



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
PROGRAMA DE POSGRADO EN ASTROFÍSICA

INSTITUTO DE RADIOASTRONOMÍA Y ASTROFÍSICA

OPTIMIZACIÓN DE MEDICIONES DE PARALAJE
CON VLBI

T E S I S

PARA OPTAR POR EL GRADO DE
MAESTRO EN CIENCIAS

PRESENTA

AURORA DE LOS ANGELES DURÁN MARTÍNEZ

DIRECTOR DE TESIS

DR. LAURENT LOINARD

INSTITUTO DE RADIOASTRONOMÍA Y ASTROFÍSICA

MORELIA, MICHOACÁN, MÉXICO. JUNIO 2020



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
PROGRAMA DE POSGRADO EN ASTROFÍSICA

INSTITUTO DE RADIOASTRONOMÍA Y ASTROFÍSICA

OPTIMIZATION OF VLBI PARALLAX
MEASUREMENTS

T E S I S

PARA OPTAR POR EL GRADO DE
MAESTRO EN CIENCIAS

PRESENTA

AURORA DE LOS ANGELES DURÁN MARTÍNEZ

DIRECTOR DE TESIS

DR. LAURENT LOINARD

INSTITUTO DE RADIOASTRONOMÍA Y ASTROFÍSICA

MORELIA, MICHOACÁN, MÉXICO. JUNIO 2020

Contents

Resumen	iii
Summary	vii
1 Introduction	1
1.1 Astrometry and Distance Measurements	1
1.2 Interferometry	4
1.3 Previous Observations	6
1.4 Motivation	7
1.4.1 Monte Carlo Method	8
2 Methodology	11
2.1 Trajectory	11
2.2 Epoch selection	12
2.3 Astrometric errors	13
2.4 Adjust new trajectory	15
2.5 Monte Carlo	16
3 Results	19
4 Conclusions	33
A T Tau	41
B HP Tau	47
C Ophiuchus	53
D ONC	59
E Astrometry Simulation Code	65

Resumen

La astrometría es la ciencia que se encarga de estudiar las posiciones y los movimientos de los objetos celestes; tiene muchas aplicaciones importantes. Por ejemplo, analizando los movimientos aparentes en función del tiempo de un objeto, es posible conocer su distancia. La distancia es esencial para conocer las propiedades intrínsecas (tamaño, luminosidad, masa, etc.) de un astro. Por lo tanto, la determinación de la distancia siempre ha sido muy importante en la astronomía.

Existen varios métodos para calcular la distancia a un objeto, muchos de ellos indirectos. Para objetos fuera del Sistema Solar, solo hay una técnica puramente geométrica: la determinación de la paralaje trigonométrica. La paralaje se define como *el movimiento aparente de la posición de un objeto visto desde dos puntos diferentes*. La paralaje trigonométrica es el resultado del movimiento de la Tierra alrededor del Sol a través del tiempo. Debido a este movimiento, podemos ver un desplazamiento de la fuente en el cielo con respecto a las estrellas de fondo. El movimiento aparente de la fuente traza un camino elíptico en el cielo. El eje semi-mayor de esta elipse es conocida como el ángulo de *Paralaje Trigonométrica* (ϖ). La distancia al objeto se calcula a partir de la paralaje trigonométrica (ϖ) con $d = 1/\varpi''$ pc.

La paralaje trigonométrica puede ser medida en el óptico para estrellas aisladas y brillantes. Pero esto no funciona bien para estrellas localizadas en regiones de formación estelar. Por estar embebidas en nubes de polvo, dichas estrellas jóvenes son débiles en

longitudes de onda del óptico. En contraste, las nubes de polvo son prácticamente transparentes en ondas de radio. Entonces, para observar estrellas en regiones de formación estelar las observaciones en longitudes de onda en el radio son muy complementarias de observaciones en el rango óptico.

La precisión astrométrica con un solo radiotelescopio, se limita a una fracción del límite de difracción, $\theta \approx \lambda/D$, donde λ es la longitud de onda a la que se observa, y D es el diámetro de la antena. Aún así, es poco para fines de astrometría. Esta pobre resolución, se soluciona al usar interferometría. Un interferómetro se conforma de dos o más radiotelescopios, que pueden tener separaciones que van desde metros hasta miles de kilómetros. Esta última situación se conoce como interferometría de muy larga base (VLBI, por sus siglas en inglés). La resolución angular está dada por la separación máxima, B_{max} , entre los radiotelescopios, $\theta \approx \lambda/B_{max}$.

Las observaciones de paralaje con VLBI se hacen con observaciones multi-época. Revisando la literatura, notamos que varios autores han usado diversas estrategias para programar las observaciones de paralaje, particularmente en relación a la cantidad de épocas observadas y la cadencia de éstas. Dada la diversidad de estas estrategias, decidimos escribir un código para simular observaciones de paralaje con VLBI, y así poder entender en qué difiere la precisión de la paralaje bajo diferentes opciones de cadencia y número de observaciones. Este código utiliza un método Monte Carlo (i.e. utiliza números aleatorios para resolver el problema). El algoritmo de esta simulación opera de la siguiente manera: para cualquier conjunto de valores iniciales de ascensión recta inicial (α_0), declinación inicial (δ_0), componentes de movimiento propio (μ_α y μ_δ) y paralaje (ϖ), se calcula una trayectoria teórica del objeto. De esa trayectoria, extraemos un número dado de posiciones, con una cierta cadencia, tomando en cuenta posibles variaciones en la cadencia alrededor de un valor medio (por ejemplo una variación de una semana alrededor de una cadencia media de una observación cada tres meses). A

esas posiciones extraídas, les añadimos errores astrométricos realistas tanto en ascensión recta (α), como en declinación (δ). Estos errores astrométricos toman en cuenta errores sobre la posición del calibrador, la orientación de la Tierra, la posición de las antenas, así como incertidumbres sobre los parámetros de la tropósfera, y se suman a las posiciones extraídas de la trayectoria teórica. Estas nuevas posiciones serán nuestras observaciones simuladas u observaciones sintéticas. A partir de estas observaciones simuladas, se hace un ajuste astrométrico que resulta de una determinación de la paralaje recuperada (ϖ^f). Como las observaciones simuladas incorporan un término de ruido astrométrico, la ϖ^f será ligeramente diferente a la ϖ que pusimos como parámetro inicial. Este algoritmo se repite un cierto número de realizaciones, para hacer un método Monte Carlo. En cada realización los parámetros de ruido serán diferentes, ya que están programados para ser valores aleatorios en cada realización. Al terminar las realizaciones, tendremos tantos valores de ϖ^f como realizaciones hechas (ϖ_i^f). Estos ϖ_i^f son arreglados en un histograma, al que se le ajusta una Gaussiana para calcular el promedio de los valores de ϖ_i^f así como la desviación estándar (σ) de alrededor del promedio. El valor promedio obtenido de ϖ_i^f , es muy similar al ϖ de los parámetros iniciales. El valor obtenido para σ proporciona una estimación del error en la determinación del valor de la paralaje, para la estrategia (número y cadencia de observaciones) implementada en dicha simulación. Al repetir simulaciones para diferentes estrategias, podemos comparar las diferentes estrategias entre sí.

Corrimos simulaciones para casos de regiones de formación estelar ya conocidas: Tauro (Tauro C, Tauro S y Tauro E), Ofiuco y Orión. En cada caso, se hicieron 1000 realizaciones en un método Monte Carlo para todas las combinaciones posibles entre número de épocas y cadencia. Se simuló observar para cadencias de cada 1, 2, 3, 4, 5, 6, 8, 9, 10 y 12 meses, cada una con 6, 12, 18, 24, 30 y 36 épocas observadas. Para cada una de éstas combinaciones, nuestra simulación permite medir el valor del error

esperado, σ , sobre la paralaje trigonométrica medida. Hemos graficado los resultados de dos maneras distintas: épocas vs σ y cadencia vs σ . De las primeras, se obtiene una relación de $\sigma \approx 1/\sqrt{n}$, donde n es el número de épocas observadas. Esto significa que las mediciones de paralaje mejoran como la raíz cuadrada inversa del número de observaciones. De las gráficas de cadencia vs σ , se nota una disminución en σ si se hacen las observaciones con una cadencia de 6 meses. Esto significa que la paralaje se ajusta mejor si se hacen las observaciones cada 6 meses. Con este trabajo queda cuantificado cuál es la diferencia entre diversas estrategias. Se concluye que es mejor hacer las observaciones de paralaje con VLBI cada 6 meses y la paralaje podrá ir mejorando con la raíz cuadrada inversa del número de observaciones. Aunque este trabajo no cambia las mediciones ya obtenidas anteriormente por otros autores, sí servirá para perfeccionar futuras observaciones de paralaje con VLBI. La herramienta que construimos para obtener los resultados de esta tesis podrá servir a futuros trabajos para darse una idea del error que se esperará al programar las observaciones de cierta manera.

Summary

Astrometry is the science that studies positions and proper motions of celestial bodies; it has many important applications. For example, by analyzing the apparent motion in time of an object, it is possible to know its distance. Distance is fundamental for the estimation of the intrinsic properties of any object, such as its size, luminosity and mass. Therefore, determining distances has always been very important in astronomy.

There are several methods to calculate distances, many of them indirect. For objects outside the Solar System, there is only one purely geometric technique: the determination of the trigonometric parallax. Parallax is defined as *the apparent displacement of an object as seen from two different points not on a straight line with the object*. In astronomy, parallax is converted into distance by calculating the angular difference in the position of a source observed from two different perspectives. The trigonometric parallax is the result of the movement of the Earth around the Sun through time. Because of this movement we see a displacement of the source on the sky with respect to the background stars. The apparent movement of the source traces an elliptical path, and the semi-major axis of this ellipse is known as the *Trigonometric Parallax* (ϖ) angle. The distance is calculated from this trigonometric parallax (ϖ) as $d = 1/\varpi''$ pc.

Trigonometric parallaxes can be measured at optical wavelengths for isolated bright stars. But this does not work well for stars located in star-forming regions, where they are embedded in dust clouds, and therefore weak at optical wavelengths. In contrast,

dust clouds are practically transparent in radio waves. Thus, observations at radio wavelengths are complementary to optical ones for star-forming regions.

The astrometric precision using a single radio telescope is limited to some fraction of the diffraction limit, $\theta \approx \lambda/D$, where λ is the observing wavelength and D is the antenna diameter. This is typically insufficient for astrometry purposes; This poor resolution is solved by using interferometry. An interferometer is made up of two or more radio telescopes, that can have separations from meters to thousands of kilometers. This last situation is known as very-long-baseline interferometry (VLBI). The angular resolution is given by the maximum separation, B_{max} , between the radio telescopes, $\theta \approx \lambda/B_{max}$.

VLBI parallax measurements are based on multi-epoch observations. Reviewing the literature, one finds that different authors have used different scheduling strategies, particularly with regard to the number of observations used, and the cadence of these observations. Given the wide variations among these strategies, we decided to develop a computer code to simulate VLBI parallax observations. This code uses a Monte Carlo method (i.e. it uses random numbers to solve a physical problem). The algorithm works as follows. It takes as input a set of initial values for the astrometric parameters: initial right ascension (α_0), initial declination (δ_0), components of the proper motion (μ_α and μ_δ), and parallax (ϖ). With those initial parameters, the theoretical trajectory of the object is calculated. From this trajectory, we extract a given number of positions with a certain cadence, accounting for possible variations around the mean cadence (for instance, a variation of a week around a mean cadence of one observation every three months). To these extracted positions we then add realistic astrometric errors, both in right ascension (α) and declination (δ). These astrometric errors take into account errors on the calibrator position, station coordinate, Earth orientation as well as troposphere parameters uncertainties; these errors are added to the extracted positions from the theoretical trajectory. These new positions form our simulated observations

or synthetic observations. We then perform an astrometric fit to these simulated observations, obtaining a determination of a recovered parallax (ϖ^f). Since the simulated observations include noise, ϖ^f is slightly different from the ϖ we put as initial parameter. This algorithm is repeated with a certain number of realizations, to make a Monte Carlo method. In each realization, the noise parameters will be different, since they are programmed to be random values in each realization. At the end of the realizations, we will have many values of ϖ^f (one ϖ_i^f per realization). These (ϖ_i^f) are plotted as a histogram and fitted with a Gaussian, which provides a determination of the mean value and of the standard deviation (σ) around the mean. The average value obtained from ϖ_i^f is very similar to ϖ from the initial parameters. The σ obtained gives an estimation of the error in the determination of the parallax value, for the strategy (number and cadence of observations) used in the simulation. By repeating simulations for different strategies, we can compare the different strategies to each other.

The simulation was done with previously known star-forming regions: Taurus (Taurus C, Taurus S and Taurus E), Ophiuchus and Orion. For each case, 1000 realizations were made in a Monte Carlo method for every possible combinations between number of epochs and cadence. We simulated cadences of 1, 2, 3, 4, 5, 6, 8, 9, 10 and 12 months, each with 6, 12, 18, 24, 30 and 36 epochs observed. For each one of these combinations, our simulation provides an estimate of the error, σ , expected on the parallax measurement. We plotted these results in two different ways: epoch vs σ and cadence vs σ . From the first representation, we obtained a relation $\sigma \approx 1/\sqrt{n}$, where n is the number of observed epochs. This means that the parallax measurements get better as the inverse square root of the number of observations. From the plots of cadence vs σ we find a decrease in sigma if the observations are made every 6 months. This means that the parallax fits are better if the observations are made with a cadence of 6 months. With this work, we quantified the differences between different strategies. We concluded that

it is better to make the VLBI parallax observations every 6 months, and the parallax determination will improve as the inverse square root of the number of observations. Even though this work does not change the measurements made previously by different authors, it will help to improve future VLBI parallax observations. The tool we built to obtain the results from this thesis may help future observers to predict the expected error by scheduling the observations in a certain way.

Chapter 1

Introduction

1.1 Astrometry and Distance Measurements

Astrometry is the science that studies the positions and motions of celestial bodies; it has many applications. For instance, and as we will see below, it is possible to know the distance to a celestial object by analyzing its apparent position as a function of time. In turn, distances are essential to estimate intrinsic properties, such as size, luminosity, mass, age, etc. For example, the correct temperature and luminosity are necessary to place stars on a Hertzsprung–Russell diagram (H-R diagram), and say something about their mass or evolutionary status. As a consequence, the determination of distances in the universe has always played a central role in astronomy. In particular the *distance ladder* plays a very important role in astrophysics. This distance ladder goes from the Solar System to the furthest extragalactic objects. Every step of the ladder is calibrated from the one below. Thus, the first steps are extremely important to the entire ladder, and distances to nearby stars are fundamental. Of course, in the era of precise astrometry the first steps of this *distance ladder* will have to be reevaluated.

Different methods to calculate the distance to a celestial body have been devel-

oped through time. Most of these methods are indirect and for objects outside the Solar System there is only one purely geometric technique: the determination of the trigonometric parallax.

Parallax is defined by the Merriam-Webster dictionary as *the apparent displacement or the difference in apparent position of an object as seen from two different points not on a straight line with the object*. In astronomy, parallaxes are converted to distances by calculating the angular difference in the apparent position of a given source observed from two different perspectives. The trigonometric parallax is the result of the motion of the Earth around the Sun over time. The first attempts to measure trigonometric parallaxes were made in order to confirm the heliocentric theory. Indeed, if the Earth were at the center of the Solar System, distant stars would not display any trigonometric parallax.

In the 1830s, Friedrich Bessel made the first reliable trigonometric parallax measurement to the star 61 Cygni. This was shortly followed by similar measurements by Thomas Henderson to α Centauri and Friedrich von Struve to Vega. Although the values measured by these three astronomers were very uncertain, they showed that even the closest stars are at considerable distances. This explained the lack of earlier trigonometric parallax detection (Loinard, 2014). By the end of the 1990s, the Hipparcos space astrometry mission had measured a large number of trigonometric parallaxes with milli-arcsecond (*mas*) precision, resulting in progress in astrometry. In 2013, Gaia, a new astrometric mission, was launched. Gaia is providing positions, distances, space motions and many physical characteristics of some one billion stars in our Galaxy and beyond. It is a scanning satellite that will survey in a systematic way the whole sky (Mignard, 2005). The performance of Gaia for astrometric accuracy is $4 \mu\text{as}$ at $V = 10$ mag, $10 \mu\text{as}$ at $V = 15$ mag, and $200 \mu\text{as}$ at $V = 20$ mag. The improvement compared with Hipparcos is huge, Gaia is expected to observe about 10 000 times as many objects

and to achieve 100 times higher accuracy for stars that are 100 times fainter (Lindgren and de Bruijne, 2005). Hipparcos and Gaia observe at optical wavelengths. For stars embedded in dusty star-forming regions, it is useful to observe at lower frequencies where the extinction is significantly lower (as explained later in this chapter) with the radio technique of *Very-long-baseline interferometry* (VLBI). With that technique, an astrometric accuracy of 10 micro-arcsecond (μas) can be achieved (Reid and Honma, 2014).

Due to proper motion of the source and the annual rotation of the Earth around the Sun, we can see a displacement of the source on the sky with respect to the background stars. The proper motion traces the true displacement of the source across the sky, and is generally assumed to be linear and uniform. The only exception is if the star belongs to a binary or multiple system; in that case the proper motion will not be uniform. Because of the annual rotation of the Earth, the apparent position of the source describes an elliptical path on the sky (see Fig. 1.1). The semi-major axis of that ellipse is known as the *Trigonometric Parallax* (ϖ). From trigonometry we can see that

$$d = \frac{1 \text{ AU}}{\tan \varpi} \simeq \frac{1}{\varpi} \text{ AU} \quad (1.1)$$

taking into account the small-angle approximation¹ $\tan \varpi \simeq \varpi$, where ϖ is measured in *radians*. Using $1 \text{ radian} = 57.2957795^\circ = 206264.806''$, we obtain

$$d \simeq \frac{206,264}{\varpi''} \text{ AU}. \quad (1.2)$$

The parsec (**par**allax-**sec**ond, pc) is defined as the distance to a source whose parallax is 1 arcsec. From the previous equation we see that $1 \text{ pc} = 2.06264806 \times 10^5 \text{ AU}$.

¹ $\tan \varpi \simeq \varpi$ is true for small angles. For instance if we assume $\varpi = 1''$, $\tan \varpi / \varpi = 1.00000000001$, very close to unity.

Introducing the parsec simplifies the equation to:

$$d = \frac{1}{\varpi''} \text{ pc} \quad (1.3)$$

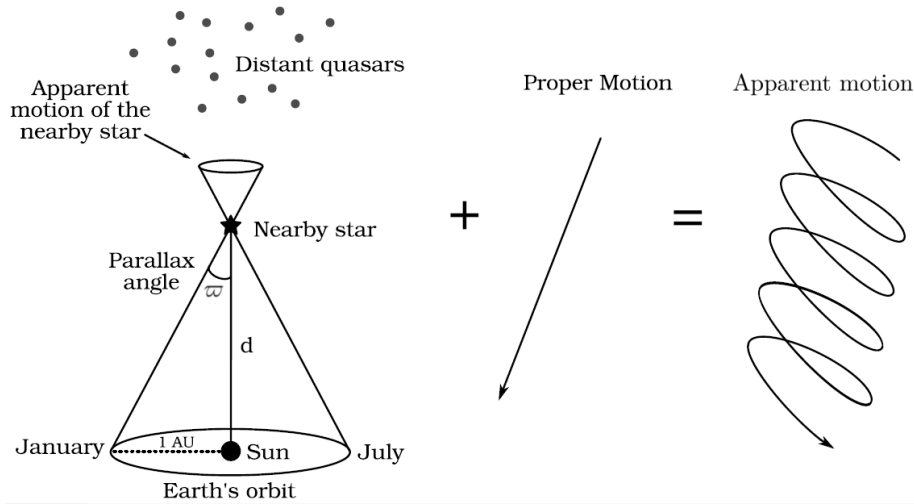


Figure 1.1: Trigonometric Parallax and Proper Motion (Torres, 2009)

1.2 Interferometry

Trigonometric parallaxes can be measured at optical wavelengths (for instance with Gaia) for isolated and bright stars. But this can become difficult for stars located in star-forming regions. This is because young stars in star-forming regions are embedded in dust clouds, which makes them faint at optical wavelengths. Radio waves penetrate many magnitudes of visual extinction (Draine, 2006). In this situation radio-wavelength observations can be used advantageously.

The astrometric precision of a single radio telescope is limited to some fraction of the diffraction limit, $\theta \approx \lambda/D$, where θ is the angular diameter in radians of the region that the radio telescope “sees”, λ is the observing wavelength and D is the antenna diameter. For example, a radio telescope of 100 m in diameter observing at a wavelength

of 10 cm has an angular resolution of ≈ 200 arcseconds (Rodríguez, 1984), very poor for astrometry purposes. This low resolution “problem” is solved by using interferometry.

An interferometer is made of two or more radio telescopes. In a typical interferometer, the shortest separation between the individual telescopes is of the order of their diameter. The longest separation, on the other hand, can be up to thousands of meters for conventional interferometers, and up to thousands of kilometers for VLBI configurations. For conventional interferometers, the signals from the individual telescopes are combined in real time, using a correlator co-located with the antennas. The correlator calculates the mutual coherence function of the electric field for each pair of antennas. When expressed as a function of baseline length and orientation, this quantity is called a visibility. The reconstruction of the image is made from the visibilities through an inverse Fourier transform. In VLBI observations, the correlation is not made in real time. Instead, the signals are time tagged using independent atomic clocks at each antenna and recorded on magnetic tape, or more recently on magnetic disks. These tapes or disks are brought to a correlator (which is usually not co-located with any of the antennas) to be combined weeks or even months after the observations.

In an interferometer, the angular resolution, θ , is given by the maximum separation between the radio telescopes, or the maximum baseline, B_{max} .

$$\theta \approx \frac{\lambda}{B_{max}}. \quad (1.4)$$

With radio telescopes located thousands of kilometers apart, it is possible to obtain an angular resolution of ~ 1 mas. The corresponding astrometric accuracy is given by θ/SNR , where SNR is the signal-to-noise with which the source is detected (Condon, 1997). Thus, if we have a SNR of a hundred, the astrometric accuracy would be of order of $10 \mu\text{as}$, comparable to the accuracy of the European space astrometry mission

Gaia (Bourda et al., 2011).

