Saha-Boltzmann calibration of the Havard sequence

Maren Rasmussen [1]

[1] Institute of Theoretical Astrophysics, P.O. Box 1029, Blindern, N–0315 Oslo, Norway

Abstract. In this article we have studied the Saha and Boltzmann distributions for a fictional element called Schadeenium and for hydrogen and calcium to achieve more information about spectral lines and spectral line strengths. We have shown that the spectral lines have a close connection to temperature, because both ionization and excitation is closely connected to the temperature. We have also found that the Harvard sequence basically classifies stars after temperature.

1. Introduction

Stellar spectra gives a lot of information about the conditions in different stars. Absorption lines in the spectra appears if the atoms in the outer layers of the stars absorbs photons emitted from the star. Therefore the absorption lines can tell something about the content of elements in the star. A classification method called the Harvard spectral sequence is based on this information from the stellar spectra. Figure 1 shows some stellar spectra illustrating the Harvard spectral sequence for wavelengths between around 3800-5000 Å. The spectra can be compared to energy level diagrams for different elements to recognize which elements that have made the different absorption lines.

The energy level diagram for hydrogen is shown in figure 2. According to this diagram, the Balmer β line should absorb wavelengths of 4861 Å, and is the only line with a wavelength within the same range as the spectra in figure 1. At this wavelength in the spectra, there is a clearly, strong black line as expected for the five uppermost stars. The stars further down has no line at all for the Balmer β line, or just a very weak one. This article will among other discuss why this is the case.

This article also will study the Harvard spectral sequence and spectral lines. It will discuss the relationship between the population density of particles with different ionization level and the strength of the absorption lines in stellar spectra. The article will mainly focus on a fictional element called *Schadeenium*, but will also look at real elements.

1.1. The Boltzmann and Saha laws

The Boltzmann and Saha laws describe the population density of different ionization stages and over the discrete energy levels within each stage for a specific element. Saha's law describes how many particles of an element that occupies the different ionization stages (e.g. twice-ionized). Saha distribution is given by

$$\frac{N_{r+1}}{N_r} = \frac{1}{N_e} \frac{2U_{r+1}}{U_r} \left(\frac{2\pi m_e kT}{h^2}\right)^{3/2} e^{-\chi_r/kT}, \tag{1}$$

where T is the temperature, k is the Boltzmann constant, h is the Planck constant, N_e is the electron density, m_e is the electron mass, χ_r is the threshold energy needed to ionize a particle from stage r to $r+1$, and U_{r+1} and U_r are the partition functions of the the respective ionization stages. $N_r = \sum_s n_{r,s}$ is the total particle density in all levels of ionization stage r. At last, the partition function U_r is defined by

$$U_r \equiv \sum_s g_{r,s} e^{-\chi_{r,s}/kT} \tag{2}$$

The Boltzmann law describes the distribution of the different ionized particles over the discrete energy levels within each ionization (e.g. a neutral atom with one electron excited to the second level). The Boltzmann distribution is given by

$$\frac{n_{r,s}}{N_r} = \frac{g_{r,s}}{U_r} e^{-\chi_{r,s}/kT}. \tag{3}$$

Here r and s represent the level s of the ionization stage r, n is the number density in unit m^{-3}, g is the statistical weight and χ is the excitation energy for the given level measured from the ground state $(r, 1)$.

1.2. Schadeenium

To analyze how the absorption lines are connected to the population density of particles with different ionization stages and excitation levels, a simple fictional element is studied. The element we call Schadeenium and it will be referred to as E. Its ionization energies are $\chi_1 = 7$ eV for neutral stage, $\chi_2 = 16$ eV for the first ionized stage, $\chi_3 = 31$ eV for the second ionized stage and $\chi_4 = 51$ eV for the this ionized stage. The excitation energies is given by $\chi_{r,s} = s - 1$ eV. This means that there is 7 excitation levels in the neutral stage, 16 in the first ionized stage etc. For Schadeenium it is also assumed that the statistical weights is $g_{r,s} = 1$ for all levels (r, s).

2. Observations

Before analyzing the fictional element Schadeenium, an assumption needs to be made. From observations, it seems like the strength of the absorption lines scales linearly with the population density of the lower level of the corresponding transition. If the stellar outer layers have a lot of atoms with the lower level of the corresponding transition, it is natural to conclude that there would be a lot of atoms which could excite and absorb photons. This is proven to not be correct, although absorption lines do get stronger at larger lower-level population. In this report we will anyways assume that the scaling is linear. With this assumption, we can find initial rough estimates of the
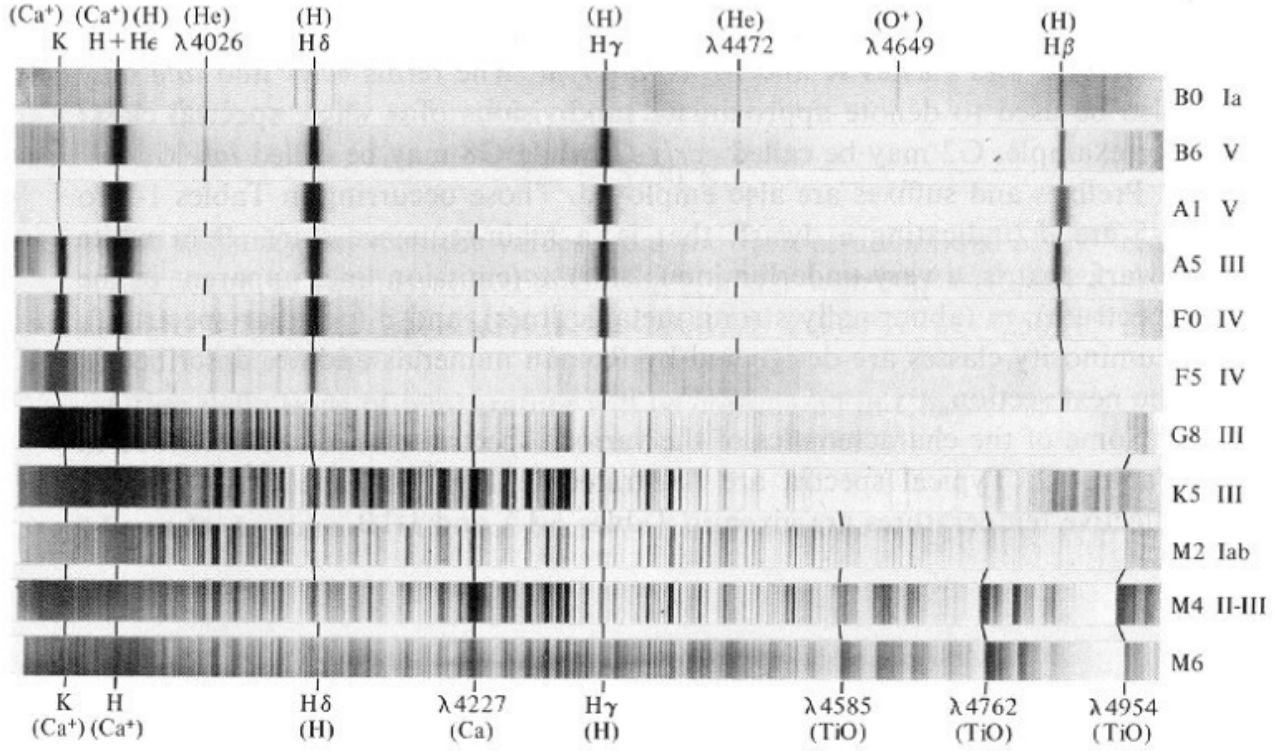
Fig. 1: Stellar spectra illustrating the Harvard spectral sequence. The dark lines represents the absorption lines for different wavelengths. The Harvard classification of the different stars, are written on the right. The stars are classified as either O, B, A, F, G, K or M, where the hottest stars are classified as O and the coolest as M. Novotny (1973)

strength ratios of different absorption lines.

Take a neutral hydrogen atom as an example. The ratio between the strengths for the different excitation lines for hydrogen can be found using equation 3. The statistical weight for a neutral hydrogen is given by $g = 2s^2$. If we then assume we have a temperature equal to the the Sun's surface temperature (T = 5777 K), we get a ration between the Lyman α and Balmer α line of

$$\frac{n_{1,1}}{n_{1,2}} = \frac{2e^0}{8e^{-10.20/kT}} \approx 2 \cdot 10^8.$$

The same calculation can be done for the Balmer α and Paschen α line strength ratio,

$$\frac{n_{1,2}}{n_{1,3}} = \frac{8e^{-10.20/kT}}{18e^{-12.09/kT}} \approx 20.$$

And for the Paschen α and Brackett α lines,

$$\frac{n_{1,2}}{n_{1,3}} = \frac{18e^{-12.09/kT}}{32e^{-12.75/kT}} \approx 2.$$

So for a sun-like star, the Lyman α line would be much stronger than the other lines.

3. Method

The fictional element Schadeenium and the two realistic elements hydrogen and calcium is studied by numerically calculating the Saha and Boltzmann distributions and the line strength

U_r	5000 K	10 000 K	20 000 K
U_1	1.11	1.46	2.23
U_2	1.11	1.46	2.27
U_3	1.11	1.46	2.27
U_4	1.11	1.46	2.27

Table 1: The partition function values for Schadeenium for different ionization stages and temperature.

ratio when making the assumption in the previous section. All the figures are made by simple implementations of the functions. The algorithms can be found in appendix C.

4. Results and discussion

4.1. Schadeenium

Table 1 shows the calculations of the partition function (equation 2) for Schadeenium for different ionization stages and temperature. It shows that the partition function is almost invariant to temperature and is almost constant for different ionization stages. It is also worth noting that the partition function are of order unity.

