CHAPTER 1

AUTOMATIC FITTING OF OPTICAL TYPE IA SUPERNOVA SPECTRA - THE DALEK PROJECT

The last chapters (Chapters **??** were dedicated to the hunt for donor stars and did not use the measurements from the SN Ia-phenomenon itself. In this chapter we will describe the extraction of yields and energies from optical spectra.

The two main sources of information in spectra, are the spectra themselves as well as their time evolution. There have been a few attempts to extract the details of the stellar explosions from one or two of these sources. All of them employ the technique of fitting the spectra using synthetic spectra. One of the main parts is the radiative transfer program that creates the synthetic spectra. There are several different radiative transfer-codes in the community.

Fisher (2000) wrote a very simple radiative transfer code called SYNOW. SYNOWis a highly parametrized code and thus is mainly used for line identification rather than actual fitting of supernova spectra. It runs The main code (henceforth ML MONTE CARLO) used in this work is an evolved code of Mazzali & Lucy (1993); Mazzali (2000). Compared to the SYNOW-code the ML MONTE CARLO-code calculates a radiative equlibirum tepmerature and uses this to compute internally consistent ionization ratios. In addition ML MONTE CARLOtakes electron scattering into account as well as allowing for photon branching.

Codes such as PHOENIXHauschildt & Baron (1999), SEDONA Kasen et al. (2006) and ARTIS Kromer & Sim (2009) are powerful 3D radiative transfer codes. They are the most "physical" codes available but take hours on supercomputers to produce spectra. These codes, however, are not feasible for fitting observed spectra as they take too long for each iteration.

The main aim of this work is to automatically fit the torrent of observed spectra expected from the current and next generation of supernova searches. We opted to use the ML MONTE CARLO-code as it provides a good compromise between speed and "realism".

In section **??**we will introduce a the inner-workings of the ML MONTE CARLO-code. We will discuss the properties of the search space in **??** and will introduce our optimisation strategies in **??**. Finally we will conclude and give an outlook over future work for this unfinished project in section **??**.

1.1. The ML MONTE CARLO-Code

The supernova can be divided in two different phases: the photospheric phase and the nebular phase. The ML MONTE CARLO-code only models the photospheric phase. In this photospheric phase the supernova is treated like a sharp photosphere emitting a black-body spectrum with a fast moving layer of ejecta above that.

There are many physical processes in radiative transfer. Of those the Bound-free opacity has the biggest contribution to the final spectrum. In addition, Thompson scattering is thought to have an important effect in redistributing the flux. As ML MONTE CARLOis required to run fast only Bound-Free opacity as well as Thompson scattering is implemented in the code.

Unlike stellar atmospheres in supernova ejecta one needs to consider the photon's doppler shift in relation to the surrounding medium. One major assumption that the code makes is that of the Sobolev approximation. This means that at the interaction between photon and line resonance happens only at one specific point (thus disregarding any broadening effects to the line). For example a photon in free flight from the photosphere will be able to interact with resonance lines of lower and lower frequencies. The Sobolev approximation makes the code relatively fast

Another assumption that ML MONTE CARLOmakes is that the ejecta is in homologous expansion. This means that the velocity is a linear function of the radius:

$$v = r/t.$$

Combining both the Sobolev approximation with the assumption of homologous expansion yields this relatively simple formulat for line opacities:

$$\tau_{ul} = \frac{\pi e^2}{m_e c} f \lambda t_{\exp} n_l \left(1 - \frac{g_l n_u}{g_u n_l}\right),$$

where τ_{ul} denotes the opacity going from the u-state to the l-state, e is the electron charge, m_e is the electron mass, f is the oscilator strength of the line, λ denotes the wavelength, $t_e xp$ the time since explosion, n_x the number of atoms in the state x and g_x is the statistical weight of the state x. Both homologous expansion and Sobolev approximation have their caveats. In the case of homologous expansion it is thought to be a very good approximations after the first few minutes after the explosion. The main caveat for Sobolev approximation is that a line is not a delta-function, as assumed in the Sobolev approximation. If too strong bound-bound lines are close in frequency space it can lead to the first line shielding the second line. In summary for fast supernova fitting both approximations seem to still allow for a relatively well fitting spectrum.