1.3 Previous Observations

VLBI parallax measurements are obtained through observations of a given source at several epochs. Reviewing the literature, one finds that different authors have used different scheduling strategies (for instance concerning the number of observations used, or the cadence of these observations). For example, Loinard et al. (2007) made a parallax measurement for the star T Tau Sb with observations scheduled every 2 months during a total period of 2 years (between 2003 September and 2005 July), for HP Tau/G2 Torres et al. (2009) used a total of 9 epochs between 2005 and 2007 separated by 2 or 3 months. For sources like Hubble 4 and HDE 283572 in Torres et al. (2007), the parallax was measured with 6 epochs (between 2004 September and 2005 December) separated by 2 to 4 months. Menten et al. (2007) measured the trigonometric parallax of several member stars of the Orion Nebula Cluster with observations conducted on 2005 September 25, 2006 March 02, 2006 September 09, and 2007 March 6. These dates sample well the maximum extent of the Earth's orbit as viewed by the source. Reid et al. (2009b) have measured the positions of H₂O masers in Sgr B2, a massive star-forming region near the Galactic center. The positions were measured on 12 epochs over a time span of one year.

A similar disparity is found for VLBI distance measurements to pulsars. For the double neutron star system PSR B1913+16, Deller et al. (2018) used observations obtained every 6 months close to the parallax extremas. On the other hand, the measurements of nine pulsar parallaxes made by Brisken et al. (2002) used five epochs separated by about 3 months over the course of one year.

More recently, the BeSSeL Survey (Bar and Spiral Structure Legacy Survey) was

performed to study the spiral structure and kinematics of the Milky Way galaxy. The target sources were methanol and water masers that are associated with young massive stars and compact HII regions, that trace the spiral structure. Some examples of parallax measurements for this survey are reported by Reid et al. (2009a). For the parallax measurement of the methanol masers in S 252 and G232.6+1.0, they made observations at four epochs, from 2005 October to 2007 March with a cadence of approximately 6 months. These observations sample the peaks of the sinusoidal trigonometric parallax signature in right ascension. Xu et al. (2006) have measured the distance to the massive star-forming region W3(OH) in the Perseus spiral arm of the Milky Way using measurements at 5 epochs every 6 months approximately. Finally, we should mention GOBELINS (The Gould's Belt Distances Survey), a project aimed at measuring the proper motion and trigonometric parallax of a large sample of young stars in nearby regions. Ortiz-León et al. (2017) measured parallax and distance to twelve stellar systems located in the Lynds 1688 dark cloud in the Ophiuchus region. Kounkel et al. (2017) presented the results of young star-forming regions toward the Orion Molecular Cloud Complex and Galli et al. (2018) presented new trigonometric parallaxes and proper motions of young stellar objects in the Taurus molecular cloud complex. Ortiz-León et al. (2018) derived the distance and structure of the Perseus molecular cloud. All these GOBELINS measurements use a cadence of 6 months, with a total of 5 to 8 epochs.

1.4 Motivation

Given the variety of observation strategies used in previous works, we decided to make a simulation program in order to understand how the parallax precision differs under different choices of cadence and number of epochs. As will be described in more details in the next chapter, for a given set of astrometric parameters [right ascension (α_0) and

declination (δ_0) at a reference epoch, components of the proper motion (μ_α and μ_δ), and parallax (ϖ), we calculated the expected trajectory of the object on the plane of the sky. From that trajectory we extracted positions at a given number of epochs with a certain cadence (taking into account observation epochs to be within 10 days of the expected observing date). Having these extracted positions, we now add an astrometric error in right ascension and declination; it is calculated from Pradel et al. (2006) and takes into account errors in calibrator position, station coordinate, Earth orientation and troposphere parameter uncertainties. With these synthetic observations a new trajectory is sampled. We then calculated the recovered parallax for this set of synthetic measurement. Repeating this for different choices of cadence and number of observations in a Monte Carlo method, we estimate the error on the parallax measurement associated with any specific scheduling strategy

1.4.1 Monte Carlo Method

Named after the city famous for its casinos, the Monte Carlo method is a numerical technique that makes use of random numbers to solve statistical physics problems in order to estimate some functions of a probability distribution. A problem that does not have a stochastic component sometimes may also be posed as a problem with a component that can be identified with an expectation of some function of a random variable. The problem is then solved by estimating the expected value by the use of a simulated sample from the distribution of the random variable (Gentle, 2010). The first real applications of this statistical method were from Enrico Fermi, with his work on neutron diffusion in the early 1930s, later the Monte Carlo method was used by Ulam and Metropolis and their collaborators in the construction of the hydrogen bomb. They were the first to propose and employ the Monte Carlo method as a viable numerical technique for solving practical problems (Murthy, 2001). Nowadays, this method is used

in a very large number of fields to solve all kinds of problems. To mention just a few, in mathematics: to solve integrals with complex boundary conditions; in astronomy: generation of initial conditions for N-body simulations, modeling light scattering by dust grains; and for the work presented in this thesis, to estimate astrometric errors on parallax measurement.

From these Monte Carlo simulation (explained in more details in the following chapter), we will be able to determine the best way to schedule the observations of VLBI parallax measurements. It is well known that astrometry at radio wavelengths has made great progress over the last decade, with a positional accuracy of $\approx 10 \mu\text{as}$ for compact sources relative to background quasars. Though this thesis will not change the results from the previous observations, the tool (simulation program) we built will be very useful for future parallax observations. Gaia will be covering a lot of stars in the optical range in the Milky Way, but for stars observed at radio frequencies there is still a lot to study. Radio astrometry continues to advance. For example, the next-generation Very Large Array (ngVLA)² could provide a sensitivity increase of more than an order of magnitude. Thus, the ngVLA could achieve astrometric accuracies of $\approx 1 \mu\text{as}$ (Reid et al., 2018). With this work, future VLBI parallax observations could be optimized by knowing how to schedule properly the observations, and by knowing this, it will save telescope time.

²The ngVLA is an astronomical observatory planned to operate at centimeter wavelengths (25 to 0.26 centimeters, corresponding to a frequency range extending from 1.2 GHz to 116 GHz). The observatory will be a synthesis radio telescope constituted of approximately 214 reflector antennas each of 18 meters diameter, operating in a phased or interferometric mode (Selina et al., 2018).

Chapter 2

Methodology

In this chapter we will describe the simulation program that we made in order to create a series of synthetic observations. The algorithm we created is shown in Fig. 2.5, and the code is described schematically in the appendix E. The algorithm is explained step by step in this chapter.

2.1 Trajectory

The simulation, mentioned in the previous section, works in the following manner. We choose a set of astrometric parameters: an initial right ascension (α_0), an initial declination (δ_0), the components of the proper motion (μ_α and μ_δ) and a parallax (ϖ). This corresponds to the input of **Astro Elements** in the algorithm (Fig. 2.5). With those initial parameters the program calculates an expected trajectory on the sky of the object through time; this trajectory is calculated with (Seidelmann, 1992):

$$\alpha(t) = \alpha_0 + \mu_\alpha t + \varpi f_\alpha(t), \tag{2.1}$$

$$\delta(t) = \delta_0 + \mu_\delta t + \varpi f_\delta(t), \tag{2.2}$$

where f_α and f_δ are the projections of the parallactic ellipse over α and δ , and are given by

$$f_\alpha = (X \sin \alpha - Y \cos \alpha) / \cos \delta, \quad (2.3)$$

$$f_\delta = X \cos \alpha \sin \delta + Y \sin \alpha \sin \delta - Z \cos \delta. \quad (2.4)$$

Here (X, Y, Z) are the barycentric coordinates of the Earth in a specific reference frame associated with the Solar System; these were calculated with the library `NOVAS` in python (Kaplan et al., 2012). After this step, we will have a theoretical trajectory of the apparent motion from the chosen source. This is represented by the process *Trajectory* in the flux diagram (Fig. 2.5).

2.2 Epoch selection

From the theoretical trajectory of the apparent motion on the sky of a certain object, we extract a given number N of positions (later to be named epochs) for a given cadence M (taken to be the separation between successive epochs). In practice, observations are never scheduled exactly as requested because of practical constraints at the observatories, or of meteorological requirements. We take this into account by choosing the observation epochs to be within 10 days of the expected observing date. In our simulations, this is achieved by choosing a random number uniformly distributed between -10 days and $+10$ days, using the function in the python library `random.uniform`. In the program, the specific scheduling strategy is defined in the **Simulation parameters** (Fig. 2.5). With these parameters we proceed to extract the positions, shown in our flux diagram as the process of *Extract positions*. At this point, we have N positions with a cadence of M months ± 10 days.

In Fig. 2.1, we can see an example of the theoretical trajectory described before, and the extracted positions at a certain epochs from this trajectory, in other words, we have N positions every M months ± 10 days.

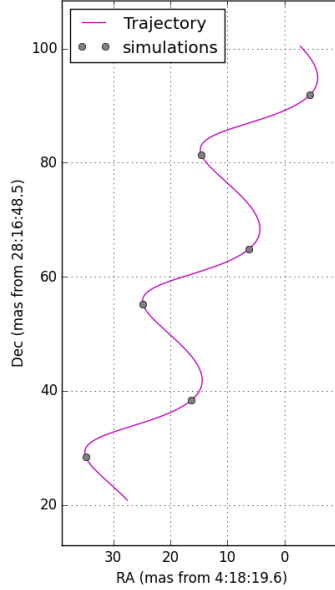


Figure 2.1: Example of the theoretical trajectory represented by the purple line, and the extracted positions of observations are the grey dots.

2.3 Astrometric errors

Having the positions now extracted from the theoretical trajectory, an astrometric error will be added to both right ascension and declination. These astrometric errors were chosen based on Pradel et al. (2006) and take into account errors on the calibrator position, station coordinate, Earth orientation as well as troposphere parameters uncertainties. Their simulation was performed for sources observed with the Very Long Baseline Array (VLBA) and the European VLBI Network (EVN) at various declinations and angular separations. From Pradel et al. (2006) we use Tables 3 and 4, which contain VLBA rms astrometric error for a relative source separation $(\alpha - \alpha_0) \cos \delta_0 = 1^\circ$ and

$\delta - \delta_0 = 1^\circ$, respectively. Such errors were calculated for sources at declinations -25° , 0° , 25° , 50° , 75° and 85° between the target source and the calibrator; there is an astrometric error¹ for right ascension ($\Delta\alpha \cos \delta$) and declination ($\Delta\delta$). We do a quadratic sum from the astrometric errors from the Table 3 (source separation $(\alpha - \alpha_0) \cos \delta_0 = 1^\circ$) with the values from the Table 4 (source separation $\delta - \delta_0 = 1^\circ$), for every value of $\Delta\alpha \cos \delta$. and $\Delta\delta$ at the given declination of source. Then, we do a polynomial curve fitting, for right ascension and for declination, shown in Fig. 2.2. This polynomial fit enable us to interpolate the correct value of the errors for any value of the declination and right ascension. The random error number will be obtained from the function `random.gauss` in the python library. This function creates a random number taken from a Gaussian distribution. The value for the standard deviation (σ) required for this Gaussian distribution will be given from the fit in Fig. 2.2. As we see in the figure, it will depend on the declination of the source we want to simulate. We will center the Gaussian at a mean (μ) of $\mu = 0 \mu as$. As an example, for a source of 0° in declination, the astrometric error added in right ascension will be a random number created with a Gaussian distribution with $\sigma \approx 50 \mu as$. Similarly, for the astrometric error in declination, will be a random number created with a Gaussian distribution with $\sigma \approx 100 \mu as$. These errors correspond to the input of **Noise parameters**, and then we will proceed to add these errors to the positions extracted in the steps above. This belongs to the process *Add errors* from the flux diagram (Fig. 2.5). At this stage in the simulation, we now have our simulated, or synthetic, observations.

¹The values we used are when all the error components are combined together.

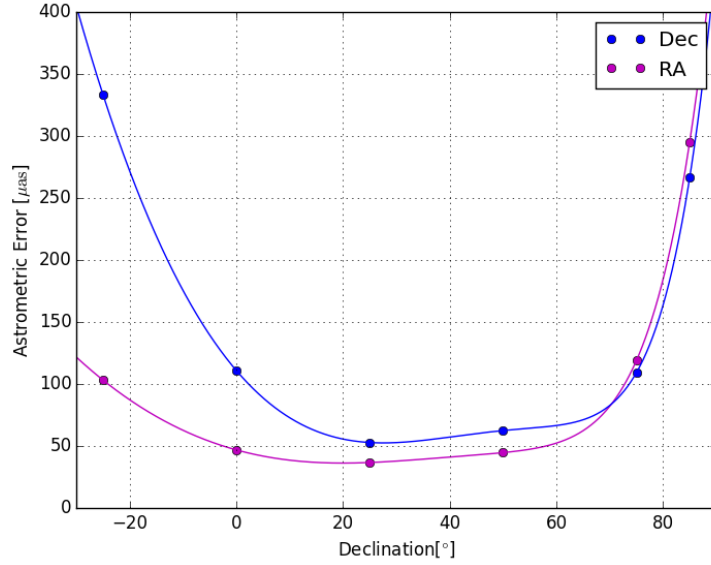


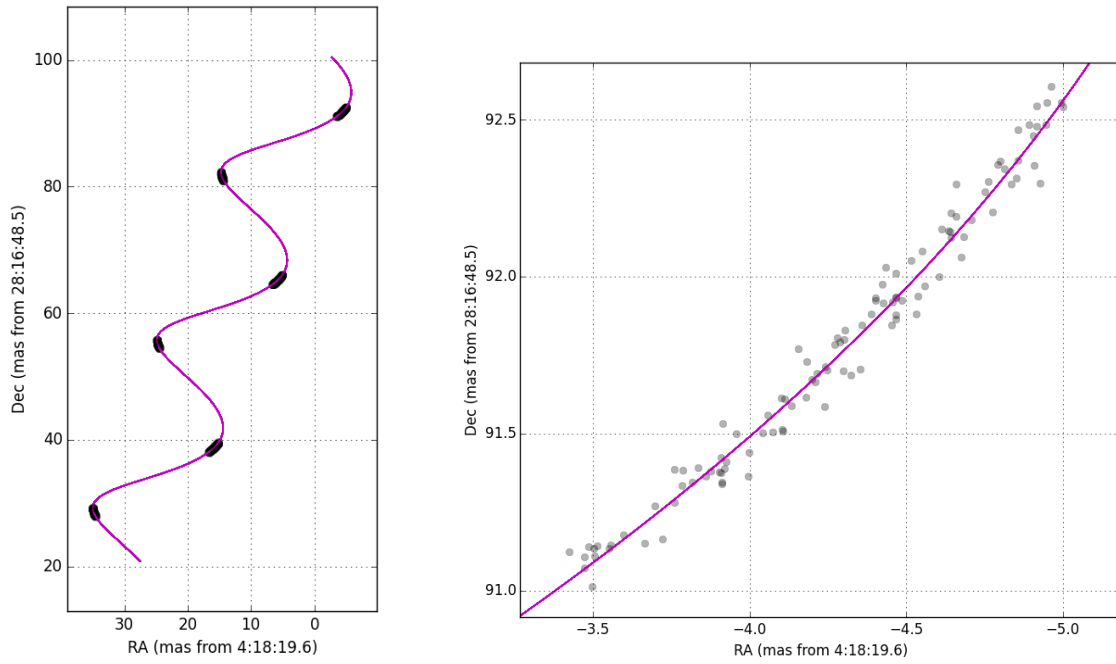
Figure 2.2: Curve fitting of values of tables 3 and 4 from Pradel et al. (2006). The purple line represents the fit for the astrometric error in right ascension and the blue line for declination.

2.4 Adjust new trajectory

Once we have these new synthetic observations, an astrometric fit is made (similar to the purple line in Fig. 2.1). The trajectory recovered from this fit yields a recovered parallax (ϖ^f), as well as recovered values for the components of the proper motion (μ_α^f and μ_δ^f), the initial right ascension (α_0^f), and initial declination (δ_0^f). Since these simulated observations include added errors (*Noise parameters*), the recovered parameters will be slightly different from the true ones we put in the initial values (*Astro Elements*). In the algorithm in Fig. 2.5, this step is shown with the internal storage of α_0^f , δ_0^f , μ_α^f , μ_δ^f , ϖ^f . For the moment we will only focus on the result from the recovered parallax (ϖ^f).

2.5 Monte Carlo

The steps described above, from *Extract positions* to the recovered ϖ^f , are repeated i times in a Monte Carlo simulation. In each realization, the **Noise parameters** and the exact choice of observing epochs will change, since they are set to be random values. Thus, for every realization made, we will have different synthetic observations and therefore a different value for ϖ^f . As an example, Fig. 2.3 shows a simulation with $i = 100$ (i.e. 100 realizations). In the left panel we can see the synthetic observations over the theoretical trajectory, they all pile up in every epoch. In the right panel, we see a close up from the left panel, and we note the different positions, in right ascension and in declination, of the simulated observations, due to the **Noise Parameters**. As we said before, the simulation is run i times, and for every realization we recover a different ϖ^f . All these values (ϖ_i^f) are arranged in a histogram, as shown in Fig. 2.4. From this histogram we fit a Gaussian to extract the mean parallax ($\overline{\varpi_i}$), and the standard deviation (σ). This corresponds to the process of *Calculates $\overline{\varpi_i}$ and σ* from the flux diagram in Fig. 2.5. The mean parallax obtained from the Gaussian fit is, of course, very similar to the input parallax from the **Astro Elements**. The result for σ is the one we will be analyzing in the next chapter, it provides an estimate of the error in our determination of the parallax value for a given simulation. It will be the final display in the flux diagram shown in Fig. 2.5.



(a) Simulated observations (grey dots) through the theoretical trajectory (purple line).

(b) Close up of the simulated observations, in this case we can note different positions due to astrometric errors for every point.

Figure 2.3: For this example the simulation was run $i = 100$ times.

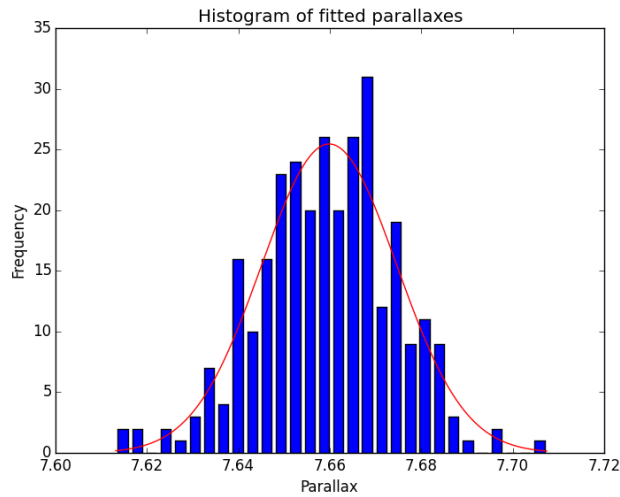


Figure 2.4: This histogram is calculated with the same parameters previously mentioned and for $i = 100$ times, so this is for 100 different parallaxes.

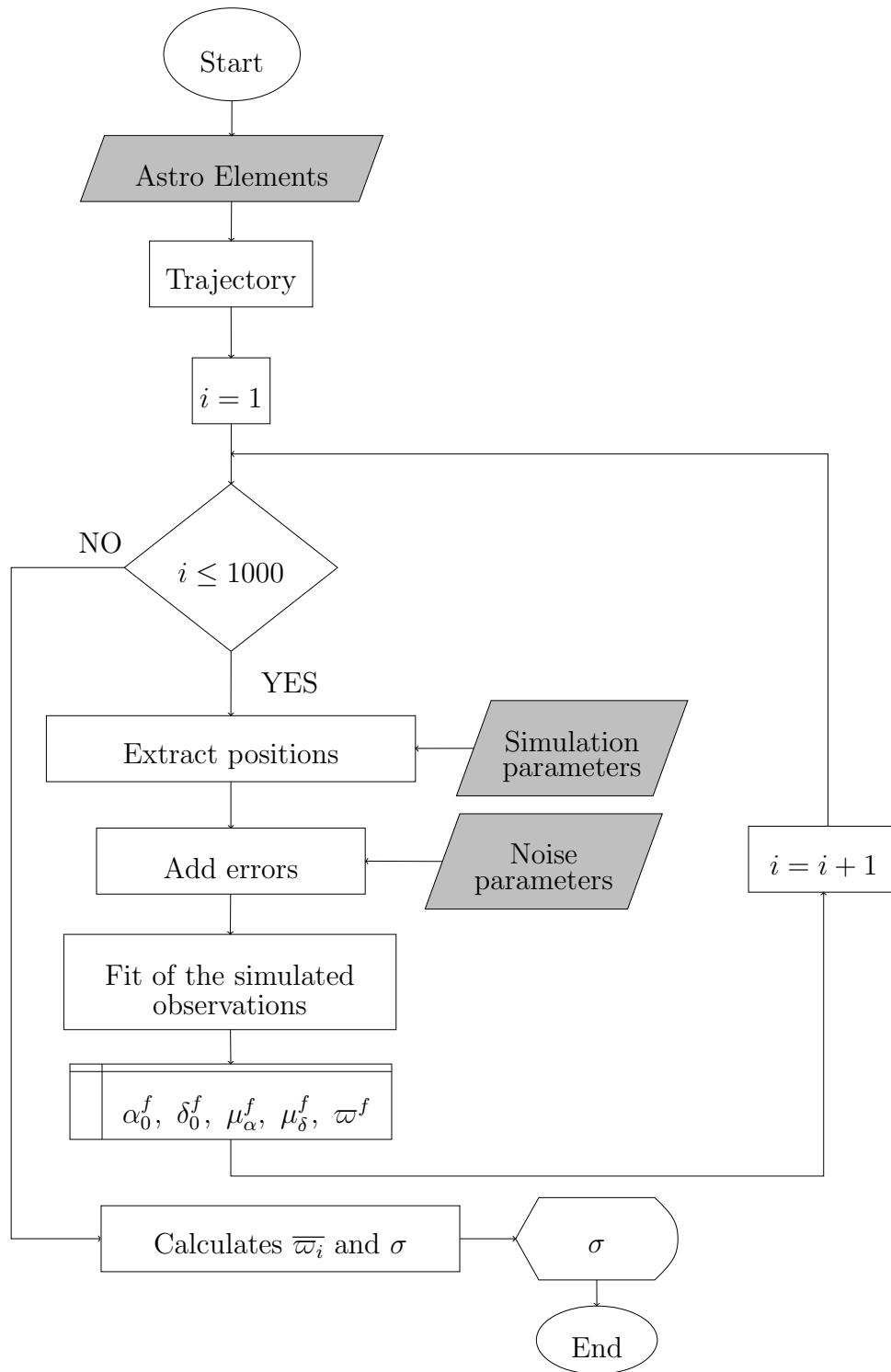


Figure 2.5: Flow diagram of the simulation.

