Single-voxel debugging

Craig Yanitski

November 9, 2021

1 Single-voxel setup

Here we will examine the current setup of the calculations of KOSMA-τ3d. These will give a basis on how the probabilistic calculation works for a single voxel, and what assumptions are present when using the single-voxel model. The observed intensity is calculated from the radiative transfer equation,

$$dI_\nu = -I_\nu \kappa_\nu ds + \epsilon_n u ds,$$

we can assume κ and ϵ (the absorption and emissivity coefficients, respectively) are constant in the voxel. Therefore they are calculated using the voxel-averaged emmision:

$$\epsilon_\nu(v_{obs}) = \frac{\langle I_\nu \rangle (v_{obs})}{\Delta s},$$

$$\kappa_\nu(v_{obs}) = \frac{\langle \tau_\nu \rangle (v_{obs})}{\Delta s},$$

In order to properly understand these voxel-averaged values, it is necessary to have some insight for what is happening inside the voxel. The clumpy PDR is approximated by an ensemble of spherically-symetric KOSMA-τ'clumps'. The ensemble of clumps has a Gaussian velocity distribution with dispersion σ_{ens}, and each clump has an intrinsic velocity dispersion σ_{cl} (this value depends on which underlying KOSMA-τgrid is used, but currently it is $\sim 0.71 \frac{km}{s}$). The number of clumps with a particular mass (N_j) follows the mass spectrum:

$$\frac{\mathrm{d}N}{\mathrm{d}M} = AM^{-\alpha}$$

Here we adopt $\alpha = 1.84$ from Heithausen et al. (1998), and A is a constant. Henceforth the subscript j will refer to a property specific for clumps with mass M_j. Clumps can be observed with a particular radial velocity, so we also use the formalism of the subscript i to refer to a property with radial velocity v_i. Note that this is a velocity within the voxel, not the observed velocity. The observed radial velocity is v_{obs}.

Now the number of clumps with a particular radial velocity v_i is determined by the number of clumps of a particular mass N_j and velocity distribution of the ensemble σ_{ens}.

$$\Delta N_{j,i}(v_{obs}) = \frac{N_j}{\sqrt{2\pi}\sigma_{ens}} \exp\left(-\frac{(v_{vox} - v_i)^2}{2\sigma_{ens}}\right) \Delta v$$

$p_{j,i}$ is the probability of having $k_{j,i}$ clumps in a line-of-sight observed at velocity v_i, determined by Binomial distribution of the number of clumps at that velocity:

$$p_{j,i} = \binom{N_{j,i}}{k_{j,i}} p_j^{k_{j,i}} (1 - p_j)^{N_{j,i} - k_{j,i}}$$

Where p_j is the area fraction of clump j. If $p_j N_{j,i} > 5$, then a gaussian expression is used with $\mu_{j,i} = p_j N_{j,i}$ and $\sigma_{j,i} = \sqrt{N_{j,i} p_j (1 - p_j)}$.

For having $k_{j,i}$ clumps in a line-of-sight, the intensity and optical depth ($I_{\nu,j,i}$ & $\tau_{\nu,j,i}$) can be calculated. Here the subscript ν means the properties refer to a particular transition (which is what we get from the KOSMA-τgrid).

$$I_{\nu,j,i}(v_{obs}) = k_{j,i} I_{cl,j} \text{exp}\left(\frac{(v_i - v_{obs})^2}{2\sigma_{cl}^2}\right)$$

$$\tau_{\nu,j,i}(v_{obs}) = k_{j,i} \tau_{cl,j} \text{exp}\left(\frac{(v_i - v_{obs})^2}{2\sigma_{cl}^2}\right)$$

Finally the intensity and optical depth can be averaged over the voxel using all possible combinations and their corresponding probabilities to determine voxel properties we require.

$$\langle I_\nu \rangle (v_{obs}) = \sum_i \left(\prod_j p_{j,i}(k_{j,i}) \right) \left(\sum_j k_{j,i} I_{\nu,j,i}(v_{obs}) \right)$$

$$\langle \tau_\nu \rangle (v_{obs}) = -\ln \left[\sum_i \left(\prod_j p_{j,i}(k_{j,i}) \right) \ \exp \left(- \sum_j k_{j,i} \tau_{\nu,j,i}(v_{obs}) \right) \right]$$

Here the sums and products ensure we consider contributions from all clump masses at all radial velocities, since there is a velocity dispersion to each clump's intensity and optical depth. Now we can finally substitute these into the radiative transfer equation to determine the intensity we expect to observe from this voxel.

$$I_{\text{vox},\nu}(v_{obs}) = \frac{\epsilon_\nu(v_{obs})}{\kappa_\nu(v_{obs})} \left(1 - e^{-\kappa_\nu(v_{obs})\Delta s}\right) + I_{bg} e^{-\kappa_\nu(v_{obs})\Delta s}$$

2 Calculation error

There is clearly some issue with the single voxel calculation, since there is an absorption feature that should not be present. Perhaps this was noticed initially when setting up the single-voxel model, but at the time it was dismissed due to the nonphysical model parameters (a volume filling factor $f_V > 1$). Yoko has explored more of the parameter space to compare with observations, though, and encountered this problem again. In this use-case, an over-filled voxel is used to compare to various lines-of-sight. This makes enough sense as the clumps overflow behind the voxel. I now need to revise my intensity calculations not only to correct this absorption feature, but to ensure that the previous calculations (without this error) also agree.