We have discussed the propagation of the photons in the plasma but have not discussed the state of the plasma yet. The simplest assumption for the state one can make is local thermodynamic equilibrium. In this case the Boltzmann formula describes the level populations in a single ion:

$$\frac{n_j}{n_{\text{ground}}} = \frac{g_j}{g_{\text{ground}}} e^{-(\epsilon_j - \epsilon_{\text{ground}})/kT}$$

Similarly we can calculate the ionization state using the Saha-equation:

$$\frac{N_j}{N_{j+1}} = n_e \frac{U_j(T)}{U_{j+1}(T)} C_I T^{-3/2} e^{\chi/kT},$$

where U_j is the partition function and C_I is a constant. As the ionization likelyhood depends on the internal electronic state of the atom the partition function sums up over the different states:

$$U_j = \sum_i g_{i,j} e^{-\frac{E_{i,j}}{kT}},$$

where i describes the excitation states and j the ionization states. The other symbols have their usual meaning. The sum normally diverges slowly so one in practice just sums up until a highly excited state.

The ML MONTE CARLO uses the so called *nebular approximation* which will calculate the excitation and ionization state of the SNe at nearly LTE cost. In this nebular approximation they introduce a dilution factor W. This is a purely geometrical factor. Treating the photosphere as a point source the factor would result in $W = 1/r^2$ with r being the distance from the center. As the photosphere is expanded the dilution factor has a slightly more complex formula. An important point to note is that purely theoretical at the photosphere the dilution factor is 0.5.

The mean intensity for the supernova at a specific zone is given as:

$$J = WB(T_R),$$

where T_R is the radiative temperature. The radiative temperature is estimated in the ML MONTE CARLO by matching the mean frequency of $B(T_R)$ with the mean frequency of the photon packets in the current zone (Wien approximation). W is chosen so that the frequency-integrated intensity matches the photon distribution.

Using W and J one now can calculate the electronic and ionization states of the plasma:

$$\frac{n_j}{n_{\text{ground}}} = W \left(\frac{n_j}{n_{\text{ground}}} \right)_{T_R}^{\text{LTE}}$$

and

$$\frac{N_j}{N_{j+1} n_e} = W \left(\frac{N_j}{N_{j+1} n_e} \right)_{T_R}^{\text{LTE}}$$

In the simplest case we can treat the ejecta as homogeneous in temperature and abundance. For now we will also assume a pure scattering line interaction. This means that the photon is absorbed at a resonance frequency and then instantaneously reemitted with the same frequency into a random direction. This in in contrast to photon branching which we will discuss later.

We assume a time since explosion t_e, a photospheric velocity v_{ph}, $L_{\text{bol}}$and an abundance distribution for the chosen elements. W7 (Nomoto et al., 1984) is used in the ML MONTE CARLO as a density structure.

The one dimensional model is divided into multiple zones that each have the same abundance but a different density. Using an initial guess of T_{eff} for the photosphere, one can calculate the plasma condition in each shell.

The Monte-Carlo simulation begins. A photon packet is emitted with a random frequency and a random angle drawn from a Blackbody distribution $B(T)$. Each photon packet contains the same energy (more photons per packet in the red than in the blue). An event optical depth is calculated from a uniform random distribution so that $\tau_{\text{event}} = -ln(z); z \in (0, 1]$. In the next step there are three possible outcomes. We calculate the length

of the path (s_e) that the packet can travel freely before τ_{event} is equal to the Thompson scattering opacity $\tau_{\text{event}} = \sigma_T n_e s_e$. Next we calculate the same path length for the lines s_l using as a target opacity $\tau_e + \tau_{\text{line}}$. If both paths are longer than the path to exit the current zone, then the photon exits the current zone and a new Monte Carlo process begins. If however s_e is the shortest then Thompson scattering occurs and the photon is assigned a new direction and a new τ_{event} is drawn and the process begins anew.

In the case of line scattering the excited atom can de-excite through many lines. ML MONTE CARLOrandomly chooses a downward transition for the whole packet (taking the appropriate weights into account). The number of photons in the packet is adjusted to ensure that the energy is conserved in the comoving frame.

There are two possibilities for the final fate of the photon. Either it is reabsorbed into the photosphere or is emitted from the supernova.