Chapter 3

Results

Simulations were run for the well-known star-forming regions Taurus (Taurus C, Taurus S and Taurus E), Ophiuchus and Orion. For Taurus C we use an average of the astrometric elements of the stars V773 Tau, Hubble 4 and HDE 283572 from the dark cloud L1495. For Taurus S, we used the values of T Tau and for Taurus E, we used the values of HP Tau. The parameters used for the simulation are all summarized in Table 3.1. These parameters correspond to the input of **Astro Elements**, mentioned in the previous chapter.

	Parameters				
	L1495	T Tau	HP Tau	Ophiuchus	ONC
α_0	$04^h18^m19.6^s$	$04^h21^m59.42^s$	$04^h35^m54.16^s$	$16^h29^m30.19^s$	$05^h41^m29.9^s$
δ_0	$28^\circ16'48.6''$	$19^\circ32'05.71''$	$22^\circ54'13.49''$	$-24^\circ39'42.1''$	$-03^\circ49'22.5''$
μ_α [mas/yr]	10.09	4.02	13.85	-6.52	0.06
μ_δ [mas/yr]	-26.51	-1.18	-15.4	-25.05	-0.12
π [mas]	7.67	6.82	6.20	7.01	2.41

Table 3.1: Inputs for the Astro Elements used for the simulation. All the elements were compiled from Torres et al. (2007), Torres et al. (2012), Torres et al. (2009), Loinard et al. (2007), Ortiz-León et al. (2017), Kounkel et al. (2017).

For every case mentioned in Table 3.1, the simulation was run for a given number

of epochs observed at a certain cadence. We made 1000 realizations in a Monte Carlo method for all the possible combinations of number of epochs and cadence. We simulate cadences of 1, 2, 3, 4, 5, 6, 8, 9, 10 and 12 months each at 6, 12, 18, 24, 30 and 36 epochs. As specified in the last chapter, the final display of the simulation will be the standard deviation (σ), obtained from the statistics made from the recovered parallaxes (ϖ_i). For every case mentioned in Table 3.1, this σ is shown in the next Tables (from Table 3.2 to Table 3.6).

Cadence (months)	Epochs					
	6	12	18	24	30	36
1	170.65	14.06	10.85	9.24	8.53	7.8
2	19.1	13.61	10.88	9.38	8.62	7.52
3	19.05	14.15	11.07	9.58	8.58	7.78
4	20.15	13.68	11.12	9.82	8.5	8.03
5	18.45	14.08	10.93	9.53	8.84	7.79
6	15.39	10.61	8.82	7.54	6.66	5.81
8	21.64	13.63	11.29	10.28	8.15	7.73
9	20.16	13.18	10.72	9.35	9.02	7.78
10	25.51	14.18	11.54	9.64	8.5	7.9
12	464.5	269.24	225.31	183.52	153.43	146.7

Table 3.2: Standard deviation ($\sigma \times 10^{-3} mas$) for all the combinations of cadence and epochs for L1495.

From the values obtained in the tables, we produced two types of plots: the number of epochs vs σ , and cadence vs σ . In figures 3.1 to 3.10, we see the first type of plot (epochs vs σ), these figures belong to the L1495 case. The plots for the T Tau, HP Tau, Ophiuchus and ONC cases are displayed in Appendix A, B, C and D, respectively.

For each cadence we make a plot of the values of σ . Then we make a linear fit to the points. Analyzing the latter plots we can see a trend on the slope from the linear fit. The green line is the best fit obtained by allowing both the slope and the intercept to vary. The black line is the best fit obtained by fixing the slope at -0.5 , varying only the intercept. A slope of -0.5 is the value expected from standard error propagation.

Cadence (months)	Epochs					
	6	12	18	24	30	36
1	175.95	13.46	11.12	10.39	9.16	8.13
2	20.62	14.15	11.49	10.66	8.82	8.39
3	20.46	13.21	11.31	9.9	8.53	8.08
4	21.31	14.93	11.78	10.24	8.45	7.8
5	19.75	14.04	11.08	10.23	9.55	8.53
6	15.5	10.34	8.98	7.22	6.77	6.11
8	22.44	15.1	11.72	9.8	9.44	8.24
9	20.03	14.32	11.61	9.79	9.02	8.06
10	26.97	15.35	11.59	10.31	9.01	8.41
12	449.37	300.5	212.97	182.78	168.42	151.79

Table 3.3: Standard deviation ($\sigma \times 10^{-3} mas$) for all the combinations of cadence and epochs for T Tau.

Cadence (months)	Epochs					
	6	12	18	24	30	36
1	200.61	13.13	11.43	9.17	8.81	7.74
2	20.02	14.13	12.2	10.12	8.89	8.0
3	19.96	13.54	10.82	9.82	9.1	7.9
4	20.25	14.38	11.34	9.9	8.55	8.49
5	18.79	13.24	11.48	9.69	9.01	8.46
6	15.1	10.85	8.28	7.7	6.22	6.01
8	22.38	13.94	11.61	9.87	8.35	7.79
9	19.75	14.17	11.41	9.8	8.82	7.56
10	24.54	15.03	11.74	9.79	8.54	7.9
12	512.95	318.88	270.59	219.82	186.73	162.2

Table 3.4: Standard deviation ($\sigma \times 10^{-3} mas$) for all the combinations of cadence and epochs for HP Tau.

Cadence (months)	Epochs					
	6	12	18	24	30	36
1	544.75	38.1	31.39	28.44	23.8	21.83
2	55.73	38.42	30.53	27.18	23.84	22.51
3	53.05	38.68	31.87	28.04	25.49	22.73
4	57.33	41.64	32.16	27.35	23.96	21.28
5	57.37	39.49	31.44	26.37	25.64	22.39
6	45.62	27.7	25.04	20.16	19.31	17.14
8	64.04	40.29	31.04	26.36	24.05	20.74
9	52.43	41.2	31.62	27.96	24.27	21.9
10	70.87	39.65	34.2	27.4	24.81	22.07
12	1359.19	859.86	684.14	591.45	466.68	445.08

Table 3.5: Standard deviation ($\sigma \times 10^{-3} mas$) for all the combinations of cadence and epochs for Ophiuchus.

Cadence (months)	Epochs					
	6	12	18	24	30	36
1	219.71	21.23	16.1	16.0	12.96	12.14
2	29.31	23.0	16.57	14.43	12.85	12.29
3	31.57	20.82	17.71	15.57	13.97	11.79
4	30.98	22.64	17.02	14.19	14.71	13.0
5	28.94	21.53	16.57	14.47	13.29	12.04
6	21.25	16.28	12.33	10.34	9.49	8.45
8	34.64	21.25	16.04	15.09	12.83	11.28
9	30.96	22.38	17.01	14.45	12.54	11.47
10	31.43	20.45	16.52	15.03	13.02	11.84
12	1368.92	806.62	628.01	542.38	512.43	462.07

Table 3.6: Standard deviation ($\sigma \times 10^{-3} mas$) for all the combinations of cadence and epochs for ONC.

Note that for the plots for a cadence of 1 month (Fig. 3.1 and from the appendices Figs. A.1, B.11, C.21, D.31) the blue point for 6 epochs was not taken into account for the fit. It shows a higher σ since the parallactic ellipse is not complete for this particular combination for 6 epochs of observations every month.

From the slope trend of ≈ -0.5 we can get that

$$\log \sigma \approx -0.5 \log(n)$$

where n is the number of epochs, and now we have that

$$\sigma \approx \frac{1}{\sqrt{n}}.$$

This means that the parallax measurements improve as the square root of the number of observations. This is a value one would expect from standard error propagation and indicates that our simulation is made correctly.

The other type of plots that we examined were cadence vs σ , for each case from Table 3.1. These plots are displayed in Figures 3.11 to 3.15. For all other cases (L1495, T Tau, HP Tau, Ophiuchus, and ONC) we can observe a similar behavior in the plots. Analyzing these plots, we see a noticeable trend, a decrease in the value of σ for all the data points corresponding to a cadence of 6 months. That case corresponds to observations that systematically target close to the extremes of the parallactic trajectory. Furthermore we see a peculiar point corresponding to a cadence of 1 month for 6 epochs, this point is anomalous because for this combination of cadence-epochs, the simulation only cover a 6 month period, this means that it does not track the entire parallactic ellipse. Thus, we will not have a good measure for the recovered ϖ^f from the simulation, and a larger value of σ is obtained. A similar thing is happening with the observations for a cadence of 12 months, since it is only observing once a year, the

entire parallactic ellipse is not well computed, and the σ tends to be bigger for those cases.

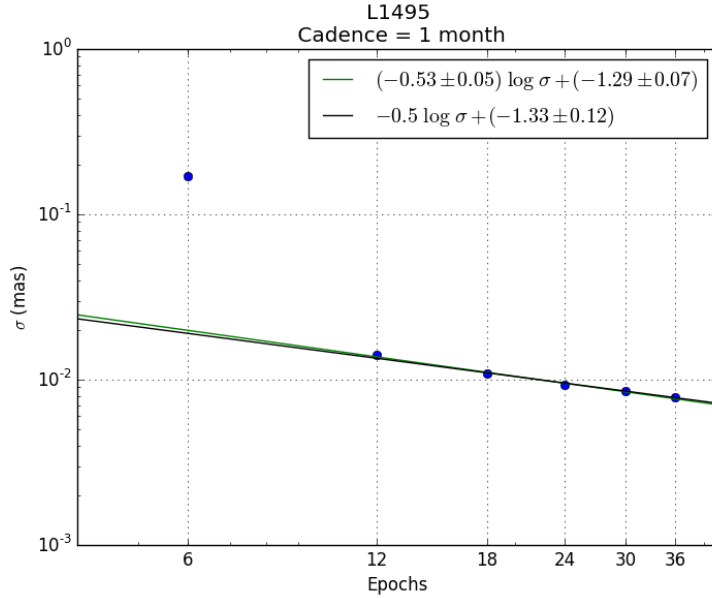


Figure 3.1: Relation of epochs and σ for cadence of 1 month, each blue point represent a value of σ , the green line is the linear adjust that passes through the points, and the black line is the linear adjust that passes through the points with a slope of -0.5 . Note: The blue point for 6 epochs was not taken into account for the fit.

The results obtained here show that the best strategy to measure trigonometric parallaxes from astrometric observations is to observe with a cadence of six months, targeting the extremes of the parallactic ellipse. Although this result is expected, we now can quantify how much the parallax measurements varies with the observing strategy, as we now illustrate comparing cadences of 3 and 6 months. Comparing Fig. 3.6 with Fig. 3.3, we see that in order to get a similar value for the rms error, of $\sigma \approx 10^{-2}$ mas, observing with a cadence of three months, we would need observations at 24 epochs, whereas we would only need 12 epochs if we observed with a cadence of 6 months.

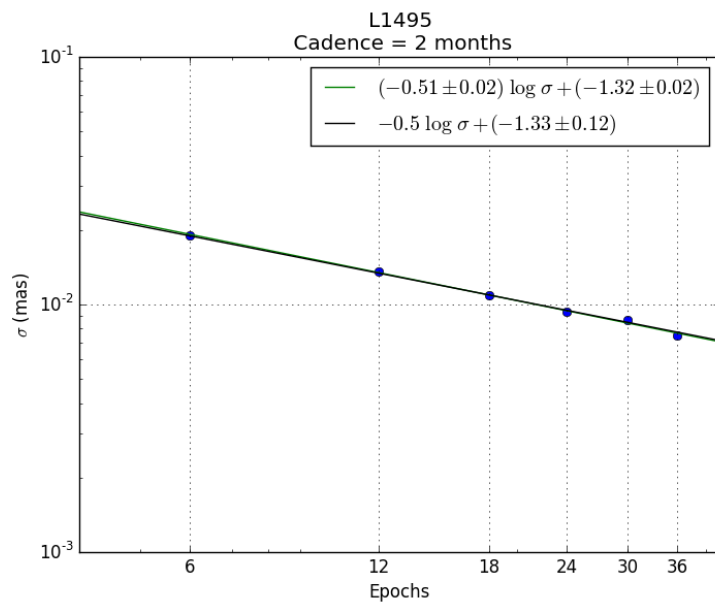


Figure 3.2: Relation of epochs and σ for cadence of 2 months, each blue point represent a value of σ , the green line is the linear adjust that passes through the points, and the black line is the linear adjust that passes through the points with a slope of -0.5 .

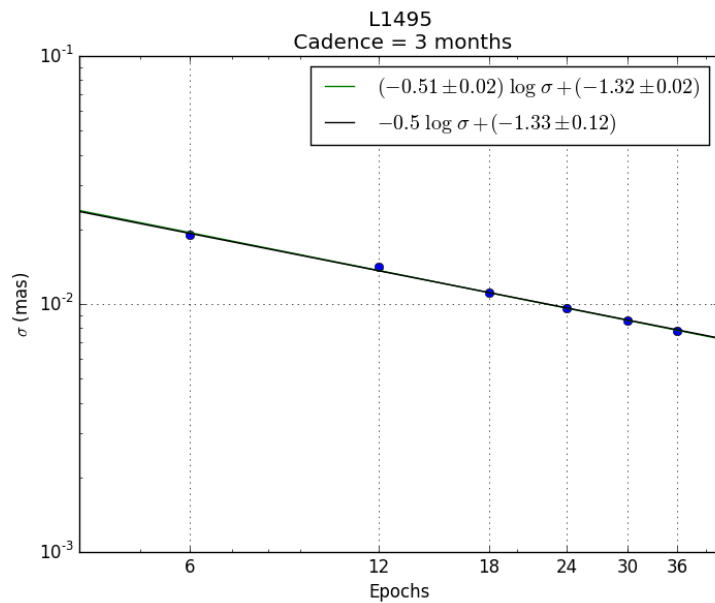


Figure 3.3: Relation of epochs and σ for cadence of 3 months, each blue point represent a value of σ , the green line is the linear adjust that passes through the points, and the black line is the linear adjust that passes through the points with a slope of -0.5 .

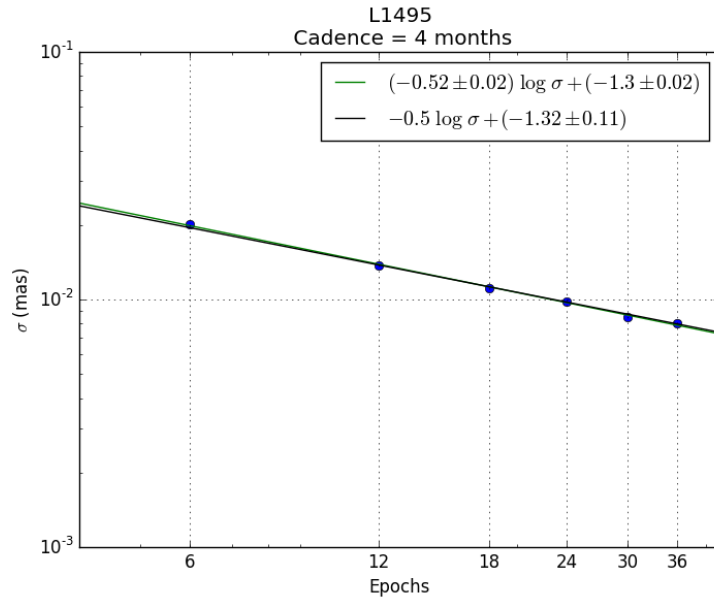


Figure 3.4: Relation of epochs and σ for cadence of 4 months, each blue point represent a value of σ , the green line is the linear adjust that passes through the points, and the black line is the linear adjust that passes through the points with a slope of -0.5 .

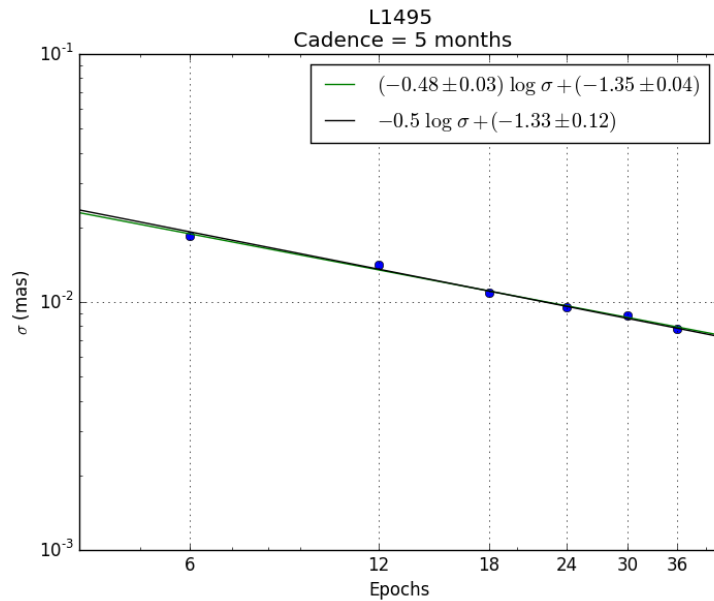


Figure 3.5: Relation of epochs and σ for cadence of 5 months, each blue point represent a value of σ , the green line is the linear adjust that passes through the points, and the black line is the linear adjust that passes through the points with a slope of -0.5 .

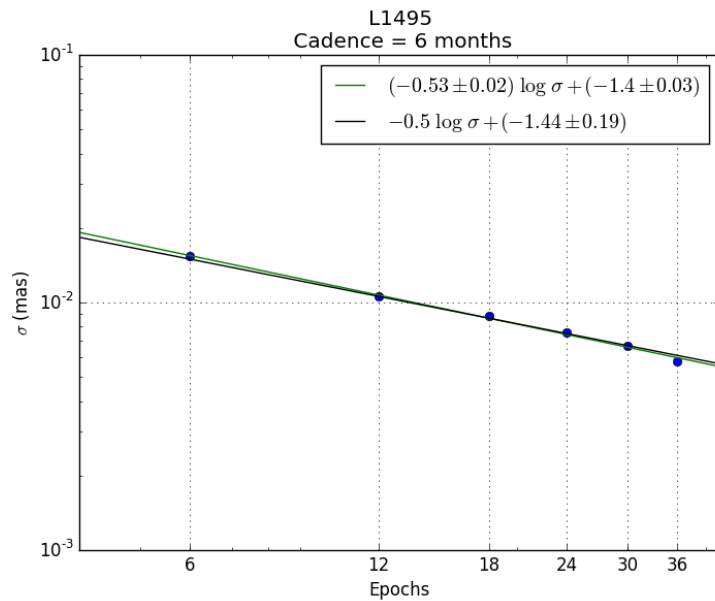


Figure 3.6: Relation of epochs and σ for cadence of 6 months, each blue point represent a value of σ , the green line is the linear adjust that passes through the points, and the black line is the linear adjust that passes through the points with a slope of -0.5 .

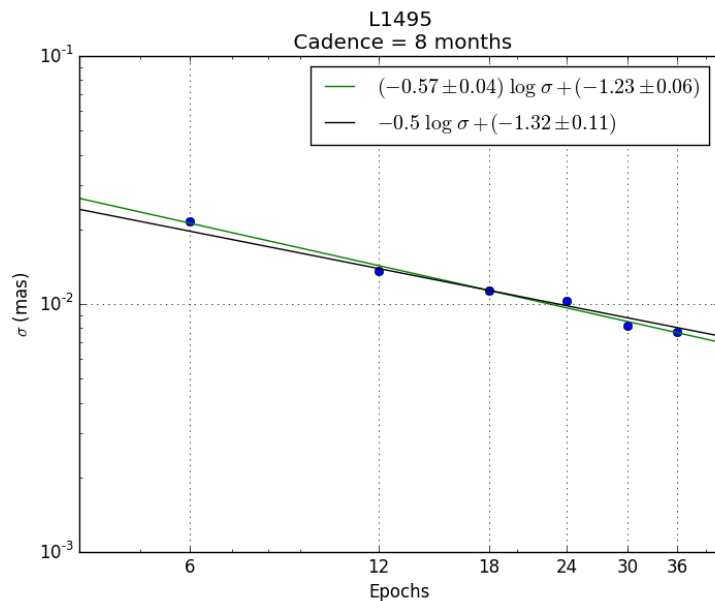


Figure 3.7: Relation of epochs and σ for cadence of 8 months, each blue point represent a value of σ , the green line is the linear adjust that passes through the points, and the black line is the linear adjust that passes through the points with a slope of -0.5 .

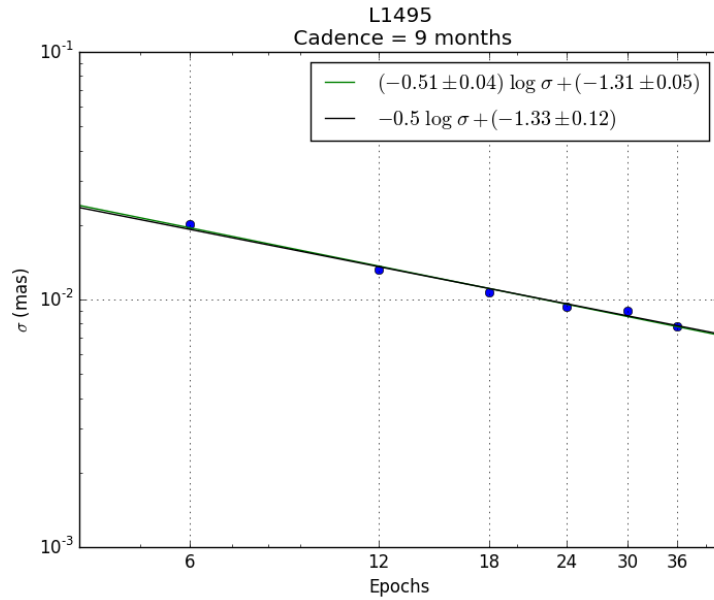


Figure 3.8: Relation of epochs and σ for cadence of 9 months, each blue point represent a value of σ , the green line is the linear adjust that passes through the points, and the black line is the linear adjust that passes through the points with a slope of -0.5 .