Figure 3 shows the Boltzmann distribution for the seven excitation levels in the ground state or neutral stage of Schadeenium. The figure shows that the lowest excitation level (s=1) is the most populated level at all temperatures. The energy for this level is $\chi_{1,1} = 0$ eV, so by looking at equation 3, we can see that
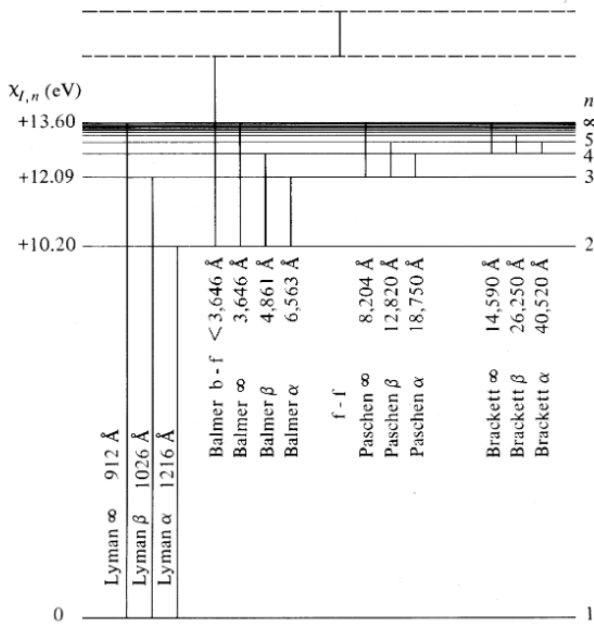
2

Fig. 2: Energy levels for hydrogen, each horizontal line represents one energy level. The excitation energies $\chi_{1,n}$ from the ground state to the respective energy levels are written in eV on the left. n denotes the energy level. The figure also shows the wavelengths the energy levels represent when light is emitted or absorbed from the atom. Novotny (1973)

for this level the Boltzmann distribution becomes temperature independent. This means that for this level the Boltzmann distribution is only dependent on the partition function, which also is almost invariant of temperature.
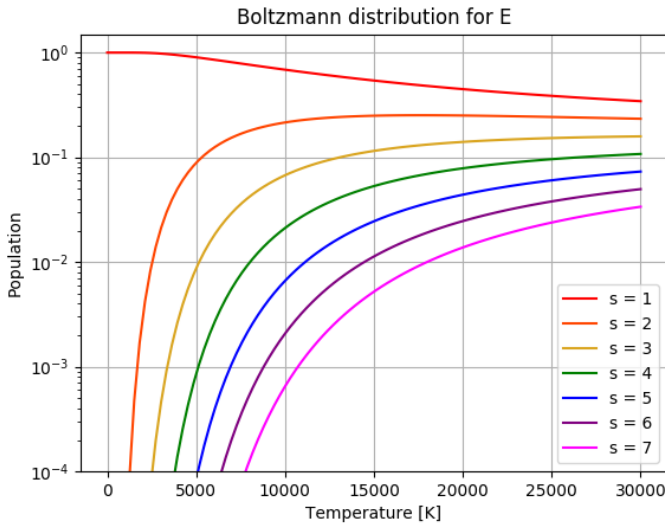

Fig. 3: Boltzmann's distribution for the seven excitation levels in the neutral/ground stage of Schadeenium plotted with respect to temperature. The first excitation level dominates at all temperatures.

Figure 4 shows the Saha distribution for Schadeenium for the neutral or ground state and the three first ionization

stages. From the distribution function we can see that there is at all temperatures maximum two stages of ionization. This is very different compared to the Boltzmann function, where there all the different energy states can be represented at the same temperature. The reason for their different dependence of temperature can be seen by looking at the equations for the Saha and Boltzmann distribution (equation 4 and equation 3. Here we can see that the Saha distribution is dependent on the temperature T to a factor of 3/2 in addition to the temperature in the exponent, which they both have in common.

Another thing worth noting, is that the Saha distribution also depends on the electron pressure. In fact, with a lower pressure of electrons, the ionization happens at lower temperatures. If we have more electrons in the gas, it would be harder for an atom to let go of another electron. But with a lower density of electrons, it is easier for the atom to "push away" a electron. A plot comparing the difference between Saha distributions for two different electron pressures, can be found in appendix A.
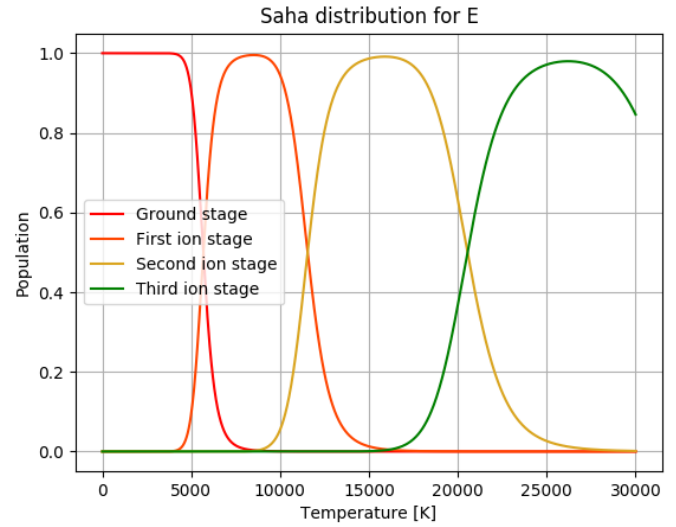

Fig. 4: The Saha distribution for Schadeenium for the neutral/-ground stage and the three first ionization stages plotted with respect to temperature. Only two ionization stages dominates at each temperature.

By comparing the Saha distribution in figure 4 and the Boltzmann distribution in figure 3, we can see that according to the Saha distribution, it is fully possible to ionize the atom. But in the Boltzmann distribution this probability is low because the ground state is always the most populated at all temperatures. The reason to this contradiction can be found by looking at equations for the Saha distributions (equation 1) and how it changes for different values at a constant temperature T. The partition function we have seen in table 1 that does not vary much with temperature, so we can ignore it. If we only look at a given constant temperature, the temperature can be ignored as well. The exponent is more or less the same for high ionization energies, so we are only left with the electron density N_e. Low N_e gives a higher likelihood to get to next ionization stage. If the electron density is low it is easy to give away electrons.

By making a plot of the Saha distribution multiplied with the Boltzmann distribution (Payne's curves), we can obtain

even more information about how the ionization and excitation change with temperature. This is done in figure 5. Here the Boltzmann distribution for the first level of excitation multiplied with different ionization stages is plotted. It is clearly that the ground stage dominates here, and even though the temperature increases, the population of the higher states is quite low, because we look at atoms with electrons in the first level of excitation. The steep flanks on each side of each peak is a result of the Saha distribution's shape.

In appendix B it is attached two more plots of the Saha-Boltzmann distributions of Schadeenium. These two are plotted for the the second level of excitation and the fourth level of excitation. For second level excitation, the population of higher ionized atoms decrease because the energy left to fully ionize the atom is now lower. For the forth level excitation, the electrons are excited in a high enough level to make more or less all the atoms ionized for temperatures over 5000 K.

For elements with higher ionization energies than E, there would be an even higher population of atoms in the ground stage, because it would require more energy to ionize the atom. For elements with lower ionization energies than E, the higher ionized stages would be more populated, and the ground stage would be less populated.
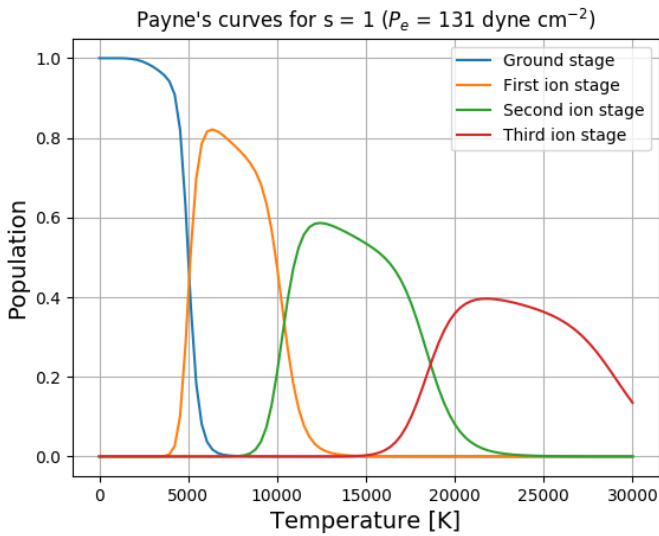

Fig. 5: The Saha distribution multiplied with the Boltzmann distribution plotted with respect to temperature at electron pressure $P_e = 131$ dyne cm^{-2}. We can recognize the behavior of the Saha distribution.

4.2. Hydrogen and calcium

Further we will look at two more realistic elements, hydrogen and calcium. If the assumption we made in section 2 holds, we will find the strength ratio between the Ca$^+$K and Hα balmer absorption lines by plotting the ratio between the respective Saha-Boltzmann distributions. We would expect the Ca$^+$K line to be stronger at temperatures around 4000-6000 K because the Ca$^+$K line is in the ground level of excitation, but the Hα Balmer line comes from the second to third transition of excitation. The Boltzmann distributions of Ca$^+$K and Hα will behave similar to

the Boltzmann distribution of E, even though this is a simplified atom, and therefor we would expect the ground level to be much more populated than the other levels as seen in figure 3.
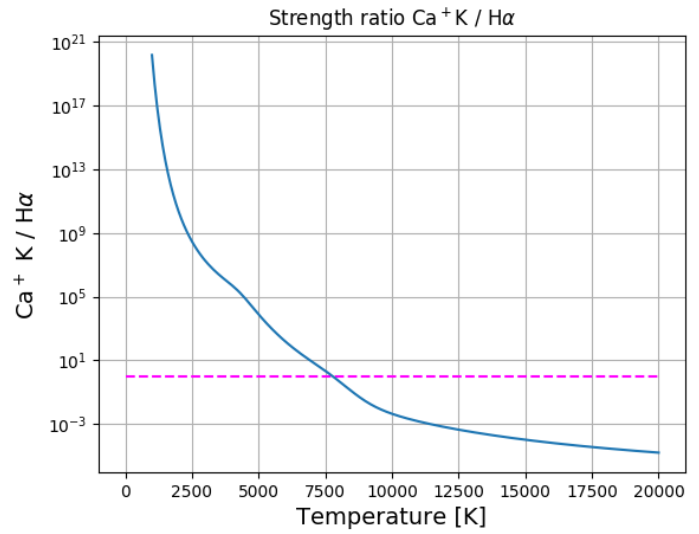

Fig. 6: The strength ratio between the Ca$^+$K and Hα balmer lines plotted with respect to the temperature. The ratio is found based by the assumption made in section 2.

