

Radiative transfer and its implementation in Python

Nick Higginbottom

31st January 2012

Chapter 1

Radiative Transfer

1.1 Introduction

The aim of this document is to provide a primer to the principles of radiative transfer and how the equations are implemented in the python code for the simulation of objects with winds. This was written at the StSci in the fall of 2010

1.2 Basic definitions of terms used in radiative transfer

Symbol	Quantity	Unit	Description
L	Luminosity	$ergs^{-1}$	this is the total radiated power of an object, integrated over all angles and all frequencies
I	Intensity	$erg\,cm^{-2}s^{-1}sr^{-1}$	The flux in a given direction across a surface normal to that direction
I_ν	Specific Intensity or brightness	$erg\,cm^{-2}s^{-1}sr^{-1}Hz^{-1}$	Flux per unit frequency, this is only measurable if an object is resolved.
B_ν	Specific intensity of a Blackbody	$erg\,cm^{-2}s^{-1}sr^{-1}Hz^{-1}$	BB flux per unit frequency
J_ν	Mean Intensity	$ergs^{-1}\,cm^{-2}Hz^{-1}$	Directional average of the specific intensity, Flux per unit frequency averaged over all solid angles
\bar{J}	Mean Intensity	$ergs^{-1}\,cm^{-2}$	Flux per unit frequency averaged over all solid angles and averaged over a range of frequencies (e.g. the frequencies over which a particular line absorbs).
F	Surface flux	$erg\,cm^{-2}s^{-1}$	Intensity integrated over all solid angles, this is usually all we can measure of an object.
F	Emittance	$erg\,cm^{-2}s^{-1}$	Very similar to surface flux but integrated only over the forward direction. π times the intensity
F_ν	Specific Emittance	$erg\,cm^{-2}s^{-1}Hz^{-1}$	
u	Radiation density	$erg\,cm^{-3}$	
u_ν	Specific radiation density	$erg\,cm^{-3}Hz^{-1}$	

Much of the following is a rewriting of the descriptions to be found in Rybicki and Lightman (1979), Osterbrock (1989) and Sobolev (1969). Radiation at its most fundamental level is simply the transfer of energy from one place to another. If we consider a source which radiates in all directions, then we start by defining the concept of an energy flux F ; simply how much energy is passing through a unit area in a unit time. F is measured in $\text{ergs s}^{-1}\text{cm}^{-2}$, and if we want to know how much energy has passed through a small surface element dA in time dt , then this energy is simply equal to $FdAdt \text{ ergs}$. This supposes that the surface element is normal to the radiation field.

If we take the case of an object which radiates energy equally in all directions, then this flux will be spread out over spherical surfaces of ever increasing surface area as the observer moves away from the source. The energy per second is constant, (it is equivalent to the Luminosity L of the source) so we can write

$$F = \frac{L}{4\pi r^2}$$

This very simple minded description of radiation runs into serious problems as soon as one wants to know how much energy is radiated in a particular frequency band, and we have to introduce a more complex treatment in order to allow proper consideration of this. We begin this by considering how much energy is transported by a group of rays travelling in roughly the same direction. We will consider a small surface dA normal to the rays and ask how much energy passes through it in coming from a certain direction. This small surface could be a detector on Earth or some theoretical surface in space. In this case, direction means angle, and so a small group of rays coming from roughly the same direction describes a small element of solid angle $d\Omega$ as seen by our surface. The energy radiated at a single frequency is zero, so we also have to take a small range of frequencies $d\nu$. The energy passing through the surface, from a small range of solid angles, at a small range of frequencies is defined as I_ν , the specific intensity or brightness of the radiation field at the surface dA . Therefore, if we go back to asking how much energy passes through the surface, we can write

$$dE = I_\nu dA dt d\Omega d\nu$$

More generally, the surface could be inclined at an angle θ to the incoming rays. In this

case, the area of the surface will be reduced by a factor $\cos\theta$. So more generally

$$dE = I_\nu \cos\theta dA dt d\Omega d\nu$$

The relation between flux F and specific intensity I_ν is therefore simply

$$F = \int I_\nu \cos\theta d\Omega d\nu$$

One can also define a specific flux, F_ν which is simply the flux per unit frequency interval.

$$F_\nu = \int I_\nu \cos\theta d\Omega$$

This quantity I_ν is particularly useful because it is independent distance from the source. This is because as you get further away, the energy is spread out over a larger surface, which goes as $1/r^2$, however the object subtends a smaller solid angle, which cancels it out. This means that the specific intensity (or brightness) is a property of the source which is the same at the surface of the object or at the place where it is being observed. This is only true if there is nothing between the source and the observer. In general, this is not true, and so we need to consider how radiation is transported through a medium. This brings us to the equations of radiative transfer.

1.2.1 Radiative energy density

Since radiation carries energy, and area of space through which radiation is passing will have an energy density associated with it. This quantity can be used to compute the probabilities for electron transitions to take place in a material through which the radiation is passing and so it is an important quantity for radiative transfer problems. We begin the definition by defining a quantity $u_\nu(\Omega)$ which is the energy per unit volume per unit solid angle per frequency interval. This is related to the energy in a given volume of space by

$$dE = u_\nu(\Omega) dV d\Omega d\nu$$

We define a cylinder of length cdt , the volume is therefore $dA cdt$ and so the energy contained within it is

$$dE = u_\nu(\Omega) dA cdt d\Omega d\nu$$

This energy is all contained in a ray, which moves with velocity c , and so in the time dt , the ray will sweep out this volume. The energy per unit time, per unit area, per unit solid angle per unit frequency is simply the specific intensity I_ν so we can write an alternative expression for the energy within the cylinder.

$$dE = I_\nu dA dt d\Omega d\nu$$

Equating the two, we get an expression relating energy density per unit solid angle with specific intensity

$$u_\nu(\Omega) = \frac{I_\nu}{c}$$

In order to go from energy density per unit solid angle to a straight energy density, we must clearly integrate over solid angle

$$u_\nu = \frac{1}{c} \int I_\nu d\Omega$$

We define a new quantity here, J_ν which is the mean intensity, taken over all directions from the current location where we want to work out the energy density. This is a useful quantity in its own right, since it represents the intensity in a region coming from all angles.

$$J_\nu = \frac{1}{4\pi} \int I_\nu d\Omega \tag{1.1}$$

There are 4π steradians in a sphere, so the equation becomes

$$u_\nu = \frac{4\pi}{c} J_\nu$$

At this stage u_ν is the energy per unit volume, per unit frequency, so in order to produce the required radiation energy density u , we must simply integrate over all frequencies

$$u = \frac{4\pi}{c} \int J_\nu d\nu$$

In words, we are saying that the energy per unit volume due to radiation entering the volume from outside, or the *radiation density* is $4\pi/c$ times the mean intensity of radiation entering that volume from all directions.

1.3 The equations of radiative transfer

1.3.1 Introduction

In the absence of any absorbing medium, the specific intensity of a radiation field remains constant, however in real situations, the intensity of a beam from a source will be affected in four ways (we introduce the concept of photons here to aid the discussion)

- Photons may be absorbed by the medium through which the ray is passing
- Photons may be emitted by the medium
- Photons may be scattered out of the direction of the ray
- Photons may be scattered into the direction of the ray from other rays

The first two processes may be summarised by the following simple equation

$$\frac{dI_\nu}{ds} = -\kappa_\nu I_\nu + \epsilon_\nu \quad (1.2)$$

The equation may also be written without the subscripts and represents the frequency integrated quantities, however in general both the emissivity and absorption are frequency dependent. The second two processes are scattering type processes and will be dealt with separately. In equation (1.2) the first term refers to absorption by the medium, and the second to emission from the medium.

1.3.2 Absorption basics

The quantity κ_ν is the absorption coefficient measured in cm^{-1} and represents the loss in intensity in a beam as it travels a distance ds through the medium. It can also be thought of as the probability of a photon being absorbed per distance that the beam travels through the medium. We can therefore write

$$dI_\nu = -\kappa_\nu I_\nu ds$$

We can now integrate this equation along the path that the ray takes from the inner edge of the medium s_{in} to the outer edge of the medium s_{out}

$$\int_{s_{in}}^{s_{out}} \frac{dI_\nu}{I_\nu} = - \int_{s_{in}}^{s_{out}} \kappa_\nu(s') ds'$$

this gives

$$\ln I_{\nu_{out}} - \ln I_{\nu_{in}} = - \int_{s_{in}}^{s_{out}} \kappa_{\nu}(s') ds'$$

where $I_{\nu_{in}}$ is the intensity of the radiation as it enters the cloud and $I_{\nu_{out}}$ is the intensity as it leaves. Rearranging

$$\ln \left(\frac{I_{\nu_{out}}}{I_{\nu_{in}}} \right) = - \int_{s_{in}}^{s_{out}} \kappa_{\nu}(s') ds'$$

and removing the ln

$$\left(\frac{I_{\nu_{out}}}{I_{\nu_{in}}} \right) = \exp \left[- \int_{s_{in}}^{s_{out}} \kappa_{\nu}(s') ds' \right]$$

and so we can write an equation for the intensity of the beam leaving the medium in terms of the input intensity

$$I_{\nu_{out}} = I_{\nu_{in}} \times \exp \left[- \int_{s_{in}}^{s_{out}} \kappa_{\nu}(s') ds' \right]$$

We define the integral of the absorption coefficient over the path of the ray to be equal to the optical depth τ_{ν}

$$\tau_{\nu} = \int_{s_{in}}^{s_{out}} \kappa_{\nu}(s') ds' \quad (1.3)$$

and so we can write

$$I_{\nu_{out}} = I_{\nu_{in}} \times e^{-\tau_{\nu}}$$

If $\tau_{\nu} \ll 1$ the cloud is said to be optically thin and if $\tau_{\nu} \gg 1$ the cloud is optically thick. Since in general τ is dependant on frequency, it is possible for the cloud to be opaque to some frequencies and transparent to others. If there is a specific spectral line, it is possible get extremely high optical depths.

