ASTP 720 - Homework 2 - Multiwavelength Galactic Structure

Due February 11, 2020

The Structure of Galaxies

The mass of galaxies is dominated by dark matter, which explains many observables in galactic rotation, galaxy cluster interactions, and cosmology. One of the earliest pieces of evidence was that of galaxy rotation curves, in which they are observed to be flat out at some radius, i.e., stars and gas are moving at the same material whether they are at 10 or 20 kpc, for example.

The Navarro-Frenk-White (NFW; 1995c) density profile is a model for how the mass density ρ of a dark matter halo behaves as a function of radius r. They proposed a form

$$\rho(r) \propto \frac{1}{(r/r_s)(1+r/r_s)^2}, \tag{1}$$

where r_s is a characteristic radius that is further parameterized as $r_s = r_{200}/c$, where r_{200} is the "virial radius" where the density reaches 200 times the critical density of the Universe, $\rho_{\rm crit}$ (so $200\rho_{\rm crit} = M_{200}/(4\pi r_{200}^3/3)$), and c is a dimensionless "concentration" factor. The higher c is, the smaller r_s is and so the more concentrated the halo.

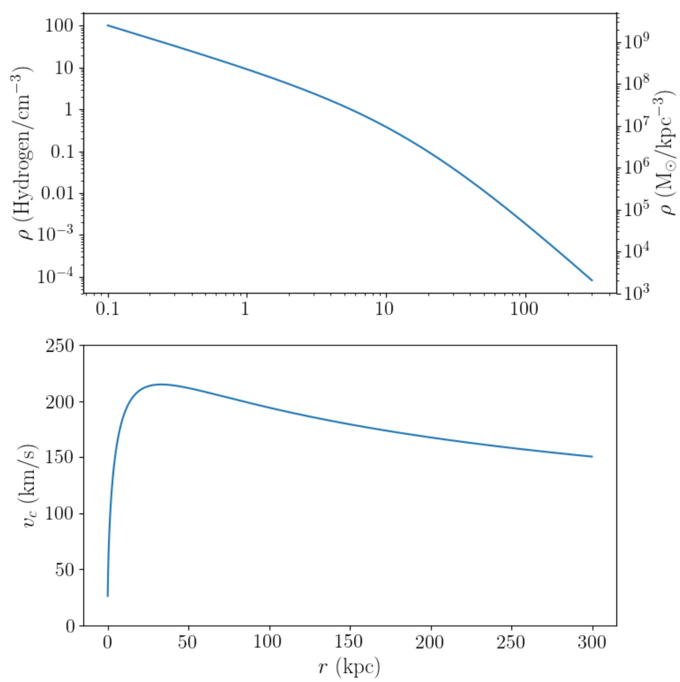
The circular velocity predicted from this fit is given by

$$\left[\frac{v_c(r)}{v_{200}}\right]^2 = \frac{1}{x}\frac{\ln(1+cx) - \frac{cx}{1+cx}}{\ln(1+c) - \frac{c}{1+c}} \tag{2}$$

where v_{200} is the value of the velocity at the radius r_{200} and $x \equiv r/r_{200}$. The circular velocity is related to the mass interior to that radius by equating the gravitational force with the centripetal force, i.e.,

$$v_c(r) = \sqrt{\frac{GM_{\rm enc}(r)}{r}} \quad \rightarrow \quad M_{\rm enc}(r) = \frac{rv_c^2(r)}{G}. \tag{3}$$

And so, the mass of the halo interior to 200 kpc sets the circular velocity v_{200}. The two functions might look something like what is shown to the right ($c = 15$, $v_{200} = 160$ km/s).



[1.] Write a numerical calculus library, one that can perform at least one derivative (e.g., the symmetric one) and at least three integration functions inclding the midpoint rule, trapezoidal rule, and Simpson's rule.

