

UNIVERSITY OF SOUTHAMPTON

# Disc Winds Matter: Modelling Accretion and Outflow on All Scales

by

James Matthews

A thesis submitted in partial fulfillment for the  
degree of Doctor of Philosophy

in the

Faculty of Physical Sciences and Engineering  
Department of Physics & Astronomy

April 2016

*“Here, on the edge of what we know, in contact with the ocean of the unknown, shines the mystery and the beauty of the world.*

*And it’s breathtaking.”*

Seven Brief Lessons on Physics, Carlo Rovelli

*“Good enough for government work.”*

Christian Knigge

UNIVERSITY OF SOUTHAMPTON

# *Abstract*

Faculty of Physical Sciences and Engineering  
Department of Physics & Astronomy

Doctor of Philosophy

by James Matthews

Outflows are ubiquitous in accreting systems across 10 orders of magnitude in mass. They can take the form of highly collimated radio jets, or less collimated, mass loaded winds emanating from the accretion disc. Perhaps the most spectacular evidence for accretion disc winds is the blue-shifted, broad absorption lines (BALs) in UV resonance lines, seen in cataclysmic variables (CVs) and approximately 20% of quasars. In addition to directly producing absorption in the spectrum, it is possible that accretion disc winds may significantly affect the line and continuum *emission* from CVs and quasars – as a result, they may even dominate the spectral appearance of such objects. When one considers that disc winds are also a possible mechanism for AGN feedback, it becomes clear that understanding the physics and true spectral imprint of these winds is of wide-ranging astrophysical significance.

In this thesis I use the confusingly named Monte Carlo radiative transfer (MCRT) code, PYTHON, to conduct a series of MCRT and photoionization simulations designed to test simple biconical disc wind models. I provide a detailed description of these methods, focusing particularly on the macro-atom implementation developed by Leon Lucy. First, I apply them to the optical spectra of CVs. Second, I conduct tests of quasar unification models. Finally, informed by the previous study, I use Sloan Digital Sky Survey and Hubble Space Telescope data to test the models in an empirical way, by using emission line equivalent widths as a probe of unification geometries.

Overall, the work presented here suggests that *disc winds matter*. They not only act as a spectral ‘filter’ for the underlying accretion continuum, but may actually dominate the emergent spectrum from accreting objects. As a result, unveiling their driving mechanisms, mass-loss rates, and ionization structure is an important goal for the astronomical community.

# Contents

<b>Abstract</b>	<b>ii</b>
<b>List of Figures</b>	<b>v</b>
<b>List of Tables</b>	<b>vi</b>
<b>1 Monte Carlo Radiative Transfer and Ionization</b>	<b>1</b>
1.1 Fundamentals of Radiative Transfer	2
1.1.1 Spectral Line Formation	3
1.1.2 Local thermodynamic equilibrium	4
1.1.2.1 Dilute approximation	5
1.1.3 The Two Level Atom and Escape Probabilities	5
1.1.3.1 Einstein coefficients	6
1.1.4 The Sobolev Approximation	7
1.1.4.1 Two-level Atom with Escape Probabilities	7
1.1.5 Monte Carlo approaches	8
1.2 PYTHON: A Monte Carlo Ionization and Radiative Transfer Code	8
1.2.1 Basics	9
1.2.2 Radiation Packets	10
1.2.3 Radiative Transfer procedure	11
1.2.3.1 Continuum opacities	12
1.2.3.2 Doppler Shifts	13
1.2.3.3 Choosing packet directions	14
1.3 Macro-atoms	15
1.3.1 Transition Probabilities	16
1.3.2 Rate equations	17
1.3.2.1 Collision strengths	18
1.3.3 Macro-atom estimators	19
1.3.3.1 Bound-free estimators	20
1.3.3.2 Bound-bound estimators	21
1.3.3.3 Other heating and cooling estimators	21
1.3.4 $k$ -packets	22
1.3.5 Putting it all together	23
1.3.6 Ionization Fractions and Level Populations	23
1.3.7 Numerical Issues and Population Inversions	25
1.4 A hybrid line transfer scheme: including simple-atoms	25
1.4.1 Line Transfer	26

---

1.4.2	Heating and Cooling Estimators . . . . .	26
1.4.2.1	Radiation Field Estimators . . . . .	27
1.4.3	Ionization and Excitation . . . . .	28
1.5	Heating And Cooling Balance . . . . .	28
1.5.1	Convergence . . . . .	29
1.6	Spectral Cycles . . . . .	29
1.6.1	Macro-atom Emissivity Calculation . . . . .	31
1.7	Atomic Data . . . . .	32
1.8	Code Validation . . . . .	33
1.8.1	Testing against CLOUDY . . . . .	33
1.8.2	Macro-atom testing against TARDIS and theory . . . . .	34
1.8.3	Testing line transfer modes . . . . .	37
1.9	Code Maintenance and Version Control . . . . .	38
1.9.1	Parallelisation . . . . .	39

**Bibliography****41**

# List of Figures

1.1	A schematic showing a ray obliquely incident on a surface of area $dA$ . The labeled quantities are used in the definition of specific intensity. . . .	2
1.2	A flowchart showing the basic operation of PYTHON. . . . .	9
1.3	The process of choosing a scattering location in a cell. . . . .	12
1.4	The decision tree traversed by an energy packet in macro-atom mode. . .	24
1.5	The average temperature and fraction of converged cells in a typical CV model, shown as a function of the number of ionization cycles completed.	30
1.6	A comparison of synthetic spectra produced in extract and live or die modes. . . . .	31
1.7	The effect of extrapolated cross-sections on the soft X-ray spectrum . . .	33
1.8	Relative ion fractions for Hydrogen in CLOUDY,PYTHON in standard mode and PYTHON in hybrid mode. . . . .	34
1.9	As figure 1.8, but for Helium. . . . .	35
1.10	As figure 1.8, but for Carbon. . . . .	35
1.11	As figure 1.8, but for Nitrogen. . . . .	36
1.12	As figure 1.8, but for Oxygen. . . . .	36
1.13	Balmer decrements in PYTHON compared to Seaton (1959), and He level populations compared to TARDIS. . . . .	38
1.14	Comparison between TARDIS and PYTHON synthetic spectra from a sim- ple 1D supernova model. . . . .	39
1.15	Github commit history for PYTHON from 2013-2016. . . . .	39

# List of Tables

# Chapter 1

## Monte Carlo Radiative Transfer and Ionization

“I’m splashing greys where once was  
glowing white”

*Mike Vennart, Silent/Transparent*

In the previous chapters I have given an introduction to the field and some relevant background relating to accretion discs and their associated outflows. Now it proves useful to discuss some of the specific *methods* I will use in order to answer some of the questions raised in the previous sections. In particular, I will discuss radiative transfer techniques and their potential applications.

*Notation: This section contains a lot of algebraic quantities and sums over ions, levels, and so on. Throughout, I use  $N$  to denote fractional populations of ions and  $n$  to denote fractional populations of levels. The primed quantities  $\ell'$  and  $u'$  follow the convention of [Lucy \(2002\)](#) in that they denote sums over all lower/upper levels. The symbol  $\mathcal{R}$  denotes a total rate (radiative + collisional), and the symbol  $C$  is a collisional rate, whereas  $\mathcal{C}$  is a cooling rate. Starred quantities are evaluated at the stated temperature but in local thermodynamic equilibrium, following [Mihalas \(1978\)](#).*



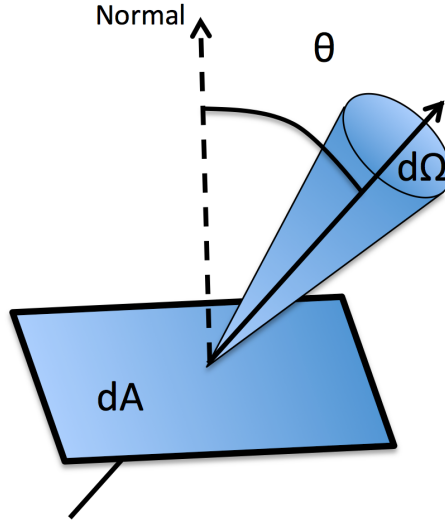


FIGURE 1.1: A schematic showing a ray obliquely incident on a surface of area  $dA$ . The labeled quantities are used in the definition of specific intensity.

## 1.1 Fundamentals of Radiative Transfer

The most fundamental quantity of radiative transfer is the *specific intensity*,  $I_\nu$ , defined as

$$I_\nu = \frac{dE}{d\Omega \, dt \, dA \, d\nu}, \quad (1.1)$$

which has units of  $\text{erg s}^{-1} \text{ Hz}^{-1} \text{ sr}^{-1} \text{ cm}^{-2}$ . By successively multiplying by  $\cos \theta$  and integrating over solid angle we can obtain the first and second ‘moments’ of the radiation field. These are the flux,  $F_\nu$  and momentum flux,  $p_\nu$ , respectively, given by

$$F_\nu = \int I_\nu \cos \theta \, d\Omega, \quad (1.2)$$

$$p_\nu = \frac{1}{c} \int I_\nu \cos^2 \theta \, d\Omega \quad (1.3)$$

We can also define the *mean intensity*,  $J_\nu$ , as

$$J_\nu = \frac{1}{4\pi} \int I_\nu \, d\Omega \quad (1.4)$$

The mean intensity is particularly useful when one wants to ignore the solid angle dependence of the radiation, for example when considering the impact of an ionizing radiation field.

The equation describing the specific intensity change along a path element  $ds$  is the radiative transfer equation,

$$\frac{dI_\nu}{ds} = -\kappa_\nu I_\nu + j_\nu, \quad (1.5)$$

where  $\kappa_\nu$  and  $j_\nu$  are the absorption and emission coefficients respectively. If we define the optical depth  $d\tau_\nu = \kappa_\nu ds$  we can recast this as

$$\frac{dI_\nu}{d\tau_\nu} = -I_\nu + S_\nu \quad (1.6)$$

where  $S_\nu = j_\nu/\kappa_\nu$  is the source function. This equation is called the *formal radiative transfer equation*, and can be solved to give

$$I_\nu = I_{\nu,0} e^{-\tau_\nu} + \int_0^{\tau_\nu} S_\nu(\tau'_\nu) e^{\tau'_\nu - \tau_\nu} d\tau'_\nu. \quad (1.7)$$

A useful limit is when the source function is constant in the absorbing medium, in which case the integral can be easily evaluated to give

$$I_\nu = I_{\nu,0} e^{-\tau_\nu} + S_\nu(1 - e^{-\tau_\nu}). \quad (1.8)$$

### 1.1.1 Spectral Line Formation

From the above equations, it is trivial to show how emission and absorption lines form when the source function is approximately constant. Say we have a plasma illuminated by a blackbody of temperature  $T_0$ , such that  $I_{\nu,0} = B_\nu(T_0)$ . The plasma layer then has a different temperature,  $T$ , such that  $S_\nu = B_\nu(T)$  in that medium. By inspecting equation 1.8 we can see that if we are optically thick within the line, but optically thin in the continuum, then inside the line the source term is dominant and outside the line the first  $I_{\nu,0} e^{-\tau_\nu}$  term dominates. Therefore, if  $T > T_0$  we will see an emission line, and

if  $T < T_0$  we will see an absorption line. This approach describes line emission in the blackbody limit; for more complicated SED shapes it is necessary to construct simple model atoms.

### 1.1.2 Local thermodynamic equilibrium

An important physical limit is that of local thermodynamic equilibrium (LTE). This is a first-order way to describe the physical conditions of a plasma, and assumes that all the properties of the plasma, such as the level populations and source function, are the same as those in thermodynamic equilibrium for local values of temperature and density. For this to be the case, the principle of *detailed balance* must also apply, in which every process by which electrons transition in state must be exactly balanced by its inverse process. LTE also assumes that  $T_e = T_R$ , and that the source function is given by a blackbody, i.e.  $S_\nu = B_\nu(T_R)$ . Three *microscopic* requirements of LTE also follow (Mihalas 1978):

a) The velocities of the electrons and ions in the plasma obey Maxwellian distributions, such that

$$f(v) = 4\pi \left( \frac{m}{2\pi kT} \right)^{3/2} v^2 \exp \left( -\frac{mv^2}{2kT} \right) \quad (1.9)$$

b) the ionization state of the plasma is governed by the *Saha equation*, which states that two adjacent ions have relative populations given by

$$\frac{N_{i+1}n_e}{N_i} = \frac{2g_{i+1}}{g_i} \left( \frac{2\pi m_e kT}{h^2} \right)^{3/2} \exp(-h\nu_0/kT) \quad (1.10)$$

c) the excitation state of the plasma is governed by *Boltzmann statistics*. Two adjacent levels then have relative populations given by

$$\frac{n_j}{n_i} = \frac{g_j}{z_i(T_R)} \exp(-E_j/kT_R), \quad (1.11)$$

Although these three assumptions are often valid, in most astrophysical situations there can be large departures from LTE. A good example of these departures is when the SED is not a blackbody and is affected by absorption – as is the case in AGN and other accreting systems. The Maxwellian assumption is probably the most reliable,

but even this may break down when high-energy photons create suprathermal electron distributions (Humphrey & Binette 2014).