1.3.3 Emission basics

Similarly to the way we dealt with absorption, we define an emission coefficient for the medium, ϵ_{ν} which is the energy emitted by the medium per unit time, per unit solid angle, per unit volume in a frequency interval $d\nu$ so its units are $ergs s^{-1} \Omega^{-1} m^{-3} Hz^{-1}$. This takes account of the fact that in most cases an emitting medium will produce different amounts of energy in different directions, and at different frequencies. In

the special case of an isotropic scatterer (like a blackbody) we can write the following equation

$$\epsilon_\nu = \frac{1}{4\pi} P_\nu \quad (1.4)$$

where P_ν is the radiated power of the medium per unit volume per unit frequency interval.

If we are considering a beam of radiation that moves through a region of space where emission is taking place, the intensity of the beam will increase. The beam has a cross-section dA as defined earlier, so the volume of emitting medium swept out by the beam is $dA ds$ where ds is the path length through the medium, exactly as defined in the preceding discussion of emission. We can therefore write an expression for the change in intensity of the beam

$$dI_\nu = \epsilon_\nu ds \quad (1.5)$$

1.3.4 An alternative form of the radiative transfer equation

We can use the concept of optical depth to rewrite the equation of radiative transfer in a different way. From (1.3) we can write

$$d\tau_\nu = \kappa_\nu ds$$

By multiplying both sides of (1.2) by ds we get

$$dI_\nu = -\kappa_\nu I_\nu ds + \epsilon_\nu ds$$

and substituting for $d\tau_\nu$

$$dI_\nu = -I_\nu d\tau_\nu + \epsilon_\nu ds$$

dividing through by $d\tau_\nu$ we get

$$\frac{dI_\nu}{d\tau_\nu} = -I_\nu + \frac{\epsilon_\nu ds}{d\tau_\nu}$$

but, since $\kappa_\nu = d\tau_\nu/ds$ we can write

$$\frac{dI_\nu}{d\tau_\nu} = -I_\nu + \frac{\epsilon_\nu}{\kappa_\nu}$$

Clearly the last term in this equation has the units of specific intensity, and by examining the equation one can see that it tells us by how much the input radiation is modified by travel through the medium per unit optical depth. We call this term the *source function* of the medium, defined as follows

$$S_\nu = \frac{\epsilon_\nu}{\kappa_\nu} \quad (1.6)$$

We can split up this source function back into two parts. A an absorption fraction q which is the proportion of incoming photons which are absorbed, and the scattering fraction $(1 - q)$ which represents the rest of the photons which are scattered by the medium. The source function can be written

$$S_\nu = (1 - q)J_\nu + qB_\nu(T_e)$$

Where the first term is the scattered photons which depends on the mean intensity of the radiation incident upon the bit of cloud we are interested in and the second term represents the energy absorbed and collisionally de-excited being re-radiated to form a dilute black body at the temperature of the electrons doing the de-excitation.

1.4 Radiative transfer details

1.4.1 Introduction

In the last section we saw the simple radiative transfer equation, which is simply a statement of the fact that it is possible for a beam of radiation to both gain and lose energy as it passes through a medium. The detail of solving this equation is in considering the physical processes by which this happens. Thinking of photons as packets of energy, this energy can be deposited into the medium, heating it, removed from the medium, cooling it, or more often both, leaving the medium at a steady temperature but redistributing the photon energy. We will consider first the absorption processes and then the emission processes.

1.4.2 Absorption processes

There are three main absorption processes which much be considered, and these will be discussed in detail in the next three sections.

- free-free absorption (or inverse Bremsstrahlung.
- photoionisation or bound-free
- bound-bound or resonance-line scattering

In brief, free free absorption is allied to Bremsstrahlung radiation. It is the interaction between free electrons and photons. The electrons start off free, and end up free (hence free free) and since free electrons can have any energy, the spectrum thus produced is continuum.

Photoionisation is the process whereby an ion absorbs a photon, and an electron is ejected from the ion to become a free electron. Since the electron thus produced can have any energy, this is a continuum process. This is also called free-bound or bound-free depending on the sense of the transition.

Resonance-line scattering is the process of a photon being absorbed by an electron and that electron being promoted to a higher bound energy level within the ion. Theoretically this electron can then fall back to its original level via a series of transitions emitting several photons of different frequencies. In Python, it is assumed that only one transition takes place, back to the original level. Since the electron is at all times bound to the ion, this is also called bound-bound absorption.

Free-free

A photon cannot be absorbed by a free electron, since it is impossible to conserve momentum, however it is possible for an electron close to an ion to do so. This absorption process is called free-free absorption since the electron is free before and after the event. It is also called inverse Bremsstrahlung. The equation used to compute the free-free crosssection presented by a cell to an incoming photon taking into account stimulated

emission is

$$\kappa_{ff} = 3.692 \times 10^8 [1 - \exp(-h\nu/kT)] Z^2 g T^{-1/2} \nu^{-3} N_e N_i$$

from Allen's astrophysical quantities. Where g is the Gaunt factor, N_e and N_i are the electron and ion densities in cm^{-3} . The mean velocity for the electrons is taken account of in the numerical constant and the $T^{-1/2}$ term.

Free-bound or photoionisation

Here are some important quantities:

- χ - The energy threshold for an ionisation or excitation process = $h\nu_0$ also clearly the energy difference between two states.
- g_1 The statistical weight of an energy level - basically the number of states with the same energy - also called the degeneracy of the state.

Photoionisation is the process where photons above a threshold frequency ν_0 cause electrons to be freed from an atom, thereby increasing its ionisation state and increasing the number density of electrons. Because there are an infinite number of different energy states of free electrons photons of frequency ν_0 and above are absorbed, although it turns out that the likelihood of absorption peaks at ν_0 and then drops off as the frequency increases. This process is how the wind is heated, and python calculates the photoionisation of the flow during the ionisation cycles to calculate the temperature of the flow and the ionisation state of the gas. This in turn is important so one is able to calculate the correct probabilities for resonance line scattering.

Following Osterbrock (1989) we start by considering a small region in a hydrogen cloud and ask what the rate of ionisation is. Clearly it will depend on the number of photons available and this is in turn proportional to the local mean specific intensity J_ν . This is the energy flowing through the region per unit time, per unit area, per unit frequency, per unit solid angle averaged over all solid angles. Radiation coming from all directions will contribute to the energy in the zone, so we integrate over all solid angles, and therefore the energy available is $4\pi J_\nu$. Photoionisation is a quantum process, so we need to know how many photons are available, and we simply divide the energy by the

energy carried by one photon $h\nu$ to get the following expression for number of incident photons per unit time, per unit area per unit frequency.

$$N_{photons} = \frac{4\pi J_\nu}{h\nu}$$

We need an area at this point, and we use the ionisation cross section of hydrogen $a_\nu(H^0)$. This is effectively a probability that a photon of a given frequency will interact with the hydrogen atom, and has units of area. It depends on frequency and is approximately proportional to ν^{-3} . Thus The number of photons per unit area per unit time per unit frequency multiplied by the interaction cross section tells us how many photons per unit time per unit frequency will interact with the hydrogen atom. If the photons have frequencies greater than ν_0 , the threshold frequency for ionisation, the hydrogen atom will become ionised. So we integrate over all frequencies over the threshold frequency to give us the number of photons per second which ionise the hydrogen atoms. By multiplying further by the number density of hydrogen N_{H^0} , we get the number of ionisations per unit volume per unit time.

$$Ionisation\ rate = N_{H^0} \int_{\nu_0}^{\infty} \frac{4\pi J_\nu}{h\nu} a_\nu(H^0) d\nu$$

If the cloud is in equilibrium, each ionisation must be followed by a recombination - this is *detailed balance* and the recombination rate is simply the number density of ions multiplied by the number density of electrons multiplied by the recombination coefficient - a number which describes the probability of a recombination taking place. This coefficient depends on the local temperature. Equating this to our ionisation rate we get

$$N_{H^0} \int_{\nu_0}^{\infty} \frac{4\pi J_\nu}{h\nu} a_\nu(H^0) d\nu = N_e N_p \alpha(H_0, T)$$

This simple equation is for a pure hydrogen cloud, but in reality we need to model more realistic situations where there are many other elements present.

Bound-bound or resonance-line scattering

In general, resonance line scattering in a material in LTE would not tend to remove energy from a radiation field since any photon absorbed would be quickly re-emitted.

however in the special case of a cloud which is moving with respect to the source, the radiation will be absorbed at one (red shifted) frequency and re-emitted at the comoving line frequency. This will therefore cause blueshifted absorption lines and redshifted emission lines (relative to the observer).

The absorption and consequent reemission of photons is a probabilistic process governed by the populations of the various energy levels in the ions present, and the chance of photons being absorbed or re-emitted. Rybicki and Lightman (1979) begin their discussion of the process of resonance scattering by considering Kirchhoff's Law $\epsilon_\nu = \kappa_\nu B_\nu$. This is a macroscopic statement of energy conservation, but on the microscopic level it implies a link between absorption and emission. Einstein analysed this problem and suggested three processes that need to be considered, each with its own probability of occurring.

