

SDSS-II: DETERMINATION OF SHAPE AND COLOR PARAMETER COEFFICIENTS FOR SALT-II FIT MODEL

L. Dojcsak,¹ J. Marriner²
 August 14, 2010

ABSTRACT

In this study we look at the SALT-II model of Type IA supernova analysis, which determines the distance moduli based on the known absolute standard candle magnitude of the Type IA supernovae. We take a look at the determination of the shape and color parameter coefficients, α and β respectively, in the SALT-II model with the intrinsic error that is determined from the data. Using the SNANA software package provided for the analysis of Type IA supernovae, we use a standard Monte Carlo simulation to generate data with known parameters to use as a tool for analyzing the trends in the model based on certain assumptions about the intrinsic error. In order to find the best standard candle model, we try to minimize the residuals on the Hubble diagram by calculating the correct shape and color parameter coefficients. We can estimate the magnitude of the intrinsic errors required to obtain results with $\chi^2/\text{degree of freedom} = 1$. We can use the simulation to estimate the amount of color smearing as indicated by the data for our model. We find that the color smearing model works as a general estimate of the color smearing, and that we are able to use the RMS distribution in the variables as one method of estimating the correct intrinsic errors needed by the data to obtain the correct results for α and β . We then apply the resultant intrinsic error matrix to the real data and show our results.

INTRODUCTION

In an effort to expand upon the already vast Sloan Digital Sky Survey (SDSS)—which imaged more than 8000 square degrees of the sky from 2000 to 2005 using the 2.5 meter telescope (Gunn et al. 2006) at the Apache Point Observatory (APO) in New Mexico—the SDSS-II Supernova Survey (Frieman et al. 2008) was undertaken to detect and measure the light curves for Type IA supernovae (SNe) to study supernova (SN) properties and to use them to measure the expansion of the universe. The CCD camera (Gunn et al. 1998) of the SDSS telescope obtains images in five optical passbands of *ugriz* (Fukugita et al. 1996), which are then processed by the PHOTO photometric reduction pipeline (Lupton et al. 2001; Ivezić et al. 2004). Previously taken co-added images (Sako et al. 2008) are subtracted from the new images in order to correctly identify Type IA supernovae.

Type IA supernovae, colossal explosions of white dwarf stars, are just one subcategory of SNe. These white dwarves are believed to explode by undergoing a thermonuclear reaction caused by accumulating mass from a binary companion star and exceeding the Chandrasekhar limit, causing an explosion that is more luminous than a galaxy. These supernovae are studied because they have a narrow range of absolute magnitudes that are well known. In addition, they have distinctive spectra due to the conditions under which the dwarf star becomes a supernova without the presence of Hydrogen, producing heavier elements such as Sulfur and Silicon. Since the absolute magnitude of an SN IA is known, the apparent magnitudes in this and other surveys are used to determine the distances to the SNe. The supernovae received rapid spectroscopic follow-up by supplementary telescopes in order to determine redshift and to make sure data was restricted to high-quality SN

candidates for analyses. An additional benefit of SDSS-II was to complement other surveys of this type by providing data between the low and high redshift ranges, where Type IA SNe had not been observed before.

Over the course of three fall seasons (September 1 – November 30) in 2005, 2006, and 2007, the SDSS-II SN Survey (Frieman et al. 2008) did repeat imaging of a 300 square degree equatorial stripe, 2.5 degrees wide, in the Southern Galactic hemisphere, designated as SDSS stripe 82 (Kessler et al. 2009a). The survey discovered and obtained data on approximately 500 Type IA SNe in the 0.05-0.40 redshift range for the use of studying the accelerating cosmic expansion as well as to improve the statistical scatter about the Hubble fit diagram, which plots the distance of cosmological objects as a function of redshift.

In order to measure the light curve of each SN from the survey, we chose to apply the Spectral Adaptive Light Curve Template Fitting (SALT-II) (Guy et al. 2007) model in this research as presented in the Supernova Analysis (SNANA) (Kessler et al. 2009b) software package for the SDSS. This fitting method uses a two-parameter model of light curve shape (x_1) and excess color (c) in order to compare it to a well measured SN with known parameters. The SALT-II fitter uses the following base equation for the absolute magnitude (M_b) of the SNe:

$$M_b = m_b - \mu(z) + \alpha x_1 + \beta c$$

The equation shows the correlation between the absolute standard candle magnitude (M_b) to be proportional to the apparent magnitude that we measure in the B band m_b , the distance modulus $\mu(z)$, the shape of the light curve (x_1), and the excess color offset in the light curve (c). The parameters x_1 and

¹ Lawrence Technological University, 21000 West Ten Mile Road, Southfield, MI 48075-1058

² Center for Particle Astrophysics, Fermi National Accelerator Laboratory, P.O. Box 500, Batavia, IL 60510

c do not have any particular physical significance in the model; they account for anything that may change the shape or color of the SN light curves. The model for the absolute standard candle magnitude depends linearly on these shape (x_1) and color (c) parameters. The coefficients are α for the shape parameter (x_1) and β for the color parameter (c). Our purpose in this research is to observe trends in the results from altering the intrinsic error model and to try and calculate the correct α and β coefficients for use in determining the absolute magnitude (M_b) and the distance moduli ($\mu(z)$).

Before applying the model to the real data, we test it with a Monte Carlo simulation of SN light curves, provided in the SNANA package. The program SALT2mu is what we use to calculate the α and β parameter coefficients. Since we generate the Monte Carlo simulation with known values of α and β , we are able to test the program on how well it is able to calculate the shape (α) and excess color (β) coefficients. Thus, we can say with certainty that the correct values in the simulation are distributed about $\alpha = 0.11$ and $\beta = 2.60$. We use this method to determine trends in altering specific parameters in the program SALT2mu as well as to improve the calculations on the α and β parameter for this type of analysis. We do this in order to find the best standard candle model by minimizing the residuals in the Hubble diagram.

