Laboratory IVb work : Time-delay measurement with lensed supernovae

Y. Carteret

Supervising Assistant: M. Millon, Supervising Professor: F. Courbin.

June 11, 2021

ABSTRACT

The discovery of the first lensed Supernova in 2014, SN Refsdal, makes the original idea of Refsdal (1964) to measure the Hubble constant H_0 achievable. We can measure the time delays between the five images of the SN, in particular we will take the reference of S1 light curve. We propose a comparison with an existing framework, PyCS3, and our own implementation that fits a template of SNe to measure the time delays of SN Refsdal. We study the accuracy and precision of the algorithms on a dataset of generated light curves from SN Refsdal, we blindly measure the time delays and compare them to the true ones. From the analysis made with PyCS3 we established that microlensing was negligeable in a first approach of the time-delay measurements. We finally measured the time delays with the best implementations and achieved S1-S2: 27.33 days; S1-S3: -27.85 days; S1-S4: 20.86 days; S1-Sx: 388.99 days.

1. Introduction

The ΛCDM model is the most favored cosmological model in the sens it can explain a wide range of phenomenons underlined by observations or experiments such as Cosmic Microwave Background (CMB) or large scale galaxy distribution. For a flat topology of the Universe this model is described by six parameters, which are tighly constrained by the CMB observations. The expansion rate of the Universe at the present time, H_0, can either be derived from these six parameters or measured directly at the present time, providing a stringent test for the ΛCDM model. The latest results of the *Planck* mission (Planck Collaboration et al. 2020), which uses the CMB map, gives a value $H_0 = 67.4 \pm 0.5$ km s^{-1} Mpc^{-1} using the so called inverse distance ladder method. The measurements made from background radiation, such as CMB or BAO, emitted during the early stage of the Universe are called Early Universe measurements.

The Late Universe measurements rely on observations of nearby objects. One of the most precise method to measure H_0 from the Late Universe is through the type Ia supernovae (SNe Ia) that are used as standard candles. It is believed that the explosion is the result of a white dwarf reaching the Chandrasekhar mass (Hillebrandt and Niemeyer 2000). The exact scenario of formation is still unclear but two models are preferred. The single degenerate model stands that the white dwarf accretes matter from a companion, that usually is a red giant close enough (Whelan and Iben 1973). The double degenerate model considers a binary system of white dwarfs orbiting each others. Due to emission of gravitational waves they end up to collide and form a new object that can be heavier that the Chandrasekhar mass (Webbink 1984).

Since the mechanism is similar each time, the luminosity, thus the absolute magnitude, can be determined from the light curve of the SNe Ia. With the appropriated calibration it is possible to determine the distance of the SNe Ia and measure H_0 from the distance-redshift relation. In practice the calibration is done through the distance ladder method. The parallax of nearby Cepheid variable stars is used to measure their distance. The Cepheid stars are also standard candles for which a relation between absolute magnitude and period exists. Once calibrated from nearby measurements, further Cepheid stars can then be used to calibrate brighter standard candles such as SNe Ia. The most precise measurements with this method (Riess et al. 2021) gives $H_0 = 73.2 \pm 1.3$ km s^{-1} Mpc^{-1}.

These two different methods differ by 4.4σ. A general trend of the observations made so far (Verde et al. 2019) establishes that the most precise measurements of H_0 from the Early Universe have the same order as Planck Collaboration et al. (2020) whereas from the Late Universe measurements have the same order as Riess et al. (2021). This rises a tension between these two values of H_0, and indicates that there exist either some new physic or some unknown systematics errors in the measurements. It is necessary to have as many independent meaurements of H_0 as possible in order to determine if the tension is real of if it results from systematic errors affecting some of the experiments.

Gravitational lensing provides an independant way to measure H_0 in the Late Universe relying on the time-delay cosmography. Refsdal (1964) proposed to use lensed SNe Ia to measure the time delay between multiple images produced by the gravitational lensing effect. Let's consider the case of one galaxy and a very massive object almost aligned along the line of sight and the host galaxy of the SN Ia being much further away. The General Relativity establishes that the mass can bend the space-time and thus the path of the light rays emitted by the host galaxy. Multiple images of the latest can appear for the observer, the signal emitted by the source will reach the observer at different time. The shift in time depends on the length of the path taken by the light and gravitational delays that both depend on the mass distribution along the path. A SN Ia is a variable object and thus the time delay can be measured from the several light curves observed. Unluckily, finding lensed SN is a rare event and only two have been discovered so far (Kelly et al. 2015; Goobar et al. 2017). It is the main reason why lensed quasars, which are much more commonly found, are usually used to measure time delays despite the advantages that SNe Ia can provide. SNe Ia are also convenient for lens modeling. They are short events, they disappear a few months after the explosion which makes simpler to model the lens since the bright point source is no longer present in the Einstein ring. As mentioned they also are standard candles so, with the right calibration, it provides another estimation of the distance and thus helps breaking the degeneracies of the lensing model. With the use of future imaging time-domain surveys it is believed that detection of lensed SNe will become more common (Goldstein and Nugent 2017).

The time delay is linked to the so called "time-delay distance", $D_{\Delta t}$, that depends on the lens potential, the mass distribution along the line of sight and cosmological parameters. An important remark is that $D_{\Delta t}$ is mainly sensitive to the H_0 value and weakly to the other parameters (Treu and Marshall 2016). The most efficient way to improve the precision of the H_0 measurement is thus through the time-delay measurements.

Kelly et al. (2015) discovered the first strongly lensed SN with multiple images, SN Refsdal. It appeared behind the MACS J1149.5 + 2223 cluster, at the moment of its discovery it was composed of 4 images (S1-S4, see Fig. 1) of the same SN forming an Einstein-cross. Using lens models it has been possible to predict its reappearance (Sx) a couple of months later. Kelly et al. (2015c) found that SN Refsdal is a SN 1987A-like type II peculiar supernovae (SNe II). Type II SNe, which are not standard candles, are triggered by the collapse of the stellar cores of massive stars, the observed light curves are thus higly influenced by the characteristics of the progenitor. Podsiadlowski (1992) highlighted that SN 1987A might be the result of interactions between the progenitor and a binary companion, which would explain the peculiar light curve observed. SNe II are much less convenient than SNe Ia to measure time delays but SN Refsdal is the only avaible data to train different measurement techniques.

The dataset used for the present paper is a set of 991 light curves in the IR F125W and F160W HST filters generated from the SN Refsdal and used by Kelly et al. (2021). Evaluation of time delays using a fitting algorithm will be provided. The idea that motivates the following work is that giving a template of SNe the fitting algorithm should increase its performances.