- Absorption: A photon of the correct frequency $E = h\nu$ is absorbed by the ion and an electronic transition requiring that energy takes place. There is a certain probability assigned to the process (the Einstein B coefficient), and the chance of it happening is simply the mean intensity of radiation at the frequency (or more properly in a band centred on that frequency) multiplied by the B coefficient.
- Spontaneous Emission: An electron in an excited state of an ion drops back to a lower energy level releasing a photon of frequency $\nu = E/h$. There is a chance of this happening, which is simply given by the Einstein A coefficient.
- Stimulated Emission: This process is where the presence of photons of the correct frequency in the same place as excited ions causes a de-excitation which produces more photons of that frequency. This is the process used in a laser, and like absorption, depends on a different Einstein B coefficient and also the mean intensity at the frequency in question

The first new concept that needs explanation is the mean intensity. We begin by recalling the definition of mean intensity J_ν .

$$J_\nu = \frac{1}{4\pi} \int I_\nu d\Omega$$

This quantity is the mean specific intensity passing through a volume from all directions. The new quantity \bar{J} is the mean intensity integrated over the line width and takes account of the fact that the line is not infinitely narrow but has a width in frequency over which photons can be accepted. Figure 1.1 shows the usual situation, in that

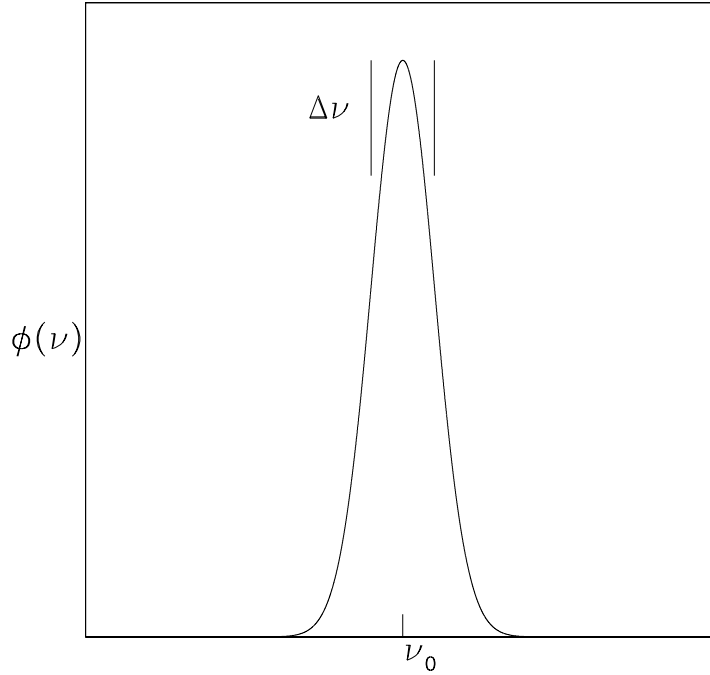


Figure 1.1: Line profile after Rybicki and Lightman (1979)

there is some range of frequencies $\Delta\nu$ which will cause either an absorption process or a stimulated emission. $\phi(\nu)$ is the line profile function which is normalised so the area under the graph is 1. Using this idea we can write an expression for \bar{J} as follows

$$\bar{J} \equiv \int_0^\infty J_\nu \phi(\nu) d\nu \quad (1.7)$$

So \bar{J} has the units of intensity and describes how much energy is available at suitable frequencies coming from all directions to cause the particular absorption or stimulated emission process to occur. When this number is multiplied by the relevant Einstein coefficient, we get the probability per unit time of the particular process to occur. So,

for the three processes, we have three equations

$$\text{transition probability per unit time for absorption} = B_{12}\bar{J}$$

$$\text{transition probability per unit time for spontaneous emission} = A_{21}$$

$$\text{transition probability per unit time for stimulated emission} = B_{21}\bar{J}$$

In the case where the function J_ν varies slowly with frequency (or at least varies little over the interval $d\nu$) as is the case with a blackbody spectrum, we can take it as a constant in the integral. The integral of $\phi(\nu)$ from 0 to ∞ is by definition 1 so $\bar{J} = J_{\nu_0}$. It is also possible to express the equations in terms of energy density u_ν in which case two equations that depend on the energy density become

$$\text{transition probability per unit time for absorption} = \frac{c}{4\pi}B_{12}u_\nu$$

$$\text{transition probability per unit time for stimulated emission} = \frac{c}{4\pi}B_{21}u_\nu$$

Our interest in this process is to discover how much energy is removed from a beam moving through the medium by absorption and how much is added by emission. The python code looks at this in terms of optical depth, so it is useful to consider how we can derive an optical depth from these equations. The equation we need to consider for absorption is the first one in our list. $B_{12}\bar{J}$ tells us the transition probability per unit time, and each transition will remove $h\nu_0$ energy from the beam. We now need to consider how many suitable ions the beam will encounter as it passes through the medium. We define n_1 to be the number density of ions in the lower state. There are therefore n_1dV ions in a volume dV traversed by the beam. So, we can write the following equation for the energy removed from the beam in time dt

$$dE = dV dth\nu_0 n_1 B_{12}\bar{J}$$

However, in order to express this in terms of optical depth, we need to go back to considering specific intensity I_ν . Recalling how specific intensity is related to mean intensity from equation 1.1 and how \bar{J} is defined in equation 1.7 we can get this expression in

terms of the input specific intensity

$$dE = dV dth\nu_0 n_1 B_{12} \frac{1}{4\pi} \int d\Omega \int d\nu \phi(\nu) I_\nu$$

We are interested in how the specific intensity varies, so we don't want to integrate over frequency or solid angle, so we can write the following equation for the loss of energy per unit volume, per unit solid angle, per unit frequency, per unit time

$$dE = dV dtd\nu d\Omega n_1 B_{12} \frac{h\nu_0}{4\pi} \phi(\nu) I_\nu$$

if we define our volume as the volume swept out by a bundle of rays of cross sectional area dA travelling a distance ds in a time dt ($dV=dA ds$) we can make use of the basic definition of energy flow

$$dE = I_\nu dA dtd\Omega d\nu$$

by dividing both sides by $dA dtd\Omega d\nu$ to make the LHS equal to the loss of specific intensity,

$$dI_\nu = n_1 B_{12} \frac{h\nu_0}{4\pi} \phi(\nu) I_\nu ds$$

compare this with our equation for absorption

$$dI_\nu = -\kappa_\nu I_\nu ds$$

and we can see that the absorption coefficient due to line absorption is simply

$$\kappa_\nu = \frac{h\nu_0}{4\pi} n_1 B_{12} \phi(\nu)$$

The same radiation will also cause stimulated emission, which will tend to reinforce the beam. The treatment is exactly the same and it turns out that we can simply modify the last equation to insert a negative absorption term to take account of stimulated emission. The full equation is

$$\kappa_\nu = \frac{h\nu_0}{4\pi} \phi(\nu) (n_1 B_{12} - n_2 B_{21})$$

We can therefore write down the optical depth of this transition line

1.4.3 Emission processes

The photons produced in a python model come from three sources

- blackbody (thermal emission both from the central star, disk and wind)
- solar atmosphere models (used for the central star and disk)
- recombination or free-bound (within the wind)
- Line emission - excited ions de-excite emitting photons of a particular frequency

Blackbody radiation

Since much of the following relies upon terms which relate to the blackbody radiation, it makes sense to carefully describe what we mean by this emission process and what useful parameters are defined by it.

By definition, blackbody radiation is the radiation produced by matter in thermal equilibrium. The form of the specific intensity spectrum produced by a black body depends only on its temperature, and in this case, the specific intensity of a black body I_ν is given the special symbol B_ν and takes the form of the planck law

$$B_\nu(T) = \frac{2h\nu^3}{c^2} \frac{1}{\exp\left(\frac{h\nu}{kT}\right) - 1} \quad (1.8)$$

which can also be written in terms of wavelength

$$B_\lambda(T) = \frac{2hc^2}{\lambda^5} \frac{1}{\exp\left(\frac{hc}{\lambda kT}\right) - 1} \quad (1.9)$$

Another important relationship for radiative transfer comes from a thought experiment carried out where a sample of material is placed at the entrance to a black body cavity. If the temperature of the material is equal to that of the blackbody cavity, then the whole system will produce exactly the same blackbody spectrum, since this depends only on the temperature. If the source function of the material is S_ν we can write

$$S_\nu = B_\nu(T)$$

and since $S_\nu = \epsilon_\nu / \kappa_\nu$

$$\epsilon_\nu = \kappa_\nu B_\nu(T) \quad (1.10)$$

In other words, the emissivity of a medium in thermal equilibrium (energy emitted by the medium per unit time, per unit solid angle, per unit volume in a frequency interval) is equal to the blackbody specific intensity multiplied by its absorption coefficient . This is effectively a statement of energy conservation and is known as Kirchoff's Law. It allows us to calculate the emissivity from the absorption coefficient or vice versa for a medium in local thermodynamic equilibrium (LTE). LTE refers to the situation where the radiation field may have a different effective temperature to the medium, but the medium itself is in thermal equilibrium.

Stellar Atmosphere models

Recombination or free-bound

Line emission or bound-bound

We can write an emission coefficient for a medium where line emission is taking place by considering the Einstein coefficient spontaneous de-excitation, A_{21} . If ν_0 is the rest frequency of the centre of the line, and $\phi(\nu)$ is the line profile (assumed to be the same as for absorption) then we can write

$$\epsilon_\nu = \frac{h\nu_0}{4\pi} \phi(\nu) (n_2 A_{21})$$

where n_2 is the number density of ions in the upper state and A_{21} is the chance of spontaneous de-excitation. $h\nu$ is the energy of each photon produced, we assume they are reradiated isotropically, we divide the energy by 4π steradians so then the equation gives energy per unit volume, per unit solid angle, per second.

Chapter 2

Balance

2.1 Introduction

The balance program provides a testbed for the ionisation cycle of the python code. A plane parallel wind is created in a grid of 9 cells (3x3) which is then illuminated and the ionisation state calculated. This simple geometry is designed to allow detailed interrogation of the ionisation state of the wind in order to facilitate comparison with codes like cloudy.

2.2 Inputs

Balance uses by default the standard39 set of atomic data held in the data66 directory. The files contained in this database are:

- `elem_ions_ver.py` : This file contains elemental abundances and the list of ions that will be read in. Where the actual data comes from depends on what is available. If nothing 'better' comes from the following three files, the data here is used to work out the populations. The data format is IonV : Element : Atomic number : Ionisation state (1=neutral) : Degeneracy : Ground ionization potential : plus some more.
- `topbase_levels_hhe.py` : This file contains the ion data for Hydrogen and Helium.

