

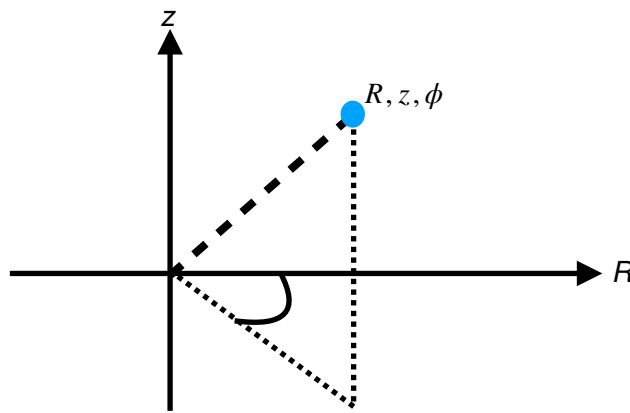
Synthetic Observations for ALMA

A simple parametric disc model

This tutorial will set you up with the tools to produce synthetic observations of axisymmetric protoplanetary discs. You will use TORUS to set up a model disc and compute a synthetic molecular line data cube. You will then process this using CASA to account for the ALMA instrumentation response. Successful ALMA proposals have been supported using models just like these.

1. Preamble: the simple disc model

We will set up a disc following the simple parametric approach of [Williams and Best \(2014\)](#). We will be working on a 2D-cylindrical model. It is in an R, z coordinate system that is axisymmetric.



In the radial direction (R), circumstellar discs have a structure set by the balance between pressure, gravity and centrifugal force (this can be solved for, but we will introduce a parametric description for it below). In the vertical direction the structure is set by the balance between pressure and gravity and so the density at some R varies with z as

$$\rho(R, z) = \rho_{mid} \exp(-z^2/(2H^2)) \quad (1)$$

Where H is the scale height, defined as

$$H = \frac{c_s}{\Omega} \quad (2)$$

Where c_s is the mid-plane sound speed and ω the Keplerian angular velocity

$$\Omega = \left(\frac{GM_*}{R^3} \right)^{0.5} \quad (3)$$

G here is the gravitational constant, M_* the stellar mass.

The radial variation of the disc density is described in terms of the surface density, which is what you get by integrating the density through the disc at some R . It has the form

$$\Sigma = \Sigma_0 \left(\frac{R}{R_c} \right)^{-\gamma} \exp \left[- \left(\frac{R}{R_c} \right)^{2-\gamma} \right] \quad (4)$$

At some R, z we translate between this surface density and the local density using equation 1 above and $\rho_{mid} = \Sigma/H$.

The thermal structure is parameterised in terms of a mid-plane and upper temperature profile, which we smoothly transition between
 The mid-plane temperature profile is

$$T_{mid} = T_{mid,1AU} \left(\frac{R}{AU} \right)^q \tag{5}$$

And that of the “atmosphere”

$$T_{atm} = T_{atm,1AU} \left(\frac{R}{AU} \right)^q \tag{6}$$

We transfer from one to the other through

$$T(r, z) = \begin{cases} T_{mid} + (T_{atm} - T_{mid}) \left[\sin \left(\frac{\pi z}{2z_q} \right) \right]^{2\delta} & \text{if } z < z_q \\ T_{atm} & \text{if } z \geq z_q \end{cases} \tag{7}$$

Where Z_q is 4 times the vertical scale height H . The above combines in TORUS to set up a disc as in Figure 1.

We are going to be producing position-position-velocity (PPV) data cubes in a line of carbon monoxide (CO). CO can actually be heavily depleted from the gas phase in cold, dense regions, where it “freezes out onto dust. There is the option to account for this in the model by removing CO where the temperature is less than 30K and the hydrogen density more than $3 \times 10^4 \text{ cm}^{-3}$.

