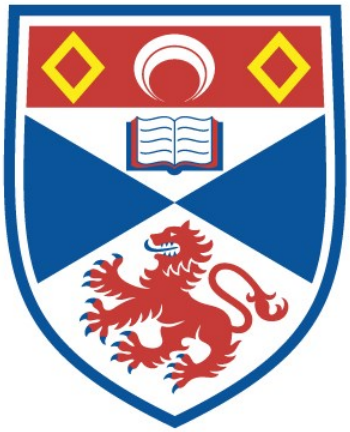
PROBALISTIC MODELLING OF ASTROPHYSICAL TIME SERIES: GRAVITATIONAL MICROLENSING AND OCCULTATION MAPPING OF PLANETS AND MOONS

Fran Bartolić

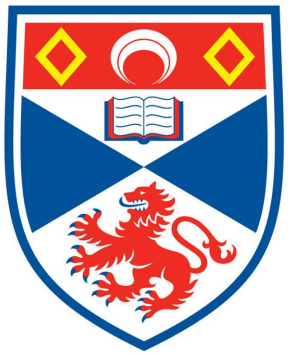
A Thesis Submitted for the Degree of PhD
at the
University of St Andrews



2023

Probabilistic modelling of astrophysical time series: gravitational microlensing and occultation mapping of planets and moons

Fran Bartolić



University of
St Andrews

School of Physics & Astronomy

This thesis is submitted in partial fulfilment for the degree of
Doctor of Philosophy (PhD)
at the University of St Andrews

March 1, 2023

Candidate's declaration

I, Fran Bartolić, do hereby certify that this thesis, submitted for the degree of PhD, which is approximately 35,461 words in length, has been written by me, and that it is the record of work carried out by me, or principally by myself in collaboration with others as acknowledged, and that it has not been submitted in any previous application for any degree. I confirm that any appendices included in my thesis contain only material permitted by the 'Assessment of Postgraduate Research Students' policy.

I was admitted as a research student at the University of St Andrews in September 2017.

I received funding from an organisation or institution and have acknowledged the funder(s) in the full text of my thesis.

Date: 01.03.2023 **Signature of candidate**:

Supervisor's declaration

I hereby certify that the candidate has fulfilled the conditions of the Resolution and Regulations appropriate for the degree of PhD in the University of St Andrews and that the candidate is qualified to submit this thesis in application for that degree. I confirm that any appendices included in the thesis contain only material permitted by the 'Assessment of Postgraduate Research Students' policy.

Date: 01.03.2022 **Signature of supervisor**:

Permission for publication

In submitting this thesis to the University of St Andrews we understand that we are giving permission for it to be made available for use in accordance with the regulations of the University Library for the time being in force, subject to any copyright vested in the work not being affected thereby. We also understand, unless exempt by an award of an embargo as requested below, that the title and the abstract will be published, and that a copy of the work may be made and supplied to any bona fide library or research worker, that this thesis will be electronically accessible for personal or research use and that the library has the right to migrate this thesis into new electronic forms as required to ensure continued access to the thesis.

I, Fran Bartolić, confirm that my thesis does not contain any third-party material that requires copyright clearance.

The following is an agreed request by candidate and supervisor regarding the publication of this thesis:

Printed copy

No embargo on print copy.

Electronic copy

No embargo on electronic copy.

Date: 01.03.2022 **Signature of candidate**:

Date: 01.03.2022 **Signature of supervisor**:

Underpinning Research Data or Digital Outputs

Candidate's declaration

I, Fran Bartolić, hereby certify that no requirements to deposit original research data or digital outputs apply to this thesis and that, where appropriate, secondary data used have been referenced in the full text of my thesis.

Date: 17.11.2022 **Signature of candidate**:

Acknowledgements

I would like to express my gratitude to the following people. My supervisor, Martin Dominik, for giving me the chance and the freedom to work on my own ideas and for his helpful guidance and support throughout my PhD. Rodrigo Luger and Daniel Foreman-Mackey from the Center for Computational Astrophysics (CCA) at the Flatiron institute, for their useful feedback and collaboration and for showing me what true excellence in research on the intersection of astronomy, statistics and computing looks like. I also want to thank all the wonderful people at the St Andrews astronomy department, for providing a stimulating and enjoyable atmosphere, especially my fellow PhD students who went through this journey with me. I appreciate the support and understanding of my family, who always motivated me to follow my goals.

Most of all, I would like to thank Marguerite, for bringing so much joy and happiness to my life and for being my support during the difficult times.

Funding

I acknowledge studentship funding support from the United Kingdom Science and Technology Facilities Council (STFC) and the Scottish Data-Intensive Science Triangle (ScotDIST).

Contents

List of Figures

List of Tables

Abstract

Progress in Astronomy in the 21st century is contingent on the ability to extract useful information from complex and noisy datasets. This requires modeling the data-generating process – a complex combination of the physical phenomenon of interest and the "noise". The goal is to create an approximate model that captures the essence of this process and then fit it to the data. This thesis covers the development of new methods and tools for almost all aspects of the data analysis process in two fields of astronomy: gravitational microlensing and occultation/eclipse mapping. In both fields, the objective is to infer the physical properties of exoplanets, stars, or dark compact objects by measuring the brightness variations of a light source as a function of time. Building on recent advancements in statistics, machine learning, and computer science, I developed a new open-source package called caustics[1] for computing the microlensing magnification in single, binary, and triple-lens microlensing events. I also tackled foundational questions on the statistical analysis of single-lens and multiple-lens microlensing events, developing a new paradigm for modeling degenerate single-lens microlensing events and demonstrating the flaws of commonly used methods for analyzing multiple-lens microlensing events. Moreover, I built models for reconstructing two-dimensional emission maps of spherical bodies, exoplanets, and Solar System objects from one-dimensional photometric occultation light curves. Together with collaborators, I developed a novel method for reconstructing spatial maps of volcanic emission on Jupiter's moon Io from occultation light curves and used the same method for exoplanet eclipse mapping to explore the possibility of detecting weather and climate change on Hot Jupiters using simulated photometric JWST secondary eclipse light curves.

[1] https://github.com/fbartolic/caustics

Reproducibility

All of the code used in this thesis is available at https://github.com/fbartolic/thesis under a very permissive open-source license (MIT). **A special feature of this thesis is that all important figures have a small clickable icon in the form of a GitHub logo in the bottom left corner. Clicking on this icon will take you to the exact code used to generate that figure.** I aimed to make this thesis as reproducible as possible. I initially aimed for full reproducibility, but due to the complexity of the problem and the time constraints, I was not able to achieve this. However, I believe that the code is sufficiently well-documented and the figures are sufficiently well-explained that the reader can follow the steps and reproduce the results.

Chapter 1

Introduction

1.1 Context

The most important consequence of the scientific revolution, which started in the 16th century was operationalising a set of ideas we now call the scientific method. This process relies on very carefully observing the world, building quantitative *models* that describe some aspect of the observed phenomenon, and finally and most importantly, checking if the predictions of the model agree with the observations. Since the time of Galileo, the scientific method has given rise to the modern world and completely revolutionised our understanding of the universe. The result is that today we have some amazingly accurate models (theories) of the universe. Einstein's theory of *General Relativity* describes the universe at a large scale and the *Standard Model* describes the universe at the atomic and subatomic scales.

Two key questions in the physical sciences remain unanswered. The first is: how do we combine the Standard Model and General Relativity to get a complete physical theory of the Universe? And the second: what is the origin and distribution of complex life in the Universe? It used to be the case that conducting experiments that have the potential to answer questions of such importance could be accomplished by a single person or a small group of researchers. An example would be, for instance, Kepler's observations of planetary orbits, Ernest Rutherford's experiments with atoms or Arthur Eddington's observations of the Solar eclipse to test General Relativity. The scientific method mostly consisted of of writing down observations in a physical notebook, and the analysis hardly required complex statistics.

Since then, the rate of progress has slowed down and much of the "low-hanging fruit" of scientific discovery (in the physical sciences at least) has been exhausted. Today, making substantial progress often requires coordination between many scientists, physical engineers and software engineers who are working with complex instruments. Most importantly for this thesis, writing code and doing numerical simulations is ubiquitous and often necessary. Consider this list of some of the most notable scientific discoveries in the past few decades:

- Discovery of accelerated expansion of the Universe.
- First sequencing of the human genome.
- Detection of the Higgs Boson.

- Paleogenomics studies of the origin of Homo Sapiens.
- Detection of gravitational waves by LIGO.
- Reconstruction of the first image of a Black Hole.

All of these discoveries required collecting substantial amounts of data and the use of relatively complex statistical analysis techniques. Computation and statistical analysis are now absolutely central to the process of scientific discovery.

Additionally, since the early 2010s, there has been a revolution (still ongoing and showing no signs of slowing down) in machine learning/AI centred around deep learning and neural networks. Besides the incredible progress in predicting patterns in language and vision, deep learning has also been used in science to solve the protein folding problem (Jumper et al., 2021) and in mathematics to discover novel conjectures and theorems (Davies et al., 2021). Deep learning has been less useful in physics and astronomy so far. However, as I will argue in this thesis, some of the technologies which underlie deep learning such as automatic differentiation and GPU computing can be very useful for processing and understanding complex datasets in physics and astronomy.

Another revolution of sorts is happening in the physical sciences, especially in astronomy. In the past two decades, there has been a substantial increase in the popularity of Bayesian statistics, as opposed to frequentist statistics. Most of this thesis is devoted to the application of Bayesian methods to specific astrophysical problems. Although these methods have been around for a very long time, they only started to become computationally feasible relatively recently.

Having situated the work presented in this thesis in the present moment and pointed out relevant scientific and technological developments in recent decades, I now focus on astronomy in particular. The kinds of questions in astronomy that most excite me are those which lead us closer to answering one of those two fundamental questions I stated at the beginning of this chapter. In particular, the question about the origin of life in the universe. Since the first discoveries of planets outside of our Solar System in the early 1990s (Wolszczan and Frail, 1992; Mayor and Queloz, 1995), thousands more have been confirmed[1] using methods such as transits, radial velocity, microlensing and direct imaging. Thanks to gravitational microlensing, we now know (Cassan et al., 2012) that there is, on average, at least one planet per star in the Milky Way. In addition to detecting the presence of the planets and inferring their properties such as mass, radius, and orbital period, we can now also measure the transmission and emission spectra of their atmospheres and even reconstruct very crude maps of their surfaces (the subject of Chapter 7) (Knutson et al., 2007; Majeau et al., 2012). With the James Webb Space Telescope, we might even be able to detect biosignatures in the atmospheres of Earth-size planets.

Answering a great question such as "Are there biologically produced complex molecules in this exoplanet atmosphere" will be a very difficult task. It will also almost certainly not be a clear yes or no kind of answer. Rather, it will require a deep understanding (i.e. good *models*) of the physics of exoplanet atmospheres, stellar variability, the response of the instrument, sophisticated statistical analysis of the data drawing on multiple independent pieces

[1] At the time of writing NASA Exoplanet Archive contains more than 5000 exoplanets.

of evidence, and clear definitions of what it means to have detected something. Building a good model for the thing we really care about requires understanding also things we may intrinsically care less about (instrumental systematics, details of stellar variability, variations in Earth's atmosphere etc.). The focus of this thesis building the methods and tools that will enables us to get closer to answering the fundamental questions. Wherever possible, I approach problems using first principles thinking but with a heavily computational/statistical approach and a healthy dose of pragmatism.

In the next two sections, I briefly introduce specific areas of astronomy I worked on and I provide an outline of the rest of the thesis.

1.2 Gravitational microlensing

"Do not bodies act upon light at a distance and by their action bend its rays, and is not this action strongest at the least distance?" asked Newton (1704). Much later, in 1911, even before he published his theory of General Relativity (GR), Einstein considered the deflection of light from a distant star passing close to a massive object in the foreground. He derived an expression for the deflection angle of the light ray which was off by a factor of 2 relative to the correct value. In 1916, the same year Albert Einstein published his General theory of Relativity (in which he corrected his earlier mistake), the English physicist Sir Oliver Lodge suggested the light-bending phenomenon could produce a *gravitational lens*.

The first experimental confirmation of the deflection of light by a mass came in 1919 during the Solar eclipse when the astronomer Arthur Eddington famously measured the deflection of light from distant stars passing close to the limb of the Sun[2]. Much later, in 1936, a young amateur scientist R.W. Mandl convinced Einstein to write a short paper on the lensing effect by a massive star acting as the lens instead of the Sun. In the paper (Einstein, 1936), Einstein derived an expression for the magnification of the distant star when it is closely aligned to the foreground lens star relative to an observer and he predicted that a luminous ring would form if the two stars were perfectly aligned. He considered this effect curious but useless, stating that *"Of course, there is no hope of observing this phenomenon directly. First, we shall scarcely ever approach closely enough to such a central line. Second, [the angles] will defy the resolving power of our instruments"*.

In this matter, Einstein was wrong. The famous astronomer Fritz Zwicky noticed the phenomenon is likely to be observable at galactic scales if the lens was a massive galaxy (Zwicky, 1937a,b) and that one could use the measurement of the deflection angle to weigh the lens. Zwicky was right, and the first definitive observation came in 1979 by Young et al. (1980), who observed a double image of the quasar Q0957+561 and concluded that the two images originate from the same object whose light had been distorted by a massive galaxy. Many other observations followed, including visible Einstein rings. Far from being just a curious phenomenon, galactic scale lensing is now one of the key methods of observational cosmology used for inferring the parameters of the standard model and the distribution of dark matter in the universe.

[2]He concluded that the observed deflection was in agreement with GR. However, it is doubtful that the data were actually good enough to distinguish between the GR prediction for the deflection of light and the Newtonian prediction, which is derived using the equivalence principle and is equal to one-half the GR value.

Figure 1.1: A stellar field of a microlensing event GLE-2012-BLG-0406 (centred), imaged by one of Las Cumbres Observatory's 2m telescopes, showing the high density of stars typical in microlensing observations. Credit: Y. Tsapras. Taken from http://microlensing-source.org/pictures/.

That the same effect could be used to detect the presence of planets orbiting around the lens star was first theorised by Liebes (1964), who wrote that *"the primary effect of planetary deflectors bound to stars other than the Sun will be to slightly perturb the lens action of these stars"*. However, he was also sceptical about the possibility of detection, saying that *"associated pulses would be so weak and infrequent and of such fleeting duration – perhaps a few hours – as to defy detection"*. Gravitational lensing as a method for discovering exoplanets really took off with the work of Paczyński (Paczynski, 1986b,a; Mao and Paczynski, 1991), who also coined the term *microlensing*, referring to gravitational lensing in a regime where the images of the background light source cannot be resolved but one can nevertheless measure its magnification as a function of time. Microlensing as a method for detecting exoplanets has some unique aspects. First, it is a one-off event that happens on a timescale of a few minutes up to several months depending on the distances to the background star, the lensing star, and the mass of the lens. In addition to the fact that there is only one chance to observe such an event, for it to happen at all we need extremely precise alignment between the lens and the background star. The chance of observing a stellar microlensing event is roughly one-in-a-million for a typical star within the Milky Way. Observing a planetary signal is about an order of magnitude less likely than that. Hence, obtaining a decent sample of planetary events requires continuous monitoring on the order of 10^8 stars. Thus, microlensing observations focus on the densest region of the Milky Way – the Galactic bulge. Figure 1.1 shows a picture of such a dense stellar field. Finally, in the vast majority of cases, we detect no light from the lens itself. The collected photons originate from a background star completely unrelated to and distant from the lensing star. This is very different from other exoplanet discovery methods and means that we can only obtain the dynamical properties of planets such as their masses and periods.

The properties of microlensing events that make them difficult to observe also mean that they provide a unique lens on exoplanet systems. Relative to other methods such as transits and radial velocity, microlensing is sensitive to planets located at substantially greater distances, well outside our Solar System neighbourhood and even potentially to Milky Way's satellite galaxies such as the Magellanic Clouds and the nearby Andromeda galaxy. Microlensing is also sensitive to very small planets and planets that are further out from the star than those typically detected using transits and radial velocity. Microlensing surveys such as OGLE (Udalski et al., 1993) and MOA (Muraki et al., 1999) continuously monitored crowded stellar fields in the Milky Way since the 1990s[3] discovering thousands of stellar events and dozens of planetary events. To detect planetary deviations in the observed light curve, it is essential to have high-cadence observations of the source star. The way survey campaigns traditionally work is that once a particular star starts to become magnified by a large factor, many additional small telescopes join in to improve coverage of the key parts of the light curve. Sometimes space-based observatories get involved as well. The vast majority of microlensing events analysed so far consist of observations from multiple observatories, each with its unique aspects, such as noise properties, cadence and photometric quality.

Future surveys such as the ground-based Rubin Observatory (Ivezić et al., 2019) telescope and the space-based Roman Telescope (Penny et al., 2019) and Euclid (Bachelet et al., 2022) will detect tens of thousands of events in total. Although most of the microlensing literature is focused on characterising multiple-lens events, answering questions about *populations* of objects with these new, but also with existing datasets, requires scalable data analysis methods and a clear set of guidelines on how to interpret the analysis products. This is a substantial challenge because microlensing events are notoriously difficult to model. Even though the datasets are relatively simple (multiple time series photometric light curves in different bands), the parameter space for even the simplest models is highly non-linear, correlated, relatively high dimensional and there are often near-perfect degeneracies in the solutions. The assumptions relied on by existing methods for modelling microlensing events are often opaque and unquestioned. Discussions on model "degeneracies" (Song et al., 2014; Hwang et al., 2019; Skowron et al., 2018; Dominik, 2009), correlated noise (Bachelet et al., 2015; Li et al., 2019), and model comparison (Hwang et al., 2018; Dominik et al., 2019), have been ongoing in the microlensing literature for decades without a clear solution in sight and without a proper framing of the issue.

In this thesis, I revisit these sorts of questions while taking into account many recent developments in the fields of computational statistics and machine learning. The microlensing portion of this thesis is structured as follows. In Chapter 3 I introduce the caustics code for computing the extended source magnification of single, binary, and triple-lens systems. This chapter deals with the *forward modelling* problem of microlensing, i.e. computing the magnification of a source star as a function of time given a set of system parameters. The *inverse problem* – how to infer the system parameters from the observed light curve – is discussed in Chapter 4 (for single lens events), and in Chapter 5 (for multiple-lens events).

[3]Initially the focus was on finding dark matter candidate particles – so-called MACHOs (Massive Compact Halo Objects).

1.3 Occultation and phase curve mapping

Interestingly, the history of the second topic of this thesis – occultation and phase curve mapping – is not unlike that of microlensing. The idea of using occultations and phase curves to reconstruct a two-dimensional "map" of spherical astronomical bodies was proposed in the early 20th century. It was only much later that technology caught up with the idea. In 1906, Russell (1906) pointed out that certain features in light curves of Solar System satellites can be attributed to inhomogeneities of their surfaces. The key idea is as follows. Although at any given time we observe only the total light from an unresolved satellite or planet, different portions of the surface are visible at different times. Thus, we may expect that some of the information about the light emitted or reflected from the surface will be imprinted onto the light curve. Russell (1906) also considered the inverse problem – can we learn something about the surface of these objects starting from a light curve? The method he proposed is now known as *phase curve mapping*. It was first attempted by Lacis and Fix (1972), who analysed photometric light curves of Pluto in reflected light, attempting to constrain variations in the *albedo* of the surface with inconclusive results.

Later works such as Dunbar and Tedesco (1986); Buie et al. (1992); Young and Binzel (1993); Young et al. (1999) went a step further by using both phase curves and also light curves of mutual *occultations* of Pluto by its moon Charon to reconstruct albedo maps of Pluto. The significant advantage of occultations relative to just phase curves is that occultation light curves encode more information about the surface because the sharp limb of the occultor sweeps over the disc of the occulted body and exposes its different parts. Besides Pluto, occultations of another Solar System body – the Jovian moon Io – were also analysed in the 1990s. Starting with the work of Spencer et al. (1994), who observed occultations of Io by Europa and Jupiter, Io has been a regular target for near-infrared observations using ground-based telescopes such as NASA's Infrared Telescope Facility (IRTF). The goal of these observations is to understand the volcanic activity on Io's surface which is covered with many time-varying and bright volcanic features. The observing campaign of Io has yielded insights into the nature of its volcanic activity and it continues to this day.

A natural question arises, can we do this with objects outside of the Solar System? The answer is yes. Knutson et al. (2007), Majeau et al. (2012), and de Wit et al. (2012b) used Spitzer mid-infrared observations of secondary eclipses of the Hot Jupiter HD189733b and found that surface emission is best described by the presence of a large hot spot on the dayside of the planet longitudinally offset from the substellar point. Similarly, Stevenson et al. (2014) produced temperature maps of the Hot Jupiter WASP-43b, Demory et al. (2013) mapped the Hot Jupiter Kepler-7b in reflected light and Demory et al. (2016a) mapped the thermal emission from the Super Earth 55 Cancri e. These studies were able to only capture longitudinal variations in intensity. Real exoplanet atmospheres are certain to have three-dimensional spatial inhomogeneities in emission more complex than a single hot spot due to the presence of clouds, zonal jets, storms, waves, etc. (Showman et al., 2020).

In recent years there have been significant advances in the statistical modelling of phase curves and eclipse light curves. Most notably, Luger et al. (2019) introduced the `starry` code which enables analytic computation of phase curves and occultation light curves for bodies with arbitrary emission maps expressed in a spherical harmonic basis (an idea dating back to Russell (1906)). Luger et al. (2022a) expanded the algorithm for the (considerably more

complicated) case of reflected light. In this thesis, I will present the work I have done in collaboration with other researchers from planetary science and exoplanet communities. I have used starry to map the surface of Io and investigate the prospects for detecting fine spatial structure in the atmospheres of Hot Jupiters using simulated JWST data. In Chapter 6 I describe a novel model for inferring emission maps of Io from occultation light curves. This was previously published in Bartolić et al. (2022). In Chapter 7 I draw on the results of this work and investigate how feasible is it to use the same techniques in the context of mapping Hot Jupiter atmospheres with the goal of detecting weather and climate patterns. The application of the occultation/eclipse mapping method to Io can be seen as the best-case scenario for the application of the same method to exoplanets.

Chapter 2

The theoretical minimum

In this chapter, I review the basic concepts in microlensing, occultation mapping, and computational statistics. These concepts are necessary for understanding the subsequent chapters.

2.1 Gravitational microlensing

Gravitational lensing is generally divided into multiple classes depending on whether the effect is discernible at the level of individual objects or at a level of a statistical sample of objects. The primary division is between *weak lensing* and *strong lensing*. The former results in a subtle distortion of the images of a background source that can only be teased out in a statistical sense. The latter refers to the lensing of individual objects where the lensing effect is less subtle. It is further divided into *macrolensing* – lensing of galaxies where the multiple images of the source are resolved, and *microlensing* – where the images are generally not resolved and the observable is the total magnification of the source as a function of time (a light curve). In the case of microlensing, the light source is either a quasar or a star (sometimes a binary star) and the lenses are stars, brown dwarfs, planets, or compact objects such as black holes, neutron stars and white dwarfs. In this thesis, I focus specifically on stellar microlensing instead of quasar microlensing[1].

2.1.1 Deflection of light by gravity

Gravitational lensing is a phenomenon fully described by Einstein's General Theory of Relativity (GR) which predicts that light changes direction when passing close to a massive body. According to GR:

- The presence of mass changes the spacetime geometry from the flat (Minkowski) metric $\eta_{\mu\nu}$ to a curved metric, specified by the tensor $g_{\mu\nu}$.
- Massless particles such as photons follow the null geodesics, paths we can obtain by solving the *geodesic equation*.

[1] Quasar microlensing is a somewhat separate community from the rest of the microlensing community.

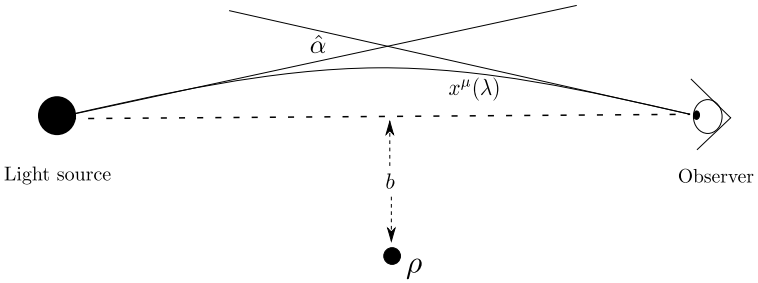


Figure 2.1: Geometry of gravitational lensing. A geodesic curve $x^\mu(\lambda)$ is deflected by an angle $\hat{\alpha}$ from its initial trajectory due to the presence of a massive body with mass density ρ. Figure adapted from Carroll (2019).

If we restrict ourselves to the regime where the metric is time-independent and the test particles are allowed to travel at any velocity less than c, also known as the *weak field* or *linearised approximation* of GR, the metric tensor is

$$g_{\mu\nu} = \eta_{\mu\nu} + h_{\mu\nu} \quad , \tag{2.1}$$

where (assuming that the metric sign convention is $(-,+,+,+)$ and that we are working in Cartesian coordinates (t,x,y,z)) $h_{\mu\nu} = \text{diag}(-2\Phi, -2\Phi, -2\Phi, -2\Phi)$ (Carroll, 2019). The (static) Newtonian gravitational potential Φ obeys the Poisson equation

$$\nabla^2\Phi = 4\pi G\rho \quad , \tag{2.2}$$

where ρ is the mass density of the distribution of mass located between the observer and the light source.

The geometry of the lensing system consists of a distant source of light, the observer, and distribution of mass with density ρ located between the source and the observer are shown in Figure 2.1. A photon is deflected, as it travels from a source to an observer by a *deflection angle* $\hat{\alpha}$, a vector in the plane perpendicular to the wave vector $\mathbf{k}$ pointing in the direction of photon propagation. The deflection angle is (Carroll, 2019):

$$\hat{\alpha} = 2\int \nabla_\perp\Phi \, \mathrm{d}s \quad , \tag{2.3}$$

where $\nabla_\perp\Phi$ is the gradient of the potential in the direction transverse to the path of the photon and s is the spatial distance travelled. For a point mass M, the potential is

$$\Phi = -\frac{GM}{(b^2+x^2)^{1/2}} \quad , \tag{2.4}$$

where x parametrises the straight line connecting the observer and the lens and b is the impact parameter of the light ray. Integrating from $-\infty$ to ∞ (i.e. assuming that both the source and the observer are located far away from the deflecting mass and that the deflection angle is small), we obtain the deflection angle:

$$\hat{\alpha} = \frac{4GM}{c^2 b} = \frac{2R_s}{b} \quad , \tag{2.5}$$

where $R_s = 2GM/c^2$ is the Schwarzschild radius. $\hat{\alpha}$ is directly proportional to the mass of the lens and it is independent of the wavelength of the light. It also increases with the proximity

of the light ray to the lensing mass which is the opposite behaviour to that of a classical convex lens for which the deflection angle vanishes at the centre of the lens. The resulting deflection angle is very small, for example, for the Sun we have $GM/c^2 = 1.48 \times 10^5$cm (2.95 km), $R = 6.96 \times 10^{10}$cm and $\hat{\alpha} = 1.75$ arc seconds. This is the angle that was claimed to have been observed by Eddington during the 1919 total solar eclipse.

Equation 2.3 is valid when the linearised approximation of GR is sufficient. In this approximation, the deflection angles are additive, and the total deflection caused by a group of massive bodies is just the sum of the individual deflection angles. The linearised approximation breaks down when the impact parameter b approaches the Schwarzschild radius of the lensing mass because the linearised metric is no longer sufficient. Thus, except for modelling lensing in the close vicinity of a black, the linearised approximation is adequate.

2.1.2 The magnification of a point source by a point lens

Consider a system consisting of an observer O, a point mass M, and a point light source S. The geometry of this system is shown in Figure 2.2. The lens is located in the *lens plane*, perpendicular to the observer-lens axis, at a distance D_L away from the observer. The light source is located in the *source plane* perpendicular to the observer-source axis, at a distance D_S away from the observer. The observer is in the *observer plane*. The assumption that the location of the source and the lens can be parametrised using angular coordinates as seen by the observer (or Cartesian coordinates if the distance to the lens and the source is known) in their respective planes is justifiable because the deflection angles involved are tiny and the distance between the observer and any point of interest in the lens plane is approximately constant. Let θ be the apparent angular position of the source in the sky with respect to the observer–lens axis and β be the actual angular position of the source. Assuming a metric for spacetime in between the source and the observer that is approximately Euclidian[2] the distance from the lens to the source is $D_S - D_L$.

From Figure 2.2 and using Equation 2.3 it follows that the relationship between the observed and the actual location of the source is

$$\beta = \theta - 2R_s \frac{D_S - D_L}{D_L D_S} \frac{1}{\theta} \quad , \tag{2.6}$$

which is known as the *lens equation* or sometimes also the *ray-tracing* equation. If the source and the lens are perfectly aligned with respect to the observer, then $\beta = 0$ and $\theta \equiv \theta_E$ where

$$\theta_E = \sqrt{\frac{4GM}{c^2}\left(\frac{1}{D_L} - \frac{1}{D_S}\right)} = \sqrt{\kappa M \pi_{LS}} \quad , \tag{2.7}$$

where $\kappa \equiv \frac{4G}{c^2 \mathrm{au}} \simeq 8.1 \frac{\mathrm{mas}}{M_\odot}$ and

$$\pi_{LS} \equiv \pi_L - \pi_S = \frac{1\mathrm{au}}{D_L} - \frac{1\mathrm{au}}{D_S} \tag{2.8}$$

[2]Which is a valid assumption for galactic sources but not for extragalactic sources such as quasars and galaxies require a cosmological model for the metric.

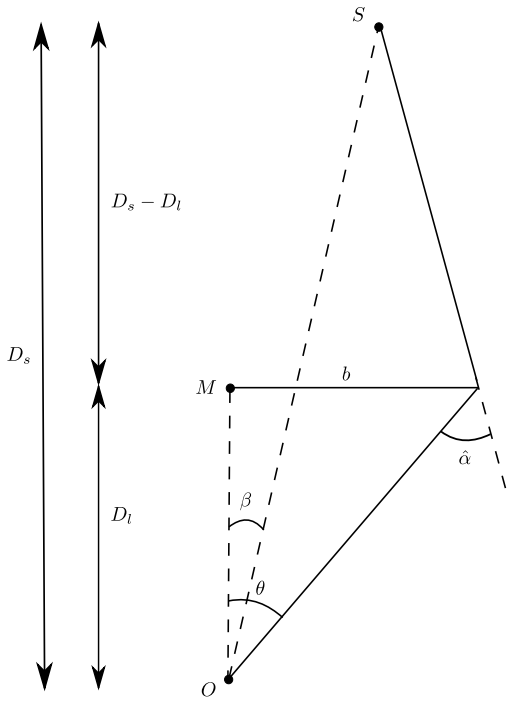


Figure 2.2: The geometry of a system consisting of a point mass lens M located at a distance D_L at angular separation β from the observer, and a single light source S located at a distance D_S, emitting a light ray which gets deflected by an angle $\hat{\alpha}$ from its original trajectory. As a result, the observer sees the source at an angular separation θ. Figure adapted from Schneider et al. (1992).

is the relative lens-source parallax.