The format is LevTop : Atomic Number : ionisation state (1=neutral) : plus lots more....

- `topbase_levels_cno.py` : This file contains the ion data for Carbon, Nitrogen and Oxygen
- `levels_ver.py` : This file contains data for all elements, however if data has already been read in from one of the topbase files, it will not be over written. The format is Level : Atomic number : Ionisation State (1=neutral) : Level number (0=ground) : Degeneracy : Energy in eV.
- `lines_linked_ver.py`
- `excited.py` : This gives data for the energy levels of the ions to be used.
- `recomb.data`
- `topbase_h1_phot.py`
- `topbase_he1_phot.py`
- `topbase_he2_phot.py`
- `topbase_cno_phot.py`
- `photo_fkvy.data`
- `coll.data`

The atomic data is read in by `get_atomic.data` (in `get_atomicdata.c`) and I think its pretty vital to understand this in order to add in more data should we need to! The first thing that `get_atomic.data` reads is the masterfile (currently `standard39`, and it then processes the files one by one, choosing what type of data is contained in each line of data from the first word. These are the switches

control word	meaning
--------------	---------

! or #	A comment line
Element	Reads in the atomic number, the name of the element and its abundance which is immediately converted into abundance relative to hydrogen.
Ion or IonV	Reads in the atomic number, ionisation state, statistical weight, and ionization energy above ground for all. If the keyword in IonV, it also reads in the maximum number of levels and the number of nonlte levels. If nlte=5, it makes levels do a non lte calculation. At present the ions with nlte=5 are H, He, C, N and O. It will only read in an ion line for elements mentioned in the element part of the file.
LevTop	This reads in ions which have a topbase type record. The program reads in atomic number, ionisation state (1=neutral), angular momentum figure, unique level number, energy below free, energy above ground state (in eV but converted to ergs), statistical weight, principle quantum number (non integer in topbase). As before, it will only read in anything if the element is known to the code. I think that the first five levels are set as nonLTE levels for use in the initialisation of the wind.
LevMacro	Macroatoms - not sure about this yet
Level	These are LTE levels. Read in is the atomic number and ionisation state (0=neutral) - these are used to identify the ion, quantum number, statistical weight and the energy above ground state. If this ion has not already been read in, or has already got TopBase levels, the data is ignored. All this data gets stored in the config data structure.
PhotMacS	

PhotTop	<p>These are photoionisation data generated by topbase. The first line is a summary file which contains the atomic number, ionisation state, angular momentum and topbase level identifier - both just used to pair up this data with levden, minimum energy required to further ionize the ion, and finally the number of values. There are then a list of paired numbers, which are the cross section for a range of photon energies. These data all go into phot_top</p>
Phot	<p>This is a 'FKVY' type data also for photoionisation. This is a way of describing the crossection for photons being captured for photoionisation using a fit. Read in is the atomic number, the number of electrons currently in the ion (i.e. 2 for helium converts to an ionisation state of 1, neutral), threshold frequency in eV, maximum applicable frequency for the fit, the the fit parameters; freq0, sigma, ya, p, yw, y0 and y1. These are all stored in the structure xphot after various multiplicative factors are applied to get the data into cgs units. The equation is (Verner et al. (1996))</p> $F(y) = [(x - 1)^2 + y_w^2] y^{0.5P-5.5} \left(1 + \sqrt{y/y_a}\right)^{-P}$ <p>where</p> $\sigma(E) = \sigma_0 F(y), \quad x = \frac{E}{E_0} - y_0, \quad y = \sqrt{x^2 - y_1^2}$
Line	<p>This reads in a resonance line. Read in are the atomic number, ionisation state (0=neutral), wavelength in angstroms, oscillator strength, statistical weight of the lower level, then the upper level, energy of the lower level and higher level and finally the lower level of the transition followed by the upper level. These data are all stored in the structure lines.</p>

LinMacro	
Frac	This reads in "ground state fractions". Read in is the atomic number, ionisation state, then 20 numbers which go into the array ground.frac. These are the fraction of transitions which return straight to the ground state, tabulated for values of T_e . They are used in the <code>lucy_mazzali</code> algorithm to improve the estimate of ionisation fractions in the cloud over the Saha abundances.
Xcol	
InPhot	

2.3 Topbase photon ionisation data

Currently topbase photoionisation data is read in for HI, HeI, HeII, C, N, and O. The data is formatted as follows.

The first table shows the first line in the topbase record for HI, stored in `topbase_h1_phot.py`, with the names of the variables that the data is read into in the second line.

PhotTopS	1	1	200	1	13.605698	50
Label	Z	istate	islp	ilv	exx	np

The label simply points out to the reading algorithm that we are dealing with a topbase record. Z is used to check that we are reading data into the correct element record, and istate is used to ensure we are reading into the correct ionisation state for that ion. The data field read into islp is a description of angular momentum used in topbase to identify ions $isp = (2s + 1) \times 100 + l \times 10 + p$. The value held in exx is the threshold energy for ionisation (although this is not used) and np is the number of discrete energies for which cross sections are provided. Following this line, the file contains a list of cross sections for energies. The table below contains the first three lines and the last line for the photoionisation data of the ground state of HI

PhotTop	13.605698	6.304e-18
PhotTop	16.627193	3.679e-18
PhotTop	19.648688	2.327e-18
PhotTop	161.658966	4.316e-21
Label	xe	xx

The data field xe is the energy in ergs, which is converted into frequency before storage in the field phot_top[ntop_phot].freq[n] (ntop_phot is a counter for all topbase photoionisation cross sections and n is the number of different frequencies for which we have information, 50 here) and xx is the photoionisation cross section in cm^2 which is stored in phot_top[ntop_phot].x[n]

For HI, HEI and HEII, the data is stored in individual files, however the topbase data for Carbon, Nitrogen and Oxygen are all in one file. The number of energy levels for each of the ions are as follows.

Element	1	2	3	4	5	6	7	8
H	10							
He	10	10						
C	220	44	78	24	53	55		
N	149	229	54	96	24	53	55	
O	65	173	243	81	96	24	53	55

not all of these levels are read in for each ion. The first level is the ground state for that ion, and this is always read in. The subsequent lines represent ionisation from excited states.

2.4 fkv photo ionisation data

This data is read in for ions which do not have topbase data. The data is a set of parameters which are used to compute the photoionisation cross section for an arbitrary frequency between the threshold frequency and a maximum frequency given as part of

the data. There is only one line for each ion, representing photoionisation from the ground state only. The format of the data is as follows, with examples for H1, He1 and He2.

(Ion)	Label	z	nelectrons	et	emax	e0	sigma	ya	p	yw	y0	y1
HI	Phot	1	1	13.6	50000	0.429	54750	32.88	2.963	0	0	0
HeI	Phot	2	2	24.59	50000	13.61	949.2	1.469	3.188	2.039	44.34	2.136
HeII	Phot	2	1	54.42	50000	1.72	13690	32.88	2.963	0	0	0

Z is the atomic number, nelectrons is confusingly the number of electrons remaining in the ion, and this is used to compute the ionisation state by simply calculating $(z - \text{nelectrons}) + 1$. The next parameter et, is the threshold energy (in ergs) which is converted to frequency units before storage, emax is the maximum energy for which the fit applies, and this is also put into frequency units. The next 7 numbers are the parameters for the fit. Data is only given for the ground state of each ion.

2.5 Ionisation Cycle

The first pass through the program, it initialises the numbers of ions using the user input electron temperature. This procedure is carried out in the program partition_functions in the file partition.c. It is a little convoluted how this gets called from balance, but basically balance calls pl_wind_define (in plane.c) which first sets up a plane parallel wind, then calls ion_abundances. This first pass through is really just to ensure that the wind has properly set parameters for the ion densities.

2.5.1 Partition function

The partition function Z for an ion is used to calculate the number of atoms in each state and is calculated using this equation

$$U_{jk} = \sum_j g_{jk} e^{-\frac{E_{jk}}{kT}}$$

where U_{jk} is the partition function of atom k in ionisation state j in the notation of Mihalas, g_j is the degeneracy of the level (simply an integer saying how many levels in the ion have the same energy) E_j is the energy of that level relative to the ground state energy and T is the electron temperature - at least first time through. The assumption is

that the cloud is in local thermodynamic equilibrium (i.e. in any location, the electrons and ions are at a temperature which can be described by the Boltzmann distribution and the photons have the same temperature and are in a blackbody distribution). Partition functions are calculated for each of the ions read in during the first phase of the program, but the way it is done is slightly different depending on what data is available. If a full set of states for a given ion is available, then Z is summed over all states. If the ion simply has an ionization potential and a degeneracy (i.e. there is no entry in either `topbase` file or `levels_ver`, the degeneracy is used for the partition function.

Once the partition functions are called, they can be used to work out the abundances of the levels within each ion. This calculation is carried out in `levels`, and is automatically called at the end of `partition`.

2.5.2 Levels

Given inputs of the temperature and the statistical weights of each of the ions, `levels` works out the population of each level within each ion that is specified to be calculated in a non-lte fashion before the ionization cycle starts. and puts it into the array `levden`. It only generates levels for ions who have `ion[nion].nlte` greater than 1. This number tells you how many levels there are which are not to be calculated on the fly, but are to be calculated at the start of the ionization cycle. These are levels which do not have detailed information from a level data. The calculation is:

$$\left(\frac{n_{ijk}}{N_{jk}}\right)^* = \frac{wg_{ijk}e^{-\chi_{ijk}/kT}}{U_{jk}(T)}$$

Where n_{ijk} is the number of ions of atomic weight k , ionisation state j in level i , N_{jk} is the number of ions of atomic weight k , in ionisation state j (so the LHS is a fraction of ions in a given excited level), g_{ijk} is the statistical weight of level i , χ_{ijk} is the difference in energy between the lowest possible state and the level i , T is the temperature which can be electron temperature, photon temperature or some black body estimate modified by a dilution factor w , and U_{jk} is the partition function which describes the abundance of that particular ion.