When initializing the state of the plasma one assumes an initial guess for the photon temperature. The Monte-Carlo simulation is run and records each packet status at the mid-point of each shell. This information is used to calculate a new photospheric temperature and an updated plasma condition (level population and ionization). This procedure is repeated until the photospheric temperature converges. Once convergence is reached the actual Monte-Carlo simulation begins.

The final spectrum is not calculated using the escaping packets. Instead we calculate the optical depths using and then calculate the emerging spectrum using the formal integral. This has the advantage of reducing noise in the spectrum due to Monte-Carlo noise and gives very good results.

A more detailed description of the code can be found in Mazzali & Lucy (1993); Mazzali (2000).

1.2. Manually fitting a Type Ia supernova

When fitting manually there are several features that help guide the direction of the fit. We will attempt to explain by using a spectrum of SN2002bo (cite?????) 10 days before maximum. In this section we will only talk about fits with no abundance stratification. Stephan Hachinger has kindly provided his manually obtained best fitting parameters (for the supernova at this stage (see Figure 1.1). Directly measurable are the redshift of the supernova (and implied distance) and the time of the spectrum relative to maximum. We assume calculate the time since explosion assuming a rise time of 19.5 days. The other parameters are initialized using empirical data.

The chosen fundamental parameters are $\log L/L_\odot = xx$, $v_{\text{ph}} = xxx$. We have listed the non-zero abundances in Table **??**.

The P-Cygni profiles of many features are easily visible. The Calcium line in the blue can be seen to be to blueshifted in relation to the model. This property is not unusual and is thought to come from high velocity component at the outer edge of the ejecta. The next major known discrepancy that can be seen is the excess of flux redwards of ≈ 6200Å. This is a region that usually does not fit well as the underlying black body spectrum overestimates the flux in this region. When fitting manually often one tries to fit the depth the lines instead of the continuum.

There are three main parameters that have the most influence on the overall fit: Luminosity, photospheric velocity and abundance in iron group elements.

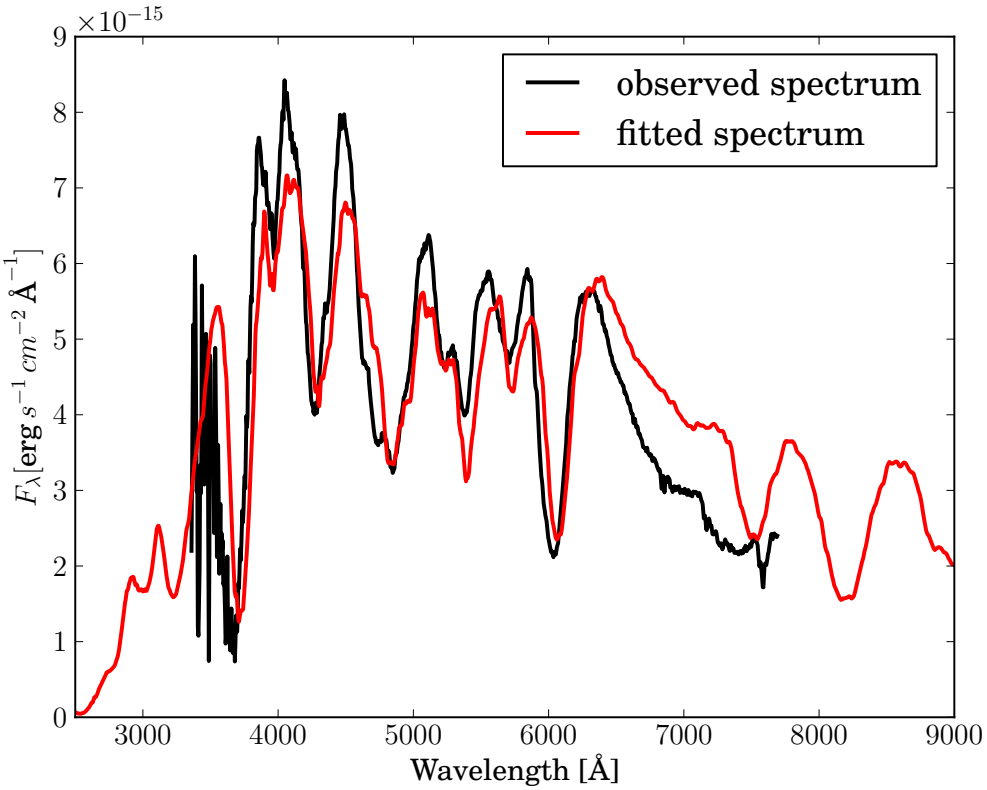

Figure 1.1 example caption

A large offset in L to the best fit paramater is easily visible as a large offset of the continuum (see Figure **??**. Thus it is easy to constrain the parameter space in L initially. L also has influence on the temperature of the model through:

$$L_{\text{bol}} = 4\pi\sigma\, R^2\, T^4 = 4\pi\sigma\, v_{\text{ph}}\, t_{\text{exp}}.$$

Velocity in astronomy is often measured using the doppler shift. In this case however it is hard to measure the photospheric velocity as lines are created at different depths and thus at different velocities. This smears out the line profiles which makes fitting velocities nearly impossible using this technique. The main impact of photospheric velocity is establishing the temperature given a luminosity. A model with a too high photospheric velocity will have expanded more than the real spectrum at that time and will be cooler. This results in a spectrum that is too luminous in the red and not luminous enough in the blue (see Figure 1.2). A secondary effect is that the ion population will be different to the actual supernova and the line spectrum differs. The initial abundances for each fit

Figure 1.2 example caption

are determined by integrating the abundances above the photosphere in the W7 model (Nomoto et al., 1984). The iron group element have a similar influence on the overall flux distribution as the photospheric velocity. As we assume no stable Cobalt and the input parameters for Nickel, Cobalt and Iron are $^{56}\text{Ni}_0$ and $^{56}\text{Fe}_0$ and calculate the abundances

using radioactive decay. Ti and Cr have no easily identifiable single lines in the observed spectra, but provide line blanketing in the blue. We often lock their ratios and only use one abundance as an input parameter.

These elements cause photons to be absorbed in the UV and be reemited in the red. A too high abundance will suppress the flux in the blue too much and will cause the spectrum to be over-luminous in the red (see Figure **??**). Although physically different from the photospheric velocity, phenomenologically these are similar. The degeneracy is broken by identifiable Fe-Lines in the red part of the spectrum as well as the ionization balance determined by the temperature (influenced by photospheric velocity). This near degeneracy causes a very complex search space.

There are six other abundances that are taken into account when fitting: Carbon, Oxygen, Magnesium, Silicon, Sulfur and Calcium. Among these Oxygen plays a special role. It does not have lines except the Oxygen triplet at 7778 Å. In our fitting routine it acts as a buffer element and is assigned the remaining fraction that is left after all elements have been given abundances.

The first element that is usually adjusted from its initial value is Calcium. The Calcium line is relatively easy to identify. Calcium also does not depend an awful lot on the right choice of temperature. One caveat however is that Calcium saturates at a certain point so if the observed line is close to that point one can only extract a lower limit for Calcium.

The next to be adjusted are Silicon and then Sulfur. Both of these elements are linked through nuclear synthesis and we don't expect there to be more Sulfur than Silicon. We also expect no less Sulfur than a third of Silicon. Silicon also provides an important measure for temperature through the ionization balance of doubly ionized Silicon to triple ionized Silicon (see Figure **??**).

Magnesium has one strong feature near 4300 Å, which constrains the Magnesium abundance.

Doubly ionized Carbon has a line at $\approx$ 6300 Å which is often not present or very weak.

Fitting the spectra involves first adjusting luminosity and then IGEs as well as photospheric velocity. This is followed by adjusting the other elemental abundances from the initial W7 values (Nomoto et al., 1984). After the elemental abundances are adjusted we readjust the luminosity, photospheric velocity and IGEagain. This loop is continued until convergence is reached.

One important factor to check is the value of the dilution factor w at the photosphere. Theoretically we would expect this to be close to 0.5 if the fit is sensible.

After having fit one spectrum we can continue to fit the other photpsheric spectra of the same supernova. We expect most parameters to increase or decrease monotonically (luminosity being the obvious exception).