### 1.1.2.1 Dilute approximation

A first step away from LTE is to introduce the dilute approximation. In this case, we relax the assumption that  $T_R = T_e$ , and assume that the mean intensity is given by a dilute blackbody, i.e.

$$J_\nu = W B_\nu(T_R), \quad (1.12)$$

where  $W$  is the dilution factor. We can then approximate the ionization state with a modified Saha equation (Abbott & Lucy 1985; Mazzali & Lucy 1993),

$$\frac{N_{i+1}n_e}{N_i} = W[\xi + W(1 - \xi)] \left(\frac{T_e}{T_R}\right)^{1/2} \left(\frac{N_{i+1}n_e}{N_i}\right)^*_{T_R}, \quad (1.13)$$

where  $\xi$  is the fraction of recombinations that go directly to the ground state. The excitation state can be approximated with a dilute Boltzmann equation (?)

$$\frac{n_{ij}}{n_i} = \frac{W g_j}{z_i(T_R)} \exp(-E_j/kT_R), \quad (1.14)$$

where  $n_{ij}$  is the population of level  $j$  in ionic stage  $i$ ,  $E_j$  is the energy difference between level  $j$  and the ground state,  $g_j$  is the statistical weight of level  $j$  and  $z_i(T_R)$  is the partition function of ionic stage  $i$ .

### 1.1.3 The Two Level Atom and Escape Probabilities

The two level atom formalism is well described by Mihalas (1978). Let us consider an atomic model consisting of two levels that are linked by radiative and collisional transitions, that can also interact with the continuum. Whilst this model is clearly a simplification, it nonetheless allows for a first step into non-LTE line transfer and proves useful for modelling the resonance lines briefly touched on in chapter 2.

To construct our simple model we must make a few assumptions. The first is the assumption of *statistical equilibrium*. This is the principle that the total rate into a given atomic level/state is equal to the total rate out of said state. This is clearly true whenever

the timescale to establish this equilibrium is shorter than the timescale on which the ambient conditions change. The second is the assumption of *complete redistribution* (CRD), which states that the emission and absorption line profiles are identical for a given transition. This assumption is somewhat analogous to the Sobolev approximation (see section 1.1.4). These assumptions allow us to formulate rate equations and derive the Einstein relations.

### 1.1.3.1 Einstein coefficients

Within a two level atom, the rate equation between the two levels in LTE can be written by invoking detailed balance, such that

$$B_{lu}\bar{J}_{ul}n_l = B_{ul}\bar{J}_{ul}n_u + A_{ul}n_u, \quad (1.15)$$

where  $B_{ul}$ ,  $B_{lu}$  and  $A_{ul}$  are the *Einstein coefficients* for absorption, stimulated emission and spontaneous emission respectively. The ‘mean intensity in the line’,  $\bar{J}_{ul}$ , is given by

$$\bar{J}_{ul} = \int \phi(\nu) J_\nu d\nu. \quad (1.16)$$

We can then rearrange equation 1.15 in terms of the mean intensity, giving

$$\bar{J}_{ul} = \frac{A_{ul}/B_{ul}}{(n_l/n_u)(B_{ul}/B_{lu}) - 1}. \quad (1.17)$$

In LTE,  $\bar{J}_{ul} = B_\nu(T)$  and the level populations obey Boltzmann statistics, so we can combine equations ??, 1.11 and 1.17 to write

$$\frac{2h\nu^3}{c^2} \frac{1}{\exp(h\nu/kT) - 1} = \frac{A_{ul}}{B_{ul}} \frac{1}{(g_l/g_u)(B_{ul}/B_{lu}) \exp(h\nu/kT) - 1}. \quad (1.18)$$

This must be true at all values of  $T$ , so we can simply equate coefficients to show that

$$\frac{A_{ul}}{B_{ul}} = 2h\nu_{ul}^3/c^2, \quad (1.19)$$

$$\frac{B_{lu}}{B_{ul}} = g_u/g_l. \quad (1.20)$$

These two equations are known as the *Einstein relations*, and have no dependence on temperature. They are therefore purely atomic properties.

### 1.1.4 The Sobolev Approximation

The Sobolev approximation (SA) is a useful limit used to treat line transfer in fast-moving flows. Originally the theory was mostly applied to stellar winds, although since then a wide variety of astrophysical objects have been modelled using Sobolev treatments, such as accreting systems (this work) and supernovae.

The Sobolev limit is when the local bulk velocity gradients in a flow dominate other any thermal broadening. In the presence of these steep velocity gradients, one can assume that the interaction of a ray with a bound-bound transition takes place over a small resonant zone, known as a ‘Sobolev surface’. The length of this zone is defined by

$$l_s = \frac{v_{th}}{dv/ds}. \quad (1.21)$$

It is important that the physical conditions of the c do not change on this scale. If this is the case, then we can assume that all line interactions for a given frequency will occur at a single ‘resonant’ point. The location at which a given photon will interact with a line of frequency  $\nu_{lu}$  is then given, in velocity space, by

$$v = c \left( \frac{\nu}{\nu_{lu}} + 1 \right). \quad (1.22)$$

The Sobolev optical depth is then

$$d\tau = \frac{\pi e^2}{mc} \left( n_l - n_u \frac{g_l}{g_u} \right) \frac{f_{lu} \lambda_{lu}}{c |dv/ds|}. \quad (1.23)$$

We can see that the physical quantities determining line opacity are therefore the level populations in the plasma, the velocity gradient and the atomic physics associated with the bound-bound transition.

#### 1.1.4.1 Two-level Atom with Escape Probabilities

Let us now write down the rate equation linking our two-level atom,

$$B_{lu} \bar{J}_{ul} n_l + C_{lu} n_l = B_{ul} \bar{J}_{ul} n_u + A_{ul} n_u + C_{ul} n_u, \quad (1.24)$$

where we have now introduced collisional rates  $C_{ul}$  and  $C_{lu}$ . We now seek to find a relation between the source function and the intensity that will simplify the coupled problem of radiative transfer and statistical equilibrium. When we consider a two-level atom plus continuum this can be written as (Mihalas 1978)

$$S = (1 - q)\bar{J}_{ul} + qB(\nu_{ul}), \quad (1.25)$$

where  $B(\nu_{ul})$  is the Planck function at line centre.

### 1.1.5 Monte Carlo approaches

Simple radiation transfer problems can be solved analytically, but with more complicated geometries it is necessary to use Monte Carlo techniques, which are easily solved with modern computing approaches and are intuitively parallelisable problems. I will describe one specific Monte Carlo radiative transfer (MCRT) code, which has been used for the majority of the work in this thesis.

## 1.2 PYTHON: A Monte Carlo Ionization and Radiative Transfer Code

PYTHON<sup>1</sup> is a confusingly named Monte Carlo ionization and radiative transfer code. The general philosophy of the code is to be able to produce synthetic spectra for astrophysical objects with outflows in 2.5D, using a self-consistent ionization treatment. The code is written in C, and has been in development since the mid-1990s. Throughout this time it has been used with application to CVs (Long & Knigge 2002; Matthews et al. 2015), YSOs (Sim et al. 2005), supernovae (Kerzendorf & Sim 2014) and AGN/quasars (Higginbottom et al. 2013, 2014; Matthews et al. 2016). It is also capable of producing spectra for stellar winds and conducting simple photoionization balance calculations for comparison with codes such as CLOUDY. Some more detail on code testing and development can be found in sections 1.8 and 1.9 respectively. Although the operation of PYTHON is well-described by the above authors, it is central to this Thesis and I will thus provide substantial detail on its operation.

---

<sup>1</sup>Named c. 1995, predating the inexorable rise of a certain widely used programming language.

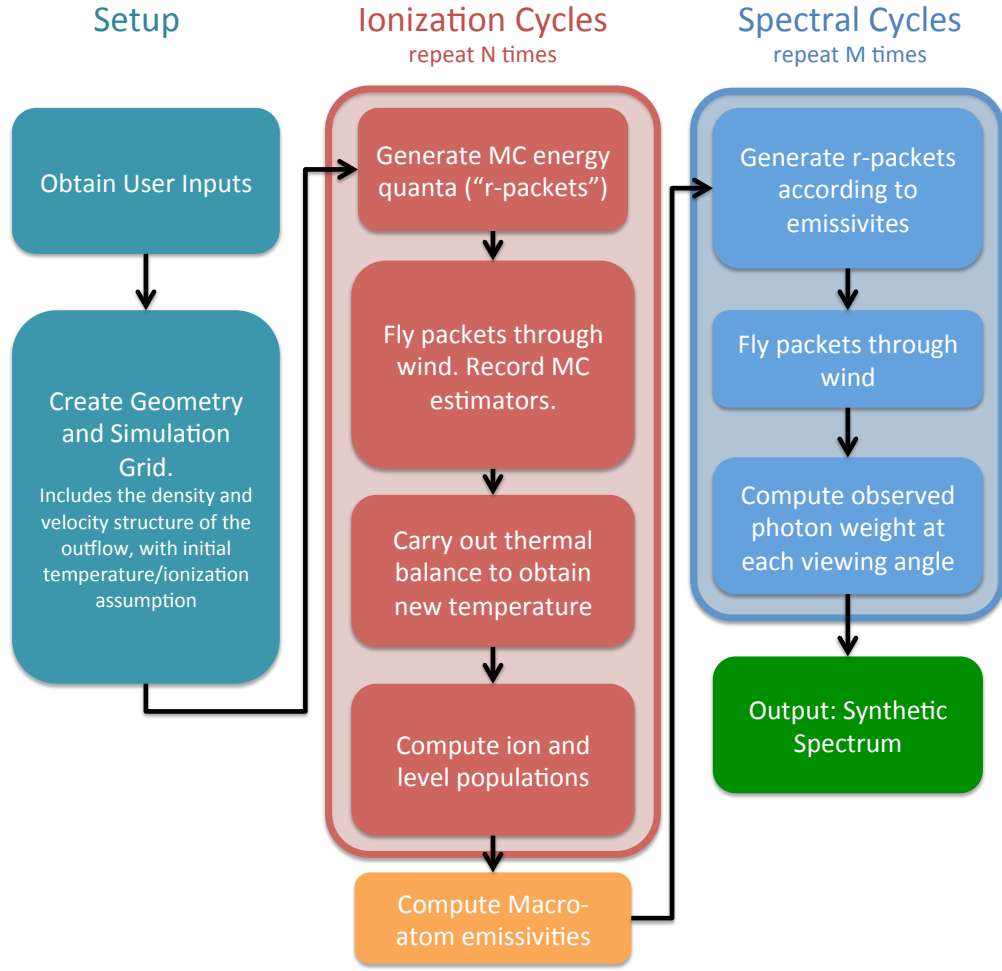


FIGURE 1.2: A flowchart showing the basic operation of PYTHON.

### 1.2.1 Basics

PYTHON operates in three distinct stages, shown in figure 1.2. First, the user specifies the photon sources, geometry and kinematics of the system, normally with a similar parameterisation to the SV93 model described in section ???. The code can operate with multiple coordinate systems (1D, spherical polar, cylindrical), but in this work I use cylindrical coordinates. In this case, the outflow is discretised into a  $n_x \times n_y$  logarithmic grid with user-specified dimensions. The co-ordinates,  $(x_i, z_i)$ , of the corner of the  $i$ th cell are then given by

$$x_i = L_x 10^{(i-1) \frac{\log(R_{max}/L_x)}{n_x}}, \quad (1.26)$$

$$z_i = L_z 10^{(i-1) \frac{\log(R_{max}/L_z)}{n_z}}, \quad (1.27)$$



where  $L_x$  and  $L_z$  are appropriately chosen (but hardwired) scale lengths. From these co-ordinates the poloidal distance can be calculated and the velocity set according to equation ???. The density is then calculated from equation ???. An initial temperature,  $T_{init}$  is set by the user. The ionization fractions throughout the wind are then to Saha (LTE) abundances at  $T_{init}$ , and the level populations are set according to the Boltzmann formula.

Once the basic setup process has been carried out, the ionization state, level populations and temperature structure are calculated. This is done via an iterative process, by transporting several populations of Monte Carlo energy quanta (‘photons’ or ‘ $r$ -packets’) through the outflow. This process is repeated until the code converges. In each of these iterations (‘ionization cycles’), the code records estimators that characterize the radiation field in each grid cell. At the end of each ionization cycle, a new electron temperature is calculated that more closely balances heating and cooling in the plasma. The radiative estimators and updated electron temperature are then used to revise the ionization state of the wind, and a new ionization cycle is started. The process is repeated until heating and cooling are balanced throughout the wind (see sections 1.5.1 and 1.5).

This converged model as the basis for the second set of iterations (‘spectral cycles’; section 1.6), in order to compute the synthetic spectrum based on the MC estimators record during the ionization cycles. The emergent spectrum over the desired spectral range is synthesized by tracking populations of energy packets through the wind and computing the emergent spectra at a number of user-specified viewing angles. In the next sections, I will describe each of the above steps in more detail, particularly with regards to the macro-atom mode of operation.

