MNRAS **000**, 1–4 (2020) Preprint 22 October 2020 Compiled using MNRAS LATEX style file v3.0

MCG Guide

N. Deg,★ K. Spekkens
[1]*CIRADA*

1 INTRODUCTION

Most spiral galaxies have reservoirs of rotating neutral HI gas. The HI gas rotation provides one of the best tracers of the galaxy potential. As such, kinematic models of the HI gas are particularly interesting.

Observations of the neutral HI emmission using radio interferometers arrives in the form of spectral cubes. These cubes have two spatial dimensions, and one frequency or velocity dimension.

There are a variety of existing kinematic modelling tools. In particular, there are three commonly used 3D tools, TiRiFiC, FAT, and 3DBAROLO. Each of these codes are built on an underlying tilted ring method of modelling galaxies. The different codes have different strengths based on the problems they were designed to solve.

In order to compare different codes, it is useful to have a standalone tool that can generate mock HI data cubes. To be sure, each of the 3D codes mentioned has the ability to build their own cubes. However, the focus of those codes is on the modelling of data, rather than on simply generating models.

For this reason, and a number of others, we have developed the MCGSuite set of programs. These can generate mock observations from tilted ring models, and, unlike other codes, can generate realistic tilted ring models from scaling relations.

This document focuses on the mock cube generator code, MockCubeGenerator (hereafter MCG). Sec. 2 describes the basics of tilted ring models and how MCG converts these models into observed cubes. For those who are already familiar with these models, Sec. 3 contains the installation instructions for MCG, while Sec. 4 describes how to use the code, and Sec. 5 describes the outputs of MCG.

2 TILTED RING MODELS

A tilted ring model is based on the idea that a disk galaxy can be approximated as a series of nested rings described by a set of geometric and kinematic parameters. Each ring is then populated by a set of tracer particles. These particles are then projected to the sky and 'observed'.

Figure 1 shows a cartoon of the process. Firstly, a set of ring parameters are describing the galaxy are given. In MCG, these are supplied in a text input file. The full set of ring parameters are listed in Table 1. Broadly speaking, there are five parameters that describe the projection of rings on the sky, 3 parameters describing the geometry of the rings relative to the galaxy plane, 5 parameters describing the motion of particles in each ring and a single parameter describing the brightness of particles within a ring.

Once the ring parameters are set, tracer particles are generated to fill each ring. The radial position of the particles is selected such that the particle density is constant across each ring. The vertical position of the particles is drawn from a $sech^2$ profile with a scale radius of z_0. The velocity of an individual particle in galaxy plane coordinates is:

$$v_{\theta,i} = V_{rot} + \frac{dv}{dz} f(z_i, z_{gradient}) \,, \tag{1}$$

$$v_{r,i} = V_{rad} \,, \tag{2}$$

$$v_{z,i} = V_{vert} \,, \tag{3}$$

where i is the i'th particle in a ring, and $f(z_i, z_{gradient})$ is 0 for $z_i < z_{gradient}$ and $z_i - z_{gradient}$ for $z_i > z_{gradient}$.

The brightness of each particle,

$$B_i = \Sigma * A/N \,, \tag{4}$$

where A is the area of the ring and N is the number of particles within a ring. However, determining the N is somewhat more complicated. For MCG, we have set

$$N = S\Sigma^{c_{mode}}/A_p \,, \tag{5}$$

where S is the cloud surface density in #/pixel units (the term cloud is used here to be consistent with the larger literature. In this context each particle represents a cloud of gas). The A_p term is the area of the ring in pixels. The use of pixel units here is due to the fact that the particles will be placed in a datacube with some pre-determined pixel size. The c_{mode} power is somewhat more complex. When $c_{mode} = 0$, the particle surface density will remain constant. However, when $c_{mode} = 1$, the individual particle brightness will remain constant across all rings. Both S and c_{mode} are set by the user for MCG.

The particles that are generated by the rings are then projected onto the sky. In practice, first each particle is inclined and then rotated by the position angle. They are then re-centered on their particle ring center coordinates. The line-of-sight velocity is calculated for each particle based on the velocity vector and geometric coordinates. Then a random line-of-sight velocity is drawn from a Gaussian with a mean of zero and $\sigma = V_{disp}$ and added to the particle's velocity. Finally, a systemic velocity is added to each line-of-sight velocity. The result of this is a set of particles with values of $(B, X^{'}, Y^{'}, v_i)$ for the brightness, projected position, and line-of-sight velocity respectively.

To generate the data cube, each particle is placed into a spatial and spectral cell based on both their projected coordinates and the datacube parameters. The cube parameters are listed in Table 2. The combination of the CRPIX and CRVAL parameterss determine the location of the pixel cube on the sky. It is important to note that if X_{cent} or Y_{cent} are not located within the cube, it is possible for the mock cube to contain no data.

