(\ell)$ is the beam window function (assuming a circular beam) or any other ℓ space filter (eg, Wiener filter). For an infinitely small pixel (or beam) one would have $P(\ell) = 1$ (resp. $B(\ell) = 1$) for any ℓ.

FORMAT   % alteralm [options] [parameter_file]

COMMAND LINE OPTIONS

-d
--double   double precision mode (see Notes on double/single precision modes on page 3)

-s
--single   single precision mode (default)

QUALIFIERS

infile_alms =	Defines the FITS file from which to read the input $a_{\ell m}$.
outfile_alms =	Defines the FITS file in which to write the altered $a_{\ell m}$.
fwhm_arcmin_in =	Defines the FWHM size in arcminutes of the Gaussian beam present in the input a_{lm}. The output a_{lm} will be corrected from it, see Eq. (1). (default= value of FWHM keyword in `infile_alms`).
beam_file_in =	Defines the FITS file describing the Legendre window function of the circular beam present in the input a_{lm}. The output a_{lm} will be corrected from it, see Eq. (1). If set to an existing file name, it will override the `fhwm_arcmin_in` given above. (default= value of the BEAM_LEG keyword in `infile_alms`)
nlmax_out =	Defines the maximum ℓ value to be used for the output a_{lm}s. (default= maximum ℓ of input $a_{\ell m}$ = value of MAX-LPOL keyword in `infile_alms`).
nsmax_in =	If it can not be determined from the input file `infile_alms`, asks for the **HEALPix** resolution parameter N_{side} whose window function is applied to the input $a_{\ell m}$
nsmax_out =	Defines the **HEALPix** resolution parameter N_{side} whose window function will be applied to the output $a_{\ell m}$. Could be set to 0 for infinitely small pixels, ie no pixel window function (default= same as input's N_{side}).
fwhm_arcmin_out =	Defines the FWHM size in arcminutes of the Gaussian beam to be applied to a_{lm}, see Eq. (1). (default= `fwhm_arcmin_in`).
beam_file_out =	Defines the FITS file describing the Legendre window function of the circular beam to be applied a_{lm}. If set to an existing file name, it will override the `fhwm_arcmin_out` given above. (default= " ")
coord_in =	Defines astrophysical coordinates used to compute the input $a_{\ell m}$. Valid choices are 'G' = Galactic, 'E' = Ecliptic, 'C'/'Q' = Celestial = eQuatorial. (default = value of COORDSYS keyword read from input FITS file)
epoch_in =	Defines astronomical epoch of input coordinate system (default=2000)

coord_out =	Defines astrophysical coordinates into which to rotate the $a_{\ell m}$ (default = `coord_in`)
epoch_out =	Defines astronomical epoch of output coordinate system (default=`epoch_in`)

DESCRIPTION

Alteralm can modify temperature as well as polarisation a_{lm}. It will also modify the error on the a_{lm} if those are provided. It works best if the input FITS file contains the relevant information on the beam size and shape, maximum multipoles, ...

DATASETS

The following datasets are involved in the **alteralm** processing.

Dataset	Description
/data/pixel_window_nxxxx.fits	Files containing pixel windows for various nsmax.

SUPPORT

This section lists those routines and facilities (including those *external* to the HEALPix distribution) which can assist in the utilisation of **alteralm**.

anafast	This **HEALPix** Fortran facility can analyse a **HEALPix** map to extract the a_{lm} that can be altered by alteralm.
synfast	This **HEALPix** facility can generate a **HEALPix** map from a power spectrum C_ℓ, with the possibility of including constraining a_{lm} as those obtained with alteralm.

EXAMPLES: #1

```
alteralm
```

Alteralm runs in interactive mode, self-explanatory.

EXAMPLES: #2

```
alteralm filename
```

When 'filename' is present, alteralm enters the non-interactive mode and parses its inputs from the file 'filename'. This has the following structure: the first entry is a qualifier which announces to the parser which input immediately follows. If this input is omitted in the input file, the parser assumes the default value. If the equality sign is omitted, then the parser ignores the entry. In this way comments may also be included in the file. In this example, the file contains the following qualifiers:

```
infile_alms = alm.fits
nlmax_out = 512
fwhm_arcmin_out = 20.0
coord_out = G
outfile_alms = newalm.fits
```

Alteralm reads the a_{lm} from 'alm.fits'. Since

```
nsmax_in
nsmax_out
fwhm_arcmin_in
beam_file_in
coord_in
epoch_in
epoch_out
```

have their default values, the pixel size will remain the same, the a_{lm} will be corrected from its input beam (whatever it was, assuming the relevant information can be found), and a gaussian beam of 20.0 arcmin will be applied instead, the a_{lm} will also be rotated from their original coordinate system (whatever it was, assuming the relevant information can be found) into Galactic coordinates, assuming a year 2000 epoch for both, and only the multipoles up to 512 will be written in 'newalm.fits'.

RELEASE NOTES

Revision 1: Initial release (**HEALPix** 2.00)

Messages

This section describes error messages generated by **alteralm**.

Message	Severity	Text
can not allocate memory for array xxx	Fatal	You do not have sufficient system resources to run this facility at the map resolution you required. Try a lower map resolution.
this is not a binary table		the fitsfile you have specified is not of the proper format
there are undefined values in the table!		the fitsfile you have specified is not of the proper format
the header in xxx is too long		the fitsfile you have specified is not of the proper format
XXX-keyword not found		the fitsfile you have specified is not of the proper format
found xxx in the file, expected:yyyy		the specified fitsfile does not contain the proper amount of data.
alteralm> no information found on input alms beam	Fatal	no information on the input beam was found, neither from parsing the FITS file header, nor from what the user provided.