### 1.2.2 Radiation Packets

Every energy packet in the simulation starts out as a radiation packet generated from one of  $N_S$  photon sources. To ensure that the frequency distribution of photons is adequately sampled in important frequency regimes, *stratified sampling* is used. A specified fraction,  $f_i$ , of photons must then emerge each band  $i$ , whose frequency boundaries can be adapted for the astrophysical situation considered. The weight,  $w_i$ , of the radiation packets in a

given energy band  $i$ , with boundaries  $\nu_i$  and  $\nu_{i+1}$  is then given by

$$w_i = \frac{\sum_j^{N_S} \int_{\nu_i}^{\nu_{i+1}} L_{\nu,j} d\nu}{f_i N_p}, \quad (1.28)$$

where  $N_p$  is the *total* number of photons desired and  $L_{\nu,j}$  is the monochromatic luminosity of photon source  $j$ . The frequency of photons is calculated by constructing a cumulative distribution function (CDF),  $f_{C,i}(\nu)$  from the spectral energy distribution in each band  $i$ :

$$f_{C,i}(\nu) = \frac{\int_{\nu_i}^{\nu} L_{\nu} d\nu}{\int_{\nu_i}^{\nu_{i+1}} L_{\nu} d\nu}. \quad (1.29)$$

A photon frequency can then be generated by cycling through the bands. In each band, a random number is chosen between 0 and 1, and then the frequency is selected by interpolating on the sampled CDF. This process is repeated until each band has the specified number of photons, with the packet weights adjusted accordingly.

PYTHON can operate in two modes concerning the approach to energy packets. In the original mode described by LK02, continuum processes attenuate the weight of the radiation packets. This attenuation is accounted for by including the wind as an additional photon source. In the second mode, energy packets are indivisible and strict radiative equilibrium is enforced. From here on I will only be discussing this indivisible packet scheme, as it is required in order to be able to use macro-atoms to accurately treat recombination in H and He.

### 1.2.3 Radiative Transfer procedure

As a photon travels through a plasma, it has a finite probability of interacting with the free or bound electrons and undergoing a scattering or absorption event. To deal with this in a Monte Carlo sense, a random optical depth is generated before an  $r$ -packet is moved,

$$\tau_R = -\ln(1 - \mathcal{Z}), \quad (1.30)$$

where  $\mathcal{Z}$  is a random number between 0 and 1. The  $r$ -packet is then gradually transported through a given cell. As it moves, the optical depth,  $\tau'$ , it experiences is incremented continuously, representing continuum processes. When the  $r$ -packet comes into resonance with a line, according to equation 1.22, then the Sobolev optical depth

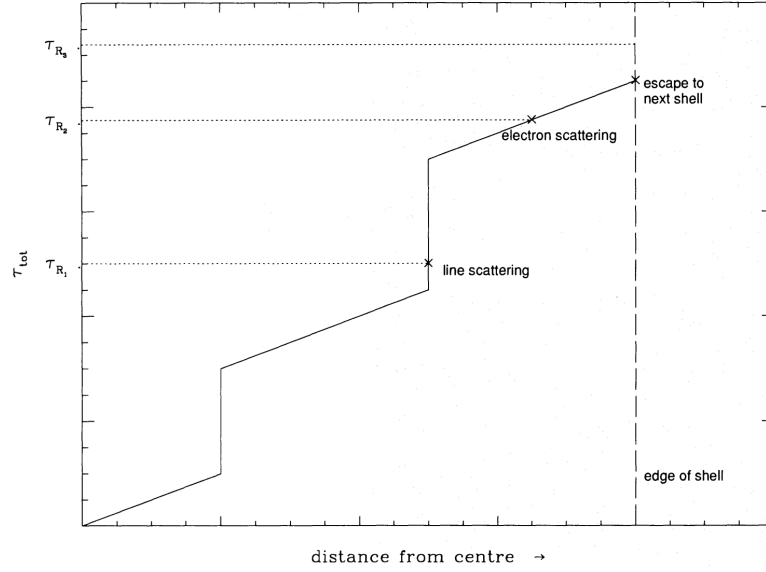


FIGURE 1.3: *Credit: Mazzali & Lucy 1993.* The process of choosing a scattering location in a cell.

is calculated from equation 1.23 and added to  $\tau'$ . This process is shown in Fig. 1.3, and continues until  $\tau' \geq \tau_R$  or the  $r$ -packet leaves the cell. If the photon leaves the cell then the values of  $\tau_R$  and  $\tau'$  are preserved, and the process continues using the conditions in the new cell. If  $\tau' \geq \tau_R$ , then an interaction with the plasma has occurred, and the process governing this interaction must be identified. This is done by randomly picking an interaction process in proportion with their contributions to  $\tau'$ . If the process is an electron scatter then a new, isotropic direction is generated for the  $r$ -packet. Otherwise, the packet must interact with either the thermal pool or the excitation energy of the plasma.

### 1.2.3.1 Continuum opacities

In order to calculate  $\tau'$  in the above approach, we need to know the opacities that will contribute to it. An opacity at a given frequency,  $\kappa(\nu)$ , is related to an optical depth,  $\tau(\nu)$ , by

$$\tau(\nu) = \kappa(\nu) \Delta s, \quad (1.31)$$

where  $\Delta s$  is the distance moved by the photon. The bound-free opacity is calculated from a sum over photoionization cross-section, such that

$$\kappa_{bf} = . \quad (1.32)$$

The free-free emission coefficient for an individual ion  $i$  is (Gayet 1970)

$$j_{ff,i}(\nu) = \bar{g}_{ff} \frac{8Z_i^2 e^6}{3m_e c^2} \frac{2\pi m_e}{3kT_e}^{1/2} N_i n_e \exp(-h\nu/kT_e). \quad (1.33)$$

The free-free opacity is then calculated from Kirchhoff's law,

$$\kappa_{ff,i}(\nu) = \frac{B_\nu(T_e)}{j_{ff,i}(\nu)}, \quad (1.34)$$

which gives

$$\kappa_{ff,i}(\nu) = n_e N_i \frac{4}{3} \left( \frac{2\pi}{3} \right)^{1/2} \frac{Z_i^2}{e^6 m_e^2 h c} \left( \frac{m_e}{kT_e} \right)^{1/2} g_{ff} \nu^{-3} [1 - \exp(-h\nu/kT_e)]. \quad (1.35)$$

The electron scattering opacity is then

$$\kappa_{es} = \sigma_T n_e \Delta s. \quad (1.36)$$

These opacities are all used in the heating and cooling estimators introduced in section 1.3.3. In addition the Compton opacity is required in order to estimate the Compton heating effect on the plasma. The Compton opacity is given by

$$\kappa_C = \sigma_{KN}(\nu) n_e \Delta s, \quad (1.37)$$

where  $\sigma_{KN}(\nu)$  is the cross-section computed from the Klein-Nishina formula (Klein & Nishina 1929), and unlike  $\sigma_T$  is frequency dependent. This opacity is not included in the actual radiative transfer in the simulations presented in this Thesis.

### 1.2.3.2 Doppler Shifts

To calculate the opacities correctly, the frequency must be shifted from the co-moving frame of the photon into the co-moving frame of the cell. This shift depends on the before and after direction of the photon. Let us denote these two directions with unit vectors  $\vec{n}_i$  and  $\vec{n}_f$  respectively, and consider a situation when a photon scatters off an electron in a region of the wind moving at velocity  $\vec{v}$ . The final frequency of the photon

with initial frequency  $\nu_i$  is then

$$\nu_f = \nu_i \frac{1 - (\vec{v} \cdot \vec{n}_i)/c}{1 - (\vec{v} \cdot \vec{n}_f)/c}. \quad (1.38)$$

In the case of a resonance scatter with line transition  $u \rightarrow j$ , the new frequency is

$$\nu_f = \frac{\nu_{uj}}{1 - (\vec{v} \cdot \vec{n}_f)/c}. \quad (1.39)$$

when we consider that the resonant point is chosen according to equation 1.22 and that  $v = \vec{v} \cdot \vec{n}_f$  in this case, it is clear that the above two equations are equivalent.

### 1.2.3.3 Choosing packet directions

The last variable, in addition to  $w_i$  and  $\nu$ , needed to define a radiation packet is the direction of travel. In the case of isotropic emission the direction of a photon packet is chosen so that the probability of emission in each bin of solid angle is the same. It follows that

$$P(\Omega)d\Omega \propto \cos\theta \sin\theta d\theta d\phi, \quad (1.40)$$

where the angles are in polar coordinates and relative to the local outward normal. Clearly, for a spherical emitting source such as a star then one must first generate a location on the star's surface, and then calculate the photon direction relative to the normal at the point. For emission from solid surfaces the above equation is modified to include linear limb darkening,  $\eta(\theta)$ :

$$p(\theta\phi)d\theta d\phi = \eta(\theta)\cos\theta \sin\theta d\theta d\phi, \quad (1.41)$$

where, under the Eddington approximation use in the code,  $\eta(\theta)$  is given by

$$\eta(\theta) = a(1 - \frac{3}{2}\cos\theta). \quad (1.42)$$

The constant  $a$  is simply normalised such that the total probability sums to 1. Whenever a radiation packet undergoes an electron scatter, the new direction is chosen to be isotropic. However, when the photon is a line photon then the new direction is chosen according to a line trapping model, which samples a probability distribution according to the Sobolev escape probability in different directions.

### 1.3 Macro-atoms

The macro-atom scheme was created by Leon Lucy and is outlined in his 2002/03 papers. It was implemented in PYTHON by Stuart Sim, initially for the study of recombination lines in YSOs (Sim et al. 2005)

Lucy (2002, 2003) has shown that it is possible to calculate the emissivity of a gas in statistical equilibrium without approximation for problems with large departures from LTE. His macro-atom scheme allows for all possible transition paths from a given level, dispensing with the two-level approximation, and provides a full non-LTE solution for the level populations based on Monte Carlo estimators. The macro-atom technique has already been used to model Wolf-Rayet star winds (Sim 2004), AGN disc winds (Sim et al. 2008; Tatum et al. 2012), supernovae (Kromer & Sim 2009; Kerzendorf & Sim 2014) and YSOs (SDL05). A full description of the approach can be found in L02 and L03.

The fundamental approach here requires somewhat of a philosophical shift. Normally MCRT is described in the most intuitive way- that is, we imagine real photons striking atoms and scattering, or photoionizing and depositing energy in a plasma. With Lucy's scheme we should instead reimagine the MC quanta as a packets of quantised energy flow, and the scheme as a *statistical* one. The amount of time a given energy quanta spends in a specific atomic level or thermal pool is then somewhat analogous to the absolute energy contained therein.

Following L02, let us consider an atomic species interacting with a radiation field. If the quantity  $\epsilon_j$  represents the ionization plus excitation energy of a level  $i$  then the rates at which the level  $j$  absorbs and emits radiant energy are given by

$$\dot{A}_j^R = R_{\ell j} \epsilon_{j\ell'} \quad \text{and} \quad \dot{E}_i^R = R_{j\ell'} \epsilon_{j\ell'} \quad , \quad (1.43)$$

Where I have adopted Lucy's convention in which the subscript  $\ell'$  denotes a summation over all lower states ( $\ell' < j$ ), and  $u'$  will thus denote a summation over all states ( $u' > j$ ). Similarly, the rates corresponding to *kinetic* (collisional) energy transport can then be written as

$$\dot{A}_j^C = C_{\ell' j} \epsilon_{j\ell'} \quad \text{and} \quad \dot{E}_j^C = C_{j\ell'} \epsilon_{j\ell'} \quad , \quad (1.44)$$

Let us define  $\mathcal{R}$  as a total rate, such that  $\mathcal{R}_{\ell'j} = R_{\ell'j} + C_{\ell'j}$ . If we now impose statistical equilibrium

$$(\mathcal{R}_{\ell'j} - \mathcal{R}_{j\ell'}) + (\mathcal{R}_{u'j} - \mathcal{R}_{ju'}) = 0 \quad . \quad (1.45)$$

we can then obtain

$$\begin{aligned} \dot{E}_j^R + \dot{E}_j^C + \mathcal{R}_{ju'}\epsilon_j + \mathcal{R}_{j\ell'}\epsilon_{\ell'} \\ = \dot{A}_j^R + \dot{A}_j^C + \mathcal{R}_{u'j}\epsilon_j + \mathcal{R}_{\ell'j}\epsilon_{\ell'}. \end{aligned} \quad (1.46)$$

This equation is the starting point for the macro-atom scheme. It shows that, when assuming only radiative equilibrium, the energy flows through a system depend only on the transition probabilities and atomic physics associated with the levels the energy flow interacts with. By quantising this energy flow into radiant ( $r$ -) and kinetic ( $k$ -) packets, we can simulate the energy transport through a plasma discretised into volume elements (“macro-atoms”), whose associated transition probabilities govern the interaction of radiant and kinetic energy with the ionization and excitation energy associated with the ions of the plasma.

Although equation 1.46 assumes strict radiative equilibrium, it is trivial to adjust it to include non-radiative source and sink terms. For example, in an expanding parcel of plasma, adiabatic cooling may be included with a simple modification to the RHS of equation 1.46.