SALT2mu ANALYSIS

We use the program SALT2mu, provided with the SNANA software package, in order to calculate the α and β coefficient values in both simulation and SDSS data. We adjust the intrinsic error to obtain χ^2 per degree of freedom equal to 1 ($\chi^2/\text{DOF} = 1$) for good statistical results. χ^2 is determined by the following equation in the SALT2mu program:

$$\chi^2 = \sum_{i=1}^N \left[\frac{m_{Bi} - m_0(z_i) - \mu(z_i) + \alpha x_{1i} - \beta c_i}{\sigma_i} \right]^2$$

The value of σ_i is equal to the measurement error in the numerator for the i^{th} supernova for all N supernovae, plus an intrinsic error that we do not fully understand. The intrinsic error is the additional error that we find in the real data, but not in the simulation, so we must add it using a covariance error matrix for the total error (σ_i^2) in m_b , x_1 , and c:

$$\sigma_i^2 = C_{11} + \alpha^2 C_{22} + \beta^2 C_{33} + 2\alpha C_{12} - 2\beta C_{13} - 2\alpha\beta C_{23}$$

This total error is divided into parameters where $C_{ab} = C_{ab}^{\text{meas.}} + C_{ab}^{\text{int.}}$ and the program makes use of altering:

$$\Sigma_1 = \sqrt{C_{11}^{\text{int.}}} \equiv \text{Intrinsic error for } m_b$$

$$\Sigma_2 = \sqrt{C_{22}^{\text{int.}}} \equiv \text{Intrinsic error for } x_1$$

$$\Sigma_3 = \sqrt{C_{33}^{\text{int.}}} \equiv \text{Intrinsic error for } c$$

Since we do not understand the source(s) of the intrinsic error, we want to know what happens to resultant values of α and β if

we place the error completely in one intrinsic error. This paper examines the dependence on the assumed cosmological parameters and how the intrinsic error matrix has an effect on the outcome, including a possible solution using the color smearing model in SNANA.

Before we can attempt to test varying the intrinsic error matrix, we must restrict the equation so that the results do not vary based on assumptions about the cosmology. The SALT2mu program is written so that it can calculate the distance moduli for Type IA SNe by separating the determination of the shape and color parameter coefficients, α and β , from the determination of the cosmological parameters. We do not want the cosmology to have an effect on the determination of the α and β coefficients since we want to observe effects only due to the error. In order to do this, we split the data into multiple equally ranged redshift bins for $z = 0.02-0.42$. By splitting the data into multiple bins, we create new M_b variables in the equation for each bin, thus minimizing the effect of the distance modulus ($\mu(z)$), which is a function of the cosmological parameters.

In order to test the number of redshift bins that we would need, we use a standard SNANA Monte Carlo simulation with the SALT2mu program and use three different assumptions for the cosmological parameters. The parameters are:

$\Omega_\lambda \equiv$ Percent of Universe Made Up of Dark Energy

$\Omega_K \equiv$ Curvature of the Universe

$w \equiv$ Pressure/Density Parameter at $z=0$

First we assume the standard observed cosmological parameters where $\Omega_\lambda = 0.7$, $\Omega_K = 0.0$, and $w = -1.0$. These are the parameters we use in all other runs of the program with the standard simulation. The other two assumed cosmologies are $\Omega_\lambda = 0.0$, $\Omega_K = 0.7$, $w = -1.0$ and $\Omega_\lambda = 0.7$, $\Omega_K = 0.0$, $w = 0.0$. By increasing the number of bins from 1 to 19, we observe in Figures 1 and 2 that the values of α and β begin to converge after approximately 4 bins.

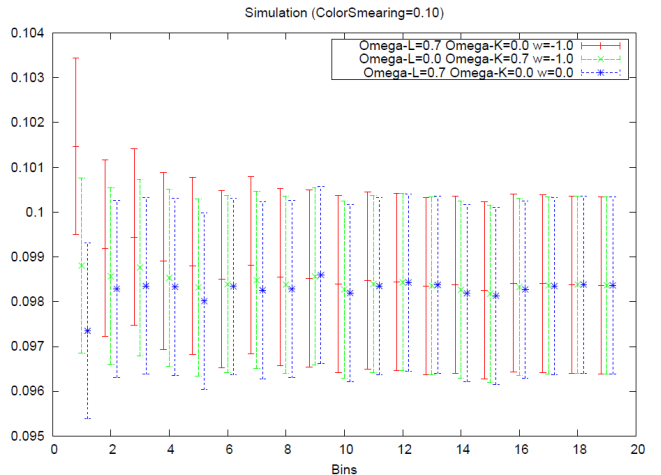


Figure 1 – Plot of α vs. number of redshift bins with error bars for three different assumed cosmologies.

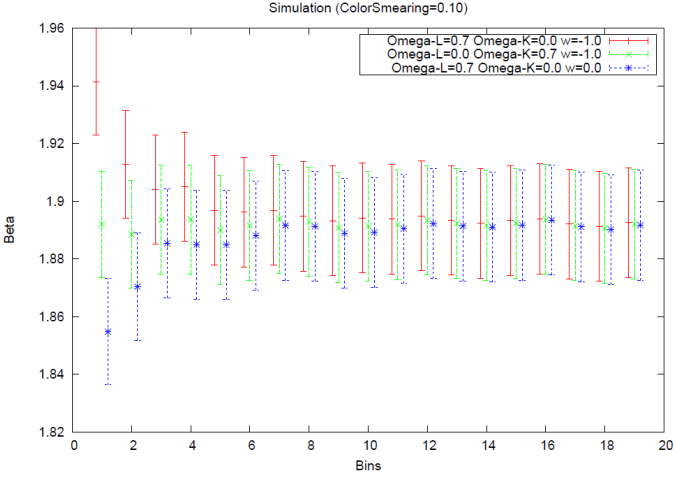


Figure 2 – Plot of β vs. number of redshift bins with error bars for three different assumed cosmologies.

The plots demonstrate that both α and β begin to converge after a low number of bins even for a simulation of 5000 SNe and that the results for the two parameters can be found independently of the cosmology parameters entered into the program. In theory, this could be performed for any size data with increasing bins, so long as it stays within the limits of the program of at least 5 entries per bin. However, this would become impractical as it would greatly increase computation time without producing significantly better results. The plots show that 4 bins are enough for further tests.