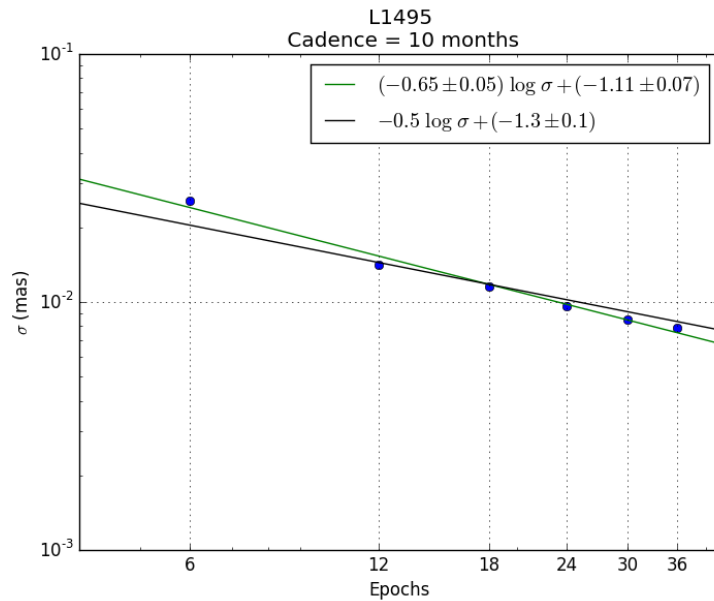


Figure 3.9: Relation of epochs and σ for cadence of 10 months, each blue point represent a value of σ , the green line is the linear adjust that passes through the points, and the black line is the linear adjust that passes through the points with a slope of -0.5 .

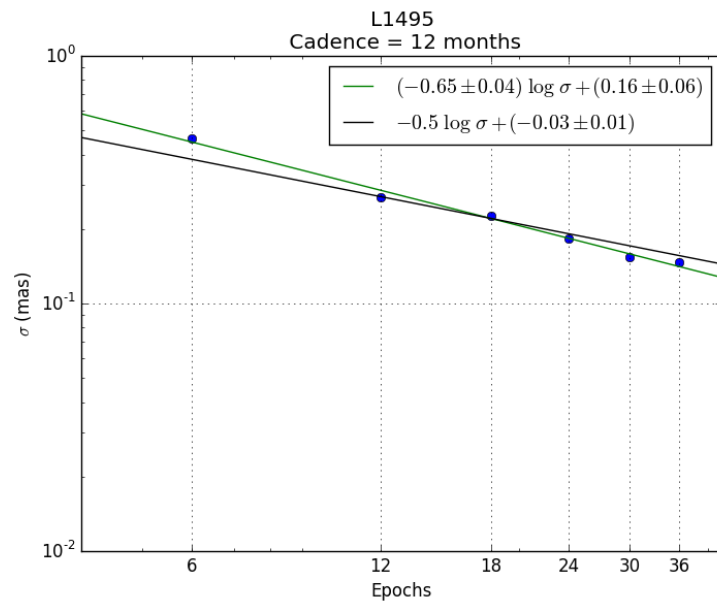


Figure 3.10: Relation of epochs and σ for cadence of 12 months, each blue point represent a value of σ , the green line is the linear adjust that passes through the points, and the black line is the linear adjust that passes through the points with a slope of -0.5 .

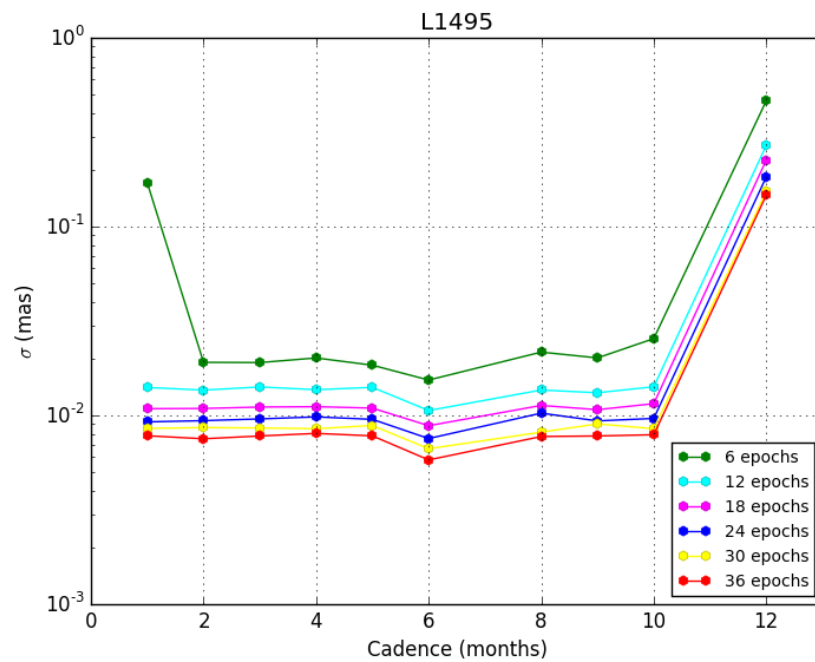


Figure 3.11: Cadence and Standard deviation(σ) for the L1495 case, plotted for different epochs.

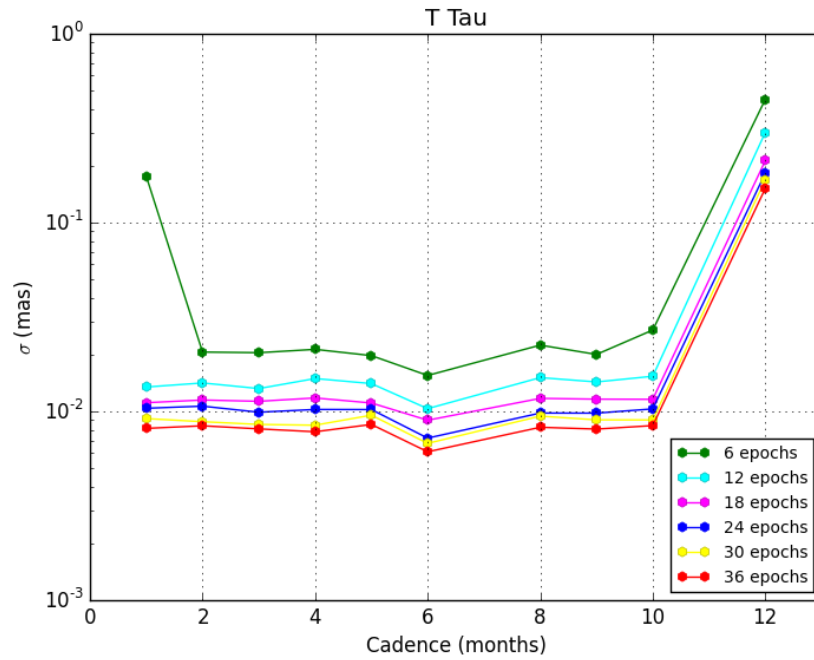


Figure 3.12: Cadence and Standard deviation(σ) for the T Tau case, plotted for different epochs.

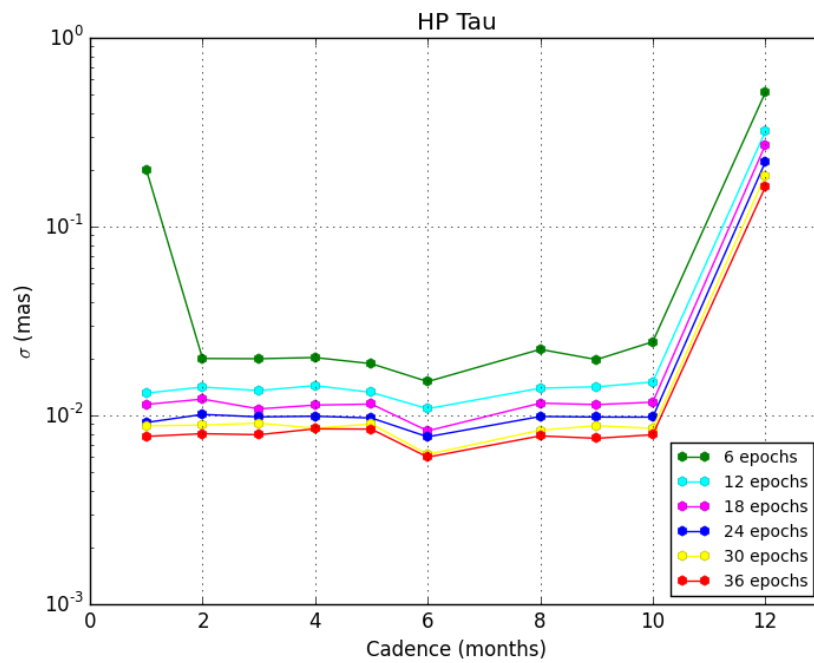


Figure 3.13: Cadence and Standard deviation(σ) for the HP Tau case, plotted for different epochs.

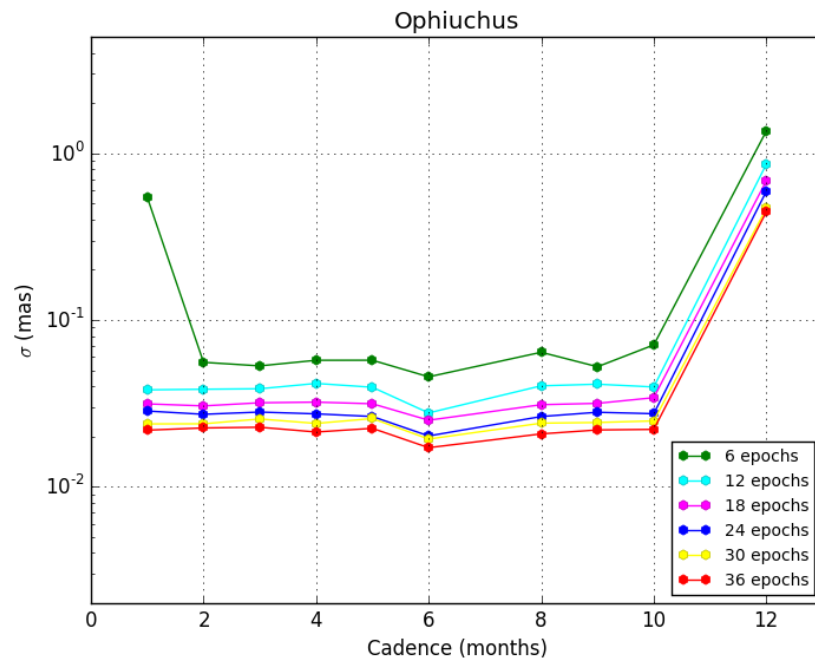


Figure 3.14: Cadence and Standard deviation(σ) for the Ophiuchus case, plotted for different epochs.

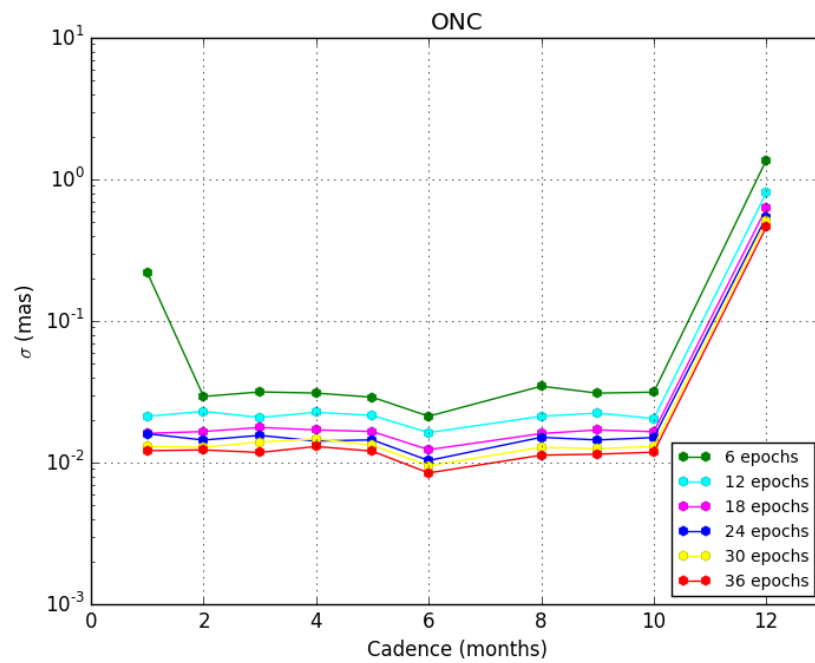


Figure 3.15: Cadence and Standard deviation(σ) for the ONC case, plotted for different epochs.

Chapter 4

Conclusions

Given the large variety of scheduling strategies used by different authors to measure parallaxes from VLBI observations, we decided to make a simulation and see how the quality of the parallax determination differs with different scheduling strategies. The simulation was done in the programming language python, and works for any Astro Elements (position, parallax, and proper motion) given as initial values. Specifically, here we applied our code to five different cases that correspond to the star-forming regions Taurus (Taurus C, Taurus S and Taurus E), Ophiuchus and Orion. These regions cover ranges from $\sim -25^\circ$ to $\sim +29^\circ$ in declination.

The simulation calculated the parallax, $i = 1000$ times¹ using a Monte Carlo Method, for every combination of cadence and number of epochs. From the ϖ_i obtained, we calculated the standard deviation (σ) from the mean, and used it as an estimate of the observations errors that would be obtained for each strategy.

Then we proceeded to plot the results in two different ways: examining the relation between σ and the number of epochs and between σ and the cadence of observation. From the first type of plots, a relation between the number of epochs and σ was found:

¹Initially, we tested the simulation for 100 realizations, then with 200, 300 and 500. We noted that from 300, the results were the same. So we decided to do a final run with 1000 realizations.

σ decreases with the square root the number of epochs observed, as expected from standard error propagation. From the second type of plot we noted an important decrease of σ by doing the observations every 6 months, and we conclude that this is the best way to schedule parallax measurement observations.

Although this work does not change the results from the past authors, it will help for future observations for VLBI parallax measurements. VLBI observations for parallax are very important in astrometry, and they will remain important. For instance, the Next Generation Very Large Array (ngVLA) is expected to reach an astrometric of 1 μas (one order of magnitude better than the GAIA satellite; Reid et al., 2018). The tool that we have developed here, perhaps adapted to include sources of errors proper to the ngVLA, will be very useful to prepare and schedule astrometric ngVLA observations. More generally, our tool can help future observers decide on the best observing strategy for astrometric observations of their source of choice. Since telescope time is very valuable, this work will help to use it in an advantageous way.

Finally, we intend to develop our tool further to examine, for instance, the effect of randomly losing some observations either to bad weather conditions or because a specific target becomes too weak to be detected at some epochs (non-thermal radio emission from young stars tends to be highly variable, so it is common that it is only detected at some fraction of the observed epochs). It will be worthwhile confirming whether a cadence of six months remains the best strategy in such cases, or whether a different strategy (for instance with some redundancy) should be adopted.

Bibliography

- Bourda, G., Collioud, A., Charlot, P., Porcas, R., and Garrington, S. (2011). VLBI observations of optically-bright extragalactic radio sources for the alignment of the radio frame with the future Gaia frame. II. Imaging candidate sources. *Astronomy and Astrophysics*, 526:A102.
- Brisken, W. F., Benson, J. M., Goss, W. M., and Thorsett, S. E. (2002). Very Long Baseline Array Measurement of Nine Pulsar Parallaxes. *The Astrophysical Journal*, 571:906–917.
- Condon, J. J. (1997). Errors in Elliptical Gaussian FITS. *Publications of the Astronomical Society of the Pacific*, 109:166–172.
- Deller, A. T., Weisberg, J. M., Nice, D. J., and Chatterjee, S. (2018). A VLBI Distance and Transverse Velocity for PSR B1913+16. *The Astrophysical Journal*, 862:139.
- Draine, B. T. (2006). On the Submillimeter Opacity of Protoplanetary Disks. *The Astrophysical Journal*, 636(2):1114–1120.
- Galli, P. A. B., Loinard, L., Ortiz-Léon, G. N., Kounkel, M., Dzib, S. A., Mioduszewski, A. J., Rodríguez, L. F., Hartmann, L., Teixeira, R., Torres, R. M., Rivera, J. L., Boden, A. F., Evans, Neal J., I., Briceño, C., Tobin, J. J., and Heyer, M. (2018). The Gould’s Belt Distances Survey (GOBELINS). IV. Distance, Depth, and Kinematics of the Taurus Star-forming Region. *The Astrophysical Journal*, 859(1):33.

- Gentle, J. E. (2010). *Computational Statistics*, volume 4 of 10. Elsevier Science, 3 edition.
- Kaplan, G., Bartlett, J. L., Monet, A., Bangert, J., Puatua, W., Harris, W., Fredericks, A., Barron, E. G., and Barrett, P. (2012). NOVAS: Naval Observatory Vector Astrometry Software.
- Kounkel, M., Hartmann, L., Loinard, L., Ortiz-León, G. N., Mioduszewski, A. J., Rodríguez, L. F., Dzib, S. A., Torres, R. M., Pech, G., Galli, P. A. B., Rivera, J. L., Boden, A. F., Evans, II, N. J., Briceño, C., and Tobin, J. J. (2017). The Gould's Belt Distances Survey (GOBELINS) II. Distances and Structure toward the Orion Molecular Clouds. *The Astrophysical Journal*, 834:142.
- Lindegren, L. and de Bruijne, J. H. J. (2005). *Performance of the Gaia mission*, volume 338 of *Astronomical Society of the Pacific Conference Series*, page 25.
- Loinard, L. (2014). Distances to star forming regions. *arXiv e-prints*, page arXiv:1410.3306.
- Loinard, L., Torres, R. M., Mioduszewski, A. J., Rodríguez, L. F., González-Lópezlira, R. A., Lachaume, R., Vázquez, V., and González, E. (2007). VLBA Determination of the Distance to Nearby Star-forming Regions. I. The Distance to T Tauri with 0.4% Accuracy. *The Astrophysical Journal*, 671:546–554.
- Menten, K. M., Reid, M. J., Forbrich, J., and Brunthaler, A. (2007). The distance to the Orion Nebula. *Astronomy and Astrophysics*, 474(2):515–520.
- Mignard, F. (2005). *The Gaia Mission: Science Highlights*, volume 338 of *Astronomical Society of the Pacific Conference Series*, page 15.

- Murthy, K. P. N. (2001). An introduction to monte carlo simulation of statistical physics problem.
- Ortiz-León, G. N., Loinard, L., Dzib, S. A., Galli, P. A. B., Kounkel, M., Mioduszewski, A. J., Rodríguez, L. F., Torres, R. M., Hartmann, L., Boden, A. F., Evans, Neal J., I., Briceño, C., and Tobin, J. J. (2018). The Gould’s Belt Distances Survey (GOBELINS). V. Distances and Kinematics of the Perseus Molecular Cloud. *The Astrophysical Journal*, 865(1):73.
- Ortiz-León, G. N., Loinard, L., Kounkel, M. A., Dzib, S. A., Mioduszewski, A. J., Rodríguez, L. F., Torres, R. M., González-Lópezlira, R. A., Pech, G., Rivera, J. L., Hartmann, L., Boden, A. F., Evans, II, N. J., Briceño, C., Tobin, J. J., Galli, P. A. B., and Gudehus, D. (2017). The Gould’s Belt Distances Survey (GOBELINS). I. Trigonometric Parallax Distances and Depth of the Ophiuchus Complex. *The Astrophysical Journal*, 834:141.
- Pradel, N., Charlot, P., and Lestrade, J.-F. (2006). Astrometric accuracy of phase-referenced observations with the VLBA and EVN. *Astronomy and Astrophysics*, 452:1099–1106.
- Reid, M., Loinard, L., and Maccarone, T. (2018). Astrometry and Long Baseline Science. In Murphy, E., editor, *Science with a Next Generation Very Large Array*, volume 517 of *Astronomical Society of the Pacific Conference Series*, page 523.
- Reid, M. J. and Honma, M. (2014). Microarcsecond Radio Astrometry. *The Annual Review of Astronomy and Astrophysics*, 52:339–372.
- Reid, M. J., Menten, K. M., Brunthaler, A., Zheng, X. W., Moscadelli, L., and Xu, Y. (2009a). Trigonometric Parallaxes of Massive Star-Forming Regions. I. S 252 & G232.6+1.0. *The Astrophysical Journal*, 693:397–405.

- Reid, M. J., Menten, K. M., Zheng, X. W., Brunthaler, A., and Xu, Y. (2009b). A Trigonometric Parallax of Sgr B2. *The Astrophysical Journal*, 705(2):1548–1553.
- Rodríguez, L. F. (1984). Formación y procesamiento de imágenes en la interferometría radioastronómica. *Ciencia*, 35:163–173.
- Seidelmann, P. K. (1992). *Explanatory Supplement to the Astronomical Almanac*. Mill Valley: University Science Books.
- Selina, R. J., Murphy, E. J., McKinnon, M., Beasley, A., Butler, B., Carilli, C., Clark, B., Erickson, A., Grammer, W., Jackson, J., Kent, B., Mason, B., Morgan, M., Ojeda, O., Shillue, W., Sturgis, S., and Urbain, D. (2018). The Next-Generation Very Large Array: a technical overview. volume 10700 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, page 107001O.
- Torres, R. M. (2009). *Midiendo Regiones de Formación Estelar Cercanas con el VLBA: de la Distancia a la Dinámica*. PhD thesis, UNAM.
- Torres, R. M., Loinard, L., Mioduszewski, A. J., Boden, A. F., Franco-Hernández, R., Vlemmings, W. H. T., and Rodríguez, L. F. (2012). VLBA Determination of the Distance to nearby Star-forming Regions. V. Dynamical Mass, Distance, and Radio Structure of V773 Tau A. *The Astrophysical Journal*, 747:18.
- Torres, R. M., Loinard, L., Mioduszewski, A. J., and Rodríguez, L. F. (2007). VLBA Determination of the Distance to Nearby Star-forming Regions. II. Hubble 4 and HDE 283572 in Taurus. *The Astrophysical Journal*, 671:1813–1819.
- Torres, R. M., Loinard, L., Mioduszewski, A. J., and Rodríguez, L. F. (2009). VLBA Determination of the Distance to Nearby Star-Forming Regions. III. HP TAU/G2 and the Three-Dimensional Structure of Taurus. *The Astrophysical Journal*, 698:242–249.

A T Tau

This appendix shows the plots of epochs vs standard deviation (σ) for the T Tau case.