ANAFAST

Location in HEALPix directory tree: src/f90/anafast/anafast.f90

This program performs harmonic analysis of the **HEALPix** maps up to a user specified maximum spherical harmonic order ℓ_{max}. The integrals are computed on the whole sphere, unless the user chooses a provided option to excise from the input map(s) a simple, constant latitude, symmetric cut, and/or apply an arbitray cut read from an external file. 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 $O(N_{\mathrm{pix}}^{3/2})$ with a prefactor depending on ℓ_{max}.

Anafast 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 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 is provided with an option to use precomputed Legendre Polynomials to decrease the total execution time.

Anafast 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.

RECOMMENDATIONS FOR USERS

Execution of anafast requires a user to specify the maximum spherical harmonic order ℓ_{max} up to which the harmonic decomposition of the input maps will be performed. Since there are no formal limits on parameter ℓ_{max} enforced by anafast, the user should make his/her choices judiciously. Hereafter it is convenient to specify ℓ_{max} in terms of the **HEALPix** map resolution parameter *nsmax*.

If the function to be analysed is strictly band-width limited, or nearly band-width limited (as in the case of a Gaussian beam smoothed signal discretized at a rate of a few pixels per beam area), it is sufficient to run anafast with $\ell_{max} \approx 2 \cdot nsmax$, with a very good C_ℓ error performance already in the raw (i.e. uncorrected quadrature) harmonic transform mode. If quadrature corrections are still desired in this case, it should be sufficient to use, at no extra cost in execution time, the ring-weighted quadrature scheme. This is the recommended mode of operation of anafast for essentially error and worry free typical applications, e.g. CPU-intensive Monte Carlo studies.

If more aggressive attempts are undertaken to extract from a map the spectral coefficients at $\ell > 2 \cdot nsmax$ (for example, as in a possible case of an attempt to analyse an existing map, which was irreversibly binned at a suboptimal resolution) the following should be kept in mind:

• Spherical harmonics discretized using **HEALPix** (either sampled at pixel centers, or avaraged over pixel areas) form a linearly independent system up to $\ell_{max} = 3 \cdot nsmax - 1$. Hence, the functions which are strictly band-width limited to $\ell_{max} = 3 \cdot nsmax - 1$ can be fully spectrally resolved with anafast, albeit with integration errors in the uncorrected quadrature mode, which grow up to $\delta C_\ell \propto \varepsilon \cdot C_\ell$, with $\varepsilon < 0.1$, at the highest values of ℓ. These integration errors can be efficiently reduced using anafast in the iterative mode. Although this ℓ_{max} range — $2 \cdot nsmax < \ell_{max} < 3 \cdot nsmax - 1$ — is easily manageable with anafast used on strictly band-width limited functions, it should be used with caution in basic and automated applications, e.g. Monte Carlo simulations.

• As with any discrete Fourier transform, anafast application to functions which are not band-width limited results with aliasing of power, which can not be remedied. If the particular case of interest may result in such a band-width violation (i.e. there is significant power in the function at $\ell > 3 \cdot nsmax - 1$), the function should be smoothed before the application of anafast, or discretized and then analysed, on a refined **HEALPix** grid (with larger *nsmax*).

• REMEMBER: A peculiar property of the sphere, which usually surprizes those whose intuition is built on experience with FFTs on a segment, or on a Euclidean multidimensional domains, is the lack of a regular and uniform point-set at arbitrary resolution, and the resulting non-commutativity of the forward and backward discrete Fourier transforms on nearly-uniform point-sets, e.g. **HEALPix** . Hence, as in any case of attempting an extreme application of an off-the-shelf software, use caution and understand your problem well *before* executing anafast under such cicumstances!

FORMAT   % anafast [options] [parameter_file]

COMMAND LINE OPTIONS

`-d`

`--double`   double precision mode (see Notes on double/single precision modes on page 3)

`-s`

`--single`   single precision mode (default)

QUALIFIERS

infile =   Defines the input map file. (default= map.fits)

outfile =   Defines the output file with the powerspectrum. (default= cl_out.fits)

simul_type =   Defines which map(s) to analyse, 1=temperature only, 2=temperature AND polarisation. (default= 1)

nlmax =   Defines the maximum ℓ value to be used. See the Recommendations for Users. (default= 64)

mask_file =   Defines the FITS file containing the pixel mask(s) or weighting scheme(s) by which the map(s) read from `infile` will be multiplied before analysis. If the file contains several fields, the first one in which at least one pixel is non-zero will be used. This option can be used to apply the WMAP Kp intensity mask to the data (see http://lambda.gsfc.nasa.gov), but it will *not* handle the WMAP composite mask correctly. Can be used in conjonction with `theta_cut_deg`. (default= ": no mask, all valid pixels are used)

theta_cut_deg =   Defines the latitude (in degrees) of an optional, straight symmetric cut around the equator. Pixels located within that cut ($|b| <$`theta_cut_deg` are ignored. (default= $0°$: no cut)

iter_order =   Defines the maximum order of quadrature iteration to be used. (default=0, no iteration)

outfile_alms =   Defines the name of the file containing the a_{lm} coefficients which can be written optionally. (default= no entry — a_{lm}s are not written to a file)

plmfile =	Defines the name for an input file containing precomputed Legendre polynomials P_{lm}. (default= no entry — anafast executes the recursive evaluation of P_{lm}s)
w8file =	Defines name for an input file containing ring weights in the improved quadrature mode (default= no entry — the name is assumed to be 'weight_ring_n0xxxx.fits' where xxxx is nsmax)
w8filedir =	Gives the directory where the ring weight files are to be found (default= no entry — anafast searches the default directories, see introduction)
won =	Set this to 1 if ring weight files are to be used, otherwise set it to 0 (or 2). (default= 0)
regression =	Sets the degree of the regression made on the input map before doing the power spectrum analysis. The regression is a minimal variance fit (assuming a uniform noise) made on valid (unflagged and unmasked) pixels, out of the symmetric cut (if any). In case of cut sky analysis, such a regression reduces the monopole and dipole leakage to higher ℓ's. 0 : no regression, does the a_lm analysis on the raw map 1 : removes the best fit monopole first 2 : removes the best fit monopole and dipole first default = 0.