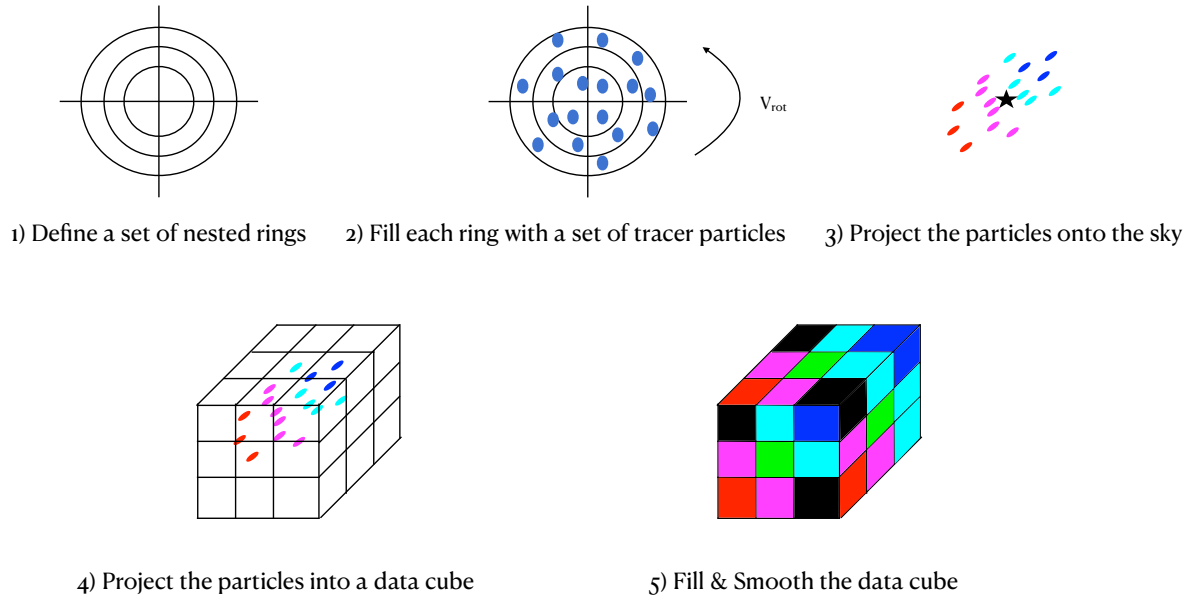

Figure 1. A cartoon showing the process of generating a data cube from a tilted ring model. In steps 3-5, the particles and cells are colored according to the line-of-sight velocity, with red receding and blue approaching.

Name	Description
R_{mid}	The midpoint radius of a ring.
R_{width}	The width of a ring.
X_{cent}	The projected X-coordinate center of the ring on the sky.
Y_{cent}	The projected Y-coordinate center of the ring on the sky.
Inc	The inclination of the ring.
PA	The position angle of the ring.
V_{sys}	The projected velocity of the ring relative to the observer.
V_{rot}	The rotational velocity of the ring in the galaxy plane.
V_{rad}	The radial velocity of the ring in the galaxy plane.
V_{vert}	The vertical velocity of the ring relative to the galaxy plane.
V_{disp}	The line-of-sight velocity dispersion of a ring.
dv/dz	The derivative of the rotational velocity as a function of height from the galaxy plane.
z_0	The height of the disk.
$z_{gradient}$	The height at which the dv/dz parameter begins affecting the rotation speed.
Σ	The surface density of the disk.

Table 1. The parameters that describe each ring in a tilted-ring model.

Name	Description
N_{pixels}	The number of pixels and channels in each dimension.
L	The dimensions of each pixel and channel.
CRPIX	The location of the reference pixels (channel).
CRVAL	The reference position of CRPIX.

Table 2. The parameters that describe the data cube.

The placement of the particles into the cube generates a 'point-source' cube. To get a realistic observation, this cube must be convolved with the beam PSF. MCG does this by calculated a Gaussian kernel based on user supplied parameters. This kernel is convolved with the cube using routines from the FFTW-3 library (Frigo & Johnson 2005). The beam PSF creates a spatial convolution, but real data may also have velocity smoothing as well. MCG implements this using a simple Gaussian smoothing kernel after doing the spatial convolution. However, it is worth noting that the velocity smoothing can be turned off in the input files. Ultimately, this convolution process produces a 'convolved-source' cube.

