



Universidad Autónoma de San Luis Potosí
Facultad de Ingeniería
Centro de Investigación y Estudios de Posgrado

Data science model to estimate accurately time delays of gravitationally lensed quasars

Para obtener el grado de:
Doctorado en Ciencias de la Computación

Presenta:
Luis Manuel León Anaya

Asesor:
Dr. Juan Carlos Cuevas Tello

San Luis Potosí, S. L. P.

Febrero de 2023





Universidad Autónoma de San Luis Potosí
Facultad de Ingeniería
Centro de Investigación y Estudios de Posgrado

Modelo de ciencia de datos para estimar con precisión los retrasos de tiempo de los cuásares con lentes gravitacionales

Para obtener el grado de:
Doctorado en Ciencias de la Computación

Presenta:
Luis Manuel León Anaya

Asesor:
Dr. Juan Carlos Cuevas Tello

San Luis Potosí, S. L. P.

Febrero de 2023



16 de junio de 2022

**M. C. LUIS MANUEL LEÓN ANAYA
P R E S E N T E.**

En atención a su solicitud de Temario, presentada por el **Dr. Juan Carlos Cuevas Tello** Asesor de la Tesis que desarrollará Usted, con el objeto de obtener el Grado de **Doctor en Ciencias de la Computación**, me es grato comunicarle que en la sesión del H. Consejo Técnico Consultivo celebrada el día 16 de junio del presente año, fue aprobado el Temario propuesto:

TEMARIO:

"Modelo de ciencia de datos para estimar con precisión los retrasos de tiempo de los cuásares con lentes gravitacionales"

1. Introducción
 2. Antecedentes de la Astronomía y la Astrofísica
 3. Complejidad de los datos provenientes de sistemas de lentes gravitacionales fuerte
 4. Métodos actuales para la estimación del retraso de tiempo
 5. Modelo de estimación de retrasos de tiempo que incluye técnicas de preprocesamiento
 6. Resultados experimentales
 7. Conclusiones y trabajo futuro
- Referencias
Apéndice

"MODOS ET CUNCTARUM RERUM MENSURAS AUDEBO"

A T E N T A M E N T E



DR. EMILIO JORGE GONZÁLEZ GALVÁN
DIRECTOR.
UNIVERSIDAD AUTÓNOMA
DE SAN LUIS POTOSÍ
FACULTAD DE INGENIERIA
DIRECCION

Copia. Archivo.
*etn.

This thesis is dedicated to my beloved son, wife, mother, and father.

Acknowledgments

First of all, I would like to express my gratitude to my adviser Dr. Juan Carlos Cuevas Tello. Since this research was possible thanks to his guidance and support.

I am also most grateful to my research committee, Dr. Luis Javier Ontañón García Pimentel, Dr. César Augusto Puente Montejano, Dr. Carlos Soubervielle Montalvo, and Dr. Octavio Valenzuela Tijerino, without her support and assistance, this investigation would not have been completed. Also, I would like to thank the Universidad Autónoma de San Luis Potosí for their support in carrying out this project, as well as for allowing me to continue with my studies.

Finally, I would like to thank the Lord for guiding me.

Abstract

The time delay measurement among multiple lensed images forthcoming from a strong gravitational lens system is considered a direct and powerful probe to estimate cosmological parameters, in particular the Hubble parameter. Since gravitational lensing systems have risen as one of the main devices to roughly calculate the mass structure of different sorts of stellar objects such as exoplanets, stars, and galaxies among others. In particular, in strong lensing, the correct projection of the distribution of matter and a precise measure of the density of dark energy relies heavily on an accurate time delay estimation of the deflected light curves. However, astronomical data is inherently complex, among its traits are the noise, unevenly sampled, highly distorted, and large observational gaps, among others. Many systematic, random, and unknown issues affect the quality of data, like defects, in photometry, telescope problems, air mass variation, and cloud passage among other atmospheric events that can undermine these signals. Furthermore, the amount of astronomical data available is beyond the capacity of traditional analytical tools. In this way, we propose to introduce a data science methodology to handle the large volume of data. To test the precision of the models used in this research, a set of artificial data from the Time Delay Challenge 1 (TDC1) is used, this dataset tries to reproduce the behavior of real light curves with a previously known delay. Despite many time delay estimation tools that have been developed to tackle these issues, is imperative to increase the precision. Therefore, we emphasize the preprocessing stage to transform the data and diminish its complexity. The empirical evidence shows that with data transformation of the mocked dataset from TDC1, there is a notable improvement in the accuracy of the time delay estimation.

Resumen

La medición del retraso de tiempo entre múltiples imágenes con lentes provenientes de un sistema de lente gravitacional fuerte se considera una prueba directa y poderosa para estimar parámetros cosmológicos, en particular el parámetro de Hubble. Dado que los sistemas de lentes gravitacionales se han convertido en uno de los principales dispositivos para calcular aproximadamente la estructura de masa de diferentes tipos de objetos estelares, como exoplanetas, estrellas y galaxias, entre otros. En particular, en las lentes fuertes, la proyección correcta de la distribución de la materia y una medida precisa de la densidad de la energía oscura dependen en gran medida de una estimación precisa del retraso de tiempo de las curvas de luz desviadas. Sin embargo, los datos astronómicos son intrínsecamente complejos, entre sus rasgos destacan el ruido, el muestreo irregular, la alta distorsión y las grandes brechas observacionales, entre otros. Muchos problemas sistemáticos, aleatorios y desconocidos afectan a la calidad de los datos, como los defectos en la fotometría, los problemas de los telescopios, la variación de la masa de aire y el paso de las nubes, entre otros fenómenos atmosféricos que pueden socavar estas señales. Además, la cantidad de datos astronómicos disponibles supera la capacidad de las herramientas analíticas tradicionales. De este modo, proponemos introducir una metodología de ciencia de datos para manejar el gran volumen de datos. Para probar la precisión de los modelos utilizados en esta investigación, se utiliza un conjunto de datos artificiales del Time Delay Challenge 1 (TDC1), este conjunto de datos intenta reproducir el comportamiento de curvas de luz reales con un retraso previamente conocido. A pesar de las muchas herramientas de estimación de retraso de tiempo se han desarrollado para abordar estas cuestiones, es imperativo aumentar la precisión. Por lo tanto, hacemos hincapié en la etapa de preprocesamiento para transformar los datos y disminuir su complejidad. Las pruebas empíricas demuestran que con la transformación de los datos del conjunto de datos simulado de TDC1, se produce una notable mejora en la precisión de la estimación del retraso de tiempo.

Contents

Abstract	i
1 Introduction	1
1.0.1 General overview	1
1.0.2 Research Question	3
1.0.3 Hypothesis	4
1.0.4 Objectives	4
1.0.5 Thesis contribution	4
1.0.6 Thesis structure	4
2 Background of Astronomy and Astrophysics	6
2.1 Background of Astronomy	6
2.1.1 Forms of gravitational lensing	9
2.1.2 Motivation	9
2.2 Justification	10
2.3 Literature review	11
3 Data complexity from strong gravitational lensing systems	13
3.1 Data complexity	13
3.1.1 Description of the datasets	14
3.1.2 Problem Statement and Dataset	14
3.1.3 Data exploration	17
4 Current methods for time delay estimation	23
4.1 Pattern matching methods	23
4.1.1 Combining time delay methods	28
4.1.2 Performance metrics	29
5 Time delay estimation model including preprocessing techniques	31
5.1 Data Science methodology	31
5.1.1 The proposed data science model	32
5.1.2 Collecting data	33
5.1.3 Cleaning the data	33

5.1.4	Data preprocessing	33
6	Experimental results	41
6.1	Visualizing and interpreting the results from TDC1	41
7	Conclusions and future work	52
7.1	Concluding remarks	52
7.2	Future work	53
	References	54
A	Appendix	65
A.1	Time delay techniques code	65
A.1.1	Cross-correlation code	65
A.1.2	Dispersion Spectra code	66
A.2	Data transformation techniques code	73
A.2.1	Wavelet Denoising code	73
A.2.2	Data differencing	75
A.2.3	Simple Net Return code	76
A.2.4	Christiano fitzgerald filter	77

Contenido

Resumen	i
1 Introducción	1
1.0.1 Visión general	1
1.0.2 Pregunta de investigación	3
1.0.3 Hipótesis	4
1.0.4 Objetivos	4
1.0.5 Contribución de la tesis	4
1.0.6 Estructura de la tesis	4
2 Antecedentes de la Astronomía y la Astrofísica	6
2.1 Antecedentes de la astronomía	6
2.1.1 Formas de lente gravitatoria	9
2.1.2 Motivación	9
2.2 Justificación	10
2.3 Reseña bibliográfica	11
3 Complejidad de los datos provenientes de sistemas de lentes gravitacionales fuerte	13
3.1 Complejidad de los datos	13
3.1.1 Descripción de los conjuntos de datos	14
3.1.2 Planteamiento del problema y conjunto de datos	14
3.1.3 Exploración de datos	17
4 Métodos actuales para la estimación del retraso de tiempo	23
4.1 Métodos de concordancia de patrones	23
4.1.1 Combinación de métodos de retraso de tiempo	28
4.1.2 Métricas de rendimiento	29
5 Modelo de estimación de retrasos de tiempo que incluye técnicas de pre-procesamiento	31
5.1 Metodología de la ciencia de datos	31
5.1.1 El modelo de ciencia de datos propuesto	32

5.1.2	Recolección de datos	33
5.1.3	Limpieza de datos	33
5.1.4	Preprocesamiento de datos	33
6	Resultados experimentales	41
6.1	Visualización e interpretación de los resultados del TDC1	41
7	Conclusiones y trabajo futuro	52
7.1	Observaciones finales	52
7.2	Trabajo futuro	53
8	Referencias	54
A	Apéndice	65
A.1	Código de las técnicas de retraso de tiempo	65
A.1.1	Código de correlación cruzada	65
A.1.2	Código de espectros de dispersión	66
A.2	Código de las técnicas de transformación de datos	73
A.2.1	Código de eliminación de ruido Wavelet	73
A.2.2	Diferenciación de datos	75
A.2.3	Código de retorno neto simple	76
A.2.4	Filtro Christiano Fitzgerald	77

List of Figures

1.1	Strong gravitational lensing effect.	2
2.1	Cosmic Microwave Background. The primal light of the universe. Source: NASA.	7
2.2	Hubble Diagram for Cepheids. Source: NASA & ESA.	8
3.1	Example plot: light Curve A and its error A and light Curve B and its error B	14
3.2	Rung subset example: time (first column), light Curve A (second column) Gaussian error of light curve A (third column), light Curve B (fourth column), Gaussian error of light curve B (fifth column)	15
3.3	Depicting the inherent complexity of the light curves of a double system pair rung 4 from the TDC1 synthetic data. With a quick glance at both light curves we can perceive some patterns like the nine seasonal gaps, as consequence we have ten data segments. However, finding patterns in a double y-axis plot through a visual inspection can result in a prone to error delay estimation or even in a unsuccessful task.	16
3.4	Disparity between two light curves from the same source.	19
3.5	Disparity between two light curves from the same source.	20
4.1	Exemplification of the drawback of the typical election method of maximum point of correlation. The possible advantages of using the average as a selec- tion technique for Cross Correlation methods.	27
5.1	Data Science Methodology for Time Delay Estimation.	32
5.2	Comparison of Wavelet denoising with level 2 and 3 applied to light curve A and B double system pair 434 from Rung 3.	35
5.3	Seeking homogeneity of light curves A and B through time series differencing.	37
5.4	Light curve A double system pair 21 from Rung 4: the first form is in raw data (blue color), second form transformed through the SNR algorithm (wine color), third form are differentiated data (red color).	38
5.5	Breaking in two components (CF-Trend and CF-Cycle) the time delay estimation issue by applying the Christiano-Fitzgerald filter to light curve A of double system pair 21 Rung 4 from TDC1 synthetic data.	40

5.6	Example of the Christiano-Fitzgerald band-pass filter applied to light curves from TDC1.	40
6.1	Performance charts of the average of all rungs 0-4, where lower is better except for the efficiency chart where higher is better : a) efficiency, b) goodness of fit, c) precision, d) accuracy.	45
6.2	Performance charts of the average of all rungs 0-4, where lower is better, e) MAD, f) MSE, g) RMSE, h) MAPE.	46

Chapter 1

Introduction

1.0.1 General overview

The Lambda Cold Dark Matter (Λ CDM) model is the current and most widely accepted model which explains the origin of the universe. The Λ CDM model is based on Einstein's general theory of relativity from the last century, which depicts the universe as a homogeneous, isotropic, and flat universe, Λ CDM is viewed as an evolution of the Big Bang hypothesis (Universe expansion). This theoretical structure is marked as a milestone for mankind as for the first time in human history a verifiable, consistent, and convincing theory was formed about the cosmos. The evolution and the current functioning of the universe as well as the phenomena which occur in it can be explained by the two main elements known as dark matter and dark energy. The Λ CDM model consists of six parameters, the Hubble constant H_0 being the key parameter in determining the rate of expansion of the Universe [88]. Even if the basic parameters are seemingly well measured, it is necessary and affordable to seek more accurate measurements to gain insight into the nature of these strange and mysterious elements [54].

There are various methods for deducing H_0 , such as the traditional three-step technique known as the distance ladder. However, it has certain disadvantages as the errors inherent in the incorrect calculation on the initial steps of the ladder could be accumulated in later stages [62]. A straightforward one-step method for measuring H_0 , thus, calculating the distance scale of the Universe, is the time delay estimate of a strong gravitational lens system [53]. Years prior to the discovery of the first gravitational lens system, Refsdal in 1964 demonstrated that gravitational lenses can be useful for calculating cosmological distances [95]. However, to use the delay technique, a strong lens system is required, this phenomenon consists of: a luminous variable source (e.g., quasar or supernova) that is redirected and magnified by a huge mass of matter like a galaxy or a collection of galaxies that acts as an optical lens, this increase in light intensity makes it possible to observe objects that are very far away or that, basically, do not have the light intensity to be visible without the lensing effect; as time passes, these images with lenses or multiple light streams increase or decrease their brightness; these changes are noticeable in all light curves, although with a delay of time (see Figure 1.1).

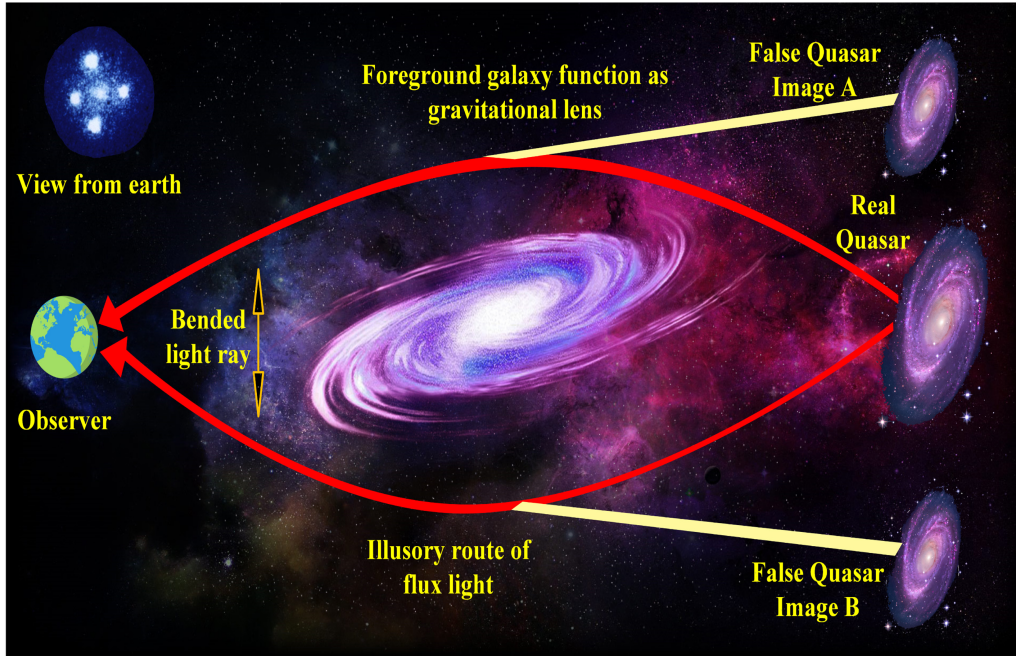


Figure 1.1: Strong gravitational lensing effect.

Over the past few years, there has been increasing interest in monitoring the strong gravitational lens effect, with the use of renewed technology. Modern monitoring campaigns above the sky, and in particular on Strong lens system, are reduced by decades to a few years or even months. Projects like the Large Synoptic Survey Telescope (LSST) and the Hubble Space Telescope (HST) in the near future may supply an enormous amount of information to the scientific community. As a result, projects such as this one can be enhanced and better realized because massive datasets on quasar samples will be available. Recognizing the abundant supply of real data, the purpose of this research is to explore the most commonly and robust methods currently used to determine strong gravitational lens time delay accurately.

The data collection process from gravitational lensing involves recording variations during an observational campaign in the form of light curves or time series (hereafter the terminology “light curves” and “time series” are used without distinction). This process has a number of complications, therefore multiple factors affect the behavior of light curves like lens galaxies intrinsic variability or anomalies caused by microlensing effect. In consequence, the recorded observations in the form of light curves have different levels of uncertainties due to the observational process. Therefore, the time delay estimation is a complex task derived from the features in data, e. g., unknown extrinsic variations, irregular sampling, noise, and seasonal gaps among other traits [58, 3]. Furthermore, uncertainty is inherent in measurements of time delay estimates between multiple images of a gravitationally lensed quasars system.

Therefore, the Time Delay Challenge (TDC) was launched in 2013 [34]. The objective was to generate synthetic data that simulates the complexity of the records obtained from

an observational campaign. The TDC is composed of two parts, i) the entry level Time Delay Challenge 0 (TDC0) with a reduced quantity of synthetic light curves intended for practical uses for the thirteen participating teams, and ii) as a filter with a deadline date of December 1, 2013. The launch date of the second level TDC1 was the deadline of TDC0, seven teams enter this level, and the deadline date was July 1 2014 [74]. The best results are compared with the proposed methodology at the visualizing and interpreting the results section (see §6.1). The TDC1 involves a dataset that contains 5,120 mock light curves to be blindly analyzed. The dataset is divided into five categories with a growing level of complexity in terms of noise level and observational gaps. This challenge was carried out to evaluate the capacities of the scientific community as well as the different approaches and methods used in the estimation of time delays, also, to produce information for the monitoring campaigns on the phenomenon of the strongly lensed quasar.

In this research, we take into account the intrinsic and extrinsic variability in the time series in conjunction with its characteristics generated by the strong gravitational lensing phenomena such as the trend and noise. It is also necessary to tackle the problem of managing a growing volume of astronomical data from a variety of sources. Hence, in this research we propose a different path to the conventional approach, as the latter may not be able to adapt to state-of-the-art technologies, processing large amounts of data, and the task of reliably converting data into meaningful information. The methodology proposed is a sequence of steps based on data science [45, 20, 103, 102]. In the proposed methodology, the preprocessing stage is the nucleus of this research, based on the hypothesis that by applying preprocessing techniques to raw time series it is possible to reduce complexity and find underlying patterns in data. Consequently, so uncovering veiled patterns in time series can lead us to find significant insight, and therefore, achieve an accurate time delay estimation.

Although many methods have been tested to address this issue, is essential to improve their accuracy. In this research, we test some of the most popular statistical models: Interpolated Cross-Correlation Function (ICCF) [47], Discrete Correlation Function (DCF) [39], Locally Normalized Discrete Correlation Function (LNDCF) [73], and Dispersion Spectra (DS) [86, 87]. Once we analyze, and estimated the time delays of the whole dataset from TDC1 with the above mentioned methods. We then compared our results to those previously provided by the Python Curve Shifting (PyCS) team, consisting of a group (Bonvin, Tewes, Courbin, and Meylan) from COSmological MONitoring of GRAvItational Lenses (COSMOGRAIL) members[108, 9]. In addition, another approach is proposed to select the most appropriate time delay estimation for cross-correlation methods. Finally, we find that the combination of delay estimations of different methods give us a better result, than the estimates obtained by a single method.

1.0.2 Research Question

How to improve the accuracy in estimating time delays with traditional statistical methods in the context of strong gravitational lenses?

1.0.3 Hypothesis

The use of data preprocessing techniques will lead to a better performance of state-of-the-art models for the time delay estimation problem under the context of the strong gravitational lensing.

1.0.4 Objectives

Main objective

To design and implement a data science model using state-of-the-art techniques to estimate accurately time delays of strong gravitational lensing systems.

Specific objectives

- To characterize and analyze with descriptive statistic the astronomical data provided by the Time Delay Challenge 1 (TDC1).
- To apply a data science methodology to the time delay estimation in order to improve the accuracy of statistical algorithms.
- To assess the feasibility of transforming the raw data with preprocessing techniques to improve the accuracy of the time delay estimation with statistical methods.

1.0.5 Thesis contribution

- A time delay estimation model based on data science with emphasis in data preprocessing techniques applied to TDC1.
- A comparison among TDC1 methods (D3CS, SPL and SDI) and state-of-the-art methods in astrophysics (DCF, LNDCF and DS).
- Analysis and estimation of the time delays from the complete TDC1 dataset.
- A study of the effectiveness of using preprocessing techniques to estimate time delays using metrics reported by state-of-the-art methods.

1.0.6 Thesis structure

The rest of the thesis is organized as follows: Chapter II provides the principles about the Universe, in order to understand the context of the time delay issue and as a consequence the extent of this research. Chapter III presents a general description of the synthetic astronomical data used in this work for testing delay estimation techniques. Chapter IV outlines some of the most widely used state-of-the-art methods and the pattern-matching methods that we used in this research, along with their different approaches for estimating

delays. Chapter V presents the proposed data science model and provides a general outline of the preprocessing stage. The experimental results are given in Chapter VI based on synthetic data. Final remarks are contained in Chapter VII.

Chapter 2

Background of Astronomy and Astrophysics

This chapter is intended to give an overview and introduce some key concepts. This chapter introduces the time delay problem and the impact on the study of the universe, including how the universe is composed and the dark matter.

2.1 Background of Astronomy

At the beginning of time, when the universe as we know it did not exist, approximately 13.8 billion years ago. All known matter and energy in the cosmos were fused into an extremely hot mixture that would fit in a receptacle the size of a pinhead. Then suddenly, in a burst of time, the genesis of the universe started in an event known as the Big Bang. Unlike the common thought this event was not an explosion, as there was no environment. In this way, the Big Bang was an expansion that generated an intense radiance and culminated in the phenomenon which causes matter, energy, and space to start co-existing [114]. The Cosmic Microwave Background (CMB) is the residual radiation generated by the Big Bang, the earliest remaining trace of the universe's birth. As a result, one of the most convincing facts to sustain the hypothesis of the Big Bang is the CMB (see Figure 2.1).

The study of the primitive Universe can be analyzed by looking at the sky, given that when astronomers gaze at the depths of space and delve into distant galaxies that are a billion light-years from Earth, they are taking a walk-through into ancient cosmic history. Therefore, these scientists are examining the looks of the primordial star system, since the light from remote stellar objects took a huge amount of time to reach the scope of their analyzing devices (telescopes, spectrographs, among others). With instruments like Galileo's telescope, Hubble Space Telescope (HST), and Planck's large telescope, astronomers have been able to capture the light emitted by the diverse remote cosmic objects since distant eras of the universe thousands of years ago. In this way, it is possible to analyze the properties of the CMB, an ancient trace from the beginning of the universe when it was much denser and hotter, as well as it is also possible to detect currently, how the universe is constantly

expanding [35].

In 1964, Robert Wilson and Arno Penzias of Bell Laboratories developed a radio receiver for scanning the intensity of the radiation emitted by the Milky Way. The received signals were weak, but with a repeat click uniformly, they initially assumed that this noise was static from the radio receiver. Nevertheless, these intriguing facts have turned out to be the remaining traces of the Big Bang, this emission is a bright burst of microwave radiation which fills all the space [113].

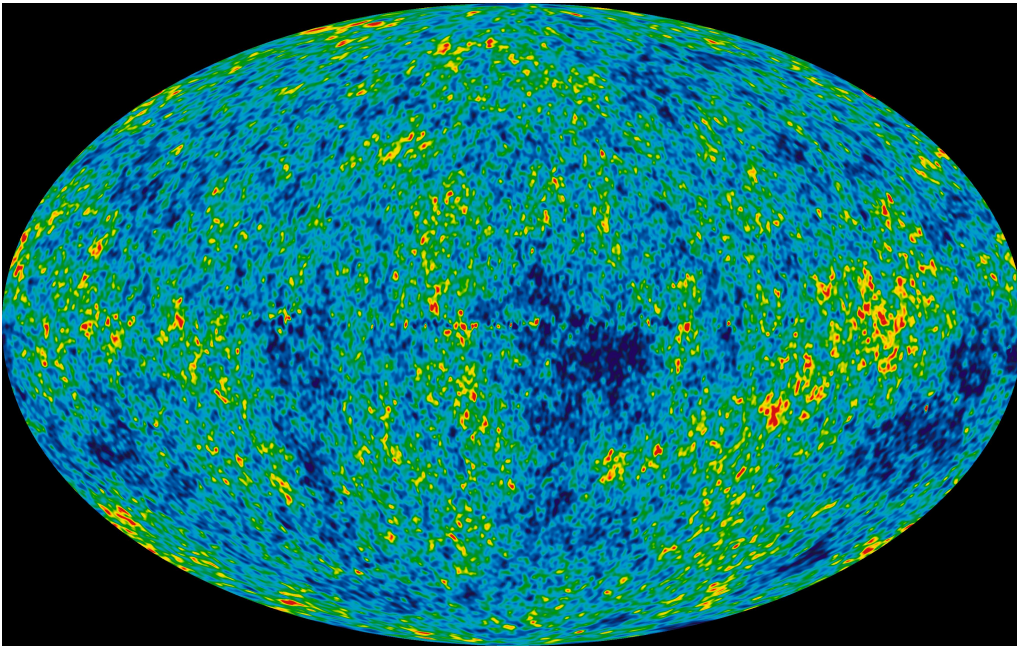


Figure 2.1: Cosmic Microwave Background. The primal light of the universe. Source: NASA.

