AST4310 - Stellar Spectra A

Simen Nyhus Bastnes
simennb

23. September 2016

1 Introduction

In this project, we will look closer at the nature and appearance of spectral lines in stellar spectra. Doing so, we will retrace some of the steps of the early pioneers in the field of stellar spectroscopy.

First, we will take on the role of Cecilia Payne, and re-act the work on Saha-Boltzmann modeling of the Harvard classification. Then, we place our self in the role of Marcel Minnaert and his work on Schuster-Schwarzschild modeling of Fraunhofer line strengths.

2 Saha-Boltzmann calibration of the Harvard sequence

In this section, we will attempt to explain the spectral-type sequence. In doing so, we will re-act the work of Cecilia Payne at Harvard. We will apply the Saha distribution for different ionization stages of an element to stellar spectra to find that the Harvard classification represents primarily a temperature scale.

2.1 The Boltzmann and Saha laws

First of all, we look at an energy level diagram for hydrogen.

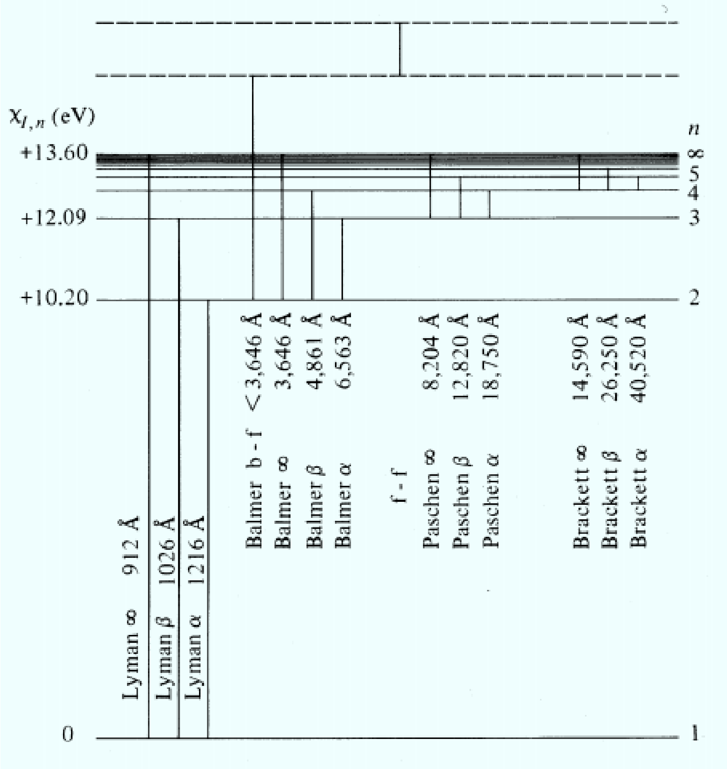


Figure 1: Energy level diagram for hydrogen, taken from figure 7 in reference[1]

We see that some of the transitions share levels. The Balmer series all share the same lower level as each other, the Lyman series share the same lower level, the Paschen share lower level, and the Brackett share lower level, so one would assume that the series are defined based on all transitions to a common lower level.

When it comes to sharing higher levels, Balmer β and Paschen β share the same upper level, as do Balmer α and Lyman β. All of the series also have an ∞ transition that share the same "upper level".

Payne assumed that the strength of the absorption lines observed in stellar spectra scales with the population density of the lower level of the corresponding transition. Even though it is not correct, it would at first glance seem likely, as a higher lower level population would mean that there is a higher amount of particles that can be excited to a higher level, and then back down again. We will in this exercise follow her example, and assume that the scaling is linear.

With this, we can maybe attempt to give some rough estimates of the strength ratios of the α lines in the HI Lyman, Balmer, Paschen and Brackett series. Since Lyman α has a lower level in the ground state, we would assume that it would be the strongest of the lines, with Balmer being a far bit weaker than that, and Paschen and Brackett being consecutively weaker, but not as large difference due to the lower lever energy differensial decreasing the further up towards ionization we go.

Boltzmann law

In TE the partitioning of a specific atom or ion stage over its discrete energy levels is given by the Boltzmann distribution

$$\frac{n_{r,s}}{N_r} = \frac{g_{r,s}}{U_r}\, e^{-\chi_{r,s}/kT} \tag{1}$$

with T the temperature, k the Boltzmann constant, $n_{r,s}$ the number of particles per cm^3 in level s of ionization stage r, $g_{r,s}$ the statistical weights of that level, and $\chi_{r,s}$ the excitation energy of that level measured from the ground state $(r, 1)$, $N_r \equiv \sum_s n_{r,s}$ the total particle density in all levels of ionization stage r, and U_r its partition function defined by

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

Saha law

In TE the particle partitioning over the various ionization stages of an element is given by the Saha distribution

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

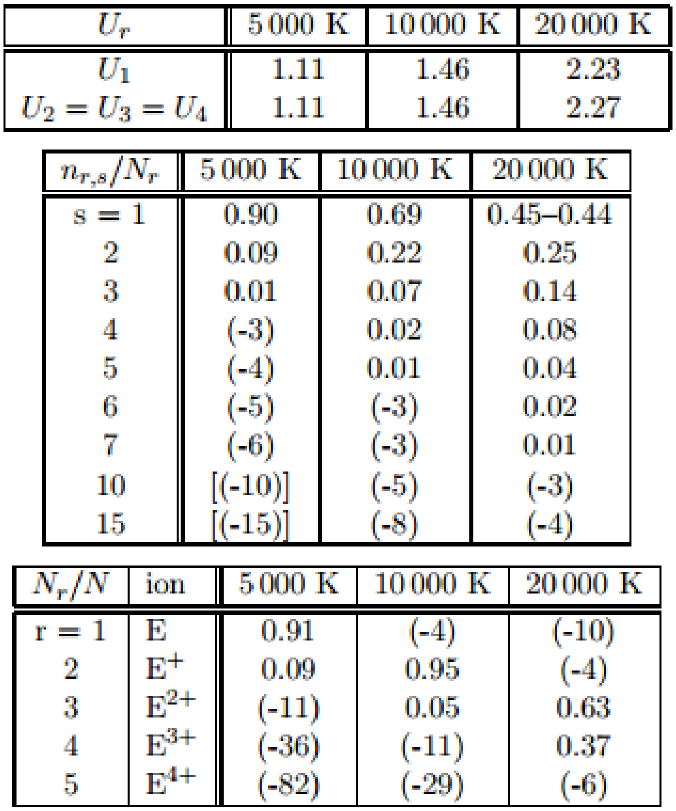
with N_e the electron density, m_e the electron mass, h the planck constant, chi_r the threshold energy needed to ionize stage r to stage $r + 1$, and U_{r+1} and U_r the partition function of ionization stages $r + 1$ and r.

2.2 Schadee's tables for schadeenium

To test the Boltzmann and Saha laws, we look at a hypothetical (but iron-like) element *schadeenium* in conditions similar to a stellar atmosphere. This element "E", has the following properties.

- ionization energies $\chi_1 = 7$ eV for neutral E, $\chi_2 = 16$ eV for E$^+$, $\chi_3 = 31$ eV for E^{2+}, $\chi_4 = 51$ eV for E^{3+}
- excitation energies that increase incrementally by 1 eV: $\chi_{r,s} = -1$ eV in each stage
- statistical weights $g_{r,s} \equiv 1$ for all levels (r,s)

We evaluate the Saha and Boltzmann laws for element E, electron pressure $P_e = 10^3$ dyne cm$^-2$ and temperature $T = 5\,000$, $10\,000$, $20\,000$, and reproduce table 1 in reference [1]. Table generated by code is appended at the end of the document.



U_r	5 000 K	10 000 K	20 000 K
U_1	1.11	1.46	2.23
$U_2 = U_3 = U_4$	1.11	1.46	2.27

$n_{r,s}/N_r$	5 000 K	10 000 K	20 000 K
s = 1	0.90	0.69	0.45–0.44
2	0.09	0.22	0.25
3	0.01	0.07	0.14
4	(-3)	0.02	0.08
5	(-4)	0.01	0.04
6	(-5)	(-3)	0.02
7	(-6)	(-3)	0.01
10	[(-10)]	(-5)	(-3)
15	[(-15)]	(-8)	(-4)

N_r/N	ion	5 000 K	10 000 K	20 000 K
r = 1	E	0.91	(-4)	(-10)
2	E^+	0.09	0.95	(-4)
3	E^{2+}	(-11)	0.05	0.63
4	E^{3+}	(-36)	(-11)	0.37
5	E^{4+}	(-82)	(-29)	(-6)

Figure 2: Table 1 from reference [1]

From the table, we see that the Saha and Boltzmann distributions behave differently for different temperature. The Boltzmann population (second table) has the highest population in its lowest state, and decreasing exponentially from there, while the Saha population has its most populated stage shifted from the ground stage for higher temperatures.

To see why this is, we look at equation (1) and equation (2). We see that the Boltzmann law has a negative exponential dependence on temperature, explaining why for increasing s, $n_{r,s}/N_r$ decreases exponentially. For the Saha law, the main difference is the factor that is proportional to $\propto T^{3/2}$. This causes the "peak" of the exponential to be shifted at higher temperatures.