DESCRIPTION   Anafast reads a binary FITS-file containing a **HEALPix** map. This file can contain a temperature map or both temperature and polarisation (Q,U) maps. Anafast analyses the map(s) and makes an output ascii-FITS file containing the C_l^Ts (and C_l^E, C_l^B, $C_l^{T \times E}$ if specified). Anafast produces C_ls up to a specified maximum ℓ-value (see Recommendations for Users). If requested, the computed a_{lm} coefficients can be written to a FITS file. This file can be used in the constrained realisation mode of synfast.

Anafast permits two execution modes that allow to improve the quadrature accuracy: (1) the ring weight corrected quadrature, and (2) the iterative scheme. Using the ring weights does not increase the execution time. The precomputed ring weights to be used for each **HEALPix** resolution *nsmax* are provided in the *Healpix/data* directory. The more sophisticated iterative scheme increases the accuracy more effectively than the weighted ring scheme, but its disadvantage is that the time for the analysis increases, 1 iteration takes 3 times as long, 2 iterations 5 times as long on so forth, since each order of iteration requires one more forward and backward transform.

The spherical harmonics evaluation uses a recurrence on associated Legendre polynomials $P_{lm}(\theta)$. This recurrence consumes most of the CPU time used by anafast. We have therefore included an option to load precomputed values for the $P_{lm}(\theta)$ from a file generated by the **HEALPix** facility plmgen. This represents a tradeoff between memory usage and execution speed of anafast.

DATASETS   The following datasets are involved in the **anafast** processing.

Dataset	Description
data/weight_ring_n0xxxx.fits	Files containing ring weights for the anafast improved quadrature mode.

SUPPORT   This section lists those routines and facilities (including those *external* to the HEALPix distribution) which can assist in the utilisation of **anafast**.

synfast	This **HEALPix** facility can generate a map for analysis by anafast.
alteralm	This **HEALPix** Fortran facility can be used to modify the a_{lm} extracted by anafast before they are used as constraints on a synfast run.
plmgen	This **HEALPix** facility can be used to generate pre-computed Legendre polynomials.

EXAMPLES: #1

```
anafast
```

Anafast runs in interactive mode — self-explanatory.

EXAMPLES: #2

```
anafast filename
```

When 'filename' is present, anafast enters the non-interactive mode and parses its inputs from the file 'filename'. This has the following structure: the first entry is a qualifier which announces to the parser which input immediately follows. If this input is omitted in the input file, the parser assumes the default value. If the equality sign is omitted, then the parser ignores the entry. In this way comments may also be included in the file. In this example, the file contains the following qualifiers:

```
simul_type = 1
nlmax = 64
theta_cut_deg = 0
iter_order = 0
infile = map.fits
outfile = cl_out.fits
regression = 0
```

Anafast reads the map from *map.fits*, makes an analysis and produces C_l^Ts up to l=64. This powerspectrum is saved in the file *cl_out.fits*. No galactic cut is excised and no iterations are performed. As regression is set to 0 (its default value) the map is analyzed as is, without prior best fit removal of the monopole nor the dipole.
Since `outfile_alms`
`w8file`
`w8filedir`
`plmfile`
were omitted, they take their default values (empty strings). This means that no file for precomputed Legendre polynomials is read and anafast does not save the a_{lm} values from the analysis.
Also since `won`
is not given, it takes it default value 2, which means that ring weights are not used.

RELEASE
NOTES

Revision 1: Initial release (**HEALPix** 0.90)

Revision 2: Optional non-interactive operation. Proper FITS file support. Improved reccurence algorithm for $P_{lm}(\theta)$ which can compute to higher ℓ values. New functionality: arbitrary order of iterations, precomputed P_{lm}, dumping of a_{lm}. (**HEALPix** 1.00)

Revision 3: New functionality : possibility of removing the best fit monopole and dipole. New Parser. Can be linked to FFTW (**HEALPix** 1.20)

Messages

This section describes error messages generated by **anafast**.

Message	Severity	Text
can not allocate memory for array xxx	Fatal	You do not have sufficient system resources to run this facility at the map resolution you required. Try a lower map resolution.

HOTSPOT

Location in HEALPix directory tree: src/f90/hotspot/HotSpots.f90

This Fortran facility provides a means to find local extrema of a map in **HEALPix** format. It also serves to illustrate the use of the following parts of the **HEALPix** toolkit: fast neighbour and extrema finding in the nested scheme, in-place conversion between RING and NESTED pixel schemes

FORMAT   % hotspot

QUALIFIERS

infile =	Defines the input map file.
extrema_outfile =	Defines the output map file.
maxima_outfile =	Defines the output ascii list of maxima.
minima_outfile=	Defines the output ascii list of minima.

DESCRIPTION   hotspot reads a healpix map in FITS format and generates the following outputs: 1) a **HEALPix** map in FITS format which is zero everywhere, except at pixels which contain local extrema. These pixels have the same values as in the input map. 2) an ASCII file which lists the pixel numbers and values of maxima, and 3) an ASCII file which lists the pixel numbers and values of minima.

The facility can be used in both an interactive mode and a command mode, where command qualifiers are fed to the facility using an input file.

Note the following limitations: hotspot (and the toolkit neighbour finder which it uses) will only work on maps with $N_{side} \geq 2$.