For many applications, a convolved-source cube is enough. However, realistic data cubes require the inclusion of noise. MCG takes a user specified value for the the RMS noise, σ. It use that value to create an initial 'noise-cube' where each cell is filled with flux from a random draw from a Gaussian. The noise cube is then smoothed using the same procedure as the 'convolved source' cube. It is important to note that a convolution process changes the RMS value of a cube. Therefore, the width of the Gaussian used for the initial random draws is

$$\sigma_{draw} = \sigma \left((2\sqrt{\pi SB_{maj} SB_{min}} \right) , \quad (6)$$

where SB_{maj} and SB_{min} are the beam's major and minor Gaussian axis size (not the Full Width Half Max [FWHM]) in pixels.

3 INSTALLATION

MCG is written in FORTRAN, and the installation is relatively painless. The larger difficulty is usually the installation of the two key dependencies. As discussed in Sec. 2, MCG uses the FFTW-3 libraries for the convolution steps. It also uses the CFITSIO libraries (Pence 1999) for writing Frrs format data cubes.

It is possible to install these dependencies in a variety of fashions, depending on the local machine. One of the simple ways to install them is from their source codes. The FFTW-3 source code and installation guide can be found here and the CFITSIO libraries can be found here. However, the MCGSuite package also includes versions of these packages in the */third_party/* directory

Installing the two libraries using the included versions of them can be done similarly for both libraries.

(i) Go to the appropriate folder in the *third_party* directory in the terminal.
(ii) Type **make clean** in the terminal.
(iii) Type **./configure** in the terminal.
(iv) Type **make** in the terminal.

Once the libraries are installed, MCG can be installed with minimal effort.

(i) Edit the *makeflags* file located in the */src/* directory so that the FFTW and Fits flags point to the locations of those libraries
(ii) In a terminal, go to the */src/* directory.
(iii) Type **make clean** in terminal.
(iv) Type **make** in terminal.

If everything proceeds correctly, this will generate an executable located in the */Programs/* directory.

4 CODE USAGE

MCG is controlled by a set of three text input files; a main file, a file describing the tilted ring model, and a file describing the data cube. Assuming that the user is in the main folder, the code itself can be run in terminal via

- ./Programs/MockCubeGenerator MainInputFile

A set of sample input files is found in the */Inputs/* directory. Running the code with those inputs should produce outputs like those located (here).

Figure 2 shows an example of the main input file. Essentially, the main file contains seven inputs. The first is the base output file name. The second is a switch containing the volume of outputs. In some cases, the user may only need the final cube, while in other cases they will need the convolved source cube, the point source cube, and the input files. This switch controls precisely how many files are saved to the main output folder. The third and fourth inputs are the names of the data cube header file (which contains parameters defining the data cube), and the tilted ring file (which contains parameters defining the tilted ring model). The next 2 inputs are a unit switch for the noise value and the noise value itself. The final input is the seed used in the random number generator. If this number is positive, the seed will be drawn using the system clock.

The next input file to discuss is the datacube input file. It is slightly more complicated than the main input file, but still relatively straightforward. Figure 3 shows an example of the datacube input file. The first input is the shape of the cube. The next input contains unit switches for the rest of the cube parameters. The third input defines the pixel and channel sizes for the cube. While the user may input differing sizes for the X and Y pixel dimensions, this feature is not fully implemented and giving different values will cause the output cubes to have an incorrect beam convolution as well as incorrect units. The next pair of input lines give the reference location and values for pixels in the cube. These are equivalent to the standard CRPIX and CRVAL parameters of a fits cube. It should be noted that the reference locations and values do not need to lie inside the cube itself.

The last four input lines in the datacube input file control the beam. The first defines beam FWHM for the major and minor axes, as well as the beam position angle. However, this feature, as with the differing pixel sizes, is not fully implemented and non-circular beams will currently result in incorrect noise profiles. The next input deals with the size of the beam kernel to calculate. The kernel will calculated on a grid extending to $2 \times N_\sigma \sigma_{beam}$ where N_σ is the specific input value and σ_{beam} is the Gaussian width of the beam. We recommend $3 \leqslant N_\sigma \leqslant 5$, as that balances realism with completeness. The final two parameters determine whether velocity smoothing will be included, and if so, the width of the smoothing Gaussian (σ_v of the velocity smoothing Gaussian kernel).

The third and final input file is the tilted ring model input file. Figure 4 shows a portion of this type of file rather than the full file as it is somewhat repetitive. The first input is the total number of rings in the tilted ring model. The second and third inputs give the cloud surface density, S, and cloud mode, c_{mode}, of the model. As discussed in Sec. 2, these two parameters determine the number of particles per ring. The fourth, and final non-profile input is the set of unit switches for the various ring parameters. The rest of the tilted ring input file is the tilted ring parameter profiles. For each of the rings, values must be given for each parameter listed in Table 1.