One can wonder why ionization can fully deplete a stage, even though excitation puts only a few atoms in levels below the ionization level. I am not sure why, but looking at the Saha law, the only parameter I see that could cause equality between high-level and next-ion population at a given temperature is the electron density N_e.

2.3 Payne curves for schadeenium

The underlying premise of Payne's analysis, was that one can think, that if TE holds in a stellar atmosphere, the expected strength of a spectral line involving level (r, s) scales with the Saha-Boltzmann prediction for the lower level population $n_{r,s}$, even if you don't know how the spectral lines are formed.

We follow the premise by plotting curves like figure 6 in reference [1] for various levels of neutral and ionization stages of E.

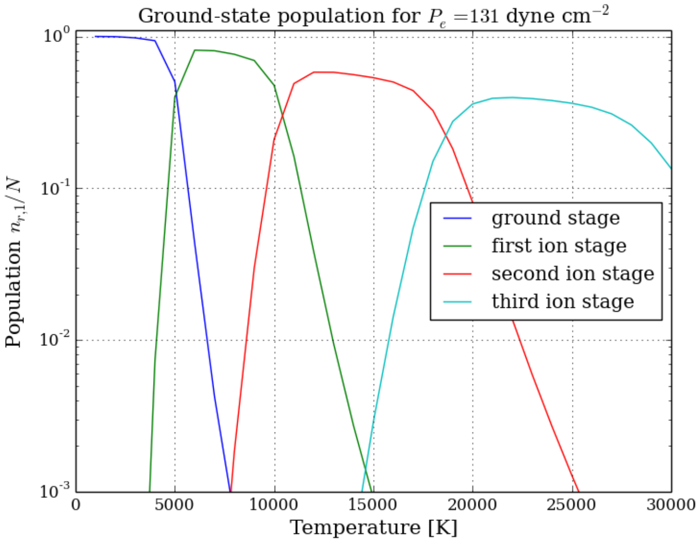


Figure 3: Populations for different ionization stages for ground state schadeenium $(s = 1)$

which looks like a Payne-like graph. We see that there are steep flanks on the left and the right side of each peak. This is caused by that there is only a specific temperature range where the specific ionization degree dominates the total population.

For $T \downarrow 0$, you would assume that most of the population would not be ionized at all, so that $n_{1,1}/N_r \to 1$, which is what we see in the graph. For $T \uparrow \infty$, we would assume that the element should be fully ionized.

We can make our plot more like what Payne did by adding curves for higher values of s to the plot.

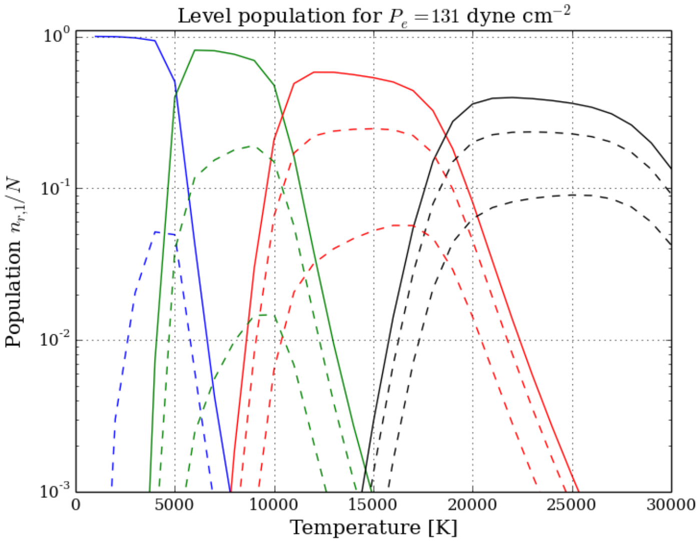


Figure 4: Population curves with higher values of s plotted with dashed lines and same color as the ground state population for the specific degree of ionization.

Looking at figure 4, we see that the curves of higher s never overlaps the lower levels, so that the population never exceeds the lower levels. This makes sense, as it requires more energy for an electron to be excited to a higher level s, and therefore less should be excited to said level.

If we were to look at an element with lower/higher ionization energies than E has, we would expect the populations to have even steeper/less steep flanks on the left and right side of the peak, and that the difference between the higher values of s to be larger/smaller.

2.4 Saha-Boltzmann populations for hydrogen

Now that we have followed in Payne's footsteps, we have (somewhat) confirmed Payne's conclusion that the Harvard classification of stellar spectra is primarily an ordering with temperature, controlled by Saha-Boltzmann population statistics.

Since we were only dealing with an hypothetical element E, the natural continuation would be to apply this to actual elements. Luckily, we can write an exact Saha-Boltzmann routine for hydrogen easily.

- Its ionization energy is $\chi_1 = 13.598$ eV.
- The statistical weights $g_{r,s}$ and level energies $\chi_{r,s}$ are given by

$$g_{1,s} = 2s^2$$
$$\chi_{1,s} = 13.598(1 - 1/s^2)\ \text{eV}$$

- Single stage ion stage (bare protons) has $U_2 = g_{2,1} = 1$

2.5 Solar Ca$^+$K versus Hα: line strength

Figure 5 (reference [1], p. 10) shows that in solar-type stars the Balmer lines become much weaker than the Ca$^+$K $s = 2 - 1$ line at $\lambda = 3933.7$ Å. Even the principal line of the Balmer sequence, Hα ($\lambda = 6563$ Å) gets much weaker than Ca$^+$K in solar-type stars. Figure 9 (reference [1], p. 22) shows how the Ca$^+$K line has a much wider absorption line than Hα.

It seems weird that the solar Ca$^+$K line is much stronger than the Hα line, even though hydrogen is not ionized in the solar photosphere and low chromosphere, where these lines are formed, and even though the solar Ca/H abundance ratio is only $N_{\rm Ca}/N_{\rm H} = 2 \cdot 10^{-6}$. Why this is the case I'm not fully certain, but since hydrogen has less possible other states that its electron can be in before ionization. Ca$^+$ has more possible configurations of its electrons, and possibly causes slight variance in the wavelength, causing a broadening.

We can compute the expected strength ratio of these two lines as function of temperature for $P_e = 10^2$ dyne cm^{-2}, by combining the actual calcium ionization energies with the schadeenium level structure we created earlier. Since the Ca$^+$K line originates from the ground state, the higher levels are only needed for the partition function, which is approximated within a factor of two by the routines we made for schadeenium.

Doing this, gives us the following plot for the expected strength ratio between the Ca$^+$K line and the Hα line.

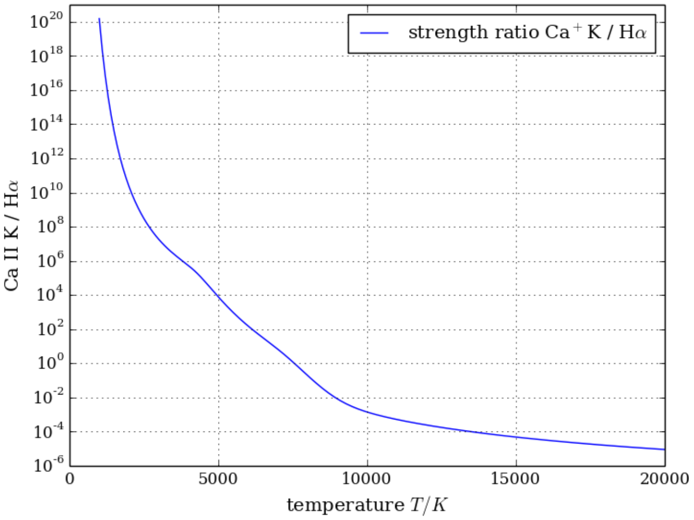


Figure 5: Plot of the expected strength ratio between Ca$^+$K line and Hα line, plotted with logarithmic y-axis

From figure 5 we see that the expected strength ratio decreases exponentially as temperature increases. For temperature $T = 5000$K, we find that the strength ratio is $N_{\rm Ca}/N_{\rm H} \approx 7840$, which shows that the Ca$^+$K line is a lot stronger than the Hα line at Sun-like temperatures.

2.6 Solar Ca$^+$K versus Hα: temperature sensitivity

Seeing that the lines differ much in their temperature sensitivity, we would be interested in plotting this. We can do this by plotting the relative population changes $(\Delta n_{\mathrm{Ca}}/\Delta T)/n_{\mathrm{Ca}}$ and $(\Delta n_{\mathrm{H}}/\Delta T)/n_{\mathrm{H}}$ for the two lower levels as function of temperature for a small temperature change ΔT. This yields the following plot.