### 1.3.1 Transition Probabilities

Having interpreted equation 1.46 in a *stochastic* way, we can now construct our Monte Carlo scheme, following Lucy (2002). A macro-atom in state  $j$  always has a finite probability of deactivating radiatively or collisionally:

$$p_j^R = \dot{E}_j^R / D_j \quad \text{and} \quad p_j^C = \dot{E}_j^C / D_j, \quad (1.47)$$

where I have defined

$$D_j = \dot{E}_j^R + \dot{E}_j^C + \mathcal{R}_{ju'}\epsilon_j + \mathcal{R}_{j\ell'}\epsilon_{\ell'} = (\mathcal{R}_{j\ell'} + \mathcal{R}_{ju'})\epsilon_j. \quad (1.48)$$

The corresponding jumping probabilities, which describe the probability that the macro-atom transitions to a different state while remaining active, are given by

$$p_{ju'} = \mathcal{R}_{ju}\epsilon_j/D_j \quad \text{and} \quad p_{j\ell'} = \mathcal{R}_{j\ell'}\epsilon_{\ell'}/D_j. \quad (1.49)$$

Note that the jumping probability is always proportional to the energy of the lower level, whereas the emission probability is proportional to the energy *difference* between the levels, as  $\dot{E}_j^R = R_{j\ell'}(\epsilon_j - \epsilon_{\ell'})$ . We can also trivially show that the probabilities are correctly normalised, as

$$\begin{aligned} p_j^R + p_j^C + p_{j\ell'} + p_{ju'} &= (1/D_j)(\mathcal{R}_{ju'}\epsilon_j + \mathcal{R}_{j\ell'}\epsilon_{\ell'} + \dot{E}_j^R + \dot{E}_j^C) \\ &= 1. \end{aligned} \quad (1.50)$$

With these transition probabilities identified, a Monte Carlo calculation can proceed by formulating the normal statistical equilibrium rate equations that will depend on the ambient conditions of the plasma. The effect of these ambient conditions is expressed through the use of Monte Carlo *estimators*.

### 1.3.2 Rate equations

The macroscopic transition probabilities above depend on the traditional rate equations formulated according to statistical equilibrium. In the framework of the Sobolev escape probability formalism (Rybicki & Hummer 1978; L02; Sim 2004), the bound-bound excitation rate,  $\mathcal{R}_{lu}$ , in an ion is given by

$$\mathcal{R}_{ju} = B_{ju}n_jJ_{est} + q_{ju}n_jn_e, \quad (1.51)$$

where  $u$  is now a specific upper level and  $q_{ju}$  is the collisional rate coefficient (see section 1.3.2.1).  $J_{est}$  is the Monte Carlo estimator for the mean intensity impinging on the Sobolev region, weighted by an angle-dependent escape probability, given by (Sim 2004)

$$J_{est} = \frac{c}{4\pi\nu_0 V} \sum_i w_i \frac{1 - e^{-\tau_{s,i}}}{\tau_{s,i}} \frac{1}{(dv/ds)_i}. \quad (1.52)$$

Here  $w$  is the photon weight (in luminosity units),  $\nu_0$  is the line frequency,  $dv/ds$  is the velocity gradient and  $\tau_s$  is the Sobolev optical depth. The sum is over all photons



that come into resonance with the line, and thus represents an integral over solid angle. This is essentially the MC estimator form of  $\beta_{uj}\bar{J}_{uj}$ , and differs from the estimator in equation (20) of [Lucy \(2002\)](#) as there is no assumption of homologous flow or symmetric escape probabilities. The corresponding de-excitation rate is then

$$\mathcal{R}_{uj} = \beta_{ju}A_{uj}n_u + B_{uj}n_uJ_{est} + q_{uj}n_un_e. \quad (1.53)$$

The photoionization and collisional ionization rates between a lower level,  $l$ , and the continuum level  $\kappa$  (or, in the case of ions with more than one bound electron, the ground state of the upper ion),  $\kappa$ , are

$$\mathcal{R}_{j\kappa} = n_j(\gamma_{j\kappa} - \alpha_{\kappa j}^{st}) + q_{j\kappa}n_jn_e. \quad (1.54)$$

Here,  $q_{j\kappa}$  is the collisional ionization rate coefficient,  $\gamma_{j\kappa}$  is the photoionization rate from  $j \rightarrow \kappa$  and  $\alpha_{\kappa j}^{st}$  is the stimulated recombination coefficient. This is included as a negative photoionization term rather than a positive recombination term as in [Lucy \(2003\)](#), which requires that there are no population inversions (see section 1.3.7). The corresponding recombination rate is given by

$$\mathcal{R}_{\kappa j} = \alpha_{\kappa j}n_\kappa n_e + q_{\kappa j}n_\kappa n_e, \quad (1.55)$$

where  $\alpha_{\kappa l}$  is the radiative recombination coefficient to level  $l$ , and is given by

$$\alpha_{\kappa j} = 4\pi\Phi_{j\kappa}^* \int_{\nu_0}^{\infty} \frac{\sigma_{j\kappa}(\nu)}{h\nu} \frac{2h\nu^3}{c^2} \exp\left(\frac{-h\nu}{kT_e}\right) d\nu. \quad (1.56)$$

This treatment means that radiative and collisional rates to and from all levels are considered when calculating both the ionization state and the level populations, although we neglect ionization directly to excited levels of the upper ion.

### 1.3.2.1 Collision strengths

The bound-bound collisional rate coefficient  $q_{lu}$  is calculated from the [van Regemorter \(1962\)](#) approximation, given by

$$q_{ju} = 2.388 \times 10^{-6} \lambda_{uj}^3 A_{uj} g_u \bar{g} \quad (1.57)$$

and the inverse rate can just be calculated by considering detailed balance, such that

$$q_{uj} = q_{ju} \frac{g_u}{g_j} \exp\left(\frac{h\nu_{uj}}{kT_e}\right) \quad (1.58)$$

Using equation 1.57 means that collisions between radiatively forbidden transitions are not taken into account when one splits levels into  $l$ - and  $s$ -subshells, as well as principal quantum number,  $n$  (as we have done with He I; see section ??). Although this approximation is, in general, a poor one, the effect is second order in the physical regime where recombination lines are formed in our models. This is because bound-free processes are dominant in determining level populations and emissivities. We have verified that this is indeed the case in the He I emission regions in our models.

The bound-free collision strengths are calculated using equation (5.79) of Mihalas (1978).

The collisional ionization rate is

$$q_{j\kappa} = 1.55 \times 10^{-13} n_e \bar{g}_i \sigma_{j\kappa}(\nu_0) \frac{h\nu_{uj}}{kT_e^{3/2}} \exp\left(\frac{-h\nu_{\kappa j}}{kT_e}\right), \quad (1.59)$$

where  $\sigma_{l\kappa}(\nu_0)$  is the photoionization cross-section at the threshold energy.  $\bar{g}_i$  is an effective gaunt factor for ion  $i$  and is approximately equal to 0.1, 0.2, 0.3 for  $Z = 1, 2$  and  $> 2$  respectively, where  $Z$  is the atomic number. Note that the use of this estimator implies a  $\nu^{-3}$  shape to the photoionization cross-section, which is only strictly true for for hydrogenic ions. The collisional (three-body) recombination rate is found using the Saha equation and given by

$$q_{\kappa j} = q_{j\kappa} \frac{n_j^*}{n_e^* n_\kappa^*}, \quad (1.60)$$

For numerical reasons, the above two expressions are combined in PYTHON where possible, to avoid multiplying two exponentials together.

### 1.3.3 Macro-atom estimators

To be able to solve the above rate equations and compute the transition probabilities, it is necessary to construct estimators for the various properties of the radiation field that appear in said equations. This is done by converting integrals over the radiation field into summations over  $r$ -packets passing through a cell. This represents the stochastic nature of a MC simulation, and is by no means unique to the macro-atom formalism.

The first step is to apply the energy-density argument of [Lucy \(1999\)](#), which gives, for a time-independent code

$$J_\nu d\nu = \frac{1}{4\pi} \frac{1}{V} \sum_{d\nu} w_i \Delta s. \quad (1.61)$$

where the summations is over all photons between  $(\nu, \nu + d\nu)$ . This allows us to formulate estimators in a MC sense, rather than in integral form.

### 1.3.3.1 Bound-free estimators

The estimator for the photoionization rate is

$$\gamma_{j\kappa} = \frac{1}{V} \sum_i^{photons} \frac{w_i \sigma_{j\kappa}(\nu)}{h\nu} \Delta s \quad (1.62)$$

and for the stimulated recombination rate is

$$\alpha_{\kappa j}^{st} = \Phi_{j\kappa}^* \frac{1}{V} \sum_i^{photons} \frac{w_i \sigma_{j\kappa}(\nu)}{h\nu} \exp(-h\nu/kT_e) \Delta s, \quad (1.63)$$

where  $\sigma_{l\kappa}(\nu)$  is the photoionization cross-section for this transition. For bound-free transitions, we define the normal photoionization and recombination rate coefficients  $\gamma$  and  $\alpha$ , where  $\alpha$  includes stimulated recombination as we do in the code. Note this differs to the approach in [Lucy \(2003\)](#), where it is instead included as a negative photoionization term, hence the notation  $\tilde{\gamma}$ . We also need to define two ‘modified rate coefficients’ which are the rates at which b-f transitions add and remove energy to the radiation field. These are denoted  $\gamma^E$  and  $\alpha^E$ , and given by

$$\gamma_{j\kappa}^E = \frac{1}{V} \sum_i^{photons} \frac{w_i \sigma_{\nu_0}}{h\nu} \Delta s \quad (1.64)$$

$$\alpha_{\kappa j}^E = 4\pi \Phi_{i\kappa}^* \int_{\nu_{\kappa j}}^{\infty} \frac{\sigma_{\kappa j}(\nu)}{h\nu_{\kappa j}} \frac{2h\nu^3}{c^2} \exp\left(\frac{-h\nu}{kT_e}\right) d\nu \quad (1.65)$$

$$\alpha_{\kappa j}^{st,E} = \Phi_{j\kappa}^* \frac{1}{V} \sum_i^{photons} \frac{w_i \sigma_{j\kappa}(\nu)}{h\nu} \exp(-h\nu/kT_e) \Delta s \quad (1.66)$$

The rate at which recombinations convert thermal *and* ionization energy into radiant energy is then  $\alpha^E h\nu_{\kappa l} n_{\kappa} n_e$ , where  $h\nu_{\kappa l}$  is the potential of the b-f transition, or the energy difference between continuum  $\kappa$  and the level  $l$  we are recombining too. The amount of this energy which is removed from the actual thermal pool therefore needs a quantity  $\alpha h\nu_{\kappa l} n_{\kappa} n_e$  subtracted from it, giving

$$\mathcal{C}_{bf} = \sum_{bfjumps} (\alpha^E - \alpha) n_e n_{\kappa} \nu_{\kappa l} V \quad (1.67)$$

where here I have also included stimulated recombination as we do in the code. Note this differs to the approach in Lucy (2003), where it is instead included as a negative photoionization term, hence the notation  $\tilde{\gamma}$ . For photoionizations, we write a similar expression. The rate of at which a level  $l$  absorbs energy by b-f transitions is given by  $\gamma^E h\nu_{\kappa l} n_{\kappa} n_e$ , but the amount  $\gamma h\nu_{\kappa l} n_l$  goes into ionization energy, giving

$$\mathcal{H}_{bf} = \sum_{bfjumps} (\gamma^E - \gamma) n_l h\nu_{\kappa l} V \quad (1.68)$$

as the rate at which radiant energy heats the plasma via b-f transitions.

### 1.3.3.2 Bound-bound estimators

The heating and cooling rates for macro-atom bound-bound transitions are the rates of collisional excitations and de-excitations - i.e. the rate at which thermal energy is converted into bound-bound excitation energy and vice versa.

$$\mathcal{C}_{bb} = \sum_{lines} q_{lu} n_l n_e h\nu_{ul} V \quad (1.69)$$

$$\mathcal{H}_{bb} = \sum_{lines} q_{ul} n_u n_e h\nu_{ul} V \quad (1.70)$$

### 1.3.3.3 Other heating and cooling estimators

Although we have now formulated the estimators required to calculate the transition probabilities and level populations, there are still a number of heating and cooling mechanisms that do not involve macro-atoms.