The ratio between the strength of the lines is plotted in figure 6. The figure shows that for temperatures lower than around 8000 K, the Ca$^+$K line should be stronger than the Hα line as we assumed. Around 8000 K, the ratio is equal to 1, which means that the lines should be approximately at the same strength. At temperatures higher than 8000 K, the Hα line is the strongest.

The Ca$^+$K line and the Hα line, also differ much in their temperature sensitivity. The temperature sensitivity is shown in figure 7 together with the Saha-Boltzmann population of the two. At a bit over 6000 K, the Ca$^+$K curve drops to very small values. And a bit under 10 000 K the same happens for Hα line. By comparing the temperature sensitivity to the population curves above in the figure, we can see that the populations have peaks at the same temperatures as where the temperature sensitivity drops. When the population reaches the maximum population, the temperature sensitivity is almost equal to zero for both of the elements. And when the population increases and decreases, the temperature sensitivity is bigger. So we can see that on the left side of the dips the population increases ($\Delta n > 0$), and on the right side of the dips the population decreases ($\Delta n < 0$).

The Saha-Boltzmann distribution for neutral hydrogen is shown in figure 8. The figure shows the fraction of hydrogen that is not ionized at different temperatures. The hydrogen is half ionized at around 9200 K. This transition divides the "hotter" from the "cooler" stars. In hot stars, most of the hydrogen is ionized, resulting in a lot of free electrons. In cool stars almost none of the hydrogen atoms is ionized.

5. Conclusions

The goal of this article was to study the Harvard spectral sequence and spectral lines. By studying the Boltzmann and Saha
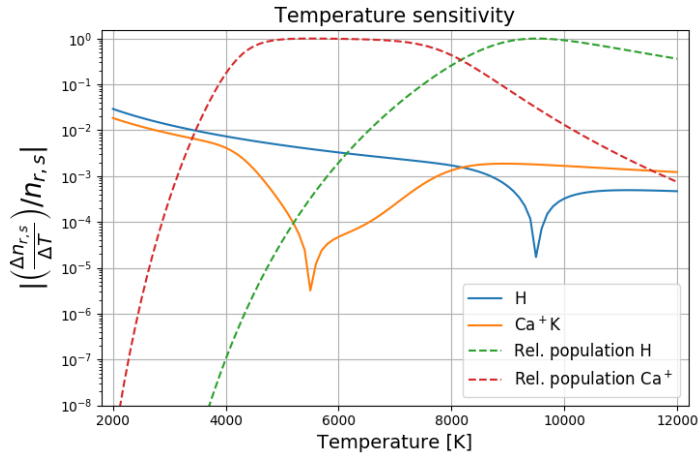
Fig. 7: The temperature sensitivity of Ca^+K line and the $H\alpha$ balmer lines plotted with respect to temperature together with the Saha-Boltzmann population of the two elements. We can see that the sensitivity have a connection with the population.
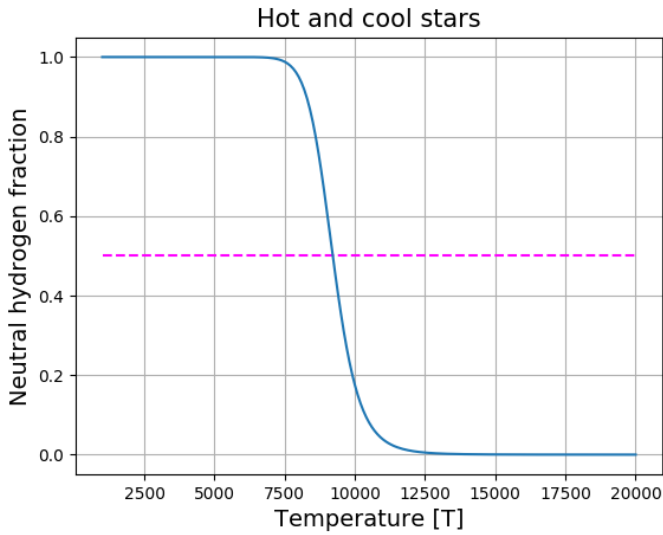

Fig. 8: The fraction of neutral hydrogen plotted with respect to temperature. Half of the hydrogen is ionized at a temperature around 9200 K.

distributions for different fictional and realistic elements we can conclude that the spectral lines have a close connection to the temperature. This means that a spectrum can tell us how the temperature is on a star. This conclusion also tells us that the Harvard spectral sequence basically classifies stars after temperature.

References

Novotny, E. 1973, Introduction to stellar atmospheres and interiors ADS

Fraunhofer line strengths and the curve of growth

1. Introduction

In the first part of this article, the Harvard classification was studied in terms of physics using the Saha and Boltzmann distributions for ionized and excited atoms. It was concluded that the variation in spectral line strength had a close connection to the change in temperature in stars. This part of the article will study how spectral lines form. This part will be less detailed and more figure oriented. All the algorithms can be found i appendix D.

2. Theory

2.1. Planck's law

The radiation intensity emitted by a gas or body in thermodynamical equilibrium is given by

$$B_\lambda(T) = \frac{2hc^2}{\lambda^5} \frac{1}{e^{hc/\lambda kT} - 1}, \tag{1}$$

where h is the Planck constant, c is the speed of light, k is the Boltzmann constant, λ is the wavelength and T is the temperature.

In figure 1 Planck's function is plotted with respect to wavelength for different temperatures between 5000 and 8000 K. The figure shows that each temperature has a peak, and that these peaks shifts to shorter wavelengths for higher temperatures as a result of *Wiens displacement law*. The highest temperature has the highest peak, so highest temperature emits the most intense radiation. The area under Planck's function for different temperatures increases a lot as the temperature increases. This means that the integrated Planck function is larger for higher temperatures.

We can also see that the function increases exponentially much faster for short wavelengths (at the left side of the figure) than at longer wavelengths. This is because when the $hc/\lambda kT$-term is sufficiently large, that is when the wavelength is short, Planck's function can be approximated with the *Wien approximation* given by

$$B_\lambda \approx \frac{2hc^2}{\lambda^5} e^{-hc/\lambda kT}. \tag{2}$$

On the right side of the figure, the curves have a slope downwards. The same can be observed if we plot the Planck function with logarithmic y-axis or logarithmic x- and y-axis as shown in 2. This slope is caused by the longer wavelengths. When the wavelengths are longer $\exp(hc/\lambda kT) \approx hc/\lambda kT$, giving the *Rayleigh-Jeans approximation*,

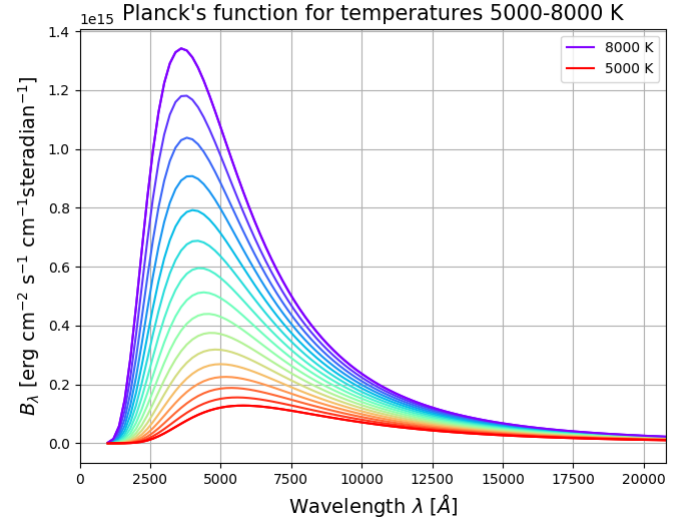$$B_\lambda \approx \frac{2ckT}{\lambda^4}. \tag{3}$$

Fig. 1: Planck's function plotted with respect to wavelength for temperatures between 5000 and 8000 K. The red curve shows Planck's function for 5000 K, and the purple line shows Planck's function for 8000 K.