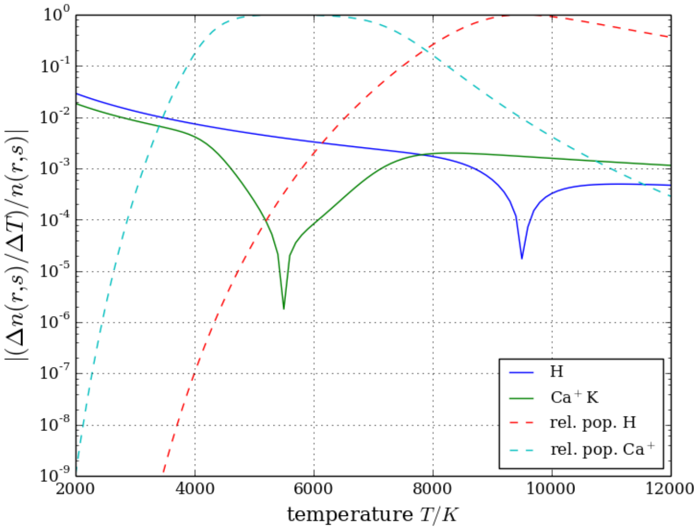


Figure 6: Plotting the relative population changes for the two lower levels of Ca and H as a function of temperature for a small temperature change ΔT. Overplotted is the relative population variation $n_{\mathrm{Ca}}/\max(n_{\mathrm{Ca}})$ and similarly for hydrogen.

In figure 6, we see that around $T = 5\,600$K, the Ca$^+$K curve has a dip, so that for that specific temperature, the temperature sensitivity of Ca$^+$K is much smaller than the temperature sensitivity of Hα. For around $T = 9\,500$K, the situation is reversed, and the Hα line has a dip. The left flank of each dip has $\Delta n < 0$, while the right flank has $\Delta n > 0$.

We can study the dips closer by overplotting the variation with temperature of each population in relative units. This is shown in figure 6 by the dashed lines. The first thing to note, is that the peaks of the relative populations coincide perfectly with the dips in the relative population changes.

The flank to the left of the peak of the population curves means that the populations is shifting from a lower ionization stage (for Ca) to Ca$^+$ as temperature is increasing. For hydrogen the fraction of atoms in the second level increases as we increase T. Similarly, increasing T further beyond the plateau, means that another energy state/ionization stage takes dominance over the relative population.

The dips in the temperature sensitivity curves means that at the lowest point in the dips, the spectral lines are at their strongest there.

2.7 Hot stars versus cool stars

One of the things that can be used to distinguish between “hot” and “cool” stars, is the temperature at which the hydrogen in stellar photospheres with $P_e = 10^2$ is about 50% ionized. This transition represents a dividing line between the mechanisms that cause the stellar continuum. In cool stars, there are few free electrons (none from hydrogen), and interactions between the few free electrons and neutral hydrogen atoms, combining them momentarily into H^- ions, dominate the formation of the stellar continuum. We can easily make a plot of the population of the ground state, as a function of the temperature.

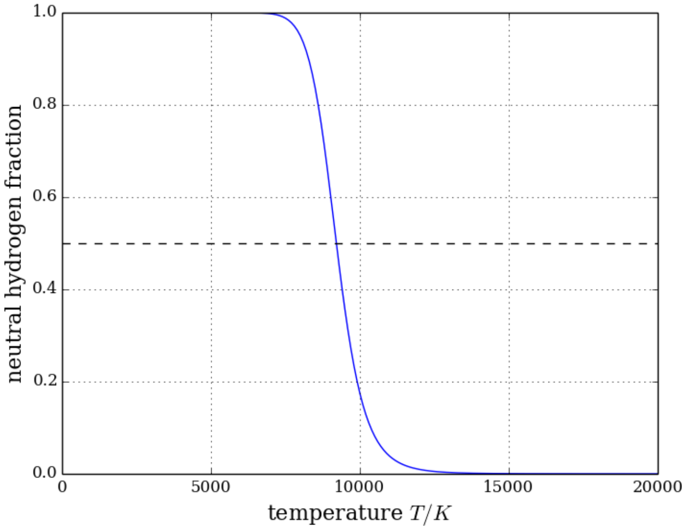


Figure 7: Plot of the fraction of ground state population versus the temperature. Overplotted, a dashed line at $y = 0.5$ to more easily see where the fraction reaches 0.5.

We see from figure 7 that as temperature increases, the fraction of neutral hydrogen decreases quite rapidly as T nears 10^3K. The temperature at which hydrogen is about 50% ionized is at $T = 9\,212\,\mathrm{K}$.

3 Fraunhofer line strengths and the curve of growth

In this section we will look at how spectral lines are formed. To adress this issue, we will look at some of the concepts introduced by Marcel Minnaert, many of which are still in use today.

We will in this task attempt to simulate a radiating body, to see how it compares to observations.

3.1 The Planck law

For electromagnetic radiation the counterparts to the material Saha and Boltzmann distributions are the Planck law and its relatives. The Planck function specifies the radiation intensity emitted by a gas or a body in TE as

$$B_\lambda(T) = \frac{2hc^2}{\lambda^5} \frac{1}{e^{hc/\lambda kT} - 1} \tag{3}$$

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

We can plot this against wavelength for different temperatures to see how the Planck curves behave.

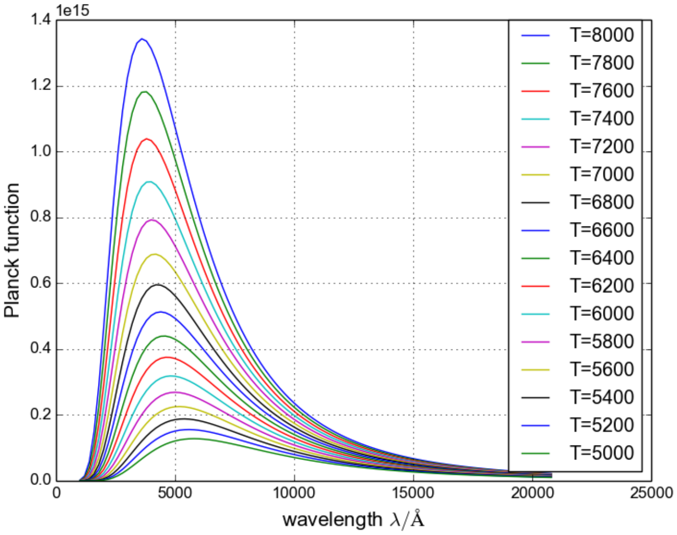


Figure 8: Planck curves plotted against wavelength in the visible spectrum for different stellar-like temperatures, linear scale

We see from figure 8 that the height of the peak increases steeply with temperature. We see that for shorter wavelengths, $B_\lambda(T)$ increases much faster (exponentially, Wien regime), than for longer wavelengths (linearly, Rayleigh-Jeans regime). The peak divides the two regimes, and shifts to shorter wavelengths for higher temperatures, as per Wien's displacement law.

We can plot with logarithmic axis.

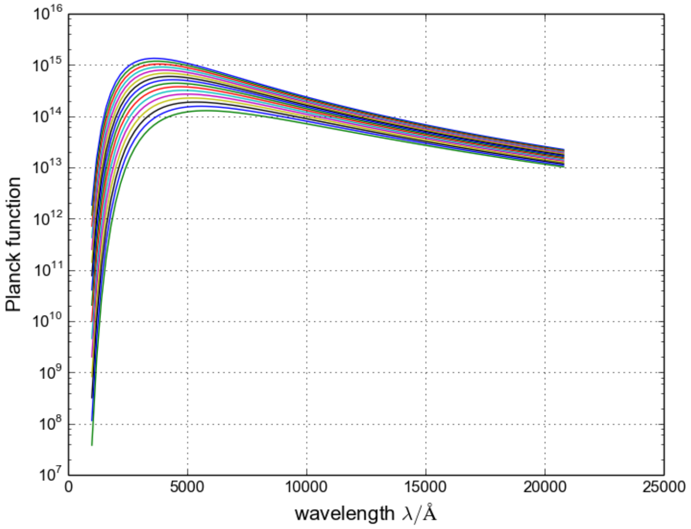


Figure 9: Plot of the Planck curves, logarithmic y-axis

We see that in figure 9, the y-axis is logarithmic, and looks about how we would expect from figure 8 when plotted with a logarithmic y-axis. The slope on the righthand side of the peak increases linearly towards the peak.

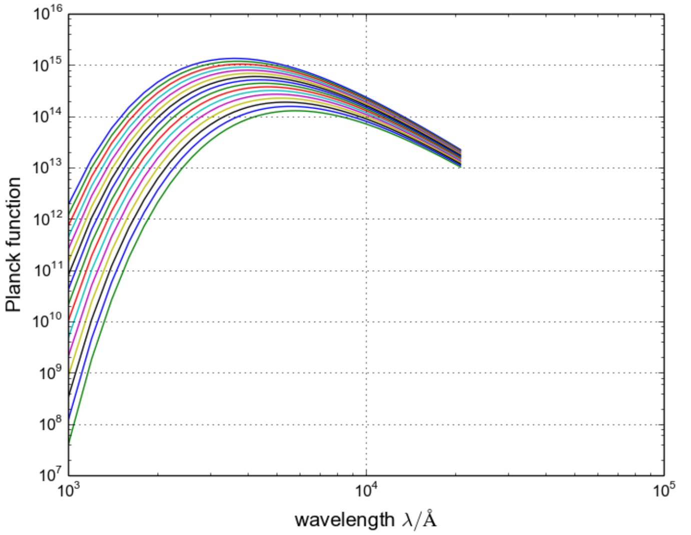


Figure 10: Plot of the Planck curves, logarithmic x- and y-axis

In figure 10, both axis are logarithmic.

