G-virial Method – ReadMe

Guang-Xing Li gxli@mpifr.de, ligx.ngc7293@gmail.com

December 15, 2014

1 Introduction

Here we provide a simple realization of the G-virial method as described in the paper *G-virial: A Gravity-based Method to Quantify the Structure of Molecular Clouds*.

2 Description of the files

1. `compute_givirial.py` The file that contains the function to calculate the G-virial. The code can be used both in the command line as a script and used as a python library.

2. `example.py` A file illustrating how to use `G-virial` code in another python script. You are recommended to read it.

3. `ngc1333.fits` A test data file from the COMPLETE survey.

3 Use the code

3.1 The easy way – as a python script

```
python compute_gvirial.py ngc1333.fits ngc1333_gvirial.fits 250 1e5
3
```

Description of the parameters:

1. 250, distance of the object, in unity of pc.

2. 1e5, c_0, a parameter in the model. It is chosen to be comparable with the sound speed.

3. 3, padding for the FFT. a larger values gives better results at the endges of the maps but requires more memory and computation time.

In this example (`compute_gvirial.py`), the column density is converted from the observed $T_{\rm b}$ through

$$n_{\rm H_2} = 5 \times 10^{20} \times \frac{T_{\rm b}}{K} \times \frac{\delta {\rm v}}{1 {\rm km\ s^{-1}}} \ , \tag{1}$$

where δv is the width of the velocity channel. You can find this conversion in the line defining `k_fco` in `compute_gvirial.py`.

3.2 Calling `G-virial` as a python function

This is illustrated in `example.py`. Users are suggested to read the program.

The major input is the column density distribution in the form of a 3D position-position-velocity data cube. The cube have to be 3D and structured as `data_density(v, y, x)`. The value at each vorxel stand for the column density at given position (x, y) within the interval $(v - 0.5\ dv, v + 0.5\ dv)$. The column density is measured in unit of g cm^{-2}.

The main function that calculates the `G-virial` is the function

```
compute_gvirial(data_dens, dxy, dv, c0, npad_fft).
```

Here `data_dens` is the density distribution, `dxy` is the vorxel size in the (x, y) direction, and `dv` is the vorxel size in the v direction. `c0` is a parameter and `npad_fft` is the parameter determining how much padding will be made during the FFT calculation.

3.3 Padding in FFT

The padding is implemented to tread the end effects. The parameter `npad_fft` determines how large the padded image is compared to the original image. As a result a larger value of `npad_fft` will make the program consume more memory.

4 Interpreting the results

The results are arranged in the same format as the input data. The axes are arranged in the order of (v, y, x). The map should look much more smooth compared to the input. The value of the map should be around the order of 1. Larger G-virial is related to a higher chance of being gravitationally bound.

If the results are very large or small (e.g. larger than 10^3 or smaller than 10^{-3}), first check the conversion into the column density and then the distance of the object.

5 Memory Usage

The amount of memory needed is about ~ 3 times the size of the cube. If the computer does not have enough memory, please try the followings:

1. Try a smaller data cube (e.g. cut the image and only keep the important parts.)
2. Try a smaller padding(smaller `npad_fft`).
3. Find a computer with more memory.

6 Dependencies

The code depends on the following python packages:
`numpy, astropy`