With perfect alignment ($\beta = 0$), the image of the source forms a ring (the Einstein ring) around the lens star with angular radius θ_E – the *Einstein radius*. θ_E depends on the mass of the lens and the distances to the lens and the source. The angular Einstein radius sets the characteristic scale of a microlensing event; it is a natural unit for most microlensing parameters. For an M-dwarf star lens in the galactic disc ($D_L \sim 3\,\mathrm{kpc}$) and a source star in the galactic bulge ($D_S \sim 8\,\mathrm{kpc}$) we have $\theta_E \sim 1\,\mathrm{mas}$ so the typical scale of microlensing is on the order of milliarcseconds. The images of the source star are only rarely observable using the most advanced optical interferometers, such as the GRAVITY instrument on ESO's Very Large Telescope, whose lower resolution limit is 2 mas (Gravity Collaboration et al., 2017). Dong et al. (2019) were the first to resolve the images of a microlensing event, finding the value of $\theta_E = 1.87\,\mathrm{mas}$ using the GRAVITY instrument.

We can rewrite Equation 2.6 by defining new dimensionless angular coordinates scaled by the angular Einstein radius:

$$z \equiv \frac{\theta}{D_L \theta_E}, \quad w \equiv \frac{\beta}{D_S \theta_E} \quad . \tag{2.9}$$

The lens equation then takes the very simple form

$$z = w - \frac{1}{w} \quad . \tag{2.10}$$

Equation 2.10 is a non-linear mapping between the source plane w and the lens plane z. If

we solve for z, we obtain two solutions for the position of the images, given by

$$z_{\pm} = \frac{1}{2}\left(w \pm \sqrt{w^2+4}\right) \quad . \tag{2.11}$$

For non zero w, the so-called *minor image* is always located within the Einstein radius ($|z_-| < 1$) while the *major image* is located outside of it ($|z_+| > 1$).

Because gravitational lensing involve no additional emission or absorption along a deflected ray of light and the there is no net wavelength shift between the emission point and the observer, the *surface brightness* (flux density per unit angular area), I, of the source image is identical to the surface brightness of the unlensed source. The flux of an infinitesimal source is a product of the surface brightness and the solid angle $\Delta\omega$ subtended by the source in the sky. This flux changes throughout the microlensing event because the source and the lens are in motion relative to an observer. The ratio of the original and the lensed source flux, the *magnification* A, is then:

$$A = \frac{\Delta\omega}{(\Delta\omega)_0} \quad , \tag{2.12}$$

where 0 denotes the unlensed solid angle. Assuming that the source has infinitesimal size and is at an angular location $\mathbf{w} \equiv (w_1, w_2)$ subtending an angle $\Delta(\omega)_0$, and an image of the source at location $\mathbf{z} \equiv (z_1, z_2)$ subtending an angle $\Delta\omega$; the ratio between the solid angle of the source and the image is given by the determinant of the Jacobian matrix of the lens mapping $\mathbf{z} \rightarrow \mathbf{w}$, evaluated at the location of the images

$$\frac{(\Delta\omega)_0}{\Delta\omega} = \left|\det\frac{\partial\mathbf{w}}{\partial\mathbf{z}}\right| \quad . \tag{2.13}$$

The magnification A is then given by the inverse Jacobian determinant of the lens mapping:

$$A = \left|\det\frac{\partial\mathbf{w}}{\partial\mathbf{z}}\right|^{-1} \quad . \tag{2.14}$$

The images for which the determinant of the Jacobian of the lens mapping is positive are said to have positive *parity* and vice versa.

Looking at Equation 2.14, we see that the magnification diverges when the Jacobian determinant of the lens mapping vanishes. Curves in the lens plane for which the determinant of the lens mapping vanishes, that is,

$$\left|\det\frac{\partial\mathbf{w}}{\partial\mathbf{z}}\right| = 0 \quad , \tag{2.15}$$

are called *critical curves*. The critical curves in the lens plane can be mapped to the source plane using the lens equation $\mathbf{z} \rightarrow \mathbf{w}$, and these curves are called *caustic curves*. Although the magnification factor in Equation 2.14 formally diverges at the points of critical curves or caustics, the divergence is not physical because light sources are not in reality point-like. If the finite angular size of the source is taken into account, the magnification is an integral of Equation 2.14 over the the extent of the source, weighted by the source brightness and that integral is always finite. The critical and caustic curves are closed curves and one show

that *the number of images changes by two if and only if the source crosses a caustic curve* (Schneider et al., 1992, Chapter 6).

We can derive a simpler expression for Equation 2.12 by switching to polar coordinates (v, ϕ) in the lens plane. The lens equation for the two vector components of the apparent source position is

$$w_1 = z_1 - \frac{z_1}{z_1^2 + z_2^2} \tag{2.16}$$

$$w_2 = z_2 - \frac{z_2}{z_1^2 + z_2^2} \quad . \tag{2.17}$$

Introducing polar coordinates $z_1 = v\cos\phi,\ z_2 = v\sin\phi$, we have

$$\left|\det\frac{\partial \mathbf{w}}{\partial \mathbf{z}}\right| = 1 - \frac{1}{v^4} \quad . \tag{2.18}$$

By substituting the locations of the source images Equation 2.11 into the above expression, we obtain the magnification of the source images

$$A_\pm = \frac{1}{2}\left(\frac{u^2+2}{u\sqrt{u^2+4}} \pm 1\right) \quad , \tag{2.19}$$

where $u = \sqrt{w_1^2 + w_2^2}$ is the magnitude of the position vector of the source star. The total magnification is then the sum of the two magnifications and is given by (Einstein, 1936)

$$A(u) = |A_-| + |A_+| = \frac{u^2+2}{u\sqrt{u^2+4}} \quad , \tag{2.20}$$

For a single point lens, the critical curve is simply the Einstein ring corresponding to $v = \sqrt{z_1^2 + z_2^2} = 1$, and the caustic curve is mapped to a single point in the source plane at $u = 0$. For source separations much smaller than the angular Einstein radius ($u \ll 1$), the magnification is approximately $A(u) \simeq 1/u$. In the opposite case ($u \gg 1$) we have $A(u) \simeq 1 + 2/u^4$, that is, the magnification falls off rapidly the further away the source is from the lens.

To evaluate Equation 2.20 in practice, we have to parametrise the position of the source on the sky u as a function of time. To derive an expression for $u(t)$, we assume that the motion of the observer, the lens, and the source is rectilinear (acceleration is neglected). We take $\mathbf{w}_S$ to be the angular position of the source star on the plane of the sky and $\boldsymbol{\mu}_L$ to be its proper motion vector. Likewise, for the lens. We thus have:

$$\mathbf{w}_S(t) = \mathbf{w}_{S,0} + (t - t_0)\,\boldsymbol{\mu}_S \tag{2.21}$$

$$\mathbf{w}_L(t) = \mathbf{w}_{L,0} + (t - t_0)\,\boldsymbol{\mu}_L \quad , \tag{2.22}$$

where t_0 is some reference time. The relative position vector of the lens with respect to the source is then

$$\boldsymbol{u}(t) \equiv \frac{\mathbf{w}_L(t) - \mathbf{w}_S(t)}{\theta_E} = \frac{\mathbf{w}_{LS,0}}{\theta_E} + \frac{t - t_0}{\theta_E}\boldsymbol{\mu}_{LS} \quad , \tag{2.23}$$

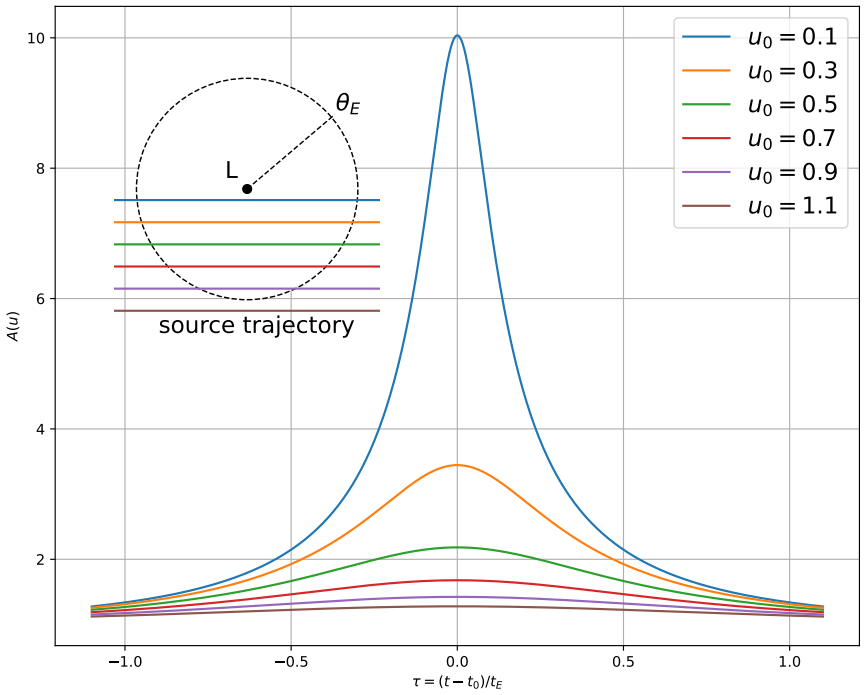


Figure 2.3: Magnification of a point source by a point lens as a function of time. The various magnification curves correspond to different impact parameters u_0. The inset figure shows the source trajectories in the lens plane, the dashed circle corresponds to the Einstein radius $v = 1$.

where $\mathbf{w}_{LS,0} \equiv \mathbf{w}_{L,0} - \mathbf{w}_{S,0}$ is the relative position at t_0, and $\boldsymbol{\mu}_{LS} \equiv \boldsymbol{\mu}_L - \boldsymbol{\mu}_S$ is the relative proper motion. Since $\boldsymbol{\mu}_{LS}$ and $\mathbf{w}_{LS,0}$ are perpendicular to each other, it follows that the magnitude of the relative separation is

$$u(t) = \sqrt{u_0^2 + \left(\frac{t - t_0}{t_E}\right)^2} \quad , \tag{2.24}$$

where we have defined $u_0 \equiv |\mathbf{w}_{LS,0}|/\theta_E$ and $t_E \equiv \theta_E/|\boldsymbol{\mu}_{LS}|$. The magnification is a function of three parameters (t_0, u_0, t_E), and the resulting curve as a function of time is often called the Paczyński curve (Paczynski, 1986b,a). The magnification as a function of time for different impact parameters u_0 is shown in Figure 2.3. Notice the very steep fall-off in magnification as we move away from the Einstein ring ($A(u) \propto 1 + 2u^{-4}$). In the limits of $u_0 \to 0$ and $u_0 \gg 1$ there is a continuous mathematical degeneracy between the parameters t_0, u_0 and t_E (Woźniak and Paczyński, 1997).

2.1.3 Observed flux

The magnification $A(t)$ is not a direct observable in microlensing events. We can measure the flux of the source star, which is usually contaminated by the flux of other nearby stars and potentially also with light from the lens and possible companions to the lens. The observed flux is then

$$F(t) = F_S\, A(t) + F_B \tag{2.25}$$

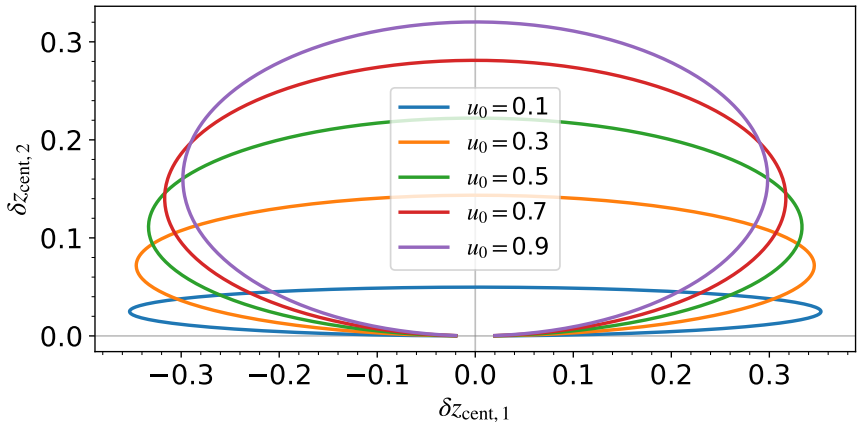


Figure 2.4: The astrometric shift in the centroid of light for a point source point lens model. The various curves correspond to different impact parameters u_0.

where F_S is the flux of the source star and F_B is the contamination flux, also called the *blending flux*. To quantify the strength of the blending, we can also define the *source flux fraction* b_S as the fraction of total (unmagnified) flux that is coming from the source star:

$$b_S \equiv F_S/(F_S + F_B) \quad . \tag{2.26}$$

For highly blended events $b_S \approx 0$ and in the absence of blending $b_S = 1$. The source flux F_S and the blending flux F_B are generally highly correlated, and it is sometimes preferable to use a slightly different parametrisation proposed by Dominik (2009). In this parametrisation, we use the *baseline flux* F_{base} and the difference between the baseline flux and the flux at peak magnification $\Delta F \equiv F_S[A(u_0) - 1]$ as the new parameters. Equation 2.25 then takes the form

$$f = \Delta F \frac{A(t) - 1}{A(t_0) - 1} + F_{\mathrm{base}} \quad . \tag{2.27}$$

It is generally straightforward to measure the flux difference ΔF, the baseline flux F_{base} and the centre of the curve t_0 but getting a good estimate of t_E even outside of the regime $u_0 \to 0$ and $u_0 \gg 1$ requires good photometric sampling along the "wings" of the light curve (Dominik, 2009). Although the parametrisation defined in Equation 2.27 best maps to the observable features in the light curve, I prefer to use the parametrisation which uses the parameters (F_S, F_{base}):

$$f = F_S\,[A(t) - 1] + F_{\mathrm{base}} \quad . \tag{2.28}$$

The advantage of this parametrisation is that it still uses the baseline flux $F_{\mathrm{base}} = F_S + F_B$ which is a direct observable, but it does not require an extra evaluation of the magnification $A(t_0)$. In addition, F_S is a parameter for which it easier to set a sensible prior. As we shall see in Section 2.3.6, the choice of parametrisation in this case is not so important because in end the we will always marginalise (integrate out) the linear parameters.

2.1.4 Astrometric microlensing

In addition to the photometric microlensing effect, there is also the astrometric microlensing effect. Astrometric microlensing refers to the relative shift in the *angular position* of the

unmagnified source star when it is in the vicinity of the lens. For a point source and a single lens, we can define the light centroid of the two images by weighting their positions (Equation 2.11) with their magnifications (Equation 2.19):

$$\mathbf{z}_{\rm cent} = \frac{A_+\mathbf{z}_+ + A_-\mathbf{z}_-}{A_+ + A_-} \quad , \tag{2.29}$$

where $\mathbf{z}_{\rm cent}$ is the position of the centroid. It follows that the magnitude of the shift is

$$z_{\rm cent} = \frac{1}{2}\left(\frac{u(u^2+4)}{u^2+2} + u\right) \quad , \tag{2.30}$$

and the magnitude $\delta z_{\rm cent}$ of the shift $\delta\mathbf{z}_{\rm cent} \equiv \mathbf{z}_{\rm cent} - \mathbf{u}$ relative to the source star is

$$\delta z_{\rm cent} = \frac{u}{u^2+2} \quad . \tag{2.31}$$

Figure 2.4 shows the centroidal shift for different values of the impact parameter u_0. The trajectories trace out ellipses (Walker, 1995). Whereas the photometric effect is strongest for $u \ll 1$, the astrometric shift peaks at $u = \sqrt{2}$ with $\delta z_{\rm cent,max} \approx 0.354$ in units of the angular Einstein radius. Importantly, the astrometric shift falls off as $1/u$ for $u \gg \sqrt{2}$, which is much slower than $A \propto 1/u^4$ for the photometric effect. In this thesis I focus exclusively on the photometric microlensing effect, although many of the methods I develop in later chapters are also applicable to astrometric microlensing.

2.1.5 Magnification of an extended source

For a relatively small subset of microlensing events, the source star passes very near to the caustic. As a result, the change in magnification over the extent of the source star disc with radius $\theta_\star$ is non-negligible and the point source approximation breaks down. This breakdown will generally happen when $u_0 \lesssim \rho_\star/2$ (Gould and Gaucherel, 1997), where $\rho_\star = \theta_\star/\theta_E$. The effect of finite source effects on the magnification curve thus matters only near the peak of an event. It results in a rounder peak of the light curves shown in Figure 2.3.

There are no analytic solutions for the magnification of an extended source magnified by a single point lens with a general surface brightness profile. Gould (1994) derived an approximate solution valid for uniform brightness sources with $\rho_\star \lesssim 0.1$, and Witt and Mao (1994) derived a general solution (also for a uniform brightness source) involving solutions to elliptic integrals of the first, second and third kinds. Lee et al. (2009) proposed a simple method for numerically integrating the point source magnification over the extent of the source disc, which is fast and accurate for arbitrary intensity profiles. Another solution involving fast Fourier transforms any source profile was recently proposed by Sugiyama (2022).

To illustrate the microlensing of an extended source, in Figure 2.5 I plot the magnification map and the images of an extended limb-darkened source (with $\rho_\star = 0.15$) as it approaches the point caustic for a single lens. The top panels in the figure show the magnification map in the source plane with a logarithmic scale colourmap. The source disc (semi-transparent grey circle) is shown at different positions relative to the caustic. The bottom panels show

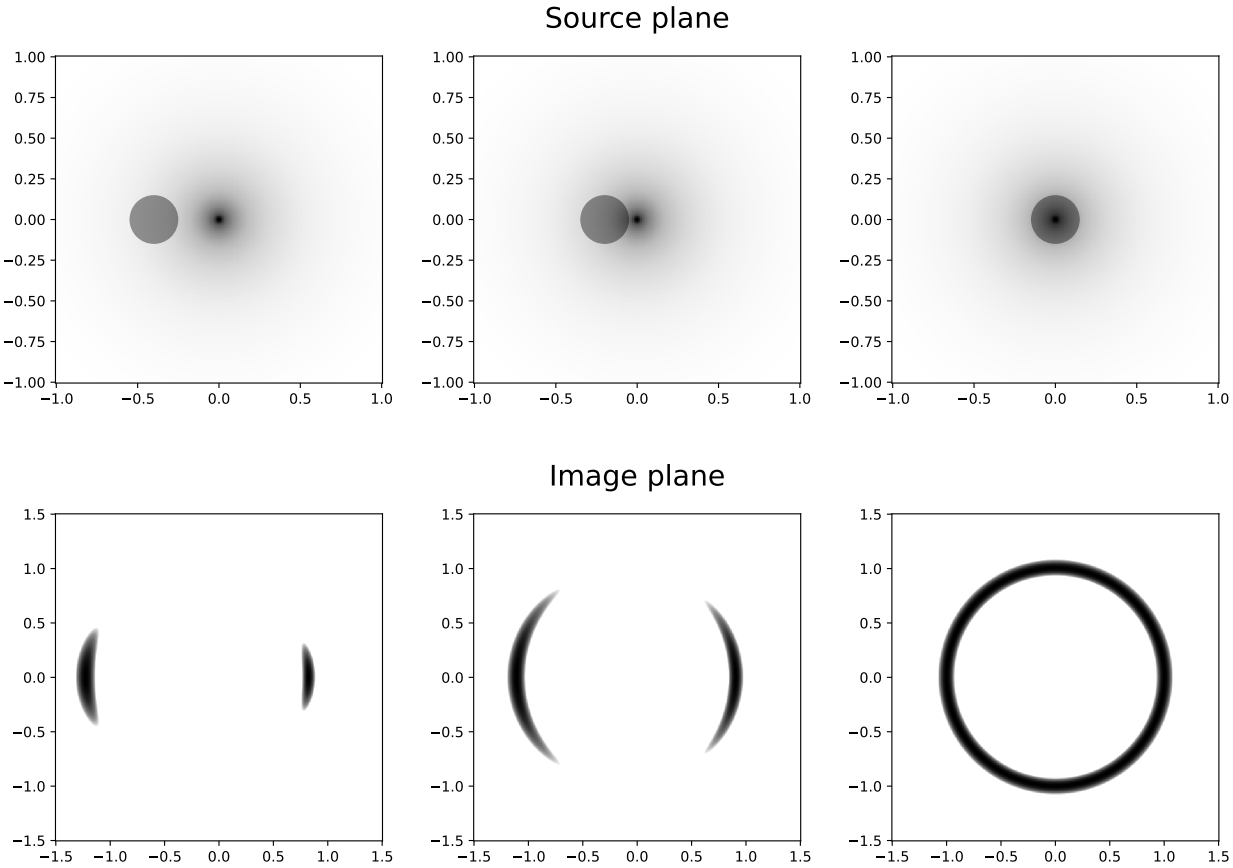


Figure 2.5: Images of an extended limb-darkened source star lensed by a single point lens for varying positions of the source star. The top panels show the magnification map with a logarithmic scale colourmap consisting of a single-point caustic. The semi-transparent circle is the source star disc with radius $\rho_\star = 0.15$. The bottom row shows the two images merging into the Einstein ring as the source moves over the caustic.

the resulting images in the image plane, which is what we would see in the sky if we could resolve the microlensing event. As the source moves closer to the caustic, the area (and hence the magnification) of the two images increases, and eventually, they merge and form an Einstein ring.

To produce these plots, I evaluated the lens mapping (Eq. 2.10) on a regular grid in the image plane and then plotted the surface brightness of the corresponding point on the source disc in the source plane. I will discuss all of this in far greater detail In Chapter 4 but, for now, I should mention a few key points:

- Every point inside the two images maps to a single point on the source disc via the lens equation. The inverse mapping is one-to-many because each point on the source disc maps to two points (point lens images) in the image plane. The points on the limbs of the two physical images correspond to points on the limb of the source disc. We cannot predict the location of the images without inverting the lens mapping

- To compute the magnification of an extended source numerically, we can either integrate the magnification function convolved with the source brightness profile over the entire source disc (a two-dimensional integral), or we can do the same integral in the

image plane. The first approach is numerically unstable because we would need to integrate over a function which diverges as $1/r$ approaching the caustic. This would require a very large number of lens equation evaluations. Hence it is preferable to integrate in the image plane where the function is smooth.

- Decreasing the source radius $\rho_\star$ results in narrower arcs of the images with smaller radial and azimuthal extent.

In Chapter 3 I discuss how things change for binary and triple lens systems where the caustics are much more complex.

2.1.6 Annual parallax

In Equation 2.23 we have assumed that the the relative motion of the lens with respect to the source on the plane of the sky is rectilinear. This is a reasonable approximation in a barycentric frame of reference if the acceleration of the source and the lens can be neglected. However, the majority of microlensing events are observed from a non-inertial geocentric frame (Earth) and sometimes from multiple locations simultaneously. In those cases, we have to take into account parallax effects. We differentiate between two kinds of parallax.

The first kind is the *annual parallax* (sometimes also called the orbital parallax). It is the change in the lens-source relative motion vector $\boldsymbol{\mu}_{LS}$ due to the local acceleration of Earth in its orbit. Annual parallax is important for long-timescale events when the event timescale is equal to some substantial fraction of a year. The effect of annual parallax is usually a slight modification of the shape of the classic Paczyński curve. The second kind of parallax is the *satellite parallax* which refers to the difference in viewpoint if we have simultaneous observations of the source star from different locations which are separated by a substantial fraction of the Einstein ring projected onto the observer plane $\tilde{r}_E \equiv D_{LS}\theta_E$ (where $D_{LS}^{-1} = D_L^{-1} - D_S^{-1}$). In practice, this means simultaneously observing a microlensing event from Earth and a space-based telescope. It was first proposed by Refsdal (1966). There is also the "terrestrial parallax", which is identical to the satellite parallax except that it involves only ground-based observations separated by a significant fraction of Earth's diameter. Measurement of these parallax effects allows for a partial breaking of the degeneracy in the event timescale t_E by providing a relationship between the mass and the distance to the lens (assuming one can estimate the distance to the source star).

In this section, I will describe the annual parallax effect because it is most relevant for this thesis. We start by modifying Equations 2.21 and 2.22 to include the projected motion of the Sun relative to the Earth on the plane of the sky. We project $\mathbf{s}(t)$ – the position vector of the Sun relative to Earth, onto a plane perpendicular to the line of sight towards the source star (plane of the sky), which is defined by the unit vector $\hat{\boldsymbol{n}}$ normal to the plane. This geocentric coordinate system is defined at some reference time t_0'. The unit vector $\hat{\boldsymbol{n}}$ depends on the sky coordinates of the source star (α, δ) (right ascension and declination). We work in geocentric equatorial coordinates defined by the spherical unit vectors $\hat{\mathbf{e}}_n$, which points North, and $\hat{\mathbf{e}}_e$ pointing eastward such that the coordinate system is right-handed.

The unit vectors are defined by

$$\hat{\mathbf{e}}_e = \hat{\mathbf{z}} \times \hat{\mathbf{n}} \tag{2.32}$$

$$\hat{\mathbf{e}}_n = \hat{\mathbf{n}} \times \hat{\mathbf{e}}_e \quad . \tag{2.33}$$

The two components of $\mathbf{s}$ projected onto the plane of the sky are then

$$\zeta_E(t; \alpha, \delta) \equiv \mathbf{s} \cdot \hat{\mathbf{e}}_e \tag{2.34}$$

$$\zeta_N(t; \alpha, \delta) \equiv \mathbf{s} \cdot \hat{\mathbf{e}}_n \quad . \tag{2.35}$$

In practice, we can retrieve $\mathbf{s}(t)$ using NASA's JPL Horizons system and compute the projected separation at any time t. The angular positions of the source and the lens (Equations 2.21 and 2.22) then become

$$\mathbf{w}_S(t) = \mathbf{w}_{S,0} + (t - t'_0)\boldsymbol{\mu}_S + \pi_S \boldsymbol{\zeta}(t) \tag{2.36}$$

$$\mathbf{w}_L(t) = \mathbf{w}_{L,0} + (t - t'_0)\boldsymbol{\mu}_L + \pi_L \boldsymbol{\zeta}(t) \quad , \tag{2.37}$$

where $\pi_S \equiv 1\,\mathrm{au}/D_S$ is the source parallax, and π_L is the lens parallax. The relative separation is

$$\boldsymbol{u}(t) = \frac{\mathbf{w}_{LS,0}}{\theta_E} + \frac{t - t'_0}{\theta_E}\boldsymbol{\mu}_{LS} + \pi_E \boldsymbol{\zeta}(t) \quad , \tag{2.38}$$

where

$$\pi_E \equiv \frac{\pi_{LS}}{\theta_E} \quad . \tag{2.39}$$

Since annual parallax is a higher-order effect affecting the apparent trajectory of the source on the sky that is often not very well constrained by the data, it makes sense to decompose the trajectory as a sum of rectilinear motion plus a deviation due to parallax (An et al., 2002; Gould, 2004). In other words, we can write (An et al., 2002)

$$\mathbf{u}(t'_0) \equiv \mathbf{u}_0 = \mathbf{w}_{LS}/\theta_E + \pi_E \boldsymbol{\zeta}(t'_0) \tag{2.40}$$

$$\dot{\mathbf{u}}(t'_0) \equiv \dot{\mathbf{u}}_0 = \boldsymbol{\mu}_{LS} + \pi_E \dot{\boldsymbol{\zeta}}(t'_0) \quad . \tag{2.41}$$

Combining the above definitions with Equation 2.38, we obtain

$$\boldsymbol{u}(t) = \mathbf{u}(t'_0) + (t - t'_0)\,\dot{\mathbf{u}}(t'_0) + \pi_E\,\delta\boldsymbol{\zeta}(t) \quad , \tag{2.42}$$

where

$$\delta\boldsymbol{\zeta}(t) = \boldsymbol{\zeta}(t) - \boldsymbol{\zeta}(t'_0) - (t - t'_0)\dot{\boldsymbol{\zeta}}(t'_0) \tag{2.43}$$

is the position offset of the Sun on the plane of the sky relative to its position at the reference time t'_0. By construction, we have $\delta\boldsymbol{\zeta}(t'_0) = 0$ and $\delta\dot{\boldsymbol{\zeta}}(t'_0) = 0$. At the reference time t'_0, the vectors $\mathbf{u}(t'_0)$ and $\dot{\mathbf{u}}(t'_0)$ are perpendicular to each other.

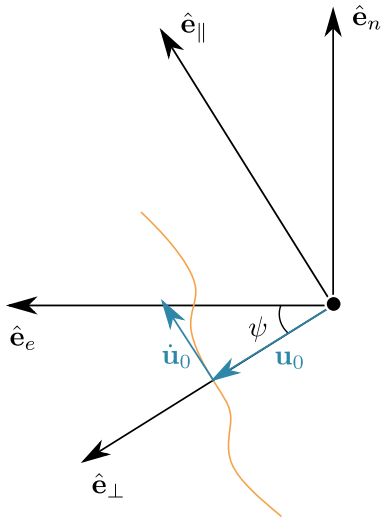


Figure 2.6: A coordinate system $(\hat{\mathbf{e}}_\perp, \hat{\mathbf{e}}_\parallel)$ parallel to the source trajectory at time t_0. The orange curve represents the source trajectory relative to the lens at the origin. The coordinate system $(\hat{\mathbf{e}}_\perp, \hat{\mathbf{e}}_\parallel)$ is related to the equatorial coordinates $(\hat{\mathbf{e}}_e, \hat{\mathbf{e}}_n)$ by a rotation through an angle ψ.