3.2 Radiation through an isothermal layer

Just the radiation produced by a gas of temperature T is not enough, we also need the amount of absorption.

We look at a beam of radiation passing through a layer of gas, where it is attenuated. The weakened beam emerging has an intensity given by

$$I = I(0)\, e^{-\tau}$$

where τ is the decay parameter specifying the attenuation by absorption in the layer. It is also called optical thickness/opaqueness.

We add the radiation that originates within the layer itself. This local contribution at a location x within the layer is subsequently attenuated by the remainder of the layer to the right. The total emergent intensity is then given by

$$I_\lambda = I_\lambda(0)e^{-\tau} + \int\limits_0^\tau B_\lambda[T(x)]\, e^{-(\tau-\tau(x))}\, d\tau(x)$$

for an isothermal layer, T and therefore $B_\lambda(T)$ is independent of x, so

$$\begin{aligned} I_\lambda &= I_\lambda(0)e^{-\tau} + B_\lambda e^{-\tau} \int\limits_0^\tau e^{\tau(x)} d\tau(x) \\ &= I_\lambda(0)e^{-\tau} + B_\lambda e^{-\tau} \underbrace{\left[e^{\tau(x)}\right]_0^\tau}_{e^\tau - 1} \\ &= I_\lambda(0)e^{-\tau} + B_\lambda e^{-\tau}(e^\tau - 1) \end{aligned}$$

so that the emergent intensity simplifies to

$$I_\lambda = I_\lambda(0) + B_\lambda\,(1 - e^{-\tau}) \tag{4}$$

Now that we have an expression for the emergent intensity, we can make a plot of I_λ for given values B_λ and $I_\lambda(0)$ against τ

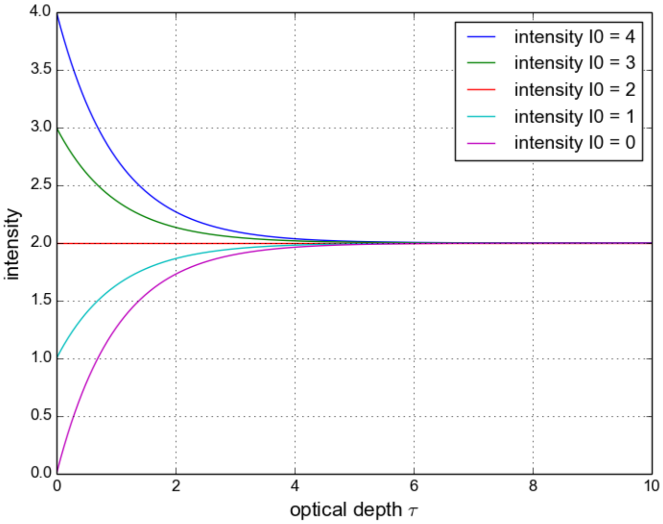


Figure 11: Plot of the emergent intensity I_λ for given values $B_\lambda = 2$ and $I_\lambda(0)$ against τ.

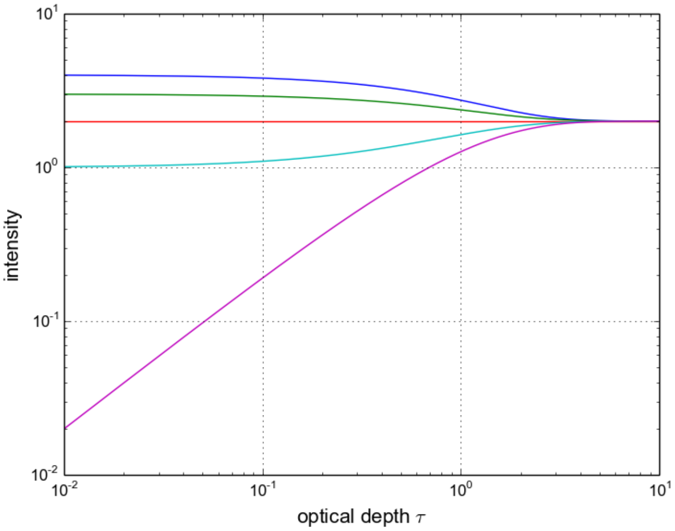


Figure 12: Plot of the emergent intensity for $B_\lambda = 2$ with logarithmic x,y axis

From looking at figure 11 and figure 12, we can see how I_λ depends on τ.

- For $\tau \ll 1$, illustrated well in figure 12, there is very little change in the intensity of the beam as it passes through the layer. For $I_\lambda(0) = 0$, as we increase τ, the emergent intensity increases steadily towards the value of B_λ. For $I_\lambda(0) > B_\lambda$, I_λ decreases as τ is increased. A layer where $tau \ll 1$ is called "optically thin", since the beam goes through the layer mostly unchanged.
- For $\tau \gg 1$, we see in figure 11 that independently of the initial intensity, the emergent intensity I_λ converges to the radiation contributed by the layer B_λ. Such a layer is called "optically thick", as the beam going through the layer will be fully absorbed by the layer, so that the radiation emitted by the layer is all we see when we look at it.

3.3 Spectral lines from a solar reversing layer

We will now apply the above result for an isothermal layer to a simple model in which the Fraunhofer lines in the solar spectrum are explained by a "reversing layer".

Schuster-Schwarzschild model

In this model, we have a few basic assumptions

- The continuous radiation, without spectral lines, is emitted by the stellar surface and irradiates a separate layer with the intensity
$$I_\lambda(0) = B_\lambda(T_{\text{surface}}) \tag{5}$$
- This layer sits as a shell around the star and causes attenuation and local emission only at the wavelengths of spectral lines.
- The star is optically thick ($\tau \gg 1$), so that the surface radiates with the $\tau \gg 1$ solution $I_\lambda = B_\lambda(T_{\text{surface}})$ of equation (4). Depending on atom concentration, the shell may be optically thin or thick.
- The line-causing atoms in the shell have temperature T_{layer} so that the local production of radiation in the layer at the line wavelengths is given by $B_\lambda(T_{\text{layer}})\Delta\tau(x)$.

The emergent radiation at the line wavelengths is then given by combining (4) and (5)

$$I_\lambda = B_\lambda(T_{\text{surface}})\, e^{-\tau_\lambda} + B_\lambda(T_{\text{layer}})\,(1 - e^{-\tau_\lambda}) \tag{6}$$

Voigt profile

The opaqueness τ in (6) has an index λ because it varies over the spectral line. We will be looking at the broadening distribution caused by Coulomb interactions by neighboring particles. This broadening distribution is described by

$$\tau(u) = \tau(0)\, V(a, u) \tag{7}$$

where V is called the Voigt function and u measures the wavelength separation $\Delta\lambda = \lambda - \lambda(0)$ from the center of the line at $\lambda = \lambda(0)$ in dimensionless units

$$u \equiv \Delta\lambda / \Delta\lambda_{\rm D}$$

where $\Delta\lambda_{\rm D}$ is the Doppler width

$$\Delta\lambda_{\rm D} \equiv \frac{\lambda}{c}\sqrt{2kT/m}$$

The Voigt function is defined as

$$V(a,u) \equiv \frac{1}{\Delta\lambda_{\rm D}\sqrt{\pi}}\,\frac{a}{\pi}\int_{-\infty}^{+\infty}\frac{e^{-y^2}}{(u-y)^2+a^2}dy \tag{8}$$

The Voigt function represents the convolution of a Gauss profile with a Lorentz profile, and therefore has a Gaussian shape close to line center. We can find a reasonable approximation to the full Voigt function

$$V(a,u) \approx \frac{1}{\Delta\lambda_{\rm D}\sqrt{\pi}}\left[e^{-u^2} + \frac{a}{\sqrt{\pi}u^2}\right] \tag{9}$$

We can plot the Voigt function (8) for $u = -10$ to $u = +10$ and for a ranging between $a = 0.001$ and $a = 1$

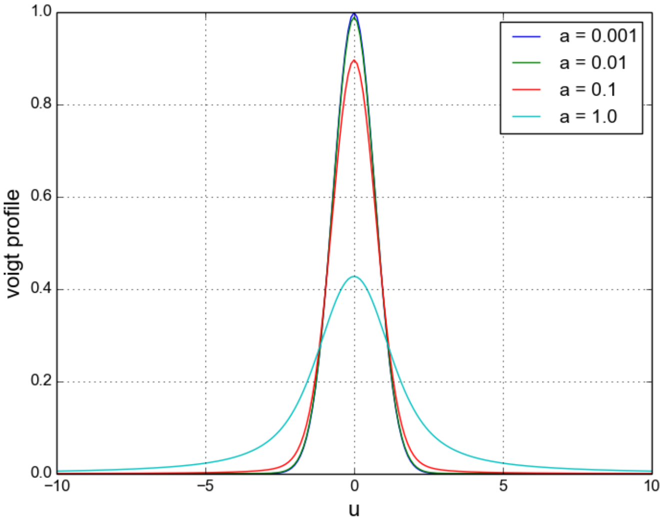


Figure 13: Plot of the Voigt function (8) for $u = -10$ to $u = 10$ and $a = 0.001$ to $a = 1$