2.5.3 Multicycle (option G)

This piece of code in `balance` calculates the abundances of light elements in order to allow easy comparison with `cloudy`. The first thing that happens is that a range of parameters are set.

Parameter	Value	Meaning
nh	1e12	Number density of hydrogen
t_r	30000	Placeholder radiation temperature, this is a parameter which is cycled over
t_e	20000	Starting electron temperature
weight	0.1	place holder dilution factor - this parameter is also cycled over
t_rmin	2000	Starting radiation temperature
t_rmax	100000	Maximum radiation temperature
t_rdelt	2000	Step in radiation temperature
w_max	0	Maximum dilution factor (log scale)
w_min	-6	Minimum dilution factor (log scale)
w_delt	2	Step in dilution factor (log scale)

As a first guess, the code uses the Mazzali & Lucy approximation of $T_e=0.9T_r$ (Mazzali and Lucy (1993)) and begins a calculation to work out the ionization fractions and correct electron temperature. This is analagous to the ionisation cycle in `python`, but the radiation field is assumed rather than calculated. It is calculated in a subroutine specific to `balance` called `cycle`

2.5.4 Cycle

This routine (contained in the file `balance_sub`) is specific to `balance` and tries to mimic what happens in the ionisation cycle of `python`. There is only one cell, and the radiation field is imposed upon that cell using a radiation temperature T_r and a weighting factor w which takes account of the fact that the cell is not necessarily optically thick, so whilst the distribution of photon energies can be modelled using the blackbody function, the

intensity is lower.

2.5.5 nebular_concentrations

This routine (contained in the file saha.c) is the driving routine for all the ionization calculations. It has three possible modes:

- 0: Use the Saha equation with the temperature set as T_r
- 1: Use the Saha equation with the temperature set as T_e
- 2: Use the Lucy and Mazzali formulation (this is the default when called from cycle)

First partition_functions is called, which populates the matrix U_{jk} , the partition function of atom j in ionisation state k, then concentrations is called.

2.5.6 concentrations

The first thing that this routine does is to estimate a value of the number density of hydrogen using the Saha equation. The Saha equation is an expression for the balance between ions in two states, based upon the number of electrons and the temperature of the surroundings. The premise is that electrons at a certain temperature will be able to ionise surrounding atoms, which will themselves contribute to the electron pool and so on until an equilibrium is found. We are therefore looking for an equation which demonstrates the statistical equilibrium between a pool of electrons and an atom in two adjacent ionisation states. The Saha equation is as follows, carrying on with the notation of Mihalas

$$\left(\frac{N_{j+1,k} n_e}{N_{j,k}} \right) = \frac{U_e U_{j+1,k}}{U_{j,k}} \left(\frac{2\pi m_e kT}{h^2} \right)^{3/2} e^{-\chi_{I_{jk}}/kT} \quad (2.1)$$

Where $N_{j+1,k}$ is the number density of atom k in ionisation state j+1, i.e. it has lost a further electron in this balance, $N_{j,k}$ is the atom in the lower ionisation state, N_e is the number density of electrons, $U_{j+1,k}$ and $U_{j,k}$ are the partition functions of the two species whilst U_e is the partition function for the electron - this is 2 (it can either be

spin up or spin down), the quantity in brackets raised to the 3/2 power is $1/\Lambda^3$ where Λ is the thermal de Broglie wavelength of an electron at temperature T , and $\chi_{I_{jk}}$ is the ionisation potential required to raise the ionisation state from j to $j+1$. Note that this equation is for LTE, we will see later how the code deals with non LTE.

This equation links two consecutive ionisation states of the same atom, however there are many possible states, and these are all linked. Also, all of the other species are linked via n_e so we actually have a large system of equations. In theory these can be solved by a matrix inversion, and this may be a way forward in the future, however at the moment an iterative scheme is used. In order to make a start on this, we need a reasonable estimate of n_e and this is produced by using the Saha equation for hydrogen alone. Firstly, in a normal gas (at least until temperatures get above about 20000K and electrons start to occupy energy levels above the ground state, most hydrogen atoms will have a degeneracy of $U_{1,h}=2$. Ionised hydrogen clearly has a degeneracy of 1 so $U_{2,h}$ is also 1. Therefore in at temperatures up to 20000K the Saha equation for hydrogen reduces to

$$\left(\frac{N_{j+1,H}n_e}{N_j}\right) = \left(\frac{2\pi m_e kT}{h^2}\right)^{3/2} e^{-\chi_{I_H}/kT} \quad (2.2)$$

We can now reduce the number of unknowns by realising that the ratio

$$\frac{N_{j+1,H}}{N_{j+1,H} + N_{j,H}}$$

is simply the ratio of hydrogen ions to hydrogen atoms in all states of ionisation (there are only 2) which is the same as n_e/n_H , since all ionised atoms release an electron. We can therefore write

$$\frac{n_e}{n_H} = \frac{N_{j+1,H}}{N_{j+1,H} + N_{j,H}}$$

and so we can write down an expression for n_e

$$n_e = \frac{N_{j+1,H}}{N_{j+1,H} + N_{j,H}} n_H$$

We can therefore rewrite the Saha equation as follows

$$\frac{N_{j+1,H}}{N_{j,H}} \frac{N_{j+1,H}}{N_{j+1,H} + N_{j,H}} n_H = \left(\frac{2\pi m_e kT}{h^2}\right)^{3/2} e^{-\chi_{I_H}/kT}$$

or

$$\frac{N_{j+1,H}^2}{N_{j,H}(N_{j+1,H} + N_{j,H})} n_H = \left(\frac{2\pi m_e kT}{h^2} \right)^{3/2} e^{-\chi_{I_H}/kT}$$

We now define a new term $N = (N_{j+1,H} + N_{j,H})$ (so $N_{j,H} = N - N_{j+1,H}$) to give us (taking the number density of hydrogen over onto the RHS)

$$\frac{N_{j+1,H}^2}{N(N - N_{j+1,H})} = \frac{1}{n_H} \left(\frac{2\pi m_e kT}{h^2} \right)^{3/2} e^{-\chi_{I_H}/kT}$$

This is solvable if we substitute x as the ionisation fraction, that is the ratio of ionised hydrogen to the total number of hydrogen atoms so

$$x = \frac{N_{j+1,H}}{N}$$

Further using the definition used in saha.c of

$$\Theta = \frac{1}{n_H} \left(\frac{2\pi m_e kT}{h^2} \right)^{3/2} e^{-\chi_{I_H}/kT}$$

We can write our problem as

$$\frac{x^2}{1-x} = \Theta$$

So

$$0 = x^2 - \Theta - \Theta x$$

and using the standard equation for solving a quadratic we get

$$x = \frac{-\Theta + \sqrt{\Theta^2 + 4\Theta}}{2}$$

We then multiply by the known number density of hydrogen in the cell to get an initial estimate of $N_{j+1,H}$ and hence N_e . This will now allow us to apply the Saha equation to get an initial estimate of the relative abundances of all of the species, prior to irradiation. The next routine that is called is saha. Note that by computing the concentrations of many more ions, n_e will change. This will require a new estimate of the concentrations and so on until n_e stabilises. As mentioned previously, it is possible in theory to solve this system of linked equations, however at the moment this iterative scheme is used.

2.5.7 Saha

This subroutine (contained in `saha.c`) applies the Saha equation to one cell of the wind with a value of temperature which can be either T_e , T_r or a weighted blackbody estimate depending on what the calling routine is assuming. Clearly the Saha equation works out the relative number density of pairs of species, so we need to make an assumption of the number density of the ground state ion. At first this is simply set to 1. The equation used is

$$\left(\frac{N_{j+1,k}}{N_{j,k}}\right) = \frac{1}{n_e} \frac{U_e U_{j+1,k}}{U_{j,k}} \left(\frac{2\pi m_e kT}{h^2}\right)^{3/2} e^{-\chi_{I_{jk}}/kT}$$

And we cycle through all the possible ionisation states of all of the elements. Once we have an estimate of all of the abundances, the electron density is updated (simply by adding on all of the electrons created from the new ions). If this has not changed significantly then Saha is not called again, and the abundances are fixed. At this stage we have estimates of the ion concentrations based upon the assumption of LTE. This is probably not true, so we need to improve our estimates. Control is handed back to `nebular_concentrations` and from there the code moves on to the subroutine `lucy`.

2.5.8 lucy

This routine is simply an iteration loop, once again looping until n_e has converged. It calls the routine `lucy_mazzali` (in `saha.c`)

2.5.9 lucy_mazzali

This is the subroutine which improves upon the saha abundances by the application of an equation suggested in Mazzali and Lucy (1993).

$$\frac{n_{j+1,k} n_e}{n_{j,k}} = W[\zeta + W(1 - \zeta)] \left(\frac{T_e}{T_r}\right)^{1/2} \left(\frac{n_{j+1,k} n_e}{n_{j,k}}\right)^*_{T_r}$$

The starred term on the left hand side is ratio of ions as calculated already using the Saha equation, assuming LTE. This equation modifies those abundances to take account of the fact that we are not in LTE, although if $W=1$ and $T_e = T_r$ then the

multiplicative term reduces to 1. This equation is based on the nebular model, which assumes that collisional de-excitation is unimportant so excited levels decay via purely radiative channels and also that ionisation occurs only from the ground state. This means that the atom is treated as a two level system, a ground state, and a photo-ionised state with a continuum of possible energy levels. The basic nebular approximation is

$$\frac{n_{j+1,k}n_e}{n_{j,k}} = W \left(\frac{T_e}{T_r} \right)^{1/2} \left(\frac{n_{j+1,k}n_e}{n_{j,k}} \right)_{T_r}^*$$

However we improve upon this by realising that in reality, not all recombinations go directly to the ground state so we multiply by the factor ζ which is the fraction of recombinations which *do* go to the ground state. A value less than 1 will tend therefore to reduce the number of ions in the $j+1$ state over what was calculated assuming LTE since there are fewer atoms in the lower state to be ionized. The second factor of $W_2(1 - \zeta)$ represents photoionisation from excited levels and further reduces the number of ions in the $j+1$ state because they have been further ionised. The appearance of both T_e and T_r in the equation allows for the departure from LTE, although T_e still describes a maxwellian distribution of electron energies, and T_r describes a black body type spectrum, although diluted from the optically thick case. In balance (G option), these two parameters are passed to `lucy_mazzali` however in python T_r is calculated from the photon distribution. We may well run into trouble here with the quasar, since it has a non blackbody spectrum, so we will need to think of new ways of setting a reasonable value for T_r here.