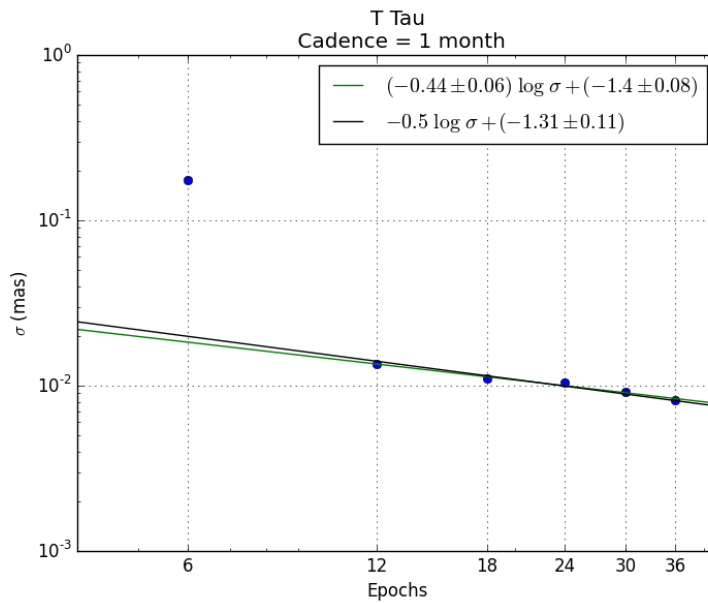


Figure 1: Relation of epochs and σ for cadence of 1 month, each blue point represent a value of σ , the green line is the linear adjust that passes through the points, and the black line is the linear adjust that passes through the points with a slope of -0.5 . Note: The blue point for 6 epochs was not taken into account for the fit.

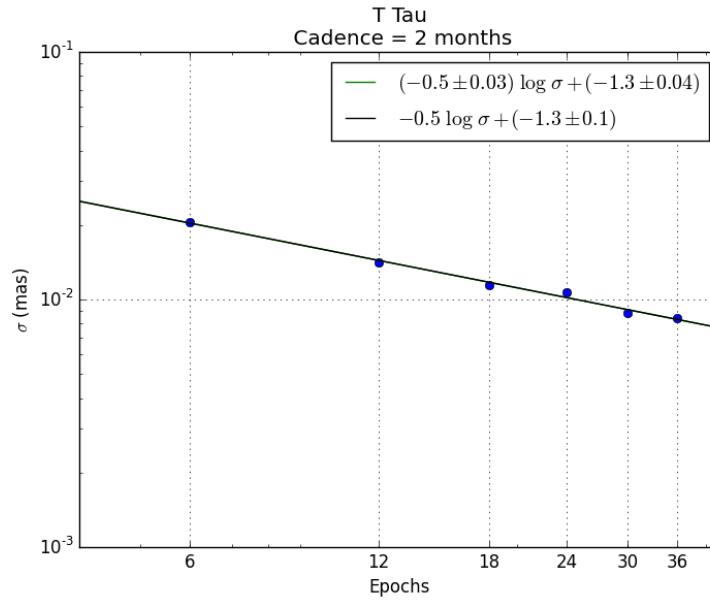


Figure 2: Relation of epochs and σ for cadence of 2 months, each blue point represent a value of σ , the green line is the linear adjust that passes through the points, and the black line is the linear adjust that passes through the points with a slope of -0.5 .

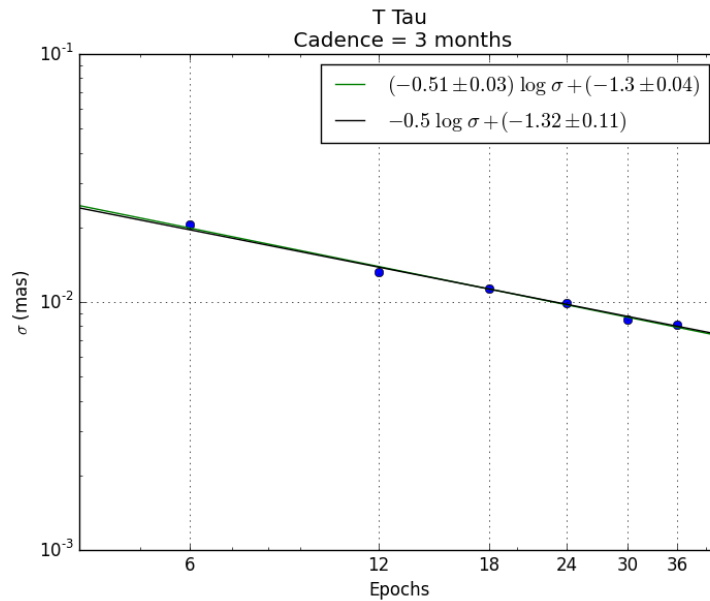


Figure 3: Relation of epochs and σ for cadence of 3 months, each blue point represent a value of σ , the green line is the linear adjust that passes through the points, and the black line is the linear adjust that passes through the points with a slope of -0.5 .

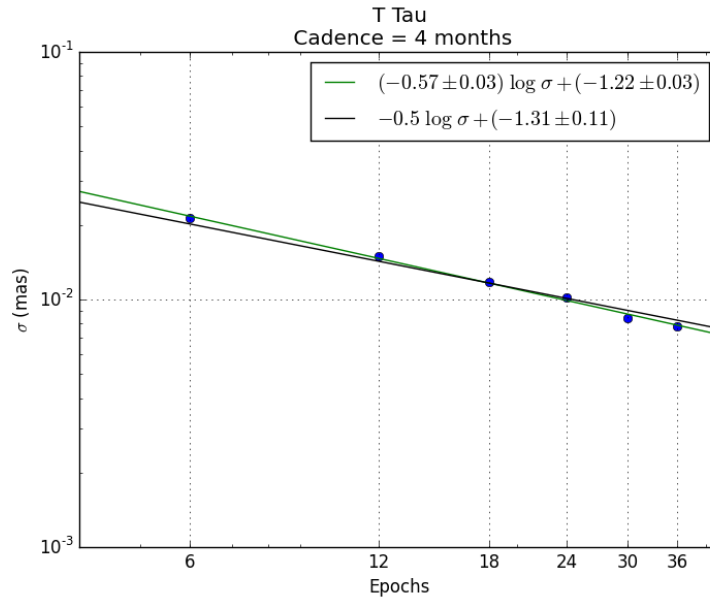


Figure 4: Relation of epochs and σ for cadence of 4 months, each blue point represent a value of σ , the green line is the linear adjust that passes through the points, and the black line is the linear adjust that passes through the points with a slope of -0.5 .

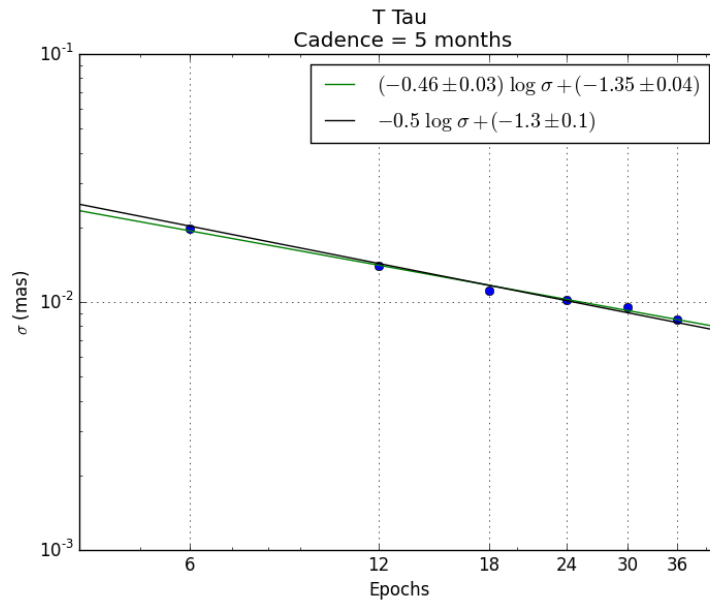


Figure 5: Relation of epochs and σ for cadence of 5 months, each blue point represent a value of σ , the green line is the linear adjust that passes through the points, and the black line is the linear adjust that passes through the points with a slope of -0.5 .

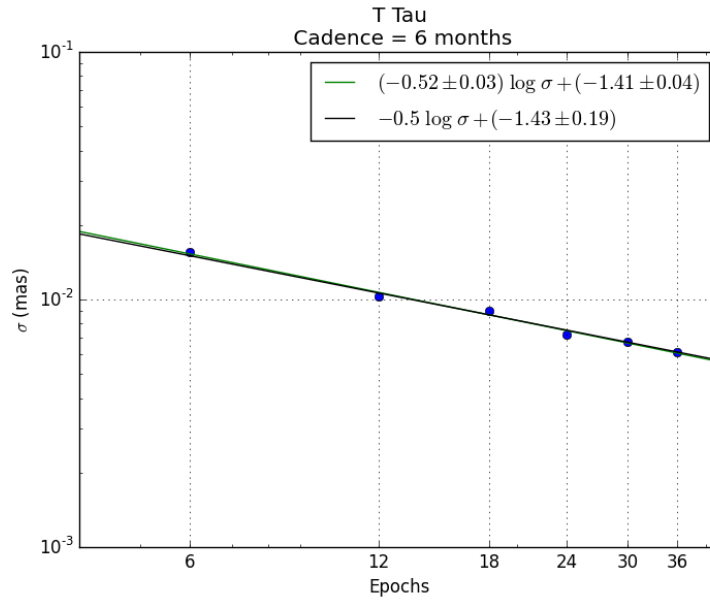


Figure 6: Relation of epochs and σ for cadence of 6 months, each blue point represent a value of σ , the green line is the linear adjust that passes through the points, and the black line is the linear adjust that passes through the points with a slope of -0.5 .

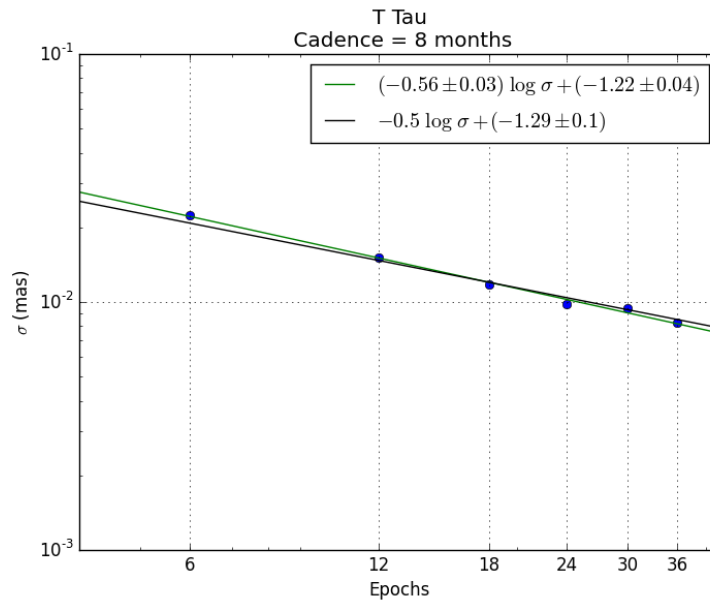


Figure 7: Relation of epochs and σ for cadence of 8 months, each blue point represent a value of σ , the green line is the linear adjust that passes through the points, and the black line is the linear adjust that passes through the points with a slope of -0.5 .

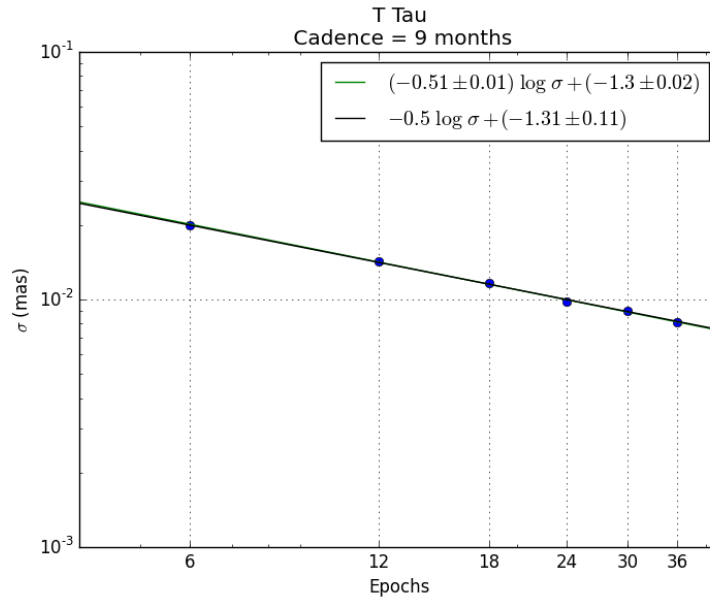


Figure 8: Relation of epochs and σ for cadence of 9 months, each blue point represent a value of σ , the green line is the linear adjust that passes through the points, and the black line is the linear adjust that passes through the points with a slope of -0.5 .

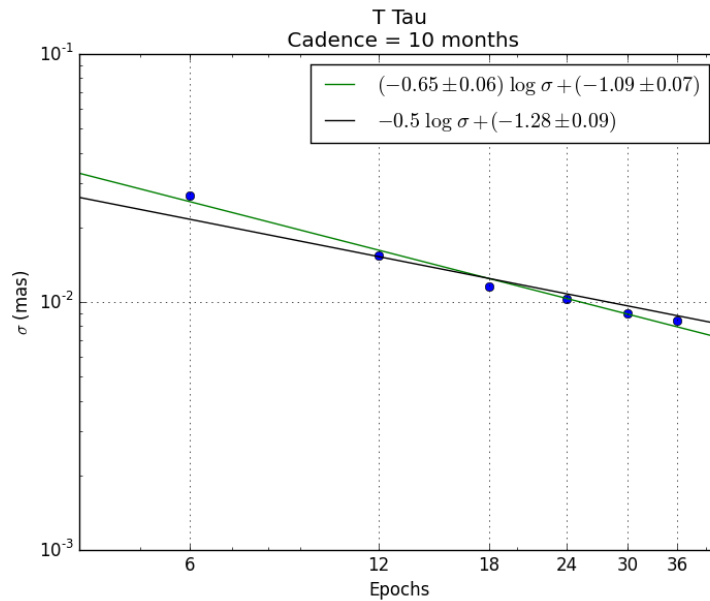


Figure 9: Relation of epochs and σ for cadence of 10 months, each blue point represent a value of σ , the green line is the linear adjust that passes through the points, and the black line is the linear adjust that passes through the points with a slope of -0.5 .

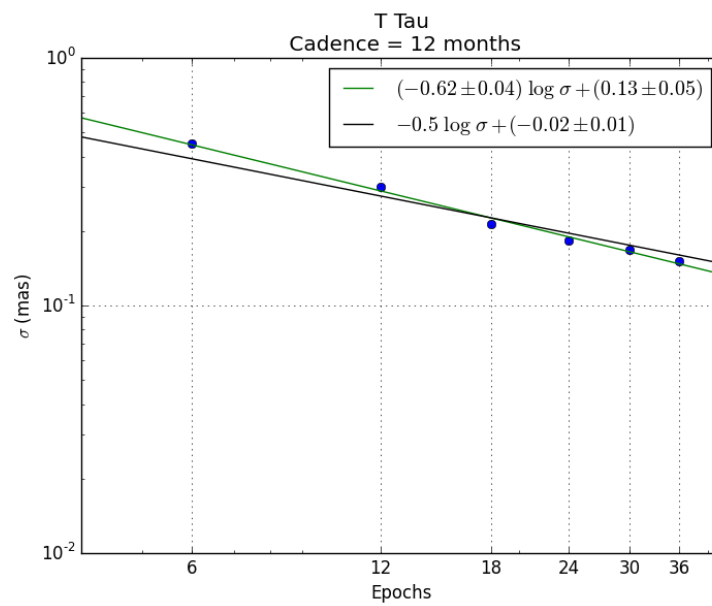


Figure 10: Relation of epochs and σ for cadence of 12 months, each blue point represent a value of σ , the green line is the linear adjust that passes through the points, and the black line is the linear adjust that passes through the points with a slope of -0.5 .

B HP Tau

This appendix shows the plots of epochs vs standard deviation (σ) for the HP Tau case.

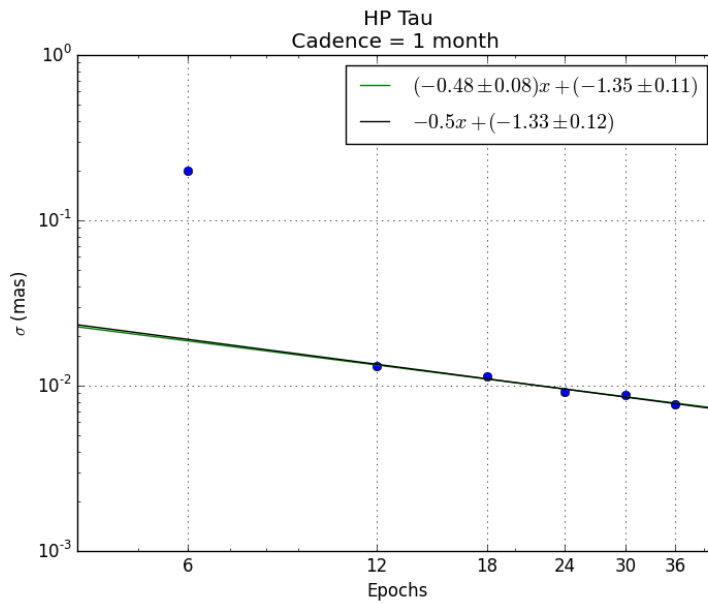


Figure 11: Relation of epochs and σ for cadence of 1 month, each blue point represent a value of σ , the green line is the linear adjust that passes through the points, and the black line is the linear adjust that passes through the points with a slope of -0.5 . Note: The blue point for 6 epochs was not taken into account for the fit.

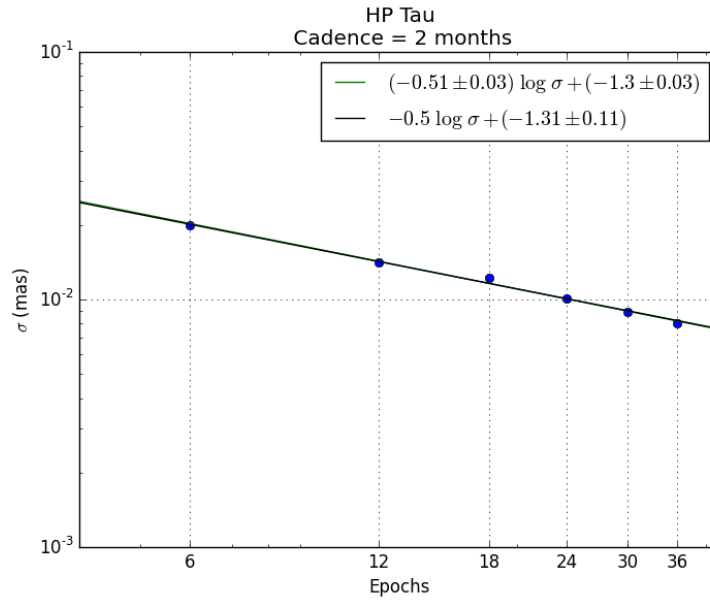


Figure 12: Relation of epochs and σ for cadence of 2 months, each blue point represent a value of σ , the green line is the linear adjust that passes through the points, and the black line is the linear adjust that passes through the points with a slope of -0.5 .

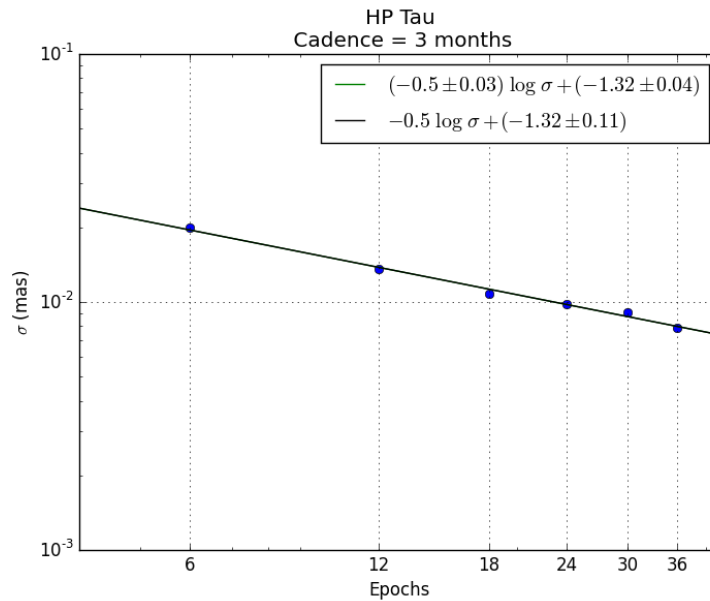


Figure 13: Relation of epochs and σ for cadence of 3 months, each blue point represent a value of σ , the green line is the linear adjust that passes through the points, and the black line is the linear adjust that passes through the points with a slope of -0.5 .

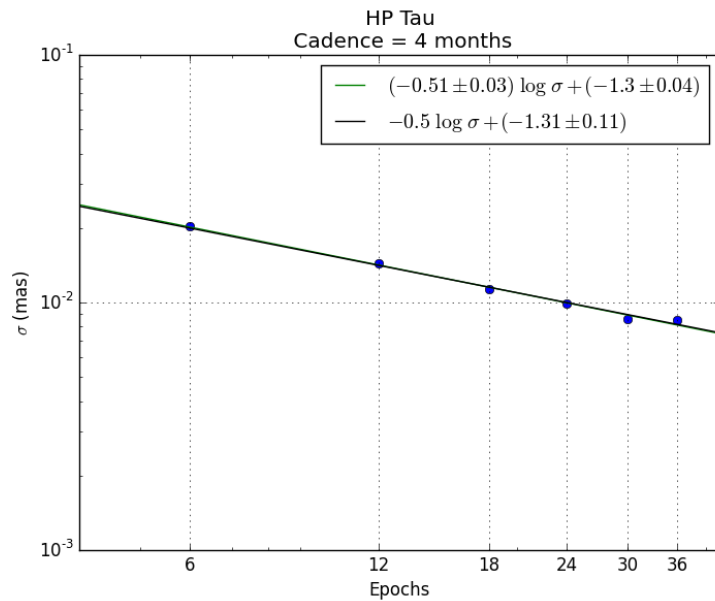


Figure 14: Relation of epochs and σ for cadence of 4 months, each blue point represent a value of σ , the green line is the linear adjust that passes through the points, and the black line is the linear adjust that passes through the points with a slope of -0.5 .