For a densely-filled voxel, there should exist a plateau in the $I(v_{obs})$ curve (see Figure **??**), however the error is that there still exists some additional absorption. Due to the calculation of the voxel intensity (see §1), there are two separate modes of calculation: standard and normalised. The standard calculation of the intensity uses the clump numbers calculated using $\Delta N_{j,i}$, while the normalised calculation resizes the voxel to ensure a given number of the largest clump are included in the voxel (typically 1 for the dense clumpy ISM). There is also part of the code which resizes the voxel if it is too small to fit the largest clump. It is the combination of these two features that make debugging this error more difficult. For that reason, I will debug these calculation methods separately.

2.1 The correction

The issue identified thus-far is that the probabilistic description has been applied to the averaged *intensity* rather than the *emissivity*, and therefore there exists some excess absorption in the plateau region. The proposed fix for this is to modify the equation for $I_{\nu,j,i}(v_{obs})$ to calculate instead $\epsilon_{\nu,j,i}(v_{obs})$:

$$\epsilon_{\nu,j,i} = I_{\nu,j,i} \frac{\tau_{\nu,j,i}}{L_j} / [1 - \exp(-\tau_{\nu,j,i})],$$

leading to,

$$\langle \epsilon_\nu \rangle (v_{obs}) = \sum_i \left(\prod_j p_{j,i}(k_{j,i}) \right) \left(\sum_j k_{j,i} \epsilon_{\nu,j,i}(v_{obs}) \right).$$

The modification for the optical depth in terms of opacity is,

$$\langle \kappa_\nu \rangle (v_{obs}) = -\ln \left[\sum_i \left(\prod_j p_{j,i}(k_{j,i}) \right) \exp \left(-\sum_j \frac{k_{j,i} \tau_{\nu,j,i}(v_{obs}) \times A_j}{V_{voxel}} \right) \right]$$

At least this is how the intended modification to the code should work. Its implementation tries to remain as faithful to this description as possible, but it was noted that there are still a couple errors. Therefore some flags were added to choose the type of implementation: `test_calc`, `test_pexp`, `test_opacity`, and `test_fv`. `test_calc` chooses whether or not the test calculation is used, and it will definitely affect the emissivity. `test_opacity` is a flag to use either the original definition of opacity or to use the modified version given above. `test_pexp` is a test to see how the calculation of opacity changes if one considers how the result is changed if one applies the probabilistic approach to κ rather than $e^{-\kappa}$. This should not be working since the probabilistic approach should work on the attenuation ($e^{-\tau}$). In any case, this is included for a test. The final flag, `test_fv`, tests a form of the calculation that should have been in the code from the beginning. Since only partially- to fully-filled voxels are considered in the 3D models, all of the computations depend on the extent of the voxel, Δs. This is not true in the case of an over-filled voxel, where the excess clumps are assumed to be located just behind the voxel. In this case of an over-filled voxel, the calculations are modified to ensure it is still physically correct. As a crude approximation, this changes equations such as the voxel-averaged opacity to $<\kappa_\nu>=\frac{<\tau_\nu>}{f_V \Delta s}$, where $f_V > 1$. This ensures all of the clumps are contained in the voxel being calculated. In the case of an over-filled voxel, the voxel calculated is actually a rectangle.

For the following tests in this section, we will use the same parameters used to identify the issue. This used four clump masses $M_{cl} \in [10^{-3}, 10^{-2}, 10^{-1}, 10^{0}] M_{\odot}$, voxel size $\Delta s = 0.1 pc$, hydrogen number density $n_H = 10^{4.8} cm^{-3}$, total mass $M_{vox} = 10^{0.8} M_{\odot}$, and far-UV radiation $\chi = 10^{1.7} \chi_0$.

2.2 Standard calculation

Setting all flags to false will perform the original calculation, which for a partially-filled voxel acts as a benchmark. The issue noted by Yoko, however, was for a voxel with $f_V \gtrsim 4$. This case is thus included to show how the error was identified. The absorption feature in CII is the error that we will be tracking.

voxel_error_original.png

2.3 Using the correct f_V

In this model we set `test_fv` True while keeping the other flags False. This should be the correct calculation of an over-filled voxel using the old calculation.

voxel_error_fv.png

2.4 Setting test_fv and test_pexp

Here I test the validity of the way the probability approach is applied to the opacity. This test uses the probability on the τ_ν rather than $e^{-\tau_\nu}$. It can be seen that this test correctly removes the absorption features, however it is at the expense of a slightly-reduced intensity. The slightly-reduced intensity is understandable considering the result without setting `test_fv` was not physically correct.

voxel_error_fv-pexp.png

2.5 Setting test_fv, test_calc, and test_opacity

Here we test the new calculation applied to both ϵ_ν and κ_ν.

voxel_error_fv-calc-opacity.png

2.6 Setting test_fv and test_calc

Here we test the new calculation applied just to ϵ_ν, but $< \kappa_\nu >$ is still calculated from $< \tau_\nu >$ like in the old calculation.

voxel_error_fv-calc-opacity.png

2.7 Setting `test_fv`, `test_calc`, `test_opacity`, and `test_pexp`

Finally we calculate a model setting all of the flags. This also correctly results in the desired plateaux, but the relative intensities are different (since C II is now larger than O I).

voxel_error_fv-calc-opacity-pexp.png