$\mathbf{u}(t)$ in the $(\hat{\mathbf{e}}_\perp, \hat{\mathbf{e}}_\parallel)$ coordinate system

To evaluate Equation 2.43, we need to choose a suitable basis. A natural coordinate system for describing the trajectory of the source relative to the lens is one defined by unit vectors $(\hat{\mathbf{e}}_\perp, \hat{\mathbf{e}}_\parallel)$ where $\hat{\mathbf{e}}_\parallel$ is parallel to the trajectory $\mathbf{u}(t)$ at time t_0' (Figure 2.6). We define the unit vectors as

$$\hat{\mathbf{e}}_\perp \equiv \frac{\mathbf{u}_0}{|\mathbf{u}_0|} \quad , \hat{\mathbf{e}}_\parallel \equiv \frac{\hat{\mathbf{n}} \times \mathbf{u}_0}{|\mathbf{u}_0|} \quad . \tag{2.44}$$

The coordinate system $(\hat{\mathbf{e}}_\perp, \hat{\mathbf{e}}_\parallel)$ is related to ecliptic coordinates by a simple rotation through an angle ψ

$$\begin{pmatrix} \hat{\mathbf{e}}_\perp \\ \hat{\mathbf{e}}_\parallel \end{pmatrix} = \begin{pmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{pmatrix} \begin{pmatrix} \hat{\mathbf{e}}_e \\ \hat{\mathbf{e}}_n \end{pmatrix} \quad . \tag{2.45}$$

By construction, at time t_0' we have $\mathbf{u}_0 \perp \dot{\mathbf{u}}_0$ so the two components of $\mathbf{u}(t)$ are then:

$$u_\perp(t) \equiv \mathbf{u}(t) \cdot \hat{\mathbf{e}}_\perp = u_0 + \pi_E \, \delta\boldsymbol{\zeta}(t) \cdot \hat{\mathbf{e}}_\perp \tag{2.46}$$

$$u_\parallel(t) \equiv \mathbf{u}(t) \cdot \hat{\mathbf{e}}_\parallel = (t - t_0') \, \dot{\mathbf{u}}_0 \cdot \hat{\mathbf{e}}_\parallel + \pi_E \, \delta\boldsymbol{\zeta}(t) \cdot \hat{\mathbf{e}}_\parallel \quad . \tag{2.47}$$

Using the definitions of the unit vectors and $\delta\boldsymbol{\zeta}(t) = \delta\zeta_E(t)\,\hat{\mathbf{e}}_e + \delta\zeta_N(t)\,\hat{\mathbf{e}}_n$, we have

$$u_\perp(t) = u_0 + \pi_E \, \cos\psi \, \delta\zeta_E(t) - \pi_E \, \sin\psi \, \delta\zeta_N(t) \tag{2.48}$$

$$u_\parallel(t) = (t - t_0')/t_E' + \pi_E \, \sin\psi \, \delta\zeta_E(t) + \pi_E \, \cos\psi \, \delta\zeta_N(t) \quad , \tag{2.49}$$

where $t_E' \equiv |\dot{\mathbf{u}}_0| = |\boldsymbol{\mu}_{LS}/\theta_E + \pi_E \, \dot{\boldsymbol{\zeta}}(t_0')|$. The model parameters that determine the magnification $A(t)$ are $(u_0, t_0', t_E', \pi_E, \psi)$. Notice that in this case, u_0 can be negative.

An alternative parametrisation which is more commonly used in the literature is obtained by defining components of a "microlensing parallax vector" as

$$\pi_{E,N} \equiv \pi_E \cos\psi \tag{2.50}$$
$$\pi_{E,E} \equiv \pi_E \sin\psi \quad , \tag{2.51}$$

in which case we have

$$u_\perp(t) = u_0 + \pi_{E,N}\,\delta\zeta_E(t) - \pi_{E,E}\,\delta\zeta_N(t) \tag{2.52}$$
$$u_\parallel(t) = (t - t'_0)/t'_E + \pi_{E,E}\,\delta\zeta_E(t) + \pi_{E,N}\,\delta\zeta_N(t) \quad , \tag{2.53}$$

and the model parameters are $(u_0, t'_0, t'_E, \pi_{E,E}, \pi_{E,N})$.

For reference, we can also write down the components of $\mathbf{u}(t)$ in equatorial coordinates by inverting the rotation matrix in Equation 2.45 and applying it to Equations 2.48 and 2.49. The two components are

$$u_e = u_0 \cos\psi + (t - t'_0)/t'_E \sin\psi + \pi_E\,\delta\zeta_E(t) \tag{2.54}$$
$$u_n = -u_0 \sin\psi + (t - t'_0)/t'_E \cos\psi + \pi_E\,\delta\zeta_N(t) \quad . \tag{2.55}$$

$\mathbf{u}(t)$ in the $(\hat{\mathbf{e}}_\parallel, \hat{\mathbf{e}}_\perp)$ coordinates using acceleration parameters

There exists another parametrisation of the trajectory in which we fit for the local acceleration of the lens at t'_0 instead of π_E and the angle ψ. I derive it here for completeness. We define the position, velocity and acceleration such that at $t = t'_0$ we have

$$\mathbf{u}(t'_0) \equiv \widetilde{u}_0\,\hat{\mathbf{e}}_\perp \tag{2.56}$$
$$\dot{\mathbf{u}}(t'_0) \equiv \frac{1}{\widetilde{t'_E}}\,\hat{\mathbf{e}}_\parallel \tag{2.57}$$
$$\ddot{\mathbf{u}}(t'_0) \equiv a_\perp\,\hat{\mathbf{e}}_\perp + a_\parallel\,\hat{\mathbf{e}}_\parallel \quad , \tag{2.58}$$

where $a_\parallel$ and $a_\perp$ are the two components of the instantaneous acceleration of the lens at t'_0. From Equations 2.48 and 2.49 it follows that

$$\mathbf{u}(t'_0) = u_0\,\hat{\mathbf{e}}_\perp \tag{2.59}$$
$$\dot{\mathbf{u}}(t'_0) = \frac{1}{t'_E}\,\hat{\mathbf{e}}_\parallel \tag{2.60}$$
$$\ddot{\mathbf{u}}(t'_0) = \left[\pi_E \cos\psi\,\delta\ddot{\zeta}_e(t'_0) - \pi_E \sin\psi\,\delta\ddot{\zeta}_n(t'_0)\right]\,\hat{\mathbf{e}}_\perp \tag{2.61}$$
$$+ \left[\pi_E \sin\psi\,\delta\ddot{\zeta}_e(t'_0) + \pi_E \cos\psi\,\delta\ddot{\zeta}_n(t'_0)\right]\,\hat{\mathbf{e}}_\parallel \quad . \tag{2.62}$$

By equating the components of the position, velocity and acceleration vectors, we obtain the expressions for the old parameters in terms of the new parameters as

$$\widetilde{u}_0 = u_0, \quad \widetilde{t'_E} = t'_E \quad \pi_E = \sqrt{\frac{a_\parallel^2 + a_\perp^2}{1}\left[\delta\ddot{\boldsymbol{\zeta}}_e(t'_0)\right]^2 + \left[\delta\ddot{\boldsymbol{\zeta}}_n(t'_0)\right]^2} \quad . \tag{2.63}$$

Similarly, we have

$$\pi_{E,N} = \pi_E \cos\psi = \frac{a_\parallel \, \delta\ddot{\zeta}_n(t_0') + a_\perp \, \delta\ddot{\zeta}_e(t_0')}{[\delta\ddot{\zeta}_e(t_0')]^2 + [\delta\ddot{\zeta}_n(t_0')]^2} \tag{2.64}$$

$$\pi_{E,E} = \pi_E \sin\psi = \frac{a_\parallel \, \delta\ddot{\zeta}_e(t_0') - a_\perp \, \delta\ddot{\zeta}_n(t_0')}{[\delta\ddot{\zeta}_e(t_0')]^2 + [\delta\ddot{\zeta}_n(t_0')]^2} \quad . \tag{2.65}$$

To obtain the trajectory in terms of these new parameters, we plug in the above expressions into Equations 2.48 and 2.49. The new parameter set is $\left(u_0, t_0', t_E', a_\parallel, a_\perp\right)$. This parametrisation makes it is evident that the parallax effect depends on the apparent local acceleration of the source at time t_0'.

2.1.7 Measuring the lens mass

Notice that the only quantity of physical interest in single lens microlensing – the lens mass M – is buried inside of the definition of the angular Einstein radius θ_E, which, when combined with the magnitude of the relative proper motion $|\boldsymbol{\mu}_{LS}|$ forms the observable t_E. In practice, depending on the microlensing event, there are several possible channels which enable an estimate of the lens mass. For example, a measurement of θ_E from finite-source effects or from a direct measurement of $|\boldsymbol{\mu}_{LS}|$ can be combined with π_E to yield (via Equations 2.7 and 2.39)

$$M = \frac{\theta_E}{\kappa \pi_E} \quad . \tag{2.66}$$

With an estimate of the distance to the source star, one can also obtain the distance to the lens. An alternative, less direct approach is to use a measurement of θ_E combined with the measurement of the lens flux to estimate the mass and distance to the lens (Bennett et al., 2007). Note that either of these approaches is possible for only a small subset of all detected microlensing events.

2.1.8 A system with N lenses

Microlensing with more than one lens is considerably more complex. Consider a system with N point mass lenses with a total mass $M = \sum_{i=1}^{N} m_i$. As before, we have the angular source position in the source plane, given by the dimensionless vector $\mathbf{z} = \boldsymbol{\beta}/(D_\mathrm{S}\theta_E)$ and the position of the images in the lens plane, given by given by $\mathbf{w} = \boldsymbol{\theta}/(D_\mathrm{L}\theta_E)$, where θ_E now refers to the angular Einstein radius corresponding to the total mass M. The lens equation then contains a sum over the deflection angles $\boldsymbol{\alpha}_i$ corresponding to each point mass m_i[3]:

$$\mathbf{z} = \mathbf{w} - \sum_{i=1}^{N} \boldsymbol{\alpha}_i(\mathbf{w}, \mathbf{w}_i) \quad , \tag{2.67}$$

[3]This is because we are working within the linearised GR framework where the deflection angles are additive.

where $\boldsymbol{\alpha}_i$ is the deflection angle due to the ith lens. Using Equation 2.10, we obtain

$$\mathbf{z} = \mathbf{w} - \sum_{i=1}^{N} \epsilon_i \frac{\mathbf{w} - \mathbf{w}_i}{|\mathbf{w} - \mathbf{w}_i|^2} \quad , \tag{2.68}$$

where $\epsilon_i \equiv m_i/M$.

It is helpful to rewrite Equation 2.68 in complex form using complex variables instead of 2D vectors $\mathbf{w}$ and $\mathbf{z}$ (Witt, 1990):

$$w \equiv w_1 + iw_2, \quad z \equiv w_1 + iw_2 \quad . \tag{2.69}$$

The lens equation then takes the form

$$w = z - \sum_{i=1}^{N} \frac{\epsilon_i}{\bar{z} - \bar{z}_i} \quad . \tag{2.70}$$

Throughout the rest of the thesis, I will use the complex notation for the lens equation, for reasons that will become apparent shortly.

The mapping from the image plane z to the source plane w is one-to-one, and it is straightforward to compute using Equation 2.70. The inverse mapping is more challenging. We start by rewriting Equation 2.70 as a complex polynomial of degree $N^2 + 1$ (see Appendix A). From the Fundamental Theorem of Algebra, it follows that such a polynomial has exactly $N^2 + 1$ complex roots. However, not all of these roots are necessarily solutions to the lens equations, so in practice, one first has to compute all of the polynomial roots and then discard those which do not satisfy Equation 2.70. For binary lenses, there are always either 3 or 5 real images, and for $N \geq 2$ the maximum number of images is $5(N-1)$ (Rhie, 2001, 2003; Khavinson and Neumann, 2004). As before, the magnification is given by the inverse Jacobian determinant of the mapping $z \to w$ evaluated at the images. We have

$$\mathbf{J} = \begin{pmatrix} \frac{\partial w}{\partial z} & \frac{\partial w}{\partial \bar{z}} \\ \frac{\partial \bar{w}}{\partial z} & \frac{\partial \bar{w}}{\partial \bar{z}} \end{pmatrix} \quad , \tag{2.71}$$

and

$$\det \mathbf{J} = \frac{\partial w}{\partial z}\frac{\partial \bar{w}}{\partial \bar{z}} - \frac{\partial w}{\partial \bar{z}}\frac{\partial \bar{w}}{\partial z} = \left|\frac{\partial w}{\partial z}\right|^2 - \left|\frac{\partial w}{\partial \bar{z}}\right|^2 \quad , \tag{2.72}$$

where I have used the identities $\overline{\frac{\partial \bar{a}}{\partial \bar{b}}} = \frac{\partial a}{\partial b}$ and $a\bar{a} = |a|^2$.

Finally, by evaluating the partial derivatives using Equation 2.70 we obtain

$$\det \mathbf{J} = 1 - \left| \sum_{i=0}^{N} \frac{\epsilon_i}{(\bar{z} - \bar{z}_i)^2} \right|^2 \quad , \tag{2.73}$$

so the magnification is given by

$$A = \sum_j \frac{1}{|\det \mathbf{J}|_j} \quad , \tag{2.74}$$

where j denotes the j-th image. The points in the image plane where the Jacobian determinant vanishes ($\det \mathbf{J} = 0$) are the critical curves:

$$\left| \sum_{i=0}^{N} \frac{\epsilon_i}{(\bar{z} - \bar{z}_i)^2} \right|^2 = 1 \quad . \tag{2.75}$$

The points on the critical curve mapped to the source plane via the lens equation (Equation 2.70) are the caustic curves. The difference compared to the single lens lens equation is that the caustics in this case are closed curves rather than a single point. The caustic curves are comprised of concave segments called *folds* which are connected at points called *cusps*. Equation 2.75 can also be written as a complex polynomial of degree $2N$ (see Appendix A.2) so there are, at most, $2N$ critical and caustic curves. The complexity of microlensing events involving multiple lenses is a direct consequence of the non-smooth nature of caustic curves.

2.1.9 Binary lenses

In this section, we will focus specifically on the binary lens. We choose a coordinate system whose origin is at the midpoint of the line connecting the two lenses, which are on the real axis. If the first lens is at distance a ($z_1 = a$) then the second lens is at $-a$ ($z_2 = -a$), the lens equation takes the form

$$w = z + \frac{\epsilon_1}{\bar{z} - a} + \frac{1 - \epsilon_1}{\bar{z} + a} \quad . \tag{2.76}$$

This equation can be rewritten as a 5th-order complex polynomial (see Appendix A). Either 3 or 5 of the complex roots of that polynomial are also solutions to the lens equation (5 inside the caustics and 3 outside). It is common to use the mass ratio $q \equiv \epsilon_2/\epsilon_1$ with $\epsilon_2 \leq \epsilon_1$ instead of ϵ_1 as a parameter and the separation between the lenses $s \equiv 2a$ instead of half the separation a. As a reminder, all angular quantities are expressed in the units of the angular Einstein radius of the total mass of the system.

Depending on the separation between the two lenses, s, the critical and the caustic curves can have three distinct topologies, which are labelled *close*, *intermediate*, and *wide*. These are shown in Figure 2.7 for a system with $q = 1$ where on the left we see the critical curves in the lens plane, and on the right are the caustic curves in the source plane. The critical and caustic curves are symmetric with respect to the x-axis. The sharp structure of the caustic curves compared to the smooth critical curves is due to the non-linearity of the lens mapping. The number of different topologies is independent of the mass ratio.

Caustics shown in Figure 2.7 correspond to an equal mass binary lens. Systems with $q \ll 1$ are of more interest because they correspond to planetary systems. The lower mass ratio does not change the topological structure of the caustics (except for shifting the boundaries between the different topologies), but it does change their shape and size. In Figure 2.8 we show the point source *magnification maps* (computed using the caustics code, which is the subject of Chapter 4) for a binary lens with $q = 5 \times 10^{-3}$, roughly corresponding to a gas giant planet orbiting an M dwarf. For the close topology (first two panels from the top in Figure 2.8), we see one caustic centred on the star, which is called a *central caustic*, and two

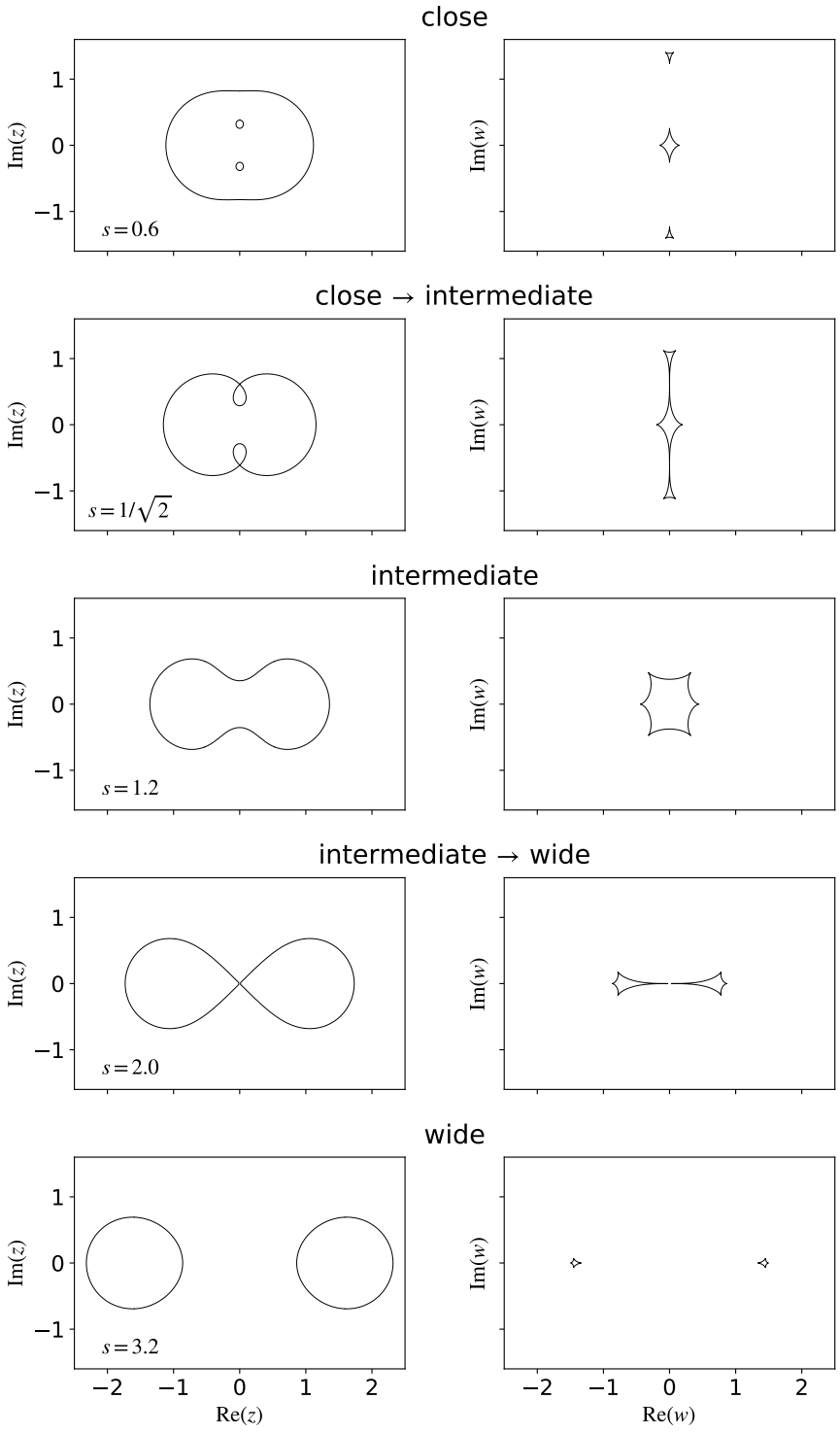


Figure 2.7: The three different topologies of the binary lens. The left panels show the critical curves in the lens plane. The right panels show the caustic curves in the source plane. The figure shows all three topologies and their transitions for an equal-mass binary lens with $q = 1$. In this case, the transition between the close and intermediate topology occurs at $s = 1/\sqrt{2}$ and the one between the intermediate and wide topology at $s = 2.0$. Figure adapted from Dominik (1999).

additional caustics on the opposite side of the planet, symmetric with respect to the star-

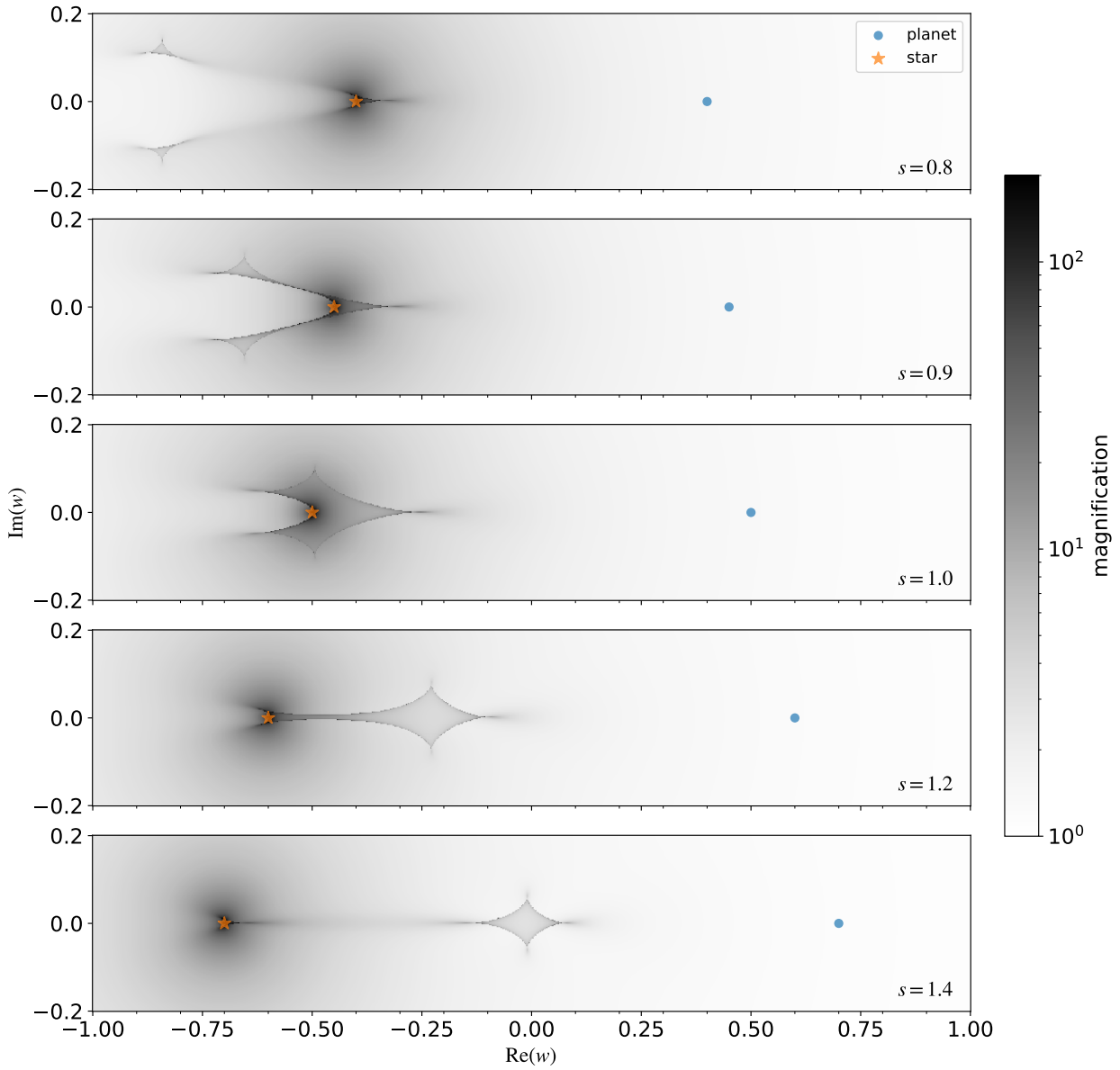


Figure 2.8: Caustic structure for a binary lens with $q = 0.003$, shown for different values of the separation s. The orange star denotes the position of the star and the blue dot designates the planet. The transition between the close and intermediate topologies occurs at $s \approx 0.92$ and the one between intermediate and wide topologies at $s \approx 1.21$. The vertical grey dashed line shows the angular Einstein radius at $y_1 = 1$.

planet axis, which are called *planetary caustics* because they are associated with the planet rather than the star. Notice that the magnification in the vicinity of these planetary caustics is significantly smaller than the magnification around the central caustic. Because of this, it is easier to search for binary events with trajectory passing close to the central caustic rather than the planetary caustics. As the separation s approaches s_c – the critical value between the close and wide topologies, the two planetary caustics merge with the central caustic into a single caustic often called the *resonant caustic*, with a much larger cross-section than either the central or the planetary caustics (third panel from the top). Finally, as the separation s increases beyond the intermediate/wide boundary, the resonant caustic separates into a

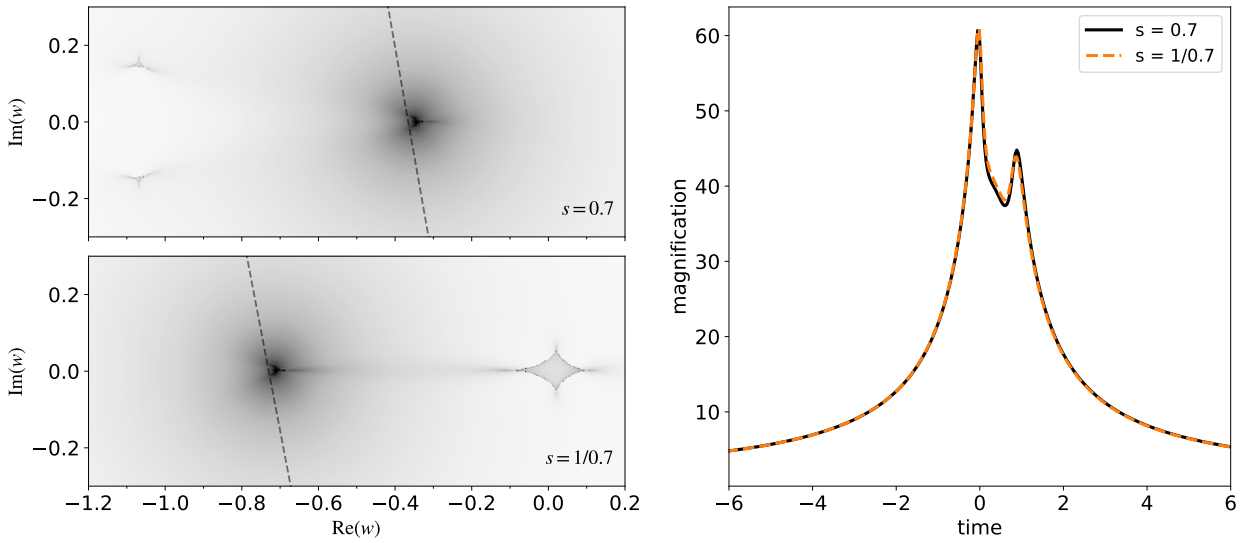


Figure 2.9: Illustration of the "close/wide" degeneracy in binary microlensing events. The two panels on the left show the magnification maps (on a logarithmic scale) for a binary lens with $q = 5 \times 10^{-3}$, $s = 0.7$ (top panel), and $s = 1/0.7$ (bottom panel). The dashed grey line indicates the source trajectory. The panel on the right shows the corresponding magnification as a function of time. We see that trajectories through two very different configurations of the binary lens result in nearly identical magnification curves. In absence of good photometric coverage near the central caustic crossing, this approximate degeneracy can become exact.

smaller central caustic centred on the star and a single, larger, planetary caustic located on the star-planet axis (bottom panel). The size of this planetary caustic scales as $q^{1/2}s^{-2}$ (see references in review by Gaudi, 2012).

Given caustics such as the ones shown in Figure 2.8, the magnification as a function of time depends on the source trajectory in the source plane. Absent parallax, it is just a slice through the magnification map convolved with the source star brightness profile. Herein lies the complexity of microlensing. *The trajectory of the source star over the caustic patterns is entirely random, and the magnification response is highly non-linear.* The consequence of this fact is there is a large variety of possible light curve morphologies and the parameter space of the models is highly complex.

One notable feature of the binary lens is that in the limit $q \ll 1$ and $|s - 1| \gg q$, the structure of the caustic curves are invariant to the $s \rightarrow s^{-1}$ transformation (Dominik, 1999), which is known in the literature as the *close/wide degeneracy*, and it often arises for microlensing events involving planets. Figure 2.9 illustrates this (approximate) degeneracy. The two panels on the left show the magnification maps of a binary lens with $q = 5 \times 10^{-3}$ and two different values of the separation s related by the transformation $s \rightarrow s^{-1}$. The dashed grey line shows the trajectory of the source passing close to the central caustic. The panel on the right shows that the magnification as a function of time is nearly identical for the two different configurations of the binary lens. In the absence of good photometric coverage near the peak of the event such approximate degeneracies can become exact. These kinds of data-dependent degeneracies (rather than exact mathematical symmetries) are a very common feature of microlensing (see, for instance Erdl and Schneider, 1993).

Parametrising the trajectory

To parametrise the trajectory $\mathbf{u}(t)$, including the annual parallax effect, we have to use an additional angle to specify the orientation of the axis containing both lenses. We can build on the work done in Section 2.1.6, and instead of setting the origin of the coordinate system as the location of the single lens, we set it to the midpoint between the two lenses which lie on the same axis. $\mathbf{u}_0$ is the point on the trajectory closest to the midpoint and $\dot{\mathbf{u}}_0$ is the velocity vector at that point. To specify the trajectory of the source in this new coordinate system, we just need to rotate the coordinate system $(\hat{\mathbf{e}}_\parallel, \hat{\mathbf{e}}_\perp)$ by α, where α is the angle between the $\hat{\mathbf{e}}_\parallel$ and the axis containing the two lenses. Using Equations 2.48 and 2.49 we have

$$\mathbf{u}(t) = \mathbf{R}(\alpha) \begin{pmatrix} u_\perp(t) \\ u_\parallel(t) \end{pmatrix} = \mathbf{R}(\alpha) \begin{pmatrix} u_0 + \pi_E \cos\psi\, \delta\zeta_E(t) - \pi_E \sin\psi\, \delta\zeta_N(t) \\ (t - t_0)/t'_E + \pi_E \sin\psi\, \delta\zeta_E(t) + \pi_E \cos\psi\, \delta\zeta_N(t) \end{pmatrix} \quad , \tag{2.77}$$

where $\mathbf{R}(\alpha)$ is the rotation matrix given by

$$\mathbf{R}(\alpha) = \begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix} \quad . \tag{2.78}$$

The magnification of a binary lens is then fully parametrised by the following parameters

$$(u_0, t_0, t'_E, \pi_E, \psi, a, q, \alpha) \quad . \tag{2.79}$$

2.1.10 Triple lenses

The first detected triple-lens microlensing event was OGLE-2006-BLG-109Lb,c (Gaudi et al., 2008; Bennett et al., 2010), a system consisting of two massive planets and a star, similar to Jupiter and Saturn in the Solar System. A circumbinary planet was discovered by (Bennett et al., 2016). There is also a possibility of detecting triple lens exomoon systems, although none have been confirmed so far (Liebig and Wambsganss, 2010).