+

In 1929, Edwin Powell Hubble made a remarkable discovery, the most noteworthy proof supporting the Big Bang theory. So far, the overwhelming majority of the scientific community claimed that the universe was a static and unchanging entity. However, Hubble probes the opposite, revealing that the universe is in expansion. In his research, Hubble considered the astronomer Vesto Slipher's measurements of the distances of the redshift galaxies and incorporated them into his own calculations. Leading to the Hubble Law, which deals with the redshift (Doppler Effect [29]) of distant galaxies, which implies that the further away one galaxy is from the other it appears that they move away between them faster [79]. In this way, galaxies distance themselves from each other, in the same form that an explosion projects fragments in all directions. However, it is not that the galaxies were cast into space, but the space between the galaxies themselves is continuously expanding, resulting in galaxies moving apart from each other. The Hubble diagram is a visual illustration of Hubble's law, this chart shows how the redshift of a galaxy increases linearly as a function of its distance from the earth. In this diagram a straight line is shown ($y = mx + b$), where

the slope (m) in the graphic measures the rate of expansion of the universe, meaning that this numerical calculus represents the Hubble constant H_0 . Figure 2.2 shows a similar plot of Edwin Hubble's original diagram, which leads to his discovery of the expansion of the universe.

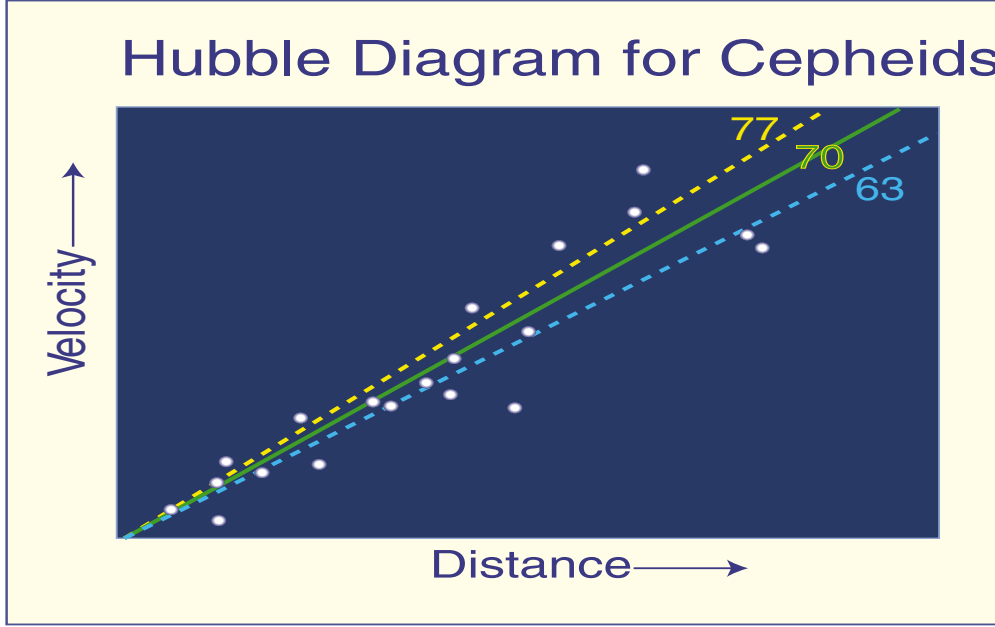


Figure 2.2: Hubble Diagram for Cepheids. Source: NASA & ESA.

The Hubble diagram presented in Figure 2.2 is a scheme of galaxy distance against the velocity at which the galaxy appears to be moving away from Earth. Cepheid variable stars are used for measuring distances, these stars have the feature that changes in their light intensity have a steady periodicity. In Figure 2.2 three lines are traced, the upper yellow dashed line shows a H_0 value of 77, the bottom blue dashed line indicates an H_0 value of 63, at last, the best-fitted green line is the continuous one, which estimates an H_0 value of 70 km/s/Mpc (kilometers/second/megaparsec).

Nowadays, the Λ CDM model, based on Einstein's theory of general relativity, can disclose a solution to reality regarding the evolution of the universe. In this way some uncertainties that haunted humanity through time might be responded. Among them it is possible to answer some questions such as: How was the universe formed? What is the universe made of? Is it the universe finite? Zwicky is considered the first to propose that galaxies can act as gravitational lenses and in this way he foresaw the importance of this topic in cosmology [125, 96, 8]. In 1964, through the estimation of the time delay among a multiple lensed quasar image, Refsdal demonstrates that the Hubble constant H_0 can be calculated [95]. The Hubble Constant is a number that describes the speed of expansion of the current universe.

2.1.1 Forms of gravitational lensing

Einstein was the first person to foresee the gravitational lensing effect. During a solar eclipse, astronomer Arthur Eddington proved Einstein's hypothesis with precision. Considering that Eddington observed that light from distant stars was curved as it passed near the Sun. In a solar eclipse, the moon blocks enough sunlight to make the stars near the sun visible. In this way, the locations of the now-visible stars seemed displaced. First, Eddington took a picture at night of a star cluster called the Hyades at Oxford. Afterwards, on May 29, 1919, Eddington photographed the Hyades which were almost immediately behind the sun during the total eclipse. Then he compared both measures and was able to prove the shift that Einstein predicts.

Gravitational lenses effect is perceivable when the light emitted from the distant source expands and increases. Moreover, it produces a notable increase in the observed brightness and given these circumstances, the location and comprehension of the stellar object analyzed is possible. Gravitational lenses are classified into three classes: a) micro-lens; b) weak-lens; c) strong-lens. The latter, the strong lens, is the ground where the present research lies. Since the light of the source variates, it is observed that the images vary with the delays between them. This effect is caused by the different routes through which the light is directed, as well as by the gravitational potential that it crosses. Based on the positional visualization of the images, the increases and the delays of time between the various images, it is possible to calculate the structure of mass density of the stellar object, as well as to measure the distance between the source and the observer. In this way, an accurate estimation of the delay period would provide the benefit of a correct calculation of the variable parameters (density of mass and age, among others) of the stellar objects, and thus, be able to classify them appropriately.

2.1.2 Motivation

At the dawn of astronomy as a science, ancient astronomers used to travel to the highest point of a mountain to gaze the different objects contained in the celestial sphere for long periods, to later formulate hypotheses about the behavior of those stellar objects analyzed by the night. Nowadays astronomers in a laboratory carry out the study of the sky in many cases with modern instruments such as computers and they access large volumes of data from around the world through the massive forms of telecommunications. In the past decade, Astronomical information has entered into the big data era, as example, with the Sloan Digital Sky Survey (SDSS) produced 100 terabytes during 2012, and in recent years obtaining half a petabyte.

The most notable and amplest future project is the Vera C. Rubin Observatory, formerly known as the Large Synoptic Survey Telescope (LSST) considered by the National Academy of Sciences of the United States as a telescopic instrument of high preponderance, with a launch date by October 2023. It is expected to obtain an amount of data of between 5 to 15 TB each night, as a consequence, it is feasible to project that in ten nights it is possible to obtain an amount of information comparable to that obtained by the SDSS during the time

it has been in operation.

As a consequence of the rapidly increasing volume of data from multiple sky surveys, the size of data repositories has grown from gigabytes to terabytes and petabytes. However, one of the most important questions that afflict the scientific community that are involved in the different research projects, is whether it is extracting all the knowledge contained in the immense volumes of information that are being obtained. Therefore, some traditional tools cannot handle the enormous flow of data. To manage the profuse opportunities and demands that exponential increases in the volumes and complexity of astronomical data, the scientific community would have to use novel techniques of research in data science, machine learning, big data, among other disciplines to be able to advance in the understanding of the Universe, due to the dynamism in the development of astronomical research, Astronomy is moving from being founded in hypothesis to being data-driven with an emphasis on data augmentation [124].

Due to the increase of data availability in the context of gravitational lensing, and their necessity to be processed in a short amount of time, new problems have risen. Therefore, we proposed a data science model, with the purpose to uncover underlying delayed patterns and with an analytical approach to the data, which seeks at the same time to obtain an accurate time delay estimation and automate the handling of a large volume of data. In this research, it was determined to emphasize the transformation of the data so that in this way the process of extracting knowledge from the data becomes a successful task (see 5.1). Also, there are some issues that have to be taken into account in order to estimate the delay in the light curves in a precise way. Given that the time series are: irregularly sampled, seasonal gaps, and noisy, among other issues. Therefore we propose to emphasize the preprocessing stage of the data science model proposed. The preprocessing step includes a transformation sub-stage where different techniques are applied, like the decomposition, noise removal of the time series. So in this way we can discover processes that are apparently hidden, in other words, find out if the transformation techniques can help uncover underlying patterns due to the intrinsic and extrinsic issues in data.

2.2 Justification

The cosmic size, age, and current rate of expansion can be inferred using the *Hubble – Lemaître* constant (H_0) [120, 97, 51] which is a key cosmological parameter. Several groups of researchers have explored different techniques to determine H_0 . Observations from Planck Satellite over the Cosmic Microwave Background (CMB) reported a value of $H_0 = 67.4 \pm 0.5 \text{ km/s/Mpc}$ [4]. In contrast, measurements based on the cosmic distance ladder have obtained a value of $H_0 = 73.0 \pm 1.4 \text{ km/s/Mpc}$ [98]. However, the results obtained by these groups and other independent measurements (for a further review see [115]), exhibit remarkable discrepancies among the early-Universe and late-Universe estimations [33], resulting in the so-called Hubble tension [115]. Considering the various and controversial outcomes produced for the value of H_0 , and for clarification of disagreements, it is essential to explore complementary and independent methods based on different physics [120, 51].

2.3 Literature review

In order to analyze the TDC1 dataset, we can follow two approaches. On one hand, the conventional one which involves light curves analysis and on the other hand the data science approach involving data transformation techniques. The conventional approach on time series analysis [36] indicates that it can be broken down into four components: the cyclic (a long-term pattern that oscillates in undulatory around the trend), seasonal (repeatable patterns in a range equal to one year or less), the trend, and finally, the random error or noise component resulting from the separation of seasonal and trend components in a time series. Noise is the effect of unknown factors. The estimated parameters for a light curve model could be significantly biased if the data includes a high degree of noise. A time series is stationary if its properties such as the mean and variance remain constant over time. To put it another way, a stationary time series is a process where its parameters do not depend on the time at which the time series is observed. On the other hand, a non-stationary series will have a mean or a variance that varies with time or both may vary [52]. The trend component is the behavior that indicates the direction in which a time series is moving in the long term. The most common trends are linear, logarithmic and exponential. Therefore, time series that have trends or seasonality or both components are non-stationary [56, 77, 14]. Many statistical techniques, analytical tools and quantitative methods depend on data that comes from a stationary process to produce a reliable result. Data from a stationary time series requires fluctuating over time around a fixed level, which means the average value of the time series must be constant without regard to time. Many statistical methods assume that parameters in a time series (e.g., the mean and variance) are constant over time, such as the power spectral density (PSD) technique which innately presupposes that data comes from a stationary process [107].

In another case, by introducing a non-stationary time series in a linear regression model, it is possible to obtain a regression that is apparently valid under typical statistical measurement, i.e., high R^2 values (goodness of fit), a very low value for the Durbin–Watson (DW) statistic test (regression residuals) among other measures. However, it produces a regression that has no validity and therefore no value, and this regression is called spurious regression [48, 14]. One of the most popular univariate methods for the analysis of astronomical time series [55, 44, 43, 100], signal processing, stock-market prediction, and in general time series modeling, is the Autoregressive Integrated Moving Average (ARIMA)(p, d, q) model [13]. An expansion of the Autoregressive Moving Average (ARMA)(p, q) model. This extension emerges, because ARMA was unable to process most of the time series produced in the real world because they are non-stationary time series. In ARIMA, the autoregressive term $AR(p)$ indicates that the explanatory variables for the prediction model are the data from the same time series, but delayed over time. The moving average term $MA(q)$ estimates the forecast model as a function of past errors or disturbances as explanatory variables. Finally, the integrated term $I(d)$ refers to the order of integration which, in other words, indicates how many differences are applied to the time series to transform it into a stationary process (see §5.1.4). Kim et al. [66] proposed a technique to remove the noise in a set of

simulated light curves of variable stars. In this research, the light curves quality is reported to be affected by noise and has to be removed, so that its variability can be adequately detected. Potential sources of noise within light curves may be photometric imperfections, cloud transit, variations in air mass, AGN flares and microlensing; among others.

In TDC1 some participants transformed the data. For example, Kumar et al. [70] and Rathna Kumar, S. et al. [94] used the difference smoothing technique, where the data is transformed by smoothing the extrinsic variability. So that in this way the imbalances caused by the noise that affects the light curves are corrected, which can mainly be due to the different phenomena such as the effect of microlensing by stars of the lens galaxy, or by the residual light of the lens galaxy. The results for difference-smoothing are divided into three datasets [70, 94]. The first two sets were submitted to TDC0 and TDC1 respectively. After TDC1, they produced a new calibrated set of results (corrected time delay) for Rung 3 and Rung 4 from TDC1 (third dataset). The results from Kumar et al. [70] and Rathna Kumar, S. et al. [94] for Rung 4 are in Table 6.2 and for Rung 3 are in Table 6.3. The publicly available results by Kumar et al. [70] and Rathna Kumar, S. et al. [94] record an amount of 1076 out of 5120 pair of light curves for time delay estimations from TDC1. For this reason, we believe that comparing their results would lead to biased inferences. Also, it is available a further review by Rathna Kumar [93] for difference-smoothing and the corrected results for TDC1. Also, Shafieloo et al. [101] and Aghamousa & Shafieloo [3] employed a smoothing algorithm in both light curves and a cross-correlation method to estimate time delay on TDC1. In the cross-correlation algorithm, to estimate the time delay, one light curve is maintained static, and the other light curve is shifted bidirectionally until the maximum correlation point is found. The time delay results from Shafieloo et al. [101] and Aghamousa & Shafieloo [3] with Smoothing and Cross-correlation are shown at; Rung 4 in Table 6.2 and Rung 3 in Table 6.3.

Chapter 3

Data complexity from strong gravitational lensing systems

The goal of this chapter is to present and characterize the data to be analyzed. Starting from the assumption that the behavior of stellar phenomena does not have a linear behavior (strong gravitational lenses). Its dynamism tends to be a complex system where many factors intervene and interact with each other, in addition to being closely interconnected by these elements, these influence and significantly affect the evolution of this phenomenon. The information collected by the astronomical event is recorded in the form of time series. The lack of information or missing data, the large time intervals without information, the intrinsically and extrinsically variants and the noisy data, are characteristic of the light curves coming from astronomical phenomena. This leads us to the conclusion that the information forthcoming from a strong lens system has a high degree of complexity, and that many of the standard tools of statistical analysis are not adequate to fulfill a reliably analysis of this stellar phenomenon. Also, in this chapter we will describe the dataset from the Time Delay Challenge 1 (TDC1). This dataset contains synthetic data simulating real quasars, where the true time delay is known (ground truth). This dataset is the core of this research. In addition, we present a step of the data science model proposed in this research, the data exploration stage that analyzes and digs into the data.

3.1 Data complexity

In this section, firstly we present a brief description of the data that is inspected. We present some of the most notable traits of the astronomical data used. Another topic to take into consideration is the way in which time is measured astronomically. The astronomical community usually measures time with the Julian date. It is founded on counting the days elapsed since January 1 4713 B. C. to 12:00 hours. In this way, the Julian date is a continuous form of recording the days and his fractions from the same beginning point. However, since the analyzed data is synthetic, it simulates sampling in Julian days, for that reason the initial sampling dates are always under the first ten days.

3.1.1 Description of the datasets

As we stated before the astronomical data manifest a large sampling gaps, low signal-to-noise ratio (SNR), missing information and other features that increase their inherently complexity. In consequence the observations registered have varying levels of error because of the observational procedure. This complexity in the collection of information arises due to factors such as the availability of equipment, physical limitations, as well as the capacity of the telescopes used, but the main cause of the difficulties is the changing climatic conditions that affect the visibility of the object. In Figure 3.1 we displayed an example of how is displayed the time series that represents in a plot a light curve A and light curve B and his corresponding errors [22].

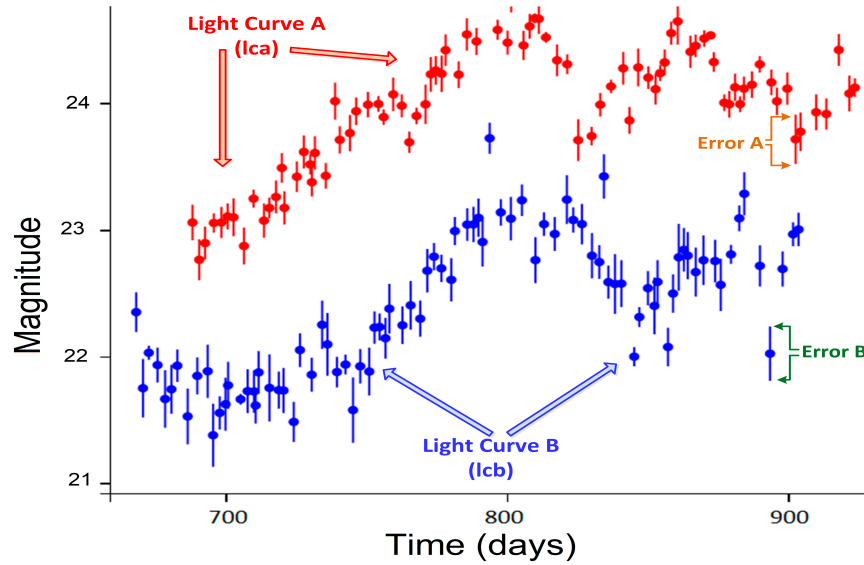


Figure 3.1: Example plot: light Curve A and its error A and light Curve B and its error B

Figure 3.2 is an example of how the information in the form of time series is published as well as is also introduced to our analysis. The first column in Figure 3.2 is the tracking time in days, the second column represents the magnitude of light curve A and its corresponding error for light curve A is displayed in the third column. The unit commonly used to register the magnitude is the nanomaggie, that is a linear unit of flux density, in other words is the star brightness measure magnitude. The fourth column shows light curve B and its associated error in the fifth column is shown.

3.1.2 Problem Statement and Dataset

The datasets under study come from the Time Delay Challenge 1 (TDC1), which were generated by a collaborative project called COSmological MONitoring of GRAvItational Lenses (COSMOGRAIL) - made up of members from multiple countries specialized in a wide range


```
## Time Delay Challenge light curves
##
## [time]=days, [lc]=[err]=flux in nanomaggies
##
##      time      lc_A      err_A      lc_B      err_B
##-----
```

time	lc_A	err_A	lc_B	err_B
1.87000	1.02659	0.03866	2.50732	0.07133
5.99000	0.91101	0.05243	2.52054	0.02973
7.32000	1.04086	0.06210	2.54686	0.06845
9.83000	1.07427	0.03612	2.44915	0.04517
....				
1691.06000	0.78862	0.05319	2.11478	0.03915
1692.50000	0.81895	0.05811	2.05563	0.05062
1696.31000	0.69337	0.05613	1.95906	0.02905
1698.65000	0.76884	0.07349	1.92158	0.04693

Figure 3.2: Rung subset example: time (first column), light Curve A (second column) Gaussian error of light curve A (third column), light Curve B (fourth column), Gaussian error of light curve B (fifth column)

of knowledge [34, 74]. The TDC1 tries to mimic the behavior of real strong gravitational lensed quasars. TDC1 consists of simulated data that was provided to participants, where the true time delays was unknown for them. The simulated data is grouped by hierarchical levels of difficulty (known as Rungs). Then, they should estimate the time delay between images of pairs of light curves. The organization of the challenge is composed of a set of scales, similar to a staircase where each group or Rung of the ladder contains simulated data for the participating teams to analyze. These data simulate a ladder of difficulty, ascending each Rung the difficulty is increased, the data intends to simulate datasets derived from light curves of strong lensed quasars. The data analyzed in this work comes from TDC1 light curve dataset [34]. There are five categories or Rungs available for the community, each dataset is labeled from the first to the last as Rung 0, Rung 1, Rung 2, Rung 3 and Rung 4; the complexity increases as the number increases. Rung 4 being the subset with more noise and gaps, so it is the one that most resembles a real strong lens system. In each TDC1 Rung, there are 720 double images and 152 quad images of synthetic systems. Also, the observational error per simulated light curve is given, simulating diverse observational sources of error including Gaussian error. Regarding the quantity of data contained in the TDC1 dataset, each Rung contains 1024 autonomous pairs of light curves with one delayed time series. The whole dataset contains a total of 5,120 files. Due to the inherent data complexity (e.g., noise, gaps, stochastic trends), we propose to preprocess the data so the hidden information in the data can be disclosed. Figure 3.3 shows the double pair 21 from Rung 4 in a time series plot, the light curve A is displayed in blue color, the light curve B is colored in red, the data is plotted with its respective error. Their known time delay (dt) is -17.45 days. The negative sign indicates that the leading light curve is B, that is why the time delay in days is presented in a negative form.

The various images emitted by a gravitational lensing system have their origin from a single light source. Therefore, it is possible to assume that multiple time series are visually and statistically similar, that is, that they are homogeneous among them, since they are

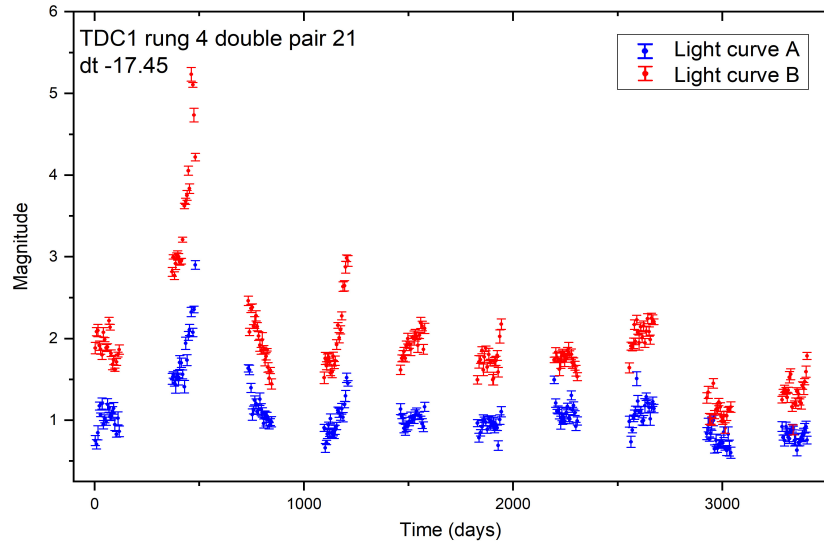


Figure 3.3: Depicting the inherent complexity of the light curves of a double system pair 21 rung 4 from the TDC1 synthetic data. With a quick glance at both light curves we can perceive some patterns like the nine seasonal gaps, as consequence we have ten data segments. However, finding patterns in a double y-axis plot through a visual inspection can result in a prone to error delay estimation or even in a unsuccessful task.

produced by the same data-generating process. However, the gravitational potential of the lens and other factors influence the photons path, affecting their behavior over time. Therefore, it is essential to consider the dynamic properties of the analyzed signals. Since the condition of stationarity or non-stationarity in the time series can play a key role in the process of choosing the parameters of an estimation model (see §4.1); for example, a model based on cross-correlation methods requires parameters such as mean and variance. Furthermore, the accuracy of the measurements of the time delay estimation may be affected if the condition of heterogeneity or homogeneity between time series from the same source is disregarded. The heterogeneity condition between two light curves from a single source occurs when the statistical properties (mean, variance, autocovariance) of one of them change over time and the other remains constant in time. Thus, by using preprocessing techniques, light curves can be homogenized (see Section 5.1.4). For heterogeneity, we also use the term disparity (see Figure 3.4), and for homogeneity, we refer to time series where the statistical properties such as mean and variance are constant over time (see Figure 5.3). Thus, by using preprocessing techniques, light curves can be homogenized. For instance, some of the different techniques for transforming data are used to stabilize the statistical properties of time series. As a result of applying the transformations to the time series, its statistical properties or indicators such as mean, variance, autocorrelation, etc. will be constant over time.

In order to obtain a deeper insight into the data, we used the Augmented Dickey-Fuller

Test (ADF) [31] to verify whether a time series is stationary or not. Therefore, the hypotheses of interest are H_0 : time series have a unit root and hence is non-stationary versus H_1 : time series is stationary. The analysis of the ADF test are presented in columns 8, 9 and 10 in Table 3.1. Table 3.1 shows some of the features of the data, detailed for each type of system (double or quad) and for each of the rung 0 to 4 of the TDC1 mock sample. Column one: list the system class double or quad; column two: indicate the sample rung number id; column three: minimum to maximum observations registered in each light curve in a lens system on their respective rung, the rung 0 and rung 1 have an average of 397.5 data points for each light curve roughly the double of rung 2, rung 3, and rung 4, theoretically, a higher amount of data might result in a more accurate estimation of time delays for the different methods applied; column four: this column displays the number of seasonal gaps in their respective system rungs, derived from this column it is possible to infer the number of data segments per system, as shown in Figure 3.3, where there are nine seasonal gaps and therefore the data is chunked in ten pieces; column five: this column indicates the minimum frequency in days with which the pairs or quad systems are sampled; column six: shows the maximum time interval in which the simulated systems are sampled or the data points are recorded in days; column seven: the average number of observations each simulated system contains recorded from both light curves A and B; eighth column: number of light curves A with the condition of stationarity or non-stationarity, for example, the first row for rung 0 with a total of 720 double-pair systems, we have 313 stationary and 407 non-stationary light curves A; ninth column; this column presents the same condition on the stationarity of the light curves in eighth column, but for light curves B; column ten: the coincidences of stationarity or non-stationarity in both light curves of a gravitational lensing system are shown, for example in the last row of Table 3.1 the system is quad and rung 4, there are 229 systems where both curves match whether they are both stationary or non-stationary and 75 light curves mismatch one of the two is stationary and the other is non-stationary; eleven column: last column or column located in the extreme right: the total of observations for each system of its corresponding rung is shown and in the lower right part the sum of the total of observations contained in the TDC1 dataset is shown. It is possible that at first glance the total quantity in thousands of mock lens systems of the TDC1 dataset is not impressive, however, the purpose of the last column located to the right of the Table 3.1 and especially the summation of all the data of TDC1, is to show that the quantity of observations that might be analyzed is of considerable dimensions.

