CTA200H project for week 2:
exploring halo models of line intensity

Dongwoo Chung

5th May 2021

The key ingredients of a halo model of line intensity are the halo mass function and the relation between halo mass and line luminosity. We'll learn how to leverage existing Python packages to explore some predictions about the total line-intensity signal.

0 Prerequisites

0.1 Software

You might need to install specific Python packages if they were not already configured during week 1 of the course. You will need to make sure `astropy` and `hmf` are available, amongst other packages. We can go about setting these up one of two ways:

1. Anaconda3 on the CITA workstations (`module load python/conda3-5.3`) includes `astropy` but not `hmf`. You can still install the latter via `pip`, but since `hmf` requires `camb`, you need to either let `pip` build and install `camb` or install the Anaconda version of it (`conda install -c conda-forge camb`). At least the former option may also require you to enable compilers in your environment (`module load gcc`).

2. To run things on a local computer, it's likely you can get away with installing vanilla Python 3 and then installing packages as necessary via `pip`. The one place where this may be troublesome is in building `camb` for `hmf` if you are on a Windows or Mac computer. Using the CITA workstations works around such problems but may result in high latency in certain contexts like interactive plotting. Another workaround is to simply install an older version of `hmf` that does not require `camb` (`pip install hmf==3.0.3`), but note that this will result in behaviour different from what is documented for the latest stable version.

The documentation for `hmf` is comprehensive and slightly intimidating but the tutorial is useful to work through and should get you up to speed on the basic usage that is necessary for this project.

0.2 Science

The halo mass function models the dark matter halo volume density dn/dM in comoving space per differential mass bin, as a function of halo mass and redshift. Often the mass is expressed in 'units' of $h^{-1}\,M_\odot$, which really means that the mass value given needs to be multiplied by h^{-1} and then is in units of $M_\odot$. Similarly comoving volume density is often in 'units' of $h^3\,\mathrm{Mpc}^{-3}$, really meaning that the density value given needs to be multiplied by h^3 and then is in units of Mpc^{-3}. This should explain units given in the documentation for `hmf`. If you find the use of h very confusing (which is normal!), arXiv:1308.4150 is an exceptionally useful read and includes a 'little-h' cheat sheet.

The main quantity we'll be calculating is the (cosmic average) specific intensity I for an emission line. The first several equations of arXiv:1706.01471 outline the appropriate calculation of I. Note however that we will deal with the [C II] (ionised carbon) line in addition to the CO lines described in that reference, and note also that we will deal with a mass-dependent form of f_{duty}. You are strongly encouraged to make use of definitions in `astropy.cosmology`, `astropy.constants`, and `astropy.units` in adapting the calculation to code.

0.3 Getting familiar with some data tables

If on a CITA workstation, navigate to `/mnt/raid-cita/dongwooc/SAM_LM_data/` and please verify that you can access the `.txt` files in the directory. These files are products of arXiv:2009.11933 (provided via private communication and not yet public), and describe simulated quantities for various lines across bins of halo mass and redshift. If you will be running scripts directly on your own computer, please copy these files to a local directory.

Open any of the `.txt` files in a text editor and inspect the contents. Note that the comments indicate the quantities described in each column of the data, and the units. For instance, the first two columns are the central value of $\log M/M_\odot$ and the average value of $L_{[\mathrm{C\,II}]}$ (units of $L_\odot$) for the mass bin described by each row. Proceed once you understand how you would interpret the contents of the file.

1 CO intensities at redshift 2.8

1. Using the `Planck13` cosmology built into `astropy.cosmology`, define a `MassFunction` instance from `hmf` with redshift set to $z = 2.8$. Then, using the formalism of arXiv:1706.01471, the halo mass function given by your `MassFunction`, and the $L(M)$ values given by `2.8.txt`, calculate the integrated mean intensity I for all CO lines (1–0 through 5–4), in janskys per steradian.

   (Some caution is warranted around the 'per steradian' bit, which is often left implicit due to the steradian being a dimensionless unit. If you are dividing at any point by 4π in this calculation, it's quite possible you are actually dividing by 4π sr.)

2. What is the corresponding mean brightness temperature $\langle T_{\mathrm{CO}}\rangle$, in μK? Plot the ratio of the temperature for each CO line ($J \to J-1$) to the temperature for CO($J = 1 \to 0$), as a function of J.

3. Try changing the minimum halo mass $M_{\min}$. Show for all CO lines how $\langle T_{\mathrm{CO}}\rangle$ changes with $M_{\min}$. At $z = 2.8$, what values of $M_{\min}$ are sufficient to capture over 95% of the $\langle T_{\mathrm{CO}}\rangle$ calculated with $M_{\min} = 10^{10}h^{-1}\,M_\odot$?

2 CO and [C II] intensities across cosmic time

1. Using all of the provided data files, and updating the `MassFunction` redshift as you go, plot $\langle T_{\mathrm{CO}}\rangle$ for each CO line as a function of redshift. How do the $(J \to J-1)/(1 \to 0)$ ratios evolve? Are higher-J lines more or less observable at high redshift compared to CO(1–0)?

2. For $z = 2.8$, you found a maximum value of $M_{\rm min}$ that still captured 95% of the total integrated temperature. Calculate and plot this value of $M_{\rm min}$ as a function of redshift.

   If the resolution of a cosmological simulation only allows us to identify simulated halos with virial mass above $3 \times 10^{10}\, M_\odot$ at each redshift, can we use it to reliably model CO emission at $z = 2$? at $z = 5$? at $z = 8$?

3. Now calculate I for [C II] as a function of redshift. Plot your calculation in comparison to other predictions in the literature (arXiv:1410.4808, arXiv:1504.06530, arXiv:1812.08135).

3 *If time allows:* [C II] observations and CO interlopers

1. Instead of as a function of redshift, now plot the [C II] intensity as a function of observing frequency $\nu_{\rm obs} \in (200, 420)$ GHz.

2. An observation of [C II] will have CO line contamination from lower redshift. Calculate the CO line intensities at each [C II] observing frequency (interpolating and extrapolating as necessary, but accounting for the fact that a redshifted CO line is not going to be observed at frequencies higher than its rest frequency), and plot the sum of the CO line intensities on top of the [C II] intensity as a function of observing frequency. At what frequencies is the [C II] dominant over the CO foregrounds, or vice versa?

3. *Extra-optional:* Extrapolate the change in CO temperature with J at each redshift to higher J (up to $J = 12$) and re-plot the sum of CO intensities to include these extrapolated values. How does this change your previous answer?

   (We would never extrapolate in such a cavalier manner for actual signal forecasting, but the exercise is an interesting one purely in the context of getting to grips with various scientific Python functions.)