2.2. Radiation through an isothermal layer

When a beam of radiation with intensity $I_\lambda(0)$ passes through a layer of gas, it is attenuated. In addition the layer contributes to the beam with radiation originated within the layer. The total radiation emitting the layer is then given by

$$I_\lambda = I_\lambda(0)e^{-\tau} + \int_0^\tau B_\lambda[T(x)]e^{-(\tau-\tau(x))}d\tau(x), \tag{4}$$

where τ is the optical thickness, B_λ is Planck's function and x is the location with the layer. For an isothermal layer the temperature T is constant, and therefore B_λ is also constant, giving

$$I_\lambda = I_\lambda(0)e^{-\tau} + B_\lambda \int_0^\tau e^{-(\tau-\tau(x))}d\tau(x)$$

Substituting $u = -(\tau - \tau(x))$ and $du = d\tau$ gives

$$I_\lambda = I_\lambda(0)e^{-\tau} + B_\lambda \int_0^\tau e^u du$$
$$= I_\lambda(0)e^{-\tau} + B_\lambda \left[e^{-\tau+\tau(x)} \right]_0^\tau$$
$$= I_\lambda(0)e^{-\tau} + B_\lambda(e^0 - e^{-\tau})$$
$$= I_\lambda(0)e^{-\tau} + B_\lambda(1 - e^{-\tau}). \tag{5}$$

Figure 3 shows the emergent intensity through an isothermal layer where equation 5 holds and $B_\lambda = 2$. Figure 4 shows the same case with logarithmic x- and y-axis. Here we can see that when the optical thickness is low, the emergent intensity is almost the same as the radiation intensity before the beam passes through the medium. Therefore such a layer is called optically
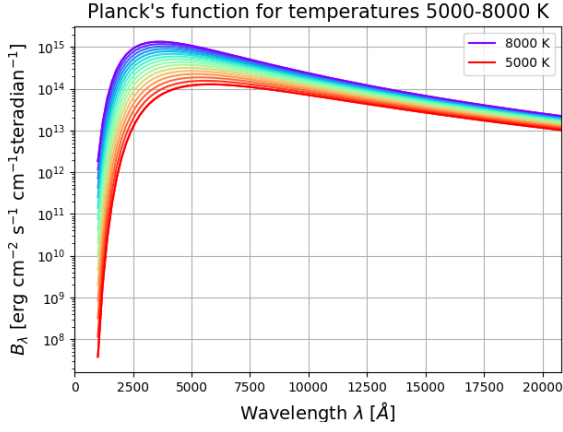
(a) Logarithmic y-axis.
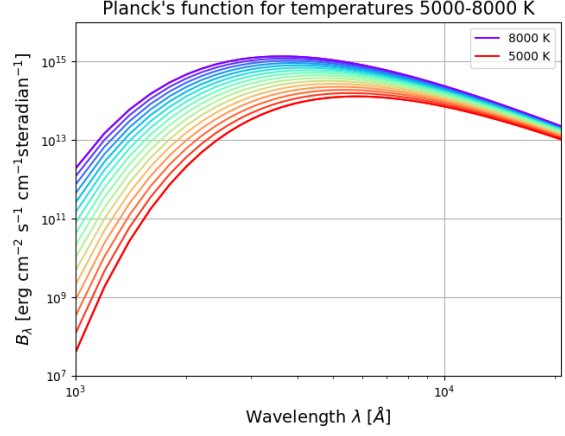
(b) Logarithmic x- and y-axis.

Fig. 2: Planck's function plotted with respect to wavelength for temperatures between 5000 and 8000 K. The red curve shows Planck's function for 5000 K, and the purple line shows Planck's function for 8000 K. The left figure has logarithmic y-axis. The right plot has logarithmic x- and y-axis.

thin. For higher optically thickness, the emergent intensity approaches the radiation intensity emitted by the medium itself ($_\lambda$). So the initial intensity decreases if $I_\lambda(0) > B_+\lambda$. Therefore such a layer is called optically thick. Even when no intensity is sent through the medium, the radiation intensity out of the medium will increase against the Planck function.

For very large optically thickness, the emergent intensity becomes independent of τ. This is because if the medium is very optically thick, no radiation will emit through the medium. So the medium will only emit with radiation originated within itself.
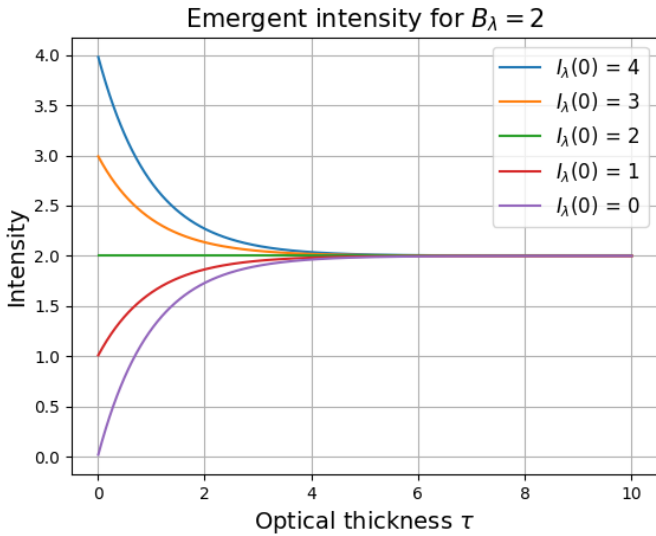

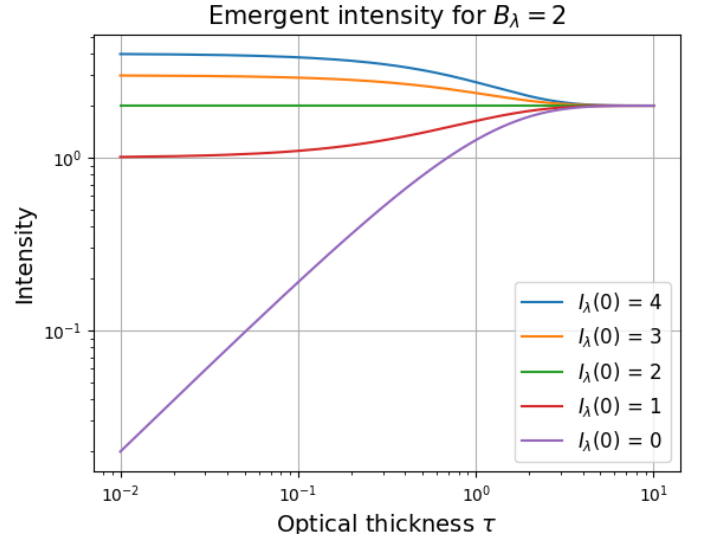Fig. 4: The emergent intensity through an isothermal layer with $B_\lambda = 2$ plotted against the optical thickness τ with logarithmic x- and y-axis.

2.3. Spectral lines from a solar reversing layer

Now we will apply the above results to a simple "reversing-layer" model, called the Schuster-Schwarzschild model. In this model we assume that the continuous radiation emitted from a stellar surface, without spectral lines, irradiates a separate layer further out with intensity

$$I_\lambda(0) = B_\lambda(T_{\text{surface}}). \tag{6}$$

The layer further out makes the spectral lines and is assumed to only contain atoms making these spectral lines. The total emergent radiation is then given by

$$I_\lambda = B_\lambda(T_{\text{surface}})e^{-\tau_\lambda} + B_\lambda(T_{\text{layer}})(1 - e^{-\tau_\lambda}), \tag{7}$$

where τ_λ varies with wavelength λ. The wavelength dependence of τ results in what we call line broadening, which means that

Fig. 3: The emergent intensity through an isothermal layer with $B_\lambda = 2$ plotted against the optical thickness τ.

the spectral lines is not exactly at one wavelength, but is a little bit spread out in wavelength. The broadening can be described by the Voigt function.

A Voigt function represents the convolution of a Gauss profile with a Lorentz profile. The function has a typical Gaussian shape close to the center of a spectral line. It is often approximated by taking the sum instead of the convolution of the two profiles

$$V(a, u) \approx \frac{1}{\Delta\lambda_D \sqrt{\pi}} \left[e^{-u^2} + \frac{a}{\sqrt{\pi}u^2} \right] \tag{8}$$

Figure 5 shows the Voigt profile with different dampening parameters plotted against u. u measures the wavelength separation $\Delta\lambda = \lambda - \lambda(0)$ from the center of the spectral line at $\lambda = \lambda(0)$ in dimensionless units

$$u \equiv \Delta\lambda / \Delta\lambda_D \tag{9}$$

where $\Delta\lambda_D$ is the *Doppler width*. Stellar atmospheres typically have a dampening parameter between $a \approx 0.01 - 0.5$. The figure shows that larger dampening parameters gives a lower and more broad peak. Smaller dampening factors on the other hand, gives higher, narrower peaks. From equation 8 we can see that when $u \to 0$ the Voigt profile will increase to infinity.
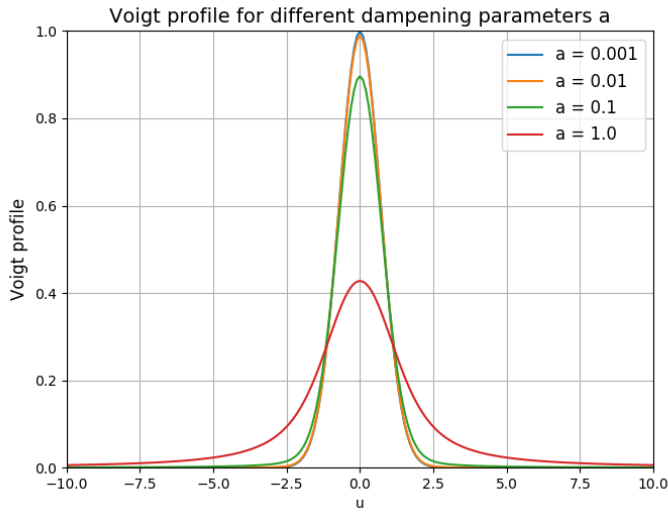

Fig. 6: The Voigt profile with different dampening parameters plotted against u, where u is a measure on the wavelength separation from the center of the spectral line. Here it is used a logarithmic y-axis.

$\lambda = 5000$Å. We can see that the figure clearly shows an upside down Voigt function, thus an absorption line.

Fig. 5: The Voigt profile with different dampening parameters plotted against u, where u is a measure on the wavelength separation from the center of the spectral line.

Figure 6 shows the same Voigt functions plotted with logarithmic y-axis. Here we can also see that the wings of the peak are different for different dampening factor a. When u grows, the exponential term in equation 8 becomes very small, and can be neglected. Then we see that higher a-values give higher values for the Voigt function for large u.