When all three input files are set, the MCG should rapidly produce a model data cube corresponding to the tilted ring model.

```
#       Base name for the output folder and all file names
MCG_TestGalaxy
#       Minimal, Moderate, of Full Outputs (0,1, and 2 respectively)
2
#       Name of the file containing the cube and beam definitions
Inputs/MockDataCubeHeader.in
#       Name of the file containing the underlying ring model
Inputs/TiltedRingModel.in
#       Noise Unit Switch (0=mJy/beam)
0
#       Noise Value
1.6
#       Random Seed
-1
```

Figure 2. An example main input file for MCG. Due to the code structure, it is necessary for all inputs to be present and in this particular order. While lines beginning with a # are not read, they are necessary for the correct input file format.

```
#       number of pixels and channels
512     512     127
#       Pixel Size Units (0=degrees, 1=arcsec) - Reference Pixel Units (0=degrees) -
Units (0=arcsec) - Beam Rotation Angle Units (0=degrees)
1       0       1       1       1       0
#       Pixel dimensions and channel size
10.     10.     8.55
#       Reference Pixel(channel) in each dimension (CRPIX values)
256.    256.    63.
#       Reference value in each dimension (CRVAL values)
2.77675430E+02    7.34348280E+01                1403.93164636
#       Beam dimensions (major, minor, position angle)
30.     30.     180.
#       number of sigma lengths to reach with the beam
5.
#       Type of velocity smoothing to use (0=none, 1=Gaussian)
0
#       The velocity smoothing sigma if using Gaussian smoothing (km/s)
4
```

Figure 3. An example datacube input file for MCG. A portion of the third line is cut off due to it's size, but the line itself lists the possible unit switch options. As with the main input file, the format of the datacube input file is fixed.

5 OUTPUTS

MCG will always place all the outputs from a particular run into a folder named by the first input in the main input file. That same input parameter will be used as the base name for all possible MCG outputs, of which there are three. The first, which is always produced by MCG, is the noisy convolved cube. This cube will be called *Name.fits*. If the output file volume switch equals zero, this file the only output from MCG. If the output volume switch is set to one, the unconvolved point-source cube is also output. That cube will be named *Name_SourceCube.fits*. Finally, if the output volume switch equals two, both those files and the noiseless, convolved will be produced. This final cube follows the naming convention of *Name_ConvolvedSourceCube.fits*.

REFERENCES

Di Teodoro, E. M. and Fraternali, F., 2015, MNRAS, 451, 3021
Frigo, M., Johnson, S. G., 2005, Proceedings of the IEEE, 93, 216
Pence, W. 1999, Astronomical Society of the Pacific Conference Series, 172, 487
Spekkens, K., Lewis, C., Deg, N., 2020, in prep

This paper has been typeset from a TEX/LATEX file prepared by the author.

```
#   Number of Rings in the model
84
#       The cloud mode you are using
0
#       The base cloud surface density
10.
#       Central Position Switch (0=degrees, 1=arcsec), Inc/PA Unit switch (0=degrees, 1=arcsec), Velocity
Units (0=m/s, 1=km/s), Brightness Units (0=Jy km/s arcsec^-2)
0       0       1       0
#       The parameters in each radial bin
#       Rmid    Rwidth  Xcent   Ycent   Inc     PA      VSys    VRot    VRad    Vvert   VDisp   dvdz
Sigma           z0      zGradStart
1.25    2.5     277.67543       73.434828       45.0    135.0   1403.93164636   11.1    0.0     0.0
8.0     0.0     0.0002097       0.2574  1.2870000000000001
3.75    2.5     277.67543       73.434828       45.0    135.0   1403.93164636   31.57   0.0     0.0
8.0     0.0     0.00024245      0.2683  1.3415
6.25    2.5     277.67543       73.434828       45.0    135.0   1403.93164636   48.86   0.0     0.0
8.0     0.0     0.00027465      0.29055 1.45275
8.75    2.5     277.67543       73.434828       45.0    135.0   1403.93164636   63.49   0.0     0.0
8.0     0.0     0.0003062       0.31375 1.5687499999999999
11.25   2.5     277.67543       73.434828       45.0    135.0   1403.93164636   75.89   0.0     0.0
8.0     0.0     0.00033695000000000003  0.3379  1.6894999999999998
13.75   2.5     277.67543       73.434828       45.0    135.0   1403.93164636   86.42   0.0     0.0
8.0     0.0     0.00036665      0.36295 1.81475
16.25   2.5     277.67543       73.434828       45.0    135.0   1403.93164636   95.38   0.0     0.0
```

Figure 4. An portion of an example TR input file. As with the other input files, the format of the file is fixed.