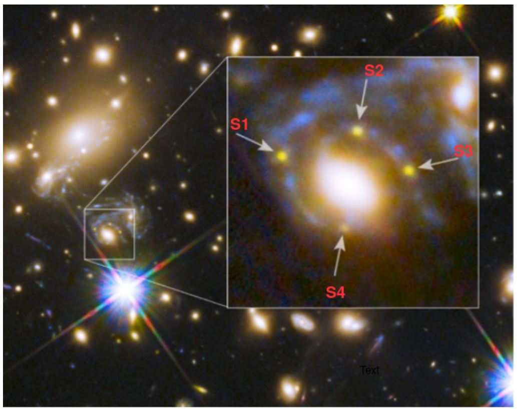

Fig. 1. Color image of S1-S4 images of SN Refsdal, Sx has not appeared yet. Released image from NASA/ESA.

This paper is organized as follows. In Section 2, we give an overview of the basis of the lensing theory. In Section 3, we will provide an overview of the PyCS3[1] package. We will provide a measure of the time delays of the lensed SN II, SN Refsdal, using the PyCS3 package. In Section 4, we will provide the measure of the time delays of the same object with our own implementation that relies mainly on the SNCosmo[2] package. In Section 5, results and ways of improvement for the implementation will be discussed.

2. Theoretical framework

We recall here the basic formalism for the gravitational lensing, the time-delay distance (Schneider et al. 1992) can be evaluated as

$$D_{\Delta t} = (1+z)\frac{D_{ol}D_{os}}{D_{ls}}, \tag{1}$$

where z is the redshift of the lens, D_{ij} are angular diameter distances between the observer, the lens and the source. Angular diameter distance is related to comsological parameters. The Hubble parameter at a given redshift is given by

$$H(z) = H_0\sqrt{\Omega_\gamma(1+z)^4+\Omega_M(1+z)^3+\Omega_k(1+z)^2+\Omega_\Lambda}, \tag{2}$$

which is also written as $E(z)H_0$. The angular diameter distance is thus given by

$$D(z) = \frac{c}{H_0}\int_0^z \frac{\mathrm{d}z'}{E(z')}, \tag{3}$$

for a flat ΛCDM model. From the following definitions $D_{\Delta t}$ is thus inversely proportional to H_0.

The light traveling time through a single gravitational lens for an object with position $\boldsymbol{\theta}$ on the sky is given by

$$t(\boldsymbol{\theta}) = \frac{D_{\Delta t}}{c}\left(\frac{1}{2}(\boldsymbol{\theta}-\boldsymbol{\beta})^2-\phi(\boldsymbol{\theta})\right), \tag{4}$$

with $\boldsymbol{\beta}$ the true position of the source on the sky and $\phi(\boldsymbol{\theta})$ the lens potential. If the alignement between the lens, the source and the observer is good enough, multiple images of the object will be observed and the time delay between two images with positions $\boldsymbol{\theta}_i$ and $\boldsymbol{\theta}_j$ on the sky is given by

$$\Delta t_{ij} = \frac{D_{\Delta t}}{c}\left[\frac{(\boldsymbol{\theta}_i-\boldsymbol{\beta})^2}{2}-\phi(\boldsymbol{\theta}_i)-\frac{(\boldsymbol{\theta}_j-\boldsymbol{\beta})^2}{2}+\phi(\boldsymbol{\theta}_j)\right]. \tag{5}$$

The term in brackets is often called the Fermat potential difference between the two image positions. From these equations it is possible to evaluate $D_{\Delta t}$ knowing the time delays and the lens potential. H_0 can be found by assuming a cosmological model.

The latest equations are valid for the ideal case of a single lens along the light path. In practice gravitational lensing is a cumulative effect. A correction for the mass distribution along line of sight should be taken into account and can be measured independently by galaxy number counts (Rusu et al. 2017) or weak lensing (Tihhonova et al. 2018). Another effect to be taken into account is microlensing, this is the phenomenon of magnification by smaller objects, stars in our case, along the line of sight and usually in the lensing galaxy. The displacement of the light due to microlensing is usually too small to be resolved by telescopes, thus this effect doesn't create a new observable image. The time evolution of magnitude of the object can be measured and the transient objects can be detected. Maps of microlensing can be made or in the study of nearby stars, orbiting planets can be detected (Bond et al. 2004) using this effect. This magnification depends also on the light's path and the time dependant correction will be different for each image of the lensed object. It can thus affect the observed light curves of the different object's images and if not accounted it can lead to wrong evaluation of the time delays.

[1] PyCS3 can be found on github
[2] SNCosmo can be found on github

3. PyCS3 framework

3.1. PyCS3 *package*

The PyCS3 package proposes an effective framework to evaluate the time-delay measurement between light curves. It implements microlensing and gives the possibility to physically constraint it, for example by considering it as a polynomial of a given order. The method that will be used for the purpose of this work is the 'free-knot spline' estimator. This method considers the intrinsic variation as a unique spline, fitted simultaneously on all light curves, whereas the extrinsic variations are modeled as independant splines for each of the light curve. PyCS3 package proposes a large amount of parameters to control the good behavior and convergence of the time delays fit.

We define the following terminology along this work:

- An *estimator* is an algorithm that returns the optimal time delay between two light curves.

- A *set of optimized light curves* is the time shifted, with respect to S1 light curve, set of light curve considering no microlensing.

- A *set of fully optimized light curves* is the time shifted, with respect to S1 light curve, set of light curve considering microlensing.

- A *time-delay estimator*, t_i, is the time delay between light curves S1 and Si as fitted by the estimator. This is given over a single set of light curves.

- A *median time-delay difference*, $\Delta t_{i-t_2}^{+t_1}$, is the median value of the difference between the fitted time delay between S1 and the light curve Si and the true time delay over the full dataset used. The errors are evaluated as the 16^{th} and 84^{th} percentiles. Note that outliers outside the [-30,30] days interval are rejected before computing the median time-delay difference.

- A *critical failure* is when the absolute difference between the time-delay estimator and the true time delay is larger than 30 days.

Most of the previous works done with PyCS3 studied the time delays of lensed quasars. Fig. 2 shows an example of the optimization made on a lensed quasar. This is only an example to illustrate what the package can do, the parameters are randomly chosen in a way to avoid over-fitting or bad behavior. We will define a small set of parameters that will be used latter:

- The knot-step, k_n, is the mean spacing between the knots. Once defined the number of knot is fixed during the optimization but they are free to move.

- The degree of the polynomial microlensing n_{ml}. Notice that a feature of PyCS3 forces us to add polynomial microlensing of degree $n_{ml} = 0$ if we want to apply a shift in magnitude even if it doesn't correspond to microlensing.

Before even fitting any SN light curves we can already raise a few ideas. The precision and accuracy of the package can be enhanced on SNe Ia objects. The algorithm has no physics implemented, it has no idea of what a quasar or a SN is. Because of their random variability this is an advantage for quasars but light curves of SNe Ia follow an almost well defined pattern. This effect might suffer from the goodness of the template that we can find for SNe II since the pattern of their light curve is mostly heterogeneous.

As mentionned the dataset used is composed of sets of light curves generated from SN Refsdal, microlensing is also introduced in these generated light curves. We have access to flux and error over the flux in function of time and the true time delays used for the generation. PyCS3 works with magnitudes so it is necessary to convert the observed flux into magnitudes. The simple formula $m = -2.5 \log F$ can be used without considering any magnitudes systems since we are only interested in the relative variations with respect to S1.