DATASETS   The following datasets are involved in the **hotspot** processing.

Dataset	Description
None required	

SUPPORT

This section lists those routines and facilities (including those *external* to the HEALPix distribution) which can assist in the utilisation of **hotspot**.

synfast	This **HEALPix** facility can generate a FITS format sky map to be input to hotspot.
map2gif	This **HEALPix** Fortran facility can be used to visualise the output map.
mollview	This **HEALPix** IDL facility can be used to visualise the output map.

EXAMPLES: #1

```
hotspot
```

hotspot runs in interactive mode.

EXAMPLES: #2

```
hotspot filename
```

When 'filename' is present, hotspot enters the non-interactive mode and parses its inputs from the file 'filename'. This has the following structure: the first entry is a qualifier which announces to the parser which input immediately follows. If this input is omitted in the input file, the parser assumes the default value shown below. If the equality sign is omitted, then the parser ignores the entry. In this way comments may also be included in the file. In this example, the file contains the following qualifiers:

```
infile = map.fits
extrema_outfile = pixlminmax.fits
maxima_outfile = maxima.dat
minima_outfile = minima.dat
```

hotspot reads in the map 'map.fits' and generates an output map with name 'pixlminmax.fits', and two ASCII files, 'maxima.dat' and 'minima.dat'.

RELEASE NOTES

Revision 1: Initial release (**HEALPix** 0.90)

Revision 2: Optional non-interactive operation. Proper FITS file support for input and output maps. (**HEALPix** 1.00)

Messages

This section describes error messages generated by **hotspot**.

Message	Severity	Text
can not allocate memory for array xxx	Fatal	You do not have sufficient system resources to run this facility at the map resolution you required. Try a lower map resolution.

MAP2GIF

Location in HEALPix directory tree: src/f90/map2gif/map2gif.f90

This Fortran facility provides a means to generate a gif 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.

FORMAT

% map2gif

QUALIFIERS

- bar	Logical which determines whether a color bar is displayed (**default:** .false.).
- col	The number of the color table to utilise (**default:** 5).
- hlp	Print on-line help for the facility.
- inp	The file name of the input FITS sky map (**default:** none).
- log	Logical to use the log of the signal when plotting (**default:** .false.)
- ash	Logical to use the hyperbolic arc sine of the signal when plotting. Cannot be true when -log is true. (**default:** .false.)
- add	Real value to add to the signal before performing any other operation to it (like taking the logarithm etc.) (**default:** 0.0)
- mul	Real value to multiply the signal with directly after adding the offset (see above). (**default:** 1.0)
- min	Set the minimum value for the plotted signal (**default:** is to use the actual signal minimum).
- max	Set the maximum value for the plotted signal (**default:** is to use the actual signal maximum).
- pro	Select the projection scheme (**default:** Mollweide).
- out	The file name of the output gif image (**default:** none).

	- sig	The identifier of the signal to plot: for a polarisation map, then the mapping is 1 = I; 2 = Q; 3 = U (**default:** 1).
	- ttl	A string specifying the title for the plot (**default:** none).
	- xsz	The x-dimension of the image in pixels (**default:** 800).

DESCRIPTION

map2gif reads in a **HEALPix** sky map in FITS format and generates an image in GIF format. map2gif allows the selection of the projection scheme (Mollweide or Gnomonic for small patches of the sky), color table, color bar inclusion, linear or log scaling, maximum and minimum range for the plot and plot-title. The facility utilises a command-line interface.

DATASETS

The following datasets are involved in the **map2gif** processing.

Dataset	Description
None required	

SUPPORT

This section lists those routines and facilities (including those *external* to the HEALPix distribution) which can assist in the utilisation of **map2gif**.

xv	xv or a similar facility is required to view the gif image generated by map2gif (a browser can also be used).
synfast	This **HEALPix** facility will generate the FITS format sky map to be input to map2gif.

EXAMPLES: #1

```
map2gif  -inp planck100GHZ-LFI.fits
         -out planck100GHZ-LFI.gif
         -bar .true.
         -min -100
         -max 100
         -ttl Simulated Planck LFI Sky Map at 100GHz
```

map2gif reads in the map ' planck100GHZ-LFI.fits' and generates an output gif image with name ' planck100GHZ-LFI.gif' in which the temperature scale has been set to lie between $\pm$ 100 (μK), a color bar has been drawn and the title ' Simulated Planck LFI Sky Map at 100GHz' appended to the image.

RELEASE NOTES

Revision 1: Initial release (**HEALPix** 1.00)

Messages

This section describes error messages generated by **map2gif**.

Message	Severity	Text
None at present		

MEDIAN_FILTER

Location in HEALPix directory tree: src/f90/median_filter/median_filter.f90

This program produces the median filtered map of an input **HEALPix** map (polarised or unpolarised). The neighborhood on the which the median is computed is defined as a disk of user-defined radius

FORMAT % median_filter [options] [parameter_file]

COMMAND LINE OPTIONS

```
        -d
```

`--double`   double precision mode (see Notes on double/single precision modes on page 3)

```
        -s
```

`--single`   single precision mode (default)

QUALIFIERS

simul_type =   Either 1 or 2. If set to 1, only the temperature component of the input map will be filtered. If set to 2, all the Stokes components available in the input file will be filtered (default = 1)

infile =   Name of the FITS file containing the map to be filtered (default = '', no default input file).

mf_radius_arcmin =   Radius in arcmin of the disk over which the median is computed (default = $3\theta_{pix}$ where θ_{pix} is the input map pixel size).

fill_holes =   If set to true, flagged pixels take for value the median of the valid pixels surrounding them (if any). Otherwise they are left unchanged. (default = .false.). Note that y, yes,t, true, .true. and 1 are interpreted as *true*, while n, no, f, false, .false. and 0 stand for *false*.

mffile =   Name of the FITS file containing the median filtered map

DESCRIPTION

Median_Filter produces a median filtered map in which the value of each pixel is the median of the input map valid pixels found within a disk of given radius centered on that pixel. A pixel flagged as 'non-valid' in the input map can either be left unchanged or 'filled in' with the same scheme, if at least one valid pixel is found among its neighbors.