To parametrise a triple-lens system, we use the same coordinate system as for the binary lens, adding a third lens at an arbitrary location z_3 in the source plane. The lens equation is

$$w = z + \frac{\epsilon_1}{\bar{z} - a} + \frac{\epsilon_2}{\bar{z} + a} + \frac{1 - \epsilon_1 - \epsilon_2}{\bar{z} - \bar{z}_3} \quad . \tag{2.80}$$

The complex polynomial derived from the triple-lens lens equation is a 10th-degree polynomial.

A triple-lens system's caustic structure is vastly more complex than in the case of the binary lens. Daněk and Heyrovský (2019) investigate three different kinds of systems: equal masses for all three lenses, two equal-mass lenses, and a low-mass third component and a hierarchical combination of the three masses. They find 11 different kinds of topologies of the caustics. Depending on the mass ratios, there could be even more kinds of topologies. In Figure 2.10 I illustrate an example caustic structure for a triple-lens system with $\epsilon_1 = 0.9$, $\epsilon_2 = \epsilon_e = 0.05$, $s = 0.8$ and $z_3 = 0.3 - i0.8$. The key features to note are that the caustic pattern is no longer symmetric about the x-axis and the caustics can be nested and self-intersecting.

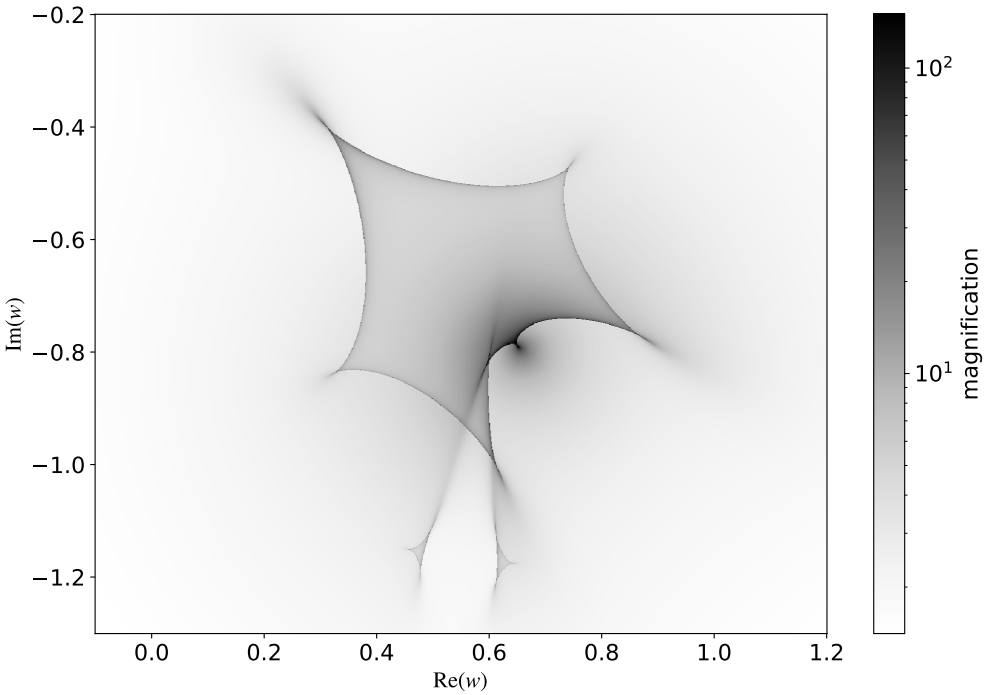


Figure 2.10: Caustic structure for a triple-lens with $\epsilon_1 = 0.9$, $\epsilon_2 = \epsilon_e = 0.05$, $s = 0.8$ and $z_3 = 0.3 - i0.8$. The caustics are considerably more complex than in the binary case and they can be nested and self-intersecting.

2.1.11 Other effects

There are a few other effects that often need to be considered in realistic microlensing models. I will not cover these in detail except for the first one but I list them below for completeness.

- **Finite source effects:** accurately computing the magnification of a limb-darkened extended source intersecting a caustic is anything but trivial. This is the subject of Chapter 4.

- **Multiple sources:** instead of a single source star, we could also have events with binary source stars (Dominik, 1998b). These can sometimes mimic binary-lens events with a single source star. Modelling binary source star events requires specifying the trajectory and separation of the binary and the total flux is then the sum of the flux from the primary and the secondary star. There is a paucity of binary source events in the microlensing literature, partly due to their intrinsic rarity (Han and Jeong, 1998), and partly because there is a bias towards focusing on binary-lens models in the community, and simply not putting as much effort into testing binary source models as often as binary-lens models (Jung et al., 2017; Dominik et al., 2019).

- **Orbital motion of the lenses:** if the orbital timescale of the lens in binary or triple-lens systems is comparable to the event timescale (this is true for a small subset of events) we need to take into account the orbital motion of the lens projected onto the plane of the sky. In the general case of a Keplerian orbit, five additional parameters are needed in the binary-lens model: the mass of the primary lens, the three components of the secondary's velocity relative to the primary, and the projected separation of the

secondary along the line of sight in units of θ_E (Dominik, 1998a). In practice, only two additional parameters can be measured: the sky projection of the two components of the velocity of the secondary relative to the primary. These are parametrised by $\gamma_{\parallel} \equiv \dot{s}/s$ – the fractional rate of change of the projected separation between the two lenses, and $\gamma_{\perp} \equiv -\dot{\alpha}$ – the angular rotation rate of the projected separation axis. The effect of $\gamma_{\perp}$ is simply rotating the magnification pattern on the sky, while a nonzero value of $\gamma_{\parallel}$ changes the magnification pattern itself. These additional parameters are partially degenerate with other parameters, such as parallax, and it is very rarely possible to constrain the complete Keplerian orbit (Skowron et al., 2011; Wyrzykowski et al., 2020).

- **Orbital motion of the sources:** the effect of the orbital motion of a binary source star is less dramatic than the orbital motion of the lens stars because the caustic structure stays fixed. Dominik (1998a) showed that six additional parameters are needed to describe the full Keplerian orbit of a binary source star. Griest and Hu (1992, 1993) pointed out that such events should be rare, although, it is not clear if that is still the case today with the advent of new microlensing surveys. There is also a possibility that the secondary source is not a star but a giant planet resulting in subtle deviations in the primary source position as it orbits the barycentre. This effect is often called *xallarap* (inverse parallax) in the literature and it is an alternative channel for detecting planets using microlensing (Rahvar and Dominik, 2009; Rota et al., 2021; Miyazaki et al., 2021).

- **Fine structure in the source profiles:** microlensing in principle allows for the measurement of the source stars' intensity and shape profile and possible planets in orbit around them. Heyrovský and Loeb (1997) investigate the microlensing of elliptical sources by a single lens, as a model for oblate stars and inclined accretion discs. Gaudi and Gould (1999) discuss the sensitivity of binary and single lens caustic-crossing events to spatial and spectral structure of stellar atmospheres. Gaudi et al. (2003) focused specifically on the possibility of inferring the intensity and shape profiles of caustic-crossing *planets* in the source plane. Deviations in the light curve caused by a non-uniform surface brightness profile or shape of the planet are only resolved during a short time window when the planet is within about one radius from the caustic. Gaudi et al. (2003) find that there is a possibility of detecting ring-like structures around the planet using high-cadence observations from a $\sim$ 30m class telescope, but detecting features such as spots and zonal bands will be very difficult. Detecting differences in the *phase* of the planet may be easier (Ashton and Lewis, 2001).

2.2 Occultation and phase curve mapping

Moving on from microlensing, in this section I will discuss the theory behind modelling phase curves and occultation (eclipse) light curves of spherical bodies for the case of emitted light (isotropic blackbody emission from the body itself) and reflected light (starlight scattered from the surface or atmosphere of the body). In a way, the problem is similar to microlensing in that the goal is to infer two-dimensional structure (caustic map in one case, emission or

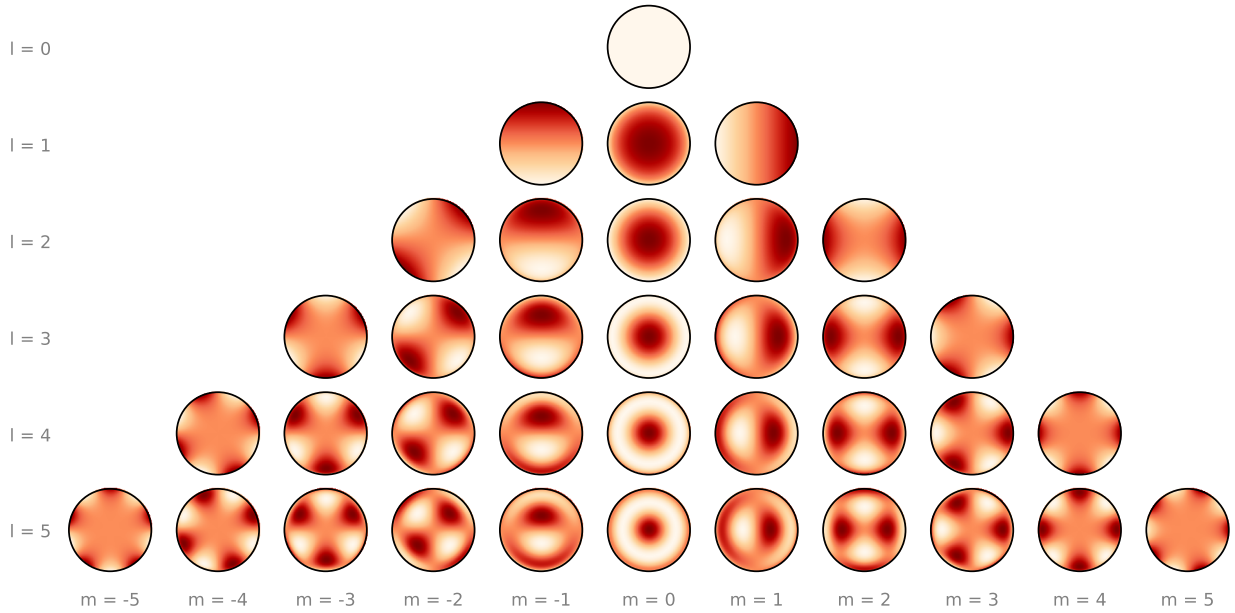


Figure 2.11: Real spherical harmonics up to degree $l = 5$ computed from Equation 2.84. Figure adapted from Figure 1 in Luger et al. (2019).

albedo map in the other) from one-dimensional photometric light curves. In both cases the inverse problem is highly degenerate but the key difference is that phase curve and occultation mapping can be cast in the form of a *linear* problem, while microlensing is highly non-linear problem. This makes all the difference in the world when it comes to solving the inverse problem.

What follows is mostly a short summary of the papers introducing the starry framework (Luger et al., 2019, 2022a).

2.2.1 The starry algorithm

Emitted light

Consider a spherical body of unit radius emitting light isotropically (a Lambertian emitter) at each point (θ, ϕ) where θ is the inclination angle and ϕ is the azimuthal angle. To compute a phase curve or an occultation light curve we need to be able to integrate the flux over the entire projected disc of the body given the distribution of specific intensity $I(\theta, \phi)$, the 3D orientation of the body in space, and the location of a possible occultor relative to the body. These two-dimensional integrals can always be computed numerically (see for instance the approaches presented in Farr et al. (2018); Louden and Kreidberg (2018)), but Luger et al. (2019) showed that it is possible to compute them analytically if the specific intensity is first expanded in the basis of spherical harmonics and then mapped to a different basis set more suitable for evaluating the integrals.

Following Luger et al. (2019), we start by setting up a cartesian coordinate system on

the unit sphere, such that

$$x = \sin\theta\cos\phi \tag{2.81}$$
$$y = \sin\theta\sin\phi \tag{2.82}$$
$$z = \cos\theta \quad . \tag{2.83}$$

The observer is located on the z-axis at ∞ such that the projected disc of the body is centred at the origin of the $x-y$ plane with the unit vector $\hat{\mathbf{x}}$ pointing to the right and $\hat{\mathbf{y}}$ pointing up. We introduce spherical harmonics $Y_{lm}(\theta,\phi)$ of degree $l \geq 0$ and order $m \in [-l, l]$ defined as

$$Y_{lm}(\theta,\phi) = \begin{cases} \bar{P}_{lm}(\cos\theta)\cos(m\phi) & m \geqslant 0 \\ \bar{P}_{l|m|}(\cos\theta)\sin(|m|\phi) & m < 0 \end{cases} \quad . \tag{2.84}$$

When the spherical harmonics defined above are rewritten in terms of x, y, and z, they become polynomials in these variables (see Appendix A in Luger et al. (2019)). We can expand the specific intensity distribution $I(x,y)$ in a spherical harmonic basis as

$$I(x,y) = \tilde{\mathbf{y}}^\intercal(x,y)\mathbf{y} \quad , \tag{2.85}$$

where $\tilde{\mathbf{y}}(x,y)$ is the basis vector of spherical harmonics arranged in increasing order:

$$\tilde{\mathbf{y}} = \big(\; Y_{0,0} \quad Y_{1,-1} \quad Y_{1,0} \quad Y_{1,1} \quad Y_{2,-2} \tag{2.86}$$
$$Y_{2,-1} \quad Y_{2,0} \quad Y_{2,1} \quad Y_{2,2} \quad \cdots \big)^\intercal \quad , \tag{2.87}$$

and $\mathbf{y}$ is a vector of scalar *spherical harmonic coefficients*. $\mathbf{y}$ is a quantity of central importance because it parametrises the body's *map* (the specific intensity at every point). Luger et al. (2019) show that we can represent $\tilde{\mathbf{y}}$ in a polynomial basis (see Luger et al. (2019) for the general expression)

$$\tilde{\boldsymbol{p}} = \big(\; 1 \quad x \quad z \quad y \quad x^2 \quad xz \quad xy \quad yz \quad y^2 \quad \cdots \big)^\intercal \quad , \tag{2.88}$$

such that

$$I(x,y) = \tilde{\mathbf{p}}^\intercal(x,y)\mathbf{p} \tag{2.89}$$
$$= \tilde{\mathbf{p}}^\intercal(x,y)\mathbf{A}_1\mathbf{y} \quad , \tag{2.90}$$

and also in the Green's basis

$$\tilde{\boldsymbol{g}} = \big(\; 1 \quad 2x \quad z \quad y \quad 3x^2 \quad -3xz \quad 2xy \quad 3yz \quad y^2 \quad \cdots \big)^\intercal \quad , \tag{2.91}$$

such that

$$I(x,y) = \tilde{\mathbf{g}}^\intercal(x,y)\mathbf{g} \tag{2.92}$$
$$= \tilde{\mathbf{g}}^\intercal(x,y)\mathbf{A}_2\mathbf{p} \tag{2.93}$$
$$= \tilde{\mathbf{g}}^\intercal(x,y)\mathbf{A}\mathbf{y} \quad , \tag{2.94}$$

where $\mathbf{A}_1$ is the change-of-basis matrix from $\mathbf{y}$ to $\mathbf{p}$, $\mathbf{A}_2$ is the change-of-basis matrix from $\mathbf{p}$ to $\mathbf{g}$, and $\mathbf{A} \equiv \mathbf{A}_1\mathbf{A}_2$.

To compute rotational light curves, we also need to be able to specify the orientation of a surface map with coefficients $\mathbf{y}$. We can rotate the map using the Wigner rotation matrix $\mathbf{R}(\mathrm{I}, \Lambda, \Theta)$ that rotates $\mathbf{y}$ given the body's inclination I, obliquity Λ and rotational phase Θ. The rotated map is then $\mathbf{R}(\mathrm{I}, \Lambda, \Theta)\mathbf{y}$, and Equation 2.90 can be rewritten more generally as

$$I(x, y) = \tilde{\mathbf{p}}^\intercal(x, y)\mathbf{A}_1\mathbf{R}\mathbf{y} \quad . \tag{2.95}$$

The total flux measured by an observer is given by an integral of the specific intensity over a region S of the projected disc of the body:

$$F = \iint_S I(x, y)\, \mathrm{d}S \tag{2.96}$$

$$= \iint_S \tilde{\mathbf{p}}^\intercal(x, y)\mathbf{A}_1\mathbf{R}\mathbf{y} dS \tag{2.97}$$

$$= \mathbf{r}^\intercal\mathbf{A}_1\mathbf{R}\mathbf{y} \quad , \tag{2.98}$$

where $\mathbf{r}$ is a column vector whose n-th element is (Luger et al., 2019)

$$r_n \equiv \iint_S \tilde{p}_n(x, y)\, \mathrm{d}S \quad . \tag{2.99}$$

When the entire disc is visible (no occultor)

$$r_n = \int_{-1}^{1} \int_{-\sqrt{1-x^2}}^{\sqrt{1+x^2}} \tilde{p}_n(x, y)\mathrm{d}y\mathrm{d}x \quad , \tag{2.100}$$

and the solution to this integral can be expressed in terms of Gamma functions (Equation 20 in Luger et al., 2019). $\mathbf{r}$ and $\mathbf{A}_1$ are independent of the map coefficients $\mathbf{y}$ so they can be pre-computed.

Computing the occultation light curves is more complicated because an occultor of radius r, centred at (x_o, y_o), partially occults the body, and the exposed portion of the disc is a function of (r, x_o, y_o). The general expression for the flux is

$$F = \iint_S I(x, y)\mathrm{d}S \tag{2.101}$$

$$= \mathbf{s}^\intercal\mathbf{A}\mathbf{R}\mathbf{y} \quad , \tag{2.102}$$

where the vector

$$\mathbf{s}^\intercal \equiv \iint_S \tilde{\mathbf{g}}^\intercal(x, y)\mathrm{d}S \tag{2.103}$$

is defined to be the solution to the integral. Luger et al. (2019) solves this integral in two steps. First, we rotate the coordinate system about the z-axis so that the occultor lies along the $+y$ axis and its centre is at a distance $b = \sqrt{x_o^2 + y_o^2}$ from the origin. This substantially simplifies the limits of integration. The second step is to use Green's theorem to express the surface integral of $\tilde{\mathbf{g}}_n$ as a line integral of a vector function $\mathbf{G}_n$ along the boundary of the visible portion of the occulted disc. The n-th element of the solution vector $\mathbf{s}^\intercal$ is

$$s_n = \iint_S \tilde{g}_n(x, y)\mathrm{d}S = \oint \mathbf{G}_n(x, y) \cdot \mathrm{d}\mathbf{r} \quad , \tag{2.104}$$

where $\mathbf{G}_n(x,y) = G_{nx}(x,y)\hat{\mathbf{x}} + G_{ny}(x,y)\hat{\mathbf{y}}$ is constructed such that

$$\mathbf{D} \wedge \mathbf{G}_n = \tilde{g}_n(x,y) \quad , \tag{2.105}$$

were $\mathbf{D} \wedge \mathbf{G}_n$ is the exterior derivative of $\mathbf{G}_n$ given by

$$\mathbf{D} \wedge \mathbf{G}_n \equiv \frac{\mathrm{d}G_{ny}}{\mathrm{d}x} - \frac{\mathrm{d}G_{n_x}}{\mathrm{d}y} \tag{2.106}$$

in Cartesian coordinates. Luger et al. (2019) provide one possible expression for $\mathbf{G}_n$ which satisfies Equation 2.105 and shows that the final solution can be written as

$$s_n = \mathcal{Q}\left(\mathbf{G}_n\right) - \mathcal{P}\left(\mathbf{G}_n\right) \quad , \tag{2.107}$$

where $\mathcal{P}(\mathbf{G}_n)$ is the line integral along the arc of the occultor of radius r and $\mathcal{Q}(\mathbf{G}_n)$ is the line integral along the arc of the occulted body of unit radius. Solutions to these integrals are long, but they are composed of only analytic functions such as sines, cosines and complete elliptic integrals. Importantly, they only need to be evaluated once for an arbitrary map with fixed geometry of an occultation (Luger et al., 2019).

The final expression for the total flux for a particular geometrical arrangement of the occulted body and the occultor can be written succinctly as

$$f = \mathbf{s}^T \mathbf{A} \mathbf{R}' \mathbf{R} \mathbf{y} \quad , \tag{2.108}$$

where $\mathbf{R}'$ is a rotation matrix which rotates the map so that the occultor is placed symmetrically on the $+y$ axis at a distance b. Notice that Equation 2.108 defines a linear operation acting on a vector of spherical harmonic coefficients $\mathbf{y}$. It is trivial to generalise this expression to a case of *spectral map*. In that case we define a matrix $\mathbf{Y}$ whose columns are the spherical harmonic coefficients at different wavelengths and the total flux (spectrum) is then

$$\mathbf{f} = \mathbf{s}^T \mathbf{A} \mathbf{R}' \mathbf{R} \mathbf{Y} \quad , \tag{2.109}$$

where $\mathbf{f}$ is a vector of fluxes, one per wavelength bin.

This formalism can be extended to account for limb darkening, which is important when modelling transits of planets across stars or the light curves of eclipsing binaries. Agol et al. (2020) show how a limb-darkening profile that is an order l polynomial function of $\mu = z = \sqrt{1 - x^2 - y^2}$ can be expressed in terms of the $m = 0$ spherical harmonics up to order l. In the case of quadratic limb-darkening ($l_{\max} = 2$), the limb-darkening profile is

$$\frac{I(\mu)}{I(1)} = 1 - u_1(1-\mu) - u_2(1-\mu)^2 \quad , \tag{2.110}$$

which can be expressed as a sum of spherical harmonics (Equation 38 in Luger et al., 2019). Limb-darkening needs to be treated separately from the map coefficients $\mathbf{y}$ because it does not rotate along with the map when rotations $\mathbf{R}$ and $\mathbf{R}'$ are applied. The limb-darkening coefficients are applied to the map $\mathbf{y}$ as a multiplicative filter following any rotations. This results in another set of spherical harmonic coefficients because products of spherical harmonics are also spherical harmonics. Applying limb-darkening raises the degree of the map by the degree of limb-darkening.

Reflected light

The results derived in the previous section are valid only for isotropic emission from the occulted body (except for the limb darkening which is anisotropic). Equation 2.108 is a good model for stellar light curves, secondary eclipse light curves, and phase curves of planets observed in the mid to far infrared wavelengths where the contribution of scattered starlight to total flux is negligible. Modelling reflected light phase curves and occultations is far more complicated because one has to consider the body's nonuniform illumination and the possible presence of a terminator line (the boundary between day and night). Luger et al. (2022a) extends the formalism described in the previous section to the case of reflected light phase curves and occultations for a body whose spatial *albedo* (specifically the *spherical albedo* – the fraction of power incident on a body at a given wavelength that is scattered into space in all directions) distribution is expanded in the basis of spherical harmonics. Thus, analogous to Equation 2.85 we have

$$A(x, y) = \tilde{\mathbf{y}}^{\intercal}(x, y)\mathbf{y} \quad , \tag{2.111}$$

where A is the albedo. We work under the assumption of isotropic (Lambertian) scattering of light at every point on the body which means that the illumination profile $\mathcal{I}$ of the surface is given by Lambert's law:

$$\mathcal{I}\left(\vartheta_{\mathrm{i}}\right) = \mathcal{I}_0 \max\left(0, \cos\vartheta_{\mathrm{i}}\right) \quad , \tag{2.112}$$

where ϑ_i is the angle between the incident radiation at the surface normal and $\mathcal{I}_0$ is the maximum illumination. In the case of, for example, a planet illuminated by its host star (Appendix A.2 in Luger et al., 2022a)

$$\mathcal{I}_0 = \frac{f_s}{\pi r_{\mathrm{s}}^2} \quad , \tag{2.113}$$

where r_s is the distance between the planet and the star in units of the planet's radius and f_s is the stellar flux measured at the observer in some arbitrary units. We assume that $f_s = 1$ so that all fluxes are defined as the fraction of the flux of the illumination source at the observer. It is the presence of the day/night terminator that complicates the calculation of total flux the most. The limits of integration end up depending on a solution to a quartic equation specifying the points of intersection between the occultor and the day/night terminator line and the solution to those integrals are a function of *incomplete* elliptic integrals as opposed to complete elliptic integrals as was the case for occultations in the emitted light. The full derivation of the total flux is provided by Luger et al. (2022a) and I will only briefly summarise the result here for completeness.

Under the Lambertian scattering assumption, the observed intensity at any point of the body's surface is proportional to the cosine of the angle ϑ_i between the incident light rays and the surface normal. All points for which $\vartheta_i \geq \pi/2$ have intensity of zero (they're unilluminated). Let's assume that the point source is placed at sky coordinates $(x_{\mathrm{s}}, y_{\mathrm{s}}, z_{\mathrm{s}})$ in units of the radius of the illuminated body. The day/night terminator on the body is a half-ellipse with a semi-major axis equal to unity and a (signed) semi-minor axis equal to

$$b = -\frac{z_{\mathrm{s}}}{r_{\mathrm{s}}} \quad , \tag{2.114}$$

where $r_\mathrm{s} = \sqrt{x_\mathrm{s}^2 + y_\mathrm{s}^2 + z_\mathrm{s}^2}$ is the distance to the source. The angle by which the semi-major axis of this ellipse is rotated away from the $+x$-axis is

$$\theta = -\arctan 2\left(x_\mathrm{s}, y_\mathrm{s}\right) \quad . \tag{2.115}$$

Under the assumption that $rs \gg 1$, the illumination $\mathcal{I}$ at point (x, y) on the projected disc is (Luger et al., 2022a)

$$\mathcal{I}\left(b, \theta, r_\mathrm{s}; x, y\right) = \max\left(0, I\left(b, \theta, r_\mathrm{s}; x, y\right)\right) \quad , \tag{2.116}$$

where

$$I\left(b, \theta, r_\mathrm{s}; x, y\right) = \frac{1}{\pi r_\mathrm{s}^2} \cos\vartheta_\mathrm{i} \tag{2.117}$$

$$= \frac{1}{\pi r_\mathrm{s}^2}\left(-b_\mathrm{c} \sin\theta x + b_\mathrm{c} \cos\theta y - b z(x, y)\right) \quad , \tag{2.118}$$

with $b_\mathrm{c} \equiv \sqrt{1 - b^2}$ and $z(x, y) = \sqrt{1 - x^2 - y^2}$. The illumination $\mathcal{I}$ is a dimensionless quantity normalised such that the integral of $A\mathcal{I}$ over the unit disc is equal to the flux measured by the observer as a fraction of the flux of the illumination source. To compute the total flux, we need to weigh each of the terms in Green's basis and integrate them over the visible portion of the body's disc to obtain the reflected light solution vector $\mathbb{s}^\intercal$. Evaluating these integrals is, unfortunately, intractable because of the piecewise function in Equation 2.116. It is more tractable to weigh the basis terms by the function I and modify the limits of integration such that the nightside of the body is excluded. Luger et al. (2022a) show that I can be expressed in the form of a linear operator $\mathbf{I}$ to weight a map vector in the polynomial basis $\tilde{\mathbf{p}}$ by the illumination profile. The total flux is then

$$f = \mathbb{s}^\intercal\left(b, \theta', b_\mathrm{o}, r_\mathrm{o}\right) \mathbf{A_2 I}\left(b, \theta', r_\mathrm{s}\right) \mathbf{A_1 R}'\left(x_\mathrm{o}, y_\mathrm{o}\right) \mathbf{R}(\mathrm{I}, \Lambda, \Theta)\mathbf{y} \quad , \tag{2.119}$$

where

$$\theta' = \arctan 2\left(x_\mathrm{o}, y_\mathrm{o}\right) - \arctan 2\left(x_\mathrm{s}, y_\mathrm{s}\right) \quad , \tag{2.120}$$

is the angle of the terminator in frame $\mathcal{F}'$. In case the illuminated body is not occulted we have

$$f_0 = \mathbb{r}^\intercal(b)\mathbf{I}\left(b, \theta'', r_\mathrm{s}\right) \mathbf{A_1 R}''\left(x_\mathrm{s}, y_\mathrm{s}\right) \mathbf{R}(\mathrm{I}, \Lambda, \Theta)\mathbf{y} \quad , \tag{2.121}$$

where $\theta'' = 0$ is the angle of the terminator in frame $\mathcal{F}''$ by construction. $\mathbf{R}''$ rotates the body through an angle $\arctan 2\left(x_\mathrm{s}, y_\mathrm{s}\right)$ so the semi-major axis of the terminator is aligned with the x'' axis. The solutions to integrals $\mathbb{r}^\intercal(b)$ and $\mathbb{s}^\intercal\left(b, \theta', b_\mathrm{o}, r_\mathrm{o}\right)$ are anything but trivial. They are derived in the appendices of Luger et al. (2022a). The key point is that the model is once again linear and the matrices in Equations 2.120 and 2.121 need to be computed only once for a specific occultation/illumination geometry. Luger et al. (2022a) also derive a solution for the total flux in case the light source is extended (the assumption that $r_s \gg 1$ is not satisfied) and in case the body is not a perfect Lambertian reflector.

2.2.2 The starry code

The method described in the previous section is implemented in the Python package starry[4]. starry is many orders of magnitude faster and more accurate than similar codes which rely on numerical integration. It enables the computation of rotational light curves, planetary transits, secondary eclipse light curves and planet-planet occultations in both emitted and reflected light. Recent extensions to starry, not mentioned in Section 2.2, extended the spherical harmonic formalism to the problem of Doppler imaging (Luger et al., 2021a). There is also added support for computing occultation light curves of oblate bodies (Dholakia et al., 2022). In Chapters 6 and 7 I use starry to explore solutions to the *inverse problem* – how can we obtain an estimate of the map coefficients $\mathbf{y}$ which best explain the observed light curve? starry abstracts the complexities of computing the total flux and allows us to treat this problem as a linear problem of the form

$$f = \mathbf{X}\mathbf{y} \quad , \tag{2.122}$$

where $\mathbf{X}$ is the design matrix computed by starry which depends on the geometry of the occultation or transit event. The inverse problem is highly degenerate and making progress requires imposing constraints on the solution $\mathbf{y}$.

In Chapter 6 I use starry to model occultation light curves of a Solar System object – Jupiter's moon Io, as it is occulted by Jupiter. Thanks to the comparatively close distance to Io relative to objects outside of the Solar System, these light curves are very high quality. I also discuss possible approaches to modelling time-dependent maps ($\mathbf{y} = \mathbf{y}(t)$). In Chapter 7 I tackle a very similar problem but in a very different regime of exoplanet eclipse mapping where the signal-to-noise is orders of magnitude worse than in the case of Io. In both cases, the focus is on on thermal light curves rather than light curves in reflected light.