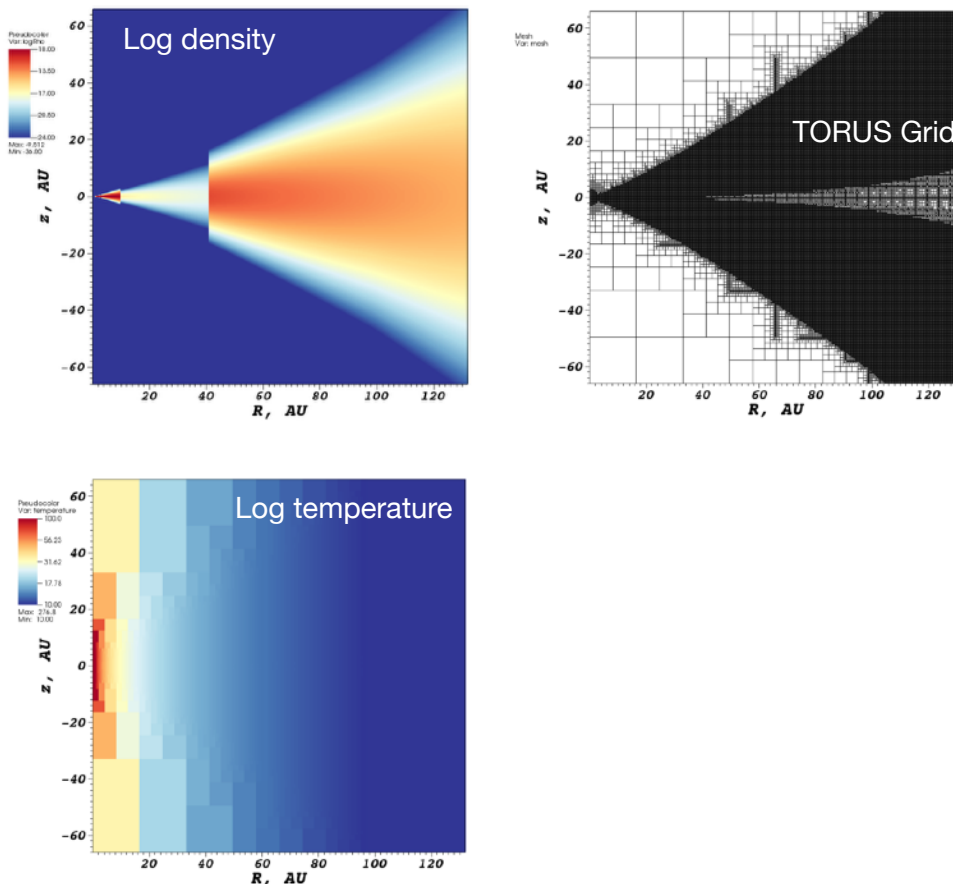


Figure 1. The density, grid, temperature and molecular abundance in an example parametric disc

3. Producing a synthetic CO datacube with TORUS

TORUS has a routine to set up a parametric disc as described above. This is done in such a way that you can customise the disc model (the density, temperature, size, etc.) from an input file (parameters.dat) rather than having to go into the guts of the code to make changes¹.

Our objective: use TORUS to produce a synthetic CO datacube for our parametric disc model.

To do this you need to compile the code (I believe this should have already been covered) and construct a parameter.dat file. This latter basically tells TORUS what kind of calculation is being run, the nature of the grid and what kind of outputs to produce. I have provided one that will work for us in this tutorial, but you can play around with it, changing things like the disc size, molecular line/species to observe, resolution, and so on. The parameters themselves are all well commented in the parameters.dat file.

Copy your own torus executable and the parameters.dat file and other scripts provided here

<https://drive.google.com/open?id=1bINcnTezCcVgNrKtCkqM3oIAfoLUja6t>

into some directory where you will run things and produce outputs. Take a moment to look at the parameters file and then run it from the terminal with, e.g.

```
./torus.openmp
```

Where I say e.g. because your executable might have a different extension (e.g. torus.gfortran). TORUS will give all the usual setup output and then start producing a bunch of information as follows

```
! DELTAV(v/c): -1.3343E-5   V (km/s):   -4.0000   Average Intensity:  5.1664E-16   FLUX:    5.1059E-16   Trad (K):  -7.1183E-01
! DELTAV(v/c): -1.3073E-5   V (km/s):   -3.9192   Average Intensity:  8.9890E-16   FLUX:    8.8837E-16   Trad (K):  -6.1820E-01
! DELTAV(v/c): -1.2803E-5   V (km/s):   -3.8384   Average Intensity:  8.9896E-16   FLUX:    8.8843E-16   Trad (K):  -6.1818E-01
```

This is the code going through the different velocity channels and constructing the image.

