K-Corrections

R. C. Thomas

Motivation

Radiation emitted in the rest frame of a distant SN redshifts as it propagates to the observer frame and impinges upon some instrument. Thus, the instrumental throughput function samples a bluer wavelength portion of the SN SED than it would if the originating event occurred in the observer frame.

The bulk of our understanding about the spectroscopic and photometric behavior of SNe Ia results from observations of low-redshift events made in the optical regime. Hence, it is necessary to correct, or more accurately to transform, the observer-frame photometry back to the rest-frame B and V band values in order to use all the events the same way regardless of redshift. These transformations are called K-corrections.

The mechanics of computing K-corrections are trivial, and are not included in this Note. We begin with a reminder of some of the general problems associated with K-corrections, describe the database and software being developed to streamline the process, and mention a couple of the foreseen problems and some random thoughts on them.

Instant Soup Level K-corrections

This is the easiest sort of K-correction: One has an event that has been observed both spectroscopically and photometrically, and the temporal coverage is good. In the most ideal case, the spectroscopic wavelength coverage is also good. This is most likely to occur in the case of a nearby event. Here, one simply uses the observed spectra to construct K-corrections. Inevitably, interpolation on the spectra will be required, and the amount of interpolation can range from bad to worse.

Spaghetti Level K-corrections

Suppose one has plenty of photometric data of an event, but has perhaps only one spectrum confirming type. For good measure, suppose also that the wavelength coverage of the confirming spectrum is not that useful for K-corrections. Here is where we mention a small bifurcation in K-correction technique.

In one case, one could search through a library of SN Ia spectra and select those events which have spectra similar to the observed one. We then assume that this is enough to declare one SN to be like another, though there are many caveats (for example, after maximum they tend to all look similar). Using these spectra, we compute K-corrections. Using several template events gives us a sense of what the uncertainties are. Again, interpolation is probably required.

One can also use all the events that apply. That is, a selection of the spectra with the appropriate wavelength coverage can be made without any decision. This might tend to spread out the uncertainties in the final K-corrections, so it might be a good idea to consider doing both.

On the other hand, one can form template spectra representative of a specific subtype of SN Ia (spectroscopically normal, 91bg-like, or 91T-like, for example). Such a set of spectra has come to be called an *überspectrum*. In this case, the choice of which SN to use goes away, one just uses the überspectrum. Of course, the choices have really already been made by whoever generated the überspectrum.

In both cases, since no two events are really perfectly identical, an adjustment to the SED of the template spectrum or überspectrum is needed. Such an SED-adjustment means that target colors are specified (either observed or from a color model) and the template SED is wiggled in a smooth way to become consistent. See the Problems section.

Souffle Level K-Corrections

Suppose one has a type-confirming spectrum, but really only has a lightcurve in a single band. This makes it hard to do SED-adjustment for either the library or überspectrum approaches. This calls for faith-based K-corrections, or K-corrections with large uncertainties, since very little of a spectrum library can be ruled out. An even worse case would be one without spectroscopy, but we won't discuss that.

Requirements – Database

The mundane problems discussed above can be addressed with an organized database of SN lightcurves and spectra, which is easy to expand both in terms of new data and new observables correlations. Toward this goal, I have assembled a database based largely on the spectra available from the OU Spectrum Archive from the past 25 years. Also included are the lightcurve metrics $\Delta m15$ from Lifan's lightcurve fits, helpful for finding alikeness between SNe. The database is a PostgreSQL relational database, and uses relational principles in its design. There are currently 233 Type Ia spectra in the database from SNe 1983G, 1986G, 1989B, 1990G, 1990M, 1990N, 1990O, 1990R, 1991B, 1991F, 1991K, 1991M, 1991S, 1991T, 1991X, 1991ad, 1991bc, 1991bd, 1991bg, 1991bj, 1992G, 1992ac 1992ah 1994D, 1994M, 1994Q, 1994S, 1994U, 1996X, 1997cn, 1998bu, 1999ee and 2000cx.

The database is multipurpose, it will also be used for classifying spectra automatically in the SNfactory followup pipeline. There are some rather disturbing gaps in the database however, that I am working to fill in. Normal events like 1981B, 1990N and 1992A are missing as I have only access to ASCII versions of these SNe and they are of dubious pedigree. However, I don't think the situation is going to improve so I will just add what I have. This is a time-intensive process, and I am working to make it easy to add new data.

The überspectrum is also included in the database, but is kept separate from the real spectra. Also, it is possible to have multiple versions of the same überspectrum in the database.

Requirements – Software Framework

I am developing a software framework written in Perl/PDL (originally written by Karl Glazebrook) for developing tools needed for making K-corrections. These tools will include spectrum/lightcurve browsers for selecting events from the database, synthetic photometry tools, and toys for seeing why a K-correction curve or synthetic color curve does something (it's always because of the spectrum shape). More advanced tools, like all-in-one K-correctors that do spectrum selection and SED-Adjustment in one go, will also be developed... probably.

The framework consists of a set of classes to be distributed as a sort of K-correction package. Dependencies on other Perl packages are easy to resolve, so this should be a trivial issue. Where necessary, I have written certain algorithms using the Inline::PDLPP module which lets one write "threaded" C code for improving performance (such as the Reddening class with its many exponentials, and the filter convolution function). Access to the PostgreSQL relational database is accomplished through the Perl DBI module. GUI's are written easily using the Tk module and its extensive library of widgets. The plotting tool is the PGPLOT library.

At the present I have written some prototype toys and scripts, the most important one being that used for computing the Albinoni K-corrections.

Problems – Dealing with Uncertainty

Setting aside the issue of faith-based K-corrections, how does one propagate uncertainties through K-corrections? Using the library approach, one can use all the spectra applicable to get a sense of what the uncertainty is like. Using the überspectrum approach, Monte Carlo can be used on the input parameters (stretch, color excess, etc.) to derive K-correction uncertainties. Does this sound okay?

Problems – SED-Adjustment

Based on some simple arguments, we have been applying "warpings" to template spectra in order to match colors and then compute K-corrections. But the warpings can induce some weird artifacts, depending on the technique. We want to be as safe as possible. This is not a huge deal as long as one can see what's going on. But the actual question is what the best warping to do is?

Some people use splines, or just redden the spectra somehow. A nice approach would be to just arrive at the correct adjustment immediately. Clearly it is desirable to apply a non-wiggly function to an SED to get the colors, and Lifan has suggested that we apply second-order polynomials. This results in a simple 3x3 matrix to solve, but still one has to use some judgment to throw out ridiculous warpings.

I propose trying to extend this approach and use SVD (with a variety of subsets of the polynomial basis set) to target colors. The nice thing about using SVD would be that one obtains a covariance matrix as a side-effect. This could be used to determine how robust the targeting of colors is. If the targeting is not robust, it just may not be possible to adjust the spectrum to fit the data. Sound crazy?