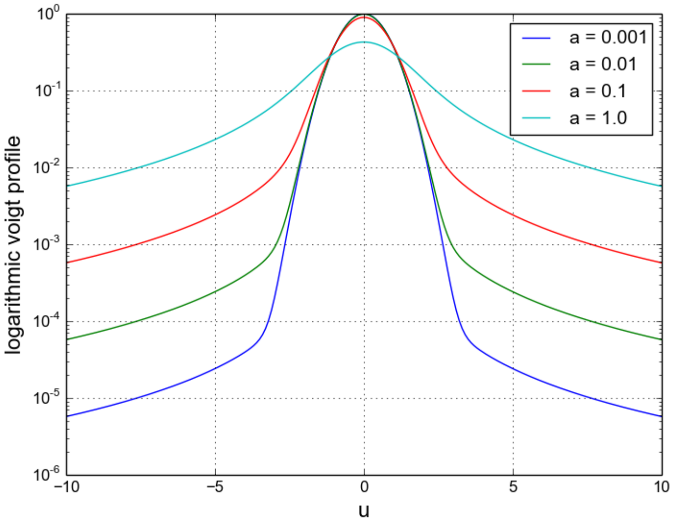


Figure 14: Plot of the Voigt function (8) with logarithmic y-axis

We inspect the far wings of the profile by looking at figure 14. For u far from the center of the profile $u = 0$, we see in our approximation, eq. (9), that the second term $(a/\sqrt{\pi}u^2)$ is proportional to a, so increasing a would make this term larger, resulting in extended/wider wings, which is what we see in the plot.

Emergent line profiles

We can now compute and plot stellar spectral line profiles by combining (6) and (7). Again using dimensionless u units for the wavelength scale, thereby avoiding having to calculate the Doppler width $\Delta\lambda_{\mathrm{D}}$.

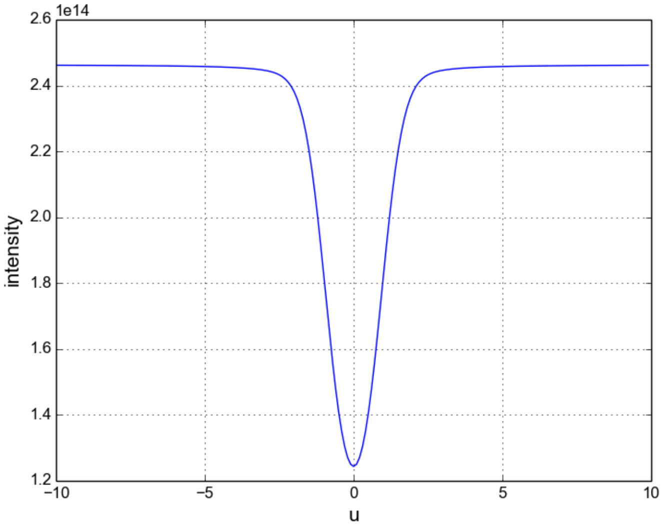


Figure 15: Plot of profile I against u for $\tau(0) = 1$, $T_{\text{surface}} = 5600$ K, $T_{\text{layer}} = 4200$ K, $a = 0.1$, $\lambda = 5000$ Å

Then we study the appearance of the line in the spectrum as a function of $\tau(0)$ over the range $\log \tau(0) = -2$ to $\log \tau(0) = 2$.

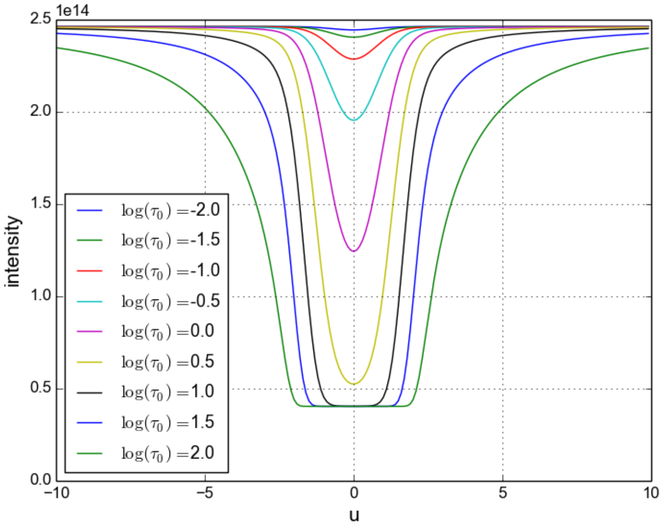


Figure 16: Plot of I against u for a range of $\tau(0)$, $T_{\text{surface}} = 5600$ K, $T_{\text{layer}} = 4200$ K, $a = 0.1$, $\lambda = 5000$ Å

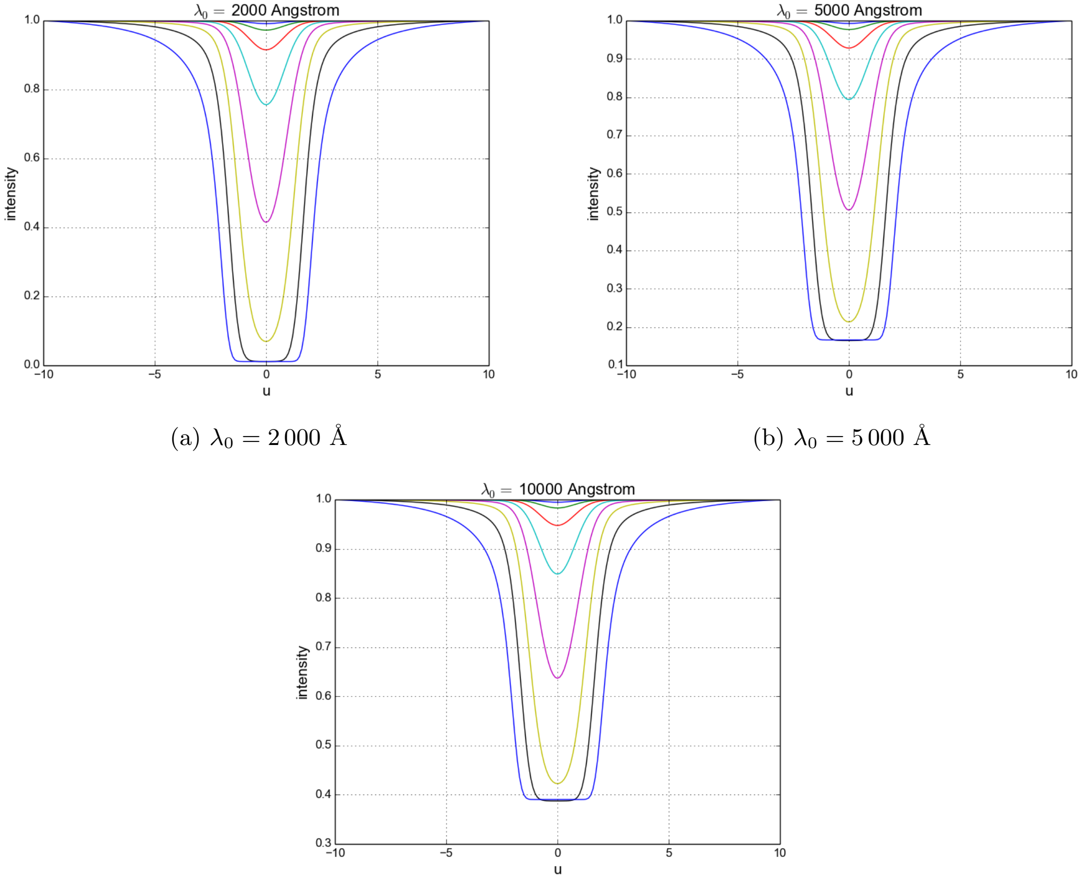
Looking at figure 16, we can hopefully unveil some of the properties of the line profiles.

For $\tau \ll 1$, the profiles are very weak absorption lines. For example, for $\log \tau(0) = -2$, the profile is almost a straight line, indicating that the reversing layer is optically thin, and $\tau_\lambda \ll 1$ is satisfied for all λ. Increasing $\tau(0)$ slightly, we see that the profile gets deeper, and the wings slowly get wider.

This continues until the profile reaches a low-intensity saturation limit for $\tau \gg 1$. When $\tau(0)$ gets large, we recall from earlier that the emergent radiation ends up following the radiation emitted by the layer itself. Due to this, there is a limit to how low the intensity can get in the center of the spectral line for an optically thick layer, since we will at some point start seeing the radiation from the layer.

The line wings only develop for very large $\tau(0)$, as τ_λ will decrease the further out from $u = 0$ we go, but much more slowly than for lower $\tau(0)$, so there will be a natural broadening of the line due to this.

We can attempt to study the dependence of these line profiles on wavelength by repeating the above for $\lambda = 2\,000$ Å, and $\lambda = 10\,000$ Å.