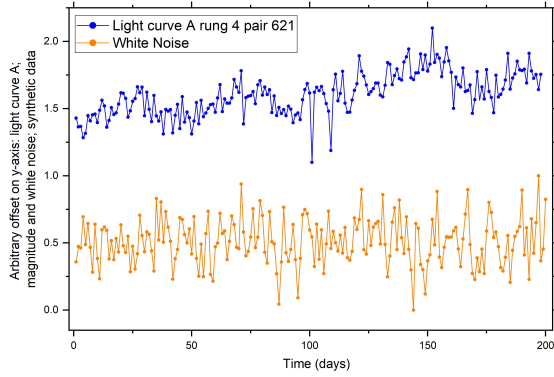
3.1.3 Data exploration

The data exploration stage is divided into two parts. 1) The *exploratory visualization*, there are many visual inspection tools, like the frequency histogram, correlogram, among others, but one of the most revealing and simple tool is plotting a time series. With this tool, it is possible to have a general description of some of the significant features the series contains or not, such as the trend, the cyclical component, and the seasonality. In addition, we can perceive if there are missing values (could arise due to a possible error in the insertion of

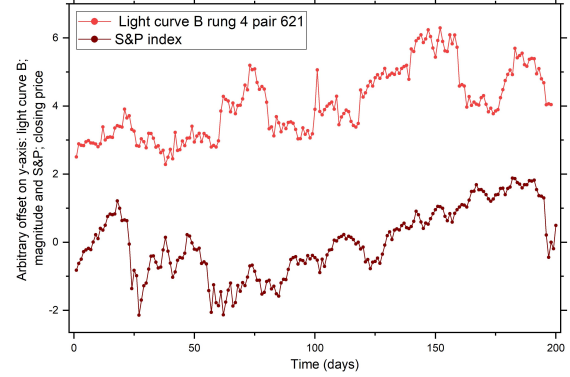
Table 3.1: Brief analysis of characteristics of TDC1 synthetic data. First column: type of simulated system double or quad; second column: rung identifier number; third column: the minimum and maximum number of observations per light curve registered for each system in the rung; fourth column: quantity of seasonal gaps in each system; fifth column: minimum sampling time interval in days; sixth column: minimum and maximum time interval of the shortest and longest sampling gaps between two observations in days; seventh column: average of observations (both light curves A and B) registered for each system in the rung; eighth: quantity of light curves A with the condition of stationarity or non-stationarity (720 for double systems and 304 for quad systems); ninth: quantity of light curves B with the condition of stationarity or non-stationarity (720 for double system and 304 for quad); tenth: amount of light curves A and light curves B that match or mismatch the condition of stationarity or non-stationarity in both light curves; eleventh column: the total summation of observations of each rung.

System	Rung	Min-Max observations in light curve	Seasonal gaps per system	Minimum sampling interval	Maximum sampling interval	Avg.-qty of observations per system	light curve A stationary vs non-stationary	light curve B stationary vs non-stationary	lc_A & lc_B match or mismatch	Quantity observations in rung
Double	0	377-420	4	0-0.82	126.72-134.09	797	313 vs 407	302 vs 418	543 vs 177	574,984
Quad	0	382-415	4	0-0.76	126.47-133.14	797	128 vs 176	155 vs 149	227 vs 77	243,248
Double	1	382-418	9	0-0.82	247.85-255.19	800	341 vs 379	371 vs 349	532 vs 188	575,168
Quad	1	372-414	9	0-0.82	248.26-254.23	786	157 vs 147	175 vs 129	226 vs 78	242,776
Double	2	199-199	4	3-3	249.0-249.0	398	223 vs 497	228 vs 497	501 vs 219	286,560
Quad	2	199-199	4	3-3	249.0-249.0	398	105 vs 199	127 vs 177	226 vs 78	120,992
Double	3	185-215	4	0-1.24	246.68-255.24	400	225 vs 495	214 vs 506	511 vs 209	287,730
Quad	3	188-216	4	0-0.99	247.11-253.14	404	103 vs 201	126 vs 178	215 vs 89	121,452
Double	4	195-212	9	1.70-4.24	250.97-261.17	404	349 vs 371	377 vs 343	524 vs 196	287,796
Quad	4	192-207	9	2.16-4.08	251.29-260.12	399	146 vs 158	177 vs 127	229 vs 75	121,460

$\Sigma = 2,862,166$



(a) Light curve A from Rung 4 pair 621 (offset 1.1) and White Gaussian Noise (offset 0.0).



(b) Light curve B from Rung 4 pair 621 (offset 4.05) and S&P stock market index (offset 0.0).

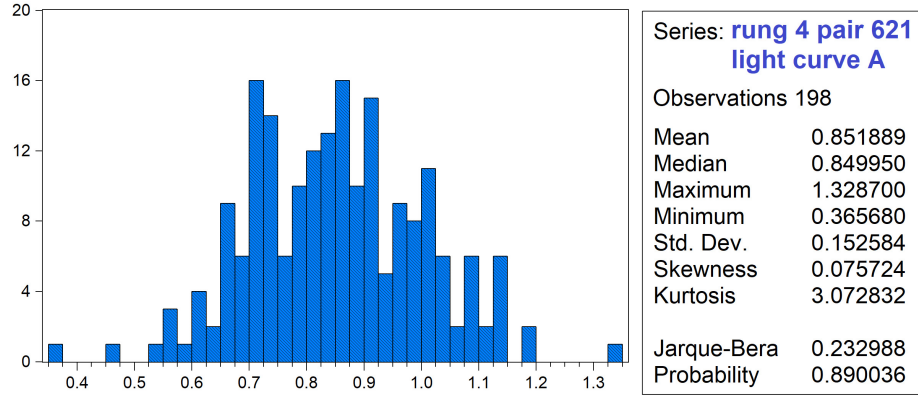
Figure 3.4: Disparity between two light curves from the same source.

data), data out of range among other issues. 2) The *exploratory statistics* part corresponds to descriptive statistics, which shows the main parameters of the time series, such as mean, median, maximum and minimum value, standard deviation, etc.

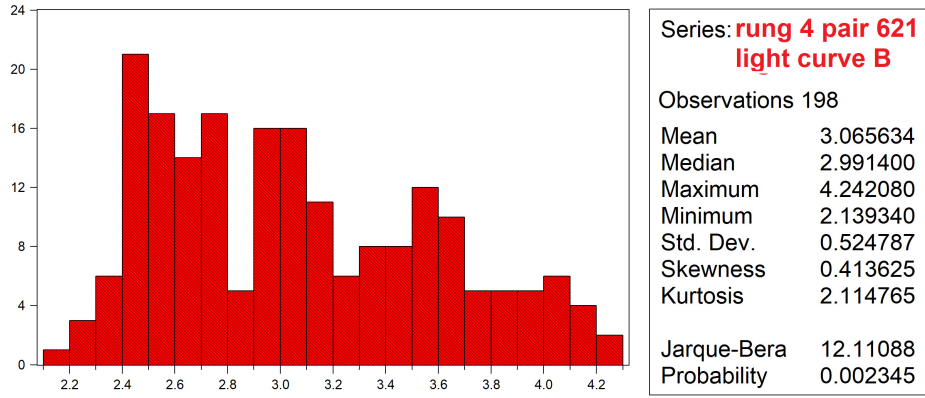
Visual Exploration

With both parts, visual and statistic, it is possible to determine one of the fundamental concepts in time series analysis, the stationarity. The graphic inspection can be useful for the detection of a stationary time series, however, in many cases, it is not enough. This is why other more robust statistical tests emerge, such as the tests proposed by Dickey and Fuller [30], the Phillips and Perron test [89], among others. Which are auxiliary to determine whether a time series is a stationary process or not. This topic becomes relevant when we refer to the formulation of cross-correlation methods (DCF, LNDCF, and ICCF) that rely heavily on the parameters of the time series (mean and variance). Figure 3.4 illustrates the complexity of the light curves produced by the strongly lensed quasar system in 2D line plots, the plots show the disparity between two light curves from the same source. For visual purposes, the light curves are displayed with points and a continuous line simulating a daily step-by-step sampling and not as the observations are recorded in a traditional way with an irregular sampling. In Figure 3.4a two time series are plotted, in orange the light curve A of pair 621 of Rung 4 and in blue color a synthetic white noise time series. Gaussian white noise is a sequence of uncorrelated random values with zero mean and unit variance [78]. Through visual inspection, it is possible to see obvious similarities between the time series illustrated in the Figure 3.4a.

Also, in the Figure 3.4b two time series are illustrated in the same plot, the time series shown in red (at the top) is the light curve B pair 621 of Rung 4 and in wine color (at the bottom) the movement of the stock market index Standard & Poor's 500 (S&P 500). S&P 500 is an indicator weighted by market capitalization (weighted participation average so the



(a) Frequency distribution of light curve A from rung 4 pair 621.



(b) Frequency distribution light curve B from rung 4 pair 621.

Figure 3.5: Disparity between two light curves from the same source.

most valuable and largest companies influence the movement of the index to a greater degree) whose function is to expose the behavior of a group made up of 500 companies listed on the stock market from the U.S., the period of observation is from 01/02/2018 to 10/16/2018. In the same way as in Figure 3.4a, the time series shown in Figure 3.4b shows remarkable resemblance between the light curve B pair 621 of Rung 4 and the S&P 500. Both time series are non-stationary and show a stochastic trend, where a stochastic trend is a random shift in a time series over time, there is not a clear pattern for periods of growth or decline.

The rung pair 621 light curve A in the descriptive statistic section shows in Figure 3.5a has a skewness of 0.0757 a value near zero, that shows a skew distribution symmetric around his mean and a kurtosis value of 3.073, indicating a mesokurtic distribution, pointing out that the value of a normal distribution is 3 or near this number. On the other side, the descriptive statistic from Figure 3.5b displays information from the rung pair 621 light curve B, it has a skewness of 0.413 that implies more positive values and the tail is slightly on the right side of the distribution, also a kurtosis of 2.11, this values means that this sample has a platykurtic negative kurtosis.

Exploratory statistics

To gain a deeper insight into analysis of the four time series presented in Figure 3.4, we used two statistical metrics. First, the unit root test mentioned above is used to determine if a time series is stationary, whereas if the test is accepted, the null hypothesis indicates that the time series has a unit root and thus is non-stationary. The second statistical tool is the Hurst coefficient or exponent (H) that is a metric based on a calculation of a numerical value. The Hurst exponent H is a tool used to measure the independence of a time series, which means that this instrument is useful for verifying whether the time series has memory (the current status of a period affects the future status of subsequent periods). The Hurst exponent H is a statistical method initially proposed by the British scientist Harold Edwin Hurst (1880-1978). After some time, Hurst's approaches and ideas were taken up by mathematician Benoit B. Mandelbrot (1924–2010) and generalized later, and called Rescaled Range Analysis (R/S) method. To calculate the Hurst exponent is used the R/S method as in [78] and [82]. Using R/S , the Hurst exponent is calculated, which is a scalar that determines the autocorrelation in a time series. Through the resulting scalar the corresponding propensity of a time series is assessed to either strongly reverse to the mean (mean-regression), to group in a particular direction (trending), or being categorized as a random process [68, 92]. The range in which the Hurst exponent is bounded is between zero and one. As mentioned above a time series is classified into three categories based on the estimated value of the Hurst coefficient.

- If the value of $H = 0.5$ or very close to this number, the series is random, there is an autocorrelation with a value of 0, that is, the data are independent, a decrease or increase in the series is not related. A geometric random walk.
- An exponent in the range $0 < H < 0.5$ indicates anti-persistent behavior. Where a period of growth is followed by another period of decrease, in the same way a period of decrease is followed by a period of growth, the autocorrelation of the data has a negative value. Negative and positive variations in the periods, a short term memory.
- When the exponent is located between $0.5 < H < 1$, the time series is persistent, the data shows positive autocorrelation. There is a high probability that increases in data will be followed by increases, similarly decreasing fluctuations are followed by decreases. A totally deterministic behavior in the data is given when the exponent has obtained a value of $H = 1$, long term memory.

On many occasions, visual inspection does not detect all features and properties in a time series. As a result, Table 3.2 below is briefly described, where two statistical tools are presented to analyze the four time series presented in Figure 3.4. The header and named rows reference the same Table 3.2. The results of these two tests show us that the two light curves of pair 621 Rung 4 are dissimilar between them. Despite the fact that both light curves come from the same source. The light curve A is quite similar to a white Gaussian noise process, both visually and statistically. According to the Hurst test, both time series are classified as anti-persistent with a value near zero. In the same way, the unit root test

Table 3.2: Hurst exponent and unit root test for light curves A and B from double pair 621 Rung 4, white noise and S&P index.

Time series	lc_A Rung 4 pair 621	white noise	lc_B Rung 4 pair 621	S&P 500 index
Hurst exponent	0.126	0.01	0.488	0.496
Hurst Categories	anti-persistent	anti-persistent	random walk	random walk
Unit root test p-val	0.01	0.00	0.11	0.24
Type of process	stationarity	stationarity	non-stationarity	non-stationarity

indicates that the two time series are stationary, because the results indicate that the null hypothesis should be rejected as they have a p-value below 0.05 as the level of significance. In contrast, the Rung 4 pair 621 light curve B and the S&P 500 time series are similar. In the Hurst test, both get an H value of almost 0.5, which classifies them as a random walk process. In the unit root test, both time series have a p-value higher than 0.05, indicating acceptance of the null hypothesis that both are non-stationary.

Considering both the visual inspection tool (see Figure 3.4) as well as the statistical tests like the Hurst coefficient and the unit root test (see Table 3.2), it is possible to infer that both light curves A and B that mimic the behavior of real light curves have a high degree of heterogeneity between them, despite the fact that both signals come from the same source. In addition, the visual inspection shows a stochastic and slightly positive trend of the light curve B (Figure 3.4b), which is confirmed by the unit root test (Table 3.2) which leads us to infer that the parameters (mean and variance) of the time series are not constant over time. For this reason, we believe there is a need to include a preprocessing step to homogenize time series (see §5.1.4) and facilitate the extraction of hidden patterns. Also, it should be noted that the data exploration stage (graph and statistics) has the function of trying to find relationships, connections, and patterns in the data. Moreover, tools at the graphic sub-stage should be meaningful and not just look good.

Chapter 4

Current methods for time delay estimation

Numerous techniques have been used to tackle these issues. These techniques take into account some of the different aspects that characterize the data coming from a lensed quasar, among these traits that distinguished these time series, are the unevenly sampled data, missing gaps between the data, intrinsic noise, scanty sampling, disturbed data caused by the effect in microlensing, therefore, to conclude a successful analysis these effects must be pondered. This chapter will present state-of-the-art methods for time delay estimation. This methods are classified as following: I) Methods from Cosmograil (SDI, SPL and D3cs), II) The cross-correlation based methods: DCF, LNDCF, and ICCF, III) Dispersion Spectra DS_1^2 .

4.1 Pattern matching methods

We can assume that to obtain an accurate estimation we may require complex methods to solve complexity in data. However, in some cases, simple models have yielded better results than more complicated models [106]. Hence, in this research, we focus on the analysis and in particular the preprocessing and the ever-increasing handling of data. We test some of the most widely used traditional tools, on one side the minimum mutual dispersion, and on the other the maximum cross-correlation, as a basis to select the delay estimation. These methods were compared with curve-shifting techniques that are based on different fundamentals. In this section, the first three time delay estimation methods used and proposed by the COSMOGRAIL team in the TDC1 are described: D3CS, SPL and SDI. Later, the four estimation techniques we used in this research are briefly reviewed: DS, ICCF, DCF and LNDCF.

D3CS : D3 Curve Shifting

The time delay estimation tool D3CS is founded on the basis of human visual inspection of the light curves. The implementation of this method was performed on the interface of a web browser using the JavaScript library D3.js by Bostock et al. [11]. This tool is available to the general public. With this public interface the user is able to move in real time and interactively the pair of light curves that are associated, allowing the user to alter the time, the magnitude, as well as the flux and even expand the segment of the plane required by the user. This tool was implemented for two reasons, the first was to obtain a preliminary estimate of the delay and its related uncertainty. The second is which level of confidence can be obtained with the estimate of the delay with this visual inspection and that the delay estimates are not catastrophic outliers [74, 9].

SPL: Free-Knot Spline Fit

A cubic spline is piecewise polynomial function which is joined by knots, where only third-degree splines are used. The approximation of functions through splines can be significantly enhanced if the knots arrangement and polynomial coefficients are free parameters to be optimized. The SPL method optimizes the knot positions based on the algorithm Bounded Optimal Knots (BOK) proposed by Molinari et al. [81]. With a previously defined fixed number of knots, the least square spline approximations are wrapped and computed using the BOK algorithm. BOK is responsible for selecting the iterative bounded optimal location of knots. The optimization function of SPL relies upon the wrappers from the scipy [83] implementation of FITPACK [32]. The strategy for estimating time delays of TDC1 with SPL as in Bonvin et al. [9] is briefly outlined in the following paragraph. To fit the SPL, it was iterated 20 times, beginning with mixed initial conditions, in order to improve the result of the time delay. For every set of two light curves, a bounded-optimal-knot spline $s(t)$ is matched or adjusted to the first selected light curve. After that, the remaining second light curve $l(t)$ is then displaced or shifted by a quantity δt in time and δm in magnitude. As a result, for a given observing time t (data point), the value of the displaced light curve can be composed by $l(t - \delta t)_{\delta m}$ [9]. The time delay estimation begins with the use of the D3CS tool, the SPL method in parallel adjust a singular spline that portrays the intrinsic quasar variability, at the same time, another smoother extrinsic spline is fitted, this second spline depicts the differential microlensing variability correspondent to the light curves. This shift mechanism to adjust the light curves over time is an iterated n -times process [74, 9]. The SPL method is further explained in Tewes et al. [108] and Bonvin et al. [9].

SDI: spline difference

In the same way that the SPL method used as the starting point the D3CS tool to estimate the time delay, the SDI technique mimics this strategy by utilizing the light curves produced by the SPL technique to outline the time delay intervals in which SDI enhance or optimizes its cost function [74]. The technique consists of adjusting a different spline for the first and

second light curves, to later try to decrease the variability of the difference between the two splines by changing them in time with respect to each other.

Dispersion Spectra

The Dispersion Spectra (DS) method is characterized by the simplicity and speed in its estimation as well as by the low demand of computational resources, without neglecting that it is considered a robust technique [85, 86, 87]. Some of the causes that make it reliable, is that the interpolation is not used as well as there is no need to smooth and neither remove the trend in data. Therefore, the computed error for each data point is autonomous.

Given a pair of light curves, represented by the time series $\vec{x}_A$ and $\vec{x}_B$, a single composite light curve $\vec{y}$ is built. The time series are combined with trial time delays $\Delta_t = [\Delta_{min}, \Delta_{max}]$ and an invariant value of the offset/ratio M among $\vec{x}_A$ and $\vec{x}_B$. Hence, for each pair of time series, the result is another time series $\vec{y}$ with $2n$ data points. The parameter M is obtained through the computed difference between the mean of image $\vec{x}_A$ and the mean of image $\vec{x}_B$, in the case of this research the value of M is very small due to data normalization. There are different versions of the dispersion method, so we use this version [85]:

$$D_1^2(\Delta_t) = \min_M \frac{\sum_{a=1}^{2n-1} w_a (y(t_{a+1}) - y(t_a))^2}{2 \sum_{a=1}^{2n-1} w_a} \quad (4.1)$$

where

$$w_a = \frac{1}{\sigma^2(t_{a+1}) + \sigma^2(t_a)} \quad (4.2)$$

Subsequently, the first of the light curves is used as the base and the second curve is displaced with respect to it by a time delay interval. The mean dispersion is calculated between the displaced points and their nearest temporal neighbors in the base light curve. The time delay considered as the best is that which minimized a statistical quantity or dispersion function associated with the combined light curve [40].

Cross Correlation Methods

Cross-correlation is a metric that explores the movements of two or more time series against each other, as a function of a shift in time between both time series. Especially at what point the highest degree of correlation between both time series occurs. The Interpolated Cross-Correlation Function (ICCF) [47, 46, 69, 38] is based on the structure of linear interpolation to produce the missing data in the light curve. This cross-correlation technique is performed in two phases, in the first the light curve A is interpolated and correlated with the light curve B, the second phase is the contrary, the light curve B is interpolated and correlated with the light curve A, to calculate the final result the two partial results are averaged. Where $\bar{s}$, $\bar{p}$ are the means of each time series, σ_s, σ_p are standard deviations of the corresponding light curve, N is the number of data in time series s , L_p is a piecewise linear interpolation of the time series p at $t + \tau$.

$$ICCF(\tau) = \frac{1}{N} \sum \frac{(s_i(t) - \bar{s}[L_p[p(t + \tau)] - \bar{p}])}{\sqrt{\sigma_s \sigma_p}} \quad (4.3)$$

The Discrete Correlation Function (DCF) method was planned particularly to estimate the time delays of gravitational lenses, it was proposed by [39] and [38]. Where, contrary to the technique designed by [91], interpolation does not play a key role. In this way, Edelson and Krolik place emphasis on optimizing the binning of data points. First, time differences (lags), $\delta t_{ij} = |t_j - t_i|$, between all pairs of observations are binned into discrete bins. Given a bin size $\delta\tau$, the bin centred at lag τ is the time interval $[\tau - \delta\tau/2, \tau + \delta\tau/2]$. $P(\tau)$ is the number of observational pairs in the bin centred at τ . The DCF at lag τ is given by

$$DCF(\tau) = \frac{1}{P(\tau)} \sum_{i,j}^{t_i, t_j \in I_\tau} \frac{(x_A(t_i) - \bar{a})(x_B(t_j) - \bar{b})}{\sqrt{(\sigma_a^2 - e_a^2(t_i))(\sigma_b^2 - e_b^2(t_j))}}, \quad (4.4)$$

where $\bar{a}$ and $\bar{b}$ are means of the observed data fluxes $x_A(t_i)$ and $x_B(t_j)$, and their respective variances are σ_a^2 and σ_b^2 . The observational errors associated to the data $\vec{x}_A$ and $\vec{x}_B$ are $\vec{e}_a$ and $\vec{e}_b$, respectively.

The Locally Normalized Discrete Correlation Function (LNDCF) introduced by [73], in the same pace as the DCF, also evade the use of the interpolation in the gaps between missing data. However, Lehar et al. introduces some changes on the method due to the fact that he perceives that the data from the quasar variations are not statistically stationary over the observed time scales.

$$LNDCF(\tau) = \frac{1}{P(\tau)} \sum_{i,j}^{t_i, t_j \in I_\tau} \frac{(x_A(t_i) - \bar{a}(\tau))(x_B(t_j) - \bar{b}(\tau))}{\sqrt{(\sigma_a^2(\tau) - e_a^2(t_i))(\sigma_b^2(\tau) - e_b^2(t_j))}} \quad (4.5)$$

where $\bar{a}(\tau)$, $\bar{b}(\tau)$, $\sigma_a^2(\tau)$ and $\sigma_b^2(\tau)$ are the lag means and variances in the bin centred at τ .

Mean DCF-LNDCF-ICCF

The cross-correlation of two-time series, both in simultaneous time, as well as out of phase in time, is a measure of the similarity that exists between them. The value of the delay with the highest correlation peak determines the best fit (greatest similarity) between the two-time series. However, the traditional way of selecting the best point of correlation may have certain weaknesses if we consider irregular sampling and seasonal gaps. Since the selection method is based on the assumption of regular sampling and stationarity in data. Contrary to the nature of the data and how the data is binned for the correct functioning of correlation methods (ICCF, DCF, and LNDCF). Consequently, it is proposed to average a set of time delay estimations given by the correlation methods (ICCF, DCF, and LNDCF). The average increases the accuracy and reduce the uncertainty in the time delay estimation. Particularly, in this research a significant number of estimates were obtained using correlation methods.

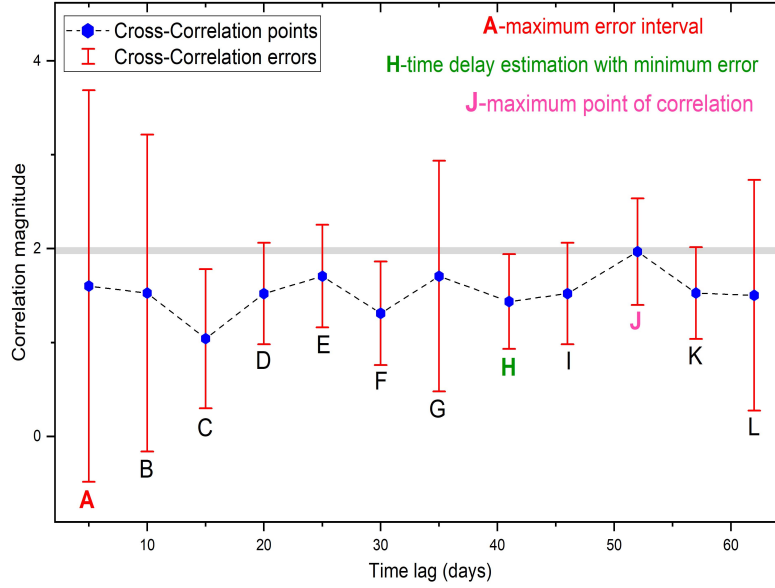


Figure 4.1: Exemplification of the drawback of the typical election method of maximum point of correlation. The possible advantages of using the average as a selection technique for Cross Correlation methods.

Compared with the DS method, correlation methods (ICCF, DCF, and LND CF) use different sizes of binned data and produce a larger number of time delay estimations.

Figure 4.1 is a graphical representation of an example produced by a time delay estimation illustrating the two rows of the Table 4.1 (*Cross-correlation point* and *Cross-correlation error*), where a set of twelve estimation points produced by a Cross-correlation method is presented. The *Cross-correlation point* parameter indicates the maximum point of correlation between the two light curves with different estimations. Labeled as *Cross-correlation error* the error produced by a Cross-correlation technique indicates the quantity or degree of uncertainty in the estimation. Also in Table 4.1 the row with the parameter *CCF/CCP* row is a compound row where the quotient of the division between *CCF/CCP* is shown in percentage terms, the results of this row show that the error is very large that even on some occasions the error has a value greater than the point of greatest correlation *CCP*. In Table 4.1 the rows labeled as: *Estimated DT*, *Bin Size*, *Range min-max*; the first shows the value of the estimated time delay in said correlation point, the second as its label indicates the length of the bin, the third row indicates the search range for the corresponding estimated correlation point, and finally the row labeled *MAD* shows the results of using the performance metric Mean Absolute Deviation (see 4.1.2), in this row in column with the heading *H* the point with the lowest error is shown, unlike the heading *J* (point of highest correlation) with a higher error that the estimation point mentioned above.

Table 4.1: Example of twelve results produced with a cross-correlation method (ICCF, DCF, and LNDCF). *Cross-Correlation Point (CCP)*: degree of correlation between two light curves; *Cross-Correlation Error (CCE)*: magnitude of error associated with its corresponding time delay estimate; *CCE/CCP*: value of the quotient in percentage terms; *Estimated TD*: value of the estimated time delay; *Bin size*: length of the bin; *Search range*: indicates the minimum and maximum search range for the corresponding estimated correlation point; *Mean Absolute Deviation (MAD)*: performance metric (see 4.1.2). Note: the traditional form for selecting the best estimation is the peak of maximum correlation, which is labeled *J* in pink, however this *J* point does not have the lowest error.

