

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
HAYSTACK OBSERVATORY

Westford, Massachusetts 01886

February 24, 2011

BRIGEN: Generator of Sky Brightness Images in FITS Format

Abstract

The **brigen** program is a part of the MAPS (MIT Array Performance Simulator) package. On input, it reads a list of celestial sources, and outputs a simulated brightness image in the form of a FITS file. The celestial sources are specified as elliptical Gaussians with the intensity, minor and major axes, position angle, and other parameters. Optionally, the user may provide one or more external images in FITS file (or files) to be added to the output as its background. The **brigen**-generated images are suitable for further processing with the **MAPS_im2uv** program, which applies the FFT to convert the brightness image into the visibility image. The latter in turn are used by the **visgen** program. The **brigen** program was created to replace the **LOsim** program. It is mainly based on \$SIM/LOsim/source/SumSources.c program.

1 BRIGEN Command line files and options

Usage:

brigen infile [outfile] [OPTIONS]

The **infile** is either a simulation project file or a source list file. The **outfile** is the name of FITS file to hold the generated image. If **outfile** is not present on the command line, **brigen** creates FITS file with the same base name as **infile**, but with the **.fits** extension.

-m number, --mgrid number	image RA grid size in pixels
-n number, --ngrid number	image Dec grid size in pixels
-a angle, --rasize angle	image RA size in angular units
-b angle, --decsz angle	image Dec size in angular units

```

-x angle, --racenter angle  image RA center in angular units
-y angle, --decenter angle  image Dec center in angular units
-s filename, --srcfile filename  name of the text file with
                                specification of sources
-i filename, --addimage filename  name of a FITS file with
                                image to be added to the output
-h, --help  help page

```

2 Angular units

The coordinates of the image center and its extents in Right Ascension (RA) and Declination (Dec) are specified in angular units. The **angle** format consists of a numerical value followed by the units specifier without whitespace in between. There are two kinds of numerical values: a real number and a triad, $\pm\text{ddd}:\text{mm}:\text{ss}$ or $\text{hh}:\text{mm}:\text{ss}$. The units specifiers after a real number are one of the following:

```

rad    radians;
deg    degrees;
'      arcminutes ( $\frac{1}{60}$  of a degree);
" or ' ' arcseconds (one double quote or two single quotes,  $\frac{1}{60}$  of an arcminute);

```

Examples of the angles in this format are: 0.17rad, 10deg, 112.35 ', 18000 " , 36000 ' ' .

The triads are traditionally used for RA and Dec specification. The RA is usually given in $\text{hh}:\text{mm}:\text{ss}$, where **hh** is “hours”, **mm** “minutes of hour”, and **ss** seconds as units of time. The seconds can be given as a fraction with decimal point. The $\text{hh}:\text{mm}:\text{ss}$ triad must be immediately followed by the **hms** units specifier. The Dec is preferably specified as a $\pm\text{ddd}:\text{mm}:\text{ss}$ triad, where $\pm\text{ddd}$ is “degrees”, **mm** is “arcminutes”, and **ss** is “arcseconds”, which may be a fraction with the decimal point.

Examples of this format are: 1:7:5dms; -21:52:43.765dms; 3:0:0hms; 17:23:34.29hms.

It is important to note that on the command line an angle parameter cannot contain whitespace. In the simulation project files the blanks and tab-

ulations are allowed between number and units. For example, 132 MHz; 15:30:00 hms etc.

3 Celestial source list

The celestial sources generated by **brigen** have the elliptical form. Within each ellipse the brightness is distributed as the elliptical Gaussian function. It has its maximum at the ellipse center and falls off towards the periphery. The list of Gaussian elliptical celestial sources can be provided either in a separate text file or as a section in the simulation project file. In either case it consists of lines, each line specifying a single source. The line consists of the following items:

```
ID Intens. Q U V SI x y Maj.ax. Min.ax. PA,
```

where

ID is any identifier not longer than 10 characters,
Intens. is the source intensity (i.e. the I Stokes component),
Q, U, V are the Stokes components describing the polarization,
SI is the source spectral index,
x is the ellipse center RA coordinate in arcseconds,
y is the ellipse center Dec coordinate in arcseconds,
Maj.ax. is the ellipse major axis in arcseconds,
Min.ax. is the ellipse minor axis in arcseconds,
PA is the position angle of the ellipse major axis in degrees.

