MAPS_im2uv: Sky Brightness to Visibility File Converter

Leonid Benkevitch

June 22, 2010

Abstract

A part of the MAPS (MIT Array Performance Simulator) package, the MAPS_im2uv program reads a FITS sky brightness image file, reorders the data, Fourier transform it and writes it out in the binary format expected by `visgen`, a kernel MAPS program. If an appropriate source file is given, a point source is added to the uv grid for each source in the file. In this manual main features of MAPS_im2uv are explained.

Running from the command line

```
MAPS_im2uv -i infile -o outfile [OPTIONS]...
```

In its most basic mode of operation, MAPS_im2uv will read a brightness image from the FITS file `infile,`convert it into a visibility image using an FFT, and save the result in the binary file `outfile.`
It is optionally possible to embed into the visibility file the FFT image of one or more point sources, specified in a sourcefile (option `-s`). In this case, the specifications of the frequency, observing latitude, and LST are mandatory. [WHAT ABOUT SKY COORDINATES AND FLUX DENSITY??]
If the brightness of the input FITS image is in units other than `Jy steradian`$^{-1}$, a multiplicative constant invoked by the option `-n` ("normalizer") must be used to convert the brightness to default units. If the input FITS file is more than two-dimensional and it contains several brightness images for different frequencies, Stokes parameters etc., the indices in option `-a` specify which

image plane is to be read. Option `-q` prints a table of all the axes in the input FITS file. With option `-d` the program prints diagnostics at runtime.

Options and interpretations

`-i, --inputfilename`	input FITS file name
`-o, --outputfilename`	output file name (usually *.dat)
`-a, --axes`	provide subscripts for multidimensional FITS files For example, if file holds images for 12 different frequencies and 4 Stokes parameters for each frequency, then the option `--axes [5,2]` asks the program to read the image for the 5th frequency channel and 2nd Stokes parameter
`-s, --sourcefilename`	point source file name
`-f, --frequencyofsource`	frequency of source in MHz
`-h, --lstofsource`	local sidereal time UNITS????
`-l, --latitude`	observing latitude of point sources?????
`-n, --normalizer`	multiplicative constant to bring brightness to units of Jy/rad^2
`-t, --ewtranspose`	flip image east-west before processing
`-d, --debug`	print some variables and other information
`-m, --help`	print the help message similar to this text
`-q, --query`	print table of all axis names and dimensions from the header of the input FITS file (at option `-i`)

Details

The input FITS file

The input FITS file name is placed after the `-i` option. The brightness image data must reside in the Primary HDU (Primary Array) of the file. The current version of MAPS_im2uv requires that the image be square, having equal numbers of pixels numbers along `RA` and `DEC` axes. In the simplest case the input FITS file only contains one two-dimensional image with the axes order:

`RA` - right ascension,
`DEC` - declination.

However, the input FITS file may store multiple images for several frequencies, polarizations, etc. The Primary Array in such FITS files is multidimensional. MAPS_im2uv can read up to 16-dimensional FITS files with one restriction: the image axes, `RA` and `DEC`, must be the very first axes of the array (this implies that the multiple images are stored contiguously). All other axes can be ordered arbtrarily. For example, an input FITS file can have the axes as follows:

`RA`	2048	- right ascension,
`DEC`	2048	- declination,
`FREQ`	29	- frequency,
`STOKES`	4	- Stokes parameters,

where the numbers are for dimensions along each of the axes.
Over one run, MAPS_im2uv can only process one image indexed by the `-a` option. Continuing the example, in order to read the image at the 17^{th} frequency and 3^{rd} Stokes parameter, one needs to include the option

`-a[17,3]` or `-axes [17,3]`.

Note that the indices of the rightmost axes can be omitted to be regarded as equal to one by default. In the previous example, options

`-a[17]` and `-a[17,1]`

are equivalent. Also, if the input FITS file is multidimensional and there is no `-a` option on the command line, then the first brightnness image will be read, which is equivalent to the action of `-a[1,1,1,...,1]`.