If the map is polarized, each of the three Stokes components is filtered separately.

SUPPORT

This section lists those routines and facilities (including those *external* to the HEALPix distribution) which can assist in the utilisation of **median_filter**.

anafast   This **HEALPix** Fortran facility can analyse a **HEALPix** map.

synfast   This **HEALPix** facility can generate a **HEALPix** map from a power spectrum C_ℓ.

EXAMPLES: #1

    median_filter [option]

Median_Filter runs in interactive mode, self-explanatory.

EXAMPLES: #2

    median_filter [option] filename

When 'filename' is present, median filter enters the non-interactive mode and parses its inputs from the file 'filename'. This has the following structure: the first entry is a qualifier which announces to the parser which input immediately follows. If this input is omitted in the input file, the parser assumes the default value. If the equality sign is omitted, then the parser ignores the entry. In this way comments may also be included in the file. In this example, the file contains the following qualifiers:

```
simul_type = 1
infile = map.fits
mf_radius_arcmin = 20.0
mffile = med.fits
```

Median_Filter reads the sky map from 'map.fits'. Since
```
fill_holes
```
has its default value, ..., The median will be computed on a disk of 20 arcmin in radius, and the result will be written in 'med.fits'.

RELEASE NOTES

Revision 1: Initial release (**HEALPix** 2.00)

Messages

This section describes error messages generated by **median filter**.

Message	Severity	Text
can not allocate memory for array xxx	Fatal	You do not have sufficient system resources to run this facility at the map resolution you required. Try a lower map resolution.
this is not a binary table		the fitsfile you have specified is not of the proper format
there are undefined values in the table!		the fitsfile you have specified is not of the proper format
the header in xxx is too long		the fitsfile you have specified is not of the proper format
XXX-keyword not found		the fitsfile you have specified is not of the proper format
found xxx in the file, expected:yyyy		the specified fitsfile does not contain the proper amount of data.

PLMGEN

Location in HEALPix directory tree: src/f90/plmgen/plmgen.f90

This program can be used to create a file containing the precomputed values of the associated Legendre polynomials $P_{lm}(\theta)$ (and, if requested, of the tensor spherical harmonics) for faster execution of the **HEALPix** map analysis/synthesis. The map resolution parameter, *nsmax*, and the maximum value of the spherical harmonic order ℓ_{max} must be specified.

FORMAT   % plmgen

QUALIFIERS

nsmax =	Defines the resolution parameter for the map to be analysed/synthesized with the precomputed harmonics. (default= 32)
nlmax =	Defines the ℓ_{max} value for the execution. (default= 64)
simul_type =	Defines whether only scalar, or scalar and tensor harmonics are to be precomputed, 1=scalar only, 2=scalar AND tensor. (default=1)
outfile =	Defines the name for the file that will contain the precomputed harmonics. (default='plm.fits')

DESCRIPTION

The recursion of Legendre polynomials and tensor harmonics during the analysis and synthesis of **HEALPix** maps can be time consuming. Especially when repetitive applications are desired there is no need to compute the recursions every time. For such applications the values of $P_{\ell m}(\theta)$ can be precomputed with plmgen and stored in a file. When using synfast or anafast this file can be read in to shorten the analysis/synthesis execution time.

DATASETS

The following datasets are involved in the **plmgen** processing.

Dataset	Description
None required	

SUPPORT

This section lists those routines and facilities (including those *external* to the HEALPix distribution) which can assist in the utilisation of **plmgen**.

synfast	This **HEALPix** facility can generate a map using pre-computed harmonics made from plmgen.
anafast	This **HEALPix** facility can analyse a map using pre-computed harmonics.
plm_gen	Fortran subroutine used to generate the harmonics

EXAMPLES: #1

```
plmgen
```

plmgen runs in interactive mode, self-explanatory.

EXAMPLES: #2

```
plmgen filename
```

When 'filename' is present, plmgen enters the non-interactive mode and parses its inputs from the file 'filename'. This has the following structure: the first entry is a qualifier which announces to the parser which input immediately follows. If this input is omitted in the input file, the parser assumes the default value. If the equality sign is omitted, then the parser ignores the entry. In this way comments may also be included in the file. In this example, the file contains the following qualifiers:
```
simul_type = 1
nsmax = 32
nlmax = 86
outfile = 'plm.fits'
```

Creates a binary FITS file called 'plm.fits' containing Legendre polynomials up to ℓ and m values of 86 for a *nsmax* = 32 map. Legendre polynomials for all ℓ and m values for each angle θ corresponding to all of the **HEALPix** pixel center rings will be created.

RELEASE NOTES

Revision 1: Initial release **HEALPix** 1.00

Messages

This section describes error messages generated by **plmgen**.

Message	Severity	Text
can not allocate memory for array xxx	Fatal	You do not have sufficient system resources to run this facility at the map resolution you required. Try a lower map resolution.

SMOOTHING

Location in HEALPix directory tree: src/f90/smoothing/smoothing.f90

This program can be used to convolve a map with a gaussian beam. The input map can be given in RING or NESTED scheme and the smoothed map is written to a FITS file in the RING scheme.