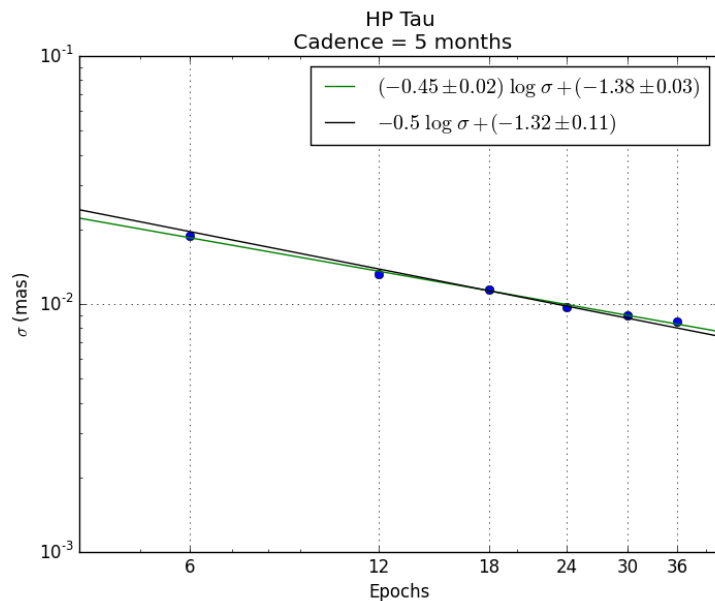


Figure 15: Relation of epochs and σ for cadence of 5 months, each blue point represent a value of σ , the green line is the linear adjust that passes through the points, and the black line is the linear adjust that passes through the points with a slope of -0.5 .

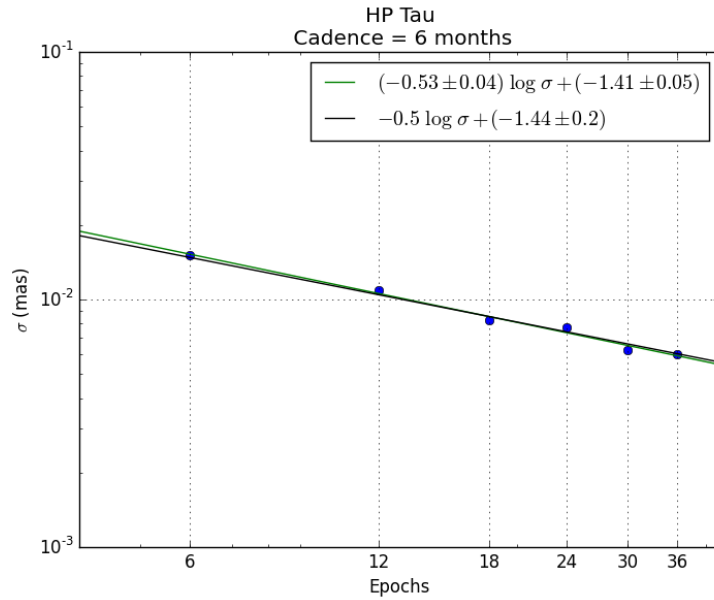


Figure 16: Relation of epochs and σ for cadence of 6 months, each blue point represent a value of σ , the green line is the linear adjust that passes through the points, and the black line is the linear adjust that passes through the points with a slope of -0.5 .

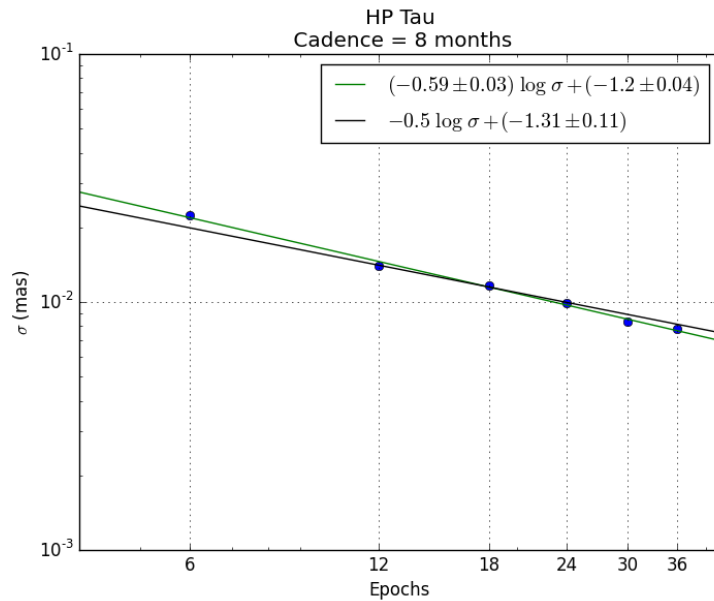


Figure 17: Relation of epochs and σ for cadence of 8 months, each blue point represent a value of σ , the green line is the linear adjust that passes through the points, and the black line is the linear adjust that passes through the points with a slope of -0.5 .

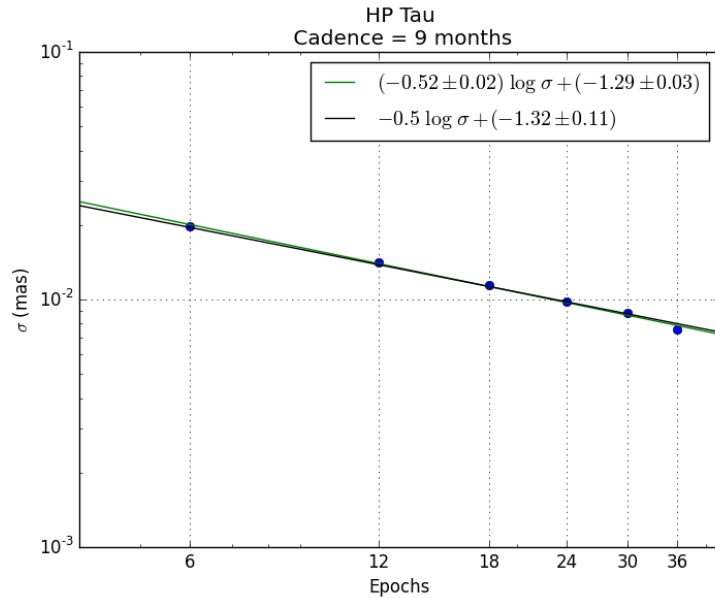


Figure 18: Relation of epochs and σ for cadence of 9 months, each blue point represent a value of σ , the green line is the linear adjust that passes through the points, and the black line is the linear adjust that passes through the points with a slope of -0.5 .

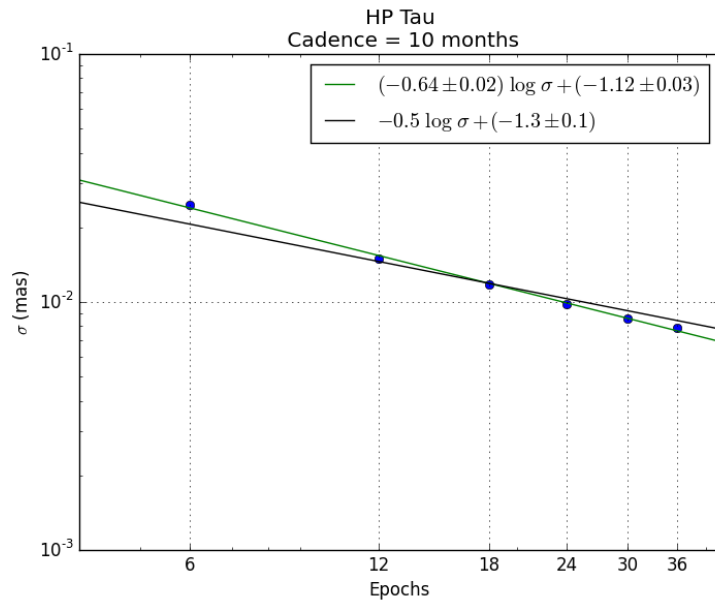


Figure 19: Relation of epochs and σ for cadence of 10 months, each blue point represent a value of σ , the green line is the linear adjust that passes through the points, and the black line is the linear adjust that passes through the points with a slope of -0.5 .

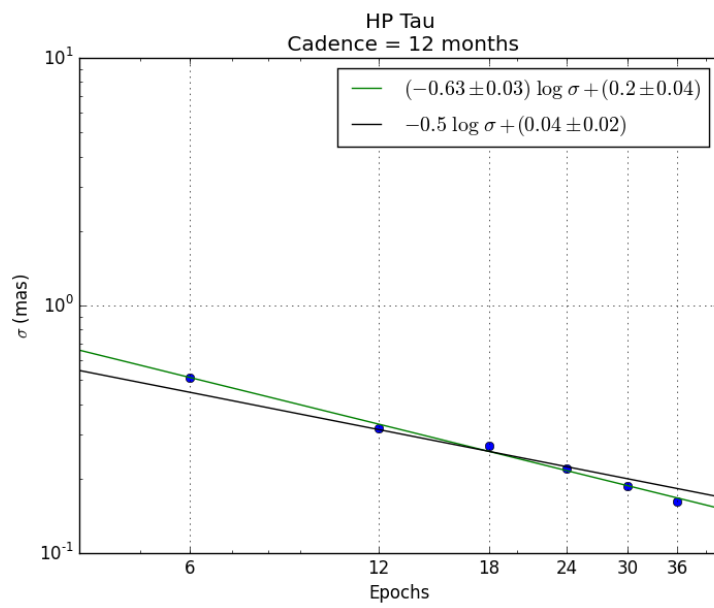


Figure 20: Relation of epochs and σ for cadence of 12 months, each blue point represent a value of σ , the green line is the linear adjust that passes through the points, and the black line is the linear adjust that passes through the points with a slope of -0.5 .

C Ophiuchus

This appendix shows the plots of epochs vs standard deviation (σ) for the Ophiuchus case.

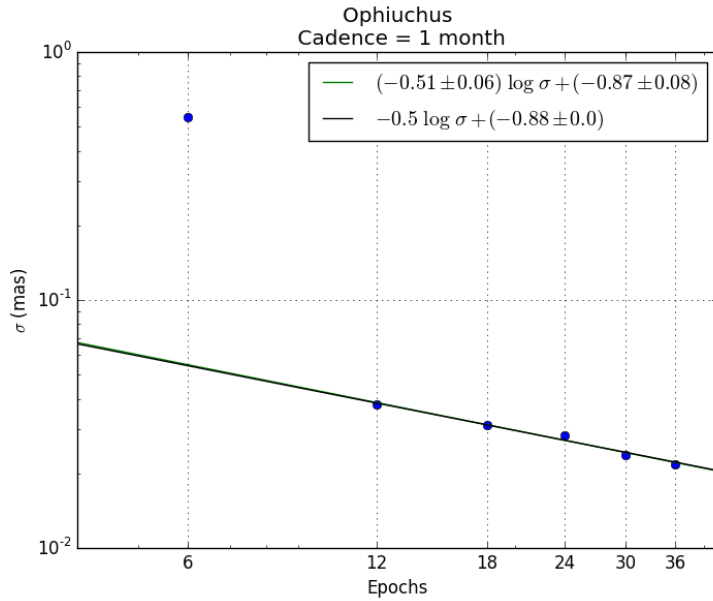


Figure 21: Relation of epochs and σ for cadence of 1 month, each blue point represent a value of σ , the green line is the linear adjust that passes through the points, and the black line is the linear adjust that passes through the points with a slope of -0.5 . Note: The blue point for 6 epochs was not taken into account for the fit.

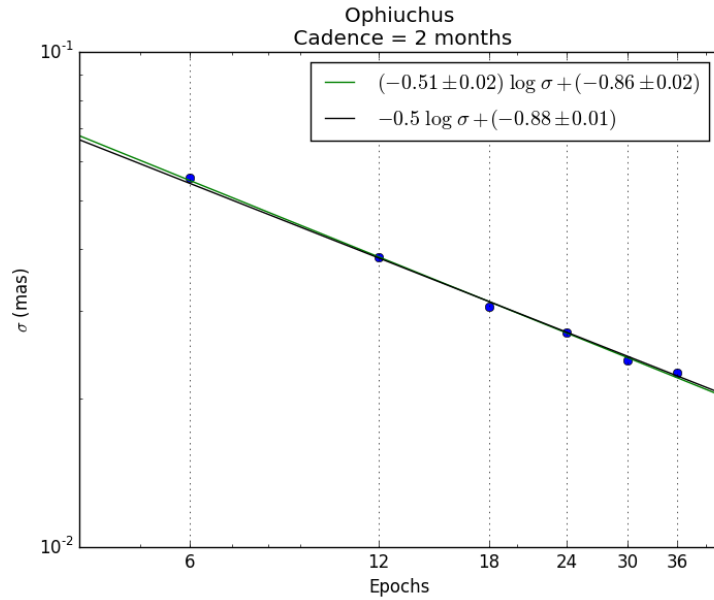


Figure 22: Relation of epochs and σ for cadence of 2 months, each blue point represent a value of σ , the green line is the linear adjust that passes through the points, and the black line is the linear adjust that passes through the points with a slope of -0.5 .

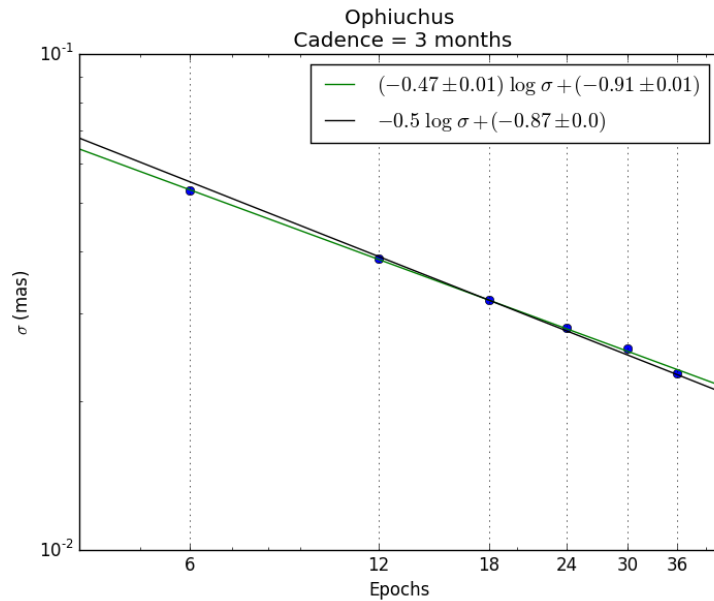


Figure 23: Relation of epochs and σ for cadence of 3 months, each blue point represent a value of σ , the green line is the linear adjust that passes through the points, and the black line is the linear adjust that passes through the points with a slope of -0.5 .

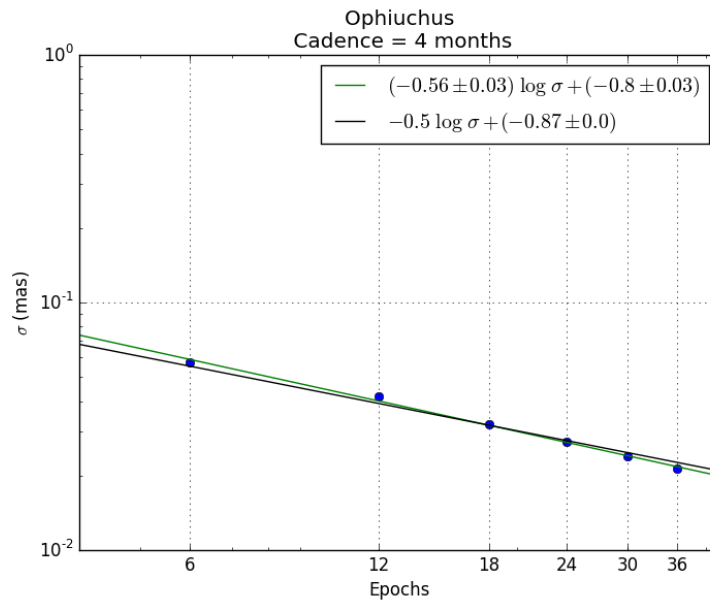


Figure 24: Relation of epochs and σ for cadence of 4 months, each blue point represent a value of σ , the green line is the linear adjust that passes through the points, and the black line is the linear adjust that passes through the points with a slope of -0.5 .

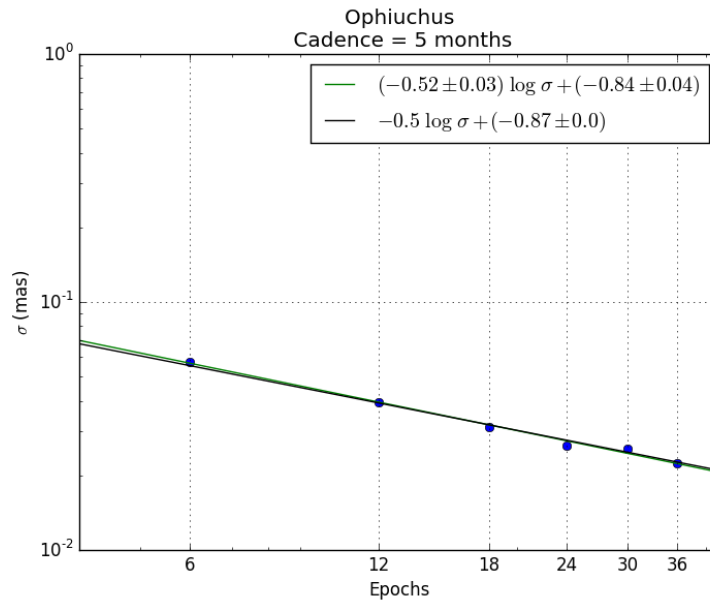


Figure 25: Relation of epochs and σ for cadence of 5 months, each blue point represent a value of σ , the green line is the linear adjust that passes through the points, and the black line is the linear adjust that passes through the points with a slope of -0.5 .

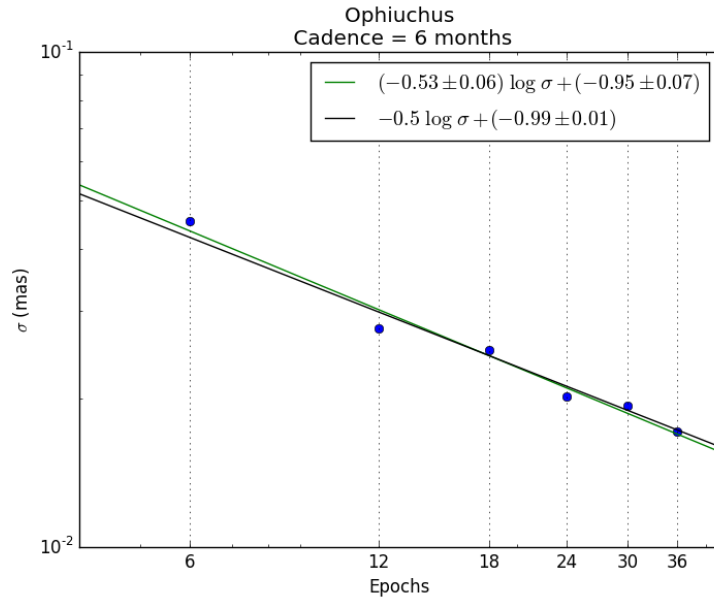


Figure 26: Relation of epochs and σ for cadence of 6 months, each blue point represent a value of σ , the green line is the linear adjust that passes through the points, and the black line is the linear adjust that passes through the points with a slope of -0.5 .

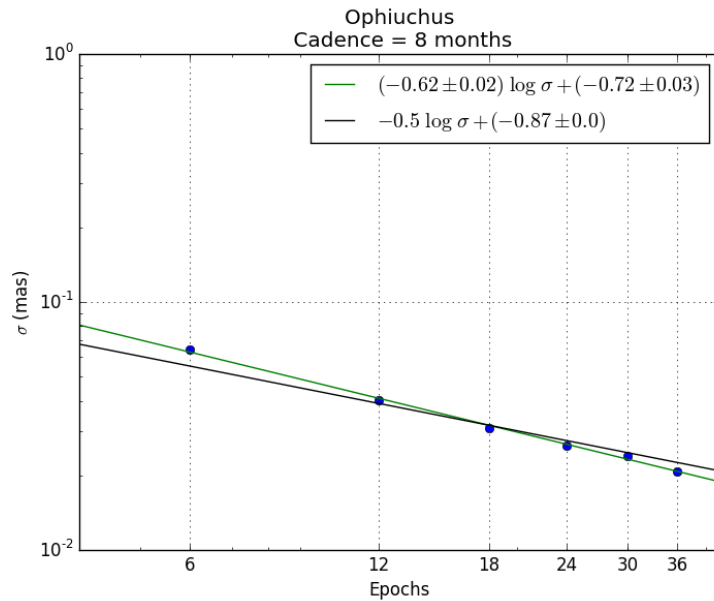


Figure 27: Relation of epochs and σ for cadence of 8 months, each blue point represent a value of σ , the green line is the linear adjust that passes through the points, and the black line is the linear adjust that passes through the points with a slope of -0.5 .

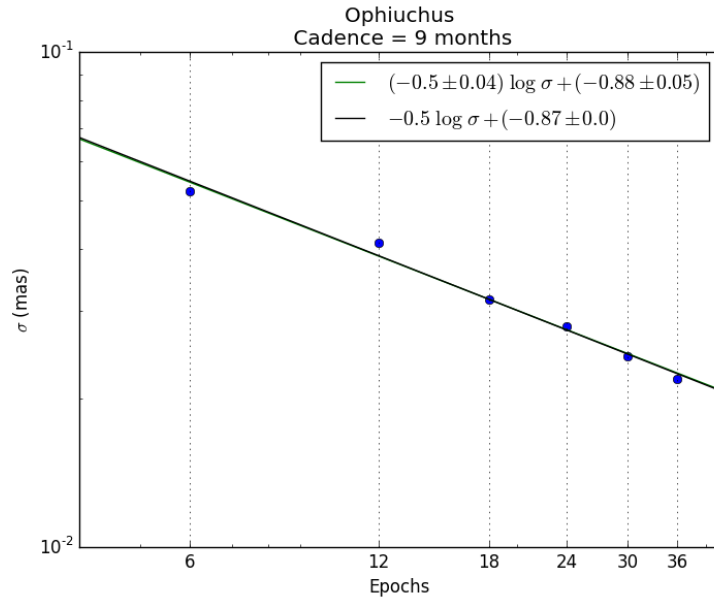


Figure 28: Relation of epochs and σ for cadence of 9 months, each blue point represent a value of σ , the green line is the linear adjust that passes through the points, and the black line is the linear adjust that passes through the points with a slope of -0.5 .