Correlation ID Parameters	A	B	C	D	E	F	G	H	I	J	K	L
<i>Cross-correlation point</i>	0.205	0.057	-0.908	0.036	0.438	-0.409	0.438	-0.134	0.036	0.909	0.057	-0.007
<i>Cross-correlation error</i>	0.735	0.592	0.259	0.189	0.193	0.192	0.431	0.175	0.189	0.199	0.172	0.431
<i>CCE/CCP</i>	130%	110%	71%	35%	32%	42%	72%	35%	35%	29%	32%	82%
<i>Estimated TD</i>	68.4	68.5	68.0	67.8	68.3	68.5	68.3	67.4	67.8	68.0	68.5	68.0
<i>Bin size</i>	2	2	4.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	2	1.5
<i>Search range (min-max)</i>	2-105	20-115	25-120	30-70	30-75	30-95	30-105	35-80	35-85	40-90	40-95	40-105
<i>MAD</i>	2.0	2.09	1.57	1.37	1.85	2.15	1.85	1.00	1.35	1.63	2.08	1.56

4.1.1 Combining time delay methods

The combination of estimates from different sources is not a recent topic, there are references that Laplace considered using the combination of estimates of the regression coefficients [104]. To estimate the time delay of a strong gravitational lens there are a large number of different approaches and ways to calculate them, in other words, for the same strong lens quasar system there may be several time delay estimations. However, there are several unanswered topics that are complicated to be resolved, such as: it is possible that the selected model has a better performance than the other methods with artificially generated data and that with real data it is not the most efficient, either due to an over-fit to the data, structural changes of the phenomenon, a bias or inconsistencies at the time of building the model, omissions or involuntary errors when choosing the parameters of the estimation model. In many cases this issues could be addressed with the results generated with a combination of different estimation methods. As a consequence of the above, in this research, we proposed to use the combination of time delay estimates. With a subset of data, several tests were performed in order to find the best combination of estimates. It was obtained as the best combination of time delay estimates two methods, the LNDCF with the transformation technique of differencing time series, and the DS with the decomposition method CF and their decompositions of trend and cycle, the result is a mix between DS with CF -Trend/Cycle and LNDCF-Mean with Differencing.

4.1.2 Performance metrics

This section summarizes different performance measures. Error measurements are a form for evaluating the performance of an estimate produced by a technique and comparing the capacity between the various methods. In the following list, the metrics for measuring the proficiencies of the different methods are displayed, and metrics one to four are proposed by COSMOGRAIL. In addition, four other performance measures are used. These latest four measurement techniques are widely used in the time series literature.

Where N is the total amount of light curves estimated for analysis in the rung.

1) Efficiency : $f \equiv \frac{N_{\text{available}}}{N}$

$\tilde{\Delta}t_i$ is the collection of estimated time delays.

Δt_i are the true time delays.

δ are the corresponding uncertainties.

2) Godness of fit : $\chi^2 = \frac{1}{fN} \sum_i \left(\frac{\tilde{\Delta}t_i - \Delta t_i}{\delta_i} \right)^2$

3) Precision : $P = \frac{1}{fN} \sum_i \frac{\delta_i}{\Delta t_i}$

4) Accuracy : $A = \frac{1}{fN} \sum_i \frac{\tilde{\Delta}t_i - \Delta t_i}{\delta_i}$

Mean absolute deviation (MAD) is the average of the absolute value, or the sum of absolute differences between the value of the true delay and the estimated delays divided by the number of observations. Where n is the sample size, E_i the i^{th} estimated delay value, T_i the i^{th} true delay.

5) $MAD = \sum_{i=1}^n |T_i - E_i|$

Mean squared error (MSE) is the summation of the squared errors divided by the number of observations, or the average of the square of the difference between true delay and the estimated delay values. Where n is the sample size, E_i the i^{th} estimated delay value, T_i the i^{th} true delay.

6) $MSE = \frac{\sum_{i=1}^n (T_i - E_i)^2}{n}$

Root mean square error (RMSE) is the square root of MSE, is a measure error that squares the deviations. Where n is the number of observations available in sample, T_i the true delay value of a point for a given time period i , E_i the estimated delay value of period i .

7) $RMSE = \sqrt{\frac{\sum_{i=1}^n (T_i - E_i)^2}{n}}$

Mean absolute percentage error (MAPE) is a relative error measure in terms of percentage absolute error, in this formula the average of absolute errors are divided by the i true delay observation value. Where n is the total number of data points in sample, T_i the true delay value in given period i , E_i the estimated delay value in period i .

$$8) \text{ MAPE} = \frac{\sum_{i=1}^n \left| \frac{T_i - E_i}{T_i} \right|}{n}$$

Chapter 5

Time delay estimation model including preprocessing techniques

This chapter presents the proposed methodology based on data science, but adapted for the time delay estimation of strong gravitational lensing.

5.1 Data Science methodology

The great quantity of astronomical data available over this decade has grown well beyond that available in the past century. The data features and the immense dimension exceeded the possibilities and capacities for manual analysis. As a result, new techniques were developed and various strategies were merged as well. Methodologies such as Data Science (DS) [103, 15], Knowledge Discovery in Databases (KDD) [27, 119, 71], and Data Mining (DM)[105, 123] have been developed to conduct a more profuse analysis in search of knowledge discovery in the data. Hence, we introduce a data science methodology adapted for the strong gravitational lensing phenomena that consists of successive steps, being the central element of this proposal the data preprocessing stage. In an effort to lessen the data complexity and be able to discover knowledge hidden within the data. The following list presents the stages of the proposed Data Science methodology for this research:

1. Problem statement (see Section 3.1.2).
2. Collecting data (see Section 5.1.2).
3. Cleaning the data (see Section 5.1.3).
4. Data exploration (see Section 3.1.3).
 - (a) Exploratory visualization.
 - (b) Exploratory statistics.

5. Data preprocessing (see Section 5.1.4).
 - (a) Data integration.
 - (b) Anomaly detection.
 - (c) Data transformation.
 - (d) Data normalization.
6. Pattern matching methods. (see Section 4.1).
7. Visualizing and interpreting the results (see Section 6.1).

5.1.1 The proposed data science model

In this section, we introduce the proposed methodology (see Figure 5.1) to analyze the TDC1 dataset. The proposed data science methodology is a guideline consisting of sequential steps in order to ease the development of the time delay estimation process in a successful way. Some of the steps of the proposed methodology are described in the next subsections since the other steps are included in other chapters of this research. Some of the methodological steps are not used in this research, such as the data integration step (see 5.1.4), as the TDC1 is a simulated dataset created and supplied by a single source. Although all of these methodological steps are linked, omissions or errors in one step can lead to misleading results or catastrophic errors.

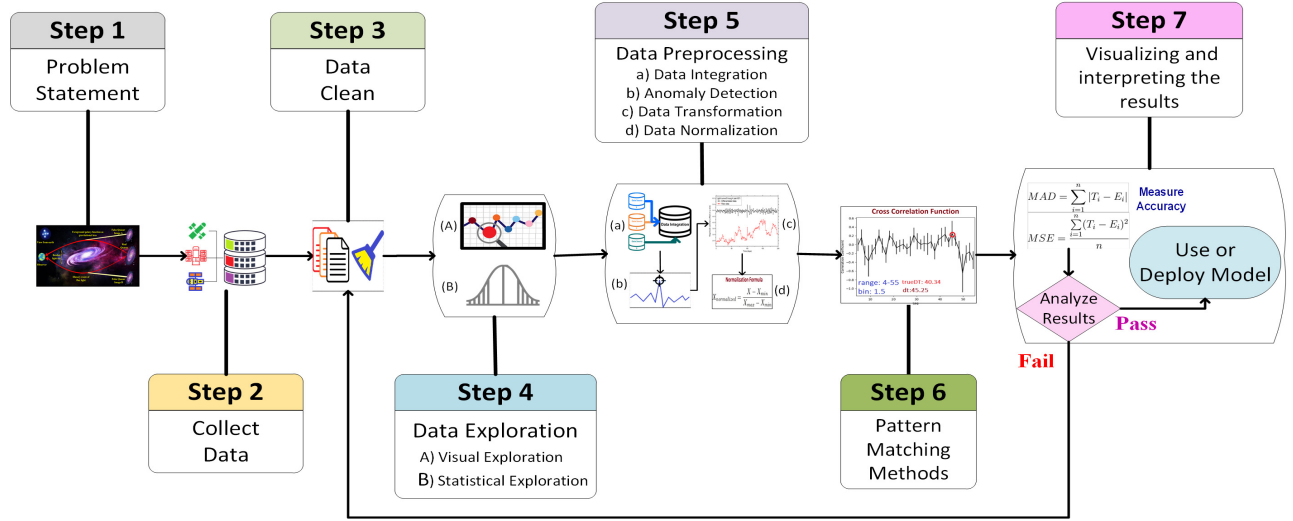


Figure 5.1: Data Science Methodology for Time Delay Estimation.

5.1.2 Collecting data

One of the most relevant issues in a data science project is to obtain a reliable dataset. Three sources are responsible for the generation and compilation of universal data: companies, government and academic [103], the latter being the data supplier for this search. The astronomical community has a philosophy of co-operation, so they promote open data culture, so the data sets might be available in a public domain and at no cost. The data collection process is composed of obtaining new observations (primary) and storing them, also querying datasets previously-stored by other parties (secondary), as well as combining data [6].

5.1.3 Cleaning the data

The data gathered from real-life phenomena are commonly not free from errors. In order to achieve a valid result based on a time delay estimation method, unprocessed data needs to be cleaned up and transformed to be reliable [26, 18]. At this stage, to improve the data quality all inconsistencies in the data need to be fixed, such as handling missing values, corrupted values, non-numeric values when numerical data are expected, and replicated entries, among other issues. The search and solution of defects in the data can be done manually by an expert or a group of experts, by crowdsourcing [80], as well as by a software solution or a combination of both.

5.1.4 Data preprocessing

This stage consists of four steps, starting with data integration (data comes from multiple sources), anomaly data detection (is the process that seek the identification of observations with unexpected behavior), data transformation (the data is transformed with the goal of converting them into more robust, stable and better adapted data for a particular algorithm) and data normalization (data is scaled in a common range).

Data integration

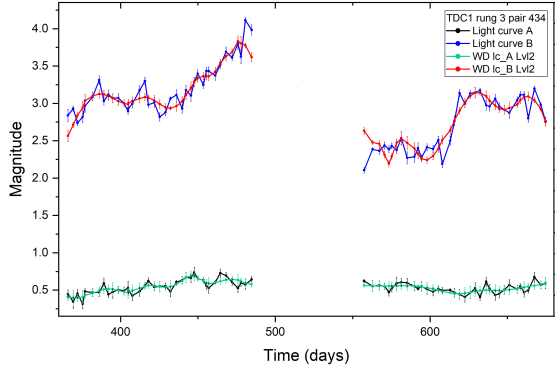
The purpose of data integration is to make available to the researcher dispersed datasets from a variety of independent data providers [37]. In the field of analysis and estimation of strong lensed system, the topic of merging data becomes relevant since, in order to obtain a single dataset with a greater number of observations and a more regular sampling, it is usual for there to be cooperation in a monitoring campaign over the same strong lensed system with different telescopes [70]. On the other hand, on occasions the data produced from independent monitoring campaigns on the same strong gravitational lensing system at different time periods are collected and integrated to obtain autonomous estimations, then these estimations are merged with various techniques to improve accuracy [5, 10].

Anomaly detection

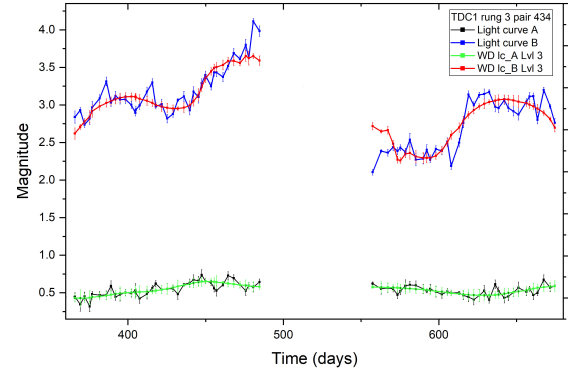
Also called outlier detection, an anomaly or outlier in a time series is an observation with a behaviour that deviates considerably from the rest of the observations [45]. The anomalous value of this observation can lead to speculation that it was generated through an external mechanism [26]. Anomaly detection aims to capture the data points that presents an extreme value, in time series analysis it relies on historical data to encounter unusualness in data. The main process causing anomalies in a monitoring campaign on a strong gravitational lensing system is the microlensing effect [1]. In a gravitational lensed quasar, intrinsic variations appear in two or more images given at different times of observation. However, in many cases the extrinsic variability (microlensing effect) produced by stellar mass objects in the lensing galaxy might corrupt a light curve [49]. Over a wide field of time scales variations in microlensing can take place. Moreover, extrinsic variability arising on significantly larger time scales than the intrinsic variability of the quasar is known as slow microlensing [9]. For this reason, to the extent possible and to obtain a high degree of accuracy in estimating the time between pairs of light curves, intrinsic and extrinsic variability must be decoupled to obtain filtered intrinsic variability. Consequently, this adverse effect has been addressed by various strategies and techniques [84, 41, 108, 2, 112, 9, 49, 10]. The precise isolation of intrinsic variability with respect to exogenous variability of the microlens is not an easy task because it is not possible to measure accurately the effect of the microlensing over the lightcurves [2]. In this research, we tackle the unwanted effect of gravitational microlensing, with data transformation and normalization. Furthermore, the methodological structure of some estimation techniques is able to reduce the effect of microlensing (e.g., the LNDCF technique uses a trimmed mean to perform its measurements and estimates, hence, this method is capable of dealing effectively with outliers and contaminated values [26]).

Data transformation

The time series used to estimate the time delay will always include some noise due to the behavior of the phenomenon itself as well as random exogenous influences. Conventionally, it is considered that noise may have some affectation on the extraction of patterns in time series. For this reason, there are techniques to remove noise, known as denoising techniques, because some methodological approaches for time delay estimation have sensibility to noise. The idea behind using a denoised time series is to help time delay estimation methods to find meaningful features by extracting the underlying patterns. Also, one expect that the attained knowledge is unaffected by the completely random component of the time series. In this way, it is proposed to use various transformation methods in order to improve the performance of the time delay estimation methods. We used some of the most popular and easy to implement transformation techniques, with the purpose of handling the noise and trend in time series. There are four different transformation techniques employed in a separate form.



(a) Light curve A and B double system pair 434 from Rung 3: light curve A and B first version with raw data in black color and blue color, second version smoothed through Wavelet denoising with a level 2 in green color for light curve A and red color for light curve B.



(b) Light curve A and B double system pair 434 from Rung 3: light curve A and B first version with raw data in black color and blue color, second version smoothed through Wavelet denoising with a level 3 in green color for light curve A and red color for light curve B.

Figure 5.2: Comparison of Wavelet denoising with level 2 and 3 applied to light curve A and B double system pair 434 from Rung 3.

Wavelet Denoising (WD) [90, 116, 122] The different smoothing methods are intended to remove noise from a time series, thus allowing patterns and tendencies to be identified in a more simplified way. A large number of methods have been proposed to smooth the time series, such as Holt-Winters smoothing [118, 60], exponential moving average smoothing, simple moving average smoothing [67], among others, the last being the simplest but at the same time the most used. However, this last method has the drawback that there is data loss. Consequently, the use of the WD technique is proposed. The wavelet transform is a mathematical method that consists of a set of orthogonal functions (wavelets). The time series is examined and decomposed into a collection of scaled and mutually orthogonal wavelets. These decompositions are produced through the translation and scaling of a function called the wavelet mother. The translation and scaling is controlled with coefficients. The wavelet transform condenses the main time series characteristics into a small number of large-magnitude wavelet coefficients. The noise is considered to be wavelet coefficients with small value. This noise can be removed or attenuated without affecting the time series quality. Once you establish the threshold of the wavelet coefficients, the inverse wavelet transform is applied to rebuild the time series.

We use the Python scikit-image library for noise removal based on the wavelet method [116]. One of the parameters of the wavelet denoise function is the election of the wavelet mother. The wavelet denoise function can support various mother wavelets, such as Harr, Daubechies, Symlets, Mexican hat, Shannon, Coiflets among others. We evaluated discrete wavelets and found that Coiflets showed the best performance with lower fitting error and then validated visually. The denoise function uses the PyWavelets software module to include one of the 14 different wavelet families for Python [72]. There are two techniques for wavelet coefficient threshold selection: BayesShrink and VisuShrink. Both are widely

Table 5.1: Order of differencing

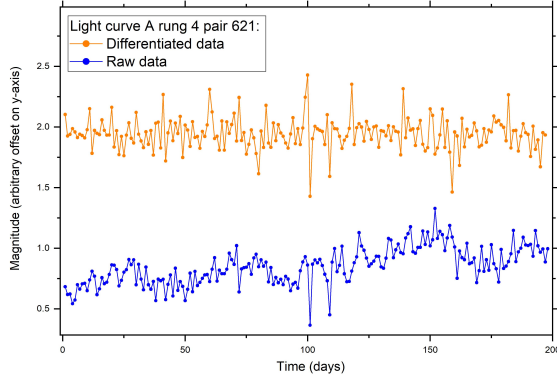
Integration order	linear equation
1st order	$Y' = Y_t - Y_{t-1}$
2nd order	$Y'' = Y_t - 2Y_{t-1} + Y_{t-2}$

used in computer vision tasks to remove noise in images, being BayesShrink the most used. Nevertheless, we found that for time series VisuShrink gives better results, because with BayesShrink we obtain a non uniformly smoothed representation of the time series, since the BayesShrink attempt to find the optimal threshold for each subband [42]. Two kinds of thresholding can be selected, soft and hard. In the hard threshold procedure, the coefficients that are set at zero are those with absolute values below the threshold. The first stage of soft thresholding consists of setting to zero those coefficients with an absolute value below the threshold. Next, the coefficients with an absolute value equal to or exceeding the threshold value are reduced to zero. The parameters were selected with a visual inspection of the time series, we used the soft coefficient threshold, that uses an individual universal threshold to every single wavelet coefficient and produces smoother results. Figure 5.2 shows the light curve A and B from the TDC1 Rung 3 pair 434 in a denoised version and in its original raw form. For visual reasons, the day sampling in Figure 5.2 has been modified, where 59 of 198 simulated observations are shown. In order to better appreciate the difference between the noisy light curves and the smoothed version of this time series, and the Gaussian error associated to the light curves. Figure 5.2 presents only two data segments, starting at day 365.96, for visual purposes, the seasonal gap is trimmed, the original starting point of the second data segment is 732.43 and is shifted for -117 days starting new point 615.43, the last modified day is 679.67 (original 797.51). The parameter `wavelet_levels` is set to 2 in Figure 5.2a and is set to 3 in Figure 5.2b. As the parameter `wavelet_levels` increase we obtain a more smoothed time series.

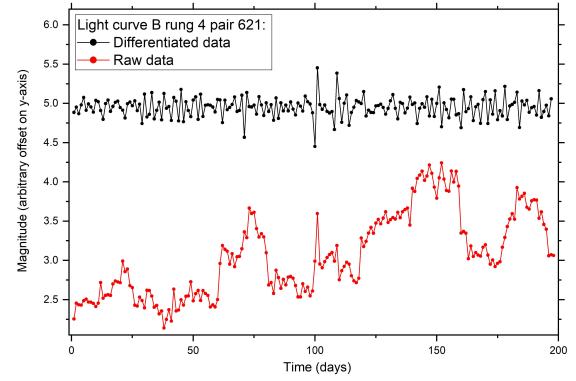
Differencing time series [12, 76, 111]. However, in many cases, a visual inspection does not clearly show that there is a deterministic trend in a time series. Rather, if there is a trend component present in a time series in most cases, this is a stochastic trend. The time series that are transformed through differencing are called integrated series.

Depending on the amount of differentiations that are used on the data to convert them into a stationary series, it will be granted the distinctive of integrated in order n (see Table 5.1), where n is the number of differentiations that were applied to the series, for example; if a difference is applied to the series, it will be an integrated series in order one and so on. Figure 5.3 depicts how the light curves are transformed when applying the differentiation on raw data (red color). It is possible to see how the differentiated time series (black color) have more stable parameters, since they fluctuate within a stable range of movement.

Simple Net Return (SNR) [16, 110] An investor usually buys a financial asset (saving or investing in a bank, mutual funds, stocks, etc.) and keeps possession of said asset for a



(a) Light curve A double system pair 621 from Rung 4: first version with raw data in blue color (offset 0.0), second version transformed through differencing in orange color (offset 1.43).



(b) Light curve B of double system pair 621 from Rung 4: first version with raw data in red color (offset 0.0), second version transformed through differencing in black color (offset 4.45).

Figure 5.3: Seeking homogeneity of light curves A and B through time series differencing.

Table 5.2: Example of SNR formula with five periods of observations from light curve A from Rung 4 pair 21, from day 2.77 to 28.08 day.

Date-period	2.77	8.92	15.17	23.05	28.08
Value-observations	1.88	2.08	2.10	1.91	1.87
SNR-formula	-	$\frac{2.08 - 1.88}{1.88}$	$\frac{2.10 - 2.08}{2.08}$	$\frac{1.91 - 2.10}{2.10}$	$\frac{1.87 - 1.91}{1.91}$
Results-SNR	-	0.11	0.01	-0.09	-0.02

certain period of time. At the end of that holding period, the price of the asset is expected to be higher than at the time of purchase. A successful investment is one that achieves the highest possible revenue with certain initial funds, which are measured in terms of asset returns. This technique is a measure that displays the percentage of increase or decrease of the initial asset value, a positive rate of return means that the position has generated profits, on the contrary, a negative rate of return implies losses. The percentage change of the observations for one period from date $t - 1$ to date t results in a Simple Net Return (SNR). The corresponding one-period simple net return is

$$SNR = \frac{O_t - O_{t-1}}{O_{t-1}} \quad (5.1)$$

where O_{t-1} = observations initial value and O_t = observations final value.

Table 5.2 shows an example of the SNR formula and the results it produces. There are two motivations behind the assumption that it is preferable to use SNR data rather than raw data as input for parametric methods (cross-correlation). First, return time series are more manageable because of their statistical properties. Second, in many cases, the use of this technique converts a non-stationary time series into a stationary process.

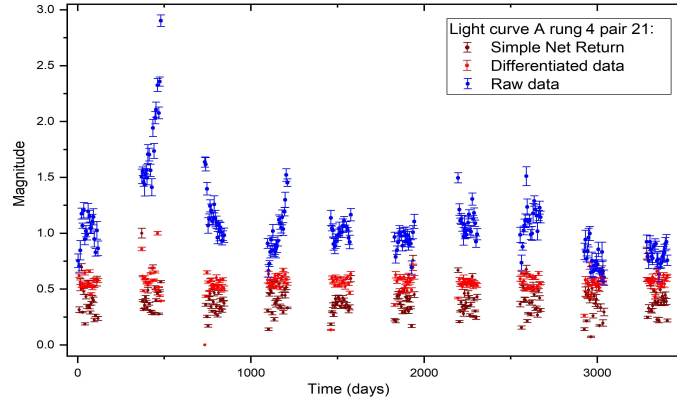


Figure 5.4: Light curve A double system pair 21 from Rung 4: the first form is in raw data (blue color), second form transformed through the SNR algorithm (wine color), third form are differentiated data (red color).

Figure 5.4 shows light curve A of the double pair 21 from Rung 4 (Raw Data), and two light curves produced from the transformation techniques: 1) Simple Net Return and 2) Differentiated Data. The differentiated data (red) and SNR data (wine) have a more stable movement in comparison to raw data (blue) that displays a non-stationary motion.

Christiano-Fitzgerald filter (CF) [17] Since CF method is based on moving averages, let us define moving average first. Moving average is obtained by calculating a sequence of averages from a collection of subsets of a time series, which is a type of mathematical convolution. When moving average used on time series, the noise is filtered, and we obtain a smooth modified version of the original time series. Moving averages are used in two ways: a) One-sided Moving Average (OMA) and b) Two-sided Moving Average (TMA). Where CF method is based on the latter. OMA are used to detect the trend direction, as a filter for random fluctuations in the short-term trend, as well as a prediction technique, among other applications. OMA integrates the past and present values of the observations to calculate each of the averages. For example, let y be a time series and t the time, with a span of 5 periods. The OMA is obtained as follows:

$$OMA = \frac{y_{t-4} + y_{t-3} + y_{t-2} + y_{t-1} + y_t}{5} \quad (5.2)$$

TMA is primarily used to identify the underlying trend, and this technique is also known as a centered filter [64]. The TMA takes into account past and future observations to compute the average at a particular point in time. In the TMA, the moving average is placed in the middle of the time range of the averaged values. For example, let y be a time series and t the time, with a span of 5 periods. The TMA is obtained as follows:

$$TMA = \frac{y_{t-2} + y_{t-1} + y_t + y_{t+1} + y_{t+2}}{5} \quad (5.3)$$

The CF method is a band pass filter that assumes that the time series y_t is a random walk (not stationary) and made up of three components. The components are: a) Stochastic trend T_t , b) Cyclical component c_t , and c) Irregular term ϵ_t . Each of these components is linked to different frequencies, so the analysis and resolution of the time series decomposition to isolate the trend of the smooth cycle are found in the frequency domain. The CF filter is calculated using a two-sided weighted moving average, the weights are changing and asymmetric, where the cycles in a "band" are delimited by a specified lower and upper limit, the cycles that are within the range are preserved, and the remaining cycles are filtered. The term weighted assigns more significance to the most recent observations over time, as they are likely to have more influence on the present values. The weighted moving average is calculated by multiplying a decreasing weight of each preceding observation in the time series and adding the resultant values. Contrary to the traditional practice where only the cyclical component is taken into account, in the present research the results of both decompositions (trend and cycle) were evaluated and averaged to produce a single time delay estimation. To filter the time series in the CF method, two parameters have to be predetermined: minimum period of oscillations P_L and maximum period of oscillations P_U . Since the CF filter was proposed for macroeconomic analysis, the P_L and P_U parameters may have different values depending on the economic data sampling frequencies. So, it is recommended to use the following settings (P_L and P_U): 18 and 96 for monthly data, 6 and 32 for quarterly data, and, 2 and 8 for annual data. To the best of our knowledge, CF method was never used in strong lensing. Hence, there were no referrals to the parameters implementation. As a result, we tested various settings and the one with the best overall performance for each Rung (pair or quad) has been selected. If there is a deterministic trend in the time series, then it has drift. The parameters setting $P_L = 8$ and $P_U = 32$ is because no drift is present, and applies for all double pairs and quad pairs from Rung 3 and Rung 4. The setting $P_L = 18$ and $P_U = 96$ applies for Rung 0, Rung 1, Rung 2 quad pairs. The variation in the selection of the quad parameters for Rung 0, Rung 1, Rung 2 is due to the fact that the results produced by DS method (see §4.1) of the time delay estimations were the same for both the cyclical component and trend component. Then it did not make sense to average the results between both components since there was no diversity in the time delay estimations. In Figure 5.5 is shown the light curve A from the double pair 434 from Rung in three different versions. The first one is the raw data colored in blue, also their two decompositions resulting from applying the Christiano-Fitzgerald filter. The CF-Trend component is colored on red and has a slightly shifted movement (hard to visualize) in time and also is smoothed, and at last the CF-Cyclic component in black color, the CF-Cycle time series has a more stable mean and variance than the other two light curves.