INTRINSIC ERROR MATRIX

Since we have determined that the α and β parameters can be found independent of the cosmology parameters with the use of multiple bins, we want to examine the trends caused by the intrinsic error matrix in the SALT2mu program. Assuming, in turn, that the intrinsic error for m_b , x_1 , or c is increased from 0 in order to see how the results of the α and β coefficients are affected, we see that when Σ_1 is increased, α and β both decrease. When the intrinsic error in Σ_2 is increased, α increases and β decreases, and when the intrinsic error in Σ_3 is increased, α decreases and β increases. Increasing any individual intrinsic error also decreases the χ^2/DOF . However, the various assumptions for the intrinsic error are not compatible, as the results depend on the intrinsic error matrix.

Next, we want to find the intrinsic errors that would be needed if the errors were completely in m_b , x_1 , or c . The method we used to come up with these results was to adjust the values of Σ_1 , Σ_2 and Σ_3 independently in the program until we obtained a $\chi^2/\text{DOF} = 1$ as shown in Tables 1, 2 and 3. The exception is the first row of each table which lists the results of setting all three intrinsic errors to 0. Table 1 lists the results for the α and β value calculations for the standard SNANA Monte Carlo simulation. It can be inferred from these results, that some sort of intrinsic error is needed, since setting the error matrix to 0 gives us a very high χ^2 value and therefore a χ^2/DOF that is much higher than 1.

TABLE 1 Standard SNANA Simulation

Intrinsic Error	χ^2	α	$\Delta\alpha$	β	$\Delta\beta$
$\Sigma_1=\Sigma_2=\Sigma_3=0$	17468	0.1510	0.00126	2.016	0.0103
$\Sigma_1=0.1275$	4319	0.1008	0.00186	1.910	0.0177
$\Sigma_2=0.8792$	4318	0.2069	0.00377	1.812	0.0257
$\Sigma_3=0.05927$	4318	0.0950	0.00209	2.424	0.0223

Table 1 – (4324 entries of 5000) The above table lists the values of α and β for the $\chi^2/\text{DOF} = 1$ output with errors, with the exception of the first row, where all three intrinsic errors are set to 0.

We were not necessarily expecting to obtain the correct results for α and β with these assumptions for the intrinsic errors. What we might hope to see here is a distribution of results around the correct values of $\alpha = 0.11$ and $\beta = 2.60$. However, we never quite get a value above 2.60 for β in the simulation. Although it would seem intuitive to simply increase Σ_3 in order to check that we are able to get a value of 2.60 for β due to the general trends, this does not attain the result that we seek since the values of α and β both converge towards an asymptote when the values of Σ are increased infinitely.

Since this result relies on using the intrinsic errors in the diagonal of the error matrix only, another test of the program is to see if the correct values for α and β can be recovered by setting the correlation coefficients for the off-diagonal values between m_b and x_1 , m_b and c , and x_1 and c to be some arbitrary value (since we do not know the correlation). Then we can alter the non-diagonal intrinsic errors in the error matrix to see if we can find a value which can recover 2.60 for β for any values, and 0.11 for α and 2.60 for β at the same time. The results show that it is possible. One example that we find is when $\Sigma_1 = 0.1325$, $\Sigma_2 = 0.0$, $\Sigma_3 = 0.07$ and with the m_b and c correlation coefficient equal to 0.9, the results we obtain are $\alpha = 0.109$ and $\beta = 2.60$. However, this does not produce a $\chi^2/\text{DOF} = 1$. Instead the value is closer to 2.

Another way to verify that the correct values of α and β can be recovered from the data can be done by using an additional simulation in SALT2mu. The program is written with an internal simulation that can generate new values for the parameters m_b , x_1 , and c from the Monte Carlo simulation that it then uses in the calculations. The results of the internal simulation are listed in Table 2 for the same SNANA Monte Carlo simulated data as before. These results verify that we can recover the correct α and β values with statistical accuracy.

TABLE 2 SALT2mu Internal Simulation with Stand. Simulation

Intrinsic Error	χ^2	α	$\Delta\alpha$	β	$\Delta\beta$
$\Sigma_1=\Sigma_2=\Sigma_3=0$	4210	0.1090	0.00096	2.620	0.0107
$\Sigma_1=0.1275$	4293	0.1084	0.00187	2.593	0.0228
$\Sigma_2=0.8792$	4268	0.1091	0.00157	2.630	0.0192
$\Sigma_3=0.05927$	4152	0.1119	0.00216	2.626	0.0270

Table 2 – (4325 entries of 5000) This table lists the values of α and β for the $\chi^2/\text{DOF} = 1$ output with new generated values for m_b , x_1 , and c by the program’s internal simulation.

Finally we run the program for the real supernova data from SDSS-II without the internal simulation. The results are listed in Table 3. The results we obtain for α and β here are actually much higher in general than the ones we get from the Monte Carlo simulation. Thus, we can conclude that the values we are entering for the simulation and in the calculation are too low.

TABLE 3 SDSS-II 2005, 2006, 2007 Data

Intrinsic Error	χ^2	α	$\Delta\alpha$	β	$\Delta\beta$
$\Sigma_1=\Sigma_2=\Sigma_3=0$	1316	0.1857	0.00619	2.826	0.0477
$\Sigma_1=0.146$	344	0.1400	0.00970	2.655	0.0779
$\Sigma_2=0.72$	345	0.2922	0.01975	2.362	0.1157
$\Sigma_3=0.05$	344	0.1223	0.01100	3.227	0.1020

Table 3 – (350 SNe) This tables shows the results for α and β by running the program on the real SDSS-II SN data.