2.3 Statistical inference – theory

Having covered the basic physics behind gravitational lensing and occultation mapping, I now turn to statistical inference in general, and, more specifically, *Bayesian* statistical inference. Bayesian (as opposed to frequentist) statistics provides a straightforward and elegant theoretical framework for solving inverse problems of the kind we mentioned in Sections 2.1 and 2.2. These are problems where we have little data per object of interest, the data are very noisy, the physical models are relatively well understood but their parameters do not straightforwardly map to observables, and there are multiple competing, sometimes physically quite different models, which provide a good explanation for the data. I am not particularly dogmatic about the use of Bayesian methods and I do occasionally steer away from the classical Bayesian approach and point out where a less Bayesian method is superior in practice. What is most important is the use of *probabilities* for encoding uncertainty about the physical world.

I start by reviewing basic probability theory and briefly mention the frequentist interpretation. I then introduce two key concepts, the likelihood function – a recipe for generating

[4] https://starry.readthedocs.io/en/latest/

plausible datasets given model parameters, and priors – what we know about the model parameters before collecting any data. Next, I discuss statistical computation and fundamental concepts such as the curse of dimensionality and the no-free-lunch-theorem. I introduce different methods for sampling a complex probability distribution and optimizing an objective function. I also discuss model comparison and the difference between statistics and machine learning Finally, I end the section with an overview of automatic differentiation, a key tool for gradient-based sampling and optimization which I use throughout the thesis.

2.3.1 Probability theory

There are two different interpretations of probability. In the *frequentist* interpretation, probabilities are defined as *frequencies of events* that are repeated in a large number of trials (think of repeated coin tosses). An alternative to the frequentist interpretation is the *Bayesian* interpretation, in which probabilities represent *degrees of belief* about the subject of interest. In both cases, probabilities are necessarily positive numbers between zero and one. The significant advantage of the Bayesian interpretation is that it allows one to reason about the probability of one-off events such as "person A wins the election" or "this microlensing event is a binary-lens event". The differences between the two interpretations are not just philosophical. The choice of interpretation one subscribes to can determine how we approach a particular statistical inference problem and the methods we use. Both approaches are commonly used within the physical sciences depending on the field and the problem. For instance, in particle physics frequentist methods are much more common because particle physics experiments often involve billions of repeated experiments in particle accelerators. On the other hand, Bayesian methods are more prevalent in astronomy because in astronomy we cannot do actual repeated experiments for obvious reasons.

The mathematical rules for manipulating probabilities are quite straightforward. In probability theory, we use *random variables*, which can be either discrete or continuous, as opposed to *deterministic variables*. The probability that a given random variable takes any value within its domain is encoded using probability distributions. For discrete random variables, we can talk of probabilities that a given variable assumes a particular value, say 5, but with continuous random variables only the probabilities that a given variable is contained within a particular range of values, say $[0, 1]$, are defined. Now let's consider a continuous *random variable* x with an associated *probability density function* (pdf) $p(x)$[5]. Since x has to take on *some value* within its domain, we have the requirement that

$$\int p(x)\ \mathrm{d}x = 1 \quad , \tag{2.123}$$

where the integral is over the entire domain of x. For Equation 2.123 to hold, $p(x)$ has to have units of x^{-1} so that the integral is a dimensionless number. When dealing with pdfs over physical random variables, it is always important to check that the units are correct.

In general, we are usually dealing with several parameters at once in which case we are interested in *joint probability distributions* over all the parameters. Consider random

[5]The notation $p(x)$ we use is somewhat overloaded. In statistics, the proper notation for a probability density function is $p_X(x)$, where X is a random variable, and x is a particular realisation of that random variable.

variables x_1 and x_2 with a joint probability density function $p(x_1, x_2)$, which, when integrated over a particular region of the (x_1, x_2) plane, gives the probability that both x_1 and x_2 are contained within that region. These joint probabilities can be decomposed into *conditional probabilities* using the so-called product rule:

$$p(x_1, x_2) = p(x_1)\, p(x_2|x_1) \tag{2.124}$$
$$p(x_1, x_2) = p(x_1|x_2)\, p(x_2) \quad , \tag{2.125}$$

where $p(x_2|x_1)$ is a pdf for x_2 *conditional on* x_1 *assuming a certain value.* $p(x_2|x_1)$ is every bit as valid a pdf as $p(x_2)$, it has the same units and has to integrate to 1. Combining Equations 2.124 and 2.125 results in the famous *Bayes's theorem*:

$$p(x_1|x_2) = \frac{p(x_2|x_1)\, p(x_1)}{p(x_2)} \quad . \tag{2.126}$$

Bayes's theorem is simply a rule for converting one kind of a conditional probability into another. Given a pdf $p(x_1, x_2)$, we can *integrate out* or *marginalise* parameters that we are not interested in to obtain a distribution over parameters of interest:

$$p(x_1) = \int p(x_1, x_2) \mathrm{d}x_2 = \int p(x_1|x_2)\, p(x_2) \mathrm{d}x_2 \quad , \tag{2.127}$$

where we applied Bayes's theorem in the second inequality. In cases where the parameter space is high dimensional, these integrals might be impossible to calculate analytically. Still, they can (sometimes) be estimated using sampling algorithms, as we shall see in subsequent sections. If x_1 and x_2 are *statistically independent*, we can factor out $p(x_1, x_2)$ as

$$p(x_1, x_2) = p(x_1)\, p(x_2) \quad . \tag{2.128}$$

Independence is a widespread assumption in statistical modelling. For instance, when modelling light curves of microlensing events or planetary transits, we often assume that flux f_i measured at time t_i is independent of flux f_j measured at time t_j for all measurement times t_i, t_j so that

$$p(\{f\}_{i=1}^{N}) = \prod_{i=1}^{N} p(f_i) \quad . \tag{2.129}$$

If the pdf-s $p(f_i)$ also happen to be equal, we say that the data are independent and identically distributed ("iid"). If the data are independent, but not identically distributed, we say that the uncertainties are *heteroscedastic*. The iid assumption is not that common in astronomy because we tend to have some idea of per-data-point uncertainties.

Expectation values and the mode

The *mean* or the *expected value* of a distribution $p(\mathbf{x})$, usually denoted by μ, is a measure of the "centre" of a distribution. It is defined as

$$\mathbb{E}[x] \equiv \int x\, p(x)\, \mathrm{d}x \quad . \tag{2.130}$$

Similarly, the *variance* is a measure of the "spread" of a distribution:

$$\mathbb{V}[x] \equiv \mathbb{E}\left[(x-\mu)^2\right] = \int (x-\mu)^2 p(x) dx \tag{2.131}$$

$$= \int x^2 p(x)\,\mathrm{d}x + \mu^2 \int p(x)\,\mathrm{d}x - 2\mu \int x p(x)\,\mathrm{d}x = \mathbb{E}\left[x^2\right] - \mu^2 \quad . \tag{2.132}$$

The variance is commonly denoted by σ^2 and the square root of σ^2 is the *standard deviation*. For a *multivariate* random variable with D dimensions then, the variance generalises to a symmetric, positive semi-definite matrix called the *covariance matrix*:

$$\mathrm{Cov}[\mathbf{x}] \equiv \mathbb{E}\left[(\mathbf{x} - \mathbb{E}[\mathbf{x}])(\mathbf{x} - \mathbb{E}[\mathbf{x}])^\intercal\right] \equiv \boldsymbol{\Sigma} \tag{2.133}$$

$$= \begin{pmatrix} \mathbb{V}[x_1] & \mathrm{Cov}[x_1, x_2] & \cdots & \mathrm{Cov}[x_1, x_D] \\ \mathrm{Cov}[x_2, x_1] & \mathbb{V}[x_2] & \cdots & \mathrm{Cov}[x_2, x_D] \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{Cov}[x_D, x_1] & \mathrm{Cov}[x_D, x_2] & \cdots & \mathbb{V}[x_D] \end{pmatrix} \quad . \tag{2.134}$$

In the general case, we can compute expectation values of an arbitrary function $g(\mathbf{x})$ under the probability distribution $p(\mathbf{x})$:

$$\mathbb{E}\left[g(\mathbf{x})\right] = \int g(\mathbf{x})\, p(\mathbf{x})\,\mathrm{d}x\,\mathrm{d}\theta \quad . \tag{2.135}$$

Some examples of g include higher moments of the distribution, the median and quantile estimates. The *mode* of $p(\mathbf{x})$

$$\mathbf{x}^* = \underset{\mathbf{x}}{\mathrm{argmax}}\, p(\mathbf{x}) \quad , \tag{2.136}$$

on the other hand, is not an expectation value. It is the global maximum of the function $p(\mathbf{x})$. Estimating the mode involves computing the *derivatives* of the density $p(\mathbf{x})$, whereas expectation values require us to compute *integrals* over the space $\mathbf{x}$. Computing either the mode or expectation values for nontrivial densities $p(\mathbf{x})$ is a very hard and often impossible problem (more on this in Section 2.4).

Central Limit Theorem

Consider N *independent and identically distributed* random variables with pdf's $p_n(x)$, which are not necessarily Gaussian. Let S_N be the sum of the random variables. The *Central Limit Theorem* (CLT) says that as N increases, the distribution of this sum approaches the Gaussian distribution

$$p\left(S_N = u\right) = \frac{1}{\sqrt{2\pi N\sigma^2}} \exp\left(-\frac{(u - N\mu)^2}{2N\sigma^2}\right) \quad , \tag{2.137}$$

and the distribution of the quantity

$$Z_N \equiv \frac{S_N - N\mu}{\sigma\sqrt{N}} = \frac{\bar{x} - \mu}{\sigma/\sqrt{N}} \quad , \tag{2.138}$$

where $\bar{x}$ is the *sample mean* $\frac{1}{N}\sum_{i=1}^{N} x_i = S_N/N$ converges to the Gaussian distribution with zero mean and unit variance. Because lots of random variables representing real-world processes can be modelled as sums of i.i.d. independent random variables, the Gaussian distribution is all over the place in statistics. One should keep in mind that the assumptions underlying the CLT are not always satisfied and the convergence is more rapid in the bulk of the distribution around the mean than in the tails of the distribution.

Change of variables

Given a univariate random variable x the effect of some deterministic transformation $y(x)$ is to stretch and shift the distribution $p(x)$. For an infinitesimal interval the *probability mass* is the same in both variables so $p(x)\,\mathrm{d}x = p(y)\,\mathrm{d}y$, which yields the *change of variables formula*

$$p_y(y) = p_x(x)\left|\frac{\mathrm{d}x}{\mathrm{d}y}\right| \quad . \tag{2.139}$$

In the multivariate case, if $\mathbf{f}$ is an invertible function mapping $\mathbb{R}^n$ to $\mathbb{R}^n$ with inverse $\mathbf{g}$, the pdf of $\mathbf{y} = \mathbf{f}(\mathbf{x})$ is (Murphy, 2022)

$$p_y(\mathbf{y}) = p_x(\mathbf{g}(\mathbf{y}))\,|\det \mathbf{J}_g(\mathbf{y})| \quad , \tag{2.140}$$

where $\mathbf{J}_g = \frac{\mathrm{d}\mathbf{g}(\mathbf{y})}{\mathrm{d}\mathbf{y}^\intercal}$ is the Jacobian of $\mathbf{g}$ and $\det|\mathbf{J}(\mathbf{y})|$ is the determinant of the Jacobian.

Kullback Leibler divergence

A common way of measuring "similarity" between two distributions $p(x)$ and $q(x)$ is using the *Kullback-Leibler divergence* (KL divergence for short):

$$D_{\mathrm{KL}}(p\|q) = \int_{-\infty}^{\infty} p(x)\ln\left(\frac{p(x)}{q(x)}\right)\mathrm{d}x \quad , \tag{2.141}$$

where $D_{\mathrm{KL}}(p\|q)$ should be read as "the KL divergence from p to q". KL divergence is not a measure of "distance" between the two distributions because it is not a symmetric quantity. It is best understood as a quantity that quantifies *how surprised one would be if one expected to encounter distribution $p(x)$ but instead saw distribution $q(x)$*[6]. This quantity is important in variational inference, which is covered in Section 2.4.

2.3.2 Bayes' theorem in practice

Consider a general inference problem where we collect some data $\mathcal{D}$ and we are interested in the joint probability distribution for parameters $\boldsymbol{\theta}$ of a particular model. We can rewrite Bayes' theorem as

$$p(\boldsymbol{\theta}|\mathcal{D}) = \frac{p(\boldsymbol{\theta})p(\mathcal{D}|\boldsymbol{\theta})}{p(\mathcal{D})} = \frac{p(\boldsymbol{\theta})p(\mathcal{D}|\boldsymbol{\theta})}{\int p\left(\boldsymbol{\theta}'\right)p\left(\mathcal{D}|\boldsymbol{\theta}'\right)d\boldsymbol{\theta}'} \quad . \tag{2.142}$$

[6] https://wiki.santafe.edu/images/a/a8/IT-for-Intelligent-People-DeDeo.pdf

In Equation 2.142 the term $p(\mathcal{D}|\boldsymbol{\theta})$ is a probability distribution over the data $\mathcal{D}$ given the parameters of the model $\boldsymbol{\theta}$. When this term is seen as a function of the parameters given a fixed dataset, it is called the *likelihood function* or the *likelihood* for short. $p(\boldsymbol{\theta})$ is a so-called *prior* distribution over $\boldsymbol{\theta}$; it represents our beliefs about $\boldsymbol{\theta}$ before having observed the data $\mathcal{D}$. When the prior distribution is multiplied by the likelihood, we obtain, up to a normalising constant $p(\mathcal{D})$, the posterior distribution over $\boldsymbol{\theta}$ given the data $\mathcal{D}$.

The best way to think about Equation 2.142 is to see it as a rule for updating beliefs. Our prior beliefs about the parameters of the model $p(\text{parameters})$ should be updated to posterior beliefs $p(\text{parameters}|\text{data})$ by making use of the likelihood $p(\text{data}|\text{parameters})$. If the data are useful, the likelihood is generally much narrower than the prior pdf, and the choice of priors does not significantly influence the final results.

The likelihood

Let's cover the three terms in Equation 2.142 in more detail, starting with the most important one – the likelihood. The likelihood can be seen as a *generative model* for the data $\mathcal{D}$ (a recipe for generating plausible datasets) given a specific realisation of the parameters $\boldsymbol{\theta}$. For the kinds of models we work with in astronomy, the likelihood consists of two parts. The first part is a (usually deterministic) function which describes the physical system of interest, for instance, the predicted flux for a binary microlensing event with a particular trajectory. The second part is the *noise model* which describes the statistical properties of the noise inherent in any physical measurement. We often assume that the noise is independent and Gaussian. It is very common to use the likelihood as is, and optimize it with respect to the model parameters $\boldsymbol{\theta}$ in a procedure called *maximum likelihood estimation* (MLE) to find a single point in the parameter space $\boldsymbol{\theta}^\star$ which maximizes the likelihood (more on this in Section 2.3.3).

We can also marginalise the likelihood over a subset of parameters. Let's say that we have a likelihood function $p(\boldsymbol{\theta}, \boldsymbol{\alpha}|\mathcal{D})$ where $\boldsymbol{\alpha}$ is a set of *nuisance parameters* which we are not particularly interested in. To marginalise over these nuisance parameters we need to introduce a prior[7] $p(\boldsymbol{\alpha}|\boldsymbol{\theta})$ which may potentially also depend on the $\boldsymbol{\theta}$. The marginalised likelihood is then

$$p(\mathcal{D}|\boldsymbol{\theta}) = \int p(\mathcal{D}|\boldsymbol{\theta}, \boldsymbol{\alpha})\, p(\boldsymbol{\alpha}|\boldsymbol{\theta})\, \mathrm{d}\boldsymbol{\alpha} \quad . \tag{2.143}$$

The problem structure where $\boldsymbol{\alpha}$ is some set of parameters we don't particularly care about, and $\boldsymbol{\theta}$ are the physical parameters of interest, is widespread in astronomy. For instance, in single lens gravitational microlensing, estimating the distribution $p(t_E)$ over the parameter of interest t_E requires marginalising over parameters such as F_S, F_B, t_0 and u_0.

Not all statistical models can be written in terms of the likelihood function. For example, most machine learning models do not have a simple probabilistic interpretation[8]. If we can write down the likelihood function for a certain problem, we say that we have a *generative model for the data* because the likelihood can be used to generate mock data sets.

[7] Not introducing a prior leads to a dimensionally incorrect result.

[8] Indeed, finding probabilistic interpretations of complex machine learning models such as deep neural networks is an active research area.

Priors

The second term of interest is the prior $p(\boldsymbol{\theta})$. In principle, the prior distribution should capture all of the prior information we might have about the model parameters, prior to collecting new data. In practice, we do not want to bias our results, so we often use "weakly informative" priors – priors which are informative enough that they exclude unreasonable parameter values, but are not so narrow that they rule out sensible values which may be favoured by the likelihood. For example, we might have prior information such as "the mass of the planet should be positive" in which case a suitable weakly informative prior on the mass parameter should, first of all, exclude negative values, but also potentially exclude values which are nonsensically large for the problem at hand. There are also the so-called "non-informative priors", such as Jeffrey's prior and maximum entropy priors. Although these priors can be of use in some cases, in general, *there is no such thing as a truly non informative prior*. Even if the prior for a particular parameter θ is uninformative in one set of coordinates, it is often "informative" for some certain transformation of the parameter $g(\theta)$. For instance, a uniform prior in θ is not uniform in $\ln\theta$. Having said that, priors are important and it is often worth the effort to construct priors which respect certain symmetries of the the problem that we know to be true.

Priors should not be seen in isolation, rather, they should be seen in the context of the likelihood (Gelman et al., 2017). To see that this is the case consider drawing samples from the prior $\boldsymbol{\theta} \sim p(\boldsymbol{\theta})$, plugging those values into the likelihood distribution $p(\mathcal{D}|\boldsymbol{\theta})$ and then generating mock datasets, a process called *prior predictive checking*. This process makes it easy to spot priors that in combination with the likelihood yield predictions of what the data should like which do not match our expectations. An in-depth study on choosing priors is available here[9].

Depending on how informative is the data, we differentiate between two different regimes. We say that the inference process is *data driven* if the data constrain the model parameters so well that the choice of the priors is effectively irrelevant. The other regime is said to be *prior driven* – when the data are so weakly informative that the shape of the posterior pdf is similar to the prior pdf. In general, in any statistical analysis involving priors, one should check the sensitivity of the scientific conclusion to assumptions about different priors. No statistical inference procedure is free of assumptions but the use of priors and likelihoods makes those assumptions more explicit.

Priors are sometimes the goal of the inference. For example, if the prior on the prior pdf $p(\boldsymbol{\theta})$ itself depends on some other set of parameters $\boldsymbol{\beta}$, then we can marginalise out all of the $\boldsymbol{\theta}$ parameters so we are left with a likelihood for the parameters $\boldsymbol{\beta}$:

$$p(\mathcal{D}|\boldsymbol{\beta}) = \int p(\mathcal{D}|\boldsymbol{\theta})\, p(\boldsymbol{\theta}|\boldsymbol{\beta})\, \mathrm{d}\boldsymbol{\theta} \quad . \tag{2.144}$$

This can then be used, using Bayes's theorem, to calculate the posterior over the prior parameters $\boldsymbol{\beta}$.

$$p(\boldsymbol{\beta}|\mathcal{D}) \propto p(\mathcal{D}|\boldsymbol{\beta})\, p(\boldsymbol{\beta}) \quad . \tag{2.145}$$

The parameters $\boldsymbol{\beta}$ which specify the prior are usually called *hyperparameters* or *population level parameters*. Similarly, the priors for those parameters are called *hyperpriors*. The whole

[9] https://betanalpha.github.io/assets/case_studies/prior_modelling.html

process is called *hierarchical Bayesian inference* (in astronomy literature), or *multilevel modelling, hierarchical modelling, nested modelling, mixed models or random effects models* (in statistics and adjacent fields). Hierarchical modelling is a very useful method for estimating population-wide properties given data about individual objects. For example, in the context of microlensing exoplanet models, the individual parameters could be the masses of individual planets, while the hyperparameters would describe the population and the shape of the planetary mass function.

The evidence

The final component of Equation 2.142 is the normalisation constant $p(\mathcal{D})$ in the denominator,

$$\mathcal{Z} \equiv p(\mathcal{D}) = \int p(\mathcal{D}|\boldsymbol{\theta})\, p(\boldsymbol{\theta}) \mathrm{d}\boldsymbol{\theta} \quad , \tag{2.146}$$

which is called the *fully marginalised likelihood* because it is a likelihood marginalised over all of the parameters of the model. Sometimes it is also called the *Bayesian evidence*. $\mathcal{Z}$ is generally very difficult to compute because it is an integral over a potentially very high dimensional parameter space $\boldsymbol{\theta}$ with complex structure. Fortunately, in most cases, we can obtain samples from the posterior distribution without having to calculate $\mathcal{Z}$.

2.3.3 Maximum likelihood estimation

One of the most popular inference methods is called *maximum likelihood estimation* (MLE). Maximum likelihood estimation answers the question: *what choice of parameters* $\mathbf{x}$ *maximizes the probability of observing the data* $\mathcal{D}$*?*, as opposed to the question, *what are the most likely values of the parameters* $\mathbf{x}$ *having observed the data* $\mathcal{D}$*?*[10], which necessitates the multiplication of the likelihood with a prior. Since maximum likelihood estimation does not use the prior, it is not a Bayesian method. The output of MLE is a *point estimate* of the parameters, $\hat{\boldsymbol{\theta}}$[11]. The MLE estimate is obtained by maximising the logarithm[12] of the likelihood $p(\mathcal{D}|\boldsymbol{\theta})$ (or equivalently, minimising the negative log-likelihood) with respect to the parameters $\boldsymbol{\theta}$:

$$\hat{\boldsymbol{\theta}}_{\mathrm{mle}} = \underset{\boldsymbol{\theta}}{\mathrm{argmax}} \ln p\left(\mathcal{D}|\boldsymbol{\theta}\right) \quad . \tag{2.147}$$

MLE for a multivariate Gaussian

The log-likelihood of multivariate Gaussian distribution without the irrelevant constants is

$$LL(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \ln p(\mathcal{D}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{N}{2} \ln |\boldsymbol{\Sigma}^{-1}| - \frac{1}{2} \sum_{n=1}^{N} \left(\boldsymbol{y}_n - \boldsymbol{\mu}\right)^{\intercal} \boldsymbol{\Sigma}^{-1} \left(\boldsymbol{y}_n - \boldsymbol{\mu}\right) \quad , \tag{2.148}$$

[10] The distinction between the two questions is subtle but important. The parameters which maximize the likelihood might be very different from those which are most "likely", in the sense that they are the most probable given the data.

[11] From a Bayesian viewpoint, the MLE is equivalent to MAP estimation under a uniform prior $p(\boldsymbol{\theta})$

[12] Because probabilities in most regions of the parameter space are very small numbers, we always use the log transform when optimising likelihoods or posteriors.

where $\mathbf{y}_n$ are the data points, and $\mathbf{\Sigma}^{-1}$ is the inverse covariance matrix, also knowns as the *precision matrix*. The MLE for the mean parameter $\hat{\boldsymbol{\mu}}$ is given by the sample mean (empirical mean) of the data (Murphy, 2022):

$$\hat{\boldsymbol{\mu}} = \frac{1}{N}\sum_{n=1}^{N} \boldsymbol{y}_n = \overline{\boldsymbol{y}} \quad . \quad . \tag{2.149}$$

Similarly, the MLE of the covariance matrix $\hat{\mathbf{\Sigma}}$ is equal to the empirical covariance matrix

$$\hat{\mathbf{\Sigma}} = \frac{1}{N}\sum_{n=1}^{N} (\boldsymbol{y}_n - \overline{\boldsymbol{y}})\,(\boldsymbol{y}_n - \overline{\boldsymbol{y}})^\intercal \quad . \tag{2.150}$$

It is a biased estimator of the true covariance matrix. Thus, whenever we compute an empirical mean and the covariance for a given dataset, we are implicitly assuming that the data are normally distributed and obtaining an MLE estimate of that distribution.

2.3.4 Least squares linear regression

I first introduce the classical, non-Bayesian formulation of linear regression (also known as *least squares regression*) before considering the Bayesian formulation. In a (multiple) linear regression model we have (noisy) measurements $\mathbf{y} \in \mathbb{R}^n$ (*dependent variable*), an $n \times p$ matrix $\mathbf{M} \in \mathbb{R}^{n\times p}$ of inputs (*design matrix* consisting of *independent variables*, *covariates*, or *features*) and a vector of *regression coefficients* or *weights* $\boldsymbol{\beta} \in \mathbb{R}^p$. This is a very general model because the matrix $\mathbf{M}$ can be composed of basis functions such as polynomials, spherical harmonics, Fourier modes, wavelets etc., so although the model is linear in the parameters (regression coefficients), it is not generally linear in the raw input data.

If we assume that the observations $\mathbf{y}$ have Gaussian noise we can write down the likelihood of this model as

$$p(\mathbf{y}|\boldsymbol{\beta}) = \mathcal{N}(\mathbf{y}|\mathbf{M}\boldsymbol{\beta}, \mathbf{C}) \quad , \tag{2.151}$$

where $\mathbf{C}$ is the data covariance matrix and $\mathcal{N}$ is shorthand notation for the multivariate normal distribution. In an astronomical application, it is usually assumed to be diagonal and we might have some idea for the scale of the of the diagonal elements of the matrix (the "error bars"). The negative log-likelihood (NLL) is then equal to (ignoring constants)

$$\mathrm{NLL}(\boldsymbol{\beta}) = \frac{1}{2}(\mathbf{M}\boldsymbol{\beta} - \mathbf{y})^\intercal \mathbf{C}^{-1}(\mathbf{M}\boldsymbol{\beta} - \mathbf{y}) \quad , \tag{2.152}$$

which is just the well-known least squares objective function, or χ^2, as it is known in astronomy.

Underdetermined case ($p < n$)

For now, let's assume that the inverse data covariance matrix $\mathbf{C}^{-1}$ is a diagonal matrix with the same variance σ^2 for every data point. In that case, the MLE solution for the weights $\boldsymbol{\beta}$ is the solution to the following (convex) optimisation problem:

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\mathrm{argmin}}\, |\mathbf{y} - \mathbf{M}\boldsymbol{\beta}|^2 \quad . \tag{2.153}$$

There is a closed-form solution for Equation 2.153 as long as the number of parameters and features p is less than the number of data points n. The solution is (Hogg and Villar, 2021a):

$$\hat{\boldsymbol{\beta}} = (\mathbf{M}^\intercal \mathbf{M})^{-1} \mathbf{M}^\intercal \mathbf{y} \quad , \tag{2.154}$$

under the assumption that $\mathbf{M}^\intercal \mathbf{M}$ is invertible which is usually the case. Equation 2.154 is called the *ordinary least squares* (OLS) solution. For a nontrivial data covariance matrix $\mathbf{C}$ Equation 2.154 generalises to

$$\hat{\boldsymbol{\beta}} = (\mathbf{M}^\intercal \mathbf{C}^{-1} \mathbf{M})^{-1} \mathbf{M}^\intercal \mathbf{C}^{-1} \mathbf{y} \tag{2.155}$$

which is the solution to the *weighted least squares* (WLS) problem. It is common to modify the least squares objective functions with the addition of a *regularisation* term to prevent overfitting. For instance, we can modify the OLS objective (Equation 2.153) by adding an *L2 regularisation* term:

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left(|\mathbf{y} - \mathbf{M}\boldsymbol{\beta}|^2 + \lambda \, |\boldsymbol{\beta}|^2 \right) \quad , \tag{2.156}$$

where $\lambda > 0$ is a regularisation parameter which penalises large magnitudes of the coefficients $\boldsymbol{\beta}$. This is also a convex optimisation problem and the solution is

$$\hat{\boldsymbol{\beta}} = (\mathbf{M}^\intercal \mathbf{M} + \lambda \mathbf{I})^{-1} \mathbf{M}^\intercal \mathbf{y} \quad , \tag{2.157}$$

where $\mathbf{I}$ is the identity matrix. Regularized OLS regression with this particular regularisation term is also known as *ridge regression* in the statistics literature[13] L2 regularisation (as well as most other forms of regularisation terms) can be seen as a particular choice of priors for the regression coefficients in the Bayesian paradigm. The above also naturally generalise to the case of weighted least squares.

Overdetermined case ($p > n$)

Contrary to popular belief within the astronomy community it is possible and often desirable to have a model with more parameters than data points (see Chapter 6 for a real-world example). In the over parametrized case of linear regression ($p > n$) there are multiple choices for $\boldsymbol{\beta}$ which perfectly fits the data. The solution, in that case, is defined to be a minimum-norm parameter vector $\boldsymbol{\beta}$ that interpolates the data ($\mathbf{y} = \mathbf{M}\boldsymbol{\beta}$), out of possibly many degenerate solutions. The OLS objective in Equation 2.153 can be modified such that it is valid for both the $p < n$ and the $p > n$ case by making use of a limit in which an L2 regularisation term tends to zero (Hogg and Villar, 2021a):

$$\hat{\boldsymbol{\beta}} = \lim_{\lambda \to 0^+} \left[\underset{\boldsymbol{\beta}}{\operatorname{argmin}} \, |\mathbf{y} - \mathbf{M}\boldsymbol{\beta}|^2 + \lambda \, |\boldsymbol{\beta}|^2 \right] \quad . \ . \tag{2.158}$$

In this case, the OLS solution becomes

$$\hat{\boldsymbol{\beta}} = \mathbf{M}^\intercal (\mathbf{M}\mathbf{M}^\intercal)^{-1} \mathbf{y} \quad . \tag{2.159}$$

[13] One of the many things that make statistics confusing is that very similar methods often have special names. The majority of statistical methods can be seen as a linear model with the data represented in a particular basis, and with a particular choice of the likelihood (noise model), priors (regularisation), and a method for obtaining the solution (usually a convex optimisation problem).

Equations 2.154 and 2.159 can be unified into a single equation by making use of the *Moore-Penrose pseudoinverse* $\mathbf{M}^\dagger$. The pseudoinverse is defined by taking the singular-value decomposition (SVD) of $\mathbf{M}$:

$$\mathbf{M} \equiv \mathbf{USV} \quad . \tag{2.160}$$

Then $\mathbf{M}^\dagger = \mathbf{V}^\intercal \mathbf{S}^\dagger \mathbf{U}^\intercal$.