For a further analysis of the CF filter Figure 5.6a and Figure 5.6b give us a more in depth look of the decomposition obtained through this method. Trough a visual inspection is evident that the decomposition (specially CF trend) are a shifted version of the original raw data.

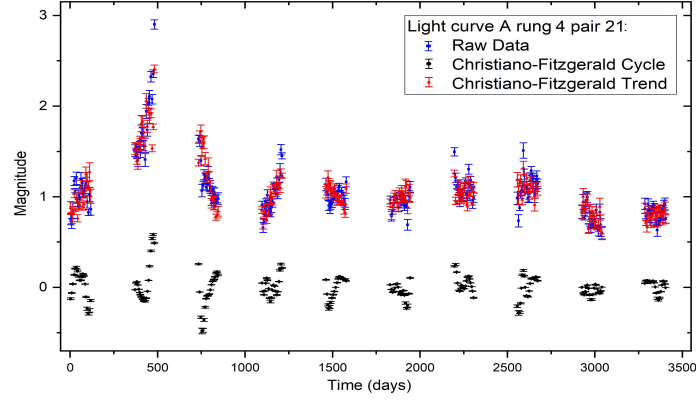
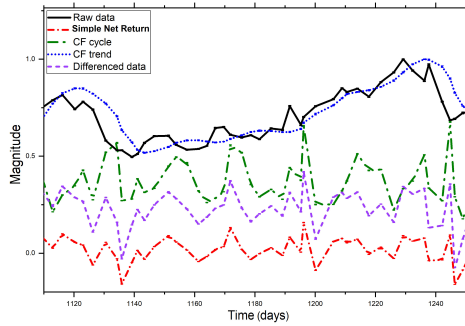
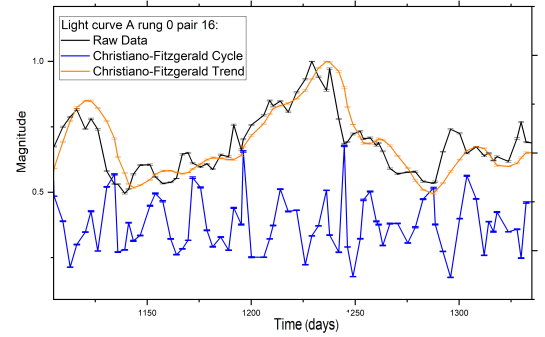


Figure 5.5: Breaking in two components (CF-Trend and CF-Cycle) the time delay estimation issue by applying the Christiano-Fitzgerald filter to light curve A of double system pair 21 Rung 4 from TDC1 synthetic data.



(a) Light curve B double pair 16 from rung 0: first form in raw data (black color), second form transformed by SNR algorithm (red color), third and fourth signals obtained by applying CF-filter; CF-Cycle (green) and CF-Trend (blue), fifth form is differenced data (purple color), only 44 data points are shown.



(b) Light curve A double pair 16 from rung 0: first version in raw data (black color), the Christiano-Fitzgerald filter decomposition obtained from light curve A of double pair 16 Rung 0 from TDC1, CF-Cycle (blue) and CF-Trend (orange), only 72 data points are portrayed.

Figure 5.6: Example of the Christiano-Fitzgerald band-pass filter applied to light curves from TDC1.

Chapter 6

Experimental results

In this section the results are summarized from different methods used in this research – DCF, LNDCF, ICCF, DS with raw data and with preprocessing techniques applied to raw data. This aforementioned methods are compared with the results of the methods from COSMOGRAIL team – D3CS, SDI and SPL – that participated in the Time Delay Challenge 1 (TDC1).

6.1 Visualizing and interpreting the results from TDC1

Before introducing each pair of light curves from Rung 0 to Rung 4 to the different estimation methods, they must have been transformed and/or formatted. Since 5,120 pairs of light curves can be considered a very large number of time delay estimates, handling this amount manually is impractical. Since the COSMOGRAIL stated that the expected period of time of observation determined will last from a few days to a few months and all known time delays of the artificial sample from TDC1 are contained within a range between 5 to 120 days.

A search for time delay estimates by range was executed, and different search ranges were performed. As for the methods used in this research, there is no single optimal overall search range where the best estimation of delays is found. Moreover, the correlation (ICCF, DCF, and LNDCF) and dispersion (DS) algorithms yield several correlation peaks and minimum dispersion values respectively. A significant number of search ranges have been used to carry out a broad scope of search ranges, so the average search range is reported; see last column in Table 6.1. Each search range represents solution zones, and within each range, the best estimate of delays can be found. With an a priori knowledge of the delay of each of the pairs of light curves, we were able to optimize the length of duration of the time delay estimation process. As a result, the strategy is focused and not an exhaustive search. Each search range is delimited by a minimum starting point (min) and an upper limit (max), the various search ranges are listed below:

- (a) Delay value between 5 to 20 days, we use 48 different search ranges e.g., min = 2 & max

= 15, min = 2 & max = 20 ... min = 15 & max = 95, min = 15 & max = 100.

- (b) Delay value between 20 to 40 days, we use 84 different search ranges e.g., min = 2 & max = 35, min = 2 & max = 40 ... min = 30 & max = 105, min = 30 & max = 110.
- (c) Delay value between 40 to 60 days, we use 99 different search ranges e.g., min = 2 & max = 35, min = 2 & max = 40 ... min = 45 & max = 120, min = 50 & max = 120.
- (d) Delay value between 60 to 80 days, we use 78 different search ranges e.g., min = 2 & max = 85, min = 2 & max = 90 ... min = 65 & max = 120, min = 70 & max = 120.
- (e) Delay value between 80 to 100 days, we use 41 different search ranges e.g., min = 2 & max = 90, min = 2 & max = 95 ... min = 85 & max = 120, min = 90 & max = 120.
- (f) Delay value between 100 to 120 days, we use 25 different search ranges e.g., min = 2 & max = 120, min = 2 & max = 125 ... min = 100 & max = 125, min = 105 & max = 125.

The various methods were evaluated automatically with a known delay file, filtering and rejecting the results with an error of more than 13 days, in order to achieve an Accuracy (A) below 0.2% ([74], which is a cosmological requirement; see [59]). The error measurements are a form to assess the performance of an estimation produced by a technique and compare the capability among different methods. In the next list, the metrics for measuring the proficiency of the different methods are displayed:

1. Efficiency (f) [50].
2. Goodness of fit (χ^2) [50].
3. Precision (P) [50].
4. Accuracy (A) [50].
5. Mean absolute deviation (MAD) [19, 76].
6. Mean squared error (MSE) [75, 99].
7. Root mean square error (RMSE) [117, 65].
8. Mean absolute percentage error (MAPE) [75, 109].

The metrics one to four are proposed by COSMOGRAIL. Four additional performance metrics are also used. These last four measurement techniques are widely applied in the time series literature, for a further review of the metrics see section 4.1.2. The Table 6.1 shows a brief summary of the results from the five Rungs from a selected group of methods. The first column (Methods) displays the methods of estimating the time delays, while the second column (Preprocessing) indicates the transformation technique applied to the time series

(except Raw Data), the third column (Rung) shows the Rung considered from TDC1. The fourth column (f) displays the fraction of light curves where the time delay was estimated using the corresponding technique, we can regard that each Rung contains 1024 mocked pairs of light curves. For example, in Rung 4 the SPL technique estimated the time delay of 657 and did not present the time delay calculation of 367 pairs of light curves, the result obtained in the metric efficiency marked in the fourth column f is $657/1024 = 0.642$ by SPL. On the other hand, the maximum value of this metric is equal to unity or 1.0, which means that all the curves of light of the corresponding Rung have been analyzed and estimated in their time delay. Like the estimations calculated by the DS methods and the combined results produced by DS&CF-Trend/Cycle and LNDCF-Mean&Differencing, so we can stand, that this metric is a fraction calculation. The next three metrics showed in columns fifth (χ^2), sixth (P) and seventh (A), are also proposed by the COSMOGRAIL team, where the uncertainty (δ) parameter plays a key role with these three metrics, at column 12th. The performance measures presented in columns 8th (MAD), 9th (MSE), 10th (RMSE), and 11th (MAPE) are commonly employed in the scientific literature. Nevertheless, these techniques do not take into consideration the uncertainty (δ) parameter. The 13th column shows the average search range in days. For example the ranges: 5-30, 30-65, 10-50 yields 33.33 (days) average search range. The different search ranges used by the PyCS team are not reported. For visual purposes, a summary of results is displayed in Table 6.1. To compare the performance of the different time delay estimation methods, it is important to clarify that lower is better; except the Efficiency metric (f), where the higher the better.

In Table 6.1, the results from D3Cs, SDI and SPL methods are obtained from [9] and from the Time Delay Challenge website¹. Three correlation methods were employed in this research: DCF, ICCF, and LNDCF, However, in Table 6.1 are only shown the results from LNDCF, because the LNDCF-Mean technique employing differencing time series was the one that showed the best performance. The DCF-mean method generated the best overall results with the data transformed by the differencing time series technique. Therefore, the ICCF and LNDCF methods used only this transformation technique. For the DS algorithm, the most enhanced results were generated with the decomposed data from the CF-Trend/Cycle technique. On the other hand, the method that stands out from the COSMOGRAIL team is the SPL. In addition, for comparison purposes, the results generated by LNDCF using raw data and the traditional selection form to choose the estimated time delay (maximum correlation point) are provided. Similarly, the results generated by the DS method are displayed using unprocessed data. The summary and average of the five Rungs from the COSMOGRAIL method the SPL and our best method (DS&CF-Trend/Cyc mixed with LNDCF-Mean&Differencing) proposed in this research are shown at the bottom of the Table 6.1, The method that has the best overall performance is the DS&CF-Trend/Cycle combined with LNDCF-Mean&Differencing, since in seven of eight different metrics this method has the best value, followed by the SPL method in one metric. Regarding the 13th column of Table 6.1 shows the average search length. In Table 6.1 the average search length indicates that the LNDCF method has an average search length of 47.38 days for the five

¹<https://www.epfl.ch/labs/lastro/scientific-activities/cosmograil/data/tdc1/>

Table 6.1: Summary of results from all Rungs. Lower is better (except column labeled f where higher is better). Preprocessing: 1) Christiano-Fitzgerald filter (CF-Trend/Cycle), combines two components: Trend-(CF-Trend) and Cycle-(CF-Cycle); 2) Differencing; 3) Raw Data (preprocessing technique was not applied). Time delay estimation methods from COSMOGRAIL: a) D3 Curve Shifting (D3CS); b) Free-Knot Spline Fit (SPL); c) Spline Difference (SDI). Time delay estimation methods employed in this research: I) Dispersion Spectra (DS_1^2); II) Locally Normalized Discrete Correlation Function (LNDCF); III) Mix of time delay estimation of methods: DS_1^2 using data produced from CF-Trend/Cycle and LNDCF using data produced from differencing time series. Selection techniques for Cross Correlation methods: A) Mean (proposed alternative, section 4.1); B) Max, typical selection procedure of estimation (maximum point of correlation).

Methods	Preprocessing	Rung	f	χ^2	P	A	MAD	MSE	$RMSE$	$MAPE$	uncertainty	Average search length
D3CS		0	0.819	0.46	0.32	0.11	2.76	31.00	5.57	11.42	6.95	
SDI		0	0.778	1.00	0.11	0.07	1.52	10.27	3.21	6.82	2.48	
SPL		0	0.807	1.01	0.09	0.04	1.21	15.15	3.89	4.34	1.88	
DS_1^2	CF-Trend/Cycle	0	1.0	1.38	0.32	0.09	1.84	6.09	2.47	8.60	7.29	25.07
DS_1^2	Raw Data	0	1.0	4.46	0.32	0.10	2.21	9.49	3.08	10.06	6.97	29.30
LNDCF-Mean	Differencing	0	0.979	0.57	0.27	0.07	1.48	5.40	2.32	6.63	8.60	52.04
LNDCF-Max	Raw Data	0	0.997	0.99	0.32	0.12	2.48	14.22	3.77	11.87	10.04	57.04
MIX-DS_1^2 with CF-Trend/Cycle and LNDCF-Mean with Differencing		0	1.0	0.17	0.30	0.059	1.30	3.60	1.90	5.89	7.95	38.55
D3CS		1	0.655	0.38	0.31	0.12	2.72	26.27	5.13	11.84	6.75	
SDI		1	0.618	1.66	0.12	0.07	1.51	11.07	3.33	7.55	3.21	
SPL		1	0.651	0.86	0.08	0.05	1.25	7.56	2.75	4.55	1.82	
DS_1^2	CF-Trend/Cycle	1	1.0	1.56	0.37	0.09	2.04	7.26	2.69	8.61	8.11	21.21
DS_1^2	Raw Data	1	1.0	5.91	0.34	0.11	2.39	10.98	3.31	11.34	7.67	27.60
LNDCF-Mean	Differencing	1	0.978	0.66	0.29	0.07	1.73	6.50	2.55	7.16	9.49	46.13
LNDCF-Max	Raw Data	1	0.999	1.02	0.37	0.14	3.16	19.08	4.37	14.48	11.16	53.93
MIX-DS_1^2 with CF-Trend/Cycle and LNDCF-Mean with Differencing		1	1.0	0.16	0.34	0.06	1.42	3.72	1.93	6.27	8.81	33.67
D3CS		2	0.614	1.19	0.37	0.10	2.34	37.69	6.14	10.42	7.58	
SDI		2	0.576	1.68	0.14	0.08	1.81	15.54	3.94	8.01	3.01	
SPL		2	0.610	11.34	0.10	0.06	1.52	35.16	5.93	5.71	2.04	
DS_1^2	CF-Trend/Cycle	2	1.000	2.36	0.33	0.09	2.10	7.94	2.82	8.95	7.37	39.89
DS_1^2	Raw Data	2	1.000	4.83	0.34	0.10	2.33	10.24	3.20	10.21	7.50	27.05
LNDCF-Mean	Differencing	2	0.958	0.68	0.27	0.08	2.23	10.10	3.18	8.00	9.00	43.35
LNDCF-Max	Raw Data	2	0.981	1.18	0.24	0.11	2.71	15.98	4.00	11.00	8.43	42.37
MIX-DS_1^2 with CF-Trend/Cycle and LNDCF-Mean with Differencing		2	1.0	0.20	0.32	0.06	1.65	4.94	2.22	6.46	8.30	41.62
D3CS		3	0.638	0.69	0.39	0.16	3.19	65.76	8.11	15.75	7.12	
SDI		3	0.574	1.18	0.16	0.11	1.94	22.96	4.79	11.16	3.37	
SPL		3	0.624	4.40	0.14	0.09	1.92	45.66	6.76	8.86	2.42	
DS_1^2	CF-Trend/Cycle	3	1.0	1.35	0.39	0.09	2.06	7.40	2.72	8.87	9.13	20.65
DS_1^2	Raw Data	3	1.0	5.40	0.33	0.11	2.41	10.86	3.30	11.11	7.45	26.12
LNDCF-Mean	Differencing	3	0.981	1.17	0.30	0.09	2.06	8.68	2.95	8.97	9.88	42.18
LNDCF-Max	Raw Data	3	0.985	0.94	0.35	0.14	3.21	20.51	4.53	14.36	10.98	48.40
MIX-DS_1^2 with CF-Trend/Cycle and LNDCF-Mean with Differencing		3	1.0	0.21	0.35	0.07	1.57	4.38	2.09	6.78	9.47	31.42
D3CS		4	0.649	0.41	0.50	0.19	4.23	47.37	6.88	18.71	10.50	
SDI		4	0.616	1.94	0.19	0.13	2.91	31.06	5.57	12.80	4.07	
SPL		4	0.642	3.92	0.11	0.09	2.13	22.53	4.75	8.55	2.13	
DS_1^2	CF-Trend/Cycle	4	1.0	0.93	0.50	0.09	2.05	7.41	2.72	9.05	11.77	19.82
DS_1^2	Raw Data	4	1.0	0.84	0.55	0.11	2.52	11.92	3.45	11.04	11.21	24.91
LNDCF-Mean	Differencing	4	0.983	0.70	0.30	0.08	1.83	6.41	2.53	8.10	9.76	43.19
LNDCF-Max	Raw Data	4	0.984	0.86	0.39	0.16	4.05	27.25	5.22	16.42	11.98	45.15
MIX-DS_1^2 with CF-Trend/Cycle and LNDCF-Mean with Differencing		4	1.0	0.13	0.41	0.06	1.45	3.89	1.97	6.48	10.77	31.51
Top methods		Rungs		Average								
SPL		0-4	0.667	4.31	0.10	0.0640	1.60	25.21	4.82	6.40	2.06	
MIX-DS_1^2 with CF-Trend/Cycle and LNDCF-Mean with Differencing		0-4	1.0	0.17	0.34	0.0638	1.48	4.11	2.02	6.37	9.06	35.35

Rungs, and the DS method has an average search length of 26.16 days for the five Rungs. This indicates that the latter method has almost half the value of the average search length that the LNDCF technique has. This leads us to conclude that it is necessary to limit the boundaries of the search range when using the DS method, and that it is preferable to have a more or less accurate idea of the value of the time delay of a strong lens system. In Figure 6.1 and Figure 6.2 the average of the results from the five rungs of the different methods shown in Table 6.1 is plotted on a set of bar graphs. Figure 6.1 shows the a) efficiency measurement results, where the bar with the highest score is the bar with the highest yield, the rest of the bar graphs indicate the contrary where a lower value is better.

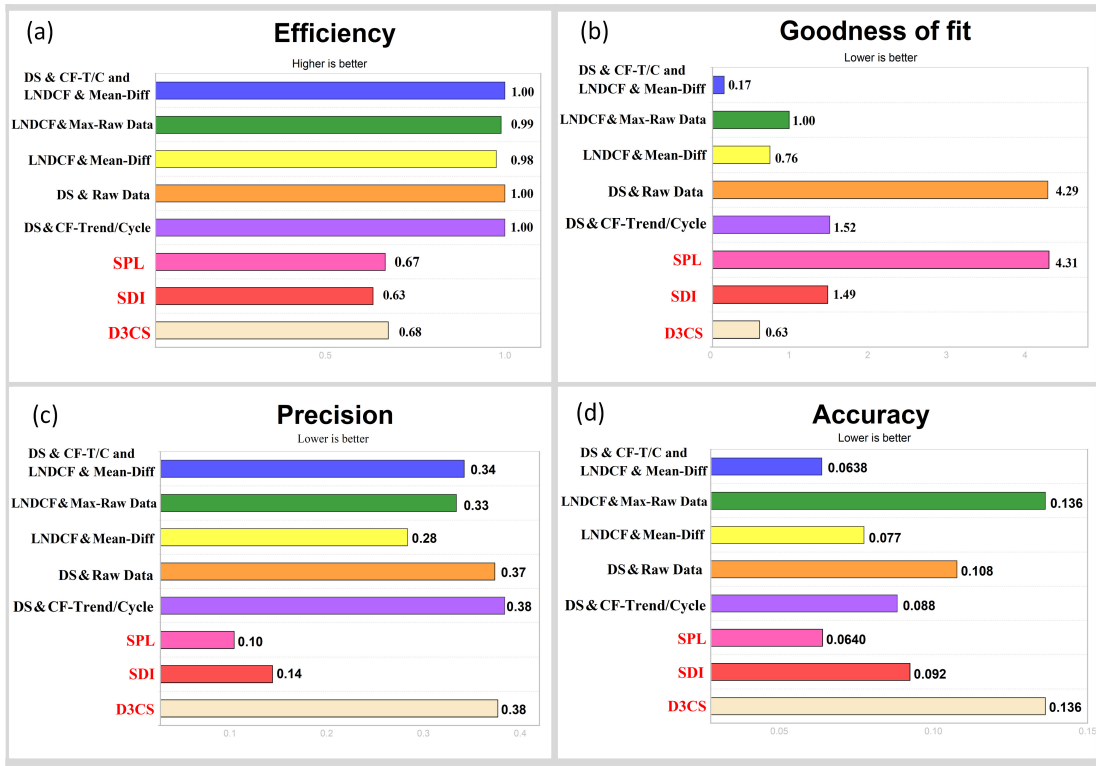


Figure 6.1: Performance charts of the average of all rungs 0-4, where lower is better except for the efficiency chart where higher is better : a) efficiency, b) goodness of fit, c) precision, d) accuracy.

Below are shown the complete results that we obtained when analyzing the synthetic data of TDC1 a dataset compound of five subsets: Table 6.6 for Rung 0, Table 6.5 for Rung 1, Table 6.4 for Rung 2, Table 6.3 for Rung 3 and Table 6.2 for Rung 4. For comparing purposes three methods (D3CS, SDI & SPL) used by the COSMOGRAIL team in the TDC1 are presented. Moreover, as we mentioned in section 2.3 the results from Kumar et al. [70] and Rathna Kumar, S. et al. [94] using the difference smoothing technique are shown for Rung 3 and Rung 4. In addition, the time delay estimations produced by Shafieloo et al. [101] and Aghamousa & Shafieloo [3] with a cross-correlation method for Rung 3 and Rung

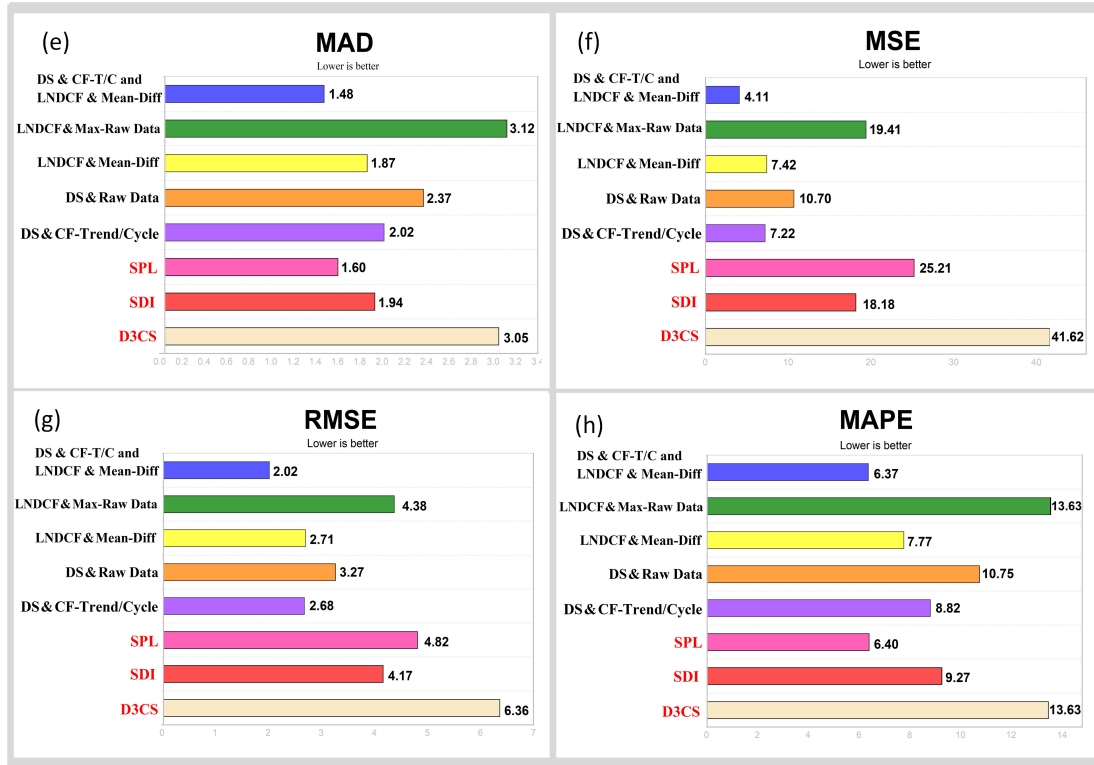


Figure 6.2: Performance charts of the average of all runs 0-4, where lower is better, e) MAD, f) MSE, g) RMSE, h) MAPE.

4 are shown.

Table 6.2: Complete results from Rung 4. Lower is better (except column labeled f where higher is better). Preprocessing: 1) Christiano-Fitzgerald filter, Trend-(CF-Trend); 2) Christiano-Fitzgerald filter, Cycle-(CF-Cycle); 3) Christiano-Fitzgerald filter (CF-Trend/Cycle), combines two components: Trend-(CF-Trend) and Cycle-(CF-Cycle); 4) Differencing; 5) Simple Net Return (SNR); 6) Wavelet Denoising (WD); 7) Raw Data (preprocessing was not applied). Delay estimation methods from COSMOGRAIL [9]: a) D3 Curve Shifting (D3CS); b) Free-Knot Spline Fit (SPL); c) Spline Difference (SDI). Delay estimation methods with preprocessed data: A) * Rathna Kumar, Stalin, & Prabhu: Difference-smoothing, first measured delay; B) **Rathna Kumar: Difference-smoothing [93], corrected delay; C) †Aghamousa & Shafieloo: Cross-correlation and smoothing, first measured delay; D) ††Aghamousa & Shafieloo: Cross-correlation and smoothing [3], calibrated measured delay. Delay estimation methods employed in this research: I) Dispersion Spectra (DS_1^2); II) Discrete Correlation Function (DCF); III) Locally Normalized Discrete Correlation Function (LNDCF); IV) Interpolated Cross-Correlation Function (ICCF). Selection techniques for Cross Correlation methods: i) Mean (proposed alternative, section 4.1); ii) Max, typical selection procedure of estimation (maximum point of correlation).