ESTIMATING INTRINSIC ERRORS

Although we observe a scatter in the Hubble diagram, we are not sure what causes the scatter, or what the form of the intrinsic error matrix should be. However, we are able to calculate the intrinsic errors in the context of the SNANA color smearing (CS) model. The scatter in the distance modulus ($\mu(z)$) is defined (Kessler et al. 2009c) as the RMS of the difference between the fitted and the generated distance modulus. In the same way, we can use this definition to estimate possible intrinsic error to each shape, color, and light magnitude by taking the difference in each simulated and generated variable for the Monte Carlo simulation to find the total error. One step we take in order to determine the correct values for the intrinsic errors is by applying the correct “color smearing” that we observe in the data to the simulations and estimate the correct intrinsic error values from the RMS of the data. The “color smearing” introduces an independent magnitude fluctuation in each passband (*ugriz*) with the fluctuation being the same for all epochs. (Kessler et al. 2009c) Since we are using a 0.10 color in the simulations as default, we use a range of values around the default to test it. The results are shown in Figure 3, A-F.

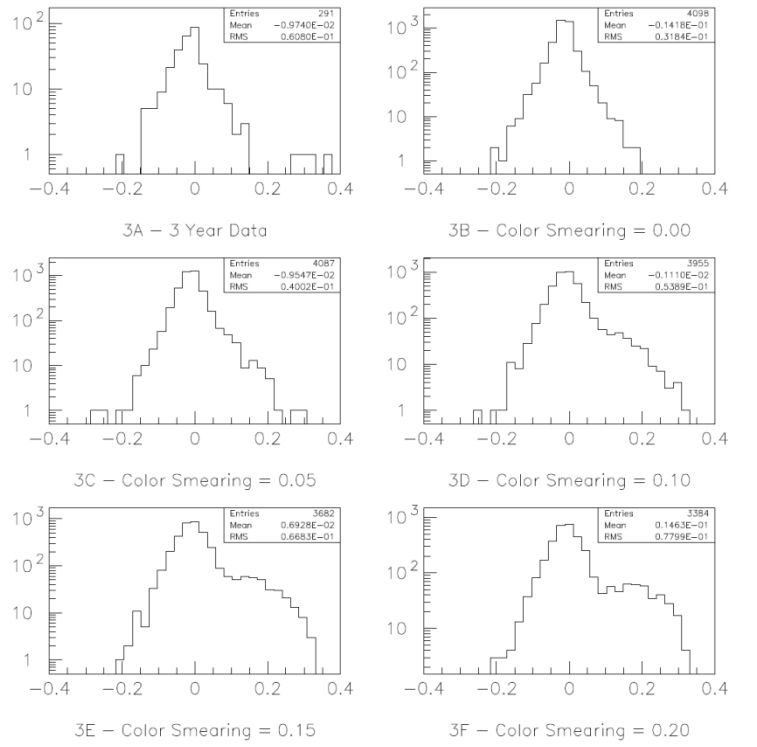
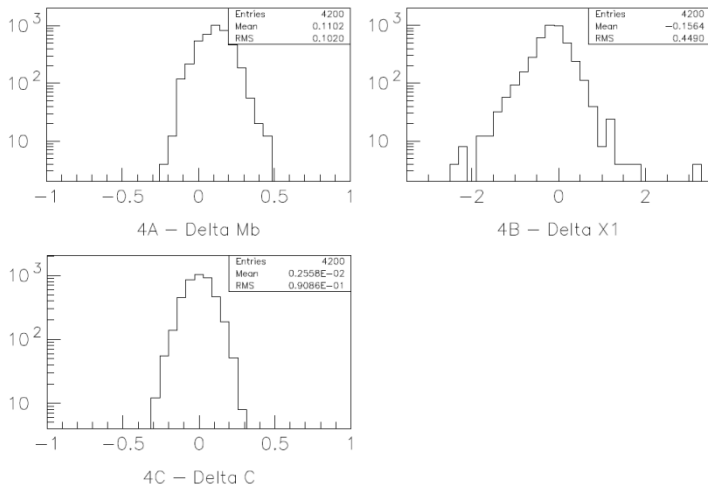


Figure 3A-3F – The above plots all show Δz for different data, where $\Delta z = \text{photometrically fit redshift} - \text{spectroscopically fit redshift}$ ($\text{photo}z - \text{spec}z$). Figure 3A shows the plot for Δz for the real data, and Figures 3B-3F show the plots for Δz for Monte Carlo simulations with color smearing 0.00, 0.05, 0.10, 0.15, and 0.20.

These plots show the RMS values for $\Delta z = \text{photometrically fit redshift} - \text{spectroscopically fit redshift}$, which we assume to be very close to or the true value of the redshift (z). The RMS values in the distribution and the plots demonstrate that color smearing = 0.10 is the closest to the real data. Although the distribution in the simulation is somewhat different from the real data, especially seen in the tail to the right of the zero, since the color smearing in the simulation is an ad hoc model designed to match the real data, we do not expect them to match exactly.

Therefore, we look at the results for the simulation with color smearing = 0.10 output of the Δ values for light magnitude (m_b), color (c), and shape (x_1) where Δ signifies that the distribution is the difference between the reconstructed value (from the light curve fit) and the true (simulated) value. These results are shown in Figure 4, A-C.

These plots indicate that according to the photometrically fit data from the SNANA software package, the RMS values show that the total error for m_b (Σ_1) should be 0.1020, error for c (Σ_2) should be 0.4490, and error for x_1 (Σ_3) should be 0.09086.



Figures 4A-4C – The above plots are all for the Monte Carlo simulated supernovae with color smearing = 0.10. Figure 4A shows the plot for Δm_b with RMS = 0.1020. Figure 4B shows the plot for Δx_1 with RMS = 0.4490. Figure 4C shows the plot for Δc with RMS = 0.09086.

The above plots demonstrate that this method is useful in estimating the total errors based on this ad hoc model. This, however, introduces a dependence on a model whose validity is uncertain. However, it allows us to use the plots in Figures 4A-4C to determine the intrinsic error matrix by finding the values for the intrinsic error that must be added to the measured error, as can be determined from the SALT2mu program, to the total error determined from the simulations. Table 4 lists the values for the error matrices for the total error, measurement error, and our final result for the intrinsic error that must be added using the equation for σ_i^2 . Similar calculations can be used to estimate the off-diagonal part of the error matrix. Based on these calculations we use the following correlations:

- m_b and $x_1 = -0.3$
- m_b and $c = 0.65$
- x_1 and $c = -0.1$

TABLE 4 Errors for Type IA Supernovae

Error for Parameter:	m_b	x_1	c
Measured Error	0.0506	0.4890	0.0384
Total Error (CS Model)	0.1020	0.4490	0.0909
Total Error (SALT2mu)	0.1022	0.4821	0.0878
Intrinsic Error Needed	0.07	0.10	0.08

Table 4 – List of the values for measurement, total, and intrinsic errors for the parameters of apparent light intensity (m_b), shape (x_1) and color (c). Total error is listed as measured by the plots from the color smearing model, as well as the values used directly by the SALT2mu program.