(a) $\lambda_0 = 2\,000$ Å

(b) $\lambda_0 = 5\,000$ Å

(c) $\lambda_0 = 10\,000$ Å

Figure 17: Profile shapes for various wavelengths

The figures in figure 17 is already plotted with scaled to the local continuum intensity $I_\lambda / I_{\text{cont}}$, so unless new plots were to be made without the scaling, it is hard to tell much about the continuum intensity.

We see from figure 17, that increasing the wavelength causes ratio between the limit value and continuum value increases.

3.4 The equivalent width of spectral lines

The profile plots in the last section demonstrate that the growth of the absorption feature in the spectrum for increasing $\tau(0)$ is faster for small $\tau(0)$ then when it saturates for larger $\tau(0)$. From Minnaert, we introduce the equivalent width W_λ as a line-strength parameter to measure this growth quantitatively.

$$W_\lambda \equiv \int \frac{I_{\rm cont} - I(\lambda)}{I_{\rm cont}} d\lambda$$

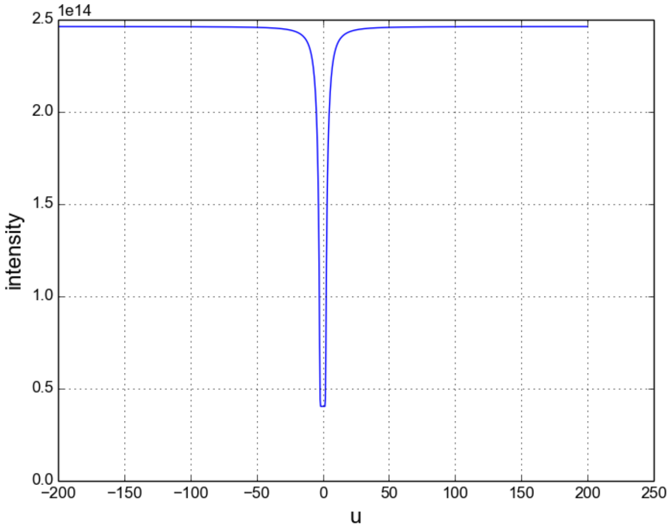


Figure 18: Plot of equivalent width line depth

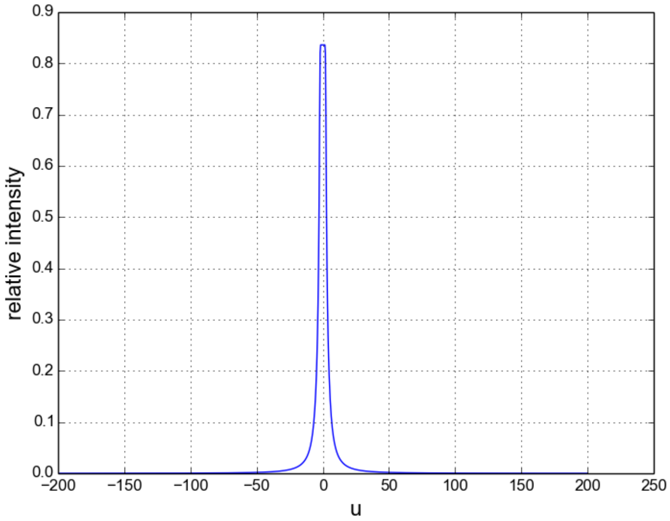


Figure 19: Line depth in relative units

We compute the equivalent width from figure 19, and find the equivalent width to be $W_\lambda = 7.54$.

We can check how this matches by integrating the plot in figure 18, which I haven't done.

3.5 The curve of growth

The idea behind equivalent width,is that the amount of spectral blocking should be a direct measure of the number of atoms in the reversing layer. Our profile plots illustrate that the profile growth is only linear with $\tau(0)$ for $\tau(0) \ll 1$. To show the full dependence, we plot the "curve of growth": the growth of the line strength with the line-causing particle density.

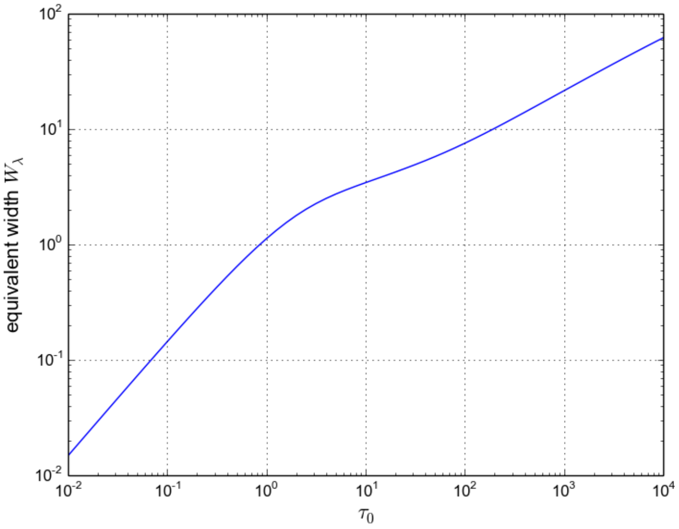


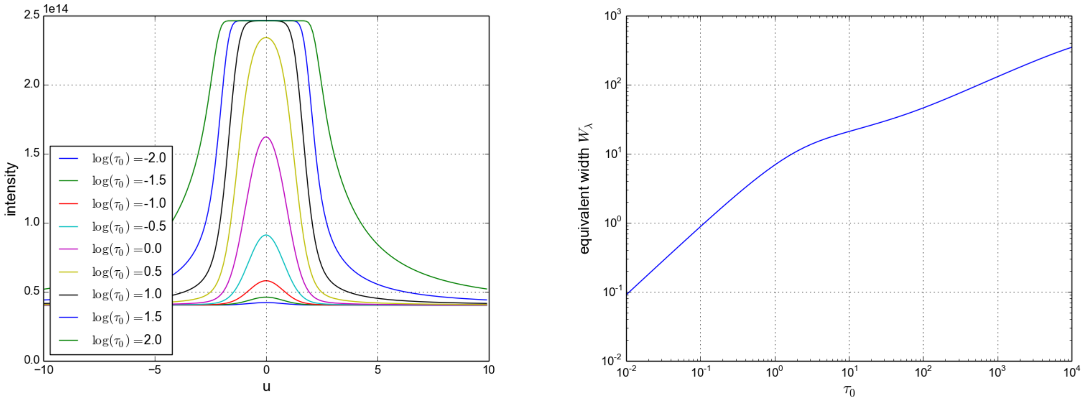
Figure 20: Curve of growth

We see that figure 20 is split into three regimes:

- Before saturation: Low absorption, small optical depth
- "During" saturation: Doppler wings barely change for a while
- After saturation: Collisional broadening takes over eventually

It makes a fair bit of sense that the curve grows fastest in the first part, as the layer is optically thin, so equivalent width of the line increases fast. Not sure which parameter that controls the location of the onset of the third part.

If we want to make emission lines instead of absorption lines, then we should switch the temperatures of the surface T_{surface} and layer T_{layer}.



(a) Emission profile

(b) Emission-line curve of growth

Figure 21: Plot of emission profiles and an emission-line curve of growth

We see that the emission profiles are pretty much just the same as the absorption profiles inverted, which is to be expected. The curve of growth remains unchanged, which is not very surprising as the equivalent width of an emission line should be the same as an absorption line under similar situations.

4 Appendix

4.1 Saha-Boltzmann tables

```
5000
U [ 1.10887454  1.10887463  1.10887463  1.10887463]
10000
U [ 1.45590251  1.45633443  1.45633445  1.45633445]
20000
U [ 2.23242942  2.27133839  2.27154946  2.27154949]
5000
s = 1 : 9.018e-01
s = 2 : 8.854e-02
s = 3 : 8.694e-03
s = 4 : 8.536e-04
s = 5 : 8.381e-05
s = 6 : 8.229e-06
s = 7 : 8.080e-07
s = 8 : 7.933e-08
s = 9 : 7.789e-09
s = 10 : 7.648e-10
10000
s = 1 : 6.869e-01
s = 2 : 2.152e-01
```

```
s = 3 : 6.744e-02
s = 4 : 2.113e-02
s = 5 : 6.622e-03
s = 6 : 2.075e-03
s = 7 : 6.501e-04
s = 8 : 2.037e-04
s = 9 : 6.383e-05
s = 10 : 2.000e-05
20000
s = 1 : 4.479e-01
s = 2 : 2.507e-01
s = 3 : 1.404e-01
s = 4 : 7.857e-02
s = 5 : 4.398e-02
s = 6 : 2.462e-02
s = 7 : 1.378e-02
s = 8 : 7.714e-03
s = 9 : 4.318e-03
s = 10 : 2.417e-03
5000
r = 1 : 8.796e-02
r = 2 : 9.120e-01
r = 3 : 8.019e-09
r = 4 : 5.357e-32
r = 5 : 4.474e-75
10000
r = 1 : 7.480e-07
r = 2 : 1.480e-01
r = 3 : 8.520e-01
r = 4 : 1.352e-07
r = 5 : 2.455e-24
20000
r = 1 : 7.288e-16
r = 2 : 4.817e-08
r = 3 : 1.689e-02
r = 4 : 9.827e-01
r = 5 : 4.594e-04
```