Methods	Preprocessing	f	χ^2	P	A	MAD	MSE	$RMSE$	$MAPE$	uncertainty	average search length
D3CS		0.649	0.41	0.50	0.19	4.23	47.37	6.88	18.71	10.50	
SDI		0.616	1.94	0.19	0.13	2.91	31.06	5.57	12.80	4.07	
SPL		0.642	3.92	0.11	0.09	2.13	22.53	4.75	8.55	2.13	
*Difference	Smoothing	0.546	0.43	0.12	0.05	1.62	25.08	5.01	5.56	2.84	
**Difference	Smoothing	0.546	0.40	0.12	0.06	1.43	5.52	2.35	6.16	2.84	
†Cross-Correlation	Smoothing	0.346	0.608	0.056	0.024						
††Cross-Correlation	Smoothing	0.346	0.868	0.040	0.0048						
DS_1^2	CF-Trend	1.0	2.19	0.51	0.14	2.83	13.78	3.71	13.55	10.74	21.79
DS_1^2	CF-Cycle	1.0	3.18	0.49	0.11	2.54	12.05	3.47	11.02	10.90	17.86
DS_1^2	CF-Trend/Cycle	1.0	0.93	0.50	0.09	2.05	7.41	2.72	9.05	11.77	19.82
DS_1^2	Differencing	1.0	3.43	0.51	0.12	2.71	13.20	3.63	12.02	11.55	17.90
DS_1^2	SNR	1.0	3.25	0.50	0.11	2.64	12.70	3.56	11.44	11.22	18.28
DS_1^2	WD	1.0	2.93	0.49	0.14	2.81	13.75	3.71	13.63	9.89	23.86
DS_1^2	Raw Data	1.0	0.84	0.55	0.11	2.52	11.92	3.45	11.04	11.21	24.91
DCF-Max	Differencing	0.992	1.37	0.30	0.15	3.56	22.43	4.74	15.21	9.50	45.46
DCF-Mean	Differencing	0.992	1.27	0.30	0.11	2.24	8.84	2.97	10.65	9.50	42.40
DCF-Max	Raw Data	0.989	1.39	0.46	0.21	4.94	34.64	5.89	21.39	13.19	48.16
DCF-Mean	Raw Data	0.989	1.27	0.46	0.14	2.71	11.43	3.38	13.67	13.19	47.26
DCF-Max	WD	0.993	2.00	0.46	0.23	5.33	38.40	6.20	23.08	13.49	44.41
DCF-Mean	WD	0.993	1.86	0.46	0.16	3.06	14.09	3.75	15.85	13.49	38.65
DCF-Max	SNR	0.997	1.76	0.46	0.21	4.90	34.29	5.86	21.03	13.29	44.17
DCF-Mean	SNR	0.997	1.61	0.46	0.14	2.76	11.53	3.40	14.21	13.29	38.74
DCF-Max	CF-Trend	0.982	1.92	0.45	0.22	5.33	38.56	6.21	22.30	13.43	44.05
DCF-Mean	CF-Trend	0.982	1.80	0.45	0.15	3.05	13.75	3.71	15.18	13.43	38.92
DCF-Max	CF-Cycle	0.985	1.71	0.45	0.21	4.98	34.95	5.91	21.24	13.36	44.86
DCF-Mean	CF-Cycle	0.985	1.61	0.45	0.15	2.94	12.79	3.58	14.54	13.36	38.76
DCF-Max	CF-Trend/Cycle	0.985	1.77	0.45	0.18	4.06	24.26	4.93	18.03	13.38	43.27
DCF-Mean	CF-Trend/Cycle	0.985	1.21	0.45	0.13	2.39	8.97	2.99	12.59	13.38	38.84
LNDCF-Max	Differencing	0.983	0.79	0.30	0.12	3.13	18.53	4.31	11.79	8.90	47.82
LNDCF-Mean	Differencing	0.983	0.70	0.30	0.08	1.83	6.41	2.53	8.10	9.76	43.19
LNDCF-Max	Raw Data	0.984	0.86	0.39	0.16	4.05	27.25	5.22	16.42	11.98	45.15
LNDCF-Mean	Raw Data	0.984	0.74	0.39	0.10	2.03	7.67	2.77	10.21	11.98	46.51
ICCF-Max	Differencing	0.958	8.73	0.22	0.21	5.24	38.50	6.21	21.22	6.00	66.18
ICCF-Mean	Differencing	0.958	8.48	0.22	0.17	4.22	25.70	5.07	17.16	6.00	42.86
ICCF-Max	Raw Data	0.977	3.90	0.28	0.26	6.68	53.25	7.30	26.40	7.68	75.43
ICCF-Mean	Raw Data	0.977	3.34	0.28	0.16	3.93	19.92	4.46	15.66	7.68	47.87
MIX- DS_1^2 with CF-Trend/Cycle and LNDCF-Mean with Differencing		1.0	0.13	0.41	0.06	1.45	3.89	1.97	6.48	10.77	41.22

Table 6.3: Complete results from Rung 3. Lower is better (except column labeled f where higher is better). Preprocessing: 1) Christiano-Fitzgerald filter, Trend-(CF-Trend); 2) Christiano-Fitzgerald filter, Cycle-(CF-Cycle); 3) Christiano-Fitzgerald filter (CF-Trend/Cycle), combines two components: Trend-(CF-Trend) and Cycle-(CF-Cycle); 4) Differencing; 5) Simple Net Return (SNR); 6) Wavelet Denoising (WD); 7) Raw Data (preprocessing was not applied). Delay estimation methods from COSMOGRAIL [9]: a) D3 Curve Shifting (D3CS); b) Free-Knot Spline Fit (SPL); c) Spline Difference (SDI). Delay estimation methods with preprocessed data: A) * Rathna Kumar, Stalin, & Prabhu: Difference-smoothing, first measured delay; B) **Rathna Kumar: Difference-smoothing [93], corrected delay; C) †Aghamousa & Shafieloo: Cross-correlation and smoothing, first measured delay; D) ††Aghamousa & Shafieloo: Cross-correlation and smoothing [3], calibrated measured delay. Delay estimation methods employed in this research: I) Dispersion Spectra (DS_1^2); II) Discrete Correlation Function (DCF); III) Locally Normalized Discrete Correlation Function (LNDCF); IV) Interpolated Cross-Correlation Function (ICCF). Selection techniques for Cross Correlation methods: i) Mean (proposed alternative, section 4.1); ii) Max, typical selection procedure of estimation (maximum point of correlation).

Methods	Preprocessing	f	χ^2	P	A	MAD	MSE	$RMSE$	$MAPE$	uncertainty	average search length
D3CS		0.638	0.69	0.39	0.16	3.19	65.76	8.11	15.75	7.12	
SDI		0.574	1.18	0.16	0.11	1.94	22.96	4.79	11.16	3.37	
SPL		0.624	4.40	0.14	0.09	1.92	45.66	6.76	8.86	2.42	
*Difference	Smoothing	0.505	0.39	0.06	0.02	0.86	26.35	5.13	2.66	1.45	
**Difference	Smoothing	0.505	0.46	0.06	0.03	0.98	26.64	5.16	3.30	1.45	
†Cross-Correlation	Smoothing	0.337	0.52	0.06	0.02						
††Cross-Correlation	Smoothing	0.337	0.77	0.04	0.001						
DS_1^2	CF-Trend	1.0	3.71	0.38	0.13	2.80	13.20	3.63	13.04	9.15	24.16
DS_1^2	CF-Cycle	1.0	4.99	0.40	0.12	2.61	12.93	3.60	11.56	9.11	17.15
DS_1^2	CF-Trend/Cycle	1.0	1.35	0.39	0.09	2.06	7.40	2.72	8.87	9.13	20.65
DS_1^2	Differencing	1.0	6.92	0.33	0.12	2.81	13.81	3.72	12.43	7.27	17.63
DS_1^2	SNR	1.0	6.63	0.33	0.12	2.66	12.62	3.55	11.85	7.29	17.59
DS_1^2	WD	1.0	5.54	0.31	0.12	2.61	11.90	3.45	12.39	6.94	26.34
DS_1^2	Raw Data	1.0	5.40	0.33	0.11	2.41	10.86	3.30	11.11	7.45	26.12
DCF-Max	Differencing	0.985	1.13	0.31	0.14	3.49	24.79	4.98	14.12	9.87	47.92
DCF-Mean	Differencing	0.985	1.02	0.31	0.09	2.10	8.81	2.97	9.35	9.86	40.82
DCF-Max	Raw Data	0.986	0.83	0.45	0.18	4.13	27.17	5.21	17.91	12.89	53.94
DCF-Mean	Raw Data	0.986	0.73	0.45	0.12	2.33	9.15	3.03	11.53	12.89	51.01
DCF-Max	WD	0.987	1.72	0.48	0.23	5.36	38.95	6.24	22.95	14.36	48.47
DCF-Mean	WD	0.987	1.60	0.48	0.15	2.92	12.99	3.60	14.97	14.36	38.17
DCF-Max	SNR	0.982	1.46	0.47	0.22	5.23	37.75	6.14	22.27	13.92	47.46
DCF-Mean	SNR	0.982	1.31	0.47	0.14	2.75	11.79	3.43	13.52	13.92	36.60
DCF-Max	CF-Trend	0.983	1.85	0.49	0.24	5.55	41.34	6.43	23.87	14.43	48.23
DCF-Mean	CF-Trend	0.983	1.70	0.49	0.15	2.96	13.41	3.66	14.92	14.43	38.74
DCF-Max	CF-Cycle	0.979	1.81	0.48	0.23	5.27	38.89	6.24	22.65	14.03	47.46
DCF-Mean	CF-Cycle	0.979	1.68	0.48	0.14	2.76	12.03	3.47	13.95	14.03	37.14
DCF-Max	CF-Trend/Cycle	0.984	1.43	0.49	0.20	4.37	27.66	5.26	20.08	14.21	47.85
DCF-Mean	CF-Trend/Cycle	0.984	1.29	0.49	0.13	2.40	9.22	3.04	13.03	14.20	37.94
LNDCF-Max	Differencing	0.981	1.28	0.30	0.13	3.26	21.55	4.64	12.81	9.88	47.05
LNDCF-Mean	Differencing	0.981	1.17	0.30	0.09	2.06	8.68	2.95	8.97	9.88	42.18
LNDCF-Max	Raw Data	0.985	0.94	0.35	0.14	3.21	20.51	4.53	14.36	10.98	48.40
LNDCF-Mean	Raw Data	0.985	0.84	0.35	0.10	1.91	7.46	2.73	9.63	10.98	49.68
ICCF-Max	Differencing	0.969	7.05	0.25	0.23	5.49	39.63	6.30	23.11	6.92	58.89
ICCF-Mean	Differencing	0.969	6.86	0.25	0.19	4.43	26.72	5.17	19.10	6.92	41.06
ICCF-Max	Raw Data	0.985	4.52	0.29	0.29	6.93	55.33	7.44	28.89	7.84	74.19
ICCF-Mean	Raw Data	0.985	4.04	0.29	0.18	4.05	20.72	4.55	17.77	7.84	46.82
MIX-DS₁² with CF-Trend/Cycle and LNDCF-Mean with Differencing		1.0	0.21	0.35	0.07	1.57	4.38	2.09	6.78	9.47	31.26

Table 6.4: Complete results from Rung 2-lower is better (except column labeled f where higher is better). Preprocessing: 1) Christiano-Fitzgerald filter, Trend-(CF-Trend); 2) Christiano-Fitzgerald filter, Cycle-(CF-Cycle); 3) Christiano-Fitzgerald filter (CF-Trend/Cycle), combines two components: Trend-(CF-Trend) and Cycle-(CF-Cycle); 4) Differencing; 5) Simple Net Return (SNR); 6) Wavelet Denoising (WD); 7) Raw Data (preprocessing was not applied). Delay estimation methods from COSMOGRAIL [9]: a) D3 Curve Shifting (D3CS); b) Free-Knot Spline Fit (SPL); c) Spline Difference (SDI). Delay estimation methods employed in this research: I) Dispersion Spectra (DS_1^2); II) Discrete Correlation Function (DCF); III) Locally Normalized Discrete Correlation Function (LNDCF); IV) Interpolated Cross-Correlation Function (ICCF). Selection techniques for Cross Correlation methods: i) Mean (proposed alternative, section 4.1); ii) Max, typical selection procedure of estimation (maximum point of correlation).

Methods	Preprocessing	f	χ^2	P	A	MAD	MSE	$RMSE$	$MAPE$	uncertainty	average search length
D3CS		0.614	1.19	0.37	0.10	2.34	37.69	6.14	10.42	7.58	
SDI		0.576	1.68	0.14	0.08	1.81	15.54	3.94	8.01	3.01	
SPL		0.610	11.34	0.10	0.06	1.52	35.16	5.93	5.71	2.04	
DS_1^2	CF-Trend	1.0	5.08	0.33	0.12	2.70	12.73	3.57	12.33	7.65	23.41
DS_1^2	CF-Cycle	1.0	7.51	0.34	0.11	2.59	13.28	3.64	11.14	7.10	16.48
DS_1^2	CF-Trend/Cycle	1.0	2.36	0.33	0.09	2.10	7.94	2.82	8.95	7.37	39.89
DS_1^2	Differencing	1.0	5.76	0.38	0.11	2.66	13.37	3.66	11.43	8.75	17.63
DS_1^2	SNR	1.0	7.45	0.35	0.11	2.66	13.47	3.67	11.44	7.52	17.28
DS_1^2	WD	1.0	5.92	0.30	0.12	2.65	12.27	3.50	12.23	6.92	25.65
DS_1^2	Raw Data	1.0	4.83	0.34	0.10	2.33	10.24	3.20	10.21	7.50	27.05
DCF-Max	Differencing	0.981	1.42	0.28	0.13	3.35	22.47	4.74	12.82	9.10	44.83
DCF-Mean	Differencing	0.981	1.63	0.28	0.10	2.36	11.25	3.35	9.96	9.10	42.51
DCF-Max	Raw Data	0.985	2.19	0.27	0.16	4.00	28.38	5.33	15.55	8.81	40.79
DCF-Mean	Raw Data	0.985	2.09	0.27	0.11	2.60	12.50	3.54	11.28	8.81	48.75
DCF-Max	WD	0.965	3.82	0.32	0.27	6.95	56.82	7.54	27.01	10.49	34.46
DCF-Mean	WD	0.965	3.56	0.32	0.17	3.93	21.24	4.61	17.40	10.49	32.83
DCF-Max	SNR	0.972	3.05	0.35	0.25	6.17	48.52	6.97	24.60	11.06	37.93
DCF-Mean	SNR	0.972	2.97	0.35	0.17	3.82	20.71	4.55	17.18	11.06	34.34
DCF-Max	CF-Trend	0.979	3.61	0.32	0.27	6.93	56.85	7.54	26.96	10.40	34.21
DCF-Mean	CF-Trend	0.979	3.36	0.32	0.18	4.11	22.33	4.73	18.35	10.40	33.88
DCF-Max	CF-Cycle	0.959	2.85	0.34	0.23	5.95	45.89	6.77	23.21	10.90	41.21
DCF-Mean	CF-Cycle	0.959	2.71	0.34	0.16	3.64	18.86	4.34	15.67	10.90	34.32
DCF-Max	CF-Trend/Cycle	0.990	1.94	0.32	0.18	4.31	27.25	5.22	18.27	10.46	37.71
DCF-Mean	CF-Trend/Cycle	0.990	1.85	0.32	0.13	2.76	11.55	3.40	13.42	10.46	34.20
LNDCF-Max	Differencing	0.958	0.71	0.27	0.11	3.25	21.49	4.64	11.10	9.00	42.88
LNDCF-Mean	Differencing	0.958	0.68	0.27	0.08	2.23	10.10	3.18	8.00	9.00	43.35
LNDCF-Max	Raw Data	0.981	1.18	0.24	0.11	2.71	15.98	4.00	11.00	8.43	42.37
LNDCF-Mean	Raw Data	0.981	1.17	0.24	0.09	1.91	7.84	2.80	8.69	8.43	51.89
ICCF-Max	Differencing	0.987	8.23	0.27	0.29	5.43	39.16	6.26	23.20	7.04	57.50
ICCF-Mean	Differencing	0.984	6.58	0.25	0.21	4.61	29.25	5.41	20.53	7.04	46.45
ICCF-Max	Raw Data	0.987	8.23	0.27	0.29	6.90	55.02	7.42	28.67	7.54	75.48
ICCF-Mean	Raw Data	0.987	5.24	0.27	0.18	4.13	21.17	4.60	17.66	7.54	47.25
MIX-DS₁² with CF-Trend/Cycle and LNDCF-Mean with Differencing		1.0	0.20	0.32	0.06	1.65	4.94	2.22	6.46	8.30	31.18

Table 6.5: Complete results from Rung 1-lower is better (except column labeled f where higher is better). Preprocessing: 1) Christiano-Fitzgerald filter, Trend-(CF-Trend); 2) Christiano-Fitzgerald filter, Cycle-(CF-Cycle); 3) Christiano-Fitzgerald filter (CF-Trend/Cycle), combines two components: Trend-(CF-Trend) and Cycle-(CF-Cycle); 4) Differencing; 5) Simple Net Return (SNR); 6) Wavelet Denoising (WD); 7) Raw Data (preprocessing was not applied). Delay estimation methods from COSMOGRAIL [9]: a) D3 Curve Shifting (D3CS); b) Free-Knot Spline Fit (SPL); c) Spline Difference (SDI). Delay estimation methods employed in this research: I) Dispersion Spectra (DS_1^2); II) Discrete Correlation Function (DCF); III) Locally Normalized Discrete Correlation Function (LNDCF); IV) Interpolated Cross-Correlation Function (ICCF). Selection techniques for Cross Correlation methods: i) Mean (proposed alternative, section 4.1); ii) Max, typical selection procedure of estimation (maximum point of correlation).

Methods	Preprocessing	f	χ^2	P	A	MAD	MSE	$RMSE$	$MAPE$	uncertainty	average search length
D3CS		0.655	0.38	0.31	0.12	2.72	26.27	5.13	11.84	6.75	
SDI		0.618	1.66	0.12	0.07	1.51	11.07	3.33	7.55	3.21	
SPL		0.651	0.86	0.08	0.05	1.25	7.56	2.75	4.55	1.82	
DS_1^2	CF-Trend	1.0	3.40	0.39	0.13	2.84	13.65	3.69	13.31	9.21	25.32
DS_1^2	CF-Cycle	1.0	6.82	0.34	0.10	2.45	12.09	3.48	10.05	7.01	17.09
DS_1^2	CF-Trend/Cycle	1.0	1.56	0.37	0.09	2.04	7.26	2.69	8.61	8.11	21.21
DS_1^2	Differencing	1.0	5.39	0.38	0.11	2.49	12.36	3.52	10.62	8.64	17.33
DS_1^2	SNR	1.0	7.30	0.35	0.11	2.54	12.66	3.56	10.83	7.44	17.37
DS_1^2	WD	1.0	6.18	0.29	0.12	2.55	12.05	3.47	12.26	6.64	27.92
DS_1^2	Raw Data	1.0	5.91	0.34	0.11	2.39	10.98	3.31	11.34	7.67	27.60
DCF-Max	Differencing	0.993	1.19	0.28	0.12	2.97	19.53	4.42	12.33	9.21	47.11
DCF-Mean	Differencing	0.993	1.11	0.28	0.09	1.86	7.49	2.74	8.63	9.21	43.37
DCF-Max	Raw Data	0.998	4.22	0.48	0.22	4.79	33.86	5.82	22.13	13.73	49.36
DCF-Mean	Raw Data	0.998	4.02	0.48	0.14	2.68	11.57	3.40	14.25	13.73	44.31
DCF-Max	WD	0.996	1.84	0.49	0.24	5.38	39.20	6.26	24.24	14.29	46.68
DCF-Mean	WD	0.996	1.72	0.49	0.16	2.95	13.16	3.63	15.52	14.29	37.48
DCF-Max	SNR	0.995	2.35	0.47	0.24	5.21	37.89	6.16	23.98	13.86	46.99
DCF-Mean	SNR	0.995	2.23	0.47	0.16	2.88	12.93	3.60	15.77	13.86	36.92
DCF-Max	CF-Trend	0.990	1.69	0.49	0.24	5.43	39.66	6.30	24.97	14.47	47.32
DCF-Mean	CF-Trend	0.990	1.58	0.49	0.15	2.95	13.38	3.66	15.25	14.47	37.97
DCF-Max	CF-Cycle	0.986	1.38	0.46	0.22	5.07	37.08	6.09	22.64	13.65	47.24
DCF-Mean	CF-Cycle	0.986	1.26	0.46	0.13	2.68	11.42	3.38	13.49	13.65	36.09
DCF-Max	CF-Trend/Cycle	0.996	0.81	0.48	0.18	3.85	21.77	4.67	18.60	13.99	47.17
DCF-Mean	CF-Trend/Cycle	0.996	0.74	0.48	0.12	2.18	7.64	2.76	12.42	13.99	36.98
LNDCF-Max	Differencing	0.978	0.73	0.29	0.10	2.70	16.95	4.12	10.39	9.49	48.98
LNDCF-Mean	Differencing	0.978	0.66	0.29	0.07	1.73	6.50	2.55	7.16	9.49	46.13
LNDCF-Max	Raw Data	0.999	1.02	0.37	0.14	3.16	19.08	4.37	14.48	11.16	53.93
LNDCF-Mean	Raw Data	0.999	0.77	0.37	0.09	1.81	6.17	2.48	9.30	11.16	52.71
ICCF-Max	Differencing	0.998	7.83	0.24	0.23	5.43	39.40	6.28	23.39	6.71	62.30
ICCF-Mean	Differencing	0.998	7.64	0.24	0.19	4.25	25.32	5.03	18.98	6.71	41.99
ICCF-Max	Raw Data	0.993	5.87	0.29	0.29	6.94	55.46	7.45	29.14	7.62	74.60
ICCF-Mean	Raw Data	0.993	4.60	0.29	0.18	4.01	20.05	4.48	17.64	7.62	46.60
MIX-DS₁² with CF-Trend/Cycle and LNDCF-Mean with Differencing		1.0	0.16	0.34	0.06	1.42	3.72	1.93	6.27	8.81	33.31

Table 6.6: Complete results from Rung 0-lower is better (except column labeled f where higher is better). Preprocessing: 1) Christiano-Fitzgerald filter, Trend-(CF-Trend); 2) Christiano-Fitzgerald filter, Cycle-(CF-Cycle); 3) Christiano-Fitzgerald filter (CF-Trend/Cycle), combines two components: Trend-(CF-Trend) and Cycle-(CF-Cycle); 4) Differencing; 5) Simple Net Return (SNR); 6) Wavelet Denoising (WD); 7) Raw Data (preprocessing was not applied). Delay estimation methods from COSMOGRAIL [9]: a) D3 Curve Shifting (D3CS); b) Free-Knot Spline Fit (SPL); c) Spline Difference (SDI). Delay estimation methods employed in this research: I) Dispersion Spectra (DS_1^2); II) Discrete Correlation Function (DCF); III) Locally Normalized Discrete Correlation Function (LNDCF); IV) Interpolated Cross-Correlation Function (ICCF). Selection techniques for Cross Correlation methods: i) Mean (proposed alternative, section 4.1); ii) Max, typical selection procedure of estimation (maximum point of correlation).

Methods	Preprocessing	f	χ^2	P	A	MAD	MSE	$RMSE$	$MAPE$	uncertainty	average search length
D3CS		0.819	0.46	0.32	0.11	2.76	31.00	5.57	11.42	6.95	
SDI		0.778	1.00	0.11	0.07	1.52	10.27	3.21	6.82	2.48	
SPL		0.807	1.01	0.09	0.04	1.21	15.15	3.89	4.34	1.88	
DS_1^2	CF-Trend	1.0	3.24	0.37	0.14	2.67	12.22	3.50	13.47	8.67	29.61
DS_1^2	CF-Cycle	1.0	5.41	0.28	0.11	2.21	10.11	3.18	10.57	5.91	20.53
DS_1^2	CF-Trend/Cycle	1.0	1.38	0.32	0.09	1.84	6.09	2.47	8.60	7.29	25.07
DS_1^2	Differencing	1.0	4.42	0.34	0.11	2.32	10.34	3.22	10.43	7.75	20.98
DS_1^2	SNR	1.0	4.98	0.31	0.10	2.21	9.75	3.12	10.26	6.46	21.47
DS_1^2	WD	1.0	4.73	0.30	0.12	2.41	10.79	3.29	11.93	6.63	30.27
DS_1^2	Raw Data	1.0	4.46	0.32	0.10	2.21	9.49	3.08	10.06	6.97	29.30
DCF-Max	Differencing	0.989	1.40	0.41	0.17	4.32	30.42	5.52	17.49	12.60	52.11
DCF-Mean	Differencing	0.989	1.33	0.41	0.13	2.71	12.83	3.58	12.77	12.60	43.70
DCF-Max	Raw Data	0.995	1.14	0.49	0.21	5.12	37.00	6.08	20.97	14.25	52.36
DCF-Mean	Raw Data	0.995	1.02	0.49	0.13	2.66	11.43	3.38	12.66	14.25	47.52
DCF-Max	WD	0.995	1.83	0.48	0.24	5.50	40.58	6.37	24.11	14.34	51.97
DCF-Mean	WD	0.995	1.67	0.48	0.15	2.96	13.52	3.68	15.19	14.34	42.55
DCF-Max	SNR	0.997	2.10	0.46	0.23	4.98	35.27	5.94	22.58	13.73	52.63
DCF-Mean	SNR	0.997	1.98	0.46	0.15	2.86	12.91	3.59	15.10	13.73	41.13
DCF-Max	CF-Trend	0.994	1.74	0.49	0.23	5.51	40.88	6.39	23.36	14.55	52.64
DCF-Mean	CF-Trend	0.994	1.60	0.49	0.15	2.94	13.36	3.65	14.88	14.55	42.65
DCF-Max	CF-Cycle	0.996	1.51	0.46	0.22	4.85	34.32	5.86	21.59	13.43	51.87
DCF-Mean	CF-Cycle	0.996	1.39	0.46	0.14	2.86	12.79	3.58	14.32	13.43	40.50
DCF-Max	CF-Trend/Cycle	0.999	0.60	0.47	0.17	3.63	20.21	4.50	17.03	13.93	52.22
DCF-Mean	CF-Trend/Cycle	0.999	0.53	0.47	0.12	2.21	7.69	2.77	11.96	13.93	41.54
LNDCF-Max	Differencing	0.979	0.62	0.27	0.09	2.17	12.77	3.57	9.44	8.60	53.23
LNDCF-Mean	Differencing	0.979	0.57	0.27	0.07	1.48	5.40	2.32	6.63	8.60	52.04
LNDCF-Max	Raw Data	0.997	0.99	0.32	0.12	2.48	14.22	3.77	11.87	10.04	57.04
LNDCF-Mean	Raw Data	0.997	0.91	0.32	0.08	1.57	5.49	2.34	8.13	10.04	58.26
ICCF-Max	Differencing	0.996	7.95	0.24	0.24	5.39	39.16	6.26	23.74	6.95	52.61
ICCF-Mean	Differencing	0.996	7.79	0.24	0.20	4.33	26.27	5.13	19.98	6.95	39.31
ICCF-Max	Raw Data	0.994	4.20	0.30	0.29	6.95	55.70	7.46	29.33	7.83	75.52
ICCF-Mean	Raw Data	0.994	3.70	0.30	0.17	3.88	18.96	4.35	17.08	7.83	47.49
MIX-DS_1^2 with CF-Trend/Cycle and LNDCF-Mean with Differencing		1.0	0.17	0.30	0.059	1.30	3.60	1.90	5.89	7.95	38.18

Chapter 7

Conclusions and future work

7.1 Concluding remarks

A great number of different elementary statistical methods have been the cornerstone of many branches of scientific research for pattern recognition. However, under the context of the study and analysis of different astronomical phenomena, and their inherent complexity in the data (data is highly dimensional due to recently developed generation of astronomical surveys, noisy, irregularly sampled, nonlinear, among other traits) produced by the diverse heterogenous and complex stellar systems. Hence, we can argue that the new era of astrophysical research could be maintained, fuelled and guided by computational algorithms, based on machine learning, big data, and data science. The proposed methodology aims to assemble the aforementioned techniques and adapt them to reach a high level of accuracy at estimating the time delay of strong gravitational lens systems. Far from being a “black box” methodology, we intend to be pragmatic, applying alternative tools and employing cutting-edge techniques, to gain deeper insight into this burgeoning field of knowledge.