[2.] Using your chosen parameters of c and v_{200}, numerically determine the mass enclosed $M_{\rm enc}(r)$ (and plot), the total mass of the dark matter halo M, and then also $M(r)$, the amount of mass in a little shell around $r \pm \Delta r$ (i.e., what does the mass profile look like?), and $dM(r)/dr$. Compare this by repeating, holding c fixed and changing v_{200}. Again compare the first by repeating, holding v_{200} fixed and changing c. Feel free to use your previous tools from last time if you find that you need to (though I don't think you should); in the future you'll be able to use built-in functions for those.

The Content of Galaxies

We discussed very briefly the equilibrium of different states of a gas like Hydrogen in the interstellar medium and the concept of *detailed balance*. Let's look at this in more detail.

Let's redefine the *Einstein coefficients* and go into them in a little more detail. For an upper state u and a lower state l, they are:

- A_{ul} is the transition probability per unit time for *spontaneous emission*.
- $B_{lu}\bar{J}$ is the transition probability per unit time for *absorption* of radiation, with $\bar{J}$ being the mean intensity of the radiation field (related to the energy density).
- $B_{ul}\bar{J}$ is the transition probability per unit time for *stimulated emission*, where an electron will jump to a lower energy configuration in the presence of other electromagnetic radiation.

Again, we will assume here that collisional excitation and de-excitation, often given with C_{lu} and C_{ul}, respectively (but not often called Einstein Coefficients), are negligible given the low densities for the purposes of this exercise.

And, again for posterity, a three-level system in equilibrium will have equations that look like:

$$
\begin{aligned}
n_1(B_{12}+B_{13})\bar{J} &= n_2(A_{21}+B_{21}\bar{J})+n_3(A_{31}+B_{31}\bar{J}) \\
n_2(B_{21}\bar{J}+A_{21}+B_{23}\bar{J}) &= n_1B_{12}\bar{J}+n_3(B_{32}\bar{J}+A_{32}) \\
n_3(B_{31}\bar{J}+A_{31}+B_{32}\bar{J}+A_{32}) &= n_1B_{13}\bar{J}+n_2B_{23}\bar{J}
\end{aligned}
$$

And also again, since the third equation is related to the other two, so in order to solve for the three densities, we need one more constraining equation. That one can just be given by the total nuber density: $N = n_1+n_2+n_3$. But if we deal with line ratios, then we don't actually care, which is often the case. So then you only have to worry about what n_2/n_1 is and n_3/n_1 is, for example (but often in observable spectral line units).

The Einstein coefficients are all related to each other. If you have a two-level system in thermodynamic equilibrium, you can work out the *Einstein relations* which *must* hold true for all temperatures and thus are *atomic properties* and not dependent on the environment. These relations are:

$$
\begin{aligned}
g_lB_{lu} &= g_uB_{ul} && (4) \\
A_{ul} &= \frac{2h\nu^3}{c^2}B_{ul} && (5)
\end{aligned}
$$

where h is Planck's constant, c is the speed of light, ν is the frequency of the emission line, and g_n are statistical weights for each level (the degeneracy for each). One can work out that for Hydrogen in state $n = 1,2,\cdots$, $g_n = 2n^2$ (Sorry about another "n").

What we see above is that once we know one of the Einstein coefficients, we know them all, which is convenient. That doesn't mean we know all of the *rates* though because those are dependent on $\bar{J}$. Technically, this is given by

$$\bar{J} \equiv \int_0^\infty J_\nu\phi(\nu)\,d\nu, \tag{6}$$

where J_ν is the spherically averaged mean intensity at a frequency ν (often written this way and not as $J(\nu)$ by convention) and $\phi(\nu)$ is the normalized line profile function that is the sum of many components, including a natural (Lorentzian) broadening, Doppler braodening, collisional broadening, etc. We can make an approximation that it is nearly a delta function in shape, centered at some frequency ν_0, and so we have that $\bar{J} \approx J_{\nu_0}$. In thermodynamic equilibrium though, $J_\nu = B_\nu$, the Planck blackbody function (sorry about another "B"), or

