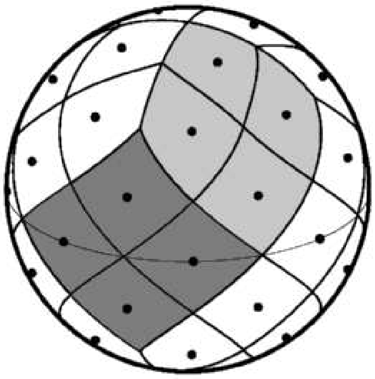
HEALPix C Subroutines Overview



Revision: Version 3.30; October 8, 2015

Prepared by: Eric Hivon, Anthony J. Banday, Matthias Bartelmann, Frode K. Hansen, Krzysztof M. Górski, Martin Reinecke and Benjamin D. Wandelt

Abstract: This document is an overview of the **HEALPix** C subroutines.

http://healpix.sf.net

Contents

Conventions

Here we list some conventions which are used in this document.

$\mathbf{N_{side}}$ **HEALPix** resolution parameter — see the **HEALPix** Primer.

θ The polar angle or colatitude on the sphere, ranging from 0 at the North Pole to π at the South Pole.

ϕ The azimuthal angle on the sphere, $\phi \in [0, 2\pi[$.

Compilation and Installation

A tentative compilation and installation script is provided in `src/C/doinstall`. If it does not work, you can try editing the `src/C/subs/Makefile` by hand.

Usage

To use in your 'C' code, include the line

```
#include "chealpix.h"
```

in your code and link with something like

```
gcc -o myprog myprog.c -I<incdir> -L<libdir> -lchealpix
```

where `<incdir>` is where you've installed the '.h' files and `<libdir>` is where you've installed the libraries (See the header of the 'subs/Makefile').

You will also need the 'cfitsio' library. See
http://heasarc.gsfc.nasa.gov/docs/software/fitsio/

Note on the C routines

This small set of C routines is provided as a start up kit to users wanting to link the **HEALPix** routines with some other languages (C, C++, IDL, perl, ...), and it was actually mainly provided by various users (see individual routines for details). As for the rest of the **HEALPix** package, all interested persons are welcome to contribute to this effort.

ang2vec

Location in HEALPix directory tree: src/C/subs/chealpix.c

Routine to convert the position angles (θ, ϕ) of a point on the sphere into its 3D position vector (x, y, z) with $x = \sin\theta\cos\phi$, $y = \sin\theta\sin\phi$, $z = \cos\theta$.

FORMAT

void vec2ang(double theta, double phi, double *vector);

ARGUMENTS

name & dimensionality	kind	in/out	description
theta	double	IN	colatitude in radians measured southward from north pole (in [0,π]).
phi	double	IN	longitude in radians measured eastward (in [0, 2π]).
vector(3)	double	OUT	three dimensional cartesian position vector (x, y, z). The north pole is $(0, 0, 1)$

RELATED ROUTINES

This section lists the routines related to **ang2vec**.

vec2ang	converts the 3D position vector of point into its position angles on the sphere.

get_fits_size

Location in HEALPix directory tree: src/C/subs/chealpix.c

This routine reads the number of pixels, the resolution parameter and the pixel ordering of a FITS file containing a **HEALPix** map.

FORMAT

long get_fits_size(char *filename, long *nside, char *ordering)

ARGUMENTS

name&dimensionality	kind	in/out	description
get_fits_size	long	OUT	number of pixels the FITS file
filename	char	IN	filename of the FITS-file containing the **HEALPix** map.
ordering	char	OUT	pixel ordering, either 'RING' or 'NESTED'
nside	long	OUT	Healpix resolution parameter Nside

EXAMPLE:

```
long npix, nside ;
char file[180]=''map.fits'' ;
char order[10] ;
npix= get_fits_size(file, &nside, order)
```

Returns in npix the number of pixel in the file 'map.fits', and read in nside or order its resolution parameter or ordering scheme

RELATED ROUTINES

This section lists the routines related to **get_fits_size**.

read_healpix_map	subroutine to read **HEALPix** maps
write_healpix_map	subroutine to write **HEALPix** maps

npix2nside

Location in HEALPix directory tree: src/C/subs/chealpix.c

Function to provide the resolution parameter $N_{\rm side}$ corresponding to the number of pixels $N_{\rm pix}$ over the full sky.