3.2. *Time-delay measurements with* PyCS3

In this section we aim to measure time delays with PyCS3. For this purpose the true time delays introduced in the generated set of light curves is collected after the fit measurement. We propose here an analysis of the impact on the quality of the fit of the two parameters introduced previously. The microlensing introduced along the simulations is taken as a polynomial of order n_{ml} applied on the full light curve, no adaptation in function of the season is done since the SN explosion event is short and spatially small. We also only present measurements made on the F125W filter, this choice is arbitrary and motivated by the better quality of the fits.

In particular two quantities will be useful to study : accuracy and precision. Accuracy describes how far the algorithm is from the true results and precision describes the dispersion of the measurements over the dataset. To avoid at most bias in our measurements the starting points of the algorithm are uniformly distributed around a solution that has been visually determined. For the purpose of this part three sets of parameters are relevant :

- Set 1: { $k_n = 110; n_{ml} = 0$ }
- Set 2: { $k_n = 110; n_{ml} = 1$ }
- Set 3: { $k_n = 65; n_{ml} = 0$ }

Fig. 3 summaries the median time-delay differences obtained with the package for the sets of parameters. We can control the performance of the fit by looking at the distribution of median time-delay differences. In term of accuracy and precision the best distribution should be centered around 0 and well peaked. The worst performances are obtained with microlensing polynomials of order 1. This behavior is confirmed with every set of parameters explored, which are not shown here, when microlensing polynomial's degree is increased the accuracy and precision both drop dramatically.

The performances of Set 1 and 3 are comparable. We observed that reducing the number of parameters, so increasing k_n, implies a trade off between the accuracy on S2, S3 and S4 and Sx images. Fig. 4 illustrates this effect, it is interesting to mention that the precision remains more or less constant (maximum 1.5 days of difference) as a function of k_n. Moreover the behaviors of the four images regarding k_n are correlated, this correlation is even higher for S2 and S3 images. There doesn't exist any knot step for which the bias of the algorithm is zero for every SN image.

The critical failure proportions are shown for our three sets of parameters in Table 1. It allows us to control that the statistical distribution given earlier, which is defined between -30 and 30 days, is complete. In practice, it indicates if the tails of the distributions really goes to 0 on a large range. Sets 1 and 3 distributions are complete for every images. But Set 2 distributions for images S_4 and S_x are influenced by larger values of the median time-delay differences. From Fig. 3 we already see that the distributions are cut at 30, so the median time-delay differences values are biased since the statistic outside the range of study is large. The median time-delay difference is thus larger if the full dataset is taken into account. Using the critical failure proportions we can once again discard the $n_{ml} > 0$ parameter.

It has been established that the Set 1 is the best set of parameters on this dataset. Notice that on Fig. 4 the value $k_n = 120$ can

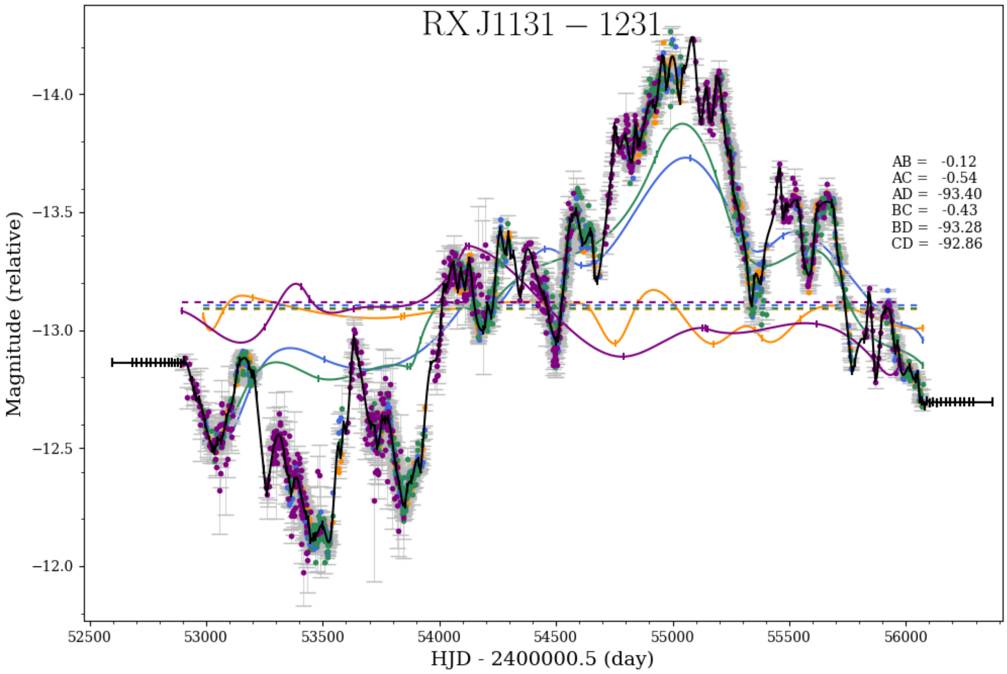

Fig. 2. Fully optimized light curves of RX J1131 − 1231 by PyCS3. Each color represents a different image of the lensed quasar, the black spline is the intrinsic variation, dashed lines are average magnitudes and colored splines represent microlensing attached to the corresponding light curves. Light curves are released data from Tewes et al. (2013).

appear to be more accurate overall. The study of its distribution reveals that it is less precise on every image, together with the large bias on S4 it makes the performances worse. The Set 1 is also characterized by a distribution that looks like Gaussian and centered not too far from 0 for each image which also justifies the choice made for the evaluation of the error in the median time-delay differences.

In order to visually control the performance and the good behavior of the fit, we provide in Fig. 5 the fit made with the Set 1 on the true data of SN Refsdal (Rodney et al. 2016). The fit is the most accurate for the bump part of the light curve.

	S_2	S_3	S_4	S_x
Set 1	1.83	0.20	0.92	0.50
Set 2	1.13	0.20	7.47	18.83
Set 3	1.63	0.31	0.92	0.41

Table 1. Percentage of critical errors with PyCS3 obtained for every images and the three sets of parameters.

From all the observations made with the results obtained with PyCS3 a general trend can be guessed, reducing the number of free parameters leads to better accuracy overall, at least untill we reach a too small number of parameters to properly fit the data. In order to enhance the performances we should be aware of the behavior of the statistic on long range, the accuracy and precision should be at least as good as already obtained and computational time should not explode. The method should not require microlensing to work since we observed poor performance with microlensing.