The parameter ζ is obtained from tabulated data in the file `recomb.data`, and is provided for a range of electron temperatures from $2000K \leq T_e \leq 40000K$. If T_e is less than 2000K, ζ is set to the lowest value in the table and if it is greater than 40000K, ζ is set to the highest value in the table. The new abundances are then simply calculated and the electron density recalculated. If the electron density doesn't change by more than the parameter `FRACTIONAL_ERROR` (currently set to 0.03) then the cycle (run by `lucy`) exits with the modified ionic abundances. Control is then returned to `nebular_concentrations`, and then back upwards to cycle. At this stage we have computed the densities of all the ions based only upon the temperature of the cloud. We can now

irradiate it and see how that changes things.

2.5.10 bands_init

The first thing we do is set up some frequency bands by calling `bands_init` with the radiation temperature and mode 0, which sets up one band (so basically just frequency limits) based upon the temperature. Limits which are set are $freq_{min} = kt/0.05h$ and $freq_{max} = kt/20h$. Then the photon structure is populated with a blackbody spectrum of photons in `xbb`

2.5.11 xbb

This is a balance specific routine which generates a blackbody spectrum of photons. Exactly what it does depends on what the chosen mode is. If we have a uniform approach, the weights of the photons are set to

$$w = W \left(\frac{4\sigma T^4}{N} \right)$$

In this case, this refers to the energy density of a blackbody cavity multiplied by W where W is the dilution factor because the radiation is diffuse and N is the number of photons. Normally then energy density equation is divided by the speed of light, so in this case each photon weight is ends up multiplied by the speed of light. This is immaterial because the whole approach is one of looking at changes, however a more rigorous geometric approach may end up with much lower numbers for photon weight. If the frequency is banded, then the blackbody function is integrated over the frequency range of interest, and the weights set so that the number of photons in that band add up to the correct fraction of the total emittance of the blackbody source.

It is probably at this stage that we will need to put a new function in to irradiate the plasma with a power law, or perhaps a blackbody combined with a power law. The best way of doing this may be to set the radiating source as a fixed object, work out the luminosity and use W to define how far away we are from it using the equation below from Sobolev (1969).

$$W = \frac{1}{2} \left[1 - \sqrt{1 - \left(\frac{r_*}{r} \right)^2} \right]$$

with r_* as the radius of an extended blackbody source and r being the distance away from the source.

2.5.12 line_heating

Once we have our first shot attempt to estimate the ionisation state of the nebula, purely based upon either the electron temperature or the radiation temperature (assuming a blackbody distribution) we are ready to expose the cloud element to photons, and see if this modifies the temperature of the cloud. There are two mechanisms by which this can happen, firstly, line heating.

The software which deals with this is in the file `balance_gen.c`. Since there is no velocity profile in the cloud as defined in `balance`, the line needs to have a range of frequencies over which it can be excited. `line_heating` is called from `cycle` once for each photon, and is sent the frequency ν of that photon. The first thing that `line_heating` does is define a range of frequencies for the photon from $0.95\nu \leq \nu \leq 1.05\nu$ which is exactly analogous to saying that each potential line with central frequency ν_{line} can be excited by a photon of frequency between $0.95\nu_{line} \leq \nu_{line} \leq 1.05\nu_{line}$. We define the scattering cross section of the line ϕ to be uniform at 0.1ν so the total area under the line $\int \phi d\nu$ is 1, as expected.

The code then calls `limit_lines` which generates a list of lines which occur within the limits. We now cycle through this list of lines, to discover how much heating of the plasma will occur from the absorption of photons in these lines.

When photons are absorbed by an ion in a resonance, there are two possible outcomes. Firstly, the ion can spontaneously de-excite with the emission of a photon identical to the original. This is a scattering process and leaves the photon field undisturbed. The second possibility is that the ion can be collisionally de-excited, in which case the energy of the photon is lost to the radiation field and instead heats up the electrons in the nebula. The fraction that goes into heating the nebula is calculated by the subroutine `scattering_fraction`.

We now calculate the actual probability of a photon being absorbed into a resonance

using the subroutine `line_nsigma` which also calculates an updated estimate of the level populations through a call to `two_level_atom`.

Once the probability of the photon being absorbed by each of the possible lines is calculated and added together to form a total possibility of the photon being absorbed this is converted into an optical depth by multiplying by the volume of the cell and the 'cross section' parameter of the photon. By multiplying this optical depth by the weight of the photon (i.e. the energy flux carried by that photon packet) we discover how much energy per second is supplied by photons collisionally de-excited from that resonance state and put into the electron population.

2.5.13 radiation

Once the heating effect of the resonance lines for our current photon of interest has been calculated, we need to make sure all the radiation parameters are up to date. Firstly, the free-free opacity of the cell is calculated using the subroutine `kappa_ff`

Next, radiation calculates the opacity of the cell due to photoionisations. Firstly, species with `topbase` data are dealt with, using `sigma_phot_topbase`, then species with `VFKY` values using `sigma_phot`. Both of these routines return a cross-section σ which gives the probability of a photon being absorbed through ionisation. The cross section is then multiplied by the density of that ion and the resulting quantity added onto the ongoing absorption coefficient κ .

We now have an absorption coefficient for the photon we are currently considering, passing through the current cell. This absorption coefficient tells us the probability that the photon will be absorbed as it passes through the current region. In the python formulation, we reduce the weight of a photon bundle by a suitable amount to reflect the proportionate loss of actual photons. The quantity we use to calculate this loss of intensity is the optical depth, calculated from

$$\tau = \kappa ds$$

Where ds is the path length through the cell. To improve computational efficiency, there are two regimes considered. In the optically thick case (defined here as $\tau > 0.001$)

we need to use the equation

$$W_{out} = W_{in}e^{-\tau}$$

and the factor $W_{in} - W_{out}$ is stored since this energy will go into heating the electron population. A quantity W_{ave} is also calculated:

$$W_{ave} = \frac{W_{in} - W_{out}}{\tau}$$

This is used to calculate the weighted frequency, used elsewhere.

In the optically thin case, we can get away with just calculating the exponential to the second order of the Taylor expansion which is computationally faster.

$$W_{out} = W_{in} \left(1 + \frac{(-\tau)}{1!} + \frac{(-\tau)^2}{2!} \right)$$

Once again, the factor $W_{in} - W_{out}$ is calculated, along with W_{ave} , in this case the equation used is

$$W_{ave} = W_{in}(1 - 0.5\tau + 1/6\tau^2)$$

Before returning, the mean intensity in the cell is updated

$$J = J + W_{ave} \times ds$$

Where ds is the distance travelled through the cell. Similarly the mean frequency in the cell (used to calculate the radiation temperature) is updated

$$\nu_{ave} = \nu_{ave} + W_{ave} \times \nu \times ds$$

We have now reduced the weight of the photon by a suitable quantity based upon the free free opacity, and any photoionisations. The energy lost by the photon heats the cell, and the energy put into the cell from the various mechanisms is stored in the xplasma array in `heat_ff` (free free heating) `heat_z` (heating due to ions with Z greater than 3) `heat_ion` (ionisation heating). We also note down the ionisation fraction of each ion at this point.

The next stage is to work out what emission comes from the cell given its new ionisation state and temperature. This is carried out in `total_emission`.

2.5.14 total_emission

The subroutine `total_emission` (in the file `emission.c`) steers the calculation of the luminosity of a single wind cell. It calls `total_line_emission` (luminosity due to de-excitation of resonance lines), `total_free` (luminosity due to scattering of photons by free electrons, free-free) and `total_fb` (luminosity produced by electrons recombining with ions, free-bound) and returns the total luminosity, effectively the cooling of the cell.

We next call `num_recomb` which calculates the total number of recombinations occurring in the cell.

2.5.15 num_recomb

The subroutine `num_recomb` calculates the recombination rate in a given cell. The routine cycles over all ions in the plasma, and after a simple check to see if there is sufficient density of that ion to warrant a calculation, the calculation performed is

$$n_i = n_i + n_e n_{i+1}$$

and now that we know all about the radiation and ions in the cell, we call `one_shot`.

2.5.16 one_shot

The first thing that `one_shot` (located in `ionization.c`) does is to calculate a new value for T_e . The subroutine to do this is `calc_te` and this routine simply recalculates the cooling of the cell with different values of T_e with the intention of balancing the heating and cooling and thereby get a better value for T_e away from the initial assumptions of LTE.

2.5.17 integ_fb

Chapter 3

Schmutz approach to calculating ionisation balance

3.1 introduction

In both Python and Balance, the initial guess at the ionisation levels in the nebular is made using the assumption of a radiation field that can be described by a weighted blackbody with colour temperature T_r , and an electron population which can be represented by a Boltzmann distribution with characteristic temperature T_e . With a power law input spectrum, the second assumption is probably fine, since the electrons will thermalise very quickly due to their low mass. The first assumption is clearly in error, since in the optically thin limit, the photons are not a black body. There are several possible ways of producing a non LTE estimate of ion distributions, and this chapter deals with one possibility discussed in Schaerer and Schmutz (1994).