We can now combine the Schuster-Schwarchild method in equation 7 and the Voigt profile to plot spectral line profiles. Figure 7 shows the Schuster-Schwarzschild line profile for a solar like photosphere emitting radiation with wavelength
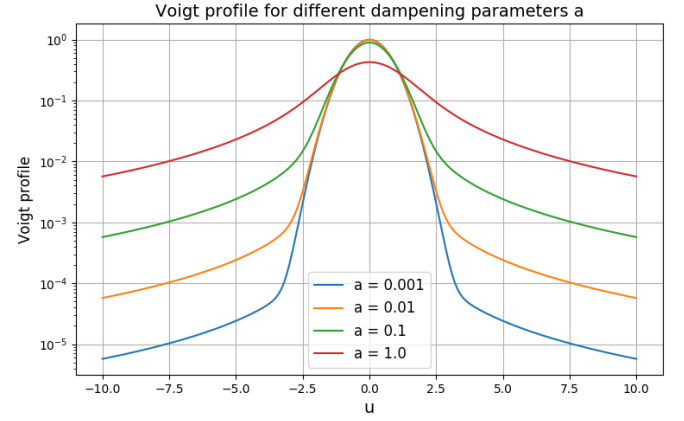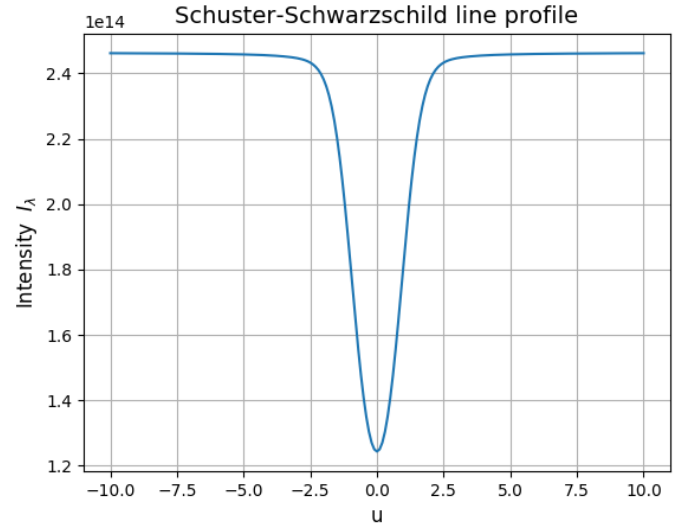

Fig. 7: The Schuster-Schwarzschild line profile for a solar like photosphere, emitting radiation with wavelength $\lambda = 5000$Å, plotted with respect to u. Here the opaqueness $\tau = 1$.

Figure 8 shows the Schuster-Schwarzschild line profile, also for a solar photosphere with the same wavelength, but with different opaqueness τ. We can see that when the opaqueness increases, the absorption line will be stronger (less intensity). For layers with low opaqueness, the absorption line will probably almost not be visible. The layer is *optically thin* if all the radiation from the surface of the star is emitted through the layer, thus we see no absorption line. This is the case when $\log(\tau) \leq -2.0$.

The layer is *optically thick* if none of the radiation from the surface of the stars is emitted through the layer. This is the case when $\log\tau \geq 1.0$. Then the only intensity from the layer, is the

	$\lambda = 2000\text{Å}$	$\lambda = 5000\text{Å}$	$\lambda = 10000\text{Å}$
$B_\lambda(T_{\text{surface}})$	$1.23 \cdot 10^{13}$	$2.46 \cdot 10^{14}$	$1.04 \cdot 10^{14}$
$B_\lambda(T_{\text{layer}})$	$1.36 \cdot 10^{11}$	$4.04 \cdot 10^{13}$	$4.00 \cdot 10^{13}$

Table 2: The Planck function of the layer temperature and the surface temperature for different wavelengths. The Planck values are given in unit [erg cm^{-2} s^{-1} cm^{-1} steradian^{-1}].

emitted intensity originated in the layer itself as we also saw in figure 3. This gives a limit on how low the intensity can by. The emitted intensity originated in the layer itself is given by the Planck function, and is $B_\lambda \approx 4 \cdot 10^{13}$ for the values used in this case. This number matches good with the intensity limit in the figure.

We can also see in the plot that only the profiles for very large opaqueness have wings, which means that the absorption lines in optically thick layers broaden more.
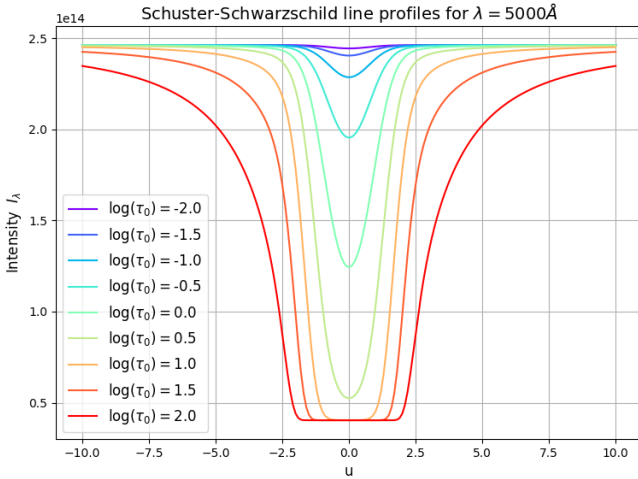

Fig. 8: The Schuster-Schwarzschild line profile for a solar like photosphere emitting radiation with wavelength $\lambda = 5000\text{Å}$ for different opaqueness τ plotted with respect to u.

If we had studied the Schuster-Schwarzchild line profile for other wavelengths, we would have gotten other results. Table 2 shows the Planck function for $\lambda = 2000\text{Å}$, $\lambda = 5000\text{Å}$ and $\lambda = 10000\text{Å}$ for both the surface temperature and the layer temperature. The Planck function of the layer temperature gives the lower limit for the intensity as already mentioned. The Planck function of the surface temperature gives the intensity of the radiation from the surface I_{cont}, therefor it sets the top value of the intensity. From the table we can see that the shortest wavelength gives the biggest difference between the two intensities, and therefore the deepest curve. The longest wavelength gives the smallest difference and therefore the less deep curve. This is also illustrated in figure 9. Here the line profiles of the three wavelengths are plotted, but they are scaled with the intensity $I_+textcont$, so it is easier to compare them.
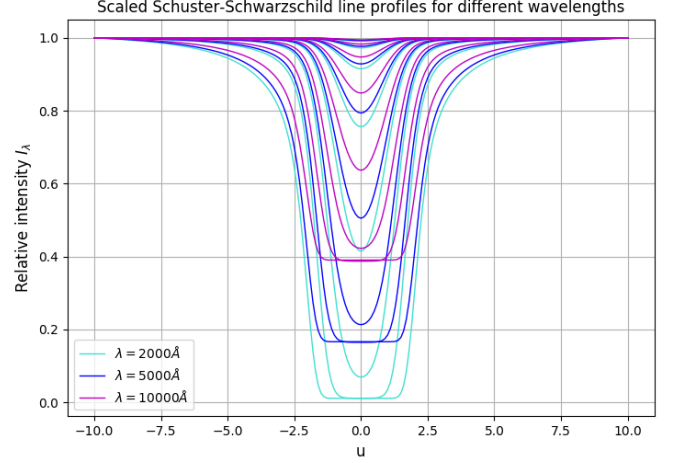

Fig. 9: The Schuster-Schwarzschild line profile for a solar like photosphere emitting radiation with wavelengths $\lambda = 2000\text{Å}$, $\lambda = 5000\text{Å}$ and $\lambda = 10000\text{Å}$ for different opaqueness τ, plotted with respect to u. The profiles are scaled with their respective I_{cont}.

2.4. The equivalent width of spectral lines

The *equivalent width W_λ* is a line-strength parameter to measure the growth of the absorption feature. The parameter measures the integrated line depression in the normalized spectrum, and is given by

$$W_\lambda \equiv \int \frac{I_{\text{cont}} - I(\lambda)}{I_{\text{cont}}} d\lambda. \tag{10}$$

In figure 10 the Schuster-Schwarzschild line profile for $\tau_0 = 100$ is plotted with a wide range of u-values. In figure 11 the same profile is plotted once more, but now relative to I_{cont}, as the integrand in 10. The large range of u-values is used to ensure that the first value at $u = -200$ is equal to I_{cont}.

2.5. The curve of growth

The main idea behind the equivalent width was to measure the number of atoms in the reversing layer. The previous profile plots have shown that the profile growth is only linear with τ_0 for $\tau_0 << 1$. The equivalent width (equation 10) can be calculated numerically by summing over the integrand in the equation for all wavelengths. Figure 12 shows W_λ plotted with respect to $\log \tau_0$. Here we can see that the first part is linear (1:1), because here $\tau_0 << 1$. When we reach $\tau(0) = 1$ the curve has a slope of 1:2. In the last part of the figure, the graph is a bit steeper again. This last part changes with the dampening parameter. Be comparing this plot with for instance the equivalent width for iron, we can see that they are very similar. Therefor iron might have an dampening parameter around the same as we have used in this plot (a=0.1).

If we would produce emission lines instead of absorption lines, we need to change the surface and layer temperatures. We have now used a surface temperature larger than the layer temperature, and have gotten absorption lines. By using a larger layer temperature and a lower surface temperature, we would get
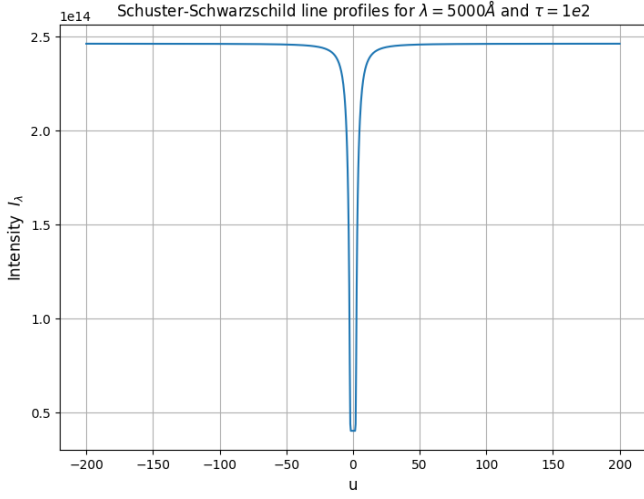
Fig. 10: The Schuster-Schwarzschild line profile for a solar like photosphere emitting radiation with wavelength $\lambda = 5000\text{Å}$ and opaqueness $\tau_0 = 100$ plotted with respect to u. Same profile as the one for $\log(\tau_0) = 2.0$ in figure 8, but width a wider range of u-values.