4. SNCosmo package

The SNCosmo package porposes a framework to fit SN light curves with a bunch of implemented templates. It includes a couple of effects that can be added such as Milky way's or host galaxy's dust. The lastest won't be needed for this work since we are only interested in relative variations, we can consider that dust along the line of sight has mostly not affected these variations. The *model* features, that describe the template, are defined by some parameters that might be different depending on which template is used. This package doesn't propose microlensing fitting nor any gravitational extrinsic variations fitting since it wasn't made to measure time delays. We propose here an implementation using SNCosmo to measure the time delay between the S1 image of SN Refsdal and S2-Sx, neglecting any microlensing. The motivating idea is to reduce the number of parameters of the intrinsic variation, the spline in PyCS3, by finding the template of the SN. In particular three models will be studied. Two parameters are common to every models, the redshift z and the time-zero-phase t_0. Depending on the model this quantity can be at the explosion or at the maximum of flux. Notice that when we refer to templates instead of models we only include the parameters of the model that are optimized, z is known thus never fitted.

4.1. SNe templates

Most of the templates that can be found in SNCosmo are light curves of previous SNe corrected from different effects. For the most of them the templates are quite simple, they depend on two parameters which correspond to a time shift and a magnitude shift. For the purpose of this study we will consider three templates. The first has been chosen because it was identified as sim-

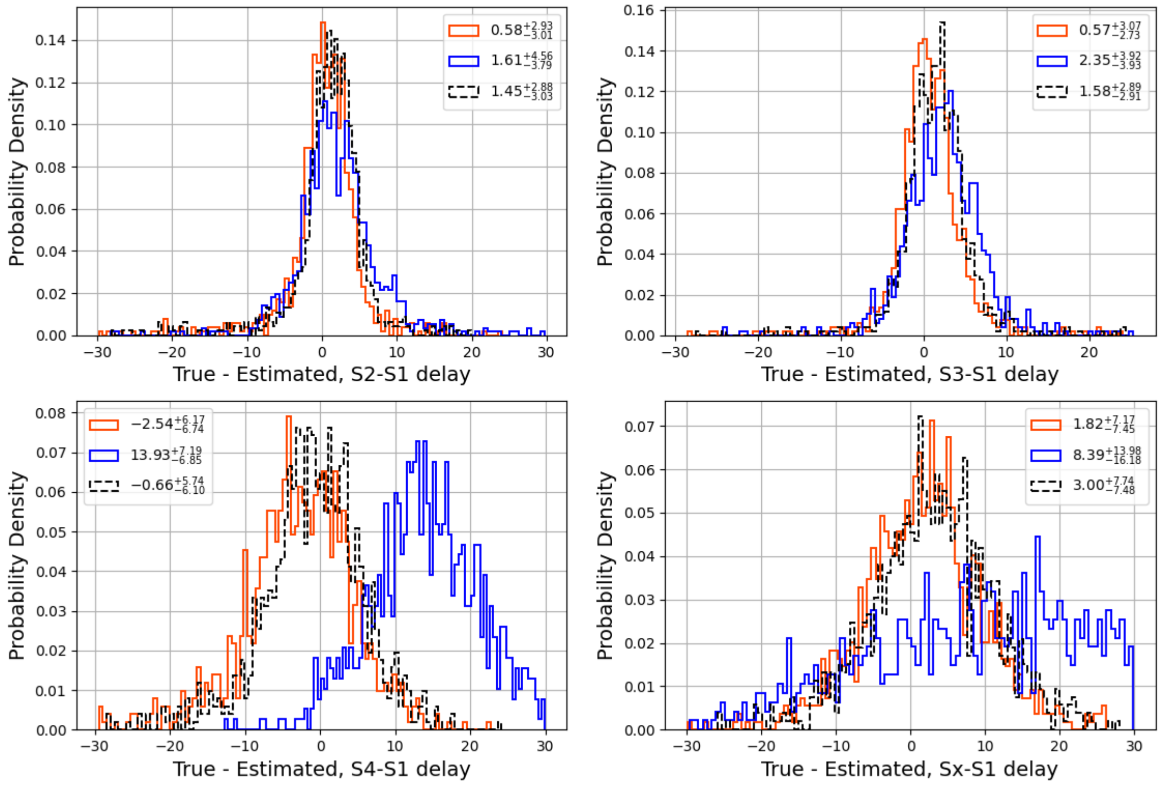

Fig. 3. Distribution of the median time-delay differences of SN Resfdal given for the four studied images and the three set of parameters as found by PyCS3. Red, blue and black curves represent respectively the Set 1, 2 and 3 of parameters. The critical failures have been filtered out from the distributions.

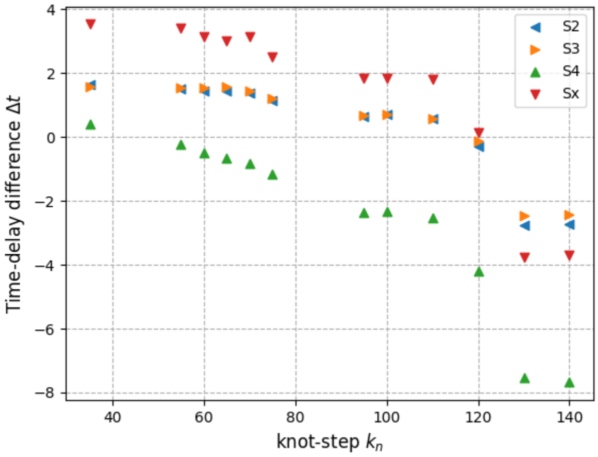

Fig. 4. Median time-delay difference obtained by PyCS3 with $n_{ml} = 0$ as a function of k_n. Errors on median time-delay differences remain almost constant as a function of k_n for every images.

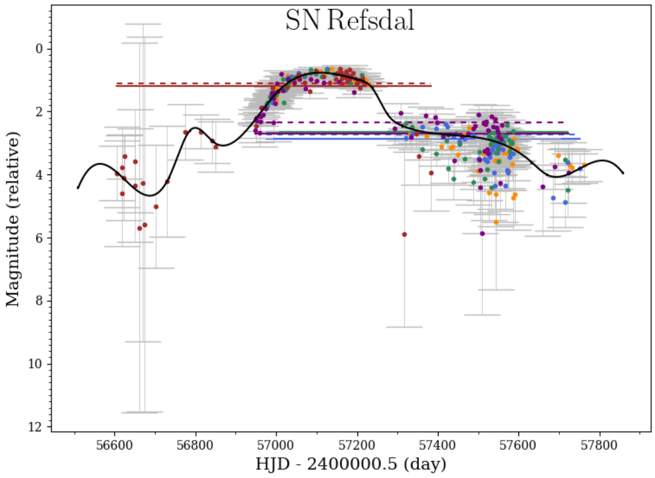

Fig. 5. Fully optimized light curves of SN Refsdal by PyCS3 with Set 1. Each color represents a different image of the lensed SN, the black spline is the intrinsic variation, dashed lines are average magnitudes and colored splines represent microlensing attached to the corresponding light curves. Respectively orange, blue, green, purple and brown datapoints are S1, S2, S3, S4 and Sx images. The time-delay estimators are : $t_2 = 20.45$, $t_3 = -36.75$, $t_4 = 27.54$ and $t_x = 380.17$.

ilar to SN Refsdal (Kelly et al. 2015c), the two last because of the good match with the data.