The position angle is counted clockwise from the meridian going through the source center. It is actually the angle between the meridian and the major axis.

Note that the current version of **brigen** ignores the parameters Q, U, V, and SI.

4 Import of external images

Additional images can be imported from FITS files whose names are given at **-i** or **--addimage** command line options. Generally, there may be several FITS files, each one at individual **-i** option. The images in these files must

have the same RA and Dec numbers of pixels as given at `-m (--mgrid)` and `-n (--ngrid)` options. The resultant image in this case is the sum of the external images and the image generated from the source list.

Note that the current **brigen** version ignores the units given at the **BUNIT** keyword in the external FITS file headers, producing no scaling before the image summation.

5 Sources as elliptical Gaussian distributions

The **brigen** program composes the brightness image as the superposition of elliptical Gaussians, one for each source,

$$G(x, y) = \frac{I}{2\pi\sigma_x\sigma_y} e^{-\left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right)}, \quad (1)$$

where x and y are in the ellipse axes coordinate system (i.e. shifted and rotated with respect to the original axes), σ_x and σ_y are its major and minor axes, and I is the source intensity (flux). A Gaussian is not calculated for every node of the image grid. To save computational resources, it is only evaluated within an ellipse drawn around the source center. The ellipse parameters are such that the Gaussian values outside the ellipse are negligibly small and can be assumed equal to zero.

6 Simulation project file

The **brigen** program interface is based on the idea of a simulation project (or simply project) file common for all the MAPS components. Usually MAPS is utilized for solving a problem through simulation. The solution may include several stages: generation of a brightness image with **brigen**; its conversion into the visibility image with **MAPS_im2uv**; array observation simulation to produce visibilities for all the array baselines with **visgen**; storing the visibilities obtained from **visgen** in the UVFITS file using **maps2uvfits**. Currently the parameters used at the stages are stored in several files. Instead of managing many different text files with the parameters for different programs describing a single simulation it is suggested that all the frequently-changed parameters be gathered in a single project file. When a MAPS program is launched, the user provides the simulation project file name on the command line. For example, if the project name is **project1.txt**, the

brigen invocation may look like:

```
$ brigen project1.txt
```

If all the necessary parameters and links to external parameter/data files are present in the project1.txt file, the program will execute and save its result.

The project file consists of several sections, one for each of the MAPS programs. At present time, only **brigen** and **visgen** sections are implemented. The project file structure may be described as follows.

```
<simulation_name> simulation
brigen              section
                    <brigen parameters>
                    end          brigen
visgen              section
                    <visgen parameters>
                    end          visgen
                    end          <simulation_name>
```

The project file can have as many sections as needed. If, for example, the user currently works with only the **brigen** program, the project file may include just the **brigen** section:

```
<simulation_name> simulation
brigen              section
                    <brigen parameters>
                    end          brigen
                    end          <simulation_name>
```

It is supposed that most of the parameters specified inside the program section in the project file can also be given on its command line with corresponding options. The command line options have higher priority over the parameters in the project file. Thus the user by entering a parameter as a command line option can override the value of the corresponding parameter in the project file. For example, suppose the brightness grid sizes are set in

the project file `neb374.txt` with the line

```
imgrid_pixels    1024, 1024
```

The user can change the grid dimensions to 2048×2048 for a particular `brigen` execution by running it with the following options:

```
$ brigen neb374.txt -m 2048 -n 2048
```

All other parameters will remain intact.

The convenience of having all the parameters in one place from the user point of view is obvious. For the MAPS developer the project-based architecture is also beneficial, because all the MAPS programs are supposed to use the same piece of software to scan and parse the project file. The lexical analyzer (scanner) and the syntax analyzer (parser) are written with the use of standard generators, `Flex` and `Bison` (former `Lex` and `Yacc`). They will be described in a separate section below.

7 Project file grammar

The simulation project file has its own formal language described with the grammar. The grammar consists of two parts: the lexical analyzer or scanner and the syntax analyzer or parser. Both are located in the directory `$SIM/parser`.