$$B_\nu(T) = \frac{2h\nu^3}{c^2}\frac{1}{e^{h\nu/kT}-1}, \tag{7}$$

where k is Boltzmann's constant. Therefore,

$$\bar{J} \approx \frac{2h\nu_0^3}{c^2}\frac{1}{e^{h\nu_0/kT}-1}. \tag{8}$$

For Hydrogen, the frequencies of the transitions between different states are easy to calculate: $E = -13.6\ \mathrm{eV}/n^2$, again for $n = 1, 2, \cdots$. The frequency of emission or absorption is just going to be the difference between the energy levels, so between two states i and j, the frequency of the transition is given by $\Delta E_{ij} = h\nu_{ij}$, where $\nu_{ij} = \nu_0$ above.

Below is a table of Einstein coefficients A_{ul} from `http://www.ioffe.ru/astro/QC/CMBR/sp_tr.html` (Kholupenko, Ivanchik, and Varshalovich 2005), sanity checked against the more specific CHIANTI database (Dere et al. 1997, Landi et al. 2013; has more specific individual transitions), and sanity checked against some other values found elsewhere. The values are given in units of s^{-1}.

l \ u	2	3	4	5	6	7	8	9
1	4.70×10^8	5.57×10^7	1.28×10^7	4.12×10^6	1.64×10^6	7.57×10^5	3.87×10^5	2.14×10^5
2	$--$	4.41×10^7	8.42×10^6	2.53×10^6	9.73×10^5	4.39×10^5	2.21×10^5	1.22×10^5
3	$--$	$--$	8.98×10^6	2.20×10^6	7.78×10^5	3.36×10^5	1.65×10^5	8.90×10^4
4	$--$	$--$	$--$	2.70×10^6	7.71×10^5	3.04×10^5	1.42×10^5	7.46×10^4
5	$--$	$--$	$--$	$--$	1.02×10^6	3.25×10^5	1.39×10^5	6.90×10^4
6	$--$	$--$	$--$	$--$	$--$	4.56×10^5	1.56×10^5	7.06×10^4
7	$--$	$--$	$--$	$--$	$--$	$--$	2.27×10^5	8.23×10^4
8	$--$	$--$	$--$	$--$	$--$	$--$	$--$	1.23×10^5

These are also provided to you in a more convenient file to read from called `A_coefficients.dat`.

[3.] Object-oriented programming time! Write a matrix library that includes a `Matrix()` class. There's lots of material online on how to write a class, and we'll go over some of this in more depth before you get started. Your class should be able to:

1. add two matrices together (you can overload `__add__()` but don't feel that you have to),
2. multiply two matrices together (again, can use `__\mult__`)
3. transpose a matrix,
4. invert a matrix,
5. calculate the trace of a matrix,
6. calculate the determinant of a matrix,
7. return the LU decomposition of a matrix (should return two `Matrix` objects).

You may find it useful to also define some function that returns a single element i, j, swap rows, etc., whatever else you think might be useful but don't feel that you must. Internally, feel free to use `numpy`'s `np.array` or even `np.matrix` if you so choose. As usual, please document, etc.

[4.] Unit testing time! Please write a unit test class (for example, in a file called `test_matrix.py` if your `Matrix` class in in a file called `matrix.py`) for your Matrix class. I *highly* recommend you use Python's built-in `unittest` framework, e.g., `https://docs.python.org/3/library/unittest.html`, though if you want to use just your own `assert` commands you may do so. Do not overdo it with these, simply compare for each of the above items that you get a sensible result, i.e., you only need to do one test per item above, so don't make 50 tests of whether you have or have not transposed a matrix or not. The goal is to just learn about the practicalities of unit testing and make it easier for you to debug your `Matrix` class. For this part of the assignment, you may absolutely use `numpy` or `scipy`'s built-in functionality to check, though if you want to put in 3×3 matrices since you can do the checks by hand, that is of course fine as well.

[5.] For a total number density of $N = 1$ cm^{-3} for convenience, calculate the different number densities as a function of temperature T. Make a plot showing the different number densities vary as a function of T.