2.3.5 Bayesian linear regression

In the Bayesian paradigm, we can write down the posterior over the regression weights $\boldsymbol{\beta}$ assuming a Gaussian likelihood and a Gaussian prior over the weights. The posterior is

$$p(\boldsymbol{\beta}|\mathbf{y}) = \frac{p(\mathbf{y}|\boldsymbol{\beta})\, p(\boldsymbol{\beta})}{\int p(\mathbf{y}|\boldsymbol{\beta}) p(\boldsymbol{\beta})\, \mathrm{d}\boldsymbol{\beta}} \tag{2.161}$$

$$= \frac{\mathcal{N}(\mathbf{y}|\mathbf{M}\boldsymbol{\beta}, \mathbf{C})\, \mathcal{N}(\boldsymbol{\beta}|\boldsymbol{\mu}, \boldsymbol{\Lambda})}{\int \mathcal{N}(\mathbf{y}|\mathbf{M}\boldsymbol{\beta}, \mathbf{C})\, \mathcal{N}(\boldsymbol{\beta}|\boldsymbol{\mu}, \boldsymbol{\Lambda})\, \mathrm{d}\boldsymbol{\beta}} \quad , \tag{2.162}$$

where $\mathbf{C}$ is the data covariance matrix as before, $\boldsymbol{\mu}$ is the mean of the Gaussian prior on the regression coefficients, and $\boldsymbol{\Lambda}$ is the prior covariance matrix. Using the fact that the product of two multivariate Gaussian distributions is also a multivariate Gaussian distribution, one can show that the solution to Equation 2.162 is (see for example Hogg et al., 2020)

$$p(\boldsymbol{\beta}|\mathbf{y}) = \mathcal{N}(\boldsymbol{\beta}|\mathbf{a}, \mathbf{A}) \quad , \tag{2.163}$$

where

$$\mathbf{A}^{-1} = \boldsymbol{\Lambda}^{-1} + \mathbf{M}^\intercal \mathbf{C}^{-1} \mathbf{M} \tag{2.164}$$

$$\mathbf{a} = \mathbf{A}\left(\boldsymbol{\Lambda}^{-1}\boldsymbol{\mu} + \mathbf{M}^\intercal \mathbf{C}^{-1} \boldsymbol{y}\right) \quad . \tag{2.165}$$

Since all starry models for occultation light curves are linear (Equation 6.4) we can use Equation 2.163 to directly obtain the posterior over the map coefficients $\mathbf{y}$ but the cost of doing so is imposing a Gaussian prior on the map coefficients.

2.3.6 Marginalising a likelihood over linear parameters

A model structure which is quite common in astronomy and physics is a model which is linear in a subset of all parameters but non-linear in the remaining parameters. This is the structure of the equation for the observed flux in microlensing (Equations 2.25 or 2.27) where the flux parameters are linear and the magnification parameters are non-linear. More generally, the likelihood for such models can be written as

$$p(\mathbf{y}|\boldsymbol{\beta}, \boldsymbol{\theta}) \quad , \tag{2.166}$$

where $\boldsymbol{\beta}$ are the linear parameters and $\boldsymbol{\theta}$ are the non-linear parameters. In some circumstances, it is possible to analytically marginalise the linear parameters $\boldsymbol{\beta}$. Doing so is often

advantageous because it reduces the size of the parameter space and the computational cost of doing inference. The marginalised likelihood is given by (Equation 2.127):

$$p(\mathbf{y}|\boldsymbol{\theta}) = \int p(\mathbf{y}|\boldsymbol{\beta}, \boldsymbol{\theta}) p(\boldsymbol{\beta}|\boldsymbol{\theta}) \, \mathrm{d}\boldsymbol{\beta} \quad . \tag{2.167}$$

Marginalisation requires the introduction of a prior $p(\boldsymbol{\beta}|\boldsymbol{\theta})$ which can potentially also depend on the non-linear parameters $\boldsymbol{\theta}$. The alternative to computing this integral is re-factorising the expression in the integrand using Bayes' theorem:

$$p(\mathbf{y}|\boldsymbol{\beta}, \boldsymbol{\theta}) p(\boldsymbol{\beta}|\boldsymbol{\theta}) = p(\boldsymbol{\beta}|\mathbf{y}, \boldsymbol{\theta}) \, p(\mathbf{y}|\boldsymbol{\theta}) \quad , \tag{2.168}$$

which yields the marginalised likelihood without the need to compute the integral:

$$p(\mathbf{y}|\boldsymbol{\theta}) = \frac{p(\mathbf{y}|\boldsymbol{\beta}, \boldsymbol{\theta}) p(\boldsymbol{\beta}|\boldsymbol{\theta})}{p(\boldsymbol{\beta}|\mathbf{y}, \boldsymbol{\theta})} \quad . \tag{2.169}$$

If both the prior and the likelihood are multivariate Gaussians with the following form:

$$p(\mathbf{y}|\boldsymbol{\beta}, \boldsymbol{\theta}) = \mathcal{N}(\mathbf{y}|\mathbf{M}\boldsymbol{\beta}, \mathbf{C}) \tag{2.170}$$

$$p(\boldsymbol{\beta}|\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\beta}|\boldsymbol{\mu}, \boldsymbol{\Lambda}) \quad , \tag{2.171}$$

Equation 2.169 then becomes (for the proof see Hogg et al., 2020)

$$\mathcal{N}(\mathbf{y}|\mathbf{M}\boldsymbol{\beta}, \mathbf{C}) \, \mathcal{N}(\boldsymbol{\beta}|\boldsymbol{\mu}, \boldsymbol{\Lambda}) = \mathcal{N}(\boldsymbol{\beta}|\mathbf{a}, \mathbf{A}) \, \mathcal{N}(\mathbf{y}|\mathbf{b}, \mathbf{B}) \quad , \tag{2.172}$$

where

$$\mathbf{A}^{-1} = \boldsymbol{\Lambda}^{-1} + \mathbf{M}^\intercal \mathbf{C}^{-1} \mathbf{M} \tag{2.173}$$

$$\mathbf{a} = \mathbf{A} \left(\boldsymbol{\Lambda}^{-1} \boldsymbol{\mu} + \mathbf{M}^\intercal \mathbf{C}^{-1} \boldsymbol{y} \right) \tag{2.174}$$

$$\mathbf{B} = \mathbf{C} + \mathbf{M} \boldsymbol{\Lambda} \mathbf{M}^\intercal \tag{2.175}$$

$$\mathbf{b} = \mathbf{M} \boldsymbol{\mu} \quad . \tag{2.176}$$

Thus the marginalised likelihood is

$$p(\mathbf{y}|\boldsymbol{\theta}) = \mathcal{N}(\mathbf{y}|\mathbf{b}, \mathbf{B}) \quad . \tag{2.177}$$

Application to microlensing

Let's apply this trick to the microlensing models (Equation 2.27). In the general case, the dataset consists of J flux vectors $\mathbf{f}_j$, each with length N_j, where j indexes different spectral bands and/or instruments. Equation 2.27 can then be written in vector form as

$$\mathbf{f}_j = \mathbf{M}_j \boldsymbol{\beta}_j \quad , \tag{2.178}$$

where the matrix $\mathbf{M}$ is defined as

$$\mathbf{M}_j \equiv \begin{pmatrix} \tilde{A}(t_1; \boldsymbol{\theta}) & 1 \\ \tilde{A}(t_2; \boldsymbol{\theta}) & 1 \\ \vdots & \vdots \\ \tilde{A}(t_{N_j}; \boldsymbol{\theta}) & 1 \end{pmatrix} \quad , \tag{2.179}$$

and $t = t_1, \ldots, t_{N_j}$ are the times of observations in band j, $\tilde{A}(t_i) \equiv (A(t_i) - 1)/(A(t_0) - 1)$, $\boldsymbol{\theta}$ are the non-linear parameters, and the weights vector $\boldsymbol{\beta}_j$ is defined as

$$\boldsymbol{\beta}_j \equiv (\Delta F_j \; F_{\text{base},j})^\intercal \quad . \tag{2.180}$$

Equation 2.178 is general enough to encompass either single or multiple-lens microlensing models. Assuming independence of noise properties between observations in different spectral bands[14], the likelihood for the complete dataset $\mathcal{D} = \{f_j\}_{j=1}^J$ factorizes as

$$p(\mathcal{D}|\boldsymbol{\beta}, \boldsymbol{\theta}) = \prod_{j=1}^{J} p(\mathbf{f}_j|\boldsymbol{\beta}_j, \boldsymbol{\theta}) \quad . \tag{2.181}$$

Using Equation 2.177 we can then write down the negative log-likelihood (ignoring constants) for the j-th band as

$$\text{NLL}(\boldsymbol{\theta}) = \frac{1}{2} (\mathbf{f}_j - \mathbf{M}_j\boldsymbol{\mu})^\intercal \left(\mathbf{C}_j + \mathbf{M}_j\boldsymbol{\Lambda}\mathbf{M}_j^\intercal\right)^{-1} (\mathbf{f}_j - \mathbf{M}_j\boldsymbol{\mu}) \tag{2.182}$$

$$+ \frac{1}{2} \ln \left|\mathbf{C}_j + \mathbf{M}_j\boldsymbol{\Lambda}\mathbf{M}_j^\intercal\right| \quad , \tag{2.183}$$

where we have assumed a common mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Lambda}$ for the prior for the linear flux parameters. To avoid computing an inverse and a determinant of J matrices with shape $N_j \times N_j$, we can simplify the terms in Equation 2.183 using the matrix inversion lemma (see for example Appendix A3 in Rasmussen and Williams, 2006)

$$\left(\mathbf{C}_j + \mathbf{M}_j\boldsymbol{\Lambda}\mathbf{M}_j^\intercal\right)^{-1} = \mathbf{C}_j^{-1} - \mathbf{C}_j^{-1}\mathbf{M}_j \left(\boldsymbol{\Lambda}^{-1} + \mathbf{M}_j^\intercal\mathbf{C}^{-1}\mathbf{M_j}\right)^{-1} \mathbf{M}_j^\intercal\mathbf{C}_j^{-1} \tag{2.184}$$

$$\ln \left|\mathbf{C}_j + \mathbf{M}_j\boldsymbol{\Lambda}\mathbf{M}_j^\intercal\right| = \ln |\mathbf{C}_j| + \ln |\boldsymbol{\Lambda}| + \ln \left|\boldsymbol{\Lambda}^{-1} + \mathbf{M}_j^\intercal\mathbf{C}_j^{-1}\mathbf{M}_j\right| \quad . \tag{2.185}$$

In most cases we can assume that $\boldsymbol{\mu} = \mathbf{0}$, $\boldsymbol{\Lambda}$ is diagonal, and in absence of correlated noise all matrices $\mathbf{C}_j$ are also diagonal so the expression in Equation 2.183 is not usually expensive to compute.

Although Equation 2.183 is the formally correct way to marginalise the linear parameters of a general microlensing model, it is not used in the microlensing literature. What most people do instead is to use MLE to maximize the joint likelihood $p(\mathbf{f}|\boldsymbol{\beta}, \boldsymbol{\theta})$ with respect to the linear parameters, by solving a least squares problem at every MCMC step in the non-linear parameters (I will cover MCMC in the next section). This procedure is approximately equivalent to using the marginalised likelihood from Equation 2.183 with two important caveats. The first is that one should not optimize for the maximum likelihood flux parameters conditional on non-linear parameters, but rather the MAP values of those parameters under Gaussian priors (although, the difference between the MLE and MAP estimates is negligible under broad priors). The second caveat is that the optimisation can be cast as a linear least squares problem under the assumption that the covariance matrix $\mathbf{C}$ for the data are

[14] This is certainly true if we have observations from different telescopes. It is also not strictly true if the observations were obtained using the same telescope with different spectral filters, but it is a good approximation.

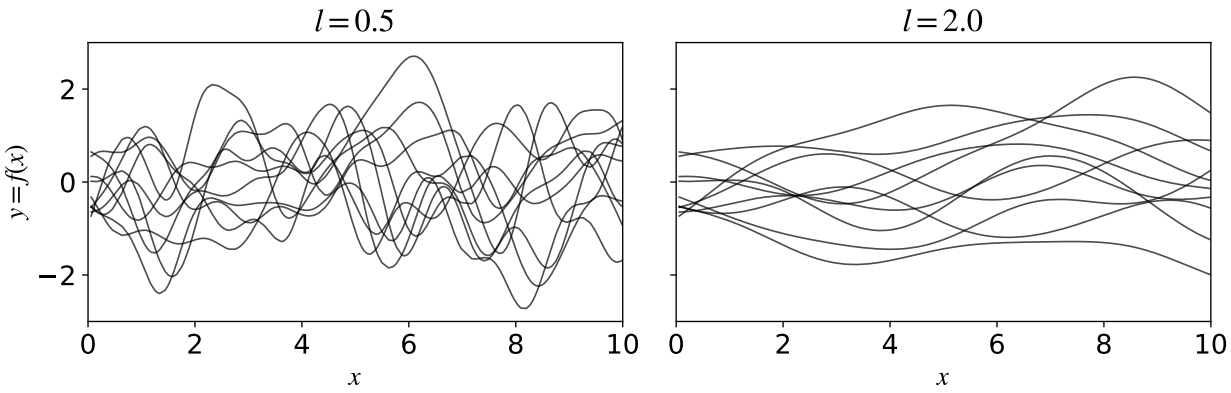


Figure 2.12: A few samples from a Gaussian process prior with a squared exponential kernel function for two different values of the length scale parameter l.

diagonal. This is a fairly restrictive assumption because it implies that there is no correlated noise in the light curve which isn't true for most real-world microlensing datasets.

Although we have marginalised the likelihood over the linear parameters, which substantially reduces the complexity of the inference problem, we can still compute the posterior distribution for those parameters *conditional on the value of the non-linear parameters*:

$$p(\boldsymbol{\beta}|\mathbf{y}, \boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\beta}|\mathbf{y}, \boldsymbol{\theta}) \quad , \tag{2.186}$$

We can use this to obtain the full posterior over the linear parameters if we happen to have samples from the posterior over the non-linear parameters $p(\boldsymbol{\theta}|\mathbf{y})$ (obtained, for example, using MCMC).

2.3.7 Gaussian processes

A final topic that we need to cover in this section is that of *Gaussian Processes*. In Section 2.3.4 I have discussed linear regression with p components (basis functions) and I have mentioned that one can fit many more components than there are data points, and that these models can be quite useful. A Gaussian process (GP) is what happens when let the number of components p go to infinity[15]. This is why GPs belong to a class of methods called *non-parametric regression* methods, although a better name would perhaps be infinite-parameter regression methods. GPs can also be seen as *prior distributions over the space of functions* such that each sample from that GP prior is a function. More formally, *a Gaussian process is a collection of random variables, any finite number of which have a joint multivariate Gaussian distribution* (Rasmussen and Williams, 2006).

GPs are completely specified by a *mean function* $\mu(x)$ and a positive-definite *covariance function* $k(x, x')$ which are functions of the input locations $x \in \mathbb{R}^D$ (for example, a time coordinate):

$$\mu(x) = \mathbb{E}[f(x)] \tag{2.187}$$

$$k(x, x') = \mathbb{E}\left[(f(x) - \mu(x))(f(x') - \mu(x'))\right] \quad , \tag{2.188}$$

[15] See Hogg and Villar (2021a) for a demonstration of this point.

where x and x' are any two different points in the input space. We write

$$f(x) \sim \mathcal{GP}\left(\mu(x), k\left(x, x'\right)\right) \quad . \tag{2.189}$$

Usually the mean function $\mu(x)$ is taken to be zero for simplicity and all the magic happens in the covariance function $k(x, x')$ which describes *how any two points in the input space relate to each other*.

One very popular choice for the kernel function is the *squared exponential* kernel:

$$k(x, x') = \exp\left(-\frac{1}{2}\frac{|x - x'|^2}{l^2}\right) \quad . \quad . \tag{2.190}$$

The above choice of a covariance function implies that the covariance of the outputs $f(x)$ evaluated at any two points will decay exponentially with the square of the distance between the two points. In other words, function values evaluated at points close to each tend to go hand in hand while function values evaluated at points far away from each other are practically independent. The parameters l set the length scale that determines how quickly the covariance decays to zero with increasing distance between the two points. Figure 2.12 shows samples from a GP prior with a squared exponential kernel function in the one-dimensional case for two choices of the length scale parameter l. One special property of this kernel is that it is *stationary*, meaning that the mean and the variance of the samples from the GP prior do not depend on the input coordinate x, they are the same everywhere because the covariance only depends on the difference between any two points x and x'[16]. Not all kernels are stationary. An example of a non-stationary kernel would be a kernel similar to the squared exponential kernel but with a variance which increases linearly in x. Kernels can be combined in any number of ways through, for example, addition and multiplication and the result is also a valid kernel. This[17] page shows informative visualisations of different kernels and their combinations.

The key property of a GP is that for any finite subset $\mathbf{X} = \{\mathbf{x}_1 \dots \mathbf{x}_n\}$ of the domain of x (that is, any marginal distribution of x) is a multivariate Gaussian:

$$f(\mathbf{x}) \sim \mathcal{N}\left(\boldsymbol{\mu}(\mathbf{X}), \mathbf{K}(\mathbf{X}, \mathbf{X})\right) \quad , \tag{2.191}$$

where $\mathbf{K}$ is a covariance matrix of shape $n \times n$ which is constructed by evaluating a kernel such as the one defined in Equation 2.190 elementwise. This is precisely how I plotted the functions in Figure 2.12; I simply chose a dense number of linearly spaced grid points, computed the covariance matrix K (assuming a squared exponential kernel), and finally sampled the multivariate Gaussian distribution.

Predictions from posterior distribution

We can go a step beyond drawing samples from the GP prior and condition the GP on a set of observations to fit the GP to data. Let $\mathbf{f}(\mathbf{X})$ be the vector of observed data points

[16] Stationarity is also a general property of time series. A stationary time series is one whose properties (mean, variance,...) do not depend on the time at which the series is observed. A GP with a stationary kernel is a very flexible model for stationary time series.

[17] https://www.cs.toronto.edu/~duvenaud/cookbook/.

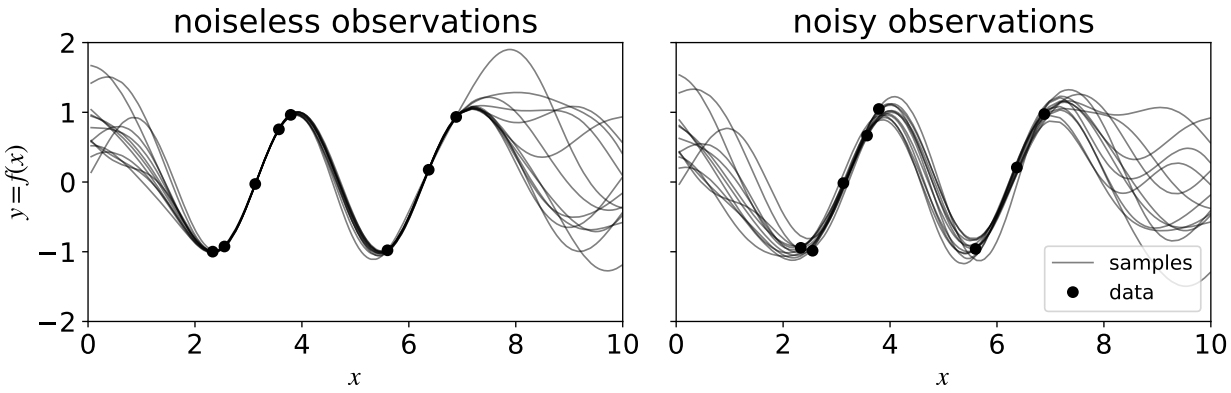


Figure 2.13: Samples from a Gaussian process with squared exponential kernel $l = 1$ conditioned on observations simulated from a simple sinusoid. The left panel shows the case when we assume that the observations are noiseless in which case the GP perfectly interpolates the data. The right panel shows the case when we assume that the observations are generated from the GP with the addition of some white Gaussian noise with variance 0.1^2.

evaluated at $\mathbf{X}$ with length n_1. If we are interested in predicting the function values $\mathbf{f}^*(\mathbf{X}^*)$ at a new vector of data points $\mathbf{X}^*$ with length n_2 (so we could, for example, make a plot), then we need to compute the posterior distribution $p(\mathbf{f}^*|\mathbf{f}, \mathbf{X}, \mathbf{X}^*)$. To compute the conditional distribution, we start with the join distribution over $\mathbf{f}$ and $\mathbf{f}^*$. Since $\mathbf{f}$ and $\mathbf{f}^*$ both come from the same multivariate Gaussian distribution, the joint distribution $p(\mathbf{f}, \mathbf{f}^*)$ is also a multivariate Gaussian:

$$\begin{pmatrix} \mathbf{f} \\ \mathbf{f}^* \end{pmatrix} \sim \mathcal{N}\left[\begin{pmatrix} \boldsymbol{\mu} \\ \boldsymbol{\mu}^* \end{pmatrix}, \begin{pmatrix} \mathbf{K}(\mathbf{X}, \mathbf{X}) & \mathbf{K}(\mathbf{X}, \mathbf{X}^*) \\ \mathbf{K}(\mathbf{X}^*, \mathbf{X}) & \mathbf{K}(\mathbf{X}^*, \mathbf{X}^*) \end{pmatrix} \right] \quad . \tag{2.192}$$

From this joint distribution, we can obtain the conditional distribution (see appendix A2 in Rasmussen and Williams, 2006) which is also a multivariate Gaussian[18]:

$$p(\mathbf{f}^*|\mathbf{X}^*, \mathbf{X}, \mathbf{f}) \sim \mathcal{N}(\mathbf{K}\left(\mathbf{X}^*, \mathbf{X}\right) \mathbf{K}(\mathbf{X}, \mathbf{X})^{-1} \mathbf{f} \tag{2.193}$$

$$\mathbf{K}\left(\mathbf{X}^*, \mathbf{X}^*\right) - \mathbf{K}\left(\mathbf{X}^*, \mathbf{X}\right) \mathbf{K}(\mathbf{X}, \mathbf{X})^{-1} \mathbf{K}(\mathbf{X}, \mathbf{X}^*)) \quad . \tag{2.194}$$

Samples from this posterior GP distribution are given by Equation 2.194 with a squared exponential kernel conditioned on a set of simulated observations $\mathbf{f}$ generated from a simple sinusoid are shown in the left panel of Figure 2.13. Each draw from the distribution perfectly interpolates all the data points. The variance is naturally larger where the data are sparse, and it grows rapidly in the extrapolative regime outside of the range of the data. The squared exponential kernel is not a great choice here because the covariance rapidly decays to zero with the distance from the first and last data point so the predictions look flat further away from the range of the data. A much better choice would be a kernel with a periodic covariance structure which would not have this problem.

The predictions shown in the left panel assume that the observations are noiseless which is why each sample perfectly interpolates the data. It is more common to assume that there is an additional white Gaussian noise component in the observed data. In that case, we

[18] This is one of many special properties of Gaussians.

add a diagonal matrix (because white noise is independent Gaussian noise) to the covariance matrix $\mathbf{K}(\mathbf{X}, \mathbf{X})$:

$$\mathbf{K}(\mathbf{X}, \mathbf{X}) + \sigma^2 \mathbf{I} \quad , \tag{2.195}$$

where $\mathbf{I}$ is the identity matrix. Predictions from a GP with this additional noise term are shown in the right panel of Figure 2.13.

Fitting for the parameters of a GP kernel

I showed how to sample a GP prior and how to compute predictions from a GP conditioned on a set of observations but I haven't discussed the parameters which define the kernel function, such as the length scale l in the squared exponential kernel. These parameters are often called the *hyperparameters* of the kernel. Given a set of observations $\mathbf{f}$ evaluated at $\mathbf{X}$ with length n, the likelihood of a GP model for the kernel hyperparameters $\boldsymbol{\theta}$ is

$$p(\mathbf{f}|\boldsymbol{\theta}) \sim \mathcal{N}(\mathbf{f}|\boldsymbol{\mu}(\boldsymbol{\theta}), \mathbf{K}(\mathbf{X}, \mathbf{X}; \boldsymbol{\theta})) \quad , \tag{2.196}$$

where $\boldsymbol{\theta}$ is a set of parameters which define the kernel function. We can treat these kernel parameters as we would any other parameters, for example, we can optimize the likelihood in Equation 2.196 and find the MLE parameters $\hat{\boldsymbol{\theta}}_{\mathrm{MLE}}$. The likelihood is not linear in these hyperparameters and the optimisation problem is generally not convex.

The major challenge with GPs and the reason they still are not as prevalent as, for instance, linear regression, is that sampling the GP prior or computing the marginal likelihood (Equation 2.196) requires an inversion of a dense $n \times n$ covariance matrix, an operation which scales as $\mathcal{O}(n^3)$. This operation becomes prohibitively expensive for datasets with a few thousand data points. Thankfully, there has been a lot of work on faster algorithms for inverting GP covariance matrices. Most of these algorithms rely on the assumption of some special structure for the covariance matrix which can be exploited to speed up linear algebra operations. The most notable example is the celerite algorithm (Foreman-Mackey et al., 2017) which enables the use of 1D GPs with linear $\mathcal{O}(n)$ scaling as long as we restrict ourselves to kernels which are sums of complex exponential functions. These kernels are sufficiently flexible for modelling physical time series, and the sums of complex exponentials also have a physical interpretation as mixtures of stochastically driven, damped simple harmonic oscillators. Thanks to celerite it is possible to fit GPs to astronomical light curves with thousands of data points.

Use of GPs in astronomy

The use of Gaussian process models within astronomy has skyrocketed in recent years. They are most commonly used to model correlated noise in astronomical light curves. In practice, this is a small step from just assuming independent Gaussian noise. All we have to do is to replace a likelihood with a diagonal covariance matrix whose entries are given by the variance estimates for individual data points ("errorbars") with a dense covariance matrix generated using some kernel function (while still including the independent noise term). The mean of the multivariate Gaussian likelihood is usually an astrophysical model such as a the predicted flux in a transit, secondary eclipse, or a microlensing event. In some cases, the mean is set

to zero and the hyperparameters of the kernel have physical meaning. For instance, GPs have been used to model stellar rotation periods using a quasi-periodic kernel specified with multiple parameters which describe the stochastic variability of the star (Angus et al., 2018). Luger et al. (2021b) went a step further, starting with properties of distributions of stellar spots expressed in the starry formalism, they derived a closed-form expression for a GP which describes a light curve of a rotating, evolving stellar surface conditioned on the distribution of stellar spot sizes, contrasts and latitudes.

The advantage of GP models relative to alternatives such as linear regression with a fixed number of basis functions is that they are far more flexible and allow us to make less rigid assumptions about the phenomenon of interest. This section was just a brief summary of their key properties and there is much more to GPs that I haven't discussed (for example, fitting GPs to multi-dimensional data). In Chapter 7 I will show two very different use cases of GPs in the context of occultation mapping.

2.4 Statistical inference - computation

In the previous section, I briefly summarised some key aspects of probability theory and Bayesian statistics, I discussed maximum likelihood estimation, GPs and linear models for which inference can be done analytically. In most cases, models of real-world interest are not well behaved, and maximum likelihood estimation or sampling the posterior distribution distribution becomes very challenging because of the complex likelihood and posterior landscapes. In those cases, computation needs to go hand-in-hand with model building and it is very important to understand how the structure of the model and the data maps to the structure of the likelihood. I start by introducing some very fundamental concepts such as the *curse of dimensionality* and the *no-free-lunch theorems*. Next, I discuss various computational inference methods such as optimisation methods, Markov chain Monte Carlo and Variational Inference. Finally, I discuss model comparison by means of Bayes factors and cross-validation.

2.4.1 The curse of dimensionality and the no-free-lunch theorem

The *curse of dimensionality* refers to a phenomenon in which the *volume of possible explanations* for the data grows exponentially with the increasing dimension of the problem. Consider the following example, borrowed from the excellent paper by Roberts (2021). There are images consisting of n pixels where each of the pixels is either black (0) or white (1). The number of possible images is 2^n and it grows exponentially with n. If each image out of the 2^n possible images is described by one of two labels, A, or B, then the total number of possible ways of labelling the set of images is $2^{(2^n)}$, a stupidly large number. If $n = 9$, the number of ways to classify the set of all 9-bit black-and-white images is greater than the number of atoms in the observable universe. If the labels do not correlate with the properties of the images, then one strategy for solving this problem would be the lookup table approach – memorize each image with its label. The famous *no-free-lunch theorem* (NFL), proved by the mathematician David Wolpert (Wolpert, 1996), demonstrates that it is impossible to construct an algorithm which performs better than the lookup table approach.

Of course, the assumption that the labels aren't correlated to features in the images is silly for any meaningful problem. For instance, all images of cats have certain things in common as do images of galaxies. The no-free-lunch theorem doesn't say that it is impossible to construct an algorithm to classify images of cats, it only says that there is no best algorithm *if we average over all possible inputs to the problem.* To construct an algorithm which works in practice we need to impose structure on the problem – what machine learning researchers call *inductive bias.* A set of assumptions which work in one domain may not work well in another domain and no single model will work well in all domains.

2.4.2 Sampling vs. optimisation

Broadly speaking, in statistical inference, we can differentiate between two classes of methods for fitting models to data. Optimisation methods cast the problem of estimating the likelihood or the posterior as a (generally, non-convex) optimisation problem. The goal in optimisation problems is to find a single point which minimises or maximizes some scalar function called a *cost function* or an *objective function.* The space of possible cost functions is infinite but in probabilistic inference, these are most often (usually unnormalised) probability distributions such as likelihoods and posteriors. Examples include MLE and MAP estimation, the Laplace approximation and variational inference.