The scanner is described as a set of regular expressions defining indivisible lexical elements (lexemae) of the project language. For example, the lexemae `ANGLE`, `FREQ`, and `DUR` are described as

```
ANGLE {NUM}[ \t]*("\\"|"'"|' '"|' '"|"deg"|"rad")
FREQ  {NUM}[ \t]*("mHz"|"Hz"|"kHz"|"MHz"|"GHz")
DUR   {NUM}[ \t]*("us"|"usec"|"ms"|"msec"|"s"|"sec"|"min"|"hr")
```

A more elementary lexema `NUM` for any integer or real number has description

```
D [0-9]
L [a-zA-Z_]
E [Ee] [+]?{D}+
FLOAT0 [+]?{D}+{E}
FLOAT1 [+]?{D}*"."{D}+{E}?
```

```

FLOAT2  [+~]?{D}+."{D}*{E}?
FLT  {FLOAT0}|{FLOAT1}|{FLOAT2}
INT  [+~]?{D}+
NUM  {FLT}|{INT}

```

The scanner is stored in the file \$SIM/parser/read_prj.l. It is compiled into the C program `lex.yy.c` with the standard UNIX tool Flex (former `Lex`). Each call to `lex.yy.c` returns another lexema read from the project file. The scanner subroutine is sequentially called by the parser.

The parser is specified as the BackusNaur form (BNF) in the file \$SIM/parser/read_prj.y. It is compiled into the C program `read_prj.tab.c` with the standard UNIX tool `Bison` (former `Yacc`). The BackusNaur form is a tree-like structure comprising all possible patterns for the project language sentence patterns.

The proposed grammar is common for all the MAPS programs. All the data formats, such as lists of sky sources, lists of array antenna coordinates, **brigen** and **visgen** parameters etc. are specified in BNF in a single file. This makes future changes or corrections to the data formats significantly easier.

One project file for several programs allows them to peek into the parameters of each other. For example, the field-of-view dimensions and its center coordinates in **visgen** section, if not specified, may be assumed equal to those from the **brigen** section.

The formal description of the project file syntax and its automated parsing allows good diagnostics of the errors in the project data. The parser catches any syntax error and reports the file name and the line where the error is encountered.

8 Image parameters in project file brigen section

In the simulation project file the parameters are specified in the format:

```
<parameter-name>    <value>[,<value>...]    [<comment>]
```

The comments are ignored by the scanner. They can have arbitrary format, standard for conventional languages: either `# <any-text>` or `// <any-text>`

or /* <any-text> */

All the **brigen** parameters are summarized below. In parentheses the corresponding command line options are given).

imgrid.pixels <number>, <number>

Brightness image size in pixels in RA and Dec dimensions (**-m** <number> and **-n** <number>).

imcenter.radec <angle>, <angle>

Brightness image center RA and Dec coordinates (**-x** <angle> and **-y** <angle>).

imsize.radec <angle>, <angle>

Brightness image extents in RA and Dec (**-a** <angle> and **-b** <angle>).

srclist <file-name>

Name of the file where the simulated celestial sources are specified (**-s** <file-name>).

addimage <file-name> or **addimage here**

Name of a FITS file with external image (**-i** <file-name>). The word **here** instead of the file name means that the source list lines immediately follow.

imfitsout <file-name>

Name of a FITS file where the simulated brightness image will be saved (a second file name on the command line).

All these parameters are optional. All the parameters must occur in the list only once, with one exception: several **addimage** lines may appear to specify several FITS files with external brightness images.

The list of celestial sources can be given directly in the simulation project file. To do this, the **srcelist** parameter should have the **here** keyword in place of the <file-name>. The contents of the source list file should follow this header, and the operator **end** **srcelist** should close the list. Here is an example of the direct source specification:

srcelist **here**

#	Type	Intens.	Q	U	V	SI	x,	y,	Maj.ax	Min.ax	PA
SrcE111	5		0	0	0	2	8500.0	6000.0	1000.0	300.0	30.0
SrcE112	7		0	0	0	2	6000.0	8500.0	3000.0	800.0	60.0


```

SrcCirc3 6      0 0 0 2 0000.0 5000.0 1000.0 1000.0 00.0
SrcCirc4 2      0 0 0 2 -5000.0 -9000.0 600.0 600.0 00.0
      end      srclist