SN 1987A is a SNe II that has exploded in 1987 in the Large Magellanic Cloud at 50 kpc (Arnett et al. 1989). It was the first SN visible by the eye of the 20th century. A SNe template, has

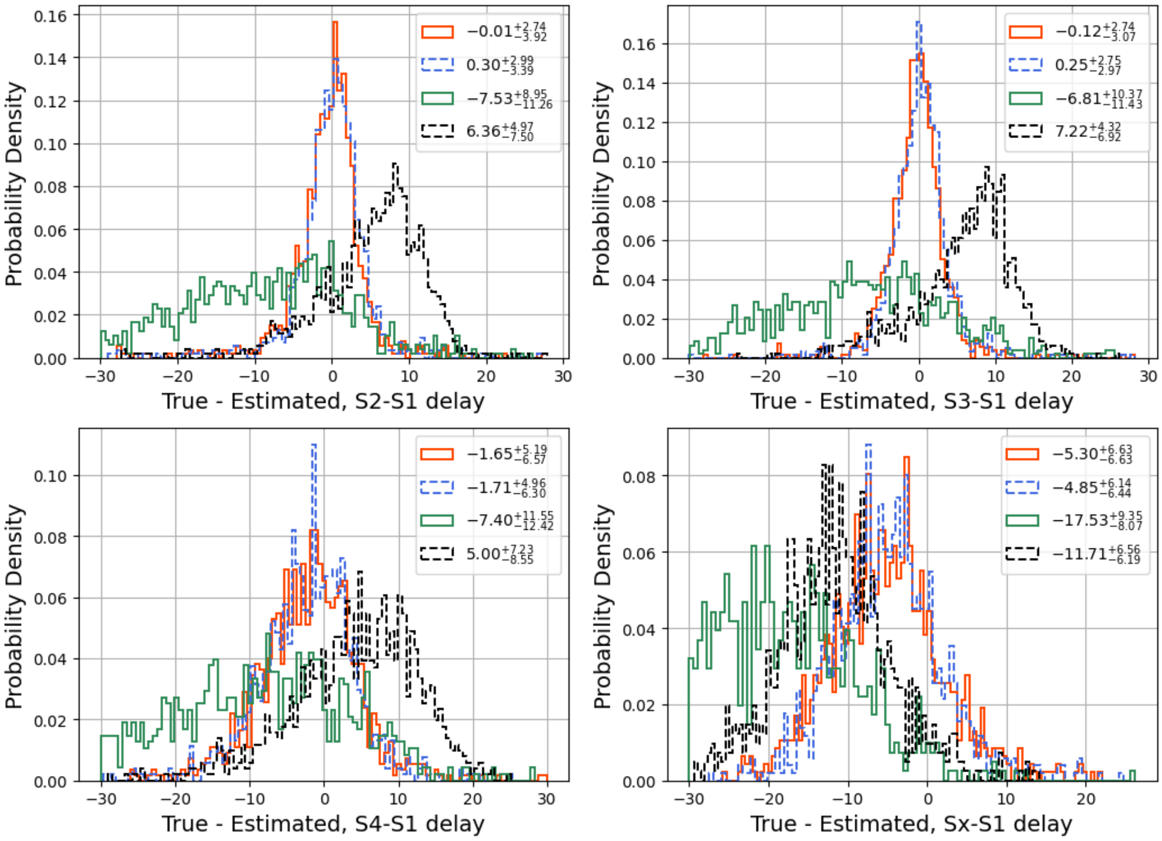

Fig. 6. Distribution of the median time-delay differences of SN Resfdal given for the four studied images and different implementations found with SNCosmo framework. Red, blue, black and green curves represent respectively the Set 4, 5, 6 and 7. The critical failures have been filtered out from the distributions.

been made from the spectrums taken in numerous bands. It is a convenient template to use since the SN used is really close, and thus the data are almost not affected by cosmoligical effects.

SN 2006aa was discovered in 2006 in NGC 3947 at around 87 Mpc. It is a type IIn SN. Its light curves are characterized by a "Gaussian" shape with an initial rise that culminates at maximum light approximately 50 days after discovery for all the filters (Taddia et al. 2013).

It has been noticed that the shape of SN Refsdal was similar to a SN Ia with a good rescaling of the time axis of the data which is an interesting feature that we decided to exploit. Despite the fact SNe Ia are similar in term of mechanism of explosion they usually have variations and the observed light curves are not perfectly superposable. Several empirical correlations have been determined like for example stretch effect (Phillips 1993) or color effect (Tripp 1998). This explains why different templates exist and how important is the research of the good template. The SALT2 model is a description of the spectral flux density as a function of time in the rest frame of the SNe, it is also the most complex SNe Ia template in the package. For a given wavelength λ of observation it takes the form :

$$F(t, \lambda) = x_0 \left(M_0(t, \lambda) + x_1 M_1(t, \lambda)\right) \times 10^{-0.4 CL(\lambda) c}, \tag{6}$$

where x_0, x_1 and c are the scaling parameters of the model. $CL(\lambda)$ is a color law and M_i is the i^{th} order deviation of the sequence. The SALT2 is an empirical model of SNe Ia, it is built from a large dataset of SNe (see Guy et al. 2007, for more information about the model).

4.2. *Implementations*

Compared to PyCS3 the light curves object in SNCosmo are different. The manipulation of the two frameworks at the same time need to be done carefully since SNCosmo requires a couple of parameters that are irrelevant for PyCS3. The redshift of the SN must be known, the zero-point and magnitude system need to be specified to scale the flux for the magnitude calculation. Since only relative magnitude matters these two parameters will be taken as default: 25 zero-point and *AB* magnitude system. Despite that these parameters are irrelevant for the study they must be specified since SNCosmo use them to compute the flux given a model and its parameters.

The proposed implementations rely on three different estimators to evaluate the time delays. For simple templates, the ones that depend on two parameters, the well optimized fitting tool of SNCosmo can be used. Five fits are made with the same model, the time delays and magnitude shifts are simply the differences and the logarithm of the ratios of the corresponding parameters with respect to S1 image. The two other estimator rely on the