Option `-q` (`-query`), when inserted after the `-i` option, can help to quickly reveal what are the dimensionality and axes of the input FITS file. Note that if the `-q` option is on the command line, MAPS_im2uv only prints out the table of FITS file axes (names and dimensions) and quits without any further processing.

Output file

The desired output file name should be specified after the `-o` option. The program stores the FFT visibility image in binary format. Commonly, the output file name is given a `.dat` extension.

Source file and related parameters

One or more point sources can be added directly to the output FFT image. The source file name specifying the properties of these point sources is specified after the option `-s`. This text file contains the list of point sources, one per line. The lines have the following format (in terms of the C scanf() function):

```
"%f %f %s %f %f %f", &az, &el, sourcename, &fd, &ref_freq, &spec_index.
```

With the source file name three other options related to the point sources added must be present on the command line:

`-f`, the frequency;
`-h`, the local sidereal time;
`-l`, the observing latitude.

Normalization constant

The brightness units of input FITS files can be expressed in arbitrary units. However, MAPS_im2uv requires that the brightness image be converted to Jansky per steradian. If the FITS file header has a keyword `BUNIT` with non-blank string value, MAPS_im2uv attempts to interpret it and convert the image to the units of `Jy/rad`^{2}. Examples of possible `BUNIT` values are

`Jy/beam`,
`Jy/pixel`,
`Jy/deg`^{2}.

If either `BUNIT` is not present in the header or it has an unrecognized value or has blank value, MAPS_im2uv issues the warning message saying that it assumes the brightness units are `Jy/rad`^{2} and continues.

The user can override this default behavior by providing the option `-n` followed by a numerical normalization constant. Then, prior to the processing, each pixel of the image will be multiplied by the constant reducing it to

Jy/rad^2, and the value of the BUNIT keyword in the file header will be disregarded.

Please note that this feature has not been implemented yet for all the units mentioned. If MAPS_im2uv does not understand the units from a particular FITS file and reports "assuming Jy/rad^2", one should manually calculate the normalization constant and apply it using the -n option.

East-West image transposition

In Astronomy, sky images are conventionally plotted in (RA,DEC) coordinates with RA increasing from right to left, and DEC increasing from bottom to top. Most of the astronomical sky images at the computer storage level obey this convention: the RA of pixels in FITS arrays decrease from left to right—i.e., from the west to the east.
However, one should be warned that the MAPS package at its current state assumes the opposite RA direction, where it increases from left to right, the west being on the left and the east on the right (like on geographical maps). This issue is planned to be corrected in near future. Until it is done, the user can use the option

-t or --ewtranspose

to make MAPS_im2uv east-west transpose (or simply east-west flip) the brightness image read from the input FITS file to render its mirror image before the processing. The action is analogous to that of the Matlab function fliplr().

Some examples

In the examples below the user command input lines start with the $ sign.

```
$ MAPS_im2uv.exe -i test_map.fits -o test_map.dat
```

Convert the brightness image test_map.fits to visibilities (i.e., FFT it) and save it in the binary file test_map.dat.

```
$ MAPS_im2uv.exe -i test4d_map.fits -q
CTYPE1:  RA---SIN, 2048
```

```
CTYPE2:  DEC--SIN, 2048
CTYPE3:  FREQ-LSR, 4
CTYPE4:  STOKES, 1
```

Print a table of all the axis names and dimensions from the file `test4d_map.fits`

```
$ MAPS_im2uv.exe -i test4d_map.fits -o test4d_map.dat -a[3,1]
```

Convert the brightness image of the 3^{rd} frequency channel and first Stokes parameter from the input file `test4d_map.fits` to visibilities and save the result in `test4d_map.dat`.

```
$ MAPS_im2uv.exe -i test4d_map.fits -o test4d_map.dat -a[3]
```

Same action as in the previous example, because all the dimensions omitted in the `-a` option are assumed to be unity by default.

```
$ MAPS_im2uv.exe -i modelsky.fits -o modelsky.dat -n 3.59e5
```

Adjust the units of the brightness image from `modelsky.fits`, converting them to `Jy/rad`^{2} via the multiplicative constant 3.59e5, using the `-n` option; then perform the FFT and save the the resulting visibility data in `modelsky.dat`.