With the results obtained, it is possible to affirm that for the models used in this research, in general the transformation of data, helps to significantly improve the precision of time delays estimation. It is also concluded that correlation models benefit from using stationary time series, specifically with the transformation technique of differencing time series. The ICCF, DCF, and LNDCF correlation methods (subsection 4.1) are an extension of the classical correlation method and have been modified to fit the irregular sampling of data. The traditional technique for selection in a correlation algorithm has the form of choosing its best estimate with the correlation point with the highest numerical magnitude over a set of correlation points. However, it is worth asking why use the traditional way of choosing the cross-correlation method; if astronomical data has its peculiarities, and therefore the methods have been modified to handle these issues.

Based on the foregoing and with the results obtained, it is possible to conclude that the alternative procedure proposed (estimation by mean) (see subsection 4.1) for the cross-correlation methods in this research to choose the time delays estimations is better than the traditional form of choice. The decomposition of data with the Christiano-Fitzgerald filter

improves the time delays estimation of Dispersion Spectra, unlike correlation methods that benefit from stationarizing a time series. This can be predicted when understanding how the correlation algorithms in the stage where both light curves A and B merge in a single time series, due to the implementation of a technique very similar to differentiation, in this way the signals would be over-differentiated, which results in something that negatively affects the estimation of time delays as in the same way as it is reported in the literature [121, 28, 61]. Of the three different correlation techniques, the one that had the best performance was the LNDCF with the choice of the delay estimation through its mean (see bottom of Table 6.1) and with the data transformed with the differencing time series algorithm (subsection 5.1.4). On the other hand, the best performance of the DS method was obtained when it was used in conjunction with the Christiano-Fitzgerald filter method (subsection 5.1.4). Finally, we find that the best overall results achieved is the combination of both aforementioned methods (DS&CF-Trend/Cycle and LNDCF-Mean&Differencing, see subsection 4.1.1).

7.2 Future work

We propose to use other techniques for evaluation, by developing alternative metrics. As a consequence of the overall performance of the DS method with the data decomposed by the CF technique, it is feasible to assume that other bandpass filters like the Baxter & King filter (BKF) [7] and Hodrick & Prescott filter (HPF) [57], could also work well. Also, signal decomposition techniques such as Fast Fourier Transform (FFT), Empirical Mode Decomposition (EMD) [63], among other methods can be used. Finally, preprocessed data may also be used in conjunction with other techniques for estimating delay over time (SDI, SPL). We studied traditional time delay estimation methods such as DCF, LNDCF and DS, and we compared the performance against those methods introduced with the TDC1 dataset, i.e. D3CS, SDI and SPL. We also would like to compare the performance against other methods such as kernel methods [21], evolutionary kernel methods [23, 25] and general regression neural networks [24]. Because these methods are based on machine learning techniques, high computational resources are required. Therefore, parallel programming on high performance computing is involved.

Bibliography

- [1] Abell, P. A., Allison, J., Anderson, S. F., Andrew, J. R., Angel, J. R. P., Armus, L., Arnett, D., Asztalos, S., Axelrod, T. S., Bailey, S., et al. (2009). Lsst science book, version 2.0. *arXiv preprint arXiv:0912.0201*.
- [2] Aghamousa, A. and Shafieloo, A. (2014). Fast and reliable time delay estimation of strong lens systems using method of smoothing and cross-correlation. *arXiv preprint arXiv:1410.8122*.
- [3] Aghamousa, A. and Shafieloo, A. (2015). Fast and reliable time delay estimation of strong lens systems using the smoothing and cross-correlation methods. *The Astrophysical Journal*, 804(1):39.
- [4] Aghanim, N., Akrami, Y., Ashdown, M., Aumont, J., Baccigalupi, C., Ballardini, M., Banday, A., Barreiro, R., Bartolo, N., Basak, S., et al. (2020). Planck 2018 results-vi. cosmological parameters. *Astronomy & Astrophysics*, 641:A6.
- [5] Al Otaibi, S., Tiño, P., Cuevas-Tello, J. C., Mandel, I., and Raychaudhury, S. (2016). Kernel regression estimates of time delays between gravitationally lensed fluxes. *Monthly Notices of the Royal Astronomical Society*, 459(1):573–584.
- [6] Ball, N. M. and Brunner, R. J. (2010). Data mining and machine learning in astronomy. *International Journal of Modern Physics D*, 19(07):1049–1106.
- [7] Baxter, M. and King, R. G. (1999). Measuring business cycles: approximate band-pass filters for economic time series. *Review of economics and statistics*, 81(4):575–593.
- [8] Blandford, R. D. and Narayan, R. (1992). Cosmological applications of gravitational lensing. *Annual Review of Astronomy and Astrophysics*, 30(1):311–358.
- [9] Bonvin, V., Tewes, M., Courbin, F., Kuntzer, T., Sluse, D., and Meylan, G. (2016). Cosmograil: the cosmological monitoring of gravitational lenses-xv. assessing the achievability and precision of time-delay measurements. *Astronomy & Astrophysics*, 585:A88.
- [10] Bonvin, V., Chan, J. H. H., Millon, M., Rojas, K., Courbin, F., Chen, G. C.-F., Fassnacht, C. D., Paic, E., Tewes, M., Chao, D. C.-Y., Chijani, M., Gilman, D., Gilmore, K., Williams, P., Buckley-Geer, E., Frieman, J., Marshall, P. J., Suyu, S. H., Treu, T.,

- Hempel, A., Kim, S., Lachaume, R., Rabus, M., Anguita, T., Meylan, G., Motta, V., and Magain, P. (2018). Cosmograil - xvii. time delays for the quadruply imaged quasar pg 1115 + 080. *A&A*, 616:A183.
- [11] Bostock, M., Ogievetsky, V., and Heer, J. (2011). D³ data-driven documents. *IEEE Transactions on Visualization & Computer Graphics*, (12):2301–2309.
- [12] Box, G. and Jenkins, G. (1976a). Holden-day series in time series analysis. *San Francisco: Holden-Day*.
- [13] Box, G. and Jenkins, G. M. (1976b). *Time Series Analysis: Forecasting and Control*. Holden-Day.
- [14] Brooks, C. (2019). *Introductory Econometrics for Finance*. Cambridge University Press.
- [15] Cady, F. (2017). *The Data Science Handbook*. ohn Wiley & Sons.
- [16] Campbell, J. Y., Lo, A. W., MacKinlay, A. C., and Whitelaw, R. F. (1998). The econometrics of financial markets. *Macroeconomic Dynamics*, 2(4):559–562.
- [17] Christiano, L. J. and Fitzgerald, T. J. (2003). The band pass filter. *International economic review*, 44(2):435–465.
- [18] Chu, X., Ilyas, I. F., Krishnan, S., and Wang, J. (2016). Data cleaning: Overview and emerging challenges. In *Proceedings of the 2016 international conference on management of data*, pages 2201–2206.
- [19] Croston, J. D. (1972). Forecasting and stock control for intermittent demands. *Journal of the Operational Research Society*, 23(3):289–303.
- [20] Cuesta, H. and Kumar, S. (2016). *Practical data analysis*. Packt Publishing Ltd.
- [21] Cuevas-Tello, J., Tino, P., and Raychaudhury, S. (2006). How accurate are the time delay estimates in gravitational lensing? *Astronomy & Astrophysics*, 454(3):695–706.
- [22] Cuevas Tello, J. C. (2007). *Estimating time delays between irregularly sampled time series*. PhD thesis, University of Birmingham.
- [23] Cuevas-Tello, J. C. (2007). Evolved kernel method for time series. In Gelbukh, A. and Kuri Morales, Á. F., editors, *MICAI 2007: Advances in Artificial Intelligence*, pages 559–569, Berlin, Heidelberg. Springer Berlin Heidelberg.
- [24] Cuevas-Tello, J. C., Gonzalez-Grimaldo, R., Rodriguez-Gonzalez, O., Perez-Gonzalez, H., and Vital-Ochoa, O. (2012). Parallel approach for time series analysis with general regression neural networks. *Journal of applied research and technology*, 10:162–179.

- [25] Cuevas-Tello, J. C., Tiño, P., Raychaudhury, S., Yao, X., and Harva, M. (2010). Uncovering delayed patterns in noisy and irregularly sampled time series: an astronomy application. *Pattern Recognition*, 43(3):1165–1179.
- [26] Dasu, T. and Johnson, T. (2003). *Exploratory data mining and data cleaning*. John Wiley & Sons.
- [27] Dasu, T. and Theodore, J. (2003). *Exploratory data mining and data cleaning*. John Wiley & Sons.
- [28] De Prado, M. L. (2018). *Advances in financial machine learning*. John Wiley & Sons.
- [29] Dicke, R. (1953). The effect of collisions upon the doppler width of spectral lines. *Physical Review*, 89(2):472.
- [30] Dickey, D. A. and Fuller, W. A. (1979). Distribution of the estimators for autoregressive time series with a unit root. *Journal of the American statistical association*, 74(366a):427–431.
- [31] Dickey, D. A. and Fuller, W. A. (1981). Likelihood ratio statistics for autoregressive time series with a unit root. *Econometrica: journal of the Econometric Society*, pages 1057–1072.
- [32] Dierckx, P. (1995). *Curve and surface fitting with splines*. Oxford University Press.
- [33] Ding, X., Liao, K., Birrer, S., Shajib, A. J., Treu, T., and Yang, L. (2021). Improved time-delay lens modelling and h 0 inference with transient sources. *Monthly Notices of the Royal Astronomical Society*, 504(4):5621–5628.
- [34] Dobler, G., Fassnacht, C. D., Treu, T., Marshall, P., Liao, K., Hojjati, A., Linder, E., and Rumbaugh, N. (2015). Strong lens time delay challenge. i. experimental design. *The Astrophysical Journal*, 799(2):168.
- [35] Dodelson, S. (2003). *Modern cosmology*. Academic Press, San Diego, CA.
- [36] Dodge, Y. (2008). *The concise encyclopedia of statistics*. Springer Science & Business Media.
- [37] Dong, X. L. and Srivastava, D. (2013). Big data integration. In *2013 IEEE 29th international conference on data engineering (ICDE)*, pages 1245–1248. IEEE.
- [38] Edelson, R., Gelbord, J., Cackett, E., Peterson, B., Horne, K., Barth, A., Starkey, D., Bentz, M., Brandt, W., Goad, M., et al. (2019). The first swift intensive agn accretion disk reverberation mapping survey. *The Astrophysical Journal*, 870(2):123.
- [39] Edelson, R. and Krolik, J. (1988). The discrete correlation function-a new method for analyzing unevenly sampled variability data. *The Astrophysical Journal*, 333:646–659.

- [40] Eigenbrod, A. (2011). *Gravitational Lensing of Quasars*. CRC Press.
- [41] Eigenbrod, A., Courbin, F., Sluse, D., Meylan, G., and Agol, E. (2008). Microlensing variability in the gravitationally lensed quasar qso 2237+ 0305≡ the einstein cross-i. spectrophotometric monitoring with the vlt. *Astronomy & Astrophysics*, 480(3):647–661.
- [42] Farhangi, N. and Ghofrani, S. (2018). Using bayesshrink, bishrink, weighted bayesshrink, and weighted bishrink in nsst and swt for despeckling sar images. *EURASIP Journal on Image and Video Processing*, 2018(1):1–18.
- [43] Feigelson, E. D., Babu, G. J., and Caceres, G. A. (2018). Autoregressive times series methods for time domain astronomy. *Frontiers in Physics*, 6:80.
- [44] Feng, T., Du, Z., Sun, Y., Wei, J., Bi, J., and Liu, J. (2017). Real-time anomaly detection of short-time-scale gwac survey light curves. In *2017 IEEE International Congress on Big Data (BigData Congress)*, pages 224–231. IEEE.
- [45] García, S., Luengo, J., and Herrera, F. (2015). *Data preprocessing in data mining*. Springer.
- [46] Gaskell, C. M. and Peterson, B. M. (1987). The accuracy of cross-correlation estimates of quasar emission-line region sizes. *The Astrophysical Journal Supplement Series*, 65:1–11.
- [47] Gaskell, C. M. and Sparke, L. S. (1986). Line variations in quasars and seyfert galaxies. *The Astrophysical Journal*, 305:175–186.
- [48] Giles, D. E. (2007). Spurious regressions with time-series data: further asymptotic results. *Communications in statistics—Theory and methods*, 36(5):967–979.
- [49] Goicoechea, L. J. and Shalyapin, V. N. (2016). Gravitational lens system sdss j1339+ 1310: microlensing factory and time delay. *Astronomy & Astrophysics*, 596:A77.
- [50] Gregory, D., Fassnacht, C. D., Treu, T., Marshall, P., Liao, K., Hojjati, A., Linder, E., and Rumbaugh, N. (2015). Strong lens time delay challenge. i. experimental design. *The Astrophysical Journal*, 799(2).
- [51] Grillo, C., Rosati, P., Suyu, S., Balestra, I., Caminha, G., Halkola, A., Kelly, P., Lombardi, M., Mercurio, A., Rodney, S., et al. (2018). Measuring the value of the hubble constant “à la refsdal”. *The Astrophysical Journal*, 860(2):94.
- [52] Gujarati, D. N. (2009). *Basic econometrics*. Tata McGraw-Hill Education.
- [53] Gürkan, G., Jackson, N., Browne, I., Koopmans, L., Fassnacht, C., and Alba, A. B. (2010). A method for efficient measurement of gravitational lens time delays. *arXiv preprint arXiv:1011.6604*.

- [54] H. Suyu, S., W. Auger, M., Hilbert, S., Marshall, P., Tewes, M., Treu, T., Fassnacht, C., V. E. Koopmans, L., Sluse, D., D. Blandford, R., Courbin, F., and Meylan, G. (2012). Two accurate time-delay distances from strong lensing: Implications for cosmology. *The Astrophysical Journal*, 766.
- [55] Hanif, A. and Protopapas, P. (2015). Recursive bayesian estimation of regularized and irregular quasar light curves. *Monthly Notices of the Royal Astronomical Society*, 448(1):390–402.
- [56] Hanke, J. and Wichern, D. (2009). *Business Forecasting*. Pearson International Edition. Pearson/Prentice Hall.
- [57] Hodrick, R. J. and Prescott, E. C. (1997). Postwar us business cycles: an empirical investigation. *Journal of Money, credit, and Banking*, pages 1–16.
- [58] Hojjati, A., Kim, A. G., and Linder, E. V. (2013). Robust strong lensing time delay estimation. *Physical Review D*, 87(12):123512.
- [59] Hojjati, A. and Linder, E. V. (2014). Next generation strong lensing time delay estimation with gaussian processes. *Phys. Rev. D*, 90:123501.
- [60] Holt, C. C. (1957). Forecasting trends and seasonals by exponentially weighted averages. carnegie institute of technology. *Pittsburgh ONR memorandum*.
- [61] Hossain, Z., Rahman, A., Hossain, M., and Karami, J. H. (2019). Over-differencing and forecasting with non-stationary time series data. *Dhaka University Journal of Science*, 67(1):21–26.
- [62] Howard, S. (2011). Cosmic distance ladder. *Journal of the Washington Academy of Sciences*, pages 47–64.
- [63] Huang, N. E., Shen, Z., Long, S. R., Wu, M. C., Shih, H. H., Zheng, Q., Yen, N.-C., Tung, C. C., and Liu, H. H. (1998). The empirical mode decomposition and the hilbert spectrum for nonlinear and non-stationary time series analysis. *Proceedings of the Royal Society of London. Series A: mathematical, physical and engineering sciences*, 454(1971):903–995.
- [64] Hyndman, R. J. (2011). *Moving Averages*, pages 866–869. Springer Berlin Heidelberg, Berlin, Heidelberg.
- [65] Icaza-Lizaola, M., Bower, R. G., Norberg, P., Cole, S., Schaller, M., and Egan, S. (2021). A sparse regression approach to modelling the relation between galaxy stellar masses and their host haloes. *Monthly Notices of the Royal Astronomical Society*, 507(3):4584–4602.
- [66] Kim, D.-W., Protopapas, P., Alcock, C., Byun, Y.-I., and Bianco, F. B. (2009). De-trending time series for astronomical variability surveys. *Monthly Notices of the Royal Astronomical Society*, 397(1):558–568.

- [67] King, W. (1912). *The Elements of Statistical Method*. Macmillan.
- [68] Kleinow, T. (2002). *Testing continuous time models in financial markets*. PhD thesis, Humboldt-Universität zu Berlin, Wirtschaftswissenschaftliche Fakultät.
- [69] Kovačević, A., Popović, L., Shapovalova, A., Ilić, D., Burenkov, A., and Chavushyan, V. (2014). Time series analysis of active galactic nuclei: The case of arp 102b, 3c 390.3, ngc 5548 and ngc 4051. *Advances in Space Research*, 54(7):1414–1428.
- [70] Kumar, S. R., Tewes, M., Stalin, C., Courbin, F., Asfandiyarov, I., Meylan, G., Eulaers, E., Prabhu, T., Magain, P., Van Winckel, H., et al. (2013). Cosmograil: the cosmological monitoring of gravitational lenses-xiv. time delay of the doubly lensed quasar sdss j1001+5027. *Astronomy & Astrophysics*, 557:A44.
- [71] Larose, D. T. (2005). *Discovering knowledge in data: an introduction to data mining*. John Wiley & Sons.
- [72] Lee, G. R., Gommers, R., Wasilewski, F., Wohlfahrt, K., and O’Leary, A. (2019). Pywavelets: A python package for wavelet analysis. *J. Open Source Softw.*, 4:1237.
- [73] Lehar, J., Hewitt, J., Burke, B., and Roberts, D. (1992). The radio time delay in the double quasar 0957+ 561. *The Astrophysical Journal*, 384:453–466.
- [74] Liao, K., Treu, T., Marshall, P., Fassnacht, C. D., Rumbaugh, N., Dobler, G., Aghamousa, A., Bonvin, V., Courbin, F., Hojjati, A., et al. (2015). Strong lens time delay challenge. ii. results of tdc1. *The Astrophysical Journal*, 800(1):11.
- [75] Makridakis, S., Andersen, A., Carbone, R., Fildes, R., Hibon, M., Lewandowski, R., Newton, J., Parzen, E., and Winkler, R. (1982). The accuracy of extrapolation (time series) methods: Results of a forecasting competition. *Journal of forecasting*, 1(2):111–153.
- [76] Makridakis, S., Wheelwright, S., and Hyndman, R. J. (1998). *Forecasting: methods and applications*. John Wiley & Sons.
- [77] Makridakis, S., Wheelwright, S. C., and Hyndman, R. J. (1997). *Forecasting methods and applications*. John wiley & sons.
- [78] Mandelbrot, B. B. and Wallis, J. R. (1969). Computer experiments with fractional gaussian noises: Part 1, averages and variances. *Water resources research*, 5(1):228–241.
- [79] Maran, S. P. (2017). *Astronomy for Dummies*. John Wiley & Sons.
- [80] Masters, K. L. and Team, G. Z. (2019). Twelve years of galaxy zoo. *Proceedings of the International Astronomical Union*, 14(S353):205–212.

- [81] Molinari, N., Durand, J.-F., and Sabatier, R. (2004). Bounded optimal knots for regression splines. *Computational Statistics and Data Analysis*, 45(2):159–178.
- [82] Na, X., Wang, N., Yuan, J., Liu, Z., Esamdin, A., Pan, J., and Xu, R. (2011). Hurst parameter analysis of radio pulsar timing residuals. *Monthly Notices of the Royal Astronomical Society*, 412(4):2678–2684.
- [83] Oliphant, T. E. (2007). Python for scientific computing. *Computing in Science and Engineering*, 9(3):10–20.
- [84] Ovaldsen, J.-E., Teuber, J., Schild, R., and Stabell, R. (2003). New aperture photometry of qso 0957+ 561; application to time delay and microlensing. *Astronomy & Astrophysics*, 402(3):891–904.
- [85] Pelt, J., Kayser, R., Refsdal, S., and Schramm, T. (1995). The light curve and the time delay of qso 0957+ 561. *Astronomy and Astrophysics*.
- [86] Pelt, J., Kayser, R., Schild, R., and Thomson, D. (1996). The time delay between qso 0957+ 561 a, b. In *Symposium-International Astronomical Union*, volume 168, pages 539–540. Cambridge University Press.
- [87] Pelt, J., Schild, R., Refsdal, S., and Stabell, R. (1998). Microlensing on different timescales in the lightcurves of qso 0957+ 561 a, b. *Astronomy and Astrophysics*, 336:829–839.
- [88] Persic, M. and Salucci, P. (1992). The baryon content of the universe. *Monthly Notices of the Royal Astronomical Society*, 258(1):14P–18P.
- [89] Phillips, P. C. and Perron, P. (1988). Testing for a unit root in time series regression. *Biometrika*, 75(2):335–346.
- [90] Polygiannakis, J., Preka-Papadema, P., and Moussas, X. (2003). On signal–noise decomposition of time-series using the continuous wavelet transform: application to sunspot index. *Monthly Notices of the Royal Astronomical Society*, 343(3):725–734.
- [91] Press, W. H., Rybicki, G. B., and Hewitt, J. N. (1992). The time delay of gravitational lens 0957+ 561. ii. analysis of radio data and combined optical-radio analysis. *The Astrophysical Journal*, 385:416.
- [92] Qian, B. and Rasheed, K. (2004). Hurst exponent and financial market predictability. In *IASTED conference on Financial Engineering and Applications*, pages 203–209. Proceedings of the IASTED International Conference Cambridge, MA.
- [93] Rathna Kumar, S. (2017). Crash testing difference-smoothing algorithm on a large sample of simulated light curves from tdc1. *Monthly Notices of the Royal Astronomical Society*, 470(3):2918–2928.

- [94] Rathna Kumar, S., Stalin, C. S., and Prabhu, T. P. (2015). H0 from ten well-measured time delay lenses. *A&A*, 580:A38.
- [95] Refsdal, S. (1964a). On the possibility of determining hubble’s parameter and the masses of galaxies from the gravitational lens effect. *Monthly Notices of the Royal Astronomical Society*, 128(4):307–310.
- [96] Refsdal, S. (1964b). On the possibility of determining hubble’s parameter and the masses of galaxies from the gravitational lens effect. *Monthly Notices of the Royal Astronomical Society*, 128(4):307–310.
- [97] Riess, A. (2018). A new measurement of the expansion rate of the universe, evidence of new physics? In *American Astronomical Society Meeting Abstracts# 231*, volume 231, pages 117–01.
- [98] Riess, A. G., Casertano, S., Yuan, W., Bowers, J. B., Macri, L., Zinn, J. C., and Scolnic, D. (2021). Cosmic distances calibrated to 1% precision with gaiaedr3 parallaxes and hubble space telescope photometry of 75 milky way cepheids confirm tension with λ cdm. *The Astrophysical Journal Letters*, 908(1):L6.
- [99] Sammut, C. and Webb, G. I., editors (2010). *Mean Squared Error*, pages 653–653. Springer US, Boston, MA.
- [100] Sarkar, A., Gupta, A. C., Chitnis, V. R., and Wiita, P. J. (2021). Multiwaveband quasi-periodic oscillation in the blazar 3c 454.3. *Monthly Notices of the Royal Astronomical Society*, 501(1):50–61.
- [101] Shafieloo, A., Alam, U., Sahni, V., and Starobinsky, A. A. (2006). Smoothing supernova data to reconstruct the expansion history of the universe and its age. *Monthly Notices of the Royal Astronomical Society*, 366(3):1081–1095.
- [102] Shah, C. (2020). *A hands-on introduction to data science*. Cambridge University Press.
- [103] Skiena, S. S. (2017). *The data science design manual*. Springer.
- [104] Stigler, S. M. (1973). Studies in the history of probability and statistics. xxxii: Laplace, fisher, and the discovery of the concept of sufficiency. *Biometrika*, 60(3):439–445.
- [105] Tan, P.-N., Steinbach, M., and Kumar, V. (2013). Data mining cluster analysis: basic concepts and algorithms. *Introduction to data mining*, 487:533.
- [106] Tandberg, D. and Qualls, C. (1994). Time series forecasts of emergency department patient volume, length of stay, and acuity. *Annals of emergency medicine*, 23(2):299–306.
- [107] Tarvainen, M. P., Ranta-Aho, P. O., and Karjalainen, P. A. (2002). An advanced detrending method with application to hrv analysis. *IEEE transactions on biomedical engineering*, 49(2):172–175.