Their approach is to define three different radiation temperatures, based upon the ionisation potential of the ion in question. A non LTE calculation of the relative abundances of HI, HII, HeI, HeII and HeIII is carried out which yields effective temperatures to get balance for an arbitrary input spectrum of photons for each of the three transitions. These temperatures are then used in a normal LTE (or modified LTE) approach but the effective temperature used depends on the particular ionisation potential of the ion

is question. In this chapter, I will first detail the non LTE calculation required.

3.2 Non LTE Photoionization Equilibrium

3.2.1 Hydrogen

It is possible to write down a series of linked equations to express the very simple statement that in a nebula in equilibrium, the absolute number of ionizations to any given state, must equal the number of de-exitations. If this were not true, the densities would be constantly changing and so the nebula would not be in equilibrium. Following Osterbrock (1989) we begin by considering the two states of Hydrogen, H^0 (neutral) and H^1 (ionised).

The equilibrium equation is

$$N_{H^0} \int_{\nu_0}^{\infty} \frac{4\pi J_{\nu}}{h\nu} a_{\nu}(H^0) d\nu = N_e N_p \alpha(H^0, T)$$

The left hand side of this equation represents the excitation from the H^0 state to the H^1 state by incoming photons. The factor a_{ν} is the ionisation cross section for a photon of frequency ν . J_{ν} is the mean intensity (energy per unit frequency, per unit time, per unit area per unit solid angle) so multiplying by 4π gives us energy per unit frequency, per unit time, per unit area and dividing by the energy of one photon gives us the number of photons per unit frequency, per unit time per unit area. Multiplying by the ionisation cross section gives us the number of ionisations per unit frequency per unit time and so integrating over frequency, and multiplying by the number density of candidate ions gives us the number of photoionisations per unit volume per unit time. The right hand side is simply the probability of recombination, $\alpha(H^0, T)$ multiplied by the number density of electrons times the number density of ionised hydrogen atoms or protons. This is the number of recombinations per unit volume per unit time.

Our J_{ν} is defined by our input photon field (in this case a power law) and T is, in our case, a free parameter which we can solve for. At the moment, we have four unknowns (N_{H^0}, N_e, N_p and T), and only one equation. Clearly we need more information. If we had a hydrogen only nebula, we could also say that $N_e = N_p$ since all the electrons

have come from the ionisation of hydrogen. This is not the case if we are to follow the Schmutz formulation, since we need to include Helium.

3.2.2 Adding Helium

Chapter 4

Important concepts

4.1 LTE-Local thermodynamic equilibrium

- Thermodynamic equilibrium is the case where all particles, and radiation in a region are at the same temperature. Local thermodynamic equilibrium is a special case where we say that although different parts of a cloud have different temperatures, things are moving slowly enough that each small part of the cloud can be adequately described by a single temperature although that temperature changes from place to place. In most cases in python, we are not in LTE, because the cloud might be fairly cool (described by T_e) but the radiation might be quite hot. A nice example would be the atmosphere of the earth. It has a temperature of perhaps 300K, but the sunlight passing through it has a temperature of 6000K (defined by the colour temperature of the sun).

4.2 On-the-spot approximation

- This is an assumption which simplifies calculations. We assume that every photon emitted by a recombining ion is absorbed by another ion close by - thereby maintaining the ionisation fraction. This is usually a good approximation since any photon emitted will have a frequency very close to that required to ionise another ion nearby and so will have a very high absorption cross section. This means that we can ignore the contribution of the so called "diffuse field" photons, and only worry about photons coming from the

star or disk.

$$J_{21} = W \frac{2h\nu^3}{e^{h\nu/kT_r} - 1}$$

4.3 Nebular approximation

- atoms collisionally excited will decay via purely radiative channels so can be modelled using the Einstein A coefficient rather than requiring collisional de-excitation to be taken into account. It also makes the assumption that most ionizations occur from the ground state and so atoms can be modelled as a two level system - a ground state and an ionised continuum state. All you then need is information about the ionisation rate and the recombination rate. This approximation, as the name suggests, holds in low density regions.

4.4 Detailed balance

Following the discussion in Rybicki and Lightman (1979) we begin by considering the link between the Einstein coefficients. If we are in thermodynamic equilibrium, the number of excitations out of a given level must equal the number of de-excitations back into it. Otherwise the atoms would be becoming more or less excited and so the effective temperature would be changing. If we say that n_1 is the number density of ions in the lower state, and n_2 is the number density in the upper state, then we can write

$$n_1 B_{12} \bar{J} = n_2 A_{21} + n_2 B_{21} \bar{J}$$

Where B_{12} is the stimulated absorption Einstein coefficient, B_{21} is the stimulated emission Einstein coefficient and A_{21} is the spontaneous emission Einstein coefficient. We then solve for \bar{J} , the mean intensity averaged over the frequency range important for that transition to get

$$\bar{J} = \frac{A_{21}/B_{21}}{(n_1/n_2)(B_{12}/B_{21}) - 1}$$

In thermodynamic equilibrium, the ratio of number density between two states excited by a photon of energy $h\nu_0$ is given by

$$\frac{n_1}{n_2} = \frac{g_1}{g_2} \exp(h\nu_0/kT)$$

where g_1 and g_2 are the level degeneracies and T is the LTE temperature. So we can write

$$\bar{J} = \frac{A_{21}/B_{21}}{(g_1 B_{12}/g_2 B_{21}) \exp(h\nu_0/kT) - 1}$$

However, in LTE, J_ν , the average intensity over all solid angles is simply equal to the black body intensity B_ν since we are immersed in the black body field, and over the typical width of a line, J_ν is a constant because B_ν varies only slowly with frequency. We can therefore write

$$\bar{J} = \frac{2h\nu^3}{c^2} \frac{1}{\exp(h\nu/kT) - 1}$$

so

$$\frac{A_{21}/B_{21}}{(g_1 B_{12}/g_2 B_{21}) \exp(h\nu_0/kT) - 1} = \frac{2h\nu^3}{c^2} \frac{1}{\exp(h\nu/kT) - 1}$$

If these two equations are to equal each other for all temperatures, then we can write two equations, the so called *Einstein relations*. The first states

$$A_{21} = \frac{2h\nu^3}{c^2} B_{21}$$

and is the connection between the spontaneous de-excitation and stimulated de-excitation at the transition frequency. The second is

$$\frac{g_1}{g_2} = \frac{B_{21}}{B_{12}}$$

and shows that the stimulated rates are simply connected by the degeneracies, so if there are more possible lower states than upper states, the stimulated de-excitation rate will be higher than the stimulated excitation rate. Since the relationships have no reference to temperature, they hold in non LTE, and so can be used to work out rates and hence populations in the non LTE state we find in most of a Python wind. This is the basis of the balance, and detailed balance subroutines.

Bibliography

C. Knigge and M. R. and Cottis C. E. Scaringi, S. and Goad. blahblashbhsah. *MNRAS*, 386:1426, 2008.

P. A. Mazzali and L. B. Lucy. The application of Monte Carl methods to the synthesis of early-time supernovae spectra. *A&A*, 279:227–456, November 1993.

Osterbrock. *Astrophysics of Gaseous Nebulae and Active Galactic Nuclei*. University Science books, 1989.

Rybicki and Lightman. *Radiative processes in astrophysics*. John Wiley and Sons, 1979.

D. Schaerer and W. Schmutz. Hydrodynamic atmosphere models for hot luminous stars. *A&A*, 288:231–254, 1994.

Sobolev. *Course in Theoretical Astrophysics*. NASA, 1969.

D. A. Verner, G. J. Ferland, K. T. Korista, and D. G. Yakovlev. Atomic Data for Astrophysics. II. New Analytic FITS for Photoionization Cross Sections of Atoms and Ions. *ApJ*, 465:487–+, July 1996.

.1 Detailed discussion of python and balance routines

.1.1 scattering_fraction

The subroutine `scattering_fraction` in the source file `lines.c` calculates the fraction of atoms which are excited by incoming photons that return to the ground state via spontaneous emission of a photon of the same frequency (and hence leaves the photon field

with the same energy). The other possibility is that the excited ion is collisionally de-excited. In this case, the photon field loses the exciting photon and the lost energy goes into heating the electrons (thereby increasing T_e). It is quite easy to write a simple equation to express this fraction:

$$\kappa_{scat} = \kappa_{tot} \frac{A_{21}}{C_{21} + A_{21}}$$

All this says is that the absorption coefficient due to scattering (or chance of a photon scattering per cm travelled) is just the total scattering coefficient times the ratio of the chance of the ion spontaneously de-exciting (the Einstein A coefficient) to the chance of it de-exciting via either path (so we add the collisional de-excitation chance C_{21} to the A coefficient). Alternatively, we can write this in the form which yields the absorption coefficient due to absorption (and subsequent collisional de-excitation) as

$$\kappa_{abs} = \kappa_{tot} \frac{C_{21}}{C_{21} + A_{21}}$$

We also define this latter ratio as q where

$$q = \frac{C_{21}}{C_{21} + A_{21}}$$

and so

$$\kappa_{abs} = \kappa_{tot} q$$

and

$$\kappa_{scat} = \kappa_{tot} (1 - q)$$

This is not however the whole story, since there is also the question of stimulated emission. In order to get a more precise equation, we first recall the equation for the source function for a medium where both scattering and absorption and collisional de-excitation is occurring.

$$S_\nu = \frac{\epsilon}{\kappa} = (1 - q)J_\nu + qB_\nu(T_e)$$

We can also write an equation for the source function in terms of the Einstein coefficients. Recalling from earlier, we have an equation for the absorption due to lines

$$\kappa_\nu = \frac{h\nu_0}{4\pi} \phi(\nu) (n_1 B_{12} - n_2 B_{21})$$

and also for the emission

$$\epsilon_\nu = \frac{h\nu_0}{4\pi} \phi(\nu)(n_2 A_{21})$$

By dividing one by the other, we get a new equation for source function

$$S_\nu = \frac{\epsilon_\nu}{\kappa_\nu} \frac{\frac{h\nu_0}{4\pi} \phi(\nu)(n_2 A_{21})}{\frac{h\nu_0}{4\pi} \phi(\nu)(n_1 B_{12} - n_2 B_{21})} = \frac{(n_2 A_{21})}{(n_1 B_{12} - n_2 B_{21})}$$

We can rewrite this equation by using the Einstein relations:

$$g_1 B_{12} = g_2 B_{21}$$

$$A_{21} = \frac{2h\nu^3}{c^2} B_{21}$$

Substituting for A_{21} then dividing top and bottom by $n_2 B_{21}$ before substituting for B_{21}/B_{21} we get

$$S_\nu = \frac{2h\nu^3}{c^2} \frac{1}{\frac{g_2 n_1}{g_1 n_2} - 1} \quad (1)$$

If we are in LTE, then the rate of excitations out of any given state must equal de-excitations back into that state, via whatever mechanism. Therefore we can write the following equation

$$n_1(c_{12} + B_{12}\bar{J}) = n_2(c_{21} + A_{21} + B_{21}\bar{J})$$

where the left hand side represents excitation from state 1 to state 2 via collisional excitation and stimulated excitation, and the right hand side is de-excitation from state 2 back to state 1.