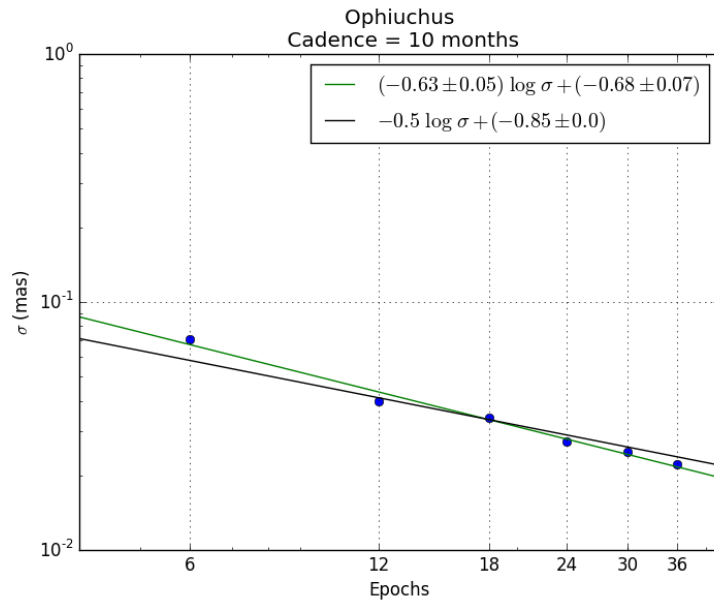


Figure 29: Relation of epochs and σ for cadence of 10 months, each blue point represent a value of σ , the green line is the linear adjust that passes through the points, and the black line is the linear adjust that passes through the points with a slope of -0.5 .

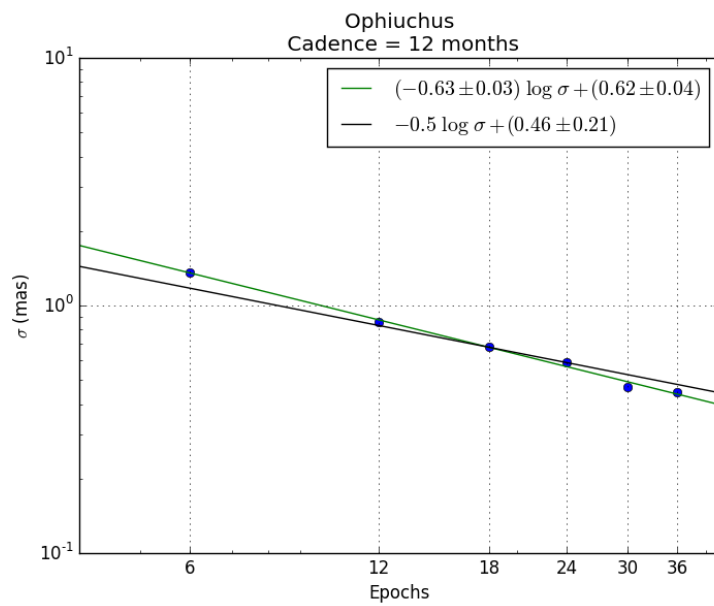


Figure 30: Relation of epochs and σ for cadence of 12 months, each blue point represent a value of σ , the green line is the linear adjust that passes through the points, and the black line is the linear adjust that passes through the points with a slope of -0.5 .

D ONC

This appendix shows the plots of epochs vs standard deviation (σ) for the ONC case.

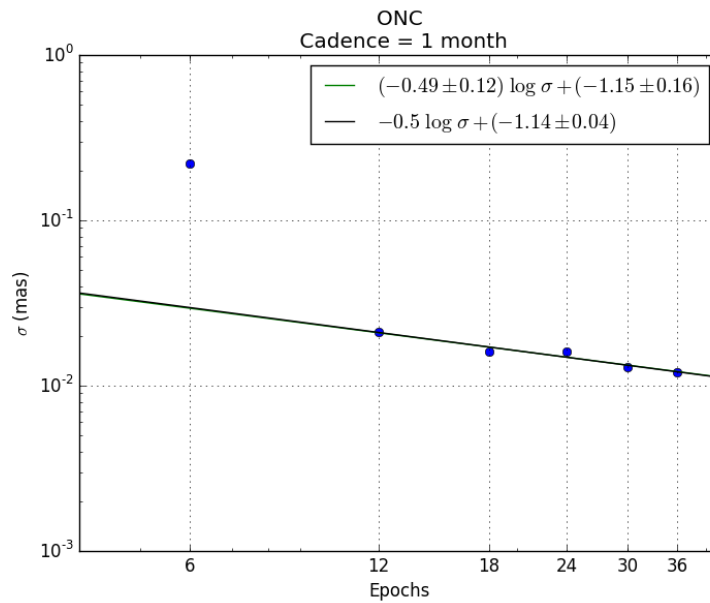


Figure 31: Relation of epochs and σ for cadence of 1 month, each blue point represent a value of σ , the green line is the linear adjust that passes through the points, and the black line is the linear adjust that passes through the points with a slope of -0.5 . Note: The blue point for 6 epochs was not taken into account for the fit.

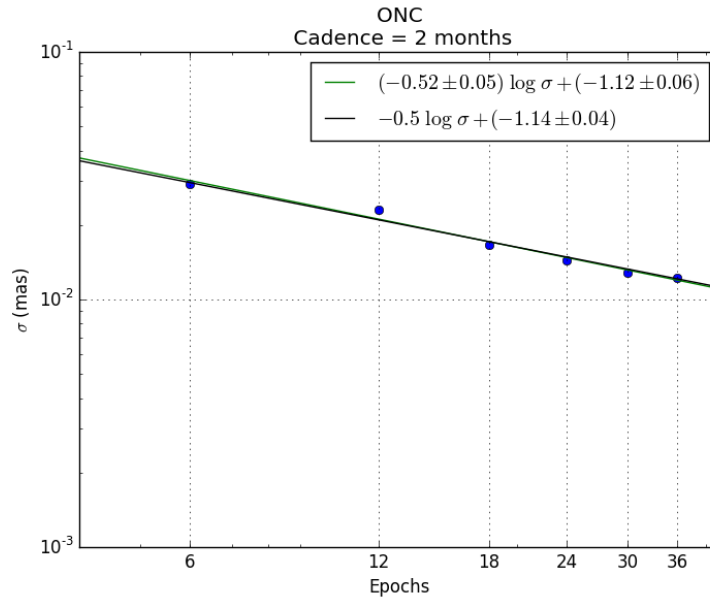


Figure 32: Relation of epochs and σ for cadence of 2 months, each blue point represent a value of σ , the green line is the linear adjust that passes through the points, and the black line is the linear adjust that passes through the points with a slope of -0.5 .

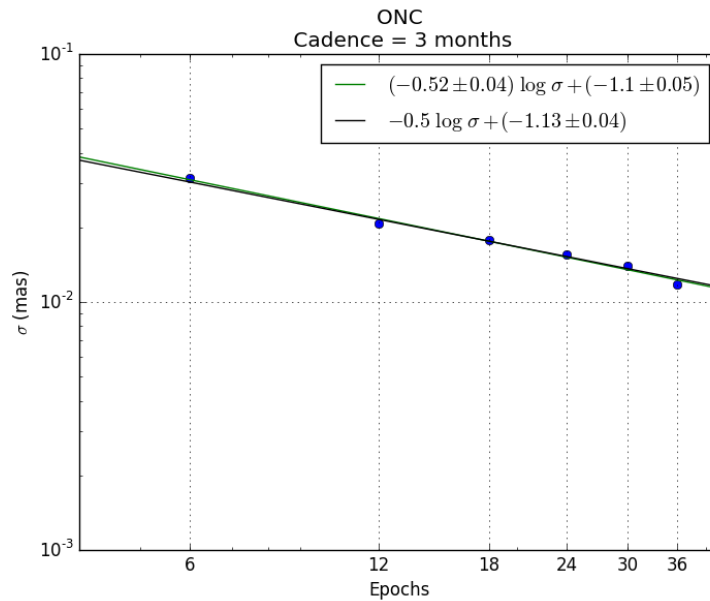


Figure 33: Relation of epochs and σ for cadence of 3 months, each blue point represent a value of σ , the green line is the linear adjust that passes through the points, and the black line is the linear adjust that passes through the points with a slope of -0.5 .

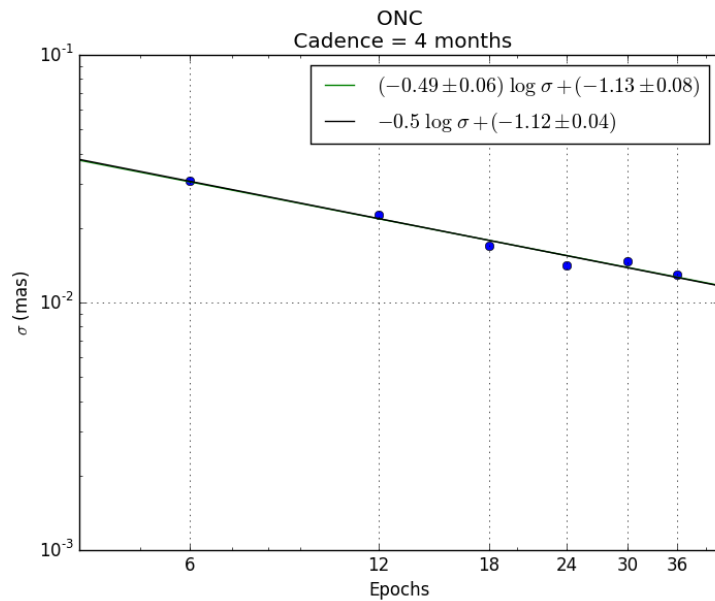


Figure 34: Relation of epochs and σ for cadence of 4 months, each blue point represent a value of σ , the green line is the linear adjust that passes through the points, and the black line is the linear adjust that passes through the points with a slope of -0.5 .

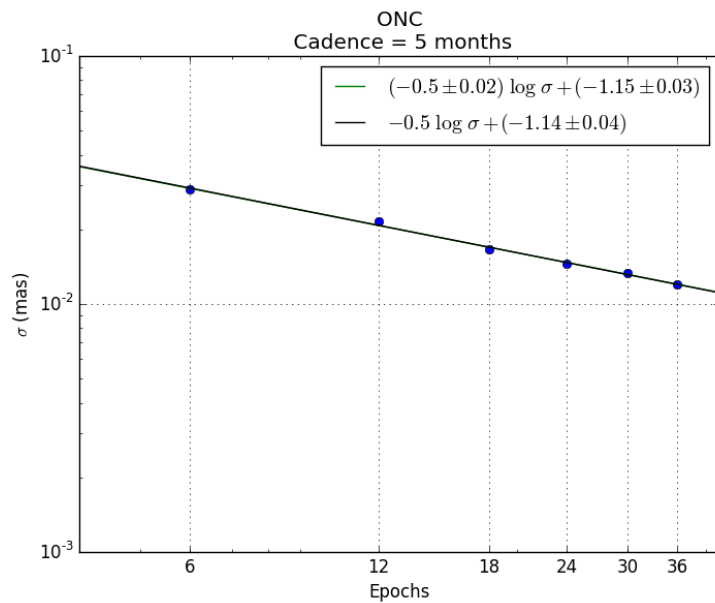


Figure 35: Relation of epochs and σ for cadence of 5 months, each blue point represent a value of σ , the green line is the linear adjust that passes through the points, and the black line is the linear adjust that passes through the points with a slope of -0.5 .

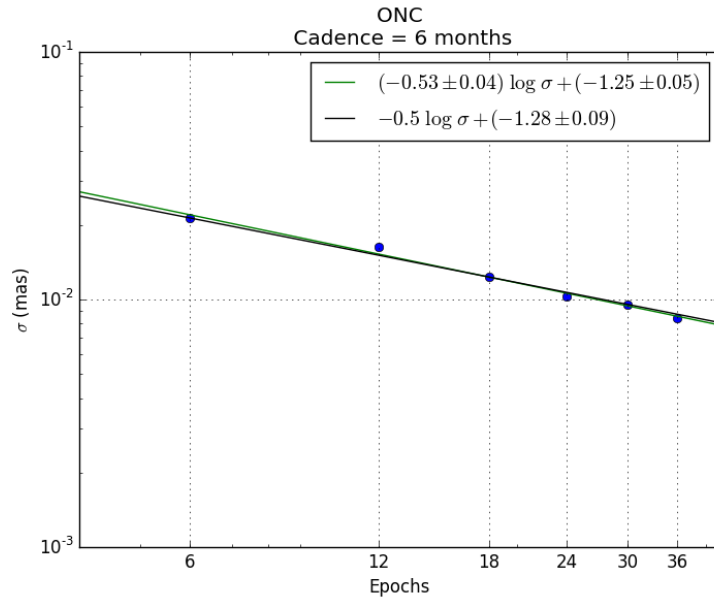


Figure 36: Relation of epochs and σ for cadence of 6 months, each blue point represent a value of σ , the green line is the linear adjust that passes through the points, and the black line is the linear adjust that passes through the points with a slope of -0.5 .

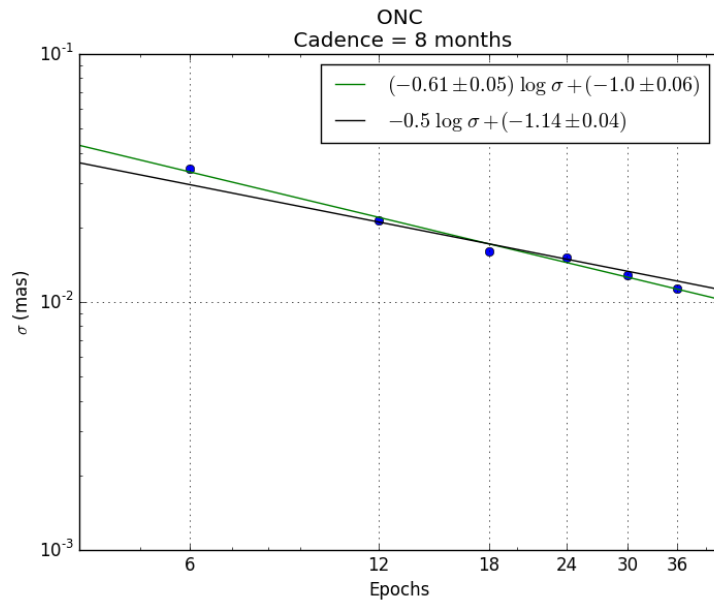


Figure 37: Relation of epochs and σ for cadence of 8 months, each blue point represent a value of σ , the green line is the linear adjust that passes through the points, and the black line is the linear adjust that passes through the points with a slope of -0.5 .

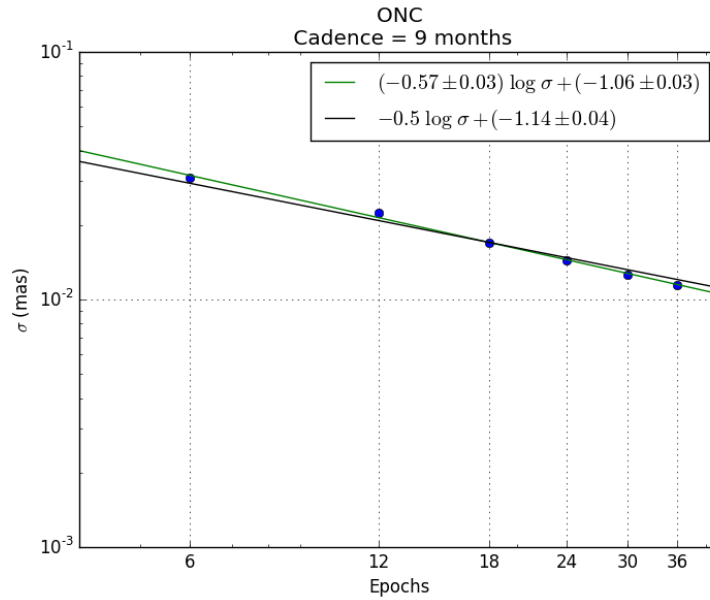


Figure 38: Relation of epochs and σ for cadence of 9 months, each blue point represent a value of σ , the green line is the linear adjust that passes through the points, and the black line is the linear adjust that passes through the points with a slope of -0.5 .

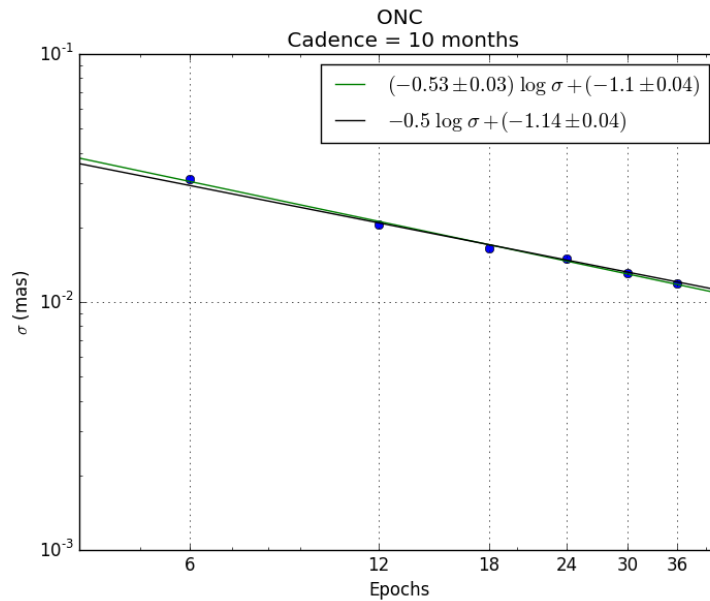


Figure 39: Relation of epochs and σ for cadence of 10 months, each blue point represent a value of σ , the green line is the linear adjust that passes through the points, and the black line is the linear adjust that passes through the points with a slope of -0.5 .

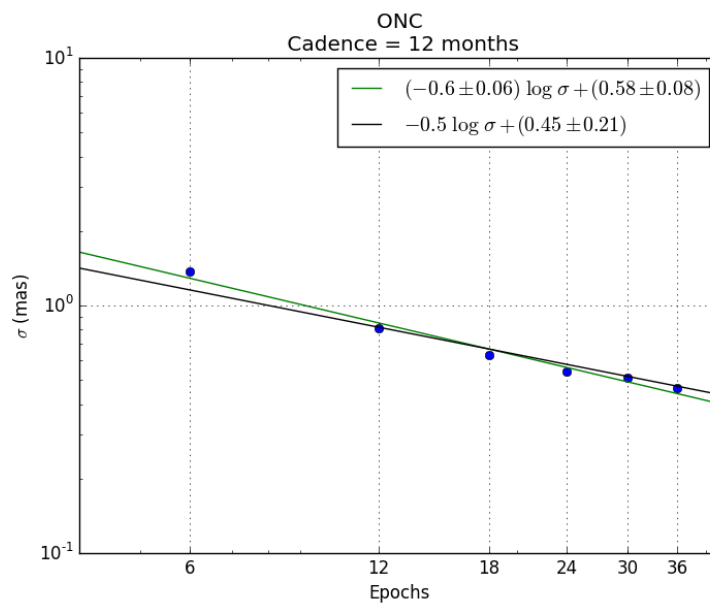


Figure 40: Relation of epochs and σ for cadence of 12 months, each blue point represent a value of σ , the green line is the linear adjust that passes through the points, and the black line is the linear adjust that passes through the points with a slope of -0.5 .

E Astrometry Simulation Code

This appendix shows the code made for the simulation, is divided in two parts. The first part is to enter the initial parameters for each case. The second part is the code that contains all the functions needed to run the simulation.

First part:

```
import numpy as np
import astrometry1000 as ast

## Here enter user specific stuff
user = 'user_name'
months,ephFile,year = ast.load_ephem(user)
print "You are using the ephemeris file",ephFile

case = 'L1495'

if case == "ONC":
    RA_0_str = '05:41:29.9' # HH:MM:SS.SS
    Dec_0_str = '-03:49:22.5' # DD:MM:SS.SS
    RA_0 = ast.str2decimal(RA_0_str,True)
    Dec_0 = ast.str2decimal(Dec_0_str,False)
    muRA = 0.06
    muDec = -0.12
    prlx = 2.41

if case == "Ophiuchus":
    RA_0_str = '16:29:30.19'
    Dec_0_str = '-24:39:42.1'
    RA_0 = ast.str2decimal(RA_0_str,True)
```



```

Dec_0 = ast.str2decimal(Dec_0_str,False)
muRA = -6.5229
muDec = -25.0564
prlx = 7.01

if case == "HP_Tau":
    RA_0_str = '04:35:54.16'
    Dec_0_str = '22:54:13.49'
    RA_0 = ast.str2decimal(RA_0_str,True)
    Dec_0 = ast.str2decimal(Dec_0_str,False)
    muRA = 13.85
    muDec = -15.4
    prlx = 6.20

if case == "T_Tau":
    RA_0_str = '04:21:59.42'
    Dec_0_str = '19:32:05.71'
    RA_0 = ast.str2decimal(RA_0_str,True)
    Dec_0 = ast.str2decimal(Dec_0_str,False)
    muRA = 4.02
    muDec = -1.18
    prlx = 6.82

if case == "L1495":
    RA_0_str = '4:18:19.6'
    Dec_0_str = '28:16:48.6'
    RA_0 = ast.str2decimal(RA_0_str,True)
    Dec_0 = ast.str2decimal(Dec_0_str,False)
    muRA = 10.0906666667
    muDec = -26.51
    prlx = 7.67

astro_elements = (RA_0,Dec_0,muRA,muDec,prlx)
results = ast.completeSimulation(user,astro_elements)
np.savetxt("results_L1495_1000.txt", results)

```

Second part:

```

from novas import compat as novas
from novas.compat import eph_manager
from novas.compat import solsys

```

```

import numpy as np
import time
import matplotlib.pyplot as plt
import random
from scipy.optimize import curve_fit
from kapteyn import wcs

year = 365.25

def load_ephem(user):
    """Load the ephemeris file for NOVAS and run some tests on it;
    also, define a python dictionary for months that will work always
    (english or spanish depending on your local version of NOVAS)"""

    if user == 'user_name1':
        ephFile = '/home/gely/Enthought/Canopy_64bit/User/lib/python2.7/ \
            site-packages/DE405.bin'
        months = {1 : 'ENE' , 2 : 'FEB' , 3 : 'MAR' , 4 : 'ABR' ,
                  5 : 'MAY' , 6 : 'JUN' , 7 : 'JUL' , 8 : 'AGO' ,
                  9 : 'SEP' , 10 : 'OCT' , 11 : 'NOV' , 12 : 'DIC'}

    elif user == 'user_name2':
        ephFile = '/Users/lloinard/Library/Enthought/Canopy_64bit/User/ \
            lib/python2.7/site-packages/novas_de405/DE405.bin'
        months = {1 : 'JAN' , 2 : 'FEB' , 3 : 'MAR' , 4 : 'APR' ,
                  5 : 'MAY' , 6 : 'JUN' , 7 : 'JUL' , 8 : 'AUG' ,
                  9 : 'SEP' , 10 : 'OCT' , 11 : 'NOV' , 12 : 'DEC'}

    jd_start, jd_end, number = eph_manager.ephem_open(ephFile)
    year = 365.25
    return months,ephFile,year

def calc_jd(date_str):
    """Calculate the Julian and modified Julian date corresponding
    to the provided date string."""
    date,dttime = date_str.split('/')
    day,month,year = date.split('-')
    month = time.strptime(month,'%b').tm_mon
    day = int(day)
    month = int(month)
    year = int(year)
    hour,minute,second = dttime.split(':')
    dec_time = float(hour)+float(minute)/60.0+float(second)/3600.0