Sampling methods are a fundamentally different class of algorithms whose purpose is accurately computing multidimensional *integrals* such as expectation values over some distribution $p(\boldsymbol{\theta})$:

$$\mu \equiv \mathbb{E}_{p(\boldsymbol{\theta})}[g(\boldsymbol{\theta})] = \int g(\boldsymbol{\theta}) p(\boldsymbol{\theta}) \, \mathrm{d}\boldsymbol{\theta} \quad , \tag{2.197}$$

where $g(\boldsymbol{\theta})$ is some arbitrary function, for instance, $g(\boldsymbol{\theta}) = \boldsymbol{\theta}$ in the case of the mean. These integrals are intractable when the dimension of $\boldsymbol{\theta}$ is greater than some small number (3-5 or so, depending on the problem) because the volume of the parameter space grows exponentially and finding the regions in parameter space where the integral is nonzero becomes ever more difficult. We can get around this problem by generating samples from the probability distribution $\boldsymbol{\theta}^{(s)} \sim p(\boldsymbol{\theta})$, and computing the integral using *Monte Carlo Integration* in regions where there is non-negligible probability:

$$\hat{\mu}_{\mathrm{MC}} = \mathbb{E}_{p(\boldsymbol{\theta})}[g(\boldsymbol{\theta})] \approx \frac{1}{S} \sum_{i=1}^{S} g\left(\boldsymbol{\theta}^{(s)}\right) \quad , \tag{2.198}$$

where $\hat{\mu}_{\mathrm{MC}}$ is the simple Monte Carlo estimator of the expectation μ. It can be shown that if the expectation μ exists, the estimator $\hat{\mu}_{\mathrm{MC}}$ is consistent (it asymptotically converges toward μ with increasing S) and unbiased (the mean of the sampling distribution of the estimator is equal to μ). If the expectation of g^2 is also finite the central limit theorem holds and the error of the simple Monte Carlo estimator decreases as $1/\sqrt{S}$. This error is independent of the dimension! The major challenge is how to generate the samples from $p(\boldsymbol{\theta})$. This is usually accomplished by sampling values from probability distributions that are easy to sample from (most often, multivariate normal distributions), and then reweighting those samples at each iteration of the algorithm so that they match the target distribution. Examples of sampling

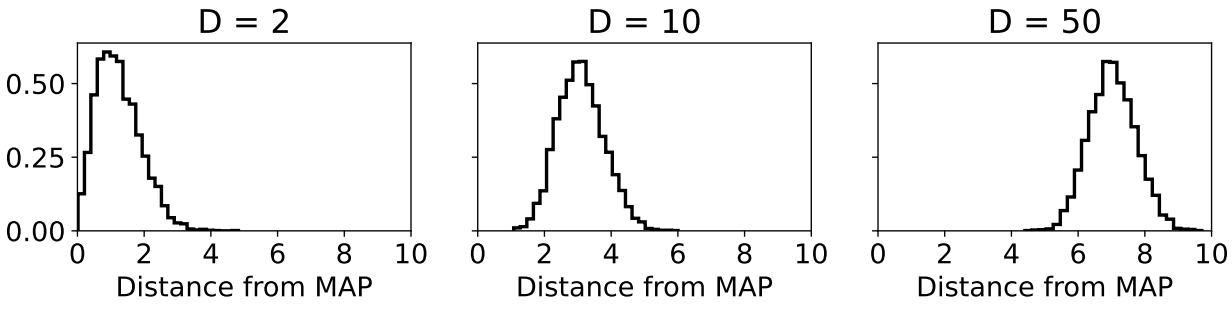


Figure 2.14: Histograms showing the distance from the MAP point for a set of samples from D-dimensional multivariate normal distribution with zero mean and unit variance. With an increasing number of dimensions, the probability mass concentrates in a thin shell centred at the MAP point called the typical set. For $D = 50$ the posterior samples are coming from a region more than 5 sigma away from the MAP point.

methods include Markov chain Monte Carlo (MCMC), importance sampling, and rejection sampling.

Contrary to what we might intuitively think, in all but the lowest dimensional models, samples from the posterior distribution are not located anywhere near a point estimate such as the MAP point in the parameter space, despite the fact that this is where the posterior density is highest. This happens because expectation values depend not only on the density of the pdf $p(\boldsymbol{\theta})$, but also on the volume term $\mathrm{d}\boldsymbol{\theta}$ and the value of the integral is non-negligible only when both are jointly maximised. In high dimensions, the volume grows exponentially and most of the volume ends up being concentrated far away from the mode so that the product of the density and the volume, the *posterior mass*, ends up being concentrated in a thin shell around the mode called the *typical set* (for a good overview of this issue see Betancourt, 2017). This point is illustrated in Figure 2.14 using samples from a D-dimensional multivariate normal distribution with zero mean and unit variance. The reason this happens is that although the most likely points are located in the neighbourhood of the mode, there are many more points further away in the typical set and those are the ones that contribute to the integral involved in the calculation of expectation values. For a toy example shown in Figure 2.14, this this does not matter much because the posterior has spherical symmetry, but for non-Gaussian posteriors we can get very different results when computing the mean (using sampling) and the MAP point (using optimisation).

2.4.3 Optimisation methods

In this section, I provide a brief overview of the two most relevant methods for estimating the posterior distribution $p(\boldsymbol{\theta}|\mathcal{D})$ using optimisation algorithms.

The Laplace approximation

One of the simplest ways of approximating a posterior distribution is the *Laplace approximation*. We start with the general posterior density of the form

$$p(\boldsymbol{\theta}|\mathcal{D}) = \frac{1}{\mathcal{Z}} e^{-\mathcal{E}(\boldsymbol{\theta})} \quad , \tag{2.199}$$

where $\mathcal{E}(\boldsymbol{\theta}) \equiv \ln p(\boldsymbol{\theta}, \mathcal{D})$ is called an *energy function* and $\mathcal{Z}$ is the fully marginalised likelihood. Following Murphy (2022), we Taylor expand $\mathcal{E}(\boldsymbol{\theta})$ around the mode $\hat{\boldsymbol{\theta}}$ as follows

$$\mathcal{E}(\boldsymbol{\theta}) = \ln\, p(\boldsymbol{\theta}, \mathcal{D}) \approx \mathcal{E}(\hat{\boldsymbol{\theta}}) + (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})^\intercal \mathbf{g} + \frac{1}{2}(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})^\intercal \mathbf{H} (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}) \quad , \tag{2.200}$$

where $\mathbf{g}$ is the gradient at the mode (the Jacobian matrix), and $\mathbf{H}$ is the *Hessian* evaluated at the mode. The gradient term by definition vanishes at the mode $\hat{\boldsymbol{\theta}}$ so we are left with

$$\hat{p}(\boldsymbol{\theta}, \mathcal{D}) = e^{-\mathcal{E}(\hat{\boldsymbol{\theta}})} \exp\left[-\frac{1}{2}(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})^\intercal \mathbf{H} (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})\right] \quad , \tag{2.201}$$

from which it follows that

$$\hat{p}(\boldsymbol{\theta}|\mathcal{D}) = \frac{1}{\mathcal{Z}}\, \hat{p}(\boldsymbol{\theta}, \mathcal{D}) = \mathcal{N}\left(\boldsymbol{\theta} | \hat{\boldsymbol{\theta}}, \mathbf{H}^{-1}\right) \quad . \tag{2.202}$$

Thus, we obtain the well-known result that a Taylor expansion of an arbitrary distribution $p(\boldsymbol{\theta}|\mathcal{D})$ around the mode $\hat{\boldsymbol{\theta}}$, truncated at the first non-vanishing term, is simply a multivariate normal distribution centred at the mode. The covariance matrix of this distribution is given by the inverse Hessian (which specifies the *curvature* of the distribution) evaluated at the mode.

Laplace approximation works very well with posteriors which are well described by a multivariate normal distribution. It is straightforward to compute because we can use an optimizer to find the MAP point and then estimate the Hessian numerically using finite difference approximation for the gradient or use automatic differentiation to obtain the exact Hessian (I will introduce automatic differentiation in Section 2.5). Both of these operations are generally very fast.

Variational inference

Another method which uses optimisation to obtain an approximation to the true posterior is *Variational inference* (VI). The advantage of VI over the Laplace approximation is that we can fit any distribution we like to the posterior. The key idea is to model an intractable density $p(\boldsymbol{\theta}|\mathcal{D})$ with some off-the-shelf distribution $q(\boldsymbol{\theta})$ such that we minimise some measure of discrepancy between the two distributions. We optimize for the parameters of the distribution $q(\boldsymbol{\theta}|\boldsymbol{\psi})$. These parameters are called the *variational parameters*. The most common discrepancy measure is the KL divergence from q to p (Equation 2.141). Thus, the optimisation problem is to find the parameters $\boldsymbol{\psi}^*$ which minimise the KL divergence:

$$\boldsymbol{\psi}^* = \underset{\boldsymbol{\psi}}{\operatorname{argmin}}\, D_{\mathbb{KL}}(q(\boldsymbol{\theta}|\boldsymbol{\psi}) \,\|\, p(\boldsymbol{\theta}|\mathcal{D})) \quad . \tag{2.203}$$

Making use of Equation 2.141, we can expand the above equation as (Murphy, 2022)

$$\boldsymbol{\psi}^* = \underset{\boldsymbol{\psi}}{\operatorname{argmin}}\, \mathbb{E}_{q(\boldsymbol{\theta}|\boldsymbol{\psi})}\left[\ln q(\boldsymbol{\theta}|\boldsymbol{\psi}) - \ln\left(\frac{p(\mathcal{D}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathcal{D})}\right)\right] \tag{2.204}$$

$$= \underset{\boldsymbol{\psi}}{\operatorname{argmin}}\, \underbrace{\mathbb{E}_{q(\boldsymbol{\theta}|\boldsymbol{\psi})}[-\ln p(\mathcal{D}|\boldsymbol{\theta}) - \ln p(\boldsymbol{\theta}) + \ln q(\boldsymbol{\theta}|\boldsymbol{\psi})]}_{-\mathrm{ELBO}(\boldsymbol{\psi})} + \ln p(\mathcal{D}) \quad . \tag{2.205}$$

Since the evidence $\ln p(\mathcal{D})$ is not a function of the variational parameters $\boldsymbol{\psi}$, the problem of solving Equation 2.203 reduces to maximising the term

$$\text{ELBO}(\boldsymbol{\psi}) \equiv \mathbb{E}_{q(\boldsymbol{\theta}|\boldsymbol{\psi})}[\ln p(\mathcal{D}|\boldsymbol{\theta}) + \ln p(\boldsymbol{\theta}) - \ln q(\boldsymbol{\theta}|\boldsymbol{\psi})] \quad , \tag{2.206}$$

which is called the *evidence lower bound* (ELBO) because $D_{\mathbb{KL}}(q\|p) \geq 0$ implies that $\text{ELBO}(\boldsymbol{\psi}) \leq \ln p(\mathcal{D})$. The most common choice for the distribution q is, unsurprisingly, a multivariate Gaussian. VI with a multivariate Gaussian differs from the Laplace approximation because in VI we are optimising for both the mean and the covariance matrix rather than equating the covariance matrix to the inverse Hessian at the MAP point. VI is most commonly used with very large datasets and high dimensional parameter spaces because it is computationally cheaper than MCMC (although VI and MCMC are not doing the same thing). It has seen little use within astronomy.

2.4.4 Sampling methods

Rejection sampling

Rejection sampling is one of the simplest algorithms for generating samples from a probability distribution. Consider a normalised pdf

$$p(\boldsymbol{\theta}) = \frac{\tilde{p}(\boldsymbol{\theta})}{\mathcal{Z}} \quad , \tag{2.207}$$

where $\tilde{\boldsymbol{\theta}}$ is the unnormalised pdf and $\mathcal{Z}$ is normalisation constant, which is not necessarily known. If a proposal distribution $q(\boldsymbol{\theta})$ (which we know how to sample from) satisfies the inequality

$$Cq(\boldsymbol{\theta}) \geq \tilde{p}(\boldsymbol{\theta}) \tag{2.208}$$

for some constant C, the function $Cq(\boldsymbol{\theta})$ provides an upper envelope for $\tilde{p}$. We can use the proposal distribution to generate samples from the target distribution using the following algorithm:

1. Sample $\boldsymbol{\theta}_0 \sim q(\boldsymbol{\theta})$ from the proposal distribution.
2. Sample $u_0 \sim \text{Unif}(0, Cq(\boldsymbol{\theta}_0))$ which corresponds to uniformly drawing a height under the envelope.
3. If $u_0 > \tilde{p}(\boldsymbol{\theta}_0)$ reject the sample, otherwise accept

See Murphy (2023) for a proof that this procedure produces independent samples from $p(\boldsymbol{\theta})$. If $\tilde{p}$ is a normalised target distribution the acceptance probability at each iteration is $1/C$ so C should be as small as possible while satisfying the constraint in Equation 2.208. In practice, if the distribution $p(\boldsymbol{\theta})$ is a posterior distribution $p(\boldsymbol{\theta}|\mathcal{D})$ rejection sampling can be accomplished by using the prior as the proposal distribution, sampling a very large number of parameter vectors from the prior, computing the value of the likelihood $p(\mathcal{D}|\boldsymbol{\theta})$ for each and then setting C to be equal to the maximum value of the likelihood across all samples.

The major drawback of rejection sampling is that it very quickly runs into problems with the curse of dimensionality. For instance, if $p(\boldsymbol{\theta}) = \mathcal{N}(\mathbf{0}, \sigma_p^2\mathbf{I})$ and $q(\boldsymbol{\theta}) = \mathcal{N}(\mathbf{0}, \sigma_q^2\mathbf{I})$ we

need to have $\sigma_q^2 \sigma_p^2 > \sigma_p^2$ to satisfy the bound. In D dimensions the optimal value of C is $C = (\sigma_q/\sigma_p)^D$ and since the acceptance probability is $1/C$ this decreases exponentially fast with dimension D. If the proposal distribution $q(\boldsymbol{\theta})$ approaches 0 in the tails at a faster rate than $p(\boldsymbol{\theta})$ as $\boldsymbol{\theta} \to \infty$, then their ratio $p(\boldsymbol{\theta})/q(\boldsymbol{\theta})$ approaches ∞, and the constraint in Equation 2.208 will not be satisfied. Thus, we must ensure that the target distribution does not have heavier tails than the proposal distribution.

Even though it is the simplest sampling algorithm, rejection sampling can be very useful for problems with small numbers of parameters and small data sets because it can deal with very pathological distributions (for example highly multi-modal distributions). The most notable application of rejection sampling in astronomy is the The Joker sampler (Price-Whelan et al., 2017) for sampling posterior pdfs with very sparse binary-star and exoplanet radial velocity measurements. They use this sampler to obtain posterior pdfs in Keplerian parameters for 232495 sources from the APOGEE catalogue. Because these measurements are very sparse (a few data points per star), the posterior pdf over the period parameter is highly multi-modal. The brute-force approach of rejection sampling works very well and it produces independent samples from the distribution. For any individual object, the samples from those pdf do not tell us much, however, when the analysis is repeated for thousands of objects in the framework of *hierarchical Bayesian inference* it is nevertheless possible to obtain very tight constrain on the population-level parameters[19]. This is because even a highly multi-modal posterior pdf represents a major contraction of the prior parameter space and thus excludes certain parameter values.

The radial velocity model in Price-Whelan et al. (2017) is interesting because structurally it very closely models the single lens microlensing model with parallax effects. Both models have seven default parameters, two of which are linear and in both cases, the posterior can be multi-modal (see Chapter 4 for more on this point). In Price-Whelan et al. (2017) they use the likelihood marginalised over the linear parameters and perform rejection sampling with the dimensionally reduced model. However, this requires an extremely large number of prior samples, on the order of hundreds of millions! Evaluating the likelihood this many times is much more computationally expensive in the microlensing case primarily because microlensing light curves have a lot more data points than stellar RV measurements.

I have tried using rejection sampling with different versions of the single lens microlensing model and I found it to be extremely inefficient. The reason is that the posterior distribution of the RV models in Price-Whelan et al. (2017) is broad because the data they use is extremely sparse (a few data points) but the posterior distribution for a typical single lens microlensing event is much narrower. As a result, the acceptance probability tends to zero even in only three dimensions (t_0, u_0, t_E) and the rejection sampling algorithm becomes unusable.

Importance sampling

Importance sampling is not an algorithm for generating samples from $p(\boldsymbol{\theta})$ but rather a method for estimating expectations $\mathbb{E}_p[g(\boldsymbol{\theta})]$ given an existing set of samples from some

[19]This point is very much relevant for microlensing as well.

other distribution $q(\boldsymbol{\theta})$. We start by rewriting the integral in Equation 2.197 as

$$\mu = \mathbb{E}_p[g(\boldsymbol{\theta})] = \int g(\boldsymbol{\theta}) \frac{p(\boldsymbol{\theta})}{q(\boldsymbol{\theta})} q(\boldsymbol{\theta}) \, \mathrm{d}\boldsymbol{\theta} \quad , \tag{2.209}$$

where we have introduced some other distribution $q(\boldsymbol{\theta})$. The importance sampling estimator of μ is then defined as

$$\hat{\mu}_{\mathrm{IS}} = \frac{1}{S} \sum_{s=1}^{S} \frac{p\left(\boldsymbol{\theta}^{(s)}\right)}{q\left(\boldsymbol{\theta}^{(s)}\right)} g\left(\boldsymbol{\theta}^{(s)}\right) \tag{2.210}$$

$$= \frac{1}{S} \sum_{s=1}^{S} w^{(s)} g\left(\boldsymbol{\theta}^{(s)}\right) \quad , \tag{2.211}$$

where $\boldsymbol{\theta}^{(s)} \sim q(\boldsymbol{\theta})$ are samples from the proposal distribution q and $w^{(s)}$ are the *importance weights*. $w^{(s)}$ are the ratios between the target distribution $p(\boldsymbol{\theta})$ and the proposal distribution $q(\boldsymbol{\theta})$ evaluated at each sample $\boldsymbol{\theta}^{(s)}$. As for rejection sampling, the requirement for the proposal distribution q is that it has support (is nonzero) wherever $p(\boldsymbol{\theta})g(\boldsymbol{\theta})$ is nonzero. Similarly to $\hat{\mu}_{\mathrm{MC}}$, $\hat{\mu}_{\mathrm{IS}}$ is also a consistent and unbiased estimator of μ but its variance is heavily dependent on how well the proposal distribution q matches p.

If we do not know the normalisation constants of p and q we can instead use the self-normalised importance sampling (SNIS) estimator

$$\hat{\mu}_{\mathrm{SNIS}} = \frac{\sum_{s=1}^{S} w^{(s)} h\left(\boldsymbol{\theta}^{(s)}\right)}{\sum_{s=1}^{S} w^{(s)}} \quad , \tag{2.212}$$

which is also consistent but it has a small bias of order $\mathcal{O}(1/S)$. There are also adaptive versions of importance sampling which all rely on iteratively adapting the proposal distribution q based on the importance weights at each iteration.

When the proposal distribution q is a poor approximation of p the distribution of importance weights can have a heavy right tail which results in a large, possibly infinite, variance of $\hat{\mu}_{\mathrm{IS}}$. This is especially true in high dimensions when the importance weights $w^{(s)}$ can vary by several orders of magnitude and the estimator in Equation 2.212 becomes dominated by a few draws from the proposal distribution. One solution to this problem is to fit a generalised Pareto distribution[20] to the distribution of importance weights and then use that distribution to produce a uniformly spaced set of importance weights. This procedure is called *Pareto Smoothed Importance Sampling* (PSIS) (Vehtari et al., 2015c). PSIS forms the basis of a very popular R package loo for high-dimensional leave-one-out cross-validation (Vehtari et al., 2015a). The PSIS algorithm also reports the estimate of the smoothness parameter in the generalised Pareto distribution (*Pareto* $\hat{k}$). $\hat{k}$ acts as a diagnostic because the PSIS estimate is likely to be unstable if $\hat{k}$ is too large (Vehtari et al., 2015c).

What if we are not just interested in computing expectations but also want to obtain posterior samples from $p(\boldsymbol{\theta})$ in order to, say, plot a histogram? We can use importance

[20]Pareto distributions are commonly used in *Extreme Value Analysis* (EVA). EVA is a branch of statistics which deals with heavy and fat-tailed distributions, a regime where many classical methods built around Gaussian assumptions fall apart.

resampling which works by resampling the original samples $\boldsymbol{\theta}^{(1)} \ldots \boldsymbol{\theta}^{(S)}$, with replacement, with probabilities proportional to the importance weights. Practically, this means drawing a set of indices $i_1, \ldots, i_S$ from a multinomial distribution where the event probabilities are the normalised importance weights $\tilde{w}_i = w_i / \sum_j w_j$:

$$i_1, \ldots, i_S \sim \text{Multinomial}\left(1, \tilde{w}_1, \ldots, \tilde{w}_S\right) \quad . \tag{2.213}$$

Samples with larger weights will appear more often than those with smaller weights. We can use Pareto smoothing when computing the weights.

In the context of astronomy, one interesting application of importance sampling is using it to simplify Bayesian hierarchical inference. Hogg et al. (2010) showed that given posterior samples for a collection of objects (obtained using, for example, MCMC), which were obtained with some set of vague priors on the parameters of interest, we can re-weight those samples using importance resampling under a different hierarchical prior which depends on some population-level parameters. Doing this, we can obtain samples from a posterior distribution over the population-level parameters. Hogg et al. (2010) applied this method to infer parameters which describe the eccentricity distribution for a catalogue of N exoplanets, given posterior samples for the eccentricity for each individual planet. This method is highly relevant to microlensing and we have used it in Golovich et al. (2022) to model the t_E distribution for a large collection of single lens microlensing events.

Rosenbluth-Metropolis-Hastings algorithm

The most popular class of algorithms used for generating samples from a general pdf are *Markov Chain Monte Carlo* algorithms, or MCMC for short. The simplest MCMC algorithm and also the very first one that was developed is the *Rosenbluth-Metropolis-Hasting algorithm* (RMH) (also known as the *Metropolis algorithm*). The RMH algorithm is one of the most famous algorithms developed in the 20th century. The original algorithm was published in Metropolis et al. (1953) and it was extended by Hastings (1970). The original version is extremely simple. Given a most recent sample $\boldsymbol{\theta}^{(s)}$, to generate the next sample

- Draw a sample $\boldsymbol{\theta}'$ from a proposal distribution $q(\boldsymbol{\theta}'|\boldsymbol{\theta}^{(s)})$
- Draw a uniform random number $r \sim \mathcal{U}(0, 1)$
- If $\frac{p(\boldsymbol{\theta}')}{p(\boldsymbol{\theta}^{(k)})} > 1$ then $\boldsymbol{\theta}^{(k+1)} \leftarrow \boldsymbol{\theta}'$, else $\boldsymbol{\theta}^{(k+1)} \leftarrow \boldsymbol{\theta}^{(k)}$

The main characteristic of this algorithm is that it defines a biased random walk through the parameter space (usually moving in only one parameter at a time) in such that the amount of time spent at a particular location $\boldsymbol{\theta}$ is proportional to the target density $p(\boldsymbol{\theta})$. These samples can then be used to approximate some expectation value of interest by means of Equation 2.198. The output samples are always, to an extent, correlated, so estimates of the expectations do not converge as quickly with an increasing number of samples as they would if the samples were independent.

This algorithm works because it defines a process called a *Markov Chain*, a process in which the probability of moving from a current state to a a successive state in the chain depends only on the current state and not on any past states. Markov chains have stationary

probability distributions over states and the stationary distribution for the Markov chain defined by the RMH algorithm turns out to be the target distribution $p(\boldsymbol{\theta})$. For this to happen, the proposal distribution q must satisfy a property called *detailed balance* which is defined as

$$q(\boldsymbol{\theta}'|\boldsymbol{\theta}) = q(\boldsymbol{\theta}|\boldsymbol{\theta}') \quad . \tag{2.214}$$

That is, the proposal has to be reversible so that the probability of going in either direction is the same.

The proposal distribution q is one of the main tunable parameters of the RMH algorithm. The simplest choice for q is a multivariate Gaussian distribution $q(\boldsymbol{\theta}'|\boldsymbol{\theta}) \sim \mathcal{N}(\boldsymbol{\theta}, \boldsymbol{\Sigma})$. The covariance matrix $\boldsymbol{\Sigma}$ sets a characteristic *step size* in the parameter space. If the step size is too large, most proposals will be rejected and the chain might miss regions of high probability mass. If on the other hand, the step size is too small, most steps are likely to be accepted, but we might not explore the parameter space fully. RMH (and MCMC samplers in general) do not converge to the stationary distribution of the target distribution immediately; instead, there is always an initial exploration phase before the chain reaches the relevant parts of the parameter space (typical set). The time it takes to reach stationarity is called the *mixing time* of the sampler or the *burn-in* period.

All MCMC algorithms have the same goal, namely, generating samples from a target pdf such that the least number of samples S are needed to accurately approximate the integral from Equation 2.197. Important aspects of different samplers include:

- **Efficiency** – what is the acceptance rate of the proposals?
- **Coverage** – is the target distribution explored completely?
- **Correlation** – how correlated are individual samples with past samples?
- **High dimensions** – can the sampler handle high dimensional spaces?
- **Gradients** – does the sampler require the gradients of the target density?

In real-world problems, MCMC samplers require very careful tuning and initialisation to work well. Although RMH and other MCMC samplers should in principle converge to the posterior distribution independent of the initialisation point in the parameter space and the particular choice of parametrisation, in practice, initialisation, careful tuning of the proposal distribution and how we parametrise the model is crucial for the performance of the sampler. It is not unusual that a minor reparametrisation of a given model leads to an increase in sampling efficiency of several orders of magnitude[21].

Hamiltonian Monte Carlo

Hamiltonian Monte Carlo (HMC) (Duane et al., 1987; Neal, 2011) is a variant of Markov Chain Monte Carlo which uses the gradient of the target density to substantially improve

[21] For this reason, it is often much better to invest time into making a given model more friendly to MCMC than to improve the evaluation time of the model by a factor of a few.

the sampling efficiency compared to gradient-free samplers such as the RMH sampler, especially in high dimensions. The gradient information is especially useful for high-dimensional densities because it enables the sampler to better explore the typical set of parameter space by making proposals along the thin shell comprising the typical set[22].

HMC works by transforming the sampling problem to a classical mechanics problem expressed using the theory of Hamiltonian mechanics. We start by transforming the parameter space $\boldsymbol{\theta}$ into the *phase space* by introducing a set of momentum parameters $\mathbf{p}$ (thus doubling the number of parameters) and defining a *Hamiltonian* of the system to be the negative logarithm of the joint density over $p(\boldsymbol{\theta}, \mathbf{p})$:

$$\mathcal{H}(\boldsymbol{\theta}, \mathbf{p}) \equiv -\ln p(\boldsymbol{\theta}, \mathbf{p}) = -\ln p(\boldsymbol{\theta}) - \ln p(\mathbf{p}|\boldsymbol{\theta}) \quad . \tag{2.215}$$

We can rewrite this Hamiltonian as

$$\mathcal{H}(\boldsymbol{\theta}, \mathbf{p}) = \mathcal{U}(\boldsymbol{\theta}) + \mathcal{K}(\boldsymbol{\theta}, \mathbf{p}) \quad , \tag{2.216}$$

where $\mathcal{U}(\boldsymbol{\theta}) = -\ln p(\boldsymbol{\theta})$ is a potential energy term, $\mathcal{K}(\boldsymbol{\theta}, \mathbf{p})$ is a kinetic energy term and $p(\boldsymbol{\theta})$ is the target density. By mapping a posterior density to a physical system, we can use the machinery of classical mechanics. We first sample the momenta conditional on the parameters $\mathbf{p} \sim p(\mathbf{p}|\boldsymbol{\theta})$ and then solve the Hamilton's equations:

$$\dot{\theta}_i = \frac{\partial \mathcal{H}}{\partial p_i} = \frac{\partial \mathcal{K}}{\partial p_i} \tag{2.217}$$

$$\dot{p}_i = -\frac{\partial \mathcal{H}}{\partial q_i} = -\frac{\partial \mathcal{U}}{\partial q_i} - \frac{\partial \mathcal{K}}{\partial q_i} \quad . \tag{2.218}$$

Symplectic integrators (integrators which respect Liouville's theorem of classical mechanics) are used to integrate Hamilton's equation because of their high accuracy. The most common choice for an integrator is the symplectic (and reversible) *Leapfrog integrator*, which is defined by

$$\mathbf{p}_{t+1/2} = \mathbf{p}_t - \frac{\eta}{2} \frac{\partial \mathcal{U}\left(\boldsymbol{\theta}_t\right)}{\partial \boldsymbol{\theta}} \tag{2.219}$$

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \eta \frac{\partial \mathcal{K}\left(\boldsymbol{p}_{t+1/2}\right)}{\partial \boldsymbol{p}} \tag{2.220}$$

$$\mathbf{p}_{t+1} = \boldsymbol{p}_{t+1/2} - \frac{\eta}{2} \frac{\partial \mathcal{U}\left(\boldsymbol{\theta}_{t+1}\right)}{\partial \boldsymbol{\theta}} \quad , \tag{2.221}$$

where η is the step size. The Leapfrog integrator works by performing a half-update of the momentum, followed by a full update of position and another half-update of momentum. Because no integrators preserve energy perfectly, an RMH acceptance criterion of the form $\alpha = \min\{1, \exp(\mathcal{H}(\boldsymbol{\theta}, \mathbf{p}) - \mathcal{H}(\boldsymbol{\theta}', \mathbf{p}'))\}$ is needed at the end of the integration step in order to restore the detailed balance property of Markov chains. For the pseudocode of a complete algorithm using the Leapfrog integrator, see, for instance, Murphy (2023).

[22] In high dimensions any perturbation away from the thin shell of the typical set is likely to be outside the typical set, which is why samplers that are not aware of the geometry of the target density such as RMH work very poorly in high dimension.

The most challenging part of implementing HMC is tuning the various hyperparameters of the algorithm, namely, the number of leapfrog integration steps L, the step size η and the kinetic energy term $\mathcal{K}$. The most common choice for the kinetic energy term is the Euclidian-Gaussian kinetic energy distribution where the conditional momentum distribution is simply a multivariate normal distribution independent of position:

$$\mathcal{K}(\boldsymbol{\theta}, \mathbf{p}) = \mathcal{N}(\mathbf{p}; 0, \mathbf{M}) \quad , \tag{2.222}$$

where the covariance matrix $\mathbf{M}$ is the *mass matrix*. The choice of the mass matrix $\mathbf{M}$ is important for efficient sampling because every transformation of the parameter space induces an inverse transformation on momentum space (Betancourt, 2017). Consider for example a cigar-shaped target density. We can either transform the posterior such that it looks more Gaussian and make it easier to sample from, or, equivalently, we can choose the momentum distribution such that the momenta are aligned with the shape of the cigar by having higher velocity proposals along the longer axis of the density. The optimal choice for $\mathbf{M}$ is a matrix that is as close as possible to the covariance matrix of the parameter space. The mass matrix is usually estimated empirically from the covariance matrix of the parameters during a burn-in phase of sampling.

As for the leapfrog parameters, we want to set the number of leapfrog steps L to be large enough so that the algorithm explores large swathes of the parameter space. The most popular variant of HMC, the No U-Turn Sampler (NUTS) algorithm (Hoffman and Gelman, 2011), adaptively chooses L such that Leapfrog integration is automatically halted when the trajectory starts going back to its starting point (the momentum vector points in the direction of the starting point). This trick ensures that parameter space updates are as large as possible and it reduces autocorrelation between successive samples. In addition to automatically selecting L, Hoffman and Gelman (2011) propose a scheme for adapting the step size η on the fly. Adding the automatic adaptation of the mass matrix by running short chains during the burn-in period means that NUTS removes any need for tuning. The lack of complicated tuning parameters and the fact that it works very well across a large set of problems is one of the reasons why NUTS (and its derivations) is the work-horse algorithm in many popular statistical modelling packages such as Stan [23] (Carpenter et al., 2017), PyMC[24], numpyro [25] and TensorFlow Probability [26].

How well MCMC works depends almost entirely on the geometry of the target density (which is parametrisation dependent). If the target density is non-Gaussian, HMC requires lots of leapfrog integration steps per iteration. It is entirely possible to have a 3-dimensional density with poor geometry (for example a multi-modal density with well separated modes) that HMC will fail to sample, but if the geometry of the distribution is well-behaved, HMC can scale to even millions of parameters.