NOTE: This automated facility is susceptible to problems with non-commutativity of discrete spherical harmonics transforms, described in the Recommendations for Users of the anafast facility. If very high accuracy of the results is required in the spectral regime of $\ell > 2 \cdot nsmax$, it is recommended to choose an iterative computation of the $a_{\ell m}$ coefficients.

FORMAT

% smoothing [options] [parameter_file]

COMMAND LINE OPTIONS

-d

--double   double precision mode (see Notes on double/single precision modes on page 3)

-s

--single   single precision mode (default)

QUALIFIERS

simul_type =   Defines which map(s) to analyse, 1=temperature only, 2=temperature AND polarisation. (default= 1)

infile =   Defines the filename for the FITS file containing the map to be smoothed. (default= 'map.fits')

nlmax =   Defines the ℓ_{max} value for the application. (default= 64)

iter_order =   Defines the maximum order of quadrature iteration to be used. (default=0, no iteration)

fwhm_arcmin =   Defines the FWHM in arcminutes of the gaussian beam for the convolution. (default=10)

beam_file =   Defines the FITS file describing the Legendre window function of the circular beam to be used for the

	simulation. If set to an existing file name, it will override the fhwm_arcmin given above. (default=" ")
outfile =	Defines the filename for the file that will contain the smoothed map. (default='map_smoothed.fits')
plmfile =	Defines the name for an input file containing precomputed Legendre polynomials P_{lm}. (default= no entry — anafast executes the recursive evaluation of P_{lm}s)
w8file =	Defines name for an input file containing ring weights in the improved quadrature mode (default= no entry — the name is assumed to be 'weight_ring_n0xxxx.fits' where xxxx is nsmax)
w8filedir =	Gives the directory where the weight files are to be found (default= no entry — smoothing searches in the default directories, see introduction)
won =	Set this to 1 if weight files are to be used, otherwise set it to 0 (or 2). (default= 0)

DESCRIPTION

A FITS file containing a **HEALPix** map in RING or NESTED scheme is read in. The map is analysed and smoothed in fourier space with a gaussian beam of a given FHWM. A new map is then synthesized using the smoothed $a_{\ell m}$ coeffecients. For a more accurate application, an iteration of arbitrary order can be applied. The output map is stored in *the same scheme* as the input map.

DATASETS

The following datasets are involved in the **smoothing** processing.

Dataset	Description
data/weight_ring_n0xxxx.fits	Files containing ring weights for the smoothing improved quadrature mode.

SUPPORT

This section lists those routines and facilities (including those *external* to the HEALPix distribution) which can assist in the utilisation of **smoothing**.

synfast	This **HEALPix** facility can generate a map and also do the smoothing.
anafast	This **HEALPix** facility can analyse a smoothed map.

EXAMPLES: #1

`smoothing`

Smoothing runs in interactive mode, self-explanatory.

EXAMPLES: #2

```
smoothing filename
```

When 'filename' is present, smoothing enters the non-interactive mode and parses its inputs from the file 'filename'. This has the following structure: the first entry is a qualifier which announces to the parser which input immediately follows. If this input is omitted in the input file, the parser assumes the default value. If the equality sign is omitted, then the parser ignores the entry. In this way comments may also be included in the file. In this example, the file contains the following qualifiers:

```
simul_type = 1
nlmax = 64
infile = 'map.fits'
outfile = 'map_smoothed.fits'
fwhm_arcmin = 10.
iter_order = 1
```

Smoothes the **HEALPix** temperature map contained in 'map.fits' with a 10 arcmin FWHM beam. The resulting map is saved in 'map_smoothed.fits'. The map analysis/synthesis was carried out using fourier coeffecients up to an ℓ value of 64. A first order iteration of the quadrature was performed.

RELEASE NOTES

Revision 1: Initial release (**HEALPix** 0.90)

Revision 2: Extension to polarization and arbitrary *circular* beams (**HEALPix** 1.20)

Messages

This section describes error messages generated by **smoothing**.

Message	Severity	Text
can not allocate memory for array xxx	Fatal	You do not have sufficient system resources to run this facility at the map resolution you required. Try a lower map resolution.

SYNFAST

Location in HEALPix directory tree: src/f90/synfast/synfast.f90

This program can be used to create **HEALPix** maps (temperature only or temperature and polarisation) computed as realisations of random Gaussian fields on a sphere characterized by the user provided theoretical power spectra, or as constrained realisations of such fields characterised by the user provided $a_{\ell m}$ coefficients and/or power spectra. Total operation count scales as $O(N_{\rm pix}^{3/2})$ with a prefactor dependent on the limiting spherical harmonics order ℓ_{max} of the actual problem. The map resolution, Gaussian beam FWHM, and random seed for the simulation can be selected by the user. Spherical harmonics are either generated using the recurrence relations during the execution of spectral synthesis, or precomputed and read in before the synthesis is executed to shorten the computation in repetitive applications.

FORMAT   % synfast [options] [parameter_file]

COMMAND LINE OPTIONS

`-d`

`--double`   double precision mode (see Notes on double/single precision modes on page 3)

`-s`

`--single`   single precision mode (default)

QUALIFIERS

infile =   Defines the input power spectrum file, (default= cl.fits). Note that `infile` is now optional : synfast can run even if only `almsfile` is provided.

outfile =   Defines the output map file, (default= map.fits). Note that `outfile` is now optional: if it set to " " (empty string), mo map is synthesized but the a_{lm} generated can be output.