In summary, the fitting of a supernova is a complex procedure and requires a lot of practice. Our initial tests were using very simple methods like Newton-Raphson and other gradient methods. We quickly discovered the search space is too complex and evaluation time takes too long (each spectrum roughly one minute on a modern computer) to use these simple methods.

1.3. Genetic Algorithms

Finding optimal solutions in complex search spaces is one of the main fields in numerical mathematics. They have wide ranging applications in engineering, bio sciences and physical sciences. There have been several advances in the last decads to optimization. Among them is the remarkable feat of "solving" the long-standing problem of the traveling salesman with simulated annealing (Kirkpatrick et al., 1983). Another major accomplishment was the development of evolutionary algorithms and subsequently genetic algorithms. The idea of an algorithm which imitates the principal of natural evolution was first introduced by Ingo Rechenberg (Rechenberg, 1973). These evolutionary algorithms have since become a sizeable subfield of numerical optimization. Genetic algorithms a subclass of evolutionary algorithms were then further developed by John Holland and David Goldberg (Goldberg, 1989).

When conquering the search space optimization algorithms have two (sometimes conflicting) goals: exploiting good leads while still exploring the search space sufficiently. Simple algorithms like Hillclimbing (randomly selecting a point in the neighbourhood and switch if it is "better" than the current one) will exploit good leads but will neglect to explore the search space. This often leads to be stuck at extrema. Whereas random searches are excellent at exploring the search space but will fail to quickly converge on an optimal solution. Genetic algorithms do strike the balance between exploration and exploiting the current best solution.

Genetic algorithms are a heuristic search technique to find optimal solutions in n-dimensional search spaces. We will consider a function $f(\vec{x})$ with the multi-dimensional solution $\vec{y}$. In terms of genetic algorithms we call $\vec{x}$ genotypes and the solution vector $\vec{y}$ phenotype (similar to biology where the $\vec{x}$ describes the DNA sequence but the solution is can not be thought of as a vector). In addition we define a fitness function $g(\vec{y}) = s$ where s is a scalar. It is our goal to choose the input parameters $\vec{x}$ to maximize s. Following the notation of (**?**) we introduce the population $P(t)$ with the individuals $\{p_1^t, \ldots, p_n^t\}$, where t denotes the iteration. Each individual (p_i^t) is a data structure consisting of a vector $\vec{x}_i$ and its corresponding fitness scalar s. When we speak of evaluating p_i^t we mean that we use $g(f(\vec{x}_i)) = s$ to determine the fitness. A new population $P(t+1)$ is formed by choosing, in the *select step*, the more fit individuals. Some or all of the new population undergo transformations (recombine step). There are unary transformations, which create new individuals by small changes in single individuals called mutations. Higher order transformations called crossovers combine the traits of multiple individuals to form a next generation individual. After the new population has been created in the recombine step, it is evaluated and the *select step* begins anew.

This procedure is repeated until the best individual has reached a certain threshold fitness (see Algorithm **??**).

Algorithm 1 Structure of a genetic algorithm

```
t ← 0
initialize P(t)
evaluate P(t)
while (not termination condition) do
  t ← t + 1
  select P(t) from P(t − 1)
  recombine P(t)
  evaluate P(t)
end while
```

1.4. Genetic Algorithms to fit Type Ia Supernovae: The Dalek Code

BIBLIOGRAPHY

Fisher, A. K. 2000, PhD thesis, THE UNIVERSITY OF OKLAHOMA

Goldberg, D. E. 1989, Genetic Algorithms in Search, Optimization, and Machine Learning, 1st edn. (Addison-Wesley Professional) [LINK]

Hauschildt, P. H. & Baron, E. 1999, Journal of Computational and Applied Mathematics, 109, 41

Kasen, D., Thomas, R. C., & Nugent, P. 2006, ApJ, 651, 366

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. 1983, Science, 220, 671 [LINK]

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

Mazzali, P. A. 2000, A&A, 363, 705

Mazzali, P. A. & Lucy, L. B. 1993, A&A, 279, 447

Nomoto, K., Thielemann, F.-K., & Yokoi, K. 1984, ApJ, 286, 644

Rechenberg, I. 1973, Evolutionsstrategie : Optimierung technischer Systeme nach Prinzipien der biologischen Evolution, Problemata No. 15 (Stuttgart-Bad Cannstatt: Frommann-Holzboog)