```

9 A simulation project example

Below is given an example of a simulation project file `mwasim01.txt`. It contains both `brigen` and `visgen` sections. Although the `visgen` program currently does not use the project files, both `visgen` and `brigen` sections are checked for incorrect syntax even if the file is used by `brigen` only. If a section is not needed at the moment, it

```

/*****
* mwasim01                                     *
* MAPS simulation example project               *
* brigen and visgen sections are present      *
*****/

mwasim01      simulation

#
# brigen: generation of the brightness image composed
#         of the sources from 'srclist'
#
brigen      section
imgrid_pixels 1024, 1024 # Brightness grid dims. in pixels
imcenter_radec 7:39:24 hms, -26:25:52.0 dms # FOV center RA, Dec
imsize_radec 10 deg, 36000" # Grid dimensions along RA, DEC
srclist      here
#Type Intens. Q U V SI x y Maj.ax Min.ax PA
Circ1 0.05 0 0 0 2 0000.0 0000.0 20.0 20.0 0.0
Circ2 0.05 0 0 0 2 -9000.0 0000.0 20.0 20.0 0.0
Circ3 0.05 0 0 0 2 0000.0 -9000.0 20.0 20.0 0.0
Circ4 0.05 0 0 0 2 9000.0 0000.0 20.0 20.0 0.0
Circ5 0.05 0 0 0 2 0000.0 9000.0 20.0 20.0 0.0
Circ6 0.05 0 0 0 2 6364.0 6364.0 20.0 20.0 0.0
Circ7 0.05 0 0 0 2 -6364.0 6364.0 20.0 20.0 0.0
Circ8 0.05 0 0 0 2 6364.0 -6364.0 20.0 20.0 0.0
Circ9 0.05 0 0 0 2 -6364.0 -6364.0 20.0 20.0 0.0
Ellip10 5 0 0 0 2 8500.0 6000.0 1000.0 300.0 30.0

```

```

Ellip11  7    0 0 0 2  6000.0  8500.0  3000.0  800.0 60.0
Circ12   6    0 0 0 2  0000.0  5000.0  1000.0 1000.0  0.0
Circ13   2    0 0 0 2 -5000.0 -9000.0   600.0  600.0  0.0
Ellip14   1    0 0 0 2 -9000.0 -5000.0   500.0  100.0  0.0
Ellip15 10    0 0 0 2 -4000.0 -4000.0 10000.0   500.0 40.0
Ellip16   1    0 0 0 2  0000.0  0000.0   700.0  300.0 10.0
        end    srclist

addimage    extimg1.fits
addimage    extimg2.fits

imfitsout    brightness_image_mwasim01.fits # Output file
        end    brigen

#
# visgen: conversion of the source visibility image after
#         MAPS_im2uv into the visibility image generated
#         by the radiotelescope specified in 'array' file.
#
visgen section
array    /home/benkev/maps/array/mwa_32.txt # File with array coords
visout    test01_Visibility.dat          # Output file

obsspec    here # Observation specifications
FOV_center_RA = 7:39:24.38467 hms
FOV_center_Dec = -26:25:52.0 dms
FOV_size_RA = 36e3"
FOV_size_Dec = 36000.0 ''
Corr_int_time = 2.0 min
Corr_chan_bw = 0.032 MHz

Scan_start = GHA -7.8237889
Scan_start = 2006:265:16:00:00.3573 ydhms
        Scan_duration = 8.0 s
        Central_freq = 140 MHz
        Bandwidth    = 32 kHz
Endscan
end    obsspec
textout    test01_UVdata.txt
        end    visgen

```

```
end    mwasim01
```

This project may be given to **brigen** on its command line as below. Note that since the project file includes all the necessary parameters, no command line options are required:

```
$ brigen mwasim01.txt
```

The source list is inside the project file. Two external images from files `exting1.fits` and `exting2.fits` will be added to the generated elliptical Gaussian sources. The result will be saved in file `brightness_image_mwasim01.fits`.

Suppose the user decides to try to generate the same image on a finer grid. Or, for example, the grid dimensions are provided as parameters in shell script. Then the command line may look like this:

```
$ brigen mwasim01.txt -m 3072 -n 3072
```

If the user wishes to add one more external image to the result, then he or she calls **brigen** with the corresponding option:

```
$ brigen mwasim01.txt -m 3072 -n 3072 -i extimage5.fits
```

If one desires to change the source list to another one that resides in an external file, it can be provided on the command line:

```
$ brigen mwasim01.txt -i extimage5.fits -s SourceList_test01.txt
```

Then the source list in the project file will be ignored, because the command line options have the higher priority over the project file parameters.