outfile_alms =   Defines the FITS file in which to output a_{lm} used for the simulation (default= " ")

simul_type =	Defines the simulation type, 1=temperature only, 2=polarisation, 3=temperature and its first spatial derivatives, 4=temperature and its first and second spatial derivatives, 5=temperature and polarisation, and their first and second derivatives, 6=same as 5 plus the second derivatives of (T,Q,U). (default= 1).
nsmax =	Defines the resolution of the map. (default= 32)
nlmax =	Defines the maximum ℓ value to be used in the simulation. WARNING: ℓ_{max} can not exceed the value $4 \cdot nsmax$, because the coefficients of the average Fourier pixel window functions are precomputed and provided up to this limit. (default= 64)
iseed =	Defines the random seed to be used for the generation of a_{lm}s from the power spectrum. (default= -1)
fwhm_arcmin =	Defines the FWHM size in arcminutes of the simulated Gaussian beam. (default= 420.0)
beam_file =	Defines the FITS file describing the Legendre window function of the circular beam to be used for the simulation. If set to an existing file name, it will override the `fhwm_arcmin` given above. (default=" ")
almsfile =	Defines the input filename for a file containing a_{lm}s for constrained realisations. (default= " "). If `apply_windows` is *false* those a_{lm}s are used as they are, without being multiplied by the beam or pixel window function (with the assumption that they already have the correct window functions). If `apply_windows` is *true*, the beam and pixel window functions chosen above are applied to the constraining a_{lm} (with the assumption that those are free of beam and pixel window function). The code does not check the validity of these asumptions; if none is true, use the alteralm facility to modify or remove the window functions contained in the constraining a_{lm}.
apply_windows=	Determines how the constraining a_{lm} read from `almsfile` are treated with respect to window functions; see above for details. y, yes, t, true, .true. and 1 are considered as *true*, while n, no, f, false, .false. and 0 are considered as *false*, (default = .false.).
plmfile =	Defines the input filename for a file containing precomputed Legendre polynomials P_{lm}. (default= " ")
windowfile =	Defines the input filename for the pixel smoothing windows (default= "pixel_window_n????.fits", see Notes on default files and directories on page 3)
winfiledir =	Defines the directory in which windowfile is located (default : see Notes on default files and directories on page 3).

DESCRIPTION Synfast reads the power spectrum from a file in ascii FITS format. This can contain either just the temperature power spectrum C_l^T s or temperature and polarisation power spectra: C_l^T, C_l^E, C_l^B and $C_l^{T \times E}$. If `simul_type = 2` synfast generates Q and U maps as well as the temperature map. The output map(s) is (are) saved in a FITS file. The C_l s are used up to the specified ℓ_{lmax}, which can not exceed $4 \times$ nsmax. If `simul_type = 3` or `4` the first derivatives of the temperature field or the first and second derivatives respectively are output as well as the temperature itself: $T(p)$, $(\partial T / \partial \theta, \partial T / \partial \phi / \sin \theta)$, $\left(\partial^2 T / \partial \theta^2, \partial^2 T / \partial \theta \partial \phi / \sin \theta, \ \partial^2 T / \partial \phi^2 / \sin^2 \theta \right)$. If `simul_type = 5` or `6` the first derivatives of the (T,Q,U) fields or the first and second derivatives respectively are output as well as the field themself: $T(p)$, $Q(p)$, $U(p)$, $(\partial T / \partial \theta, \partial Q / \partial \theta, \partial U / \partial \theta; \partial T / \partial \phi / \sin \theta, \ldots)$, $\left(\partial^2 T / \partial \theta^2, \ldots; \partial^2 T / \partial \theta \partial \phi / \sin \theta, \ldots; \ \partial^2 T / \partial \phi^2 / \sin^2 \theta \ldots \right)$

The random seed for generation of a_{lm} from the power spectrum should be a negative integer. The map can be convolved with a gaussian beam for which a beamsize can be specified, or for an arbitrary *circular* beam for which the Legendre transform is provided. The map is automatically convolved with a pixel window function. These are stored in FITS files in the `healpix/data` directory. If synfast is not run in a directory which has these files, or from a directory which can reach these files by a '`../data/`' or '`./data/`' specification, the system variable `HEALPIX` is used to locate the main **HEALPix** directory and its `data` subdirectory is scanned. Failing this, the location of these files must be specified (using winfiledir). In the interactive mode this is requested only when necessary (see Notes on default directories on page 3).

If some of the a_{lm} in the simulations are constrained eg. from observations, a FITS file with these a_{lm} can be read. This FITS file contains the a_{lm} for certain ℓ and m values and also the standard deviation for these a_{lm}. The sky realisation which synfast produces will be statistically consistent with the constraining a_{lm}.

The code can also be used to generate a set of a_{lm} matching the input power spectra, beam size and pixel size with or without actually synthesizing the map. Those a_{lm} can be used as an input (constraining a_{lm}) to another synfast run.

Spherical harmonics values in the synthesis are obtained from a recurrence on associated Legendre polynomials $P_{lm}(\theta)$. This recurrence consumes most of the CPU time used by synfast. We have therefore included the option to read in the precomputed values of $P_{lm}(\theta)$ from a file generated by the **HEALPix** facility plmgen. This represents a tradeoff between memory usage and computing time.

...

Synfast will issue a warning if the input FITS file for the power spectrum does not contain the keyword POLNORM. This keyword indicates that the convention used for polarization is consistent with CMBFAST (and consistent with **HEALPix** 1.2). See the **HEALPix** Primer for details on the polarization convention and the interface with CMBFAST. If the keyword is not found, *no attempt will be made* to renormalize the power spectrum. If the keyword is present, it will be inherited by the simulated map.

DATASETS

The following datasets are involved in the **synfast** processing.

Dataset	Description
/data/pixel_window_nxxxx.fits	Files containing pixel windows for various nsmax.

