Etat d'avancement – Septembre 2007

Utilisation du package Healpix fournis sur le site :

http://healpix.jpl.nasa.gov/healpixSoftwareCapabilities.shtml

Les méthodes java sont encore en cours d'écriture, nous avons pu demander aux auteurs une version temporaire plus avancées.

J'ai compléter le code source de ces librairies pour pouvoir commencer les tests de lecture et affichage des fichiers au format Healpix.

1 - Lecture d'un fichier au format « Healpix »

Il existe plusieurs formats de fichiers Healpix, soit table, soit image. Plusieurs exemples sont donnés dans le package initial.

Exemple entête du fichier pixel_window_n1024.fits :

```
XTENSION= 'BINTABLE'           / binary table extension
BITPIX  =                    8 / 8-bit bytes
NAXIS   =                    2 / 2-dimensional binary table
NAXIS1  =                   16 / width of table in bytes
NAXIS2  =                 4097 / number of rows in table
PCOUNT  =                    0 / size of special data area
GCOUNT  =                    1 / one data group (required keyword)
TFIELDS =                    2 / number of fields in each row
COMMENT
EXTNAME = 'PIXEL WINDOW'       /
NSIDE   =                 1024 / Resolution parameter for HEALPIX
MAX-LPOL=                 4096 / Maximum multipole l=4*nside
COMMENT
COMMENT
COMMENT
COMMENT  Contains pixel window smoothing factors
COMMENT  for temperature and polarization for NSIDE =  1024
COMMENT  for multipoles in range [0, 4096]
COMMENT
TTYPE1  = 'TEMPERATURE'        / TEMPERATURE pixel window smoothing factors
TFORM1  = '1D      '           / data format of field: 8-byte REAL
TUNIT1  = 'unknown '           / no unit
COMMENT
TTYPE2  = 'POLARIZATION'       / POLARIZATION pixel window smoothing factors
TFORM2  = '1D      '           / data format of field: 8-byte REAL
TUNIT2  = 'unknown '           / no unit
HISTORY
HISTORY  For NSIDE <= 128, the T and P windows are computed exactly
HISTORY  by averaging the individual pixel windows.
HISTORY  For NSIDE > 128 :
HISTORY  * The average T window is extrapolated from smaller NSIDE
HISTORY  assuming a scaling of the window function with the pixel radius
HISTORY  similar to the one exhibited by a tophat window
HISTORY  * The P window is assumed proportional to T
HISTORY      E. Hivon, Nov 2002
END
```

Exemple d'entête du fichier map_sm.fits :

```
XTENSION= 'BINTABLE'           / binary table extension
BITPIX  =                    8 / 8-bit bytes
NAXIS   =                    2 / 2-dimensional binary table
NAXIS1  =                12288 / width of table in bytes
NAXIS2  =                  192 / number of rows in table
PCOUNT  =                    0 / size of special data area
GCOUNT  =                    1 / one data group (required keyword)
```

```
TFIELDS =                    3 / number of fields in each row
COMMENT ------------------------------------------------
COMMENT Sky Map Pixelisation Specific Keywords
COMMENT ------------------------------------------------
PIXTYPE = 'HEALPIX '           / HEALPIX Pixelisation
ORDERING= 'RING    '           / Pixel ordering scheme, either RING or NESTED
NSIDE   =                  128 / Resolution parameter for HEALPIX
FIRSTPIX=                    0 / First pixel # (0 based)
LASTPIX =               196607 / Last pixel # (0 based)
COORDSYS= 'unknown '           / Pixelization coordinate system
COMMENT G = Galactic, E = ecliptic, C = celestial = equatorial
COMMENT
COMMENT ------------------------------------------------
COMMENT Planck Simulation Specific Keywords
COMMENT ------------------------------------------------
EXTNAME = 'SMOOTHED DATA'
CREATOR = 'SMOOTHING'          / Software creating the FITS file
VERSION = '2.0.0   '           / Version of the simulation software
MAX-LPOL=                  256 / Maximum multipole l used in map smoothing
FWHM    =       7.00000000E+00 / [deg] FWHM of gaussian symmetric beam
COMMENT
COMMENT ************************ Input data ************************
COMMENT Input Map in map.fits
COMMENT ************************************************************
COMMENT
COMMENT ------------------------------------------------
COMMENT Data Description Specific Keywords
COMMENT ------------------------------------------------
INDXSCHM= 'IMPLICIT'           / Indexing : IMPLICIT or EXPLICIT
GRAIN   =                    0 / Grain of pixel indexing
COMMENT GRAIN=0 : no indexing of pixel data                        (IMPLICIT)
COMMENT GRAIN=1 : 1 pixel index -> 1 pixel data                    (EXPLICIT)
COMMENT GRAIN>1 : 1 pixel index -> data of GRAIN consecutive pixels (EXPLICIT)
COMMENT
POLAR   =                    T / Polarisation included (True/False)
TTYPE1  = 'TEMPERATURE'        / Temperature map
TFORM1  = '1024E   '           / data format of field: 4-byte REAL
TUNIT1  = 'uK      '
COMMENT
TTYPE2  = 'Q-POLARISATION'     / Q Polarisation map
TFORM2  = '1024E   '           / data format of field: 4-byte REAL
TUNIT2  = 'uK      '
COMMENT
TTYPE3  = 'U-POLARISATION'     / U Polarisation map
TFORM3  = '1024E   '           / data format of field: 4-byte REAL
TUNIT3  = 'uK      '
COMMENT
END
```