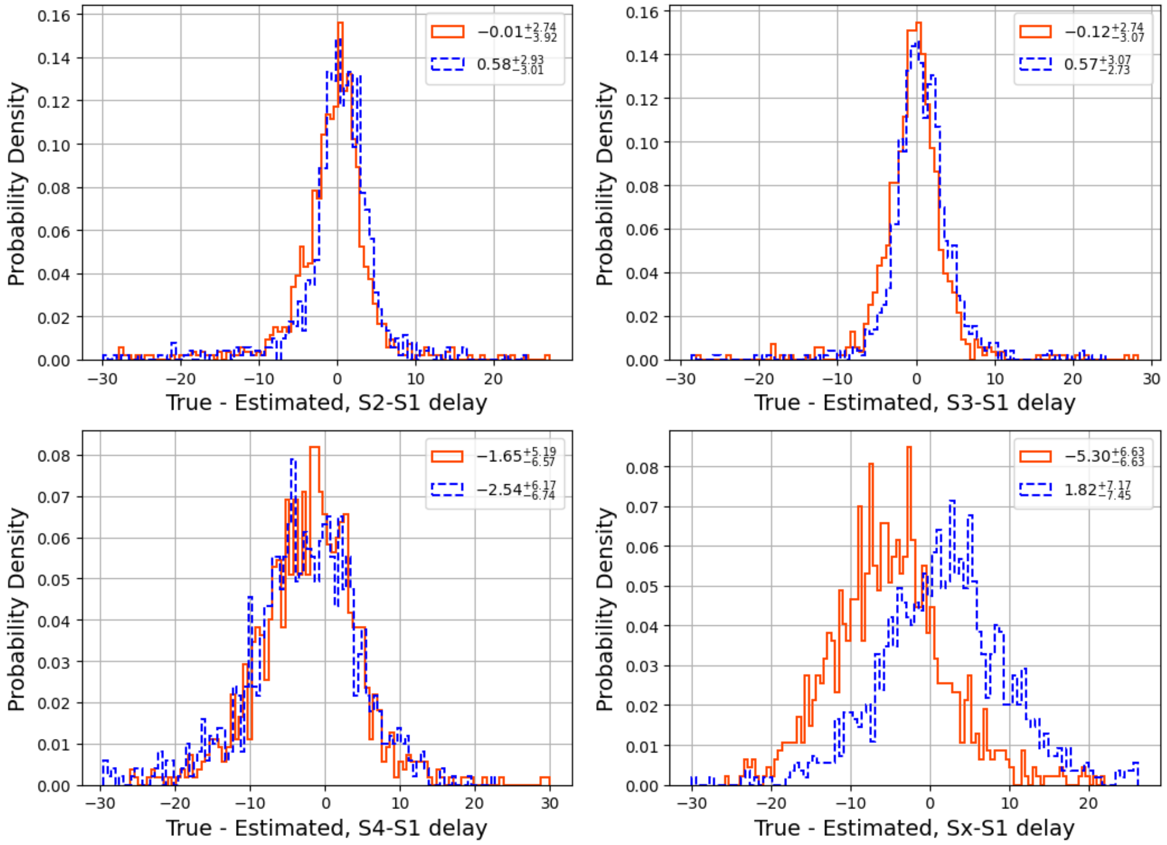

Fig. 7. Best distributions of the median time-delay differences. The red curve is the best distribution with our implementation while blue curve is obtained with PyCS3. The critical failures have been filtered out from the distributions.

computation of a χ^2. SNCosmo proposes a function to evaluate the χ^2 of a light curve with a given model and its parameters. The first method consists to merge the images' light curves, introduce independant time shifts and flux scalings for images S2-Sx, compute the χ^2 over this large light curve and minimize it. In this method the parameters of the model remain the same for each image. The idea of the method is to optimize the different time shifts and flux scalings of the data to match a template. The second method computes the χ^2 of each image's light curve independently and sum them. In particular, the parameters of the model that correspond to time and magnitude shifts varies with respect to the image whereas the rest remain fixed. The total χ^2 is given by

$$\chi^2_{tot} = \sum_{\{Si\}} \chi^2_i(\{P_i + \Delta P_i\}), \qquad (7)$$

where P_i are the model's parameters of the Si image. The idea of the method is to find the time and magnitude shifts for the modeled light curves of each image to superpose. The advantage of the latest is that the given data are not modified, flux and errors on it are not scaled at all to compute the time delays. Note that these two estimators also implements the time axis rescaling that we mentionned earlier.

4.3. Time-delay measurements with the SNCosmo *framework*

In this part we propose a similar analysis that has been performed for PyCS3. We will study the impact of two parameters: the SN model and the method of χ^2's computation. Only a few sets of parameters will be given in this work:

- Set 4: { SALT2; no merge }
- Set 5: { SALT2; merge }
- Set 6: { SN 1987A; simple }
- Set 7: { SN 2006aa; simple }

	S_2	S_3	S_4	S_x
Set 4	1.21	0.10	0.60	0.30
Set 5	1.82	0.10	1.31	1.11
Set 6	9.96	7.02	14.15	24.21
Set 7	1.05	0.00	0.84	1.36

Table 2. Percentage of critical errors with SNCosmo obtained for every images and the four sets of parameters.

The median time-delay difference distributions for the four sets can be found on Fig. 6. By just looking at these distributions the model that fits the best the SN is a SALT2 template

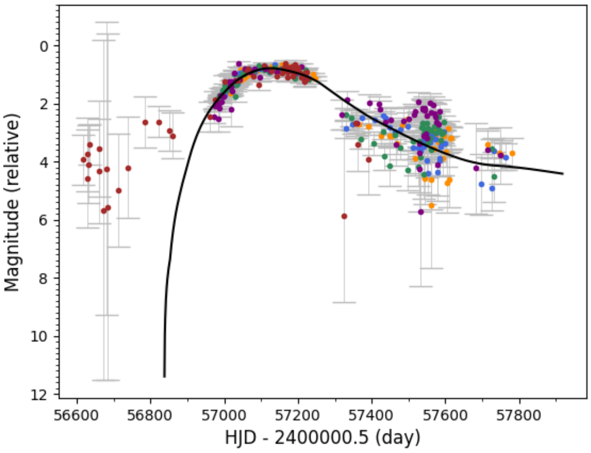

Fig. 8. Optimized light curves of SN Refsdal by our implementation Set 4. Each color represents a different image of the lensed SN, the black spline is the modeled light curve. Respectively orange, blue, green, purple and brown datapoints are S1, S2, S3, S4 and Sx images. The time-delay estimators are : $t_2 = 27.33$, $t_3 = -27.85$, $t_4 = 20.86$ and $t_x = 388.99$.

with a time rescaling of $1/(z+1)^2$, which is empirically found. The exact value might vary depending on the set of light curves used but it remains close to the value used. We chose to fix its value and not fit it to accelerate the algorithm. The SN 2006aa model seems to introduce a bias in the measurements of median time-delay. The distribution is still peaked around a value but accuracy and precision are worse. Surprisingly the worst template is SN 1987A which has distributions similar to uniformely distributed random guesses.

The two estimators that rely on the computation of a χ^2 give similar results. The two estimations of the median's value, for which the numerical error follows $\frac{\sigma}{\sqrt{N}}$, are not compatible for S2, S3 and Sx images. It can be noticed that lower error of the median time-delay differences for every set is higher than the upper one on images S2-S4. When the algorithm is driven out of the solution, which means the time-delay estimator is far from the true time delay by at least 10 days, it is more likely to under evaluate it.

The good behavior on long of the statistic have also been checked in Table. 2. Similarly to what has been observed with PyCS3, the very poor fits' distributions are incomplete. This time, despite low performances, the bad fit produced by SN 2006aa gives a more or less controlled long range statistic.