4.2 Code

4.2.1 For schadeenium

```
from pylab import *
import numpy as np
import astropy as ap
from astropy import units
```

```
rc('font',**{'family':'serif'})

def partfunc_E(temp,chiion):
    '''
    Calculates the partition function of Schadeenium for a given temperature
    '''
    temp *= units.m   # to remove error cause astropy
    # note to self, make sure to work dimensionless inside functions

#     chiion = [7, 16, 31, 51]  # Schaade ionization energies
    k = ap.constants.k_B # Boltzmann constant in J/(K)
    U = zeros(4)
    for r in range(4):
        for s in range(0, (chiion[r]).value):
            U[r] += exp(-s/((k.to('eV/K')).value*temp.value))

    return U


###########################

def boltz_E(temp,r,s,chiion):
    '''
    Computes Boltzmann distribution
    '''
    temp *= units.m   # again fixing problems caused by my astropy usage
    chiion *=units.eV # cause astropy problems

    U_r = partfunc_E(temp,chiion)
    k = 8.61734e-5    # Boltzmann constant in eV/deg

    relnrs = 1./U_r[r-1]*exp(-(s-1)/(k*temp.value))

    return relnrs

############################

def saha_E(temp,elpress,ionstage,chiion):
    # Some constants
    m_e = ap.constants.m_e # mass of electron in kg
    h = ap.constants.h    # Planck constant in J s
    k = ap.constants.k_B # Boltzmann constant in J/(K)

    # kT and electron density
    kevT = k.to('eV/K')*temp
    kergT = k.to('erg/K')*temp
    eldens = elpress/kergT

    # Partition function
#     chiion = [7, 16, 31, 51]*units.eV  # Schaade ionization energies in eV
    chiion *=units.eV # cause astropy problems

    u = partfunc_E(temp,chiion)
    u = np.append(u,2.0)

    # Saha constant
    sahaconst = (2*pi*m_e.to('g')*kergT/(h.to('erg*s')**2))**1.5*2./eldens
    # N(r)
    nstage = zeros(5)
    nstage[0] = 1

    for r in range(4):
        nstage[r+1] = (nstage[r]*sahaconst*u[r+1]/u[r]*exp(-(chiion[r]).value/kevT.value)).value
    ntotal = sum(nstage)
    nstagerel = nstage/ntotal

    return nstagerel[ionstage-1]


##########################################
def sahabolt_E(temp,elpress,ion,level,chiion):
    '''
    Computes Saha-Boltzmann population n_(r,s)/N for level r,s of E
    '''
    return saha_E(temp,elpress,ion,chiion)*boltz_E(temp,ion,level,chiion)


if __name__=='__main__':
    task = '2.5a'


    chiion = array([7,16,31,51])*units.eV
    temp_list = [5000,10000,20000]

    for temp in temp_list:
        print temp
        U = partfunc_E(temp,chiion)
        print 'U', U

    for temp in temp_list:
        print temp
        for s in range(1,11):
            print 's = %d : %.3e'%(s, boltz_E(temp,1,s,chiion))

    for temp in temp_list:
        print temp
        for r in range(1,6):
            print 'r = %d : %.3e'%(r, saha_E(temp*units.K,1e1,r,chiion))


    if task=='2.5':
        temp = np.arange(0,30001,1000)*units.K
        pop = zeros((5,31))

        '''
        for T in np.arange(1,31):
            for r in np.arange(1,5):
                pop[r,T] = sahabolt_E(temp[T],131.,r,1)

        # ground-state plot
        figure()
        for i in range(1,5):
            plot(temp,pop[i,:])
        title(r'Ground-state population for $P_e = 131$ dyne cm${}^{-2}$',size=15)
        xlabel(r'Temperature [K]',size=15)
        ylabel(r'Population $n_{r,1}/N$',size=15)
        yscale('log')
        grid('on')
        ylim([1e-3,1.1])
        legend(['ground stage','first ion stage','second ion stage','third ion stage'],loc='best')
        savefig('schadeenium_pop.png')
        show()
        '''

        # adding more lines
        #xkcd()
        figure()
        grid('on')
        colors = ['b','g','r','k']
        lines = ['-','--','--']
        s = [1,2,4]
        chiion = [7, 16, 31, 51]
        for j in range(len(s)):
            for T in np.arange(1,31):
                for r in np.arange(1,5):
                    pop[r,T] = sahabolt_E(temp[T],131.,r,s[j],chiion)

            # plot
            for i in range(1,5):
                plot(temp,pop[i,:],lines[j]+colors[i-1])
        title(r'Level population for $P_e = 131$ dyne cm${}^{-2}$',size=15)
        xlabel(r'Temperature [K]',size=15)
        ylabel(r'Population $n_{r,1}/N$',size=15)
        yscale('log')
        ylim([1e-3,1.1])
#        legend(['ground stage','first ion stage','second ion stage','third ion stage'],loc='best')
        savefig('schadeenium_pop_higher.png')
        show()
```

4.2.2 For hydrogen

```
from pylab import *
import numpy as np
import astropy as ap
from astropy import units
rc('font',**{'family':'serif'})

def sahabolt_H(temp,elpress,level):
    '''
    Computes Saha-Boltzmann population n_(r,s)/N for level r,s of H
    '''
    m_e = ap.constants.m_e # mass of electron in kg
    h = ap.constants.h    # Planck constant in J s
    k = ap.constants.k_B # Boltzmann constant in J/(K)


    # kT and electron density
    hergs = h.to('erg s')
    kevT = k.to('eV/K')*temp
    kergT = k.to('erg/K')*temp
    elmass = m_e.to('g')
    eldens = elpress/kergT

    # energy levels and weights for hydrogen
    nrlevels = 100                # reasonable partition function cut-off value
    g = zeros((2,nrlevels))       # declaration weights (too many for protons)
    chiexc = zeros((2,nrlevels))  # declaration excit. energies

    for s in range(nrlevels):
        g[0,s] = 2*(s+1)**2                     # statistical weights
        chiexc[0,s] = 13.598*(1-1./(s+1)**2)    # excitation weights

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

    # partition functions
    u = zeros(2)
    for s in range(nrlevels):
        u[0] += g[0,s]*exp(-chiexc[0,s]/kevT.value)
    u[1] = g[1,0]

    # Saha
    sahaconst = (2*pi*elmass.value*kergT.value / (hergs.value*hergs.value))**(1.5)*2./eldens.value

    nstage = zeros(2)
    nstage[0] = 1.
    nstage[1] = nstage[0]*sahaconst*u[1]/u[0] * exp(-13.598/kevT.value)
    ntotal = sum(nstage)            # sum both stages = total hydrogen density

    # Boltzmann
    nlevel = nstage[0]*g[0,level-1]/u[0]*np.exp(-chiexc[0,level-1]/kevT.value)
    nlevelrel = nlevel/ntotal       # fraction of total hydrogen density


    # Print for checking
    '''
    for s in range(6):
        print "%2d %5d %7.4f %9.7e" % (s+1, g[0,s], chiexc[0,s],
                                      g[0,s]*np.exp(-chiexc[0,s]/kevT.value))

    print
    for s in range(0,nrlevels,10):
        print "%2d %5d %6.4f %8.7e" %(s+1, g[0,s], chiexc[0,s],
                                     g[0,s]*np.exp(-chiexc[0,s]/kevT.value))
    '''

    return nlevelrel
```

4.2.3 For Ca and Halpha line

```
from pylab import *
```

```
import numpy as np
import astropy as ap
from astropy import units
rc('font',**{'family':'serif'})

from ssa2 import *
from ssa_hydrogen import *

if __name__=='__main__':
    temp = np.arange(2000,12001,100)
    dNCadT = np.zeros(temp.shape)
    dNHdT = np.zeros(temp.shape)
    dT = 1.
    chiion = np.array([6,11,50,67])

    for i in range(101):
        NCa = sahabolt_E(temp[i],1e2,2,1,chiion)
        NCa2 = sahabolt_E(temp[i]-dT,1e2,2,1,chiion)
        dNCadT[i] = (NCa - NCa2)/(dT*NCa)
        NH = sahabolt_H(temp[i],1e2,2)
        NH2 = sahabolt_H(temp[i]-dT,1e2,2)
        dNHdT[i] = (NH-NH2)/(dT*NH)

    # Plotting
    figure()
    plot(temp,np.absolute(dNHdT), label=r'H')
    plot(temp,np.absolute(dNCadT), label=r'Ca$^+$K')
    yscale('log')
    ylim(1e-9,1)
    grid('on')
    xlabel(r'temperature $T/K$', size=15)
    ylabel(r"$\left| \left( \Delta n(r,s) / \Delta T \right) / n(r,s) \right|$", size=18)
#     show()

    NCa = np.zeros(temp.shape)
    NH = np.zeros(temp.shape)
    for i in range(101):
        NCa[i] = sahabolt_E(temp[i],1e2,2,1,chiion)
        NH[i] = sahabolt_H(temp[i],1e2,2)

    plot(temp,NH/np.amax(NH),ls='--',label='rel. pop. H')
    plot(temp,NCa/np.amax(NCa),ls='--',label=r'rel. pop. Ca$^+$')
    legend(loc=4, fontsize=12)
    savefig('temp_sens_Ca_Halpha.png')
    show()
```

```
from pylab import *
import numpy as np
import astropy as ap
from astropy import units
rc('font',**{'family':'serif'})

from ssa2 import *
from ssa_hydrogen import *
```

```
for T in arange(2e3,2e4+1,2e3):
    print T, sahabolt_H(T,1e2,1)

temp = arange(1e3,2e4+1,1e2)
nH = zeros(temp.shape)
for i in range(191):
    nH[i] = sahabolt_H(temp[i],1e2,1)

plot(temp,nH)
xlabel(r'temperature $T/K$',size=16)
ylabel(r'neutral hydrogen fraction',size=16)
grid('on')
axhline(0.5,linestyle='--',color='black')
savefig('ssa_2_10.png')
show()
```