FORMAT

long npix2nside(const long npix)

ARGUMENTS

name&dimensionality	kind	in/out	description
npix	long	IN	the number of pixels $N_{\rm pix}$ of the map .
npix2nside	long	OUT	returns the $N_{\rm side}$ parameter of the map such that $N_{\rm pix} = 12 {N_{\rm side}}^2$.

EXAMPLE:

```
nside= npix2nside(786432);
```

Returns the resolution parameter (256) corresponding to 786432 **HEALPix** pixels.

RELATED ROUTINES

This section lists the routines related to **npix2nside**.

ang2vec	converts (θ, ϕ) spherical coordinates into (x, y, z) cartesian coordinates.
vec2ang	converts (x, y, z) cartesian coordinates into (θ, ϕ) spherical coordinates.
nside2npix	converts number of full sky pixels $N_{\rm pix}$ into resolution parameter $N_{\rm side}$

nside2npix

Location in HEALPix directory tree: src/C/subs/chealpix.c

Function to provide the number of pixels $N_{\rm pix}$ over the full sky corresponding to resolution parameter $N_{\rm side}$.

FORMAT

long nside2npix(const long nside)

ARGUMENTS

name&dimensionality	kind	in/out	description
nside	long	IN	the $N_{\rm side}$ parameter of the map.
nside2npix	long	OUT	returns the number of pixels $N_{\rm pix}$ of the map $N_{\rm pix} = 12{N_{\rm side}}^2$.

EXAMPLE:

```
npix= nside2npix(256);
```

Returns the number of **HEALPix** pixels (786432) for the resolution parameter 256.

RELATED ROUTINES

This section lists the routines related to **nside2npix**.

ang2vec	converts (θ, ϕ) spherical coordinates into (x, y, z) cartesian coordinates.
vec2ang	converts (x, y, z) cartesian coordinates into (θ, ϕ) spherical coordinates.
npix2nside	converts $N_{\rm side}$ into number of full sky pixels $N_{\rm pix}$.

pix2xxx, ang2xxx, vec2xxx, nest2ring, ring2nest

Location in HEALPix directory tree: src/C/subs/chealpix.c

These subroutines can be used to convert between pixel number in the **HEALPix** map and (θ, ϕ) coordinates on the sphere. This is only a subset of the routines equivalent in Fortran90 or in IDL.

ARGUMENTS

name & dimensionality	type	in/out	description
nside	long	IN	N_{side} parameter for the **HEALPix** map.
ipnest	long	—	pixel identification number in NESTED scheme over the range $\{0,N_{pix}-1\}$.
ipring	long	—	pixel identification number in RING scheme over the range $\{0,N_{pix}-1\}$.
theta	double	—	colatitude in radians measured southward from north pole in $[0,\pi]$.
phi	double	—	longitude in radians, measured eastward in $[0,2\pi]$.
vector	double	—	3D cartesian position vector (x, y, z). The north pole is $(0, 0, 1)$. An output vector is normalised to unity.

ROUTINES:

`void pix2ang_ring(long nside, long ipring, double *theta, double *phi);`

renders *theta* and *phi* coordinates of the nominal pixel center given the pixel number *ipring* and a map resolution parameter *nside*.

`void pix2vec_ring(long nside, long ipring, double *vector);`

renders cartesian vector coordinates of the nominal pixel center given the pixel number *ipring* and a map resolution parameter *nside*. Optionally renders cartesian vector coordinates of the considered pixel four vertices.

`void ang2pix_ring(long nside, double theta, double phi, long *ipring);`

renders the pixel number *ipring* for a pixel which, given the map resolution parameter *nside*, contains the point on the sphere at angular coordinates *theta* and *phi*.

`void vec2pix_ring(long nside, double *vector, long *ipring);`

renders the pixel number *ipring* for a pixel which, given the map resolution parameter *nside*, contains the point on the sphere at cartesian coordinates *vector*.

`void pix2ang_nest(long nside, long ipnest, double *theta, double *phi);`

renders *theta* and *phi* coordinates of the nominal pixel center given the pixel number *ipnest* and a map resolution parameter *nside*.

`void pix2vec_nest(long nside, long ipnest, double *vector);`

renders cartesian vector coordinates of the nominal pixel center given the pixel number *ipnest* and a map resolution parameter *nside*. Optionally renders cartesian vector coordinates of the considered pixel four vertices.

`void ang2pix_nest(long nside, double theta, double phi, long *ipnest);`

renders the pixel number *ipnest* for a pixel which, given the map resolution parameter *nside*, contains the point on the sphere at angular coordinates *theta* and *phi*.