Several tests have been performed in order to evaluate the robustness of the χ^2 computation implementations. By looking at the distributions of Set 4 and 5, we can see that the probability density starts to become negligeable around ±10 days. The starting points of the algorithm are randomised in a range of ±10 days around a typical solution, so we can ask ourselves if these two are related. By increasing the range to ±20 days and observing only small changes in the distributions we can exclude such a relation. The typical solution has been also studied, by changing it by a few days in one or another direction. Once again no major changes were detected. We also tried to look at the distribution without any fitting tool, which corresponds to the difference between the starting points and the true solution. The SALT2 increases the performances compared to random guesses, precisions is better by a factor of at least 2 and accuracies are enhanced. The solution found is on average driven away from the guess between 2 and 5 days depending on the SN image.

Compared to the performances of PyCS3, see Fig. 7, our implementation best performances are more accurate and evenly precise for images S2, S3 and S4 and less accurate but more precise for image Sx. The fit in Fig. 8 is made on the same set of light curves presented in Fig. 5, similar fits with the last sets can be found in Fig. 9, 10 and 11. Even for this single fit we didn't use the fitting tool for the time axis rescaling and used the default one provided before. Both solutions are visually acceptable and similar. Notice that the spline was more flexible, and changed by a lot from one to another set of light curves with PyCS3. With our implementation the "spline" is pretty much the same for each set of light curves. Notice also that the PyCS3's spline has 12 parameters while ours has only 5 parameters (one for the time axis rescaling) for SALT2 and 2 for the others.

The proposed implementation provides an algorithm to fit the extrinsic variation of a set of light curve. We saw that the performances are comparable and can even be better than PyCS3 on S2 and S3 without microlensing. We can also mention another feature that has not been exploited here is the possibility to use several bands and fit the parameters' model with more constraints. This feature is a difference with PyCS3 where the different bands must be treated separately.

5. Discussion and futur prospects

Using PyCS3 the low performances with microlensing indicates that the main component of the dataset is time independant variation, or at least that time dependant variations are similar in each image, and that the difference in microlensing between all the images introduced is small. This is also confirmed by the shape of the observed light curves which show no features of microlensing. The observed behavior corresponds to an overfit due to the too high number of degrees of freedom.

The correlation between the different images regarding the evolution with k_n is expected since the knot step is a feature of the intrinsic variation of the light curves. The S2 and S3 light cruves are similar, they have the same order of magnitude and are the brightest images, so the fact that their evolutions are almost identical is not a suprise.

The bump part of each light curve represents the most intense portion in term of flux, with the highest signal/noise ratio. Without extrinsinc variations, it would be possible to extrapolate the date of the maximum of each light curve and measure the time-delay estimators as the differences in the date. This implies that the first parts of the light curves are the main contributions to the time-delay estimators. We can observe this behavior on Fig. 5 where the first part of the merged light curves is well fitted by the spline while the tail is less.

The templates of SNe II are often too simple, only 2 parameters, and light curves are too different from one SNe II to another that templates don't help to find more accurately the time delays. In this context lensed SNe II are not more convenient to measure time delays than quasars. But we showed that in this particular case of SN Refsdal we could find empirically a more complex template that matches the data. The SALT2 is the most efficient template for this dataset. SALT2 is also the most flexible, it has 3 more parameters than the two other templates studied, even 4 if we count the time axis rescaling factor. Notice also that the fact the two estimators presented are numerically not compatible tends to show that they are somehow different.

From the performances obtained with our χ^2 computation implementation we can conclude that it suffers from local minima. When the time axis rescaling is fitted if the guess for this parameter isn't fine enough the solution found is irrelevant and almost the same as the guess. This probably comes from the use of *scipy.minimize* function, the gradient descent might be too simple for the purpose of this work.

The long range statistic behavior observed is expected. A template that represents well the shape of the light curve should help to avoid critical failures, just as the local minima dependance of our χ^2 implementation. A template that is not perfectly fitting the light curve, especially if the width of the SN's flux bump is different, will increase the error made on the time delay.

The Set 4 and Set 5 are very effective on S2 and S3, these two light curves are similar to S1 which is our reference while S4 and Sx are much different. In particular S4 light curve seems to suffer from demagnification. In order to simulate this effect the generated light curves of S4 have been made with higher microlensing contribution. As expected the largest signal/noise ratio light curves (S2 and S3) are the most accurate measurements. With PyCS3 the bias is larger on S4 image while with our implementation it is larger on Sx. Once again this effect could be predicted, using PyCS3 without microlensing we expect the light curve with the highest microlensing contribution to be badly fitted whereas using our implementation the overall shape of the light curve is more relevant. Sx light curve is the most noisy and incomplete light curve which makes the shape hard to properly identify and introduces a large bias on the measurements. This effect also justifies the lack of symmetry in the distribution of Set 6 which indicates a bias of the estimator.

Time-delay estimators given by PyCS3 and our implementation using the four sets of study are both visually acceptable solutions. By eyes we see that the bump of the SN is bestly fitted by Set 4, we recall that in absence of microlensing it is the most important part to fit in order to measure time delays. The first points of Sx image correspond to a period before the explosion of the SN and can be interpreted as error measurements, it is a good feature that they are not fitted with our implementation. Since no physics is implemented in PyCS3 the algorithm might have trouble to distinguish the most relevant points to fit since it uses 3 of its 12 knots only to fit the spline on these first datapoints. The spline is a common feature, it is important for it to not be highly influenced by only one of the light curves. In this sens the fit proposed by PyCS3 is visually worse than the one obtained with the Set 4. Take care that the trend of the left tail of the SN doesn't follow perfectly the data with Set 4 and Set 5 and would indicate that the choice of the time axis rescaling is probably nonoptimal for this set of light curves.

We would like to raise a couple of ideas in order to enhance the proposed implementation:

- To avoid the high dependance from the starting points of our implementation it would be interesting to implement Monte Carlo Markov's Chain (MCMC). Such a method would reduce the dependance and allows the algorithm to explore more the phase space of parameters. It wouldd also give an efficient way to evaluate the erros made on the time delays for a single set of light curves.

- The time axis rescaling empirically found for this work is a feature to explore. We can ask ourselves a couple of questions. Is an underlying physical phenomenon acting here ? Can such a feature be found for other SNe ? Is the enhancement of performances with the use of $1/(z+1)^2$ factor pure luck or is it a general trend ?

- The implementation proposed is independant of PyCS3 and thus doesn't allow the user to add microlensing to measure the time delays. The idea to fully adapt PyCS3 to lensed SNe would be to merge our implementation in PyCS3. It would require to create a super class of *spline*, depending of which object is studied the algorithm will optimize a SNe template or a spline as already implemented in PyCS3.

- The automatised research of the right SNe template should also be included.