4.2.4 Task 3

```
from pylab import *

def planck(temp, wav):
    h = 6.62607e-27  # planck in erg s
    k = 1.38065e-16  # boltzmann in erg/K
    c = 2.99792e10   # speed of light cm/s

    return 2*h*c**2/wav**5*1./(exp(h*c/(wav*k*temp))-1)

if __name__=='__main__':

    wav = arange(1000,20801,200)
    b = zeros(wav.shape)

    figure()
    xlabel(r'wavelength $\lambda / \AA$', size=15)
    ylabel(r'Planck function', size=15)
    grid('on')

    for T in range(8000,5000-1,-200):
        b[:] = planck(T, wav[:]*1e-8)
        plot(wav,b,'-',label='T=%d'%T)

    legend(loc='best')
    savefig('ssa_3_1.png')

    # log y-axis
    yscale('log')
    legend().set_visible(False)
    savefig('ssa_3_1_logy.png')

    # log-log axes
    xscale('log')
    savefig('ssa_3_1_loglog.png')
    show()
```

```
from pylab import *

B = 2.

tau = arange(0.01,10.01,0.01)
intensity = zeros(tau.shape)
for I0 in range(4,-1,-1):
    intensity[:] = I0 * exp(-tau[:]) + B*(1-exp(-tau[:]))
    plot(tau, intensity, label='intensity I0 = ' + str(I0))

xlabel(r'optical depth $\tau$', size=15)
ylabel('intensity', size=15)
legend(fontsize=14)
grid('on')
savefig('ssa_3_2.png')

# loglog
xscale('log')
yscale('log')
legend().set_visible(False)
savefig('ssa_3_2_log.png')
show()
```

```
from pylab import *
from scipy import special
import numpy as np
from ssa_3_1 import *

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

if __name__=='__main__':
    u = np.arange(-10,10.01,0.1)
    a = np.array([0.001,0.01,0.1,1])
    vau = zeros((a.shape[0],u.shape[0]))

    for i in range(4):
        vau[i,:] = voigt(a[i],u[:])
        plot(u[:], vau[i,:], label='a = ' + np.str(a[i]))

    xlim(-10,10)
    legend(fontsize=14)
    grid('on')
    xlabel('u', size=16)
```

```
ylabel('voigt profile', size=16)
savefig('ssa_voigt.png')

# ylog
ylabel('logarithmic voigt profile', size=16)
yscale('log')
savefig('ssa_voigt_ylog.png')
show()

##########################
# Schuster-Schwarzchild line profile

Ts = 5700.         # solar surface temperature
T1 = 4200.         # solar T-min temperature = 'reversing layer'
a = 0.1            # damping parameter
wav = 5000.0e-8    # wavelength in cm
tau0 = 1.          # reversing layer thickness at line center
u = arange(-10,10.0,0.1)
intensity = zeros(u.shape)

for i in range(200):
    tau = tau0 * voigt(a,u[i])
    intensity[i] = planck(Ts,wav)*exp(-tau) + planck(T1,wav)*(1.-exp(-tau))

plot(u,intensity)
grid('on')
xlabel('u', size=15)
ylabel('intensity', size=15)
savefig('ssa_3_3_1.png')
show()


logtau0 = arange(-2,2.1,0.5)
for itau in range(9):
    for i in range(200):
        tau = 10.**(logtau0[itau])* voigt(a, u[i])
        intensity[i] = planck(Ts,wav)*exp(-tau) + planck(T1,wav)*(1.-exp(-tau))
    plot(u,intensity, label=r'$\log{(\tau_0)} = $'+np.str(logtau0[itau]))

legend(loc=3, fontsize=13)
xlabel('u', size=15)
ylabel('intensity', size=15)
grid('on')
savefig('ssa_3_3_2.png')
show()

# different wavelengths
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(200):
            tau = 10.**(logtau0[itau]) * voigt(a,u[i])
            intensity[i] = planck(Ts,wav)*exp(-tau) + planck(T1,wav)*(1.-exp(-tau))
        intensity = intensity/intensity[0]
        plot(u,intensity[:], linewidth=1.)
    grid('on')
    xlabel('u',size=15)
```

```
        ylabel('intensity',size=15)
        title(r'$\lambda_0 =$ %d Angstrom'%(wav*1e8),size=16)
        savefig('ssa_3_3_wave_%d.png'%iwav)
        show()
```

```
from pylab import *
import numpy as np
from ssa_3_3 import *

def profile(a, tau0, u):
    Ts = 5700.
    T1 = 4200.
    wav = 5000.0e-8
    usize = u.size
    intensity = np.zeros(usize)

    for i in range(usize):
        tau = tau0 * voigt(a, u[i])
        intensity[i] = planck(Ts,wav)*exp(-tau) + planck(T1,wav)*(1.-exp(-tau))

    return intensity

if __name__=='__main__':

    # Checking the profile
    u = np.arange(-200,200.4,0.4)
    a = 0.1
    tau0 = 1.0e2
    intensity = profile(a,tau0,u)

    plot(u,intensity)
    xlabel('u',size=16)
    ylabel('intensity',size=16)
    grid('on')
    savefig('ssa_3_4_1.png')
    show()

    # relative
    reldepth = (intensity[0]-intensity)/intensity[0]
    plot(u,reldepth)
    xlabel('u',size=16)
    ylabel('relative intensity',size=16)
    grid('on')
    savefig('ssa_3_4_2.png')

    eqw = sum(reldepth)*0.4
    print eqw
    show()
```

```
from ssa_3_4 import *
```

```
u = np.arange(-200,200.4,0.4)
tau0 = np.logspace(-2,4,61)
a = 0.1
eqw = np.zeros(tau0.size)

for i in range(61):
    intensity = profile(a,tau0[i],u)
    reldepth = (intensity[0]-intensity) / intensity[0]
    eqw[i] = sum(reldepth)*0.4

plot(tau0,eqw)
xlabel(r'$\tau_0$',size=16)
ylabel(r'equivalent width $W_{\lambda}$',size=15)
xscale('log')
yscale('log')
grid('on')
savefig('ssa_3_5.png')
show()
```

```
from ssa_3_3 import *

def profile(a, tau0, u):
    Ts = 4200.
    T1 = 5700.
    wav = 5000.0e-8
    usize = u.size
    intensity = np.zeros(usize)

    for i in range(usize):
        tau = tau0 * voigt(a, u[i])
        intensity[i] = planck(Ts,wav)*exp(-tau) + planck(T1,wav)*(1.-exp(-tau))

    return intensity

if __name__=='__main__':
    ############################
    # Schuster-Schwarzchild line profile

    Ts = 4200.        # solar surface temperature
    T1 = 5700.        # solar T-min temperature = 'reversing layer'
    a = 0.1           # damping parameter
    wav = 5000.0e-8   # wavelength in cm
    tau0 = 1.         # reversing layer thickness at line center
    u = arange(-10,10.0,0.1)
    intensity = zeros(u.shape)

    logtau0 = arange(-2,2.1,0.5)
    for itau in range(9):
        for i in range(200):
            tau = 10.**(logtau0[itau])* voigt(a, u[i])
            intensity[i] = planck(Ts,wav)*exp(-tau) + planck(T1,wav)*(1.-exp(-tau))
        plot(u,intensity, label=r'$\log{(\tau_0)} = $'+np.str(logtau0[itau]))

    legend(loc=3, fontsize=13)
```

```
xlabel('u', size=15)
ylabel('intensity', size=15)
grid('on')
savefig('ssa_3_5_1.png')
show()

# curve of growth
u = np.arange(-200,200.4,0.4)
tau0 = np.logspace(-2,4,61)
a = 0.1
eqw = np.zeros(tau0.size)

for i in range(61):
    intensity = profile(a,tau0[i],u)
    reldepth = (intensity[0]-intensity) / intensity[0]
    eqw[i] = sum(reldepth)*0.4

plot(tau0,abs(eqw))
xlabel(r'$\tau_0$',size=16)
ylabel(r'equivalent width $W_{\lambda}$',size=15)
xscale('log')
yscale('log')
grid('on')
savefig('ssa_3_5_2.png')
show()
```

References

[1] Stellar spectra A: basic line formation
http://www.uio.no/studier/emner/matnat/astro/AST4310/h12/undervisningsmateriale/ssa.pdf