L'auteur a pu me fournir une explication concernant ces fichiers :

pixel_window_n????.fits ne sont pas des cartes Healpix habituelles, mais contiennent des informations nécessaires à certains calculs (=transformées en harmoniques sphériques), et ne devraient donc pas être nécessaires pour la manipulation ou visualisation d'image.

*Ils décrivent respectivement les fonctions fenêtres des pixels dans l'espace des polynômes (pour l=[0, 4*nside]) et des poids à appliquer sur chaque 'ring' Healpix (au nombre de 2*Nside dans chaque hémisphère).*

Par contre, le directory test contient des fichiers map.fits qui sont des cartes Healpix standard, qui elles ont npix = 12*nside*nside.*

Dans celles ci, les mots clés importants sont effectivement (cf test/map_sm.fits)

PIXTYPE
NSIDE
ORDERING
COORDSYS
INDXSCHM
GRAIN
ou les 2 derniers indiquent si les numéros de pixels sont implicites (dans le cas où le fichier contient tout le ciel) ou bien les numéros des pixels présents sont donnés

explicitement (dans les cas où seulement une petite fraction du ciel se trouve dans le fichier).

Explications du format des données dans d'un fichier :

Une fois l'entête lue, nous avons une table (ou cube) de données binaires à lire.

Nombre de champs = TFIELDS

Nombre de lignes = NAXIS2

Nombre de colonnes pour une « cellule » (une ligne x le champ *N*) = TFORM*N*

Nos avons donc pour un champ donné : TFORM1*NAXIS2 valeurs (ou aussi 12*NSIDE*NSIDE).

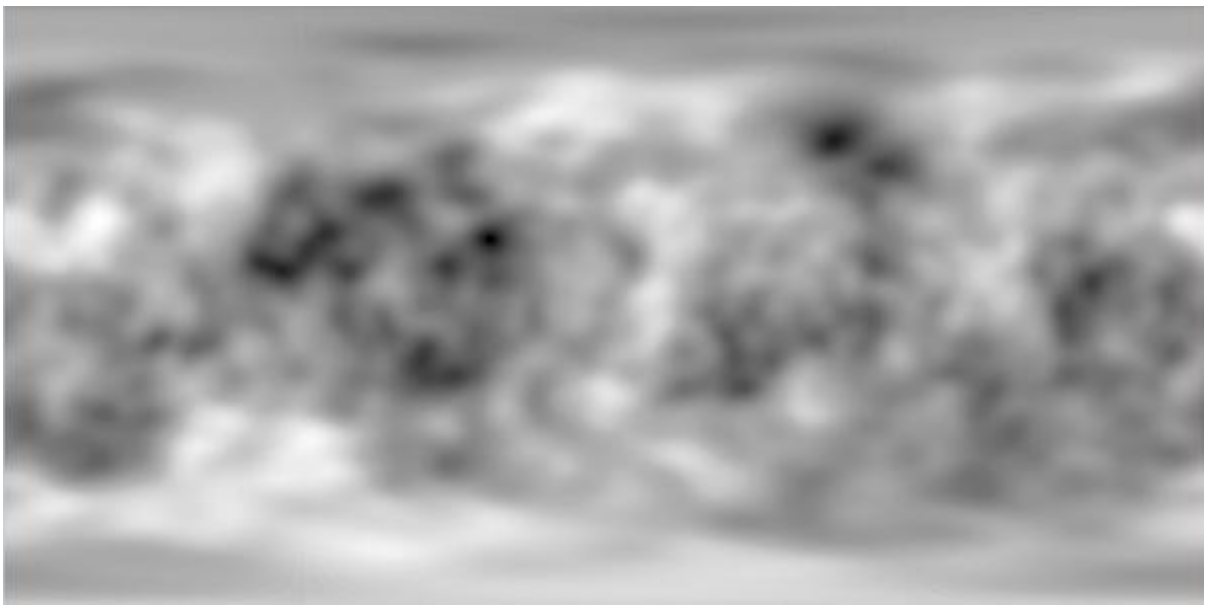
Pour l'instant, nous ne lisons que le premier champ (TFIELDS=1) dans le fichier.