The free-free cooling estimator is calculated from the emission coefficient in equation 1.33

$$\mathcal{C}_{ff} = V \sum_i^{ions} \int \frac{j_{ff,i}(\nu)}{4\pi} d\nu, \quad (1.71)$$

where the integral is over all frequencies included in the simulation, and the sum is also over all ions included in the simulation.

$$\mathcal{H}_{ff} = \sum w_i \kappa_{ff} \Delta s \quad (1.72)$$

Compton heating and cooling is included in the thermal balance and as  $(r \rightarrow k)$  and  $(k \rightarrow r)$  transitions.

$$\mathcal{C}_{comp} = 16\pi\sigma_T V J \frac{kT_e}{m_e c^2}, \quad (1.73)$$

$$\mathcal{H}_{comp} = n_e \sum \frac{h\nu}{m_e c^2} w_i \kappa_C \Delta s, \quad (1.74)$$

and induced Compton heating is then given by (Ferland et al. 2013)

$$\mathcal{H}_{ind\ comp} = n_e \sum \frac{J_\nu c^2}{h\nu^3} \frac{h\nu}{m_e c^2} w_i \kappa_C \Delta s, \quad (1.75)$$

where the first  $J_\nu c^2/h\nu^3$  term represents the ‘photon occupation number’. The adiabatic cooling rate is derived from  $PdV$  work and is given by

$$\mathcal{C}_a = kT_e V (\nabla \cdot v) \left( n_e + \sum_{i=1}^{ions} N_i \right), \quad (1.76)$$

where  $\nabla \cdot v$  is the divergence of the velocity field at the centre of the cell. The sum is over all ions included in the simulation.

### 1.3.4 $k$ -packets

$k$ -packets represent quantised kinetic or thermal energy, and any interaction chain involving a  $k$ -packet thus represents interaction with the thermal pool of ions and electrons.  $k$ -packets can be produced either directly via a continuum heating process ( $r \rightarrow k$ ), or by the collisional de-activation of a macro-atom ( $r \dots \rightarrow A^* \rightarrow k$ ) according to equation 1.47.

Once they are produced,  $k$ -packets never move, as they represent the quantised thermal energy flow in a finite volume element. Hence, when they are produced, their destruction path is decided according to the different cooling mechanisms in the plasma. A  $k$ -packet then has a probability of being destroyed by process  $i$  of

$$p_{i,destruct} = C_i / (C_{bf} + C_{ff} + C_{bb} + C_{comp} + C_a). \quad (1.77)$$

Note that only adiabatic cooling leads to an actual destruction of the energy packet, as a departure from radiative equilibrium. All other processes will lead to the creation of an active macro-atom or an  $r$ -packet.

### 1.3.5 Putting it all together

I have now defined all quantities needed to write down the transition probabilities in a macro-atom simulation. Fig. 1.4 shows the decision tree traversed by an energy packet in the simulation, showing the complicated set of interactions it can undergo each time it scatters.

### 1.3.6 Ionization Fractions and Level Populations

In section 1.1.2 I described how it is possible to calculate the ionization and excitation of a plasma under LTE or dilute approximations. Macro-atoms are not approximated – their level and ion populations are calculated by solving the rate equations formulated in section ???. This is done via matrix inversion. For an element with  $n$  ions and  $m_i$  levels in each ion, we construct a square matrix with dimensions  $m = \sum_i^n m_i$ . This element then has a total number density of  $N_{elem} = \sum_i^n N_i$ . To turn the system of rate equations for this element into matrix form, we populate the  $j$ th diagonal of the matrix with the negative of the rate out of level  $j$ ,  $-(\mathcal{R}_{j\ell'} + \mathcal{R}_{ju'})$ , and populate the off-diagonals  $(j, k)$  with the positive rate  $\mathcal{R}_{jk}$ . These are then multiplied by a vector of the fractional level populations, and must equal a vector of zeros, due to statistical equilibrium. Our matrix

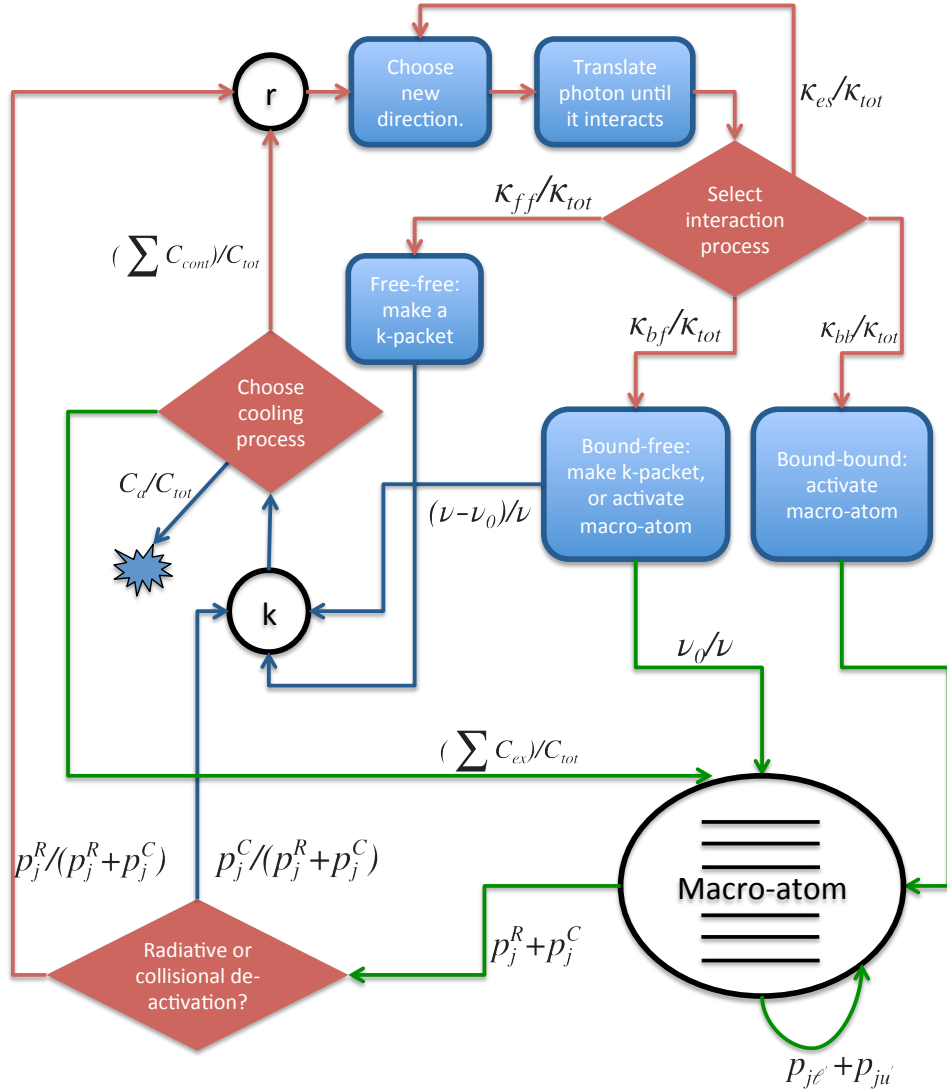


FIGURE 1.4: The decision tree traversed by an energy packet in macro-atom mode, depicting the interaction between radiation ( $r$ -packets), the thermal pool ( $k$ -packets), and ionization and ionization/excitation energy (macro-atoms). The probabilities at each decision point are marked, and are defined in the text. The red, blue and green coloured arrows represent radiant, kinetic and ionization/excitation energy respectively. The symbols are defined in the text, except  $C_{cont}$  and  $C_{ex}$  which refer to cooling contributions to radiative and excitation energy respectively.

equation is then

$$\begin{bmatrix} -\mathcal{R}_{1u'} & \mathcal{R}_{21} & \mathcal{R}_{31} & \dots & \mathcal{R}_{m1} \\ \mathcal{R}_{12} & -(\mathcal{R}_{2\ell'} + \mathcal{R}_{2u'}) & \mathcal{R}_{32} & \dots & \mathcal{R}_{m2} \\ \mathcal{R}_{13} & \mathcal{R}_{23} & -(\mathcal{R}_{3\ell'} + \mathcal{R}_{3u'}) & \dots & \mathcal{R}_{m3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathcal{R}_{1m} & \mathcal{R}_{2m} & \mathcal{R}_{3m} & \dots & -\mathcal{R}_{m\ell'} \end{bmatrix} \begin{bmatrix} n_1/N_{elem} \\ n_2/N_{elem} \\ n_3/N_{elem} \\ \vdots \\ n_m/N_{elem} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \quad (1.78)$$

This problem is not yet soluble, as a valid solution is that all levels could simply have occupation numbers of 0. To close the problem, we must impose the boundary condition, that the sum of the fractional populations is 1, i.e.

$$\sum_i \frac{N_i}{N_{elem}} = 1. \quad (1.79)$$

In matrix form, this is equivalent to replacing the entire first row of the rate matrix with 1, and the first entry of the RHS vector with a 1, so that we have

$$\begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ \mathcal{R}_{12} & -(\mathcal{R}_{2\ell'} + \mathcal{R}_{2u'}) & \mathcal{R}_{32} & \dots & \mathcal{R}_{m2} \\ \mathcal{R}_{13} & \mathcal{R}_{23} & -(\mathcal{R}_{3\ell'} + \mathcal{R}_{3u'}) & \dots & \mathcal{R}_{m3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathcal{R}_{1m} & \mathcal{R}_{2m} & \mathcal{R}_{3m} & \dots & -\mathcal{R}_{m\ell'} \end{bmatrix} \begin{bmatrix} n_1 \\ n_2 \\ n_3 \\ \vdots \\ n_m \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (1.80)$$

This matrix equation can now be solved. To do the actual matrix manipulation, the code uses the GNU scientific libraries (GSL; [Gough 2009](#)) implementation of LU decomposition ([Turing 1948](#)). This is a fast and reliable way of inverting large matrices that includes error handling and enables checking of, for example, singular rate matrices.

### 1.3.7 Numerical Issues and Population Inversions

## 1.4 A hybrid line transfer scheme: including simple-atoms

I have now described in detail how the macro-atom approach is implemented in PYTHON. A pure macro-atom approach can be easily used for some situations – for example, in the YSO application described by [Sim et al. \(2005\)](#), which uses a H-only model. However,



in accretion disc winds the densities can be very high and higher  $Z$  elements must be included. To include all these elements as macro-atoms is not currently computationally feasible in PYTHON for anything but the simplest models. I will thus describe a ‘hybrid scheme’, which treats H and He under the macro-atom approach but models all other atoms as ‘simple-atoms’.

### 1.4.1 Line Transfer

Simple-atoms still interact with  $r$ - and  $k$ -packets, but do not possess internal transition probabilities. As a result, they are analogous to the two-level atom treatment, as any excitation is immediately followed by a deactivation into an  $r$ - or  $k$ -packet. This hybrid approach allows us to preserve the fast treatment of, for example, UV resonance lines, while accurately modelling the recombination cascades that populate the levels responsible for H and He line emission. As a result of this hybrid scheme, a separate set of estimators must be recorded for simple-atoms, and the ionization and excitation of these elements is calculated with a different, approximate approach.

In order to include simple-atoms, we must add in a few extra pathways to Fig. 1.4, so that energy packets can also undergo excite simple-atoms, through either bound-free or bound-bound processes. This is done in proportion with the simple-atom opacities.

This approach does necessitate a few approximations.

### 1.4.2 Heating and Cooling Estimators

In simple-ions it is in some ways a little more complicated. First we define  $q$  which will be different for each b-b transition, following Nick’s thesis, which is given by (NB: I don’t actually know how to derive this)

$$q = \frac{q_{ul}n_e(1 - e^{-h\nu/kT_e})}{\beta_{ul}A_{ul} + q_{ul}n_e(1 - e^{-h\nu/kT_e})} \quad (1.81)$$

where  $\beta_{ul}$  is the angle-averaged escape probability.  $q$  represents *the probability that an excited bound electron will collisionally de-excite*. Our b-b heating rate is computed during the photon propagation and is a sum over photons which come into resonance

with each line, given by

$$H_{bb,simple} = \sum_{photons} \sum_{lines} (1 - q)(1 - e^{-\tau_S}) w_{photon} \quad (1.82)$$

And our bound bound cooling rate is given by

$$C_{bb,simple} = \sum_{lines} q \left( n_l \frac{g_u}{g_l} - n_u \right) q_{ul} n_e \frac{(1 - e^{-h\nu/kT_e})}{(e^{h\nu/kT_e} - 1)} h\nu_{ul} \quad (1.83)$$

The bound-free heating rate is given by

$$H_{bf,simple} = \sum_{photons} \sum_{bfjumps} w_{photon} e^{-\tau} \frac{\nu - \nu_0}{\nu} \quad (1.84)$$

where  $\nu$  here is the frequency of the photon in question, and  $\nu_0$ . The bound-free cooling rate is then

$$C_{bf,simple} = ?? \quad (1.85)$$

#### 1.4.2.1 Radiation Field Estimators

For simple-atoms we do not record radiation field estimators for discrete transitions, as for macro-atoms. Instead, we record estimators to give us a model of the radiation field. The estimators needed depend on the ionization mode employed (see section 1.4.3). The radiation temperature,  $T_R$ , is estimated by recorded the mean frequency,  $\bar{\nu}$ , of the photons passing through a cell:

$$\bar{\nu} = \frac{\sum_{photons} w_i \nu_i \Delta s}{\sum_{photons} w_i \Delta s}. \quad (1.86)$$

This is then used to calculate the radiation temperature by considering the value expected from a blackbody [Mazzali & Lucy \(1993\)](#), giving

$$T_r = \frac{h\bar{\nu}}{3.832 k} \quad (1.87)$$

The dilution factor can be calculated by comparing the estimator for the mean intensity to the Stefan-Boltzmann law:

$$W = \frac{\pi J}{\sigma T_r^4}. \quad (1.88)$$

This set of estimators is sufficient to describe the radiation field if one is adopting the dilute approximation (section 1.1.2.1). Higginbottom et al. (2013) improved on this by developing a method for modelling the SED in the cell using a series of band-limited radiation field estimators. In this scheme, a series of bands is defined in which to record these estimators. These bands are different to those discussed in section 1.2.2 as those instead govern photon generation. In H13, the band limited estimators were used to construct a correction factor that could be used in a modified Saha equation (similar in form to equation 1.13). However, the code has now been improved so that the ion populations are computed by solving the rate equations. Thus, we now simply need to calculate photoionization rate estimators for simple ions, which rely on being able to integrate a modelled form of the mean intensity.