We can use this equation to get an alternative expression for n_1/n_2 as follows

$$\frac{n_1}{n_2} = \frac{(c_{21} + A_{21} + B_{21}\bar{J})}{(c_{12} + B_{12}\bar{J})} \quad (2)$$

We can also use the Boltzmann distribution to define the ratio of transtion rates between two states separated by an energy of $h\nu$

$$\frac{c_{12}}{c_{21}} = \frac{g_2}{g_1} \exp - (h\nu/kT_e)$$

so

$$c_{12} = \frac{c_{21} g_2}{g_1} \exp - (h\nu/kT_e)$$

substituting for c_{12} into (2) and also using the Einstein equation for B_{12} we get

$$\frac{n_1}{n_2} = \frac{(c_{21} + A_{21} + B_{21}\bar{J})}{\left(\left[\frac{c_{21}g_2}{g_1} \exp - (h\nu/kT_e) \right] + \frac{B_{21}g_2}{g_1}\bar{J} \right)}$$

Taking out the common factor from the bottom, we now have an alternative formula for g_2n_1/g_1n_2 in terms of only the emission parameters.

$$\frac{g_2n_1}{g_1n_2} = \frac{(c_{21} + A_{21} + B_{21}\bar{J})}{([c_{21}\exp - (h\nu/kT_e)] + B_{21}\bar{J})}$$

and we can now substitute this back into equation (1)

$$S_\nu = \frac{2h\nu^3}{c^2} \frac{1}{\frac{(c_{21}+A_{21}+B_{21}\bar{J})}{([c_{21}\exp-(h\nu/kT_e)]+B_{21}\bar{J})} - 1}$$

CANT FOLLOW THE DERIVATION FROM HERE - NEEDS FIXING

If the routine is called with option 2 (the default when called from cycle) then the equation to work out the absorption fraction is given by the following equation.

$$q = \frac{c_{21} (1 - e^{-h\nu/kT_e})}{A_{21} + c_{21} (1 - e^{-h\nu/kT_e})}$$

The value returned by the function is actually 1-q, the scattering fraction. This represents the proportion of photons which are effectively returned to the radiation field rather than being absorbed and increasing the temperature of the electrons. Following this function, the function `line_nsigma` is called. This calculates the total line absorption cross section for the transition we are currently interested in.

.1.2 line_nsigma

The absorption cross section due to an individual line is

$$\sigma = n_1 - \frac{g_1}{g_2n_2} \left(\frac{\pi e^2}{mc} \right) f_{12} \phi_\nu$$

Where f_{12} is the oscillator strength of the line, a dimensionless quantity related to the Einstein coefficients which is a measure of the strength of a line. The quantities n_1 and n_2 are the number densities of the ion in the two states, and this is recalculated before the cross section is worked out in the function `two_level_atom` (in `lines.c`).

.1.3 two_level_atom

The number densities of the two ion levels involved in the line we are calculated is worked out using this equation,

$$\frac{n_2}{n_1} = \frac{c_{12} + B_{12}WB_\nu(T_r)}{c_{21} + A_{21} + B_{21}WB_\nu(T_r)}$$

where $J = WB_\nu(T_r)$ is the weighted blackbody function for the local radiation field. This is simply a statement that in order to maintain the ratio of ions in the two levels, the rates of transfers into and out of the levels must be in the same ratio. We can use this if we are sticking to the on-the-spot approximation. Where W is very small, this means that the radiation field has little influence in the population of the levels, and therefore only the collisional excitation and de-excitation is important, so the equation is

$$\frac{n_2}{n_1} = \frac{c_{12}}{c_{21} + A_{21}}$$

If the radiation field is high, i.e. near the disk, then the equation simplifies to

$$\frac{n_2}{n_1} = \frac{g_2}{g_1} \frac{W}{e^{h\nu/kT_r} + W - 1}$$

Clearly this equation gives us a ratio between two states, so we need to calculate a value for one or other of the states in order to convert the ratio into actual number densities. There are two different approaches to this, depending on whether n_1 represents ground state, or some excited state. If we are dealing with the ground state, then the equation we use is

.1.4 kappa_ff

This short subroutine simply calculates the free-free opacity of a cell for a particular photon. We give it the frequency of the photon, and the plasma pointer. At the moment, there are two slightly different formulations, one involving only Hydrogen, and the other involving hydrogen and helium.

The equation used is

$$\kappa_\nu^{ff} = \frac{4e^6}{3mhc} \left(\frac{2\pi}{3km_e} \right)^{1/2} T^{-1/3} Z^2 n_e n_i \nu^{-3} \left(1 - e^{\frac{-h\nu}{kT_e}} \right) \bar{g}_{ff}$$

Where Z is the charge of the ion of density n_i , n_e is the electron density, and \bar{g}_{ff} is the gaunt factor. At the moment python and balance take this to equal 1, which is a reasonable assumption, in the case where $h\nu/kT_e \sim 1$. In the case where we are considering ionised hydrogen ($Z=1$) and helium ($Z=2$) and evaluating all the constants, we get the equation used by balance and python

$$\kappa_{\nu}^{ff} = 3.7 \times 10^8 T^{-1/3} n_e (n_H + 4 * n_{He}) \nu^{-3} \left(1 - e^{-\frac{h\nu}{kT_e}}\right)$$

.1.5 sigma_phot_toplevel

This subroutine calculates the photoionisation cross section due to a line included in the toplevel set of data.

.1.6 phot_toplevel

Calculates the cross section of an ionisation using the FKVY data read in from the data files. There are three sub equations

$$x = \frac{\nu}{\nu_0} - y_0$$

$$y = \sqrt{x^2 + y_1^2}$$

$$f1 = (x - 1)^2 + y_w^2$$

$$f2 = y^{0.5P-5.5}$$

$$f3 = \left(1 + \sqrt{y/y_a}\right)^{-P}$$

and the final cross-section is calculated simply

$$\sigma = \sigma_0 f1 f2 f3$$

.1.7 total_line_emission

This subroutine steers the calculation of luminosity from lines. Firstly it calls limit_lines in order to work out what lines are important, followed by lum_lines. This in turn calls two_level_atom to recalculate the number densities of the two levels associated with

the line in question. We also call `scattering_fraction` which returns the proportion of photons which are scattered. The line luminosity is due to those photons which are 'absorbed' so we use $q = 1 - \text{scattering_fraction}$

$$q \left(\frac{g_2}{g_1 n_1} - n_2 \right) \frac{A_{21} e^{h\nu/kT_e}}{1 - e^{h\nu/kT_e}} h\nu V$$

.1.8 total_free

This subroutine (located in `emission.c`) works out the bremsstrahlung or free free radiation from a wind cell. Passed to the routine are the frequency limits, and the electron temperature in that cell. When we have only hydrogen the equation is

$$4\pi \frac{8}{3} \sqrt{\frac{2\pi}{3}} \frac{e^6}{m^2 c^3} \sqrt{\frac{m}{kT_e}} \frac{h n_e * \rho_H g_H^{ff}}{h} \sqrt{T_e V} e^{-h\nu_{min}/kT_e} - e^{-h\nu_{max}/kT_e}$$

and when we have Hydrogen and Helium

$$4\pi \frac{8}{3} \sqrt{\frac{2\pi}{3}} \frac{e^6}{m^2 c^3} \sqrt{\frac{m}{kT_e}} \frac{h n_e * (\rho_H g_H^{ff} + 4\rho_{He} g_{He}^{ff})}{h} \sqrt{T_e V} e^{-h\nu_{min}/kT_e} - e^{-h\nu_{max}/kT_e}$$

.1.9 total_fb

This subroutine (in `recomb.c`) works out the energy lost from the plasma due to recombination (free-bound transition). Firstly `init_freebound` is called to set up the relevant structures

.1.10 calc_te

This subroutine (located in `ionization.c`) attempts to change the electron temperature so as to balance the heating and cooling in the cell. To do this, the subroutine is supplied with two temperatures which bracket the current value of T_e , when called from `one_shot`, the bracket is $0.7T_e$ and $1.3T_e$. This bracketing is to prevent excessive changes in T_e in one attempt. These values are then sent to `zero_emit` a very simple routine which calls `total_emission` to discover the emission of the plasma with the new T_e and then works out the difference between the heating and the cooling.

Upon return to `calc_te`, if the values of T_e do not bracket the correct value (i.e. the one

which provides balance) then T_e is simply set to whichever of the limits is appropriate. If they do bracket the correct value (i.e. one gives a positive value for heating-cooling and one gives a negative value) then a call is made to `zbrent` - a numerical recipe which homes in on the correct value of T_e . This is then returned to the calling routine.