One other notable variant of Hamiltonian Monte Carlo which works better for densities with highly complex geometries is Riemannian Manifold Hamiltonian Monte Carlo

[23] https://mc-stan.org/

[24] https://www.pymc.io/welcome.html

[25] https://num.pyro.ai/

[26] https://www.tensorflow.org/probability

(RMHMC) (Girolami and Calderhead, 2011). RMHMC uses a position-dependent kinetic energy term $\mathcal{K}$ and second-order gradient information (quantifying the curvature of the target density) to improve sampling efficiency for strongly correlated densities by adapting to the local geometry of the target density at every point. RMHMC is very rarely used and there are no good open-source implementations of the algorithm. The reason for this is that it is difficult to construct numerically stable integrators for RMHMC[27]. Instead of using RMHMC, we can find better parametrisations for the target density and use NUTS to sample it which can end up being equivalent to RMHMC (Betancourt, 2019).

For an excellent visualisation of different MCMC samplers see here[28].

Diagnosing MCMC chains

How can we tell that our MCMC chains have converged to target density? Unfortunately, there is no point at which we can say that the chain definitively converged to the target density and that we are sampling from the typical set of the posterior density faithfully. At best, we can tell when the chains have not converged. The prime diagnostic for the convergence of MCMC chains is the *integrated autocorrelation time*.

Given S independent samples, the variance σ^2 for a Monte Carlo estimate of an expectation value $g(\boldsymbol{\theta})$ is

$$\sigma^2 = \frac{1}{S}\mathbb{V}_p\left[g(\boldsymbol{\theta})\right] \quad , \tag{2.223}$$

where $\mathbb{V}_p$ is the sample variance of $g(\boldsymbol{\theta})$ under the distribution $p(\boldsymbol{\theta})$. The standard deviation of this estimator will decrease proportional to $1/\sqrt{S}$ with the number of samples. However, samples from $p(\boldsymbol{\theta})$ produced by MCMC are not truly independent. We measure this correlation by computing the integrated autocorrelation time τ_f. If the samples are correlated, the variance in Equation 2.223 becomes

$$\sigma^2 = \frac{\tau_f}{S}\mathbb{V}_p\left[g(\boldsymbol{\theta})\right] \quad , \tag{2.224}$$

where τ_f is the average number of steps needed before the chain "forgets" its initial position. It is defined as

$$\tau_f = \sum_{\tau=-\infty}^{\infty} \rho_f(\tau) \quad , \tag{2.225}$$

where $\rho_f(\tau)$ is the normalised *autocorrelation function* (ACF) of the stochastic process which generated the chain for f[29]. Estimating integrated autocorrelation times is difficult, and in general requires long chains because the estimate of τ_f is slow to converge.

With an estimate of the integrated autocorrelation time we can compute the *Effective Sample Size* (ESS) for a given chain (Sokal, 1997)

$$\text{ESS} \equiv \frac{S}{\tau_f} \quad . \tag{2.226}$$

[27] This discussion on the Stan forums is particularly illuminating

[28] http://chi-feng.github.io/mcmc-demo/app.html.

[29] See this blog post by Daniel Foreman-Mackey for more details on autocorrelation functions: https://dfm.io/posts/autocorr/.

Since the integrated autocorrelation time depends on the function $g(\boldsymbol{\theta})$, so the the ESS. The ESS will thus be different depending on which expectation value we are interested in (mean, quantile, median, etc.). For example, if we are interested in calculating the mean and require a certain fractional precision we need to know the integrated autocorrelation time to estimate how many effective samples are necessary to achieve this precision. Vehtari et al. (2019) suggest computing ESS for both the bulk of the distribution (captured by the median) and *tail-ESS*, which they define to be the minimum of the ESS for the 5% and 95% quantiles of the distribution. Tail-ESS will generally be smaller than the ESS for the median because MCMC converges at a slower rate in the tails of a target distribution.

Since MCMC samples are correlated, some users discard all except every n-th sample in a process called *thinning* the chains. Except for reasons having to do with storage in memory, this procedure should be discouraged because discarding information never helps with obtaining better estimates and the Monte Carlo error in Equation 2.224 already incorporates the fact that the samples are correlated.

Another popular statistic for the diagnostic of MCMC chains is the *Gelman-Rubin* $\hat{R}$ statistic (Gelman and Rubin, 1992) which uses information pooled from multiple chains to determine convergence. It is defined as

$$\hat{R} = \frac{\hat{V}}{W} \quad , \tag{2.227}$$

where W is the sample variance within a single chain, and $\hat{V}$ is the variance estimate pooled across several independent chains. The $\hat{R}$ statistic converges to 1 when each of the chains accurately sample the target density. A value greater than 1 indicates that one or more chains have not yet converged.

Finally, the most sophisticated diagnostic for the convergence of MCMC chains attempts to detect the breakdown of the geometric ergodicity property of MCMC. Geometric ergodicity is a necessary condition for MCMC estimators to follow the central limit theorem. It ensures that the estimator in Equation 2.198 is consistent and unbiased. HMC algorithms enable the detection of so-called *divergences*[30] in the parameter space – points where the target density has a very large gradient indicating a suspected breakdown of geometric ergodicity. The presence of a large number of divergences during sampling can lead to biased estimates. Modern statistical modelling packages include functions for computing ESS estimators, $\hat{R}$ and divergences and without the use of these diagnostic tools, we cannot be sure that the MCMC chains have converged and that our results are correct.

Failure modes for samplers

MCMC methods work really well for distributions with relatively well-behaved geometries but tend to fail with more complex distributions, particularly multi-modal distributions with well separated modes. Buchner (2021b) lists key properties of densities for which MCMC methods tend to fail:

- **Non-Gaussian distributions**: For example banana shaped distributions, distributions with heavy tails.

[30] http://mc-stan.org/users/documentation/case-studies/divergences_and_bias.html.

- **Multimodal distributions**: If there are multiple well-separated modes in the target densities, even if each mode is individually well described by a Gaussian, MCMC methods will tend to get stuck in one of the modes and fail to jump between modes.
- **High dimensionality**: High dimensional problems can break pretty much any algorithm, they are however less of a problem for HMC-like algorithms.
- **Highly informative distributions**: If the data are particularly informative the posterior occupies an extremely small portion of the total prior volume and it may be challenging to discover those regions. Even HMC can fail here if the posterior mass is concentrated in a very narrow "canyon" and the momentum of the proposals is too large.
- **Phase transitions**: These are abrupt changes in the structure of the likelihood as we move up constant likelihood contours.

The first and second failure modes are particularly important for this thesis because the likelihoods for microlensing models are highly non-Gaussian and multimodal. In the next section, I introduce a class of algorithms which (on paper!) addresses almost all of these challenges.

Nested Sampling

Nested Sampling (Skilling, 2004) is fundamentally different from MCMC because it was designed for the purpose of estimating the Bayesian evidence $\mathcal{Z}$ (Equation 2.146), instead of drawing samples from a target distribution. Since $\mathcal{Z}$ is just a normalisation constant, it is usually ignored in traditional MCMC algorithms such as RMH which don't require the target density to be properly normalised. However, $\mathcal{Z}$ becomes important in the context of model comparison using *Bayes factors* (see Section 2.4.5) which are proportional to the ratios of the evidences of the two models.

In addition to estimating $\mathcal{Z}$, Nested Sampling can output samples from posterior $p(\boldsymbol{\theta}|\mathcal{D})$ as a sort of byproduct. The major advantage of NS over MCMC methods is that NS acts as global search algorithm, working from the "outside-in", which improves its performance with multi-modal distributions[31]. Nested Sampling has been extensively used in cosmology, particularly the MultiNest package (Feroz et al., 2009b). Ashton et al. (2022) is great introductory review of Nested Sampling methods, while Buchner (2021b) goes into a lot more depth.

What follows is a brief overview of the basic version of the Nested Sampling algorithm. We start by rewriting the evidence $\mathcal{Z}$ from Equation 2.146 as

$$\mathcal{Z} = \int \mathcal{L}(\boldsymbol{\theta})\pi(\boldsymbol{\theta})\,\mathrm{d}\boldsymbol{\theta} \quad , \tag{2.228}$$

where $\mathcal{L} \equiv p(\mathcal{D}|\boldsymbol{\theta})$ is the likelihood and $\pi(\boldsymbol{\theta})$ is the prior. The key idea behind NS is that it is possible to replace the high-dimensional integral in Equation 2.228 with a one-dimensional

[31] The extent to which this claim is true is investigated in Chapters 4 and 5.

integral over the entire prior volume:

$$\mathcal{Z} = \int_0^1 \mathcal{L}(X)\,\mathrm{d}X \quad , \tag{2.229}$$

where $\mathrm{d}X$ is the volume element of points in the parameters space which share the same likelihood $\mathcal{L}(X)$ weighted by the prior $\pi(\boldsymbol{\theta})$:

$$X\left(\mathcal{L}^\star\right) = \int_{\mathcal{L}>\mathcal{L}^\star} \pi(\boldsymbol{\theta})\,\mathrm{d}\boldsymbol{\theta} \quad . \tag{2.230}$$

$\mathcal{L} = \mathcal{L}^\star$ defines a contour of constant likelihood and $X(\mathcal{L}^\star)$ is the prior volume enclosed by that contour. *Nested Sampling thus replaces the original problem of sampling from the posterior, to a problem of sampling from a likelihood constrained prior.* For a derivation of Equation 2.229 see Ashton et al. (2022).

The standard version of the NS algorithm works by drawing K *live points* (points in the parameter space) from the prior at each iteration i, removing the point with the lowest likelihood value $\mathcal{L}_i$ and replacing it with a new live point sampled from the prior and subject to the constraint $\mathcal{L}_{i+1} > \mathcal{L}_i$ (see Ashton et al. (2022) for details). This process continuously shrinks the volume of the prior by an approximately constant factor, effectively scanning the parameter space from contours with the lowest likelihood to contours with the highest likelihood. The algorithm terminates when some fractional error tolerance on the estimation of the evidence is reached. At the end of the iterative procedure, we are left with estimates of the volume $X(\mathcal{L}^\star)$ at each iteration and we can estimate the integral in Equation 2.229 using the trapezoidal rule:

$$\mathcal{Z} = \sum_{i=1}^{K} w_i\,\mathcal{L}_i^\star \quad , \tag{2.231}$$

where the weights w_i are

$$w_i = \frac{1}{2}\left(X_{i-1} - X_{i+1}\right) \quad . \tag{2.232}$$

The posterior samples can be obtained from the discarded live points by assigning weights $\hat{w}_i$ to each live point:

$$\hat{w}_i \equiv \frac{\mathcal{L}_i\,w_i}{\hat{\mathcal{Z}}} \quad . \tag{2.233}$$

To compute the ESS for these samples, we can use the expectation value-dependent estimator provided by Elvira et al. (2018):

$$\widehat{\mathrm{ESS}} = \frac{1}{\sum_{s=1}^{S} \tilde{w}_i^2} \quad , \tag{2.234}$$

where

$$\tilde{w}_i \equiv \frac{\left|g\left(\boldsymbol{\theta}^{(s)}\right)\right| \hat{w}_s}{\sum_{i=1}^{S} \left|g\left(\boldsymbol{\theta}^{(i)}\right)\right| \hat{w}_i} \quad . \tag{2.235}$$

Buchner (2021b) identifies three distinct phases in the behaviour of the NS algorithm. In the first phase, the prior volume shrinks towards the bulk of the posterior (the typical set)

and during this phase, the live points vary in likelihood values by many orders of magnitude. All of the probability mass is associated with a single live point because the live points are all associated with an equal volume. The typical set is resolved in the second phase at which point the live points have comparable weights. In the final stage, NS reaches the region around the MLE with large values of the likelihood but very small volume. Unlike MCMC, NS has a well-defined termination criterion and running the algorithm for a long time does not generate more posterior samples (there are other ways of generating more samples after an NS run has terminated). NS is different from HMC because HMC explores level sets of energy where the Hamiltonian is constant, while NS scans through a range of energy levels. It is this feature that enables NS to discover multiple modes in the density.

Historically, NS is commonly used by physicists and almost completely ignored by statisticians. This has been changing in recent years and there is now more work from both communities aimed at understanding and improving the algorithm, and better diagnosing its failure modes (see for example Buchner (2014); Higson et al. (2019b,a); Buchner (2021b)). A recent paper by Salomone et al. (2018) reframes the NS algorithm as a special case of *Sequantial Monte Carlo*, a class of algorithms that have been studied by statisticians for decades.

There are lots of modern and open-source implementations of NS. Examples include dynesty (Speagle, 2020), JAXNS (Albert, 2020) and UltraNest (Buchner, 2021a). Out of these, UltraNest is the most developed. It is focused on the correctness of posterior inferences. Ultimately, the only way to check whether NS is doing what it promises to do is to apply it to a real-world problem. In Chapters 4 and 5 we do just that.

Ensemble samplers with affine invariance

By far the most prevalent sampler in physics and astronomy is an MCMC method called *affine invariant ensemble sampler* (AIES) (Goodman and Weare, 2010) which was popularized by an excellent Python implementation of the algorithm called emcee (Foreman-Mackey et al., 2013b, 2019). The method is similar to RMH in the sense that it does not use gradient information and it only requires the evaluation of the log-probability function. The special property of affine invariant MCMC is that it is invariante to *affine transformations* (linear transformations such as scale transformations and rotations) of the parameter space. It uses multiple MCMC chains (called *walkers*) to explore the parameter space. These chains are correlated by construction.

emcee is so popular that the original paper has been cited almost 8000 times so far. The success of emcee has less to do with its actual suitability for a wide range of problems and more to do with the fact that it is very easy to use with the kinds of models astronomers and physicists work with. emcee works with black-box physics simulators and it is also easily parallelizable. This feature makes it easy to misuse and apply to problems for which it is not suitable. For example, Huijser et al. (2015) showed that emcee does not work well in high dimensions (greater than about 10) and it can appear as if the sampler converged when in reality it has not. Nevertheless, emcee can still be very useful and fast for very low dimensional problems. As with other MCMC samplers, it will fail with multi-modal likelihoods. In Chapter 5 I show that emcee should almost certainly not be used to fit binary or triple lens microlensing models.

Other methods

There are many, many algorithms in the literature besides the ones I have listed here. The vast majority of these algorithms are variants of RHMC, HMC or NS. Most of these algorithms do not have well-tested open-source implementations, the examples presented in the papers are usually low-dimensional toy problems, and the algorithms require very careful tuning to work at all. This is why the development of general-purpose statistical modelling (probabilistic programming) packages such as Stan and PyMC has been so important. They demonstrated that methods such as NUTS work well for a wide variety of problems.

2.4.5 Model validation and comparison

So far I have discussed the inference of model parameters using either optimisation methods such as maximum likelihood, or sampling methods such as MCMC and Nested Sampling. Using these methods we can obtain some estimates for the parameters of a specific model, however, this does not tell us if the model is any good in the first place. Every time we do parameter inference we implicitly assume that the particular model is true. A major challenge in statistical inference is comparing different models which can explain the same dataset – so-called *model comparison*.

What we mean by different models is somewhat fuzzy. We could, for example, have a single model specified with one likelihood function and then treat different realisations of the model parameters as separate "models". In some cases, when a model comparison problem can be cast as a parameter inference problem it is easier to treat it as such[32]. This is often the case when two models are *nested*, meaning they share one or several parameters. For example, let's say we are fitting a microlensing model and trying to decide whether or not to include the finite source effect parametrised by a continuous parameter $\rho_\star$. We could fit two separate models, a model which uses a point source approximation for the magnification and one which includes the finite source effects via the $\rho_\star$ parameter and then decide which model better describes the data. The alternative is to simply fit the more complex model, obtain a posterior for the $\rho_\star$ parameter and check if it differs from zero in a meaningful way. Although the former approach is more elegant sometimes it is practically intractable because we might end up having trouble sampling the posterior if the additional parameter is not well constrained by the data[33] or if the model with the extra parameter is much more computationally expensive.

If the models are nested but the parameter which differentiates between the different models is discrete, it is sometimes still possible to cast the model comparison problem as a parameter inference problem using *dimension jumping* methods such as *Reversible-jump MCMC* (RJ-MCMC) (See Brewer and Donovan, 2015, for an application of a method similar to RJ-MCMC to an astronomy problem). However, these methods are computationally very difficult to implement.

[32] Although, as we will see in Chapter 4 it is advantageous to use model comparison methods when comparing different modes in a multi-modal posterior.

[33] In those cases it may make sense to put a strong prior pushing the parameter to zero which helps with inference. If the parameter is well constrained by the data the the likelihood will push the posterior away from zero.

When the models are not nested we have no choice but to do model comparison. An example would be fitting a single-lens microlensing model and a binary-lens or a binary-source star model to the same light curve. Statistics alone do not tell us what to put in the abstract of a paper after we have completed the analysis for multiple competing models[34]. The best we can hope for is assigning probabilities to each of the models, taking into account how well they fit the data and how parsimonious they are. Model comparison is one of the key problems with analyzing microlensing light curves because there are often multiple very different models which make near-identical predictions (see Jung et al., 2017; Han et al., 2020; Rota et al., 2021, for examples from microlensing).

Especially problematic is the fact that when multiple models equally well describe the data, *some models are generally more interesting to researchers than others*. As an example, a triple-lens microlensing model of an exomoon planetary system is much more exciting than a binary source star model. The problem is that the incentives of individual researchers are not always aligned with truth-seeking. The outcome of a model comparison can be strongly dependent on how much effort was invested into a given model[35]. In the extreme case, if we only consider a single model that is most interesting to us, then we will never discover that a more mundane model describes the data equally well. This is why it is very important to have a clearly defined framework for assessing the different models.

Frequentist model comparison (hypothesis testing)

The question of model comparison differs depending on whether one subscribes to frequentist or Bayesian statistics. In classical frequentist statistics, one has to form the *null hypothesis* H_0 which represents a baseline (null) model, and an *alternative hypothesis* which is the actual hypothesis of interest. For instance, the two hypotheses could be

$$\begin{aligned} H_0 &: \text{there is no evidence for a planet in the data} \\ H_1 &: \text{there is evidence for a planet in the data} \quad . \end{aligned}$$

We then form a *hypothesis test* under some *test statistic* and based on the outcome of that hypothesis test we either reject the null hypothesis or we don't, depending on an arbitrary threshold called the *p-value*.

The p-value is the probability of obtaining data equal to, or more extreme than what is observed, *assuming that the null hypothesis H_0 is true*. It is the probability of obtaining the observed dataset in an infinite set of draws of the data. *It is not a statement about the probability of the hypothesis itself.*[36] All hypothesis testing does, is answer the following question: assuming there is nothing interesting in the data, how likely were we to observe the data we were given. If a certain threshold p-value is reached (0.05 in most scientific disciplines, substantially less in physics and astronomy), the null hypothesis is rejected. The fact that the null hypothesis is rejected cannot be interpreted as evidence in favour of the alternative hypothesis because such a statement would involve using Bayes' theorem to invert

[34] This is the subject of decision theory which relies on specifying *utility functions* (utility function depend on our posterior beliefs) to quantify expected loss given a certain decision

[35] This is one of the causes of the replication crisis in many scientific disciplines.

[36] The definition of the p-value is so confusing that even some statistics textbooks get it wrong.

the probabilities. However, it is almost always (mis)interpreted in that way. Given a set of models $\{\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_n\}$, all we are formally allowed to do is to reject some of those models based on a p-value, and keep the rest.

One might wonder if there is an absolute way of judging the quality of a single isolated model. In the astronomical literature, this usually falls under the term of a *goodness-of-fit* statistic. These statistics implicitly use the hypothesis testing procedure described previously, rejecting the null hypothesis under some p-value. Popular choices include the *Chi-squared test* (not to be confused with the χ^2 loss function), the *Kolmogorov-Smirnov test*, the *Anderson-Darling test*, etc. . These tests rely on strong assumptions about the data collection process and they are often misused. It is often preferable (if possible) to use visual checks[37] on the predictions of the model (see Gabry et al., 2017, for a visualisation guide in the Bayesian paradigm) and to quantify the predictive accuracy of the model (see the discussion on pointwise cross-validation scores in Chapter 4). Probabilistic modelling is almost always an iterative process and visualisation helps us spot where the model is failing to describe the data adequately.

Bayesian model comparison using Bayes factors

In the Bayesian paradigm, the most natural way of comparing different models is using Bayes' theorem. For a given model $\mathcal{M}$, we can write down the probability of that model conditional on observed data $\mathcal{D}$ as

$$p(\mathcal{M}|\mathcal{D}) \propto p(\mathcal{M})\, p(\mathcal{D}|\mathcal{M}) \quad , \tag{2.236}$$

where $p(\mathcal{M})$ is the prior probability of the model $\mathcal{M}$ and $p(\mathcal{D}|\mathcal{M})$ is the Bayesian evidence. When comparing two models, model $\mathcal{M}_o$ with a set of parameters $\boldsymbol{\theta}_0$ and and model $\mathcal{M}_1$ with a set of parameters $\boldsymbol{\theta}_1$, we can form the ratio of their posterior probabilities given the data:

$$\frac{p(\mathcal{M}_0|\mathcal{D})}{p(\mathcal{M}_1|\mathcal{D})} = \frac{\mathcal{Z}_0}{\mathcal{Z}_1} \frac{p(\mathcal{M}_0)}{p(\mathcal{M}_1)} \quad . \tag{2.237}$$

The ratio of the two evidences on the right-hand side is called the *Bayes factor*, and the ratio of prior probabilities is the *prior odds*. Prior odds can be set to unity under the assumption that both models are equally likely a priori. A large Bayes factor $\mathrm{B}_{01} \equiv \mathcal{Z}_0/\mathcal{Z}_1$ should be interpreted as evidence in favour of model 0 relative to model 1, given the observed data $\mathcal{D}$. As in the case of hypothesis testing, one can then define arbitrary criteria for the strength of the evidence in favour of model 0 conditional on the specific value of the Bayes factor. The popular choice is using Jeffreys scale (Jeffreys, 1939).

Unlike the p-value, Bayes factors (multiplied by prior odds) are actual probabilities for the models themselves, rather than probabilities for the data in some infinite set of trials. One other useful aspect of Bayesian model comparison is the fact that it automatically implements *Occam's razor*, penalising overly complicated models if there is a simpler alternative (see Chapter 28 of Mackay, 2003). The goal when using the Bayes factor is to either select a single "best" model $\mathcal{M}_i$ or, the more Bayesian option, to average over a set of multiple models using their posterior probabilities $p(\mathcal{M}_i|\mathcal{D})$.

[37] For example plotting the predictions of the model in data space, plotting the residuals, etc. .

There are two major problems with Bayes' factor. The first one is the fact that, whereas the Bayesian evidence $\mathcal{Z}$ could often be ignored when inferring the parameters of the models, it is a crucial quantity in Bayesian model comparison. Computing $\mathcal{Z}$ is very computationally expensive, error-prone, and it requires methods such as Nested Sampling. The other problem with Bayes factors is that the estimates of $\mathcal{Z}$ are particularly sensitive to the width and the shape of the prior distribution for the parameters. For example, if the prior for the parameter θ_k in model $\mathcal{M}_0$ is $\theta_k \sim \mathcal{N}(0, s^2)$, where s is very large, the evidence will scale as $\mathcal{Z} \sim 1/s$ practically *independently of the data*. This kind of dependence of the evidences on the parameter priors is not unexpected, it means that models that "waste" prior parameter space will be penalised, but it is often undesirable[38].

Thus, to use Bayes factors for model comparison, we have to be able to compute evidences accurately and we have to think very carefully about the choice of priors. Despite these issues, Bayes factors have been often used in statistics and the natural sciences, most notably in cosmology. For example, Martin et al. (2014) estimated the Bayes factors of 193 different inflation models using the Cosmic Microwave Background data from the Planck satellite, and Higgs inflation as the reference model.

Cross validation

The more tractable and robust alternative to estimating Bayes factors is using *cross-validation*. The key idea behind cross-validation (CV) is to judge different models based on how well they predict unseen data[39]. CV can be used to assess the predictive power of a single model (as a sort of goodness of fit metric), or to compare or average multiple models. The key idea is to split the dataset into K parts or *folds*, re-fit the model including the $K-1$ folds, evaluate the objective function (log-likelihood or posterior), and then repeat the procedure for each fold. We can then, for example, add up the K values and compute an estimate the predictive performance of the model. The variant of CV with $K = N$, where N is the number of data points, is called *leave-one-out cross-validation* (LOO-CV).

For more information about Cross Validation and its approximations, see the excellent Cross-validation FAQ from Aki Vehtari. In Chapter 4 I describe CV in a lot more detail and apply LOO-CV in the context of modelling microlensing light curves.

2.4.6 What about machine learning?

How is *machine learning* (ML) different from the topics we have discussed so far? I would say the key difference is that ML, and particularly *deep learning* (DL) has a much greater focus on *prediction* than on *inference* (estimating the parameters of some model). A classic example is that of neural networks, which today routinely have millions, even billions of parameters (weights). However, the values of these weights are not important, only the prediction error on unseen data is what matters. For this reason, and because they are

[38] See the excellent essay on Bayes factors by Navarro (2020).

[39] This is the principal method for model comparison in machine learning. In machine learning, it is customary to split the original dataset into a *training dataset* and a *validation dataset*. The training dataset is used to optimize the main parameters of the model and the loss function evaluated on the validation dataset is used to compare different models or set the hyperparameters of a given model.

computationally much cheaper, optimisation methods such as *stochastic gradient descent* (SGD) are far more prevalent than sampling methods such as MCMC.

Despite the fact that the deep learning revolution started in 2012, deep learning has not had as much impact in the physical sciences as many have expected, despite the fact that the literature is full of deep learning methods applied to problems in astronomy. DL methods tend to be black boxes which are very efficient at the task they were trained to do, but their internal structure is usually illegible. This is not as much of an issue in industry applications as it is in the sciences, where we care deeply about the precise values of very few parameters. Nevertheless, all of this is changing rapidly and there is a lot of work happening on the boundary between physics and machine learning. One example is so-called *likelihood-free inference* (LFI), which aims to emulate expensive physics simulators (such as massive hydrodynamics simulations) using neural networks so that we could replace the expensive simulator with a computationally orders of magnitude cheaper emulator (Cranmer et al., 2020a, see for example the review paper by). There has been a lot of work on neural networks which respect certain physical symmetries (see Hamiltonian and Lagrangian neural networks and *geometric deep learning* in general). Neural networks have also been used to automate the discovery of succinct mathematical descriptions of the dynamical behaviour of a physical system using observations of its behaviour (Cranmer et al., 2020b).

One of the most important consequences of the atmospheric rise of machine learning for science is that the computational infrastructure developed for training large neural networks can also be used for scientific computing. In particular, two key technologies responsible for the deep learning revolution are high-powered *Graphical Processing Units* (GPUs) and *automatic differentiation* (AD). These technologies on their own are extremely useful for science. In the following section, I will introduce the latter.

2.5 Automatic differentiation

Automatic differentiation (AD or autodiff for short)(Wengert, 1964) is an old idea that has revolutionized the fields of machine learning, AI and optimisation in the recent decade. The key idea behind automatic differentiation is that any model implemented in computer code (such as C++ or Python) is fundamentally a composition of simple functions such as additions, multiplications and trigonometric functions which can be easily differentiated analytically. The chain rule from calculus tells us how to combine the derivatives of the parts of the model to obtain the (exact) derivative of complete model. Specialised automatic differentiation libraries first build a representation of the code as a graph, they then evaluate the derivatives at each node, and finally propagate them through the entire graph.

A variant of automatic differentiation called *reverse-mode automatic differentiation* is widely used in machine learning (where it goes under the name of *backpropagation*) to differentiate scalar functions with respect to millions if not billions of parameters. Autodiff has been used to differentiate through all sorts of things, including molecular dynamics simulations, (Schoenholz and Cubuk, 2019), cosmological simulations (Feng et al., 2016), simulations of the magnetic field in nuclear fusion reactors (McGreivy et al., 2021), ODE and PDE solvers, etc. . It has even been applied to differentiating an intermediate LLVM compiler representation of code (Moses and Churavy, 2020). Because there are so many

applications of autodiff, the famous AI researcher Yann LeCun popularized the term *differentiable programming* to refer to any computer code which is automatically differentiable.

Implementing AD for (some) astrophysical models enables the use of advanced MCMC methods such as Hamiltonian Monte Carlo and much faster and more stable numerical optimisation. I use autodiff throughout this thesis and in In Chapter 3 I introduce an entirely automatically differentiable code for computing the magnification of an extended source in binary and triple lens microlensing models. Although, in Chapter 5 I find that, unfortunately, autodiff is of limited utility for that particular problem.

2.5.1 Three ways of differentiating computer programs

There are three ways of computing derivatives:

1. **Symbolic differentiation**: computing *exact* derivatives of functions using either pen and paper or a computer algebra system such as Wolfram Mathematica or SymPy (Meurer et al., 2017).

2. **Numerical differentiation**: computing *approximate* derivatives using *finite difference* methods.

3. **Automatic differentiation**: computing *exact* derivatives of entire computer programs containing things like conditionals, loops, etc..

Symbolic differentiation is familiar to everyone from introductory calculus. Using either pen and paper or Computer Algebra Systems (CAS) we can compute exact derivatives of functions. If our model is expressible as a straightforward, closed-form function it is often not too difficult to compute the derivatives of interest using symbolic differentiation, indeed, this is what people have been doing for decades to speed up inference[40]. The challenge with symbolic differentiation is that it is not always possible to easily compute the gradients, especially if the model is implemented in a a code which contains control flow constructs such as loops, conditionals, or iterative solvers. In those cases, the gradients of the output of the model with respect to the input are usually computed using finite-difference methods.

In finite-difference methods, the derivative of some function $f(x)$ is defined as

$$f'(x) = \lim_{h \to 0} \frac{f(x + h/2) - f(x - h/2)}{h} \quad , \tag{2.238}$$

for some small step size h. Numerical gradients are easy to compute even with complicated models but they have two key drawbacks. First, obtaining a good approximation of the derivatives requires a small step size h which is can lead to errors due to numerical cancellations. The second is that if we want to evaluate the Jacobian of a function with lots of input parameters, we would need to compute a finite difference gradient for each partial derivative which is prohibitively expensive in high dimensions.

Automatic differentiation solves these problems. If (reverse-mode) AD is properly implemented, evaluating the exact gradient is only a factor of a few slower than evaluating

[40]Before the advent of easy-to-use open-source AD libraries, the main contribution of many papers in applied AI was computing the analytic derivatives for some specific model.

the function itself, even for models with millions of parameters. The drawback is that it is usually necessary to express the model in terms of constructs which are implemented in specialised AD libraries. It also requires writing code in a purely functional programming style. Fortunately, there are now lots of very flexible and easy-to-use