### 1.4.3 Ionization and Excitation

Regardless of ion mode, the relative excitation fractions of simple-atoms within each ionization stage of a given species are estimated via a modified (dilute) Boltzmann equation (equation 1.14). This equation is approximate, and in general this approximation is not good. We therefore endeavour to treat any species in which the excitation state of the ions is thought to be important in determining either the ionizing radiation field, or emergent spectrum, as macro-atoms.

## 1.5 Heating And Cooling Balance

I have already given the estimators used to calculate heating and cooling rates in the plasma. These are not only used in the creation and elimination of  $k$ -packets, but also in the heating and cooling balance carried out in PYTHON to achieve a self-consistent temperature structure in the wind.

At the end of each ionization cycle, the code has stored a new set of MC estimators for radiative heating of the plasma. We then assume that each cell is in thermal equilibrium then the appropriate electron temperature is simply the value of  $T_e$  that is a solution to the equation

$$\mathcal{H}_{tot} - \mathcal{C}_{tot}(T_e) = 0, \quad (1.89)$$

where  $\mathcal{H}_{tot}$  and  $\mathcal{C}_{tot}$  are the total heating and cooling rates in the plasma. A number of checks are in place to ensure numerical stability, namely a maximum temperature and a maximum change in temperature from cycle to cycle. This is especially important in cases where the initial guess at wind temperature is far from the true value.

### 1.5.1 Convergence

PYTHON always runs a fixed number of ionization cycles, rather than terminating when a convergence criterion is reached. As a result, it is up to the user to check that the simulation is converged. An individual cell is considered converged when a) the temperature stops changing significantly, i.e. both  $T_R$  and  $T_e$  satisfy

$$\frac{|T_{new} - T_{old}|}{T_{new} + T_{old}} < 0.05, \quad (1.90)$$

and b) the heating and cooling rates are well balanced such that

$$\frac{|\mathcal{H}_{tot} - \mathcal{C}_{tot}|}{\mathcal{H}_{tot} + \mathcal{C}_{tot}} < 0.05. \quad (1.91)$$

These criteria could doubtless be improved, but they are nonetheless a good way of ensuring that thermal and radiative equilibrium holds in the plasma. An example of how the average temperature and fraction of converged cells changes over the course of the ionization cycles in a typical CV model is shown in Fig. 1.5.

## 1.6 Spectral Cycles

The primary output from PYTHON is a synthetic spectrum over a specific wavelength range produced at user-specified viewing angles. This spectrum is produced in a separate cycle from the calculation of the ionization state as it is concerned with producing detailed, high-resolution spectra in a specific wavelength regime that can then be compared to observations.

The code utilises a variance reduction technique in order to minimise the amount of time spent in the portion of the code. This technique is based on a similar method implemented by (Woods 1991) and is known in the code as the ‘extract’ method. This

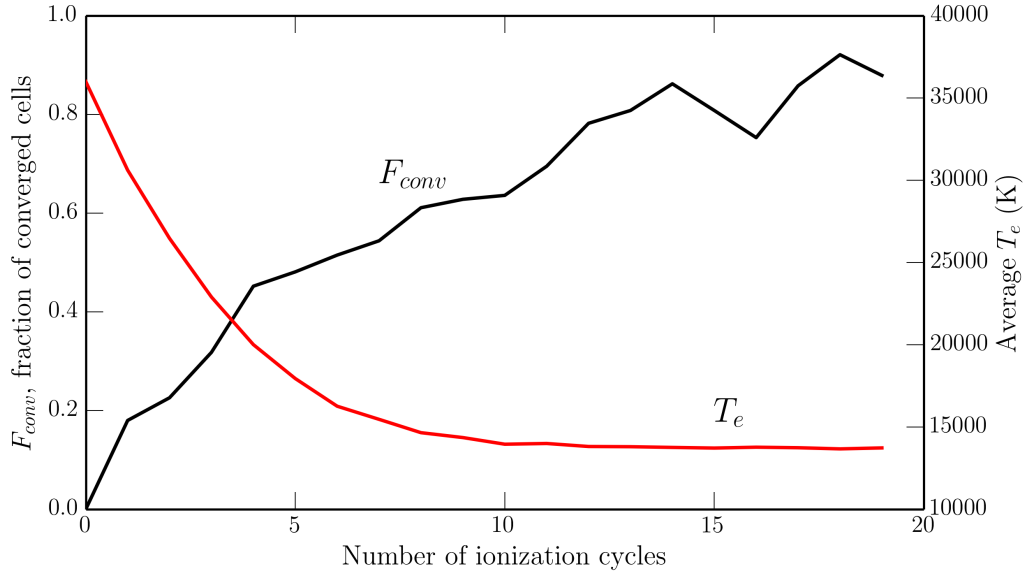


FIGURE 1.5: The average temperature and fraction of converged cells in a typical CV model, shown as a function of the number of ionization cycles completed.

method works by tracking photon packets until they scatter or interact with the plasma, according to the procedure described in section 1.2.3. At the scattering location, the optical depth the photon would experience were it to escape to infinity along each requested viewing angle,  $\tau_{extract}(\theta_i)$ , is calculated. The spectrum at each viewing angle  $\theta_i$  is then incremented by an amount

$$\Delta L = w f(\theta_i) \exp(-\tau_{extract}(\theta_i)), \quad (1.92)$$

where  $w$  is the weight of the photon and  $f(\theta_i)$  is a weighting proportional to the probability that the photon would have scattered in direction  $\theta_i$ . Once this value has been added to the corresponding wavelength bin then the photon proceeds as normal in its new random direction.

In the alternative ‘live or die’ method this extraction procedure is not carried out and a user simply has to run enough escaping photons so that enough will happen to fall into the angle bins requested. This is clearly significantly less efficient. A comparison between the two methods is shown in figure 1.6 for a standard CV model, showing that the spectrum produced is identical in shape but with significantly higher signal-to-noise (for fixed number of cycles) in the ‘extract’ case.

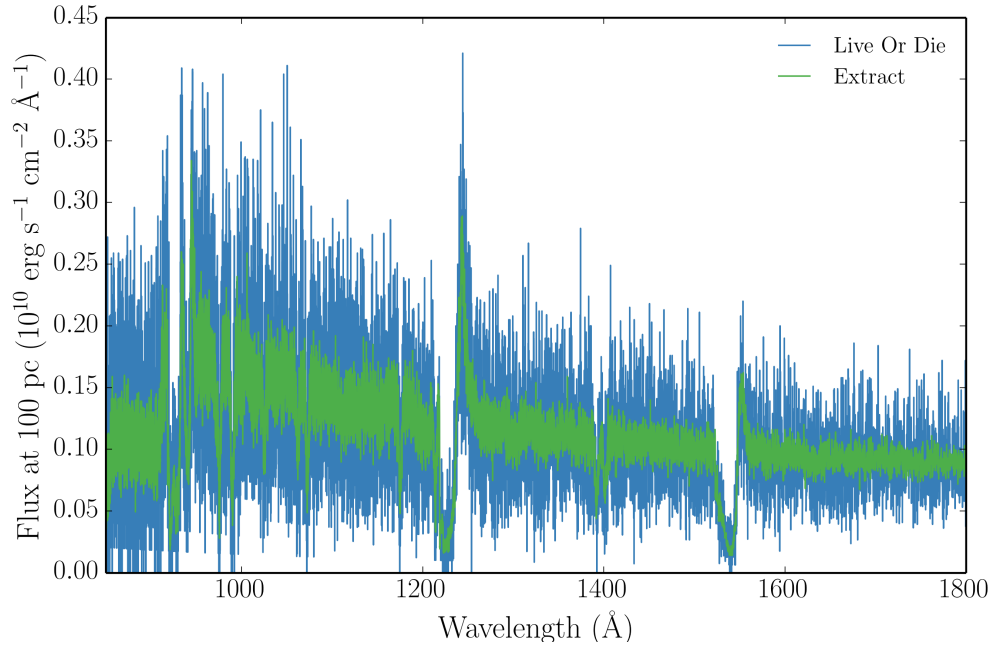


FIGURE 1.6: A synthetic spectrum after 30 spectral cycles with 100,000 photons from simple CV wind model at a  $60^\circ$  viewing angle. Spectra produced with both the extract and live or die modes are shown. The effectiveness of the extract variance reduction technique can be clearly seen, and we can see that the spectral shape is unaltered.

### 1.6.1 Macro-atom Emissivity Calculation

In order to preserve the philosophy that a detailed spectrum is calculated in a limited wavelength regime, PYTHON carries out a macro-atom emissivity calculation before the spectral cycles. This can be very computationally intensive, especially if the wavelength regime being simulated has very little emission from bound-free and line processes in the wind, but the overall broadband emissivity is high.

During the ionization cycles, the amount of energy absorbed into  $k$ -packets and every macro-atom level is recorded using MC estimators. Once the ionization cycles are finished and the model has converged, these absorption energies are split into a certain number of packets and tracked through the macro-atom machinery until a deactivation occurs. When this happens, the emissivity of the level the macro-atom de-activated from is incremented if the packet lies in the requested wavelength range. If it does not, then the packet is thrown away. It is easy to see how what is essentially a MC rejection method can be an inefficient way of sampling this parameter space. Fortunately, this problem is parallelisable (see section 1.9.1).

Once the emissivities have been calculated then the spectral synthesis can proceed. This is done in a different way to the ionization cycles. Photons are generated from the specified photon sources over the required wavelength range, but are now also generated according to the calculated macro-atom and  $k$ -packet emissivities in each cell. These photons are ‘extracted’ according to the procedure outlined above. To now ensure that radiative equilibrium holds, any photon that interacts with a macro-atom or  $k$ -packet is immediately destroyed. The photons are tracked and extracted as normal until they escape the simulation, and resonant scatters are dealt with by a combination of macro-atom photon production and destruction.

## 1.7 Atomic Data

One of the big challenges in building reliable photoionization and radiative transfer lies in the acquisition of accurate and complete atomic datasets. All of the rates described so far contain a term, such as the oscillator strength or dimensionless collision strength, that is dependent purely on the atomic physics associated with the transition. These quantities can be measured in laboratory experiments, or predicted from atomic structure codes which derive the atomic physics from quantum theory.

Photoionization cross-sections are obtained from two sources. Where possible, we use TOPBASE photoionization cross-sections. For macro-atoms, these cross-sections are partial and represent the cross-section for a photoionization from a given *level*. We neglect photoionizations to excited configurations of the upper ion. For simple-atoms they are from the ground state. The TOPBASE cross-sections have two major drawbacks in that

In order to improve the TOPBASE cross-sections, I have extrapolated them to larger energies. This was done by finding the slope, in log-log space of the cross-section at the maximum energy, and extrapolating to 100 keV. In some cases, the slope near the maximum energy was anomalous due to resonances or similar structure in the cross-section, or possibly simply due to unknown problems in the TOPBASE calculations. These cases were identified by eye, and instead a  $\nu^{-3}$  extrapolation was applied. The results of this extrapolation on the H13 model are shown in figure 1.7. Where previously there was a sharp, unphysical edge, we now observe the typical recovery to an X-ray power law we expect.