Exemple du code source pour lire et afficher une image :

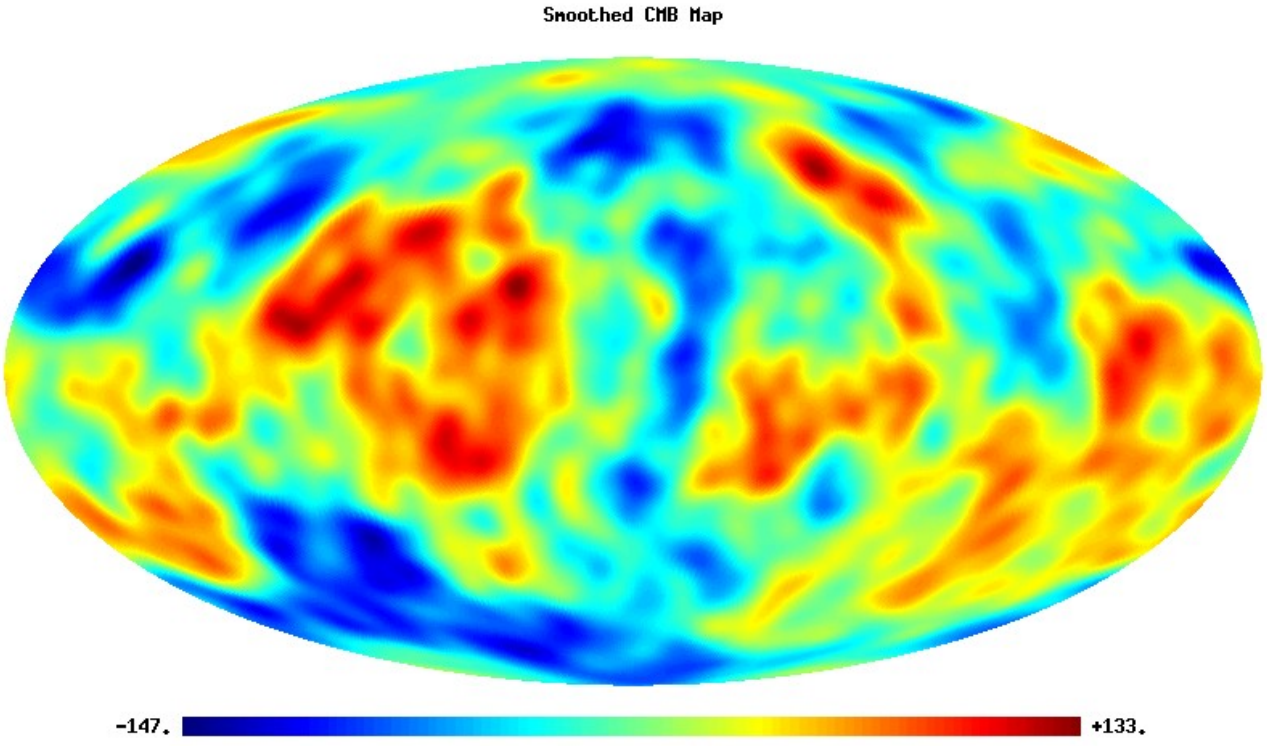
```
HealpixMap map = null;
Fits2HealpixMap reader = new Fits2HealpixMapImp();
Image img;        // lecture du fichier fits -> map
try {
   map = reader.fits2map(file);
} catch (Exception e) {
   System.err.println("Pb de lecture du fichier " + file);
   e.printStackTrace();
   return;
}

// création d'une matrice de coordonnées
byte[] pixels = new byte[width*height];
int i =0;
// pour chacun des pixels
for (int y = 0 ; y < height ; y++) {
   for (int x = 0 ; x < width ; x++) {
      int index = -1;
      // calcul les alpha, delta (ici, fixés pour les tests)
      double a = 180-x*(360.0/width); // échantillonnage entre 180;-180
      double d = y*(180.0/height); // échantillonnage entre 0 et 180
      try {
         // conversion en radians
         double theta = Math.PI*d/180.0;
         double phi = Math.PI*a/180.0;
         // récupère l'index du numéro de pixel dans la sphère
         index = map.ang2pix(theta, phi);
      } catch (Exception e) {
         System.err.println("Pb de conversion (a,d) -> #pix : " + a + " " + d);
         e.printStackTrace();
         return;
      }
      // conversion simple en 8bits pour avoir un meilleur rendu + normalisation entre min et max
      pixels[i++] = to8bits(map.get(index), map.getMin(), map.getMax());
   }
}
img = createImage( new MemoryImageSource(width,height,createDefaultCM(),pixels,0,width));
```

avec ceci on obtient l'image suivante :



l'image originale lue représentée par l'auteur est la suivante (avec une projection différente, ce qui implique la différence de mise en forme) :



2 - Evolutions

- il reste encore à savoir générer ces images à partir de celles dont nous disposons actuellement
- nous devons tester la lecture de fichiers avec seulement une zone du ciel
- il faudra prévoir de tester les vitesses d'accès en lecture de tels fichiers