SUPPORT

This section lists those routines and facilities (including those *external* to the HEALPix distribution) which can assist in the utilisation of **synfast**.

map2gif	This **HEALPix** Fortran facility can be used to visualise the output map.
mollview	This **HEALPix** IDL facility can be used to visualise the output map.
alteralm	This **HEALPix** Fortran facility can be used to implement the beam and pixel window functions on the constraining $a_l m$s (almsfile file).
anafast	This **HEALPix** Fortran facility can analyse a **HEALPix** map and save the a_{lm} and C_ls to be read by synfast.
plmgen	This **HEALPix** Fortran facility can be used to generate precomputed Legendre polynomials.

EXAMPLES: #1

    synfast

Synfast runs in interactive mode, self-explanatory.

EXAMPLES: #2

```
synfast filename
```

When 'filename' is present, synfast enters the non-interactive mode and parses its inputs from the file 'filename'. This has the following structure: the first entry is a qualifier which announces to the parser which input immediately follows. If this input is omitted in the input file, the parser assumes the default value. If the equality sign is omitted, then the parser ignores the entry. In this way comments may also be included in the file. In this example, the file contains the following qualifiers:

```
simul_type = 1
nsmax = 32
nlmax = 64
iseed = -1
fwhm_arcmin = 420.0
infile = cl.fits
outfile = map.fits
```

Synfast reads in the C_l power spectrum in $cl.fits$ up to $l = 64$, and produces the map $map.fits$ which has $N_{side} = 32$. The map is convolved with a beam of FWHM 420.0 arcminutes. The $iseed = -1$ sets the random seed for the realisation. A different $iseed$ would have given a different realisation from the same power spectrum. Since

```
outfile_alms
almsfile
apply_windows
plmfile
beam_file
windowfile
```

were omitted, they take their default values (empty strings). This means that no file for constrained realisation or precomputed Legendre polynomials are read, the a_{lm} generated in the process are not output, and synfast attempts to find the pixel window files in the default directories (see page 3).

RELEASE NOTES

Revision 1: Initial release (**HEALPix** 0.90)

Revision 2: Optional non-interactive operation. Proper FITS file support. Improved reccurence algorithm for $P_{lm}(\theta)$ which can compute to higher ℓ values. Improved pixel windows averaged over actual HEALPix pixels. New functionality: constrained realisations, precomputed P_{lm}. (**HEALPix** 1.00)

Revision 3: New functionality: constrained realisations and pixel windows are now available for polarization as well. Arbitrary circular beams can be used. New parser (**HEALPix** 1.20)

Revision 4: New functionnality: the generated a_{lm} can be output, and the map synthesis itself can be skipped. First and second derivatives of the temperature field can be produced on demand.

Messages

This section describes error messages generated by **synfast**.

Message	Severity	Text
can not allocate memory for array xxx	Fatal	You do not have sufficient system resources to run this facility at the map resolution you required. Try a lower map resolution.
this is not a binary table		the fitsfile you have specified is not of the proper format
there are undefined values in the table!		the fitsfile you have specified is not of the proper format
the header in xxx is too long		the fitsfile you have specified is not of the proper format
XXX-keyword not found		the fitsfile you have specified is not of the proper format
found xxx in the file, expected:yyyy		the specified fitsfile does not contain the proper amount of data.

UD_GRADE

Location in HEALPix directory tree: src/f90/ud_grade/ud_grade.f90

This program can upgrade or degrade the resolution of a **HEALPix** map.

FORMAT   % ud_grade [options] [parameter_file]

COMMAND LINE OPTIONS

-d

--double   double precision mode (see Notes on double/single precision modes on page 3)

-s

--single   single precision mode (default)

QUALIFIERS

nside_out =	Defines the resolution parameter for the output map. (default= 64)
infile =	Defines the name of the file containing the map to be up/degraded. (default='map.fits')
outfile =	Defines the filename for the output up/degraded map. (default='outmap.fits')

DESCRIPTION This facility transforms the resolution of an input **HEALPix** map. At each step of map resolution upgrade the four output map pixels nested in one pixel of the input map are given the values of the input pixel. At each step of map resolution degradation the four input map pixels nested in one output map pixel are averaged to produce the pixel value in the output map. **Caution** Beware that, at this stage, the parallel tranport of the polarization (Q and U Stokes vectors) that would be necessary to describe the change in local coordinates is **not** implemented.

DATASETS

The following datasets are involved in the **ud_grade** processing.

Dataset	Description
None required	

SUPPORT

This section lists those routines and facilities (including those *external* to the HEALPix distribution) which can assist in the utilisation of **ud_grade**.

mollview	IDL routine to view an up/downgraded map.
anafast	This **HEALPix** facility can analyse an up/downgraded map.

EXAMPLES: #1

```
ud_grade
```

ud_grade runs in interactive mode, self-explanatory.

EXAMPLES: #2

```
ud_grade filename
```

When 'filename' is present, ud_grade enters the non-interactive mode and parses its inputs from the file 'filename'. This has the following structure: the first entry is a qualifier which announces to the parser which input immediately follows. If this input is omitted in the input file, the parser assumes the default value. If the equality sign is omitted, then the parser ignores the entry. In this way comments may also be included in the file. In this example, the file contains the following qualifiers:

```
nside_out = 64
infile= 'map.fits'
outfile = 'outmap.fits'
```

Ud_grade transforms the **HEALPix** map in 'map.fits' to resolution *Nside* = 64, no matter what the input map resolution was. The up/downgraded map is stored in 'outmap.fits'.

RELEASE NOTES

Revision 1: (Initial release **HEALPix** 0.90)

Revision 2: Extension to multi-dimensional maps (**HEALPix** 1.20)

Messages

This section describes error messages generated by **ud_grade**.

Message	Severity	Text
can not allocate memory for array xxx	Fatal	You do not have sufficient system resources to run this facility at the map resolution you required. Try a lower map resolution.