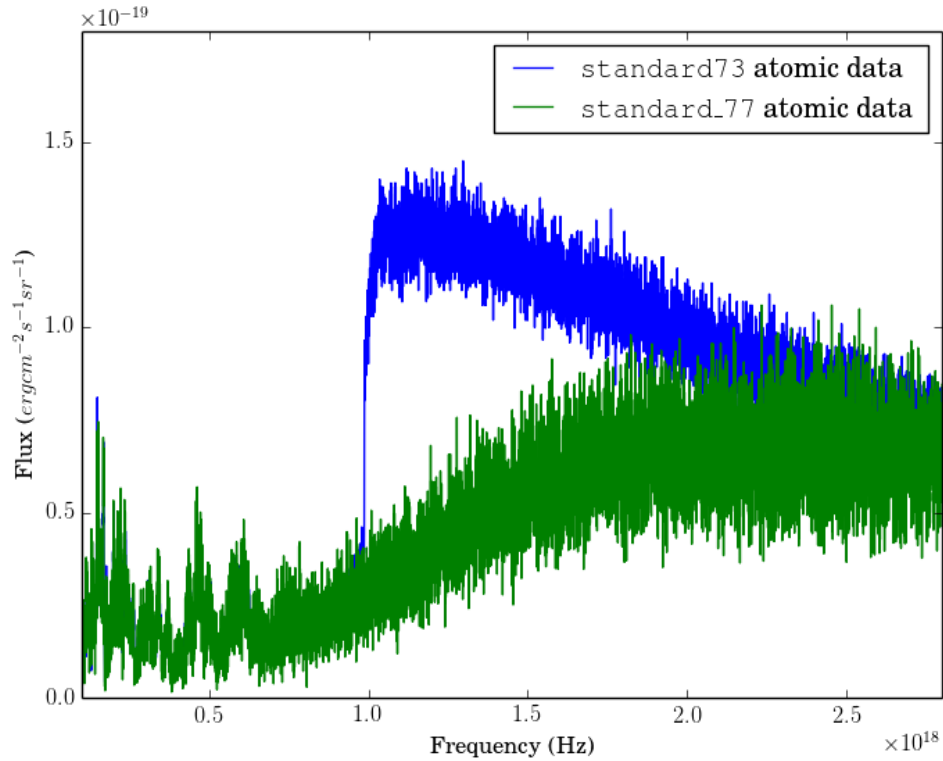


FIGURE 1.7: A comparison of the soft X-ray regime of the H13 model, with two different datasets. standard73 is the dataset with old, unextrapolated cross-sections and standard77 instead includes extrapolated cross-sections as described in the text.

## 1.8 Code Validation

The main challenge for high performance scientific computing can be elegantly summarised by Ferland’s (2002) epitaph, ‘*Reliability in the face of complexity*’. I have already delved into some of the complexity in this case, so it is important to assess whether the code is also reliable before I present results.

### 1.8.1 Testing against Cloudy

CLOUDY is a spectral synthesis and photoionization code used to simulate the emergent spectrum and ionization conditions in nebulae and other plasma environments. As a result, it uses many of the same techniques as PYTHON in order to compute ionization states, level populations and heating and cooling rates, and represents an excellent comparison tool. PYTHON has been tested extensively against CLOUDY in the past; some of these successful tests can be found in H13 and LK02.



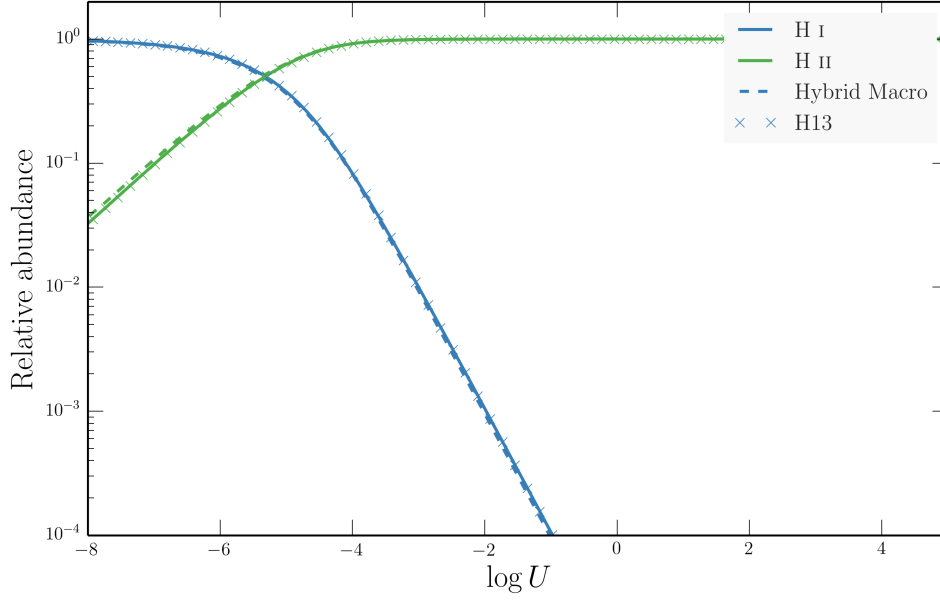


FIGURE 1.8: Relative ion fractions as a function of ionization parameter from the hybrid macro-atom scheme, with Hydrogen and Helium treated as a full macro-atom, compared to both CLOUDY and PYTHON in simple-atom only mode.

Figs. 1.8 to 1.12 show a series of ionization plots with relative ion fractions plotted as a function of ionization parameter,  $U$ . This test is designed to check that PYTHON still agrees well with CLOUDY when we turn on the full macro-atom machinery. The calculations are conducted using the same incident SEDs, densities and abundances, and PYTHON is operated in 1D, thin shell mode to simulate an optically thin plasma and facilitate comparison with CLOUDY. I have shown two separate ionization modes from PYTHON: standard mode, in which nothing is treated as a macro-atom and the spectral model ionization scheme of H13 is used to calculate ion fractions, and hybrid mode in which H and He are treated as macro-atoms and their level populations and ion fractions are solved using MC estimators according to the routine described in section 1.3.6. Thus, in both cases the simple-atoms have their populations calculated using the H13 scheme. In general, the calculated fractions are in excellent agreement, with a few exceptions.

### 1.8.2 Macro-atom testing against Tardis and theory

TARDIS is a 1D photoionization and radiative transfer designed to model SNe in a quick and easy python package, and is described in detail by Kerzendorf & Sim (2014).

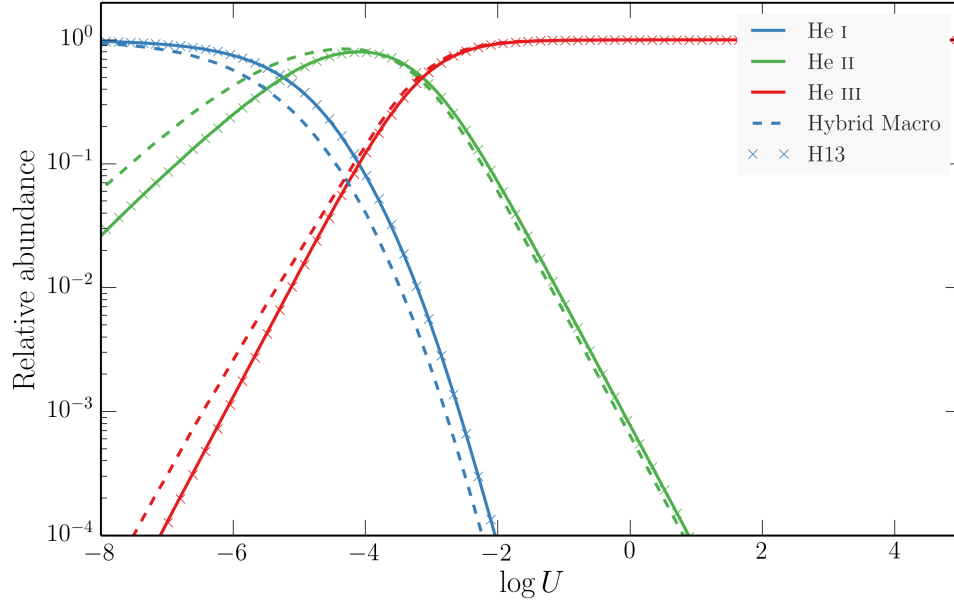


FIGURE 1.9: As figure 1.8, but for Helium.

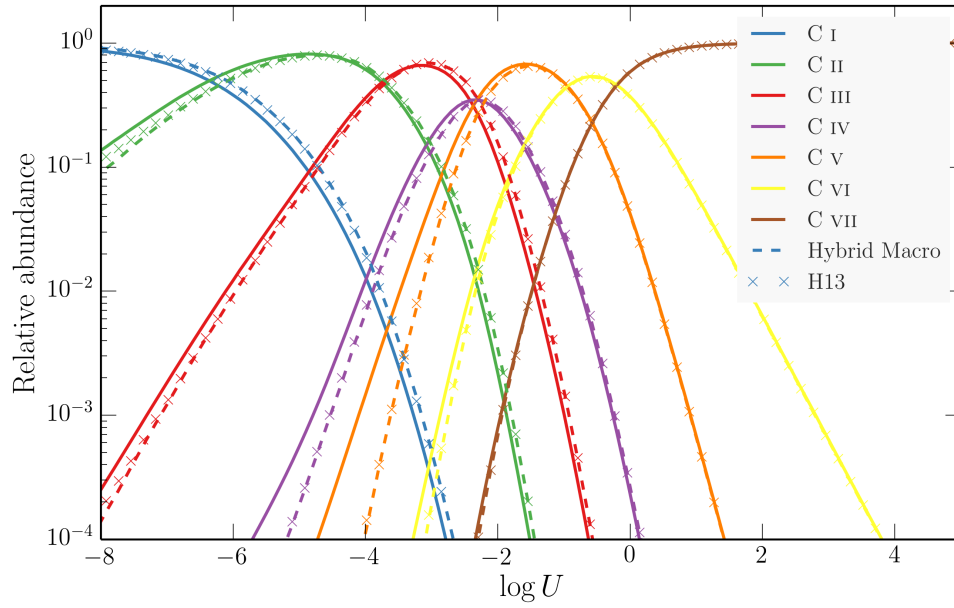


FIGURE 1.10: As figure 1.8, but for Carbon.

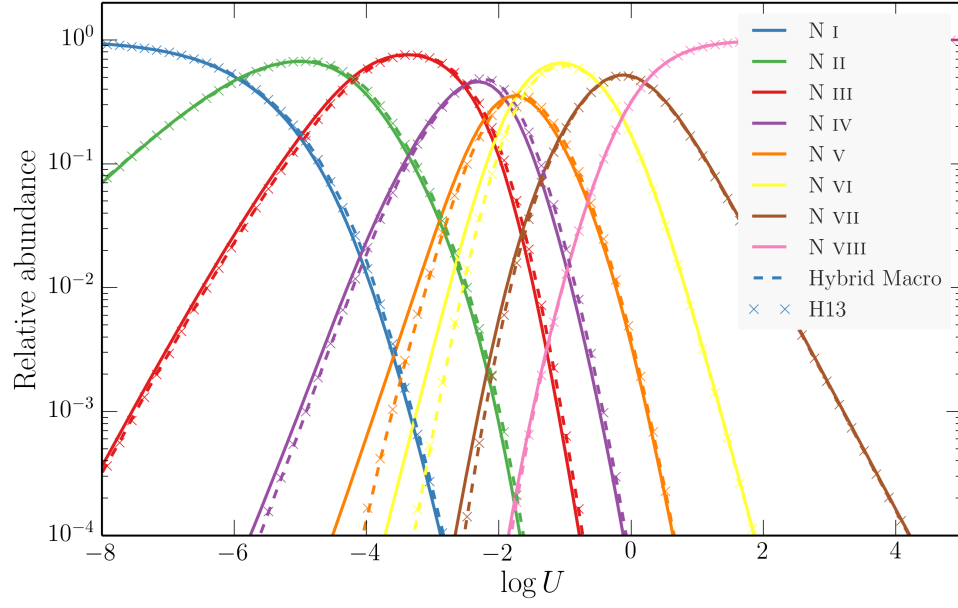


FIGURE 1.11: As figure 1.8, but for Nitrogen.

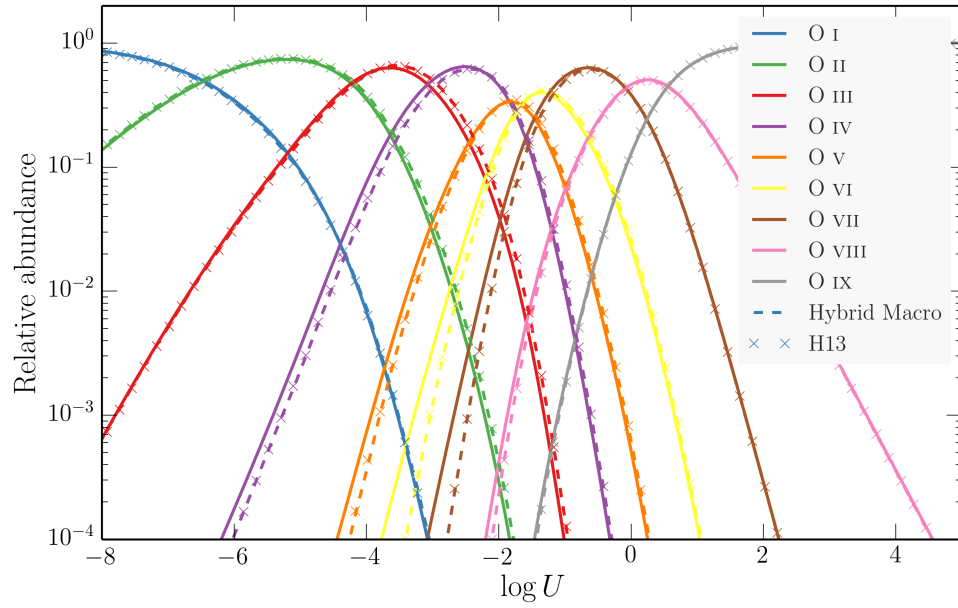


FIGURE 1.12: As figure 1.8, but for Oxygen.

Although TARDIS is simpler in terms of geometry, it has many of the same capabilities of PYTHON and thus makes for an excellent comparison.