! A quick warning that since the resulting cube is 3D, significantly increasing the spatial or spectral resolution can have a big impact on the required runtime and the size of the resulting data. The disk space requirements get even more significant in the CASA (ALMA processing) stage of this tutorial. Unfortunately, sometimes high spectral/spatial resolution is necessary. E.g. for one ALMA cycle 7 proposal I used > 150GB of disk space (dominated by CASA, not TORUS).

When the run has finished it will have produced a datacube called something like “intensity_<...>.fits”. There will also be some .vtu files that you can open with something like VisIt (<https://wci.llnl.gov/simulation/computer-codes/visit/>) or paraview (<https://www.paraview.org/>) to look at the model grid/variables themselves. For example I used VisIt to produce the plots in Figure 1 using rho.vtu and molResults.vtu. Feel free to spend some time exploring these vtu files if you want to. It is usually important to check these to a) ensure that the model is set up correctly and b) to interpret your observables).

So you should now have a first example datacube, but we want to simulate ALMA observations in this tutorial, for which we need CAS. If you already have casa installed you can skip to section 4, otherwise go to 3 and install/setup.

¹ However, if you are interested, the relevant stuff is “WB2014” and “WilliamsAndBest2014” in amr_mod.F90 of the TORUS source code.

4. Getting set up with casa and casapy

If you don't already have it, download and install casa from here <https://casa.nrao.edu/>


5. Accounting for ALMA instrumentation response in line data cubes

You should now have TORUS able to produce a synthetic datacube and have casa installed. Lets start by running casa, from the terminal:

```
casapy
```

You will then see a new prompt like

```
CASA <1>:
```

type viewer and hit return. A GUI will appear from which you can open  the data cube we produced and take a look at it. To do this, click on the intensity_xxx.fits file and select "raster image". You will need to switch back to the viewer display panel to actually see it (See Figure 2 below). There will be a slider on the right that allows you to scroll through the different velocity channels. That and some other features (zoom etc.) are highlighted in Figure 2.