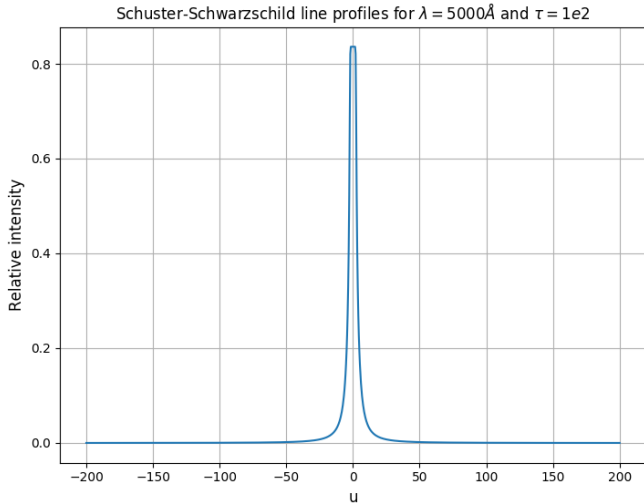Fig. 12: The equivalent width W_λ plotted with respect to τ_0 for an absorption line.

Fig. 11: The relative Schuster-Schwarzschild line profile for a solar like photosphere emitting radiation with wavelength $\lambda = 5000\text{Å}$ and opaqueness $\tau_0 = 100$ plotted with respect to u.
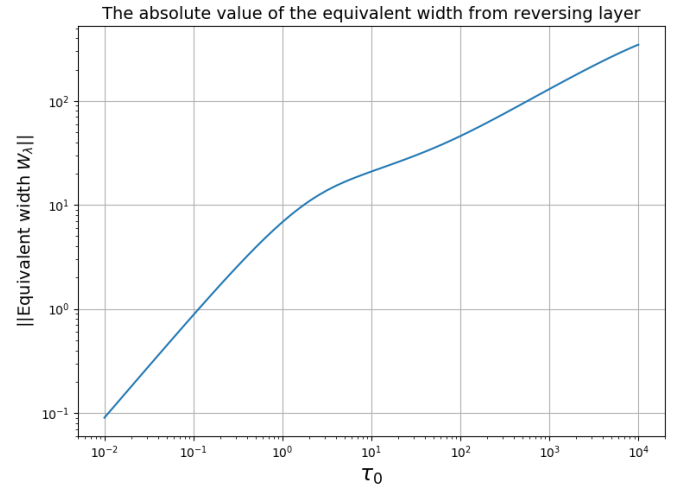
Fig. 13: The absolute value of the equivalent width W_λ plotted with respect to τ_0 for an emission line. The curve has a very similar behaviour as for the absorption line in figure 12.

the opposite case, because we have already seen in table 2 that these temperatures sets the lower and upper intensity limits. The equivalent width W_λ is plotted with respect to τ_0 for an emission line in figure 13. The line profile for the same emission line is plotted in figure 14.

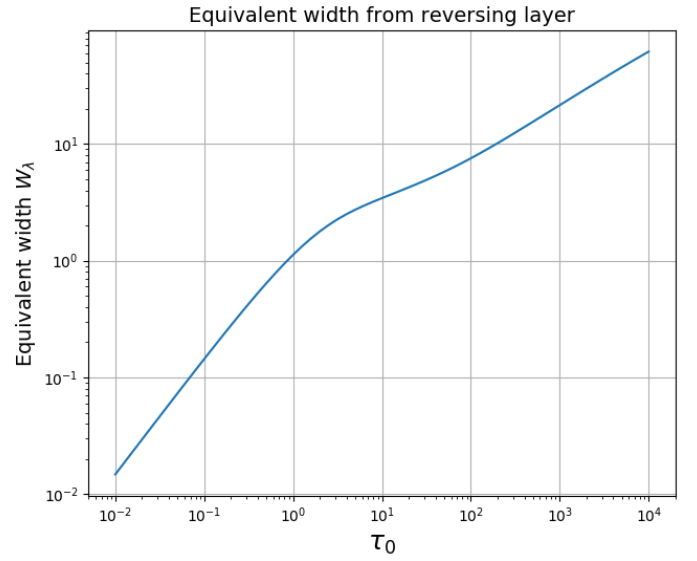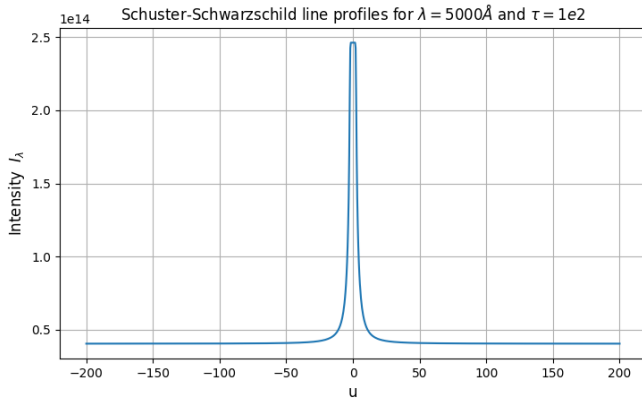Fig. 14: The Schuster-Schwarzschild line profile for an emission line with wavelength $\lambda = 5000\text{Å}$ and opaqueness $\tau_0 = 100$ plotted with respect to u.

Appendix A: Saha distribution for different electron pressures
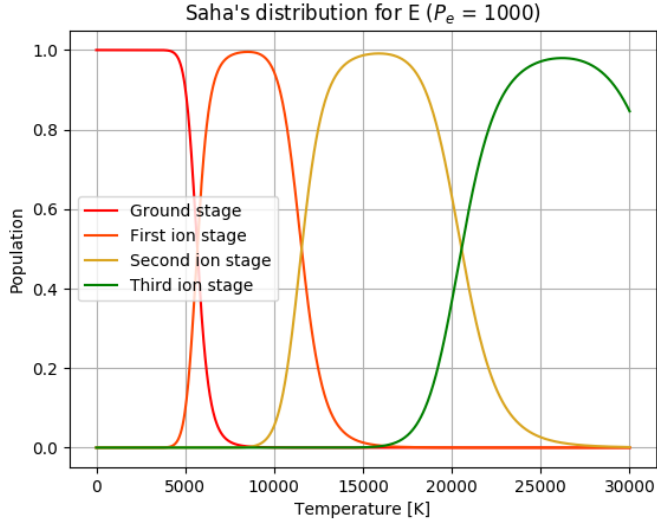
Fig. A.1: The Saha distribution for Schadeenium for the neutral/ground stage and the three first ionization stages plotted with respect to temperature at electron pressure $P_e = 1000$.
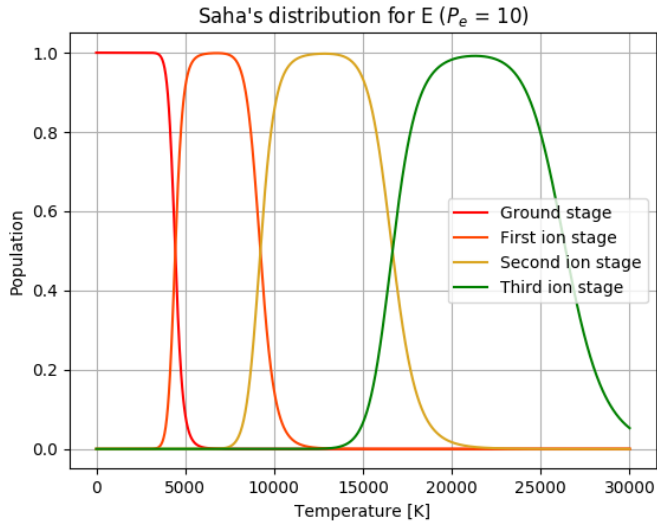
Fig. A.2: The Saha distribution for Schadeenium for the neutral/ground stage and the three first ionization stages plotted with respect to temperature at electron pressure $P_e = 10$.

Appendix B: Saha-Boltzmann distribution with excitation level s=2 and s=4
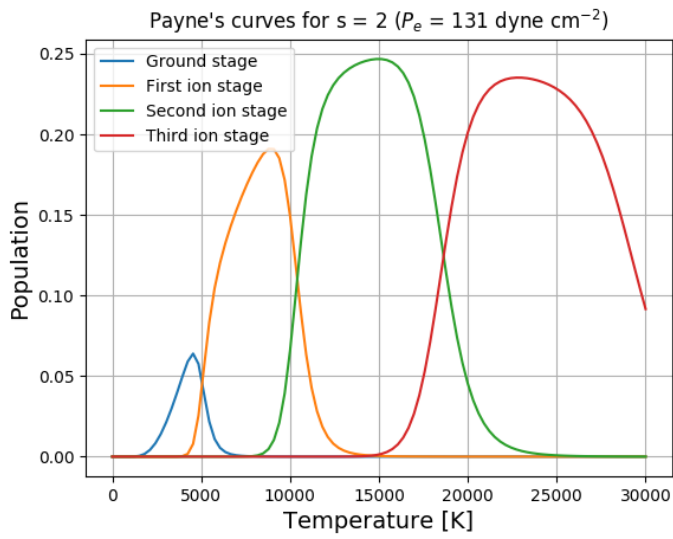
Fig. B.1: Saha-Boltzmann distribution with excitation level s=2 plotted with respect to temperature.
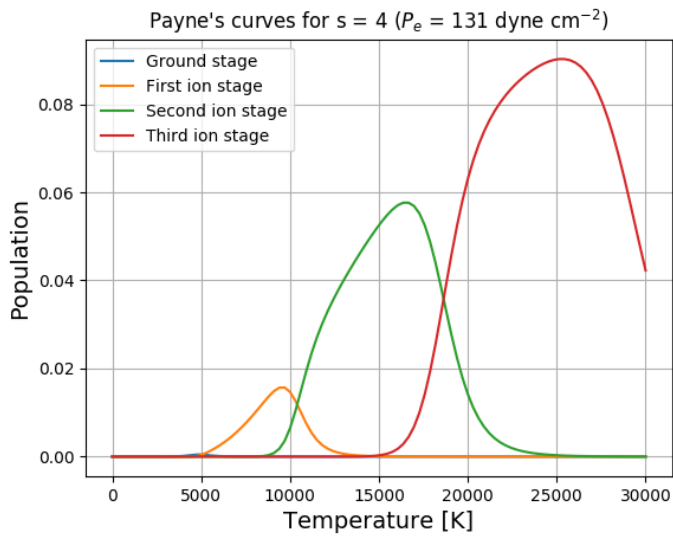
Fig. B.2: Saha-Boltzmann distribution with excitation level s=4 plotted with respect to temperature.