- [108] Tewes, M., Courbin, F., and Meylan, G. (2013). Cosmograil: the cosmological monitoring of gravitational lenses-xi. techniques for time delay measurement in presence of microlensing. *Astronomy & Astrophysics*, 553:A120.
- [109] Tian, C., Ma, J., Zhang, C., and Zhan, P. (2018). A deep neural network model for short-term load forecast based on long short-term memory network and convolutional neural network. *Energies*, 11(12):3493.
- [110] Tsay, R. S. (2014). *An introduction to analysis of financial data with R*. John Wiley & Sons.
- [111] Tseng, F.-M., Yu, H.-C., and Tzeng, G.-H. (2002). Combining neural network model with seasonal time series arima model. *Technological forecasting and social change*, 69(1):71–87.
- [112] Tsvetkova, V., Shulga, V., and Berdina, L. (2016). A simple method to determine time delays in the presence of microlensing: application to he 0435–1112 and pg 1115+ 080. *Monthly Notices of the Royal Astronomical Society*, 461(4):3714–3723.
- [113] Tyson, N. d. (2017). *Astrophysics for People in a Hurry*. WW Norton & Company.
- [114] Tyson, N. d., Tyson, N. D., and Goldsmith, D. (2004). *Origins: Fourteen billion years of cosmic evolution*. WW Norton & Company.
- [115] Valentino, E. D., Mena, O., Pan, S., Visinelli, L., Yang, W., Melchiorri, A., Mota, D. F., Riess, A. G., and Silk, J. (2021). In the realm of the hubble tension—a review of solutions*. *Classical and Quantum Gravity*, 38(15):153001.
- [116] Van der Walt, S., Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., Goullart, E., and Yu, T. (2014). scikit-image: image processing in python. *PeerJ*, 2:e453.
- [117] Willmott, C. J. (1981). On the validation of models. *Physical geography*, 2(2):184–194.
- [118] Winters, P. R. (1960). Forecasting sales by exponentially weighted moving averages. *Management science*, 6(3):324–342.
- [119] Witten, I. H., Frank, E., Hall, M. A., and Pal, C. J. (2017). *Data mining: practical machine learning tools and techniques*. Morgan Kaufmann.
- [120] Wu, P.-X., Li, Z.-X., and Yu, H.-W. (2017). Determining h_0 using a model-independent method. *Frontiers of Physics*, 12(1):1–8.
- [121] Xiu, J. and Jin, Y. (2007). Empirical study of arfima model based on fractional differencing. *Physica A: Statistical Mechanics and its Applications*, 377(1):138–154.

- [122] Xu, M., Han, M., and Lin, H. (2018). Wavelet-denoising multiple echo state networks for multivariate time series prediction. *Information Sciences*, 465:439–458.
- [123] Zaki, M. J. and Meira Jr, W. (2020). *Data mining and machine learning: Fundamental concepts and algorithms*. Cambridge University Press.
- [124] Zhang, Y. and Zhao, Y. (2015). Astronomy in the big data era. *Data Science Journal*, 14.
- [125] Zwicky, F. (1937). Nebulae as gravitational lenses. *Physical Review*, 51:290–290.

Appendix A

A.1 Time delay techniques code

In this section we present the code in Python for the Discrete Correlation Function (DCF), Locally Normalised Discrete Correlation Function (LND CF) and the Dispersion Spectra (DS_1^2) methods.

A.1.1 Cross-correlation code

Main function for the DCF and LND CF algorithms

```
import numpy as np
import os
import substring
import csv

def dcf(ts1, ts2, t, td):

    dcf = np.zeros(t.shape[0])
    dcfErr = np.zeros(t.shape[0])
    num = np.zeros(t.shape[0])

    d_st = np.empty((ts1.shape[0], ts2.shape[0]))
    for i in range(ts1.shape[0]):
        for j in range(ts2.shape[0]):
            d_st[i,j] = ts2[j,0] - ts1[i,0]

    for k in range(t.shape[0]):
        t_lo = t[k] - td/2.0
        t_hi = t[k] + td/2.0
        ts1index, ts2index = np.where((d_st < t_hi) & (d_st > t_lo))
        #Constructing the DCF formula
        avgts1 = np.mean(ts1[:,1])
        avgts2 = np.mean(ts2[:,1])
        num[k] = ts1index.shape[0]
```

```

dcfdominator = np.sqrt((np.var(ts1[:,1]) - np.mean(ts1[:,2])**2)) \
    * (np.var(ts2[:,1]) - np.mean(ts2[:,2])**2))
dcfnumerator = (ts1[ts1index, 1] - avgts1)* (ts2[ts2index, 1] - avgts2) /
    ↪ dcfdenominator

dcf[k] = np.sum(dcfnumerator) / float(n[k])
dcfErr[k] = np.sqrt(np.sum((dcfnumerator - dcf[k])**2)) / float(n[k] - 1)

return dcf, dcfErr

# use this line of code in the LND CF with local variables

dcfdominator = np.sqrt((np.var(ts1[ts1index,1]) - np.mean(ts1[ts1index,2])**2)
    ↪ \
        * (np.var(ts2[ts2index,1]) - np.mean(ts2[ts2index,2])**2))

```

A.1.2 Dispersion Spectra code

```

#Dispersion Spectra algorithm, two input variables in .csv format light curve A
    ↪ [time,lc_A,lcA_Err] and light curve B [time,lc_B,lcB_Err]
import pandas as pd
import substring
import csv
import numpy as np
import os
from re import compile, split
dre = compile(r'(\d+)')

delta_min = 0
delta_max = 0
inc = 1

all_diffflightcurves = os.listdir('rung4/rung4WavDns_csv/') # Trough this sentence we
    ↪ collect the name of files contained in the dir rung..
all_diffflightcurves.sort(key=lambda l: [int(s) if s.isdigit() else s.lower() for s in
    ↪ split(dre, 1)])
list_length = len(all_diffflightcurves)
list_length2 = int(list_length / 2)
df_truthrung = pd.read_csv("rung4/rung4truthpair/rung4truthpairABS.txt", delimiter=
    ↪ '\s+', header=None)
#M file is an external offset between two light curves, however when normalizing data we
    ↪ think is not necessary the use if this parameter
M = pd.read_csv("christianoFitzgerald/M_parameter/M_listR4pairsMeanSubsWavDns.csv",
    ↪ delimiter= '\s+', header=None)# value for offset/ratio
M_np = M.to_numpy()

print(all_diffflightcurves) # shows all files in a list

```

```

dsFinalList = pd.DataFrame()

z = 0
x = 1
M_count = 0

for j in range(list_length2):
    fuente1 = str(all_diffflightcurves[z]) # getting the name of the first file
    fuente2 = str(all_diffflightcurves[x]) # getting the name of the second file
    cadena1 = str(all_diffflightcurves[z])
    namelightcurve = substring.substringByChar(fuente1, startChar="r", endChar=".")
    lcName = namelightcurve[:-1]
    trueDT = float(df_truthrunc.iloc[j])
    print(str(trueDT))
    list_DS = []
    lcB_pctErr = pd.DataFrame(columns=['lcB_Err', 'B'])
    tope = 0
    bandera20 = 0
    bandera40 = 0
    bandera60 = 0
    bandera80 = 0
    bandera100 = 0
    bandera120 = 0
    print(fuente1)
    print(fuente2)
    if trueDT <= 20.00:
        delta_min = 2
        delta_max = 15
        bandera20 = 1
        tope = 48
    elif 20.00 < trueDT <= 40.00:
        delta_min = 2
        delta_max = 40
        bandera40 = 1
        tope = 84
    elif 40.00 < trueDT <= 60.00:
        delta_min = 2
        delta_max = 60
        bandera60 = 1
        tope = 99
    elif 60.00 < trueDT <= 80.00:
        delta_min = 2
        delta_max = 85
        bandera80 = 1
        tope = 81
    elif 80.00 < trueDT <= 100.00:
        delta_min = 2
        delta_max = 85
        bandera100 = 1
        tope = 45
    elif 100.00 < trueDT <= 120.00:

```

```

delta_min = 2
delta_max = 120
bandera120 = 1
tope = 25
print(str(tope))

for l in np.arange(0, tope):

    for dd in np.arange(0, 1, 1):
        dfA = pd.read_csv('rung4/rung4WavDnsPairs_csv/' + fuente1, header=None)
        dfB = pd.read_csv('rung4/rung4WavDnsPairs_csv/' + fuente2, header=None)
        dfA.columns = ['t', 'a', 'std_a']
        dfB.columns = ['t', 'b', 'std_b']
        dfTime = dfB.filter(['t'])
        time = dfA.loc[:, 't'] # the time
        firstImage = dfA.loc[:, 'a'] # first image
        secondImage = dfB.loc[:, 'b'] # second image
        error_lca = dfA.loc[:, 'std_a'] # quoted errors for first image
        error_lcb = dfB.loc[:, 'std_b'] # quoted errors for second image
        std_a = pd.Series(error_lca)
        sumtotalwXab = []
        sumWF = []

    for delta in np.arange(delta_min, delta_max + 1, inc): #
        ↪ delta_min:10:delta_max,
        t = pd.Series(time)
        a = pd.Series(firstImage)
        firstFrame = {'t': t, 'a': a}
        firstSignal = pd.DataFrame(firstFrame)
        firstSignal.insert(2, 'ones', 1)

        lcB_pctErr['lcB_Err'] = dfB['std_b'] / dfB['b']
        secondImage = dfB['b'] + M_np[M_count]
        secondSignal = pd.DataFrame()
        secondSignal['t'] = firstSignal['t'] - delta
        secondSignal['a'] = secondImage
        secondSignal.insert(2, 'ones', 0)
        combinedSignals = firstSignal.append(secondSignal, sort=False,
        ↪ ignore_index=True)
        combinedSignals = combinedSignals.sort_values('t')
        arraycombinedSignals = combinedSignals.to_numpy() # convierto en arreglo
        ↪ numpy para iterar el a_b
        combinedSignals_Tail = combinedSignals.tail(-1)
        combinedSignals_Head = combinedSignals.head(-1)
        combinedSignals_Tail.reset_index(drop=True)
        combinedSignals_Head.reset_index(drop=True)
        combinedSignals_Tail.index =
        ↪ pd.RangeIndex(len(combinedSignals_Tail.index))
        combinedSignals_Head.index =
        ↪ pd.RangeIndex(len(combinedSignals_Head.index))
        firstFrameError = {'t': t, 'std_ab': std_a}

```

```

firstSignalError = pd.DataFrame(firstFrameError)
secondSignalError = pd.DataFrame()
secondSignalError['t'] = firstSignal['t'] - delta
secondSignalError['std_ab'] = secondSignal['a'] * lcB_pctErr['lcB_Err']
combinedErrors = firstSignalError.append(secondSignalError, sort=False,
    ↪ ignore_index=True)
combinedErrors = combinedErrors.sort_values(['t', 'std_ab'])
combinedErrorsF = combinedErrors['std_ab']
combinedErrorsF_Tail = combinedErrorsF.tail(-1)
combinedErrorsF_Head = combinedErrorsF.head(-1)
combinedErrorsF_Tail.index =
    ↪ pd.RangeIndex(len(combinedErrorsF_Tail.index))
combinedErrorsF_Head.index =
    ↪ pd.RangeIndex(len(combinedErrorsF_Head.index))
Wdf = 1 / (combinedErrorsF_Tail ** 2) + (combinedErrorsF_Head ** 2)
arraycombinedErrorsF = combinedErrorsF.to_numpy()
combinedSumSignal = pd.DataFrame()
combinedSumSignal['t'] = combinedSignals_Tail['t']
combinedSumSignal['a1'] = combinedSignals_Head['a']
combinedSumSignal['a2'] = combinedSignals_Tail['a']
combinedSumSignal['aF'] = np.power(combinedSignals_Tail['a'] -
    ↪ combinedSignals_Head['a'], 2)
combinedSumSignal['W1'] = combinedErrorsF_Head
combinedSumSignal['W2'] = combinedErrorsF_Tail
combinedSumSignal['W3'] = 1 / (np.power(combinedErrorsF_Tail, 2) +
    ↪ np.power(combinedErrorsF_Head, 2))
combinedSumSignal['W*aF'] = (1 / (np.power(combinedErrorsF_Tail, 2) +
    ↪ np.power(combinedErrorsF_Head, 2))) *
    ↪ (np.power(combinedSignals_Tail['a'] - combinedSignals_Head['a'], 2))
totalwXab = combinedSumSignal['W*aF'].sum()
sumtotalwXab.append(totalwXab)
sumW = combinedSumSignal['W3'].sum()
sumWF.append(sumW)

dfsumWF = pd.DataFrame(sumWF, columns=['sumW'])
dfsumtotalwXab = pd.DataFrame(sumtotalwXab, columns=['sumD'])
dfsumWF['sumD'] = dfsumtotalwXab['sumD']
dfsumWF['sumF'] = dfsumWF['sumD'] / 2 * dfsumWF['sumW']
minIndex = dfsumWF['sumF'].idxmin()
dsEDelay = delta_min + minIndex * inc
list_DS.append([dsEDelay, delta_min, delta_max, inc, trueDT,
    ↪ abs(abs(dsEDelay) - abs(trueDT)), lcName])

print(str(delta_max))
if bandera20 == 1:
    print("entro al if bandera20")
    if delta_min == 15 and delta_max == 100: # 48
        delta_min = 20
        delta_max = 35
        continue
    elif delta_min == 15 and delta_max == 95: #47

```

```

        delta_min = 15
        delta_max = 100
        continue
    ....
    ....
    elif delta_min == 2 and delta_max == 20: # 2
        delta_min = 2
        delta_max = 25
        continue
    elif delta_min == 2 and delta_max == 15: # 1
        delta_min = 2
        delta_max = 20
#####
    if bandera40 == 1:
        print("entro al if bandera40")
        if delta_min == 30 and delta_max == 105: # 84
            delta_min = 30
            delta_max = 110
            continue
        elif delta_min == 30 and delta_max == 100: # 83
            delta_min = 30
            delta_max = 105
            continue
        ....
        ....
        elif delta_min == 2 and delta_max == 40: #2
            delta_min = 2
            delta_max = 45
            continue
        elif delta_min == 2 and delta_max == 35: #1
            delta_min = 2
            delta_max = 40
#####
    if bandera60 == 1:
        print("entro al if bandera60")
        if delta_min == 50 and delta_max == 115: # 99
            delta_min = 50
            delta_max = 120
            continue
        elif delta_min == 50 and delta_max == 110: # 98
            delta_min = 50
            delta_max = 115
            continue
        ....
        ....
        elif delta_min == 2 and delta_max == 65: #2
            delta_min = 2
            delta_max = 70
            continue
        elif delta_min == 2 and delta_max == 60: #1
            delta_min = 2

```

```

        delta_max = 65
#####
    if bandera80 == 1:
        print("entro al if bandera80")
        if delta_min == 70 and delta_max == 115: # 78
            delta_min = 70
            delta_max = 120
            continue
        elif delta_min == 70 and delta_max == 110: # 77
            delta_min = 70
            delta_max = 115
            continue
        ....
        ....
        elif delta_min == 2 and delta_max == 90: # 2
            delta_min = 2
            delta_max = 95
            continue
        elif delta_min == 2 and delta_max == 85: # 1
            delta_min = 2
            delta_max = 90
#####
    if bandera100 == 1:
        print("entro al if bandera100")
        if delta_min == 90 and delta_max == 115: # 41
            delta_min = 90
            delta_max = 120
            continue
        elif delta_min == 90 and delta_max == 110: # 40
            delta_min = 90
            delta_max = 115
            continue
        ....
        ....
        elif delta_min == 2 and delta_max == 90: # 2
            delta_min = 2
            delta_max = 95
            continue
        elif delta_min == 2 and delta_max == 85: # 1
            delta_min = 2
            delta_max = 90
#####
    if bandera120 == 1:
        print("entro al if bandera120")
        if delta_min == 105 and delta_max == 120: # 24
            delta_min = 105
            delta_max = 125
            continue
        elif delta_min == 100 and delta_max == 120: # 23
            delta_min = 105
            delta_max = 120

```



```

        continue
    ....
    ....
    elif delta_min == 10 and delta_max == 120: # 2
        delta_min = 20
        delta_max = 120
        continue
    elif delta_min == 2 and delta_max == 120: # 1
        delta_min = 10
        delta_max = 120
    #####
z = z + 2
x = x + 2
M_count = M_count + 1
dflist_DS = pd.DataFrame(list_DS, columns=['dsEDelay', 'delta_min', 'delta_max',
    ↪ 'increment', 'trudelay', 'error', 'nameLC'])
dflist_DS.to_csv("WavDns/rung4pairsWavDnsResults/DSv1WavDs_R4pairs_" + fuente1,
    ↪ index=True, header=True)
minimumLastIdx = dflist_DS['error'].idxmin()
dsFinalList = dsFinalList.append(dflist_DS.loc[minimumLastIdx], ignore_index=False)

DFdsFinalList = pd.DataFrame(dsFinalList, columns=['dsEDelay', 'delta_min', 'delta_max',
    ↪ 'increment', 'trudelay', 'error', 'nameLC'])
dsFinalList.to_csv("WavDns/rung4pairsWavDnsResults/DSv1WavDns_R4pairsList1.csv",
    ↪ index=True, header=True)

```

A.2 Data transformation techniques code

In this section is presented the four transformation techniques applied in this research in Python code.

A.2.1 Wavelet Denoising code

In first step we use the pandas library to load the raw data of TDC1 from a file with a dt file extension very similar to a *txt* file extension, this file is transformed into a DataFrame object. Also, in the same Python code snippet, we calculate the %percentage error for both light curves. The last line of code shows us the first five lines of the DataFrame through the head method, and is used to verify that raw data and its header were loaded correctly:

```
# In first step we use the pandas library to load the raw data of TDC1 from a file with a
↪ dt file extension very similar to a $txt$ file extension, this file is transformed
↪ into a DataFrame object. Also, in the same Python code snippet, we calculate a (%)
↪ percentage error or fraction of error for both light curves. The last line of code
↪ shows us the first five lines of the DataFrame through the head method, and is used
↪ to verify that the raw data and its header were loaded correctly:
```

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import statsmodels.api as sm
from skimage.restoration import (denoise_wavelet)
```

```
df = pd.read_csv('tdc1_rung3_double_pair434.txt', skiprows=[0,1, 2, 3, 4, 5 ],
↪ header=None, delim_whitespace=True, names=('time','lc_A','err_A','lc_B','err_B'))
#percentage error for both light curves
df['pctErr_A'] = df['err_A'] / df['lc_A']
df['pctErr_B'] = df['err_B'] / df['lc_B']
#display first five lines of the DataFrame
df.head()
```

```
# The final task is to remove the noise of both light curves and calculate his level of
↪ noise of the preprocessed data. The noise is removed with the method denoise_wavelet
↪ applying the VisuShrink approach.
```

```
x= df['time'].to_numpy()
x = np.reshape(x, (len(x), 1)) # ensure the index is in only one dimension
```

```
df_lcA_NP = df['lc_A'].to_numpy()
df_lcA_NP.flatten()
```

```
df_lcB_NP = df['lc_B'].to_numpy()
df_lcB_NP.flatten()
```

```

y_denoise = denoise_wavelet(df['lc_A'], method='VisuShrink', mode='soft',
    ↪ wavelet_levels=1, wavelet='coif17', rescale_sigma=True)
y1_denoise = denoise_wavelet(df['lc_B'], method='VisuShrink', mode='soft',
    ↪ wavelet_levels=1, wavelet='coif17', rescale_sigma=True)

dfWavDenoise_lcA = pd.DataFrame(y_denoise)
df['dfWavDenoise_lcA'] = dfWavDenoise_lcA
df['lcA_WavErr'] = df['dfWavDenoise_lcA'] * df['pctErr_A']

dfWavDenoise_lcB = pd.DataFrame(y1_denoise)
df['dfWavDenoise_lcB'] = dfWavDenoise_lcB
df['lcB_WavErr'] = df['dfWavDenoise_lcB'] * df['pctErr_B']
denoise_lcBerr = df['lcB_WavErr'].to_numpy()
"""
The different methods for delay estimation have an input of two files
lcA[time,light curve A,lcA_Error] and lcB[time,light curve B,lcB_Error]
"""
#We save light curve A in cycle and trend in two separate files
WDlcA_ts1 = df.iloc[0:,[0,7,8]].astype(float)
WDlcA_ts1.to_csv('WDlcA_ts1_r3_pair434.csv', index=False, header=False)
#We save light curve B in cycle and trend in two separate files
WDlcB_ts2 = df.iloc[0:,[0,9,10]].astype(float)
WDlcB_ts2.to_csv('WDlcB_ts2_r3_pair434.csv', index=False, header=False)

```

A.2.2 Data differencing

```

# In first step we use the pandas library to load the raw data of TDC1 from a file with a
↪ dt file extension very similar to a $txt$ file extension, this file is transformed
↪ into a DataFrame object. Also, in the same Python code snippet, we calculate a (%)
↪ percentage error or fraction of error for both light curves. The last line of code
↪ shows us the first five lines of the DataFrame through the head method, and is used
↪ to verify that the raw data and its header were loaded correctly:
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import statsmodels.api as sm
from skimage.restoration import (denoise_wavelet)

df = pd.read_csv('tdc1_rung3_double_pair434.txt', skiprows=[0,1, 2, 3, 4, 5 ],
↪ header=None, delim_whitespace=True, names=('time','lc_A','err_A','lc_B','err_B'))
#percentage error for both light curves
df['pctErr_A'] = df['err_A'] / df['lc_A']
df['pctErr_B'] = df['err_B'] / df['lc_B']
#display first five lines of the DataFrame
df.head()
# The next task consist in differencing both light curves, in a first order differencing.
df['lcA_diff'] = df['lc_A'].diff()
df['lcA_diffErr'] = df['lcA_diff'] * df['lcA_pctErr']
df['lcB_diff'] = df['lc_B'].diff()
df['lcB_diffErr'] = df['lcB_diff'] * df['lcB_pctErr']
#####

The different methods for delay estimation have an input of two files
lcA[time,light curve A,lcA_Error] and lcB[time,light curve B,lcB_Error]
#####

#We save light curve A in cycle and trend in two separate files
diff_lcA_ts1 = df.iloc[0:,[0,7,8]].astype(float)
diff_lcA_ts1.to_csv('diff_lcA_ts1_r3_pair434.csv', index=False, header=False)
#We save light curve B in cycle and trend in two separate files
diff_lcB_ts2 = df.iloc[0:,[0,9,10]].astype(float)
diff_lcB_ts2.to_csv('diff_lcB_ts2_r3_pair434.csv', index=False, header=False)

```

A.2.3 Simple Net Return code

```

import numpy as np
import pandas as pd

df = pd.read_csv('tdc1_rung3_double_pair434.txt', skiprows=[0,1, 2, 3, 4, 5 ],
↳ header=None, delim_whitespace=True, names=('time','lc_A','err_A','lc_B','err_B'))
#percentage error for both light curves
df['pctErr_A'] = df['err_A'] / df['lc_A']
df['pctErr_B'] = df['err_B'] / df['lc_B']
#display first five lines of the DataFrame
df.head()
# The task of transforming raw data into SNR and saving in two files is shown next.
df['lcA_SNR'] = df['lc_A'].pct_change()
df['lcA_SNR_Err'] = df['lcA_SNR'] * df['lcA_pctErr']
df['lcB_SNR'] = df['lc_B'].pct_change()
df['lcB_SNR_Err'] = df['lcB_SNR'] * df['lcB_pctErr']
"""
The different methods for delay estimation have an input of two files
lcA[time,light curve A,lcA_Error] and lcB[time,light curve B,lcB_Error]
"""
#We save light curve A and light curve B in two separate files
SNR_lcA_ts1 = df.iloc[0:,[0,7,8]].astype(float)
SNR_lcA_ts1.to_csv('SNR_lcA_ts1_r3_pair434.csv', index=False, header=False)
#We save light curve B in cycle and trend in two separate files
SNR_lcB_ts2 = df.iloc[0:,[0,9,10]].astype(float)
SNR_lcB_ts2.to_csv('SNR_lcB_ts2_r3_pair434.csv', index=False, header=False)

```

A.2.4 Christiano fitzgerald filter

```

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import statsmodels.api as sm

df = pd.read_csv('tdc1_rung4_double_pair21.txt', skiprows=[0,1, 2, 3, 4, 5 ],
    ↪ header=None, delim_whitespace=True, names=('time','lc_A','err_A','lc_B','err_B'))

df['pctErr_A'] = df['err_A'] / df['lc_A']
df['pctErr_B'] = df['err_B'] / df['lc_B']
df['err_A'] = df['lc_A'] * df['pctErr_A']
df['err_B'] = df['lc_B'] * df['pctErr_B']

x= df['time'].to_numpy()
x = np.reshape(x, (len(x), 1)) # ensure the index is in only one dimension

df_lcA_np = df['lc_A'].to_numpy()
df_lcA_np.flatten()
df_lcB_np = df['lc_B'].to_numpy()
df_lcB_np.flatten()

cycleA, trendA = sm.tsa.filters.cfilter(df_lcA_np, 6, 32, drift=False)
cycleB, trendB = sm.tsa.filters.cfilter(df_lcB_np, 6, 32, drift=False)

dfCycle_lcA = pd.DataFrame(cycleA)
df['dfCycle_lcA'] = dfCycle_lcA
df['lcA_CF_CyErr'] = df['dfCycle_lcA'] * df['pctErr_A']
dfTrend_lcA = pd.DataFrame(trendA)
df['dfTrend_lcA'] = dfTrend_lcA
df['lcA_CF_TrErr'] = df['dfTrend_lcA'] * df['pctErr_A']
dfCycle_lcB = pd.DataFrame(cycleB)
df['dfCycle_lcB'] = dfCycle_lcB
df['lcB_CF_CyErr'] = df['dfCycle_lcB'] * df['pctErr_B']
dfTrend_lcB = pd.DataFrame(trendB)
df['dfTrend_lcB'] = dfTrend_lcB
df['lcB_CF_TrErr'] = df['dfTrend_lcB'] * df['pctErr_B']
"""
The different methods for delay estimation have an input of two files
lcA[time,light curve A,lcA_Error] and lcB[time,light curve B,lcB_Error]
"""

#We save light curve A in cycle and trend in two separate files
CFlcACycle_ts1 = df.iloc[:, [0,7,8]].astype(float)
CFlcATrend_ts1 = df.iloc[:, [0,9,10]].astype(float)
CFlcACycle_ts1.to_csv('cfCycle_ts1_lcA_r4_pair21.csv', index=False, header=False)
CFlcATrend_ts1.to_csv('cfTrend_ts1_lcA_r4_pair21.csv', index=False, header=False)
#We save light curve B in cycle and trend in two separate files
CFlcBCycle_ts2 = df.iloc[:, [0,11,12]].astype(float)
CFlcBTrend_ts2 = df.iloc[:, [0,13,14]].astype(float)

```

```
CFlcBCycle_ts2.to_csv('cfCycle_ts2_lcb_r4_pair21.csv', index=False, header=False)  
CFlcBTrend_ts2.to_csv('cfTrend_ts2_lcb_r4_pair21.csv', index=False, header=False)
```