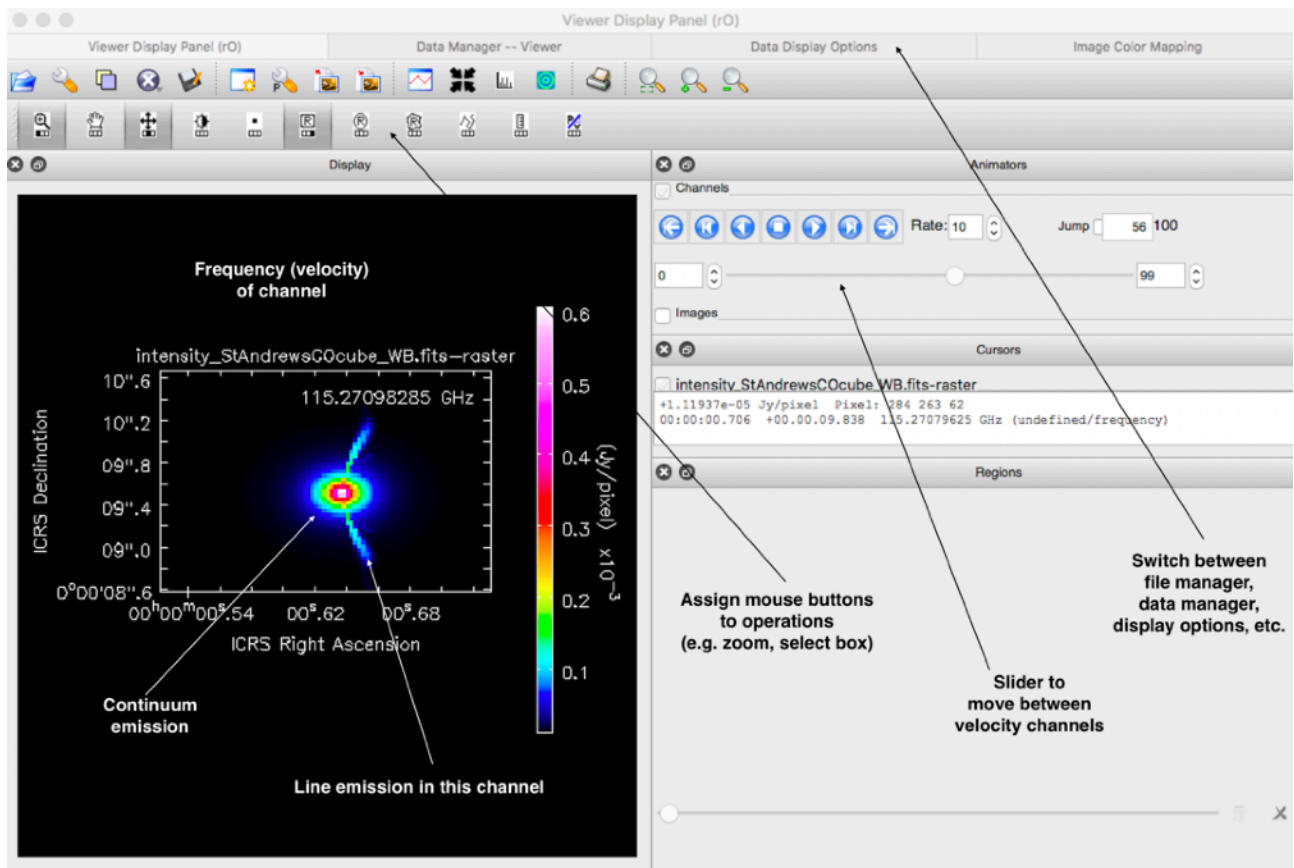


Figure 2. The CASA viewer interface

This is a raw image produced from TORUS so doesn't account for the fact that we want to predict what ALMA would see, so lets go ahead and account for ALMA as the observing tool.

! Generally this can be quite challenging as there are lots of subtle tricks. Your models will be much more robust and effective if you can collaborate with an expert on ALMA/interferometry. In all projects that I have worked on with synthetic observations for ALMA there has been someone with lots of past experience working with ALMA data involved.

5.1 Simulating an ALMA observation (simalma)

The example script that I have provided for doing the ALMA processing is called SIMALMA_CO_concat.py and can be downloaded here:

https://drive.google.com/open?id=1WcIB_ZUxRE4gqjYA-zQz30L4GqWR-ja4

The script itself is very heavily commented so I won't repeat all of the same information here, though we will go through changing some parts of it shortly. We will be using a routine called "simlama" which conveniently... simulates ALMA. For other instruments (e.g. SMA) you will probably need to use both "simobserve" and "simanalyze", which essentially split the observing and cleaning. Simalma does everything in one go. We will also automatically clean our synthetic image, which is usually fine for proposals, but if you are interpreting some real system you will probably want to manually clean the data in the way the real data was treated (and apply whatever other black magic the observers have to the real data, see my boxed point above!).

What is worth mentioning now is that the script I have provided will use two ALMA configurations to "observe" our model. One has a longer baseline, which means that it has higher resolution but also cannot retrieve larger angular scales. We therefore use a second, shorter, baseline to fill in the larger scales. Once everything is run, you can view the large, small and combined resulting images separately (discussed below).

Once you have taken a look through the script file (or simultaneously if you prefer), run the example that I provided. From the casapy prompt (not the "casa" prompt if that was what you used above) run

```
CASA <1>: execfile("SIMALMA_CO_concat.py")
```

! Caution: working with CASA to post process models can quickly use up a lot of disc space. e.g. a 700MB data cube can end up requiring 10's of GB of disc space. CASA scripts can also take a long time to run, e.g. if the resolution/time on source are high.

It will probably take around 10 minutes to run.