Appendix C: Algorithms - Saha-Boltzmann calibration of the Havard sequence

```python
import numpy as np
import matplotlib.pyplot as plt


""" Defining constants """
k_erg = 1.380658e-16                    # Boltzmann constant (erg K)
k_eV = 8.61734e-5                       # Boltzmann constant (eV/deg)
h = 6.62607e-27                         # Planck constant (erg s)
m_e = 9.109390e-28                      # Electron mass (g)
chi_ion = np.array([7, 16, 31, 51])     # Create numpy array of Schadee ionization energies
P_e = 1e3                               # Electron pressure (dyne cm^-2)

colors = ['red', 'orangered', 'goldenrod', 'green',  'blue', 'purple', 'magenta']



""" Defining functions """
def partfunc_E(T):
    U = np.zeros(4)                     # Create a 4 zero-element array

    # Calculates the partition function for the different ionization energies
    for r in range(4):
        for s in range(chi_ion[r]):
            U[r] = U[r] + np.exp(-s /(k_eV*T))
    return U                            #returns all the values of u array



def boltz_E(T, r, s):
    U = partfunc_E(T)
    return 1. / U[r-1] * np.exp(-(s - 1.) / (k_eV * T))



def saha_E(T, P_e, r): #ionstage):
    chi_ion = np.array([6, 12, 51, 67])

    el_dens = P_e / (k_erg*T)

    U = partfunc_E(T)
    U = np.append(U, 2) # Adding approximate to last term

    saha_const = ((2. * np.pi * m_e * k_erg * T) / (h**2))**(3./2) * 2. / el_dens  #(k_erg*T)/
        P_e

    N = np.zeros(5)
    N[0] = 1. # We set the first element of the array to a value 1

    for r_ in range(4):
        N[r_ + 1] = N[r_] * saha_const * U[r_ + 1] / U[r_] * np.exp(-chi_ion[r_] / (k_eV*T))

    N_total = np.sum(N)
    N_rel = N / N_total

    return N_rel[r - 1]



def saha_boltz_E(T, P_e, r, s):
        return saha_E(T, P_e, r) * boltz_E(T, r, s)



def saha_boltz_H(T, P_e, r):
    el_dens = P_e / (k_erg*T)

    # energy levels and weights for hydrogen
    num = 100                                       # reasonable partition function cut-off value
```

```python
    g = np.zeros((2, num))                          # declarations weights (too many for proton)
    chi_exc = np.zeros((2, num))                    # declaration excitation energies (idem)
    for s in range(num):
        g[0,s] = 2.*(s+1.)**2.                      # statistical weights
        chi_exc[0,s] = 13.598*(1.-1./(s+1.)**2.)    # excitation weights

    g[1,0] = 1.                                     # statistical weights free proton
    chi_exc[1,0] = 0.

    # Partition functions
    U = np.zeros([2])
    for s in range(num):
        U[0] = U[0] + g[0,s]*np.exp(-chi_exc[0,s]/(k_eV*T))
    U[1] = g[1,0]

    # Saha
    saha_const = (2*np.pi*m_e*(k_erg*T) /(h*h))**(3./2)*2./el_dens
    N = np.zeros(2)
    N[0] = 1.
    N[1] = N[0] * saha_const * U[1]/U[0] * np.exp(-13.598/(k_eV*T))
    N_total = np.sum(N)          # sum both stages = total hydrogen density

    # Boltzmann
    n = N[0]*g[0,r-1]/U[0]*np.exp(-chi_exc[0,r-1]/(k_eV*T))
    n_rel = n/N_total       # fraction of total hydrogen density

    return n_rel


""" Creating plots """
def plot_boltz():
    N = 100
    temp = np.linspace(1,30000,N)
    pop = np.zeros((8,N))
    r = 1
    for s in range(1,8):
        for T in range(N):
            pop[s,T] = boltz_E(temp[T], r, s)


    plt.figure()
    for s in range(1,8):
        plt.semilogy(temp,pop[s,:], label='s = %d' %s, color=colors[s-1])
    plt.ylim([1e-4, 1.5])
    plt.xlabel('Temperature [K]')
    plt.ylabel('Population')
    plt.title('Boltzmann distribution for E')
    plt.legend(loc='best')
    plt.grid()
    plt.show()

def plot_saha():
    #P_e = 1e1
    N = 1000
    temp = np.linspace(1,30000,N)
    pop = np.zeros((5,N))

    for r in range(1,5):
        for T in range(N):
            pop[r,T] = saha_E(temp[T], P_e, r)

    labellst = ['Ground stage', 'First ion stage', 'Second ion stage', 'Third ion stage']

    plt.figure()
    for r in range(1,5):
        plt.plot(temp,pop[r,:], label=labellst[r-1], color=colors[r-1])
    #plt.ylim([1e-4, 1.5])
    plt.xlabel('Temperature [K]')
    plt.ylabel('Population')
```

```python
    plt.title('Saha distribution for E')
    plt.legend(loc='best')
    plt.grid()
    plt.show()



def paynes_curves(s):
    P_e_payne = 131.0                       # Payne's electron pressure (dyne cm^-2)
    N = 100
    temp = np.linspace(1,30001,N)      # Creates array with temperatures between 1 and 30000 K
    pop = np.zeros((5,N))              # Creating empty matrix

    # Calculates Saha * Boltzmann for different temperatures T and different ionization states r
    for T in range(N):
        for r in range(1,5):
            pop[r,T] = saha_boltz_E(temp[T], P_e_payne, r, s)

    labellst = ['Ground stage', 'First ion stage', 'Second ion stage', 'Third ion stage']

    plt.figure()
    for r in range(1,5):
        plt.plot(temp,pop[r,:], label=labellst[r-1])#, color=colors[r+2])
    plt.title(r"Payne's curves for s = %d ($P_e$ = 131 dyne cm$^{-2}$)" %(s))
    plt.xlabel('Temperature [K]', size=14)
    plt.ylabel('Population', size=14)
    #plt.ylim([1e-3, 1.1])
    #plt.xlim([0,30000])
    plt.legend(loc='best')
    plt.grid()
    plt.show()


def line_strength_ratio():
    temp = np.arange(1000,20001,100)
    CaH = np.zeros(temp.shape)
    Caabund = 2.0e-6
    for i in range(0,191):
        NCa = saha_boltz_E(temp[i],1e2,2,1) # is equal to sahabolt_Ca
        NH = saha_boltz_H(temp[i],1e2,2)
        CaH[i] = NCa*Caabund/NH
    plt.hlines(1, 0, 20000, 'magenta', linestyle='--')
    plt.semilogy(temp,CaH)
    plt.title(r'Strength ratio Ca$^+$K / H$\alpha$')
    plt.xlabel('Temperature [K]', size=14)
    plt.ylabel(r'Ca$^+$ K / H$\alpha$', size=14)
    plt.legend(fontsize=14)
    plt.grid()
    plt.show()

    print 'Ca/H ratio at 5000 K = ', CaH[np.argwhere(temp==5000)][0][0]



def temp_sensitivity():
    temp = np.arange(2000,12001,100)
    dNCadT = np.zeros(temp.shape)
    dNHdT = np.zeros(temp.shape)
    dT = 1.
    for i in range(101):
        NCa = saha_boltz_E(temp[i],1e2,2,1)
        NCa2 = saha_boltz_E(temp[i]-dT,1e2,2,1)
        dNCadT[i] = (NCa - NCa2)/(dT*NCa)
        NH = saha_boltz_H(temp[i],1e2,2)
        NH2 = saha_boltz_H(temp[i]-dT,1e2,2)
        dNHdT[i] = (NH-NH2)/(dT*NH)

    NCa = np.zeros(temp.shape)
    NH = np.zeros(temp.shape)
    for i in range(101):
```

```python
        NCa[i] = saha_boltz_E(temp[i],1e2,2,1)
        NH[i] = saha_boltz_H(temp[i],1e2,2)

    plt.figure()
    plt.plot(temp,np.absolute(dNHdT), label=r'H')
    plt.plot(temp,np.absolute(dNCadT), label=r'Ca$^+$K')
    plt.plot(temp,NH/np.amax(NH), ls='--', label = 'Rel. population H')
    plt.plot(temp,NCa/np.amax(NCa), ls='--', label = r'Rel. population Ca$^+$')
    plt.yscale('log')
    plt.xlabel('Temperature [K]', size=14)
    plt.axis([1800,12200, 1e-8, 1.5])
    plt.ylabel(r'$\left| \left( \frac{\Delta n_{r,s}}{\Delta T} \right) /  n_{r,s} \right|$',
        size=20)
    plt.legend(loc='best', fontsize=12)
    plt.title('Temperature sensitivity', fontsize=15)
    plt.grid()
    plt.show()


def hot_vs_cool():
    for T in np.arange(2e3,2e4+1,2e3):
        print T, saha_boltz_H(T,1e2,1)
    temp = np.arange(1e3,2e4+1,1e2)
    nH = np.zeros(temp.shape)
    for i in range(191):
        nH[i] = saha_boltz_H(temp[i],1e2,1)
    plt.plot(temp,nH)
    plt.hlines(0.5, 1e3, 2e4, 'magenta', linestyle='--')
    plt.xlabel('Temperature [T]', size=14)
    plt.ylabel('Neutral hydrogen fraction', size=14)
    plt.title('Hot and cool stars', size=15)
    plt.legend()
    plt.grid()
    plt.show()
```