Fig. 1.13 shows the results of two code tests. In the top panel, I show a comparison of the Balmer series emissivities as predicted by PYTHON in the l-mixed Case B limit against the analytical calculations by [Seaton \(1959\)](#). Both calculations are calculated at  $T_e = 10,000\text{K}$ . Case B is an approximation commonly used in nebular astrophysics (see e.g. [Osterbrock 1989](#)) in which one assumes that all line transitions are optically thin, except for the  $\text{Ly}\alpha$  transition, which is taken as optically thick. Thus, this test comparison is carried out using a thin shell of plasma in which the escape probabilities,  $\beta_{ul}$  are artificially set to 1 in all transitions except  $\text{Ly}\alpha$ , which has its  $\beta_{ul}$  set to 0.

The bottom panel shows a comparison of He I level populations (the most complex ion currently treated as a macro-atom) between PYTHON and TARDIS models. The calculation is conducted with physical parameters of  $n_e = 5.96 \times 10^4 \text{ cm}^{-3}$ ,  $T_e = 30,600\text{K}$ ,  $T_R = 43,482\text{K}$  and  $W = 9.65 \times 10^{-5}$ . Considering the two codes use different atomic data and TARDIS, unlike PYTHON, currently has a complete treatment of collisions between radiatively forbidden transitions, the factor of  $< 2$  agreement is encouraging.

Fig. 1.14 shows a comparison between TARDIS and PYTHON synthetic spectra from a simple 1D SN model. This comparison was originally presented by [Kerzendorf & Sim \(2014\)](#), but I have since then discovered a bug in the Doppler shifting routine in PYTHON, introduced around PYTHON 76, which was present in this test. Fixing this issue leads to slightly better agreement between the two codes. The model involves a full computation of the ionization state in the ML93 mode, and, although run in 1D, still tests most of the radiative transfer machinery of the code. The spectra are in good agreement, considering there are differences in their excitation treatments and atomic data. This comparison is particular encouraging when we consider that [Kerzendorf & Sim \(2014\)](#) also show comparisons with other SN codes such as ARTIS ([Kromer & Sim 2009](#)).

### 1.8.3 Testing line transfer modes

Include a test of simple ion v standard mode...

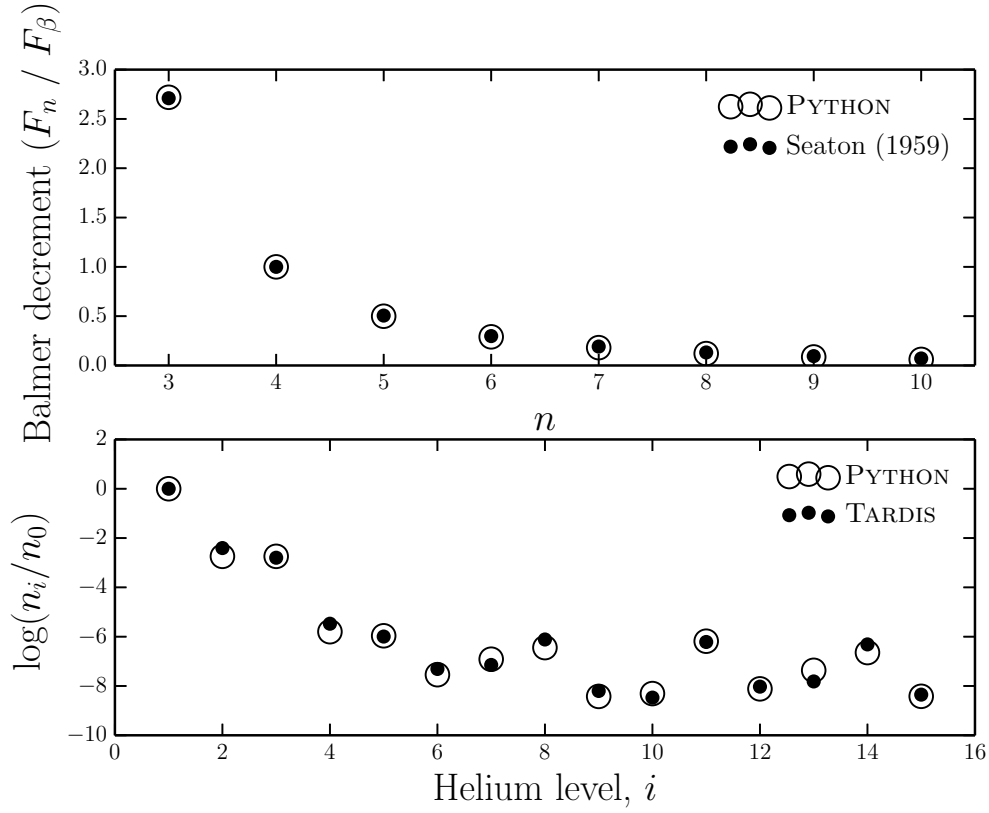


FIGURE 1.13: *Top Panel:* ‘Case B’ Balmer decrements computed with PYTHON compared to analytic calculations by Seaton (1959). Both calculations are calculated at  $T_e = 10,000\text{K}$ . *Bottom Panel:* a comparison of He I level populations (the most complex ion we currently treat as a macro-atom) between PYTHON and TARDIS models. The calculation is conducted in thin shell mode with physical parameters of  $n_e = 5.96 \times 10^4 \text{ cm}^{-3}$ ,  $T_e = 30,600\text{K}$ ,  $T_R = 43,482\text{K}$  and  $W = 9.65 \times 10^{-5}$ .

## 1.9 Code Maintenance and Version Control

As part of the expansion of the team working on PYTHON I was responsible for bringing the code under the auspices of a robust version control system. Thanks to these efforts, the code is now hosted on GitHub at <https://github.com/agnwinds/python/>. Our team uses a Pull & Fork model for collaborative code development, in which major changes are made in a forked repository before the developer submits a ‘Pull request’ to the main repository. To test the code, we use a combination of Travis CI build tests – run per commit to the upstream repo – and our own test suite which is run every night on a multi-core server.

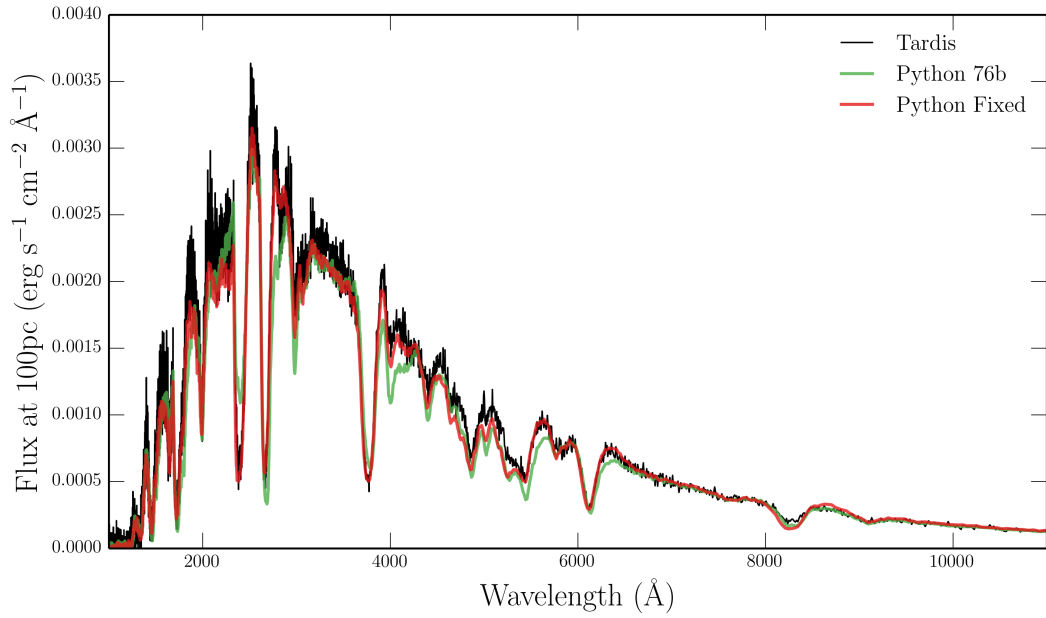


FIGURE 1.14: Comparison between TARDIS and PYTHON synthetic spectra from a simple 1D supernova model. A bug in Doppler shifting of photons was discovered around PYTHON 76, meaning that the code now gives even better agreement than presented in Kerzendorf & Sim 2014.

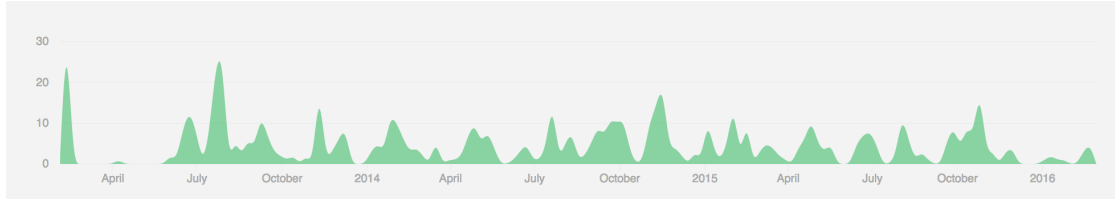


FIGURE 1.15: Commit history from Feb 3, 2013 to Feb 29, 2016, showing the regular code development that makes version control such a necessity to a collaborative code project. Produced using the Github API and plotting capability.

### 1.9.1 Parallelisation

Including macro-atoms in a simulation can have a significant impact on runtime, especially when simulating dense regions of plasma. By way of example, the CV model presented by LK02 takes approximately 118s to run one ionization cycle with  $10^6$  photons. One of the macro-atom models presented in chapter 4 takes ??s to complete the same task.

Fortunately, MCRT codes are intuitively parallelisable, as is the macro-atom emissivity calculation described above, as operations on cells or photons can simply be divided up between processors. PYTHON is parallelised using an open source Message Passing Interface (MPI) implementation known as Open MPI ([Gabriel et al. 2004](#)). This library

provides the core functions needed in order to share out computing tasks among a series of parallel processors with distributed memory. The parallelised elements of PYTHON include the photon propagation, updating of wind ionization and temperature structure and calculation of macro-atom emissivities. As a result, this involves a reasonable amount of book-keeping in that the radiation field estimators must be communicated between threads so as to correctly account for all the photons that have interacted with a given cell. I have been responsible for all of the parallelisation implemented that is specific to the macro-atom routines.

Fig. ?? shows the effect of parallelising a run. Due to the nature of MCRT, it is possible to achieve a roughly linear decrease in computing time for the photon propagation, which is the dominant factor in overall runtime. This improvement was crucial in order to be able to run the simulation grids used for chapters 4 and 5.

# Bibliography

Abbott D. C., Lucy L. B., 1985, ApJ 288, 679

Ferland G. J., Porter R. L., van Hoof P. A. M., Williams R. J. R. et al., 2013, RMXAA 49, 137

Gabriel E., Fagg G. E., Bosilca G., Angskun T. et al., 2004, in Proceedings, 11th European PVM/MPI Users' Group Meeting, Budapest, Hungary, p. 97

Gayet R., 1970, A&A 9, 312

Gough B., 2009, GNU Scientific Library Reference Manual - Third Edition, Network Theory Ltd., 3rd edition

Higginbottom N., Knigge C., Long K. S., Sim S. A., Matthews J. H., 2013, MNRAS 436, 1390

Higginbottom N., Proga D., Knigge C., Long K. S. et al., 2014, ApJ 789, 19

Humphrey A., Binette L., 2014, MNRAS 442, 753

Kerzendorf W. E., Sim S. A., 2014, MNRAS 440, 387

Klein O., Nishina T., 1929, Zeitschrift fur Physik 52, 853

Kromer M., Sim S. A., 2009, MNRAS 398, 1809

Long K. S., Knigge C., 2002, ApJ 579, 725

Lucy L. B., 1999, A&A 344, 282

Lucy L. B., 2002, A&A 384, 725

Lucy L. B., 2003, A&A 403, 261



- Matthews J. H., Knigge C., Long K. S., Sim S. A., Higginbottom N., 2015, MNRAS 450, 3331
- Matthews J. H., Knigge C., Long K. S., Sim S. A. et al., 2016, MNRAS
- Mazzali P. A., Lucy L. B., 1993, A&A 279, 447
- Mihalas D., 1978, Stellar atmospheres /2nd edition/
- Osterbrock D. E., 1989, Astrophysics of gaseous nebulae and active galactic nuclei
- Seaton M. J., 1959, MNRAS 119, 90
- Sim S. A., 2004, MNRAS 349, 899
- Sim S. A., Drew J. E., Long K. S., 2005, MNRAS 363, 615
- Sim S. A., Long K. S., Miller L., Turner T. J., 2008, MNRAS 388, 611
- Tatum M. M., Turner T. J., Sim S. A., Miller L. et al., 2012, ApJ 752, 94
- Turing A. M., 1948, QJMAM 1(1), 287
- van Regemorter H., 1962, ApJ 136, 906
- Woods J. A., 1991, Ph.D. thesis, D. Phil thesis, Univ. Oxford , (1991)