

1. This is the manual of the CHEALPix library, a small C library which implements the HEALPix tessellation of the sphere. The purpose of the HEALPix tessellation is to divide a sphere into patches of equal area which have desirable properties for a number of calculations mostly used in astrophysics and cosmology, e.g.:

- (1) Computation of the spherical Fourier transforms for a random field on the sphere;
- (2) Efficient nearest-neighbour searches of pixels.

2. The standard implementation of the HEALPix tessellation scheme is the HEALPix library developed by Gorski et al. It provides a number of functions in Fortran, C, C++, IDL and Java and is so far the most complete and widely used implementation. Still, it suffers of a number of drawbacks:

- (1) There are no connections among the language bindings. (In fact, instead of speaking of “bindings” — which imply that there is one library and many interfaces that binds to it — one would rather speak of four different libraries: one for each language.) This implies that if one wants to implement some alternative algorithm for a HEALPix routine, she has to code the algorithm once for each language supported.
- (2) The separation of the language bindings has had the effect of reduce the momentum in developing some of them. The C bindings appear to be the most neglected, for there is no facility for reading/saving/doing calculations on $a_{\ell m}$ coefficients. This is particularly limiting for those developers wanting to create bindings to other languages, as C is the *lingua franca* used for Foreign Function Interfaces (FFIs).
- (3) The C/C++ bindings of the library use Doxygen to produce the documentation. Although Doxygen is a powerful tool, the HEALPix function are usually poorly documented: there is no example of use for any of the C++ functions, and the description of each function is usually only a one-liner.
- (4) The installation of HEALPix is not straightforward: the library requires the user to install CFITSIO first, which is a rather large library if compared with what CFITSIO uses (the largest part of the CFITSIO code implements functions for reading/writing images, while HEALPix only reads and write binary tables). Moreover, there is no standard facility for a program using HEALPix to find and link the library.

3. This document presents a new C library which implements the HEALPix tessellation scheme, with the following advantages over the standard HEALPix library:

- (1) Only the C language is supported. This reduces the size of the library and eases the development.
- (2) The library tries to support all the facilities provided by the standard HEALPix library, even those implemented in languages other than C.
- (3) The library is written using CWEB, a literate programming tool developed by Donald E. Knuth and Silvio Levy which provides a powerful way to document the code.
- (4) The library supports `pkg-config`.

4. This is the main header of the library.

```
<chealpix.h 4> ≡
#ifndef CHEALPIX_H
#define CHEALPIX_H
#ifdef __cplusplus
    extern "C"
    {
#endif __cplusplus
#include <chealpix/version.h>
    typedef unsigned long pixel_num_t;
    pixel_num_t chealpix_nside_to_npixel(const unsigned short);
#ifdef __cplusplus
    }
;
#endif __cplusplus
#endif
```

5. And finally we have one of the functions of the library.

```
#include <chealpix/chealpix.h>
pixel_num_t chealpix_nside_to_npixel(const unsigned short nside)
{
    return 12 * nside * nside;
}
```

6. We put together these definitions into a header file.

```
<version.h 6> ≡
#define CHEALPIX_VERSION "0.1"

__cplusplus: 4.
CHEALPIX_H: 4.
chealpix_nside_to_npixel: 4, 5.
CHEALPIX_VERSION: 6.
CWEB: 3.
nside: 5.
pixel_num_t: 4, 5.
```

`<chealpix.h 4>`
`<version.h 6>`