Table 5 lists the final results of running the SALT2mu program with the newly determined intrinsic errors and the measurement errors in SALT2mu for both the standard Monte

Carlo simulation as well as the real data. In both instances, the χ^2/DOF is close to but lower than 1. The fact that the χ^2/DOF is close to 1 is a significant success for the color smearing model since the amount of color smearing was determined independently of the Hubble diagram (using the photo-z fits). We were able to determine a way to estimate the intrinsic errors (listed in Table 4) in order to acquire fairly close results for α and β in the simulation, where we know the correct result. Therefore, we can see from these values that the use of this model works as one method for estimating the intrinsic error for the shape and color parameter coefficients, α and β , for use in determining the distance moduli ($\mu(z)$).

TABLE 5 Final Results for α and β

	χ^2/DOF	α	$\Delta\alpha$	β	$\Delta\beta$
SNANA Sim.	3716/4318	0.0919	0.00236	2.562	0.0274
SDSS-II Data	219/344	0.1140	0.01377	3.394	0.1385

Table 5 – This table lists the final results of α and β with errors for the SNANA simulation as well as the real SDSS-II SN Survey data taken in fall seasons of 2005, 2006, and 2007.

CONCLUSIONS

The results of this project show that dividing the data into multiple redshift bins of at least 4, makes it possible to separate the determination of the α and β parameters from the determination of the cosmology. Using an extreme amount of bins, however, is unnecessary since it ultimately increases computation time without much statistical benefit. Using this method, we find that an intrinsic error is needed in order to determine the correct shape and color parameter coefficients, α and β respectively. When no intrinsic error is present, the χ^2/DOF estimation is much too high and does not yield statistically accurate results. Therefore, both α and β depend on the intrinsic errors in the SALT-II model, and cannot be determined from measurement error alone.

Since the 3x3 error matrix that we use in the program is symmetric, we were working with 6 unknown variables: the three intrinsic errors (Σ) and the correlation coefficients between m_b , x_1 , and c with values listed previously in the paper. By applying the color smearing model to the Monte Carlo simulation with the photoz reconstruction, we determine that the simulation needs a color smearing = 0.10 to match the data. We can use the non-photoz simulation to then determine the distribution of the Δ values for each variable m_b , x_1 , and c and use their RMS values as the total error. We then can apply this total error in the program SALT2mu to add in the required intrinsic error to the measurement error to match our calculated total error. The results of this method yield values for the intrinsic error matrix with $\Sigma_1 = 0.07$ $\Sigma_2 = 0.10$ $\Sigma_3 = 0.08$ with a final $\chi^2/\text{DOF} = 3716/4318$ for the values of α and β .

With the known correlations between m_b and x_1 , m_b and c , and x_1 and c , the results can be applied to the 3 year SDSS-II SN Survey data. Our final results, as listed in Table 5, yield values of $\alpha = 0.0919$ and $\beta = 2.562$ for the Monte Carlo simulation with the correct values being $\alpha = 0.11$ and $\beta = 2.60$.

For the SDSS-II data, our results are $\alpha = 0.1140$ and $\beta = 3.394$. As the tables indicate, these results show that our simulation is roughly consistent with data, and this serves as one general method for determining the shape and color parameter coefficients. However, looking at the final values by running the program with our estimated intrinsic error values implies that we are entering a value that is too low for the color parameter coefficient β .

ACKNOWLEDGEMENTS

I would also like to acknowledge and thank the Wayne State University Research Experience for Undergraduates program for this research opportunity and Scott Dodelson for his advice and discussion on the topic of maximum likelihood approximation with fitting data to lines with errors in x and y (Appendix).

REFERENCES

- Gunn, J. E. et al. 1998 AJ, 116, 3040
–. 2006, AJ, 131, 2332
Guy, J. et al. 2007, A&A, 466, 11
Frieman, J. A. et al. 2008, AJ, 135, 338
Fukugita, M. et al. 1996, AJ, 111, 1748
Hayden, B. et al. 2010, ApJS, published (arXiv:astro-ph/1001.3428)
Ivezić, Ž. et al. 2004, Astronomische Nachrichten, 325, 583
Kessler, R. et al. 2009a, ApJS, accepted for publication (arXiv:astro-ph/0908.4274)
Kessler, R. et al. 2009b, PASP, accepted for publication (arXiv:astrop-ph/0908.4280)
Kessler, R. et al. 2009c, ApJ, accepted for publication (arXiv:astro-ph/1001.0738v2)
Lupton, R. et al. 2001, in ASP Conf. Ser. 238: Astronomical Data Analysis Software and Systems X, ed. F. R. Harnden, Jr., F. A. Primini, & H. E. Payne, 269-+
Press, W. H. et al. 1988, Numerical Recipes in C, (Cambridge University Press, New York, NY)
Sako, M. et al. 2008, AJ, 135, 348

TEST 1 – Elementary χ^2 Minimization (Ignoring Errors in X)
vs. Fitting with Errors in X and Y

The most elementary method used for estimating the slope of a set of correlated data points, and the method used in our data analysis assumes error only in one variable. Our method (chisq1) minimizes a χ^2 in order to find the correct slope of the data using the method given in Press, et. al. (1988). We take a look at methods of better estimating the slope (s) through a simulation of data points distributed about a straight line $y=sx$, and by minimizing the chisq1 value between the data points and the estimated straight line. We use the pseudo-random number generator embedded in the C compiler to generate 10,000 data points with true slope value $s = 2.5$ using a Gaussian distribution for the errors in x and y as shown in Figure 5. Generally, the data we work with is analyzed by the equation:

$$M_b = m_b - \mu(z) + \alpha x_1 + \beta c$$

However, in order to explore this mathematical model and a different method of maximum likelihood estimation, we use the simplified $y=sx$ for only two variables y and x. To relate it back to the original, it could be looked at as $M_b = \alpha x_1$ where we are looking to find α with parameters M_b and x_1 .