5.2 Subtracting the continuum (imcontsub)

The script I provided first simulates the observation and cleans it automatically (SIMALMA). It then does a continuum subtraction. This means it tries to remove emission associated with the continuum, leaving us just with the (molecular) line emission. This is done with the IMCONTSUB routine. The way it operates is that you define some velocity channels that are "line free", where there is only continuum emission. It will then average over these channels and subtract it from all channels. For this reason, it is always worth ensuring your cube created by TORUS spans wide enough a velocity range so that there are some line free channels.

5.3 Creating moment moment maps/PV diagrams.

We now have a synthetic ALMA CO data cube of a disc. Useful representations of the data in these cubes are moment maps and position-velocity (PV) diagrams).

Moment maps

Provide a useful summary of the 3D data on a pixel-by-pixel basis.

A **moment 0** map is essentially a representation of $\int I_\nu d\nu$ along the velocities/frequencies in the cube. It represents the distribution of matter.

Similarly, a **moment 1** map is essentially $\int \nu I_\nu d\nu$, so is frequency/velocity weighted and hence tells us whether material at some point is typically red or blue shifted. In CASA moment maps are produced using the immoments command

Position-velocity diagrams

A position velocity diagram involves taking a cut through the spatial axes of the cube and plotting the emission along this cut as a function of velocity/frequency.

The script provided finishes by producing moment 0, 1 maps and a position velocity diagram.

This will produce new outputs that you can view in CASA.

5.3 Viewing outputs

The processed data/maps will be in your project directory (defined in the CASA script). You can view the .line files for the processed data cubes (there will be 1 for each configuration and a “concat” one that combines them). There will also be .mom0, .mom1, .pv files etc that you can view.

6. Further exploration

6.1 Different disc and observing parameters

You can try altering (in no particular order)

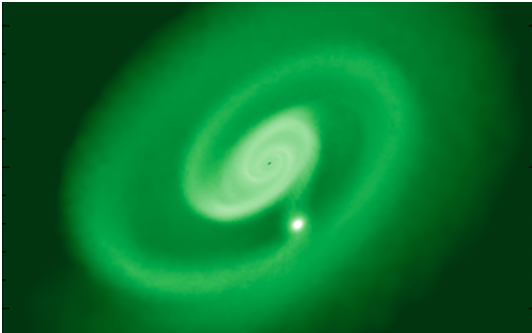
- disc mass/size
- transition (you would need to update the CASA script with the right frequency)
- spatial/spectral resolution
- inserting a ring or inner gap
- a different species
- the drop model
- different time on source
- different ALMA configurations (resolution, just using 1 configuration etc.).
- using the actual correct peak flux from the TORUS (recall that this will require a lot more time on source).

6.2 Self-gravitating discs

If you want you can download an SPH model of a self-gravitating disc with spirals (see Figure 3 for an example of this kind of scenario) and parameter file for processing it here

https://drive.google.com/open?id=1VVue4OEs7864avnUZBmXpY_L9v1TWvEp

I'll leave it to you to construct the CASA script(s).



Useful ALMA/CASA resources

Reference material

<https://almascience.nrao.edu/documents-and-tools/cycle7/alma-proposers-guide>

Table 7.1 in particular is useful

<https://almascience.nrao.edu/documents-and-tools/cycle7/alma-technical-handbook/view>

https://casaguides.nrao.edu/index.php/Main_Page

https://casa.nrao.edu/casa_cookbook.pdf

Manual pages for commands in this tutorial

<https://casa.nrao.edu/docs/TaskRef/simalma-task.html>

<https://casa.nrao.edu/docs/TaskRef/simobserve-task.html>

<https://casa.nrao.edu/docs/TaskRef/simanalyze-task.html>

<https://casa.nrao.edu/docs/TaskRef/imcontsub-task.html>

<https://casa.nrao.edu/docs/TaskRef/immoments-task.html>

ALMA sensitivity calculator

<https://almascience.eso.org/proposing/sensitivity-calculator>

Other references

Harries et al. (2019), *Astronomy & Computing*, 27, 63

Rundle et al. (2010), *MNRAS*, 407, 986

Williams & Best (2014), *ApJ*, 788, 59