Appendix D: Algorithms - Fraunhofer line strengths and the curve of growth

```python
import numpy as np
import matplotlib.pyplot as plt
import pylab
from scipy import special

colors = pylab.cm.rainbow(np.linspace(0,1,16))

""" Defining constants """
k_erg = 1.380658e-16                   # Boltzmann constant (erg K)
k_eV = 8.61734e-5                      # Boltzmann constant (eV/deg)
h = 6.62607e-27                        # Planck constant (erg s)
c = 2.99792e10                         # Speed of light (cm s^-1)



def planck(T, lmbda):
    return 2*h*c**2/lmbda**5 * 1/(np.exp(h*c/(lmbda*k_erg*T))-1)

def plot_planck(logy=False, logx=False):
    lmbdas = np.arange(1000,20801,200)
    planck_values = np.zeros(lmbdas.shape)

    plt.figure()
    i = 0
    for T in range(8000,5000-1,-200):
        planck_values[:] = planck(T, lmbdas[:]*1e-8)
        plt.plot(lmbdas, planck_values, '-', color = colors[i])
        i+=1
    plt.plot(lmbdas, planck(8000, lmbdas[:]*1e-8), '-', color = colors[0], label = '8000 K')
    plt.plot(lmbdas, planck(5000, lmbdas[:]*1e-8), '-', color = colors[-1], label = '5000 K')
    plt.xlabel(r'Wavelength $\lambda$ [$\AA$]', size=14)
    plt.ylabel(r'$B_\lambda$ [erg cm$^{-2}$ s$^{-1}$ cm$^{-1}$steradian$^{-1}$]', size=14)
    plt.title("Planck's function for temperatures 5000-8000 K", size=15)
    plt.xlim(0,20800)
    plt.legend()
    plt.grid()

    if logy:
        plt.yscale('log')
    if logx:
        plt.xscale('log')
        plt.axis([1e3,20801,1e7, 1e16])

    plt.show()


def plot_intensity(log=False):
    B = 2.
    tau = np.arange(0.01, 10.01, 0.01)
    intensity = np.zeros(tau.shape)
    for I0 in range(4,-1,-1):
        intensity[:] = I0 * np.exp(-tau[:]) + B*(1-np.exp(-tau[:]))
        plt.plot(tau, intensity, label = r'$I_\lambda(0)$ = ' + str(I0))
    plt.xlabel(r'Optical thickness $\tau$', size=14)
    plt.ylabel('Intensity', size=14)
    plt.legend(fontsize=12, loc='best')
    plt.grid()
    plt.title(r'Emergent intensity for $B_\lambda = 2$', size=15)
    if log:
        plt.yscale('log')
        plt.xscale('log')
    plt.show()



def voigt(gamma,x):
    z = (x+1j*gamma)
    V = special.wofz(z).real
```

```python
        return V

def plot_voigt():
    u = np.arange(-10,10.1,0.1)
    a = np.array([0.001,0.01,0.1,1])
    vau = np.zeros((a.shape[0],u.shape[0]))

    plt.figure()
    for i in range(4):
        vau[i,:] = voigt(a[i],u[:])
        plt.plot(u[:],vau[i,:], label = 'a = ' + np.str(a[i]))
    plt.axis([-10,10, 0,1])
    plt.legend(fontsize=12)
    plt.title('Voigt profile for different dampening parameters a', size=14)
    plt.ylabel('Voigt profile', size=12)
    plt.xlabel('u')
    plt.grid()

    plt.figure()
    for i in range(4):
        vau[i,:] = voigt(a[i],u[:])
        plt.plot(u[:],vau[i,:], label = 'a = ' + np.str(a[i]))
    plt.yscale('log')
    plt.legend(fontsize=12, loc = 8)
    plt.xlabel('u', size=14)
    plt.ylabel('Voigt profile', size=12)
    plt.title('Voigt profile for different dampening parameters a', size=14)
    plt.grid()
    plt.show()


def plot_schuster_schwarzchild():
    colors = pylab.cm.rainbow(np.linspace(0,1,9))

    Ts = 5700.             # solar surface temperature
    Tl = 4200.             # solar T-min temperature = 'reversing layer'
    a = 0.1                # damping parameter
    wav = 5000.0e-8        # wavelength in cm
    #wav = 2000.0e-8       # ultraviolet
    #wav = 10000.0e-8      # near infrared
    tau0 = 1.              # reversing layer thickness at line center

    print 'Wavelength: %d angstrom' %(wav*1e8)
    print "Planck's function of T_surface = %.2e" %(planck(Ts, wav))
    print "Planck's function of T_layer = %.2e" %(planck(Tl, wav))

    u = np.arange(-10,10.1,0.1)
    intensity = np.zeros(u.shape)

    for i in range(201):
        tau = tau0 * voigt(a, u[i])
        intensity[i] = planck(Ts,wav) * np.exp(-tau) + planck(Tl,wav)*(1.-np.exp(-tau))

    plt.figure()
    plt.plot(u,intensity)
    plt.title('Schuster-Schwarzschild line profile',size=14)
    plt.xlabel('u',size=12)
    plt.ylabel(r'Intensity  $I_\lambda$', size=12)
    plt.grid()


    logtau0 = np.arange(-2,2.1,0.5)

    plt.figure()
    for itau in range(9):
        for i in range(201):
            tau = 10.**(logtau0[itau]) * voigt(a, u[i])
            intensity[i] = planck(Ts,wav) * np.exp(-tau) + planck(Tl,wav)*(1.-np.exp(-tau))
```

```python
        plt.plot(u,intensity, label = r'$\log{(\tau_0)} = $' + np.str(logtau0[itau]), color =
            colors[itau])

    plt.legend(loc=3, fontsize=12)
    plt.title(r'Schuster-Schwarzschild line profiles for $\lambda = 5000 \AA$',size=14)
    plt.xlabel('u',size=12)
    plt.ylabel(r'Intensity  $I_\lambda$', size=12)
    plt.grid()
    plt.show()


    colors = ['turquoise', 'b', 'm']
    plt.figure()
    for iwav in range(1,4):
        wav = (iwav**2+1.)*1.0e-5    # wav = 2000, 5000, 10000 angstrom
        for itau in range(8):
            for i in range(201):
                tau = 10.**(logtau0[itau]) * voigt(a,u[i])
                intensity[i] = planck(Ts,wav) * np.exp(-tau) + planck(Tl,wav)*(1.-np.exp(-tau))
            intensity = intensity / intensity[0]
            if itau == 0:
                plt.plot(u,intensity[:],  linewidth=1., color=colors[iwav-1], label=r'$\lambda =
                    %d000 \AA$' %(wav*1e5))
            else:
                plt.plot(u,intensity[:],  linewidth=1., color=colors[iwav-1], label='_nolegend_'
                    )
    plt.legend()
    plt.xlabel('u',size=12)
    plt.ylabel(r'Relative intensity $I_\lambda$', size=12)
    plt.title('Scaled Schuster-Schwarzschild line profiles for different wavelengths', size=12)
    plt.grid()
    plt.show()




def profile(a,tau0,u):
    Tl = 5700.
    Ts = 4200.
    wav = 5000.0e-8
    intensity = np.zeros(u.size)
    usize = u.size
    for i in range(usize):
        tau = tau0 * voigt(a, u[i])
        intensity[i] = planck(Ts,wav)*np.exp(-tau) + planck(Tl,wav)*(1.-np.exp(-tau))
    return intensity

def plot_profile():
    # Checking the profile
    u = np.arange(-200,200.4,0.4)
    a = 0.1
    tau0 = 1.0e2
    intensity = profile(a,tau0,u)
    plt.figure()
    plt.plot(u,intensity)
    plt.title(r'Schuster-Schwarzschild line profiles for $\lambda = 5000 \AA$ and $\tau = 1e2$',
        size=12)
    plt.xlabel('u',size=12)
    plt.ylabel(r'Intensity  $I_\lambda$', size=12)
    plt.grid()
    plt.show()

    # relative
    reldepth = (intensity[0]-intensity)/intensity[0]
    plt.figure()
    plt.title(r'Schuster-Schwarzschild line profiles for $\lambda = 5000 \AA$ and $\tau = 1e2$',
        size=12)
    plt.xlabel('u',size=12)
    plt.ylabel(r'Relative intensity', size=12)
    plt.plot(u,reldepth)
```

```
    plt.grid()
    plt.show()
    eqw = sum(reldepth)*0.4
    print eqw

def plot_curve_of_growth():
    tau0 = np.logspace(-2, 4, 61)
    eqw = np.zeros(tau0.size)
    u = np.arange(-200,200.4,0.4)
    a = 0.1
    for i in range(61):
        intensity = profile(a,tau0[i],u)
        reldepth = (intensity[0] - intensity) / intensity[0]
        eqw[i] = sum(reldepth)*0.4

    plt.figure()
    plt.plot(tau0, abs(eqw))
    plt.xlabel(r'$\tau_0$', size=18)
    plt.title('The absolute value of the equivalent width from reversing layer', size=14)
    plt.ylabel(r'||Equivalent width $W_{\lambda}||$', size=14)
    plt.xscale('log')
    plt.yscale('log')
    plt.grid()
    plt.show()
```