`void vec2pix_nest(long nside, double *vector, long *ipnest)`

renders the pixel number *ipnest* for a pixel which, given the map resolution parameter *nside*, contains the point on the sphere at cartesian coordinates *vector* .

`void nest2ring(long nside, long ipnest, long *ipring);`

performs conversion from NESTED to RING pixel number.

```
void ring2nest(long nside, long ipring, long *ipnest);
```

performs conversion from RING to NESTED pixel number.

MODULES & ROUTINES

This section lists the modules and routines used by **pix2xxx, ang2xxx, vec2xxx, nest2ring, ring2nest**.

mk_pix2xy, mk_xy2pix	routines used in the conversion between pixel values and "cartesian" coordinates on the Healpix face.

RELATED ROUTINES

This section lists the routines related to **pix2xxx, ang2xxx, vec2xxx, nest2ring, ring2nest**.

ang2vec	converts (θ, ϕ) spherical coordinates into (x, y, z) cartesian coordinates.
vec2ang	converts (x, y, z) cartesian coordinates into (θ, ϕ) spherical coordinates.
nside2npix	converts number of full sky pixels $N_{\rm pix}$ into resolution parameter $N_{\rm side}$
npix2nside	converts $N_{\rm side}$ into number of full sky pixels $N_{\rm pix}$.

read_healpix_map

Location in HEALPix directory tree: src/C/subs/chealpix.c

This routine reads a full sky **HEALPix** map from a FITS file

FORMAT

float *read_healpix_map(char *infile, long *nside, char *coordsys, char *ordering)

ARGUMENTS

name&dimensionality	kind	in/out	description
read_healpix_map	float	OUT	array containing the map read from the file
infile	char	IN	FITS file containing a full sky to be read
nside	long	OUT	**HEALPix** resolution parameter of the map
coordsys	char	OUT	astronomical coordinate system of pixelation (either ’C’, ’E’ or ’G’ standing respectively for Celestial=equatorial, Ecliptic or Galactic)
ordering	char	OUT	**HEALPix** pixel ordering (either ’RING’ or ’NESTED’)

RELATED ROUTINES

This section lists the routines related to **read_healpix_map**.

anafast	executable that reads a **HEALPix** map and analyses it.
synfast	executable that generate full sky **HEALPix** maps
write_healpix_map	subroutine to write **HEALPix** maps
get_fits_size	subroutine to determine the size of a map

vec2ang

Location in HEALPix directory tree: src/C/subs/chealpix.c

Routine to convert the 3D position vector (x, y, z) of point into its position angles (θ, ϕ) on the sphere with $x = \sin\theta\cos\phi$, $y = \sin\theta\sin\phi$, $z = \cos\theta$.

FORMAT

void vec2ang(double *vector, double *theta, double *phi);

ARGUMENTS

name & dimensionality	kind	in/out	description
vector(3)	double	IN	three dimensional cartesian position vector (x, y, z). The north pole is $(0, 0, 1)$
theta	double	OUT	colatitude in radians measured southward from north pole (in [0,π]).
phi	double	OUT	longitude in radians measured eastward (in [0, 2π]).

RELATED ROUTINES

This section lists the routines related to **vec2ang**.

ang2vec	converts the position angles of a point on the sphere into its 3D position vector.

write_healpix_map

Location in HEALPix directory tree: src/C/subs/chealpix.c

This routine writes a full sky **HEALPix** map into a FITS file

FORMAT

int write_healpix_map(float *signal, long nside, char *filename, char nest, char *coordsys)

ARGUMENTS

name&dimensionality	kind	in/out	description
write_healpix_map	int	OUT	returns a non zero value in case of error
signal	float	IN	full sky map to be written
nside	long	IN	**HEALPix** resolution parameter of the map (the map should have 12 * nside * nside pixels).
filename	char	IN	FITS file in which to write the full sky map
nest	char	IN	flag specifing the **HEALPix** pixel ordering of the map. 0: 'RING' and 1: 'NESTED'
coordsys	char	IN	astronomical coordinate system of map (must be either 'C', 'E' or 'G' standing respectively for Celestial=equatorial, Ecliptic or Galactic)

RELATED ROUTINES

This section lists the routines related to **write_healpix_map**.

anafast	executable that reads a **HEALPix** map and analyses it.
synfast	executable that generate full sky **HEALPix** maps
read_healpix_map	subroutine to read **HEALPix** maps
get_fits_size	subroutine to determine the size of a map