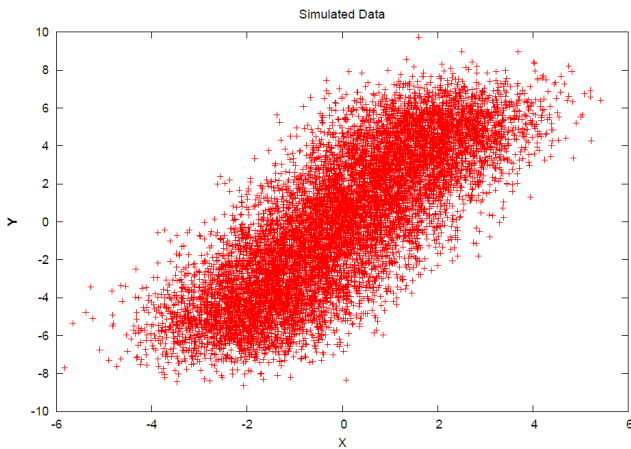


Figure 5 – The simulated data set of 10,000 points used in the exercise to compare the two fitting methods, plotted as x vs. y with a Gaussian distribution in the errors for x and y.

We then calculate the chisq1 for the data set by incrementing the slope at an assigned value of 0.05 from 0.00 to 5.00 and finding the corresponding slope with the smallest $\text{chisq1} = \sum_i \frac{(y_i - sx_i)^2}{\sigma_y^2 + s^2 \sigma_x^2}$ for i being each simulated data point. We can compare the estimated slope value found by minimizing this chisq1 output, to the new chisq2 (χ'^2) method.

The new method for estimating the slope proposed by Scott Dodelson (private communication) was added to the same program under the name chisq2 and applied to the same data set. The expression for chisq2 is obtained by integrating

analytically over all possible x and y, the likelihood $P(x,y)$ of getting each x and y value with error σ_x and σ_y :

$$P(x^{meas}, y^{meas} | s) = \frac{1}{\sigma_x \sigma_y} \int_{x_{min}}^{x_{max}} dx \int dy e^{-\frac{(x-x^{meas})^2}{2\sigma_x^2}} e^{-\frac{(y-y^{meas})^2}{2\sigma_y^2}} \delta(y - sx)$$

So that the solution is:

$$\chi'^2 \equiv \sum_i \frac{(y_i - sx_i)^2}{\sigma_y^2 + s^2 \sigma_x^2} - 2 \ln \left[\theta(x_0 - x_{min}) P\left(\frac{1}{2}, A(x_0 - x_{min})^2\right) + \theta(x_{max} - x_0) P\left(\frac{1}{2}, A(x_0 - x_{max})^2\right) \right]$$

Where:

$P(a,b) \equiv$ Incomplete gamma function

$$A = \frac{\sigma_y^2 + s^2 \sigma_x^2}{2\sigma_x^2 \sigma_y^2}$$

$$x_0 = \frac{\sigma_y^2 x^{meas} + s y^{meas} \sigma_x^2}{\sigma_y^2 + s^2 \sigma_x^2}$$

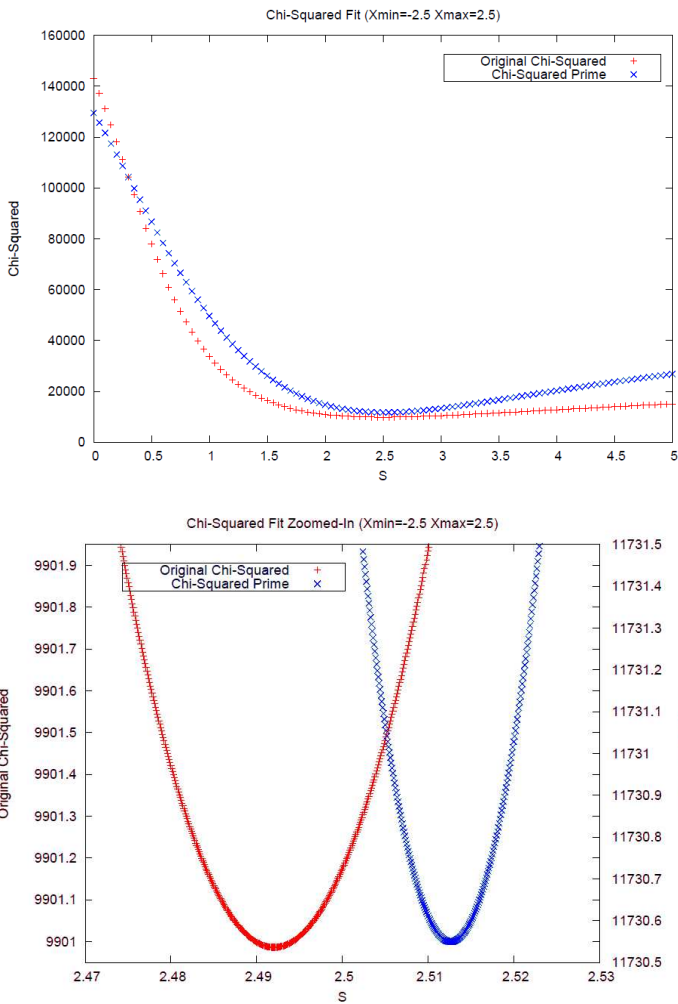
$$\theta = 1 \text{ for } x > 0 \text{ and } -1 \text{ for } x < 0$$

The program uses the simulated data to calculate both chisq1 and chisq2 for increasing slopes from 0.00 to 5.00 as before, but now prints out the standard and new methods side by side. The program can also be used to calculate the chisq1 and chisq2 values by running through possible slope values 's' at much smaller intervals (0.0001), as was done in some of the results below, in order to more precisely estimate the value of 's'. The results from the new chisq2 estimator are found for the correct values of the interval $x_{min}=-2.5$ to $x_{max}=2.5$ as well as the slightly wrong intervals of $x_{min}=-2.0$ to $x_{max}=2.0$ and $x_{min}=-3.0$ to $x_{max}=3.0$. Since the original estimator does not use these values, the outcome of the standard method is the same in each following situation but is used as a comparison.