- We recall that in our analysis we never used the fitting tool of the time axis rescaling and fixed it at the start to a solution that we found on a single set of light curves. Even if the factor remains close, the difference is around 0.02-0.03, we strongly believe that the fit on every single set of light curve would increase even more the performances.

6. Conclusion

The first part of this paper describes the PyCS3 framework which is our point of comparison for SNe time delay measurements. In the second part we proposed an implementation in the perfect case of negligeable microlensing. It has been established that our implementation has slightly better performances for images S2, S3 and S4 than PyCS3 while Sx image seems highly biased. The use of some empirically found rescaling of the SN time axis led us to enhancement of performances due to a more accurate template to fit the SN. This supports the main idea of this work, giving a good template of SNe it has been possible to obtain good performances similar to PyCS3.

The results from this paper should serve as a first approach to the developpement of PyCS3 as a lensed SNe time delay measuring tool. Indeed quite a lot of work is still required to fully adapt PyCS3 to lensed SNe. The track is promising and could help in a near future to add constraints on the H_0 measurement from the Late Universe.

References

Arnett D.W., Bahcall J.N., Kirshner R.P. and Woosley S.E., 1989, ARA&A, 27, 629

Bond I.A. et al., 2004, ApJ, 606, L155

Guy J., Astier P., Baumont S. et al., 2007, A&A, 466, 11

Goldstein D.A. and Nugent P.E., 2017, ApJ, 834, L5

Goobar A. et al., 2017, Science, 356, 291

Hillebrandt W. and Niemeyer J.C., 2000, ARAA, 38, 191

Kelly P.L. et al., 2015, Science, 347, 1123

Kelly P.L., Brammer G., Selsing J. et al., 2015c, arXiv:1512.09093

Kelly P.L. et al., 2021, being evaluated

Phillips M.M., 1993, ApJ, 413, L105

Planck Collaboration et al., 2020, arXiv:1807.06209v3

Podsiadlowski P., 1992, PASP, 104, 717

Refsdal S., 1964, MNRAS, 128, 307

Riess A.G., Casertano S., Yuan W., Bowers J.B., Macri L., Zinn J.C. and Scolnic D., 2021, ApJL, 908, L6

Rodney S., Strolger L.-G., Kelly P.L. et al., 2016, ApL, 820, 50

Rusu C.E. et al., 2017, MNRAS, 467, 4220

Schneider P., Ehlers J. and Falco E.E., 1992, Gravitational Lenses.

Taddia F. et al., 2013, A&A, 555, A10

Tewes M., Courbin F., Meylan G. et al., 2013, A&A, 556, A22

Tihhonova O. et al., 2018, MNRAS, 477, 5657

Treu T. and Marshall P.J., 2016, A&ARv, 24, 11

Tripp R., 1998, A&A, 331, 815

Verde L., Treu T. and Riess A.G., 2019, Nature Astronomy, 3, 891

Webbink R.F., 1984, ApJ, 277, 335

Whelan J. and Iben Jr.I., 1973, ApJ, 186, 1007

Appendix

We provide the fits obtained with our implementation using the Set 5, Set 6 and Set 7.

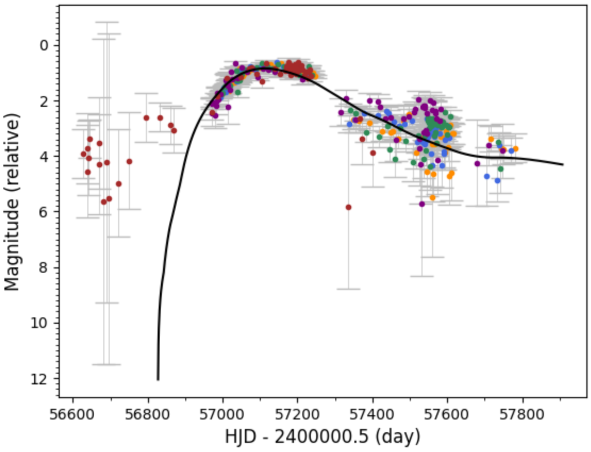

Fig. 9. Optimized light curves of SN Refsdal by our implementation Set 5. Each color represents a different image of the lensed SN, the black spline is the modeled light curve. Respectively orange, blue, green, purple and brown datapoints are S1, S2, S3, S4 and Sx images. The time-delay estimators are : $t_2 = 20.61$, $t_3 = -38.98$, $t_4 = 24.10$ and $t_x = 378.40$.

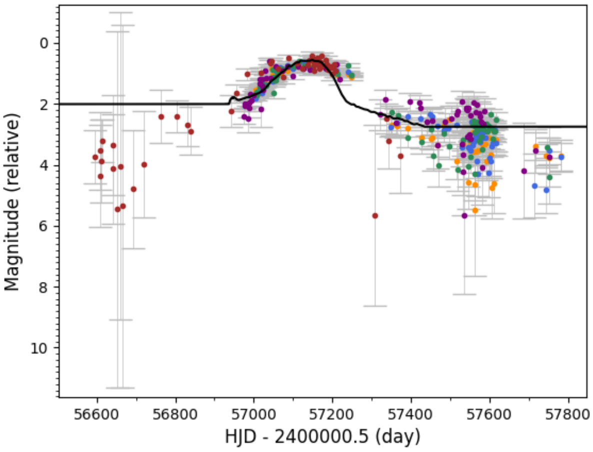

Fig. 10. Optimized light curves of SN Refsdal by our implementation Set 6. Each color represents a different image of the lensed SN, the black spline is the modeled light curve. Respectively orange, blue, green, purple and brown datapoints are S1, S2, S3, S4 and Sx images. The time-delay estimators are : $t_2 = 9.54$, $t_3 = -48.90$, $t_4 = 19.02$ and $t_x = 407.83$.

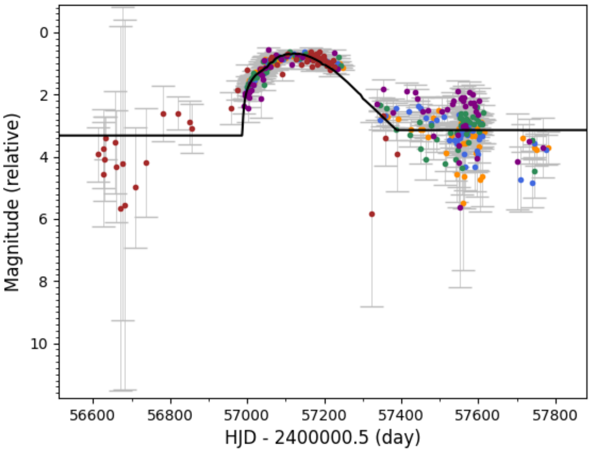

Fig. 11. Optimized light curves of SN Refsdal by our implementation Set 7. Each color represents a different image of the lensed SN, the black spline is the modeled light curve. Respectively orange, blue, green, purple and brown datapoints are S1, S2, S3, S4 and Sx images. The time-delay estimators are : $t_2 = 14.74$, $t_3 = -43.54$, $t_4 = 2.35$ and $t_x = 390.65$.