```

```

jd = novas.julian_date(year, month, day, dec_time)
mjd = jd-2400000.5
#print jd, mjd, date_str
return jd,mjd

def make_world_gal(g_long,g_lat):
    """Takes two floats assumed to be Galactic longitude
       and latitude, and transforms them into a numpy array
       of world coordinates in decimal degrees."""
    return np.array([g_long,g_lat])

def trans_gal2eq(world_gal):
    """Transforms the input numpy array containing Galactic
       world coordinates to an output numpy array containing
       equatorial world coordinates."""
    trans = wcs.Transformation(wcs.gal,wcs.eq)
    world_eq = trans(world_gal)
    return world_eq

def str2decimal(time_str,isRA):
    hour,minute,second = time_str.split(':')
    hour = float(hour)
    minute = float(minute)
    second = float(second)
    if hour > 0.0:
        coord_dec=hour+minute/60.0+second/3600.0
    else:
        print hour, minute, second
        coord_dec=-(-hour+minute/60.0+second/3600.0)
    if isRA:
        coord_dec=15*coord_dec
    return coord_dec

def decimal2str(theta,isRA):
    """Converts a (rounded value of) a decimal angle given in
       degrees to a string"""
    if isRA:
        theta = theta/15.
    neg = False
    if theta < 0:
        neg = True
        theta = -theta
    deg = np.floor(theta)

```

```

theta = 60.*(theta-deg)
amin = np.floor(theta)
theta = 60.*(theta-amin)
if isRA:
    asec = np.floor(theta*100.)/100.
else:
    asec = np.floor(theta*10.)/10.0
theta = deg+amin/60.+asec/3600.
theta_str = str(int(deg))+':' +str(int(amin))+':' +str(float(asec))
if neg:
    theta_str = "-" +theta_str
if isRA:
    theta = theta*15.
return theta_str,theta

def dec2format(theta):
    """Transforms a decimal angle into a formatted string
    (HH:MM:SS or DD:AM:AS)"""
    sign = +1
    if theta < 0:
        sign = -1
        theta = -theta
    deg,amin = divmod(theta, 1)
    deg = int(deg)
    amin = amin*60.
    amin,asec = divmod(amin,1)
    amin = int(amin)
    asec = asec*60.
    strAngleTruncated = str(deg)+":"+str(amin)
    strAngle = strAngleTruncated":"+str(round(asec,6))
    if sign < 0:
        strAngleTruncated = '-' +strAngleTruncated
        strAngle = '-' +strAngle
    return strAngle,strAngleTruncated,sign,asec

def mas2deg(mas):
    """Convert an angle given in mas to degree"""
    return 0.001*mas*(1./3600.0)

def proj_prlx(x,y,z,RA_0,Dec_0):
    """Computes the RA and Dec projections of the parallax
    ellipse, RA_0 and Dec_0 are in degrees"""
    a_prlx = (x*np.sin(np.radians(RA_0))-y*np.cos(np.radians(RA_0)))\

```

```

/np.cos(np.radians(Dec_0))
d_prlx = x*np.cos(np.radians(RA_0))*np.sin(np.radians(Dec_0))\
+y*np.sin(np.radians(RA_0))*np.sin(np.radians(Dec_0))\
-z*np.cos(np.radians(Dec_0))
return a_prlx,d_prlx

def proper_motion(RA_0,Dec_0,muRA,muDec,jd_i,jd_f):
    """Computes the change of position due to proper motions.
       RA_0 and Dec_0 are in degrees, muRA and muDec are in mas/yr,
       times are specified in Julian Days."""
    d_time = (jd_f-jd_i)/year
    muRA = mas2deg(muRA)
    muDec = mas2deg(muDec)
    Dec = Dec_0+muDec*d_time
    RA = RA_0+muRA*(d_time)/np.cos(np.radians(Dec))
    return RA,Dec

def parallax(RA_c,Dec_c,prlx,jd):
    """Computes the change of position due to parallax."""
    posEarth,velEarth = solsys.solarsystem(jd,3,0)
    x,y,z = posEarth
    RA_prlx,Dec_prlx=proj_prlx(x,y,z,RA_c,Dec_c)
    prlx = mas2deg(prlx)
    return RA_c+prlx*RA_prlx,Dec_c+prlx*Dec_prlx

def calcErrors(Dec_0,doPlot=False):
    """Calculate the expected astrometric errors from the
       theoretical predictions of Pradel et al. (A&A, 452, 1099, 2006)
       for the Declination of the source."""
    ## Input tables 3 and 4 from Pradel+06
    dec = [-25.,0.,25.,50.,75., 85.]
    erra_da = [60.,36.,33.,37.,87.,227.]
    errd_da = [175.,50.,32.,53.,103.,258.]
    erra_dd = [84.,30.,16.,25.,81.,189.]
    errd_dd = [284.,99.,42.,33.,36.,67.]
    erra = [np.sqrt(x**2+y**2) for x,y in zip(erra_da,erra_dd)]
    errd = [np.sqrt(x**2+y**2) for x,y in zip(errd_da,errd_dd)]
    ## We make high order polynomial fits
    #to those data to later interpolate
    coef_ra = np.polyfit(dec, erra, 9)
    poly_ra = np.poly1d(coef_ra)
    coef_dec = np.polyfit(dec, errd,14)
    poly_dec = np.poly1d(coef_dec)

```

```

astro_noise=(poly_ra(Dec_0)/1000,poly_dec(Dec_0)/1000)
## Stuff for the plotting of the fits
ws = np.arange(-30., 90., 0.1)
era = poly_ra(ws)
edec = poly_dec(ws)

if doPlot:
    plt.close()
    plt.plot(dec,erra, 'mo',label='Dec')
    plt.plot(ws, era,'m',label=poly_ra)
    plt.plot(dec,errd, 'bo',label='RA')
    plt.plot(ws, edec,'b',label=poly_dec)
    plt.axis([-30.,90.,0.,400.])
    plt.grid(True)
    #plt.legend(loc=1)
    plt.show()
return astro_noise

def calcTrajectory(jd0,jd1,astro_elements):
    """Calculates the positions of target between jd0 and jd1,
    given the provided astrometric elements."""
    RA_0,Dec_0,muRA,muDec,prlx = astro_elements

    jd_list = []
    RA_list = []
    RA_c_list = []
    Dec_list = []
    Dec_c_list = []

    for jd in range(int(jd0),1+int(jd1)):
        RA_c,Dec_c = proper_motion(RA_0,Dec_0,muRA,muDec,jd0,jd)
        RA,Dec = parallax(RA_c,Dec_c,prlx,jd)
        #print jd,RA_0,Dec_0,RA_c,Dec_c,RA,Dec
        jd_list.append(jd)
        RA_c_list.append(RA_c)
        Dec_c_list.append(Dec_c)
        RA_list.append(RA)
        Dec_list.append(Dec)

    jd_list = np.array(jd_list)
    RA_c_list = np.array(RA_c_list)
    Dec_c_list = np.array(Dec_c_list)
    RA_list = np.array(RA_list)

```

```

Dec_list = np.array(Dec_list)
return jd_list,RA_c_list,Dec_c_list,RA_list,Dec_list

def calcPositions(jd0,jd,astro_elements,random_astro_noise):
    """Calculates the positions of target at each of the JD
       provided in jd (which is a list)"""
    RA_0,Dec_0,muRA,muDec,prlx = astro_elements

    RA_list = []
    RA_c_list = []
    Dec_list = []
    Dec_c_list = []

    for day in jd:
        RA_c,Dec_c = proper_motion(RA_0,Dec_0,muRA,muDec,jd0,day)
        RA,Dec = parallax(RA_c,Dec_c,prlx,day)

        RA_error = random.gauss(0.,mas2deg(random_astro_noise[0]))
        Dec_error = random.gauss(0.,mas2deg(random_astro_noise[1]))

        RA_c_list.append(RA_c+RA_error)
        Dec_c_list.append(Dec_c+Dec_error)
        RA_list.append(RA+RA_error)
        Dec_list.append(Dec+Dec_error)

    RA_c_list = np.array(RA_c_list)
    Dec_c_list = np.array(Dec_c_list)
    RA_list = np.array(RA_list)
    Dec_list = np.array(Dec_list)

    return jd,RA_c_list,Dec_c_list,RA_list,Dec_list

def calcDates(sim_parameters,random_day_disp,add=0.1):
    """Given an initial equinox (equinox1) and a final
       equinox (equinox2), the routine calculates and returns
       the dates of each equinox in between (including
       equinox1 and equinox2."""
    start_date,step,n_epochs,n_per_epoch,n_skipped = sim_parameters
    jd_wo_random = []
    jd_w_random = []
    date_str = start_date+'/12:00:00'
    jd,mjd = calc_jd(date_str)
    jd = jd-step

```

```

for i in range(n_epochs):
    jd = jd+step
    for j in range(n_per_epoch):
        jd_wo_random.append(jd)
#print "jd_wo_random:", jd_wo_random, len(jd_wo_random)
for i in range(n_skipped):
    remove = random.randint(0, (len(jd_wo_random)-1))
    #print remove
    del jd_wo_random[remove]
#print "final jd_wo_random:", jd_wo_random, len(jd_wo_random)
for jd in jd_wo_random:
    random_days = random.uniform(-random_day_disp, random_day_disp)
    jd_w_random.append(jd+random_days)
#print "final jd_w_random:", jd_w_random, len(jd_wo_random)
span = jd_wo_random[len(jd_wo_random)-1]-jd_wo_random[0]
jd0 = jd_wo_random[0]-add*span
jd1 = jd_wo_random[len(jd_wo_random)-1]+add*span
return jd_wo_random, jd_w_random, jd0, jd1

def sim_one_epoch(astro_elements, sim_parameters, noise_parameters):
    """Simulation of observations around each equinox from equinox1
    to equinox2. The equinoxes are specified as tuples containing
    two elements: the type of equinox, and the year of that equinox.
    For example (spring, 2002). The routine computes the dates
    corresponding to each equinox from eq1 to eq2, adds a small
    random dispersion around each equinox (specified by
    random_day_disp) and calculates the trajectory of the source,
    as well as simulated positions at each equinox. Astrometric
    noise (specified by random_astro_noise) is added to the
    simulated positions. RE-WRITE THIS DOC... """
    jd_wo_random, jd_w_random, jd0, jd1 = calcDates(sim_parameters, \
                                                    noise_parameters[0])
    trajectory = calcTrajectory(jd0, jd1, astro_elements)
    positions = calcPositions(jd0, jd_w_random, astro_elements, \
                              noise_parameters[1])
    #print 'jd_wo_random :', jd_wo_random
    #print 'jd_w_random :', jd_w_random
    return trajectory, positions

def matrices_woacc(julianDay, alpha, err_alpha, delta, \
                  err_delta, a_prlx, d_prlx):
    """Defines the A and b matrices for motion without acceleration"""
    mean_jd = np.mean(julianDay)

```



```

d_time = (julianDay-mean_jd)/year
A_alpha = []
b_alpha = []
A_delta = []
b_delta = []

for i in range(len(julianDay)):
    row = (1./err_alpha[i],d_time[i]/err_alpha[i],\
          a_prlx[i]/err_alpha[i])
    row = np.array(row)
    A_alpha.append(row)
    b_alpha.append(alpha[i]/err_alpha[i])
    row = (1./err_delta[i],d_time[i]/err_delta[i],\
          d_prlx[i]/err_delta[i])
    row = np.array(row)
    A_delta.append(row)
    b_delta.append(delta[i]/err_delta[i])
A_alpha = np.array(A_alpha)
b_alpha = np.array(b_alpha)
A_delta = np.array(A_delta)
b_delta = np.array(b_delta)
return A_alpha,b_alpha,A_delta,b_delta

def solveChi2(A,b,isRA,cosdelta):
    """Solves the Chi2 for the given A and b matrices"""
    h = np.dot(A.T,A)
    covA = np.linalg.inv(h)
    dummy = np.dot(covA,A.T)
    p = np.dot(dummy,b)
    err = covA.diagonal()
    err = np.sqrt(err)
    if isRA:
        prlx = (p[2]*3600*1000)
        err_prlx = (err[2]*3600*1000)
        muRA = (p[1]*3600*1000)*cosdelta
        err_muRA = (err[1]*3600*1000)*cosdelta
        RA = p[0]/15.0
        err_RA = err[0]/15.0
        RA,RA_trunc,sign,asec = dec2format(RA)
        err_RA = err_RA*3600
    else:
        prlx = (p[2]*3600*1000)

```

```

    err_prlx = (err[2]*3600*1000)
    muDec = (p[1]*3600*1000)
    err_muDec = (err[1]*3600*1000)
    Dec = p[0]
    err_Dec = err[0]
    Dec,Dec_trunc,sign,asec = dec2format(Dec)
    err_Dec = err_Dec*3600
    return p[0],p[1],prlx,err_prlx

def meanPrlx(RA_prlx,err_RA_prlx,Dec_prlx,err_Dec_prlx):
    """Averages the RA and Dec parallaxes and calculates the errors"""
    RA_w = 1./err_RA_prlx**2
    Dec_w = 1./err_Dec_prlx**2
    mean_prlx = (RA_w*RA_prlx+Dec_w*Dec_prlx)/(RA_w+Dec_w)
    err_mean_prlx = np.sqrt(1./(RA_w+Dec_w))
    return mean_prlx,err_mean_prlx

def fitAstrometry(jd,RA_c,Dec_c,RA,Dec,random_astro_noise):
    """Fit an astrometric model to the data provided. Assumes that
    all data points have the same astrometric error."""

    err_RA = [mas2deg(random_astro_noise[0])]*len(RA)
    err_Dec = [mas2deg(random_astro_noise[1])]*len(Dec)
    a_prlx = []
    d_prlx = []
    for i,day in enumerate(jd):
        posEarth,velEarth = solsys.solarsystem(day,3,0)
        x,y,z = posEarth
        a_prlx_tmp,d_prlx_tmp = proj_prlx(x,y,z,RA_c[i],Dec_c[i])
        a_prlx.append(a_prlx_tmp)
        d_prlx.append(d_prlx_tmp)
    cosdelta = np.cos(np.radians(np.mean(Dec)))

    A_alpha,b_alpha,A_delta,b_delta = matrices_woacc(jd,RA,err_RA,\
                                                    Dec,err_Dec,a_prlx,d_prlx)
    RA_0,mu_RA,RA_prlx,err_RA_prlx = solveChi2(A_alpha,b_alpha,\
                                                True,cosdelta)
    Dec_0,mu_Dec,Dec_prlx,err_Dec_prlx = solveChi2(A_delta,b_delta,\
                                                    False,cosdelta)
    mean_prlx,err_mean_prlx = meanPrlx(RA_prlx,err_RA_prlx,\
                                        Dec_prlx,err_Dec_prlx)
    return mean_prlx

```

```

def fullSimulation(astro_elements,sim_parameters,noise_parameters,\
                  n_simul,plotSim=False):
    """Runs a full simulation"""
    fit_prlx = []
    for i in range(n_simul):
        if i%10 == 0:
            print i,"...",
            ## This runs the simulation (and plots it if you want)
            trajectory,positions = sim_one_epoch(astro_elements,\
                                                sim_parameters,noise_parameters)
            jd_traj,RA_c_traj,Dec_c_traj,RA_traj,Dec_traj = trajectory
            jd_pos,RA_c_pos,Dec_c_pos,RA_pos,Dec_pos = positions
            if plotSim:
                plotSimulation(jd_traj,RA_traj,RA_c_traj,Dec_traj,\
                              Dec_c_traj,jd_pos,RA_pos,Dec_pos,noise_parameters[1])
            ## This makes a fit to the simulated points
            ##(and plots the resulting histograms if you want)
            tmp_prlx = fitAstrometry(jd_pos,RA_c_pos,Dec_c_pos,RA_pos,\
                                    Dec_pos,noise_parameters[1])
            fit_prlx.append(tmp_prlx)
    print 'and done!!\n'
    return fit_prlx

def extractOffsets(RA,RA_c,Dec,Dec_c,RA_p,Dec_p):
    """Returns the offsets from mean position of the provided
    positions. The positions are passed in degrees, but the
    offsets are returned in mas. It will fail (assert False)
    if the declination changes signs during the experiment..."""

    min_Dec = min(min(Dec),min(Dec_c))
    max_Dec = max(max(Dec),max(Dec_c))
    assert np.sign(min_Dec) == np.sign(max_Dec)
    Dec_mean = np.mean(Dec_c)
    cosdelta = np.cos(np.radians(Dec_mean))
    Dec_str,Dec_mean = decimal2str(Dec_mean,False)
    #print Dec_str,Dec_mean
    Dec = (Dec-Dec_mean)*3600*1000.
    Dec_c = (Dec_c-Dec_mean)*3600*1000.
    Dec_p = (Dec_p-Dec_mean)*3600*1000.
    RA_mean = np.mean(RA_c)
    RA_str,RA_mean = decimal2str(RA_mean,True)
    #print RA_str,RA_mean
    RA = (RA-RA_mean)*3600*1000.*cosdelta

```

```

RA_c = (RA_c-RA_mean)*3600*1000.*cosdelta
RA_p = (RA_p-RA_mean)*3600*1000.*cosdelta

return RA,RA_c,Dec,Dec_c,RA_p,Dec_p,RA_str,Dec_str

def calcLimits (pos_list,add=0.1):
    """Calculates the limits for the plots"""
    x_min = np.min(pos_list)
    x_max = np.max(pos_list)
    x_size = (x_max-x_min)
    x_min = x_min-add*x_size
    x_max = x_max+add*x_size
    return x_min, x_max

def plotSimulation(jd_traj,RA_traj,RA_c_traj,Dec_traj,Dec_c_traj,\
                  jd_pos,RA_pos,Dec_pos,random_astro_noise):
    """This plots the results of one simulation.
    Don't run if you are making 1000s of simulations...."""

    RA_off_traj,RA_c_off_traj,Dec_off_traj,Dec_c_off_traj,\
    RA_p_off,Dec_p_off,RA_str,Dec_str = extractOffsets(RA_traj,RA_c_traj,\
    Dec_traj,Dec_c_traj,RA_pos,Dec_pos)
    x_min,x_max = calcLimits(RA_off_traj)
    y_min,y_max = calcLimits(Dec_off_traj)

    plt.close("all")

    fig1 = plt.figure(1,figsize=(16, 12))
    ax1 = fig1.add_subplot(211)
    ax1.plot(jd_traj, Dec_off_traj,label="w/parallax")
    ax1.plot(jd_traj, Dec_c_off_traj,label="wo/parallax")
    ax1.errorbar(jd_pos,Dec_p_off,yerr=random_astro_noise[1],\
                fmt='o',label="simulations")
    ax1.set_xlabel('JD')
    ax1.set_ylabel('Dec (mas from '+Dec_str+')')
    ax1.legend(loc=1)
    ax1.grid(True)
    ax2 = fig1.add_subplot(212)
    ax2.plot(jd_traj, RA_off_traj,label="w/parallax")
    ax2.plot(jd_traj, RA_c_off_traj,label="wo/parallax")
    ax2.errorbar(jd_pos,RA_p_off,yerr=random_astro_noise[0],\
                fmt='o',label="simulations")
    ax2.set_xlabel('JD')

```

```

ax2.set_ylabel('RA (mas from '+RA_str+')')
ax2.legend(loc=2)
ax2.grid(True)

fig2 = plt.figure(2,figsize=(16, 12))
# Plots RA - Dec (i.e. trajectory on the plane of the sky)
ax3 = fig2.add_subplot(111)
ax3.set_aspect('equal')
ax3.plot(RA_off_traj, Dec_off_traj,label="w/parallax")
ax3.plot(RA_c_off_traj, Dec_c_off_traj,label="wo/parallax")
ax3.errorbar(RA_p_off,Dec_p_off,xerr=random_astro_noise[0],\
             yerr=random_astro_noise[1],fmt='o',label="simulations")
ax3.set_xlabel('RA (mas from '+RA_str+')')
ax3.set_ylabel('Dec (mas from '+Dec_str+')')
ax3.axis([x_max,x_min,y_min,y_max])
ax3.legend(loc=2)
ax3.grid(True)
plt.show()
return None

def gauss(x,a,x0,sigma):
    return a*np.exp(-(x-x0)**2/(2*sigma**2))

def prlx_Histogram(fit_prlx,doPlot=False):
    mean_prlx = np.mean(fit_prlx)
    disp_prlx = np.std(fit_prlx)
    init_stats = (mean_prlx,disp_prlx)
    n_bins = len(fit_prlx)/10.
    hist, bins = np.histogram(fit_prlx,n_bins)
    width = 0.7 * (bins[1] - bins[0])
    center = (bins[:-1] + bins[1:]) / 2
    xvals = np.linspace(bins[0],bins[len(bins)-1],200)
    popt,pcov = curve_fit(gauss,center,hist,p0=[1,mean_prlx,disp_prlx])
    if doPlot:
        plt.close("all")
        plt.bar(center, hist, align='center', width=width)
        plt.title("Histogram of fitted parallaxes")
        plt.xlabel("Parallax")
        plt.ylabel("Frequency")
        plt.plot(xvals,gauss(xvals,*popt),'r-')
        plt.show()
    return init_stats,popt

```

```

def completeSimulation(user,astro_elements):
    months,ephFile,year = load_ephem(user)
    random_astro_noise = calcErrors(astro_elements[1])
    start_date = '21-'+months[3]+'-2001'
    n_per_epoch = 1
    n_skipped = 0
    random_day_disp = 10.
    n_simul = 1000
    noise_parameters = (random_day_disp,random_astro_noise)

    full_results = []
    for n_per_year in (1.,1.2,1.3,1.5,2.,2.4,3.,4.,6.,12.):
        partial_results = []
        for n_epochs in (6,12,18,24,30,36):
            step = year/n_per_year
            sim_parameters = (start_date,step,n_epochs,\
                             n_per_epoch,n_skipped)
            fit_prlx = fullSimulation(astro_elements,sim_parameters,\
                                     noise_parameters,n_simul,False)
            init_stats,gauss_fit = prlx_Histogram(fit_prlx,False)
            print "n_per_year = ", n_per_year
            print "n_epochs = ", n_epochs
            partial_results.append(round(abs(gauss_fit[2]),5))

        full_results.append(partial_results)
    return np.array(full_results)

```