RESULTS

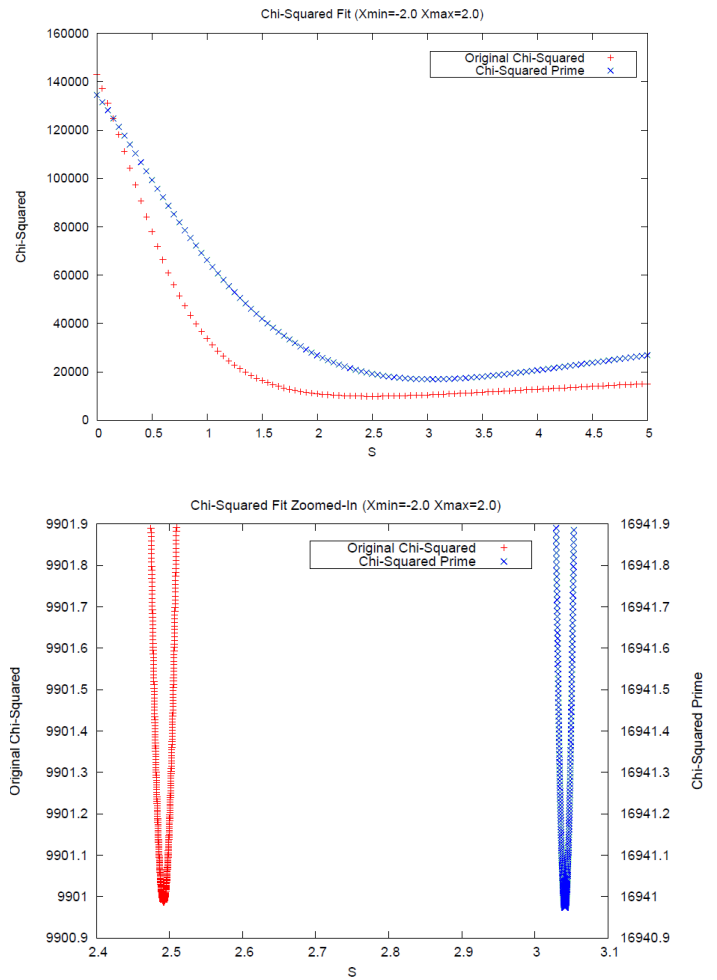
Upon first running the two estimators with $x_{min}=-2.5$ $x_{max}=2.5$ for the new method (below), the results seem to indicate that both the standard method and the new chisq2 are able to correctly estimate the value of the slope at 2.5 (Figure 6) As mentioned previously, the program was then run with slope values of smaller interval 0.0001 in order to better estimate the value of the slope (s) and to look at the error as shown by both straight line fits (Figure 7). The standard chisq1 minimization estimated the slope to be 2.4920 for all values of x_{min} and x_{max} , since it does not use these values in the calculations. The new chisq2 method estimated the slope to be 2.5160. We can see that the standard method is off from the true value by 0.0080, while the new method is off by 0.0160. However, the plot of the chisq1 data below would indicate that the error for the new method is actually less than that of the standard estimator. Figure 6 shows the results of the χ^2 estimation when the range in x is $-2.5 < x < 2.5$. Both curves are compatible with

the correct value for $s=2.5$. Figure 7 shows the regions around the minimum more precisely, indicating that the increase in χ^2 at the correct slope is less than 1 for chisq1 and only slightly larger than 1 for chisq2.



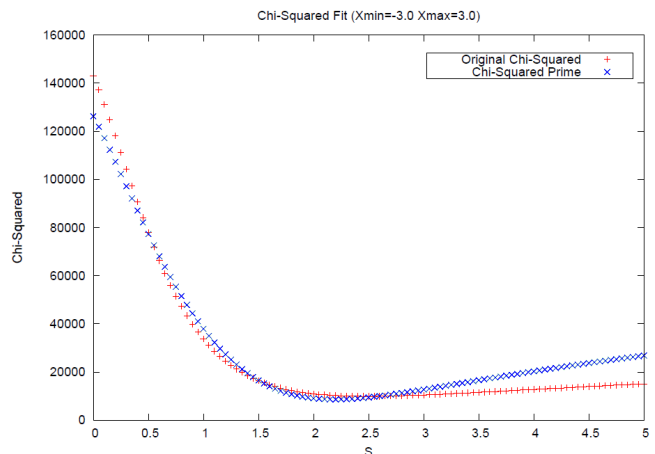
Figures 6 & 7 – Figure 6 is a plot of slope (s) vs. the χ^2 results for chisq1 and chisq2 methods for $x_{min} = -2.5$ and $x_{max} = 2.5$. Figure 7 shows the same results with a smaller range to see the results more accurately with chisq1 on the y1 axis and chisq2 on the y2 axis.

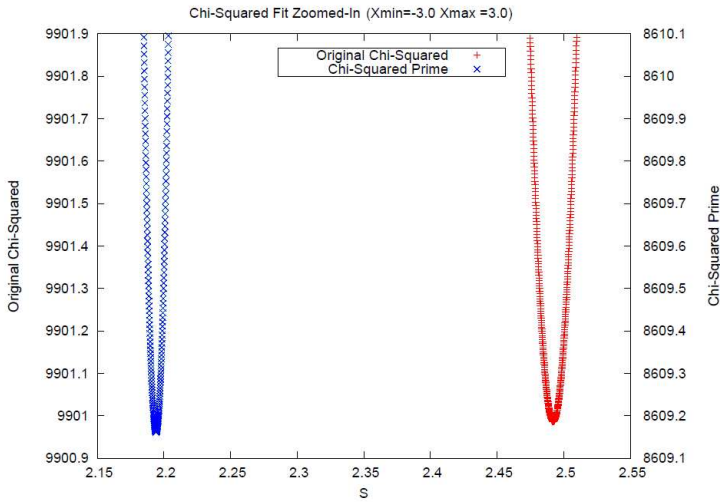
We test the method in the same manner, but change the x_{min} and x_{max} values to test the results for situations when the correct distribution in the x values is unknown. We use the values of $x_{min} = -2.0$, $x_{max} = 2.0$, and $x_{min} = -3.0$, $x_{max} = 3.0$. Figure 8 shows the results when the range in x is decreased to less than the correct distributed range (-2.0 to 2.0). The results for chisq1, do not change since these values do not play a role in the equations for that method. However, figure 9 indicates that the chisq2 is now off by 0.3059 on the positive end, and the result is therefore clearly biased.



