Problem Set 9

1 A Compton Monte Carlo

An intuitive and commonly applied method for modeling inverse Compton emission is to use a Monte Carlo approach, in which the radiation field is represented by discrete, individual packets of photons. The fate of each photon packet is determined by "rolling the dice", i.e., choosing random numbers drawn from the appropriate probability distributions. Here you will write your own simple Monte Carlo code, which could in principle be used to model real data. This is a numerical problem, so please provide a printout of your code, in addition to your plots.

   Consider a supernova that produces optical photons, which then travel through a shell of shocked circumstellar gas, where they may be scattered by non-thermal, relativistic electrons. Assume that the supernova emits a luminosity of $L_s = 10^{43}$ erg/s. The luminosity carried by each Monte Carlo packet in our code is then $L_p = L_s/N_p$, where N_p is the number of photon packets we choose to use. We will assume for simplicity that all photons are emitted with the same frequency $h\nu_{in} = E_{in} = 1$ eV. These photons travel through a shocked shell of relativistic electrons with optical depth $\tau = 0.01$.

   The Monte Carlo method proceeds by generating and scattering photon packets one by one. The energy of a scattered photon is given by applying two Doppler shifts — one into the rest frame, and one back out — which gives a final outgoing energy

$$E_{out} = E_{in}\gamma^2(1 - \beta\cos\theta_{in})(1 + \beta\cos\theta'_{out}) \tag{1}$$

where θ_{in} is the incoming angle (in the lab frame) and θ'_{out} is the outgoing angle (in the rest frame). We simulate a scattering by sampling the incoming and outgoing directions randomly. (Since we are considering $\tau \ll 1$, we will only need to scatter each packet once). Assume that $\cos\theta_{in}$ and $\cos\theta'_{out}$ are drawn uniformly (and randomly) from the range -1 to 1. This matches the distribution of $\cos\theta$ over an isotropic sphere. After a random scattering, the packet has a new luminosity

$$L_{out} = L_p\frac{E_{out}}{E_{in}}(1 - e^{-\tau}) \tag{2}$$

where the term in parenthesis takes into account the fact that only a fraction $(1 - e^{-\tau})$ of the photons from the source are actually scattered. The packet can be "observed" by collecting it and binning its energy into a spectrum array. If we repeat this procedure with the N_p packets, we will gradually build up the observed spectrum. Because of the random nature of the algorithm, this spectrum will possess noise, and we will need a fairly large value for N_p to achieve a reasonable signal to noise (try 10^5 for testing, 10^6 for the final result).

1.1 Consider the case where we only have electrons of a single energy, $\gamma = 10$. Run the Monte Carlo procedure and plot the spectrum of scattered photons on a log-log plot. Your x-axis can be either eV or Hz, and y-axis should be a monochromatic luminosity, i.e., units of ergs s^{-1} eV^{-1} or ergs s^{-1} Hz^{-1}. Check that where the spectrum cuts off at high energy makes sense.

1.2 Now consider the more realistic case where the electrons have a power law distribution in energy, between the bounds $\gamma_{min} = 10$ and $\gamma_{max} = 100$. The probability, in any given scattering, that the electron has an energy γ is

$$P = A\gamma^{-p} \tag{3}$$

for $\gamma_{min} < \gamma < \gamma_{max}$, and 0 otherwise. A is a constant, and we'll take $p = 2.5$. For each scattering, we should then choose the γ of the electron by randomly sampling from this probability distribution

(which can be easily done[1]). Run the Monte Carlo for this case and plot the spectrum due to inverse Compton scattering. Overplot the slope of the power law that you expect from analytic arguments.

By the way, your Monte Carlo code could probably already be used in a simple research paper. For example, you could model the x-ray emission from SN 2011fe and constrain the density of the circumstellar environment, as considered in Horesh et al. (2012). To improve the code, you would want to relax the assumption that all photons start with the same energy E_{in}. Instead, you can imagine randomly sampling the initial frequency of the photons from a real distribution (e.g., a blackbody). In addition, you could fairly easily generalize your code to treat multiple scatterings, or for the more realistic case where the scattering is not isotropic or inelastic. If you also kept track of the position of the photons, you could even consider the case where the scattering cloud is not spherically symmetric.

---

[1] http://mathworld.wolfram.com/RandomNumber.html