Figures 8 & 9 – Figure 8 is a plot of slope (s) vs. the χ^2 results for chisq1 and chisq2 methods for $x_{min} = -2.0$ and $x_{max} = 2.0$. Figure 9 shows the same results with a smaller range to see the results more accurately with chisq1 on the y1 axis and chisq2 on the y2 axis.

In the final test, with $x_{min} = -3.0$ and $x_{max} = 3.0$, the results are shown in Figures 10 and 11. As these two figures indicate, just as when the estimated range is decreased, when the estimated range for x is increased, the results are off by a larger number, and in this case by 0.5413.





Figures 10 & 11 – Figure 10 is a plot of slope (s) vs. the χ^2 results for chisq1 and chisq2 methods for xmin = -3.0 and xmax = 3.0. Figure 11 shows the same results with a smaller range to see the results more accurately with chisq1 on the y1 axis and chisq2 on the y2 axis.

These results would seem to indicate that when the assumed values of xmin and xmax are below the correct values, the estimated slope (s) by the chisq2 method is also below the true value of 2.5, while in the other case, when xmin and xmax are above the correct values, the slope is also estimated by the program to be above the true value.

The overall results seem to show that when the new chisq2 estimator is given the correct values of xmin and xmax, it is able to calculate the correct value of the slope (s) with minimal error, very close to the results shown from the standard chisq1 minimization. However, when the values of xmin and xmax are unknown and are off from the real values, the new method's slope (s) estimate is not as reliable as the standard method. Thus, if the statistical range of a given set of data is known, the correlation is found with better error, than with the initial method.

TEST 2 – CALCULATING TWO PARAMETERS SIMULTANEOUSLY

The final thing we wanted see is if we would be able to calculate several parameters simultaneously in regards to the slope equation, namely a parameter and the error associated with y. Therefore, for this exercise (chisq3) we changed the program and the initial error values in x and y to $\sigma_x = 0.0$ and $\sigma_y = 0.75$. The likelihood equation for this scenario becomes:

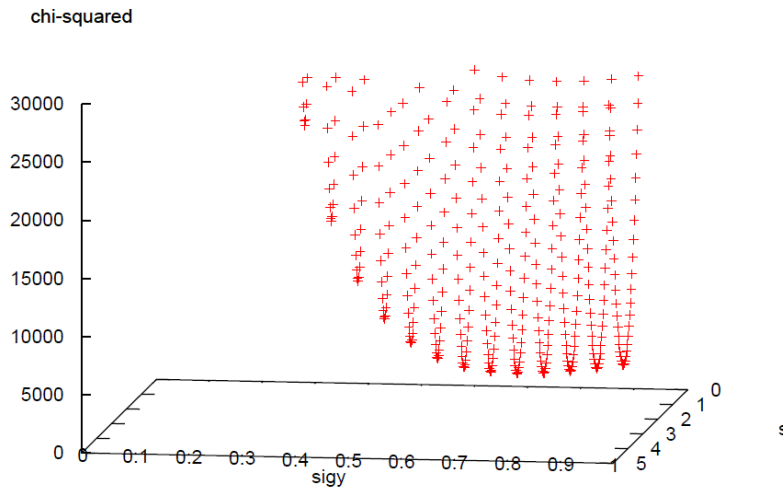
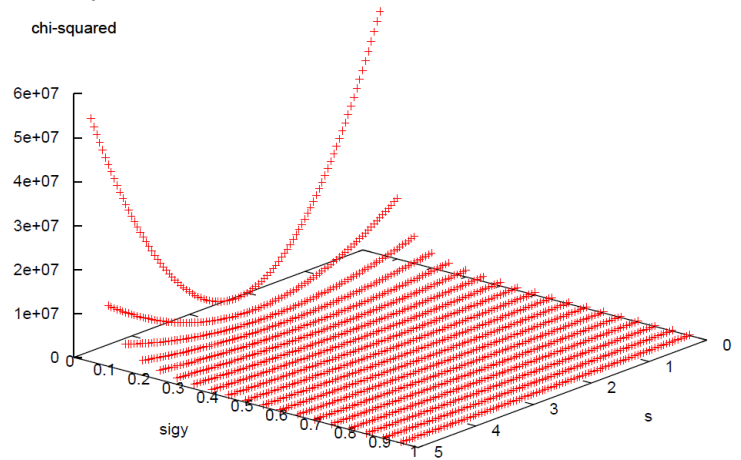
$$L(x, y) = \prod_{i=1}^N \frac{1}{\sqrt{2\pi}\sigma_y} e^{-\frac{(y_i - sx_i)^2}{2\sigma_y^2}}$$

When you take the log of both sides of this equation, the following equation can be used by incrementing both slope (s) and the error (σ_y) to determine the minimum χ^2 :

$$-2 \ln L = \sum \ln \sigma_y^2 + \frac{(y_i - sx_i)^2}{\sigma_y^2} + C$$

RESULTS

The two plots in Figures 12 and 13 show from different views that the results of the chisq3 estimation was able to correctly determine both the slope of the data set s=2.5 and the correct $\sigma_y = 0.75$.



Figures 12 & 13 – Figure 12 shows a 3 dimensional plot of s and σ_y against the χ^2 calculations for the chisq3 method and shows that the minimum is at a value of s = 2.5. Figure 13 is the same plot, but with a small range to indicate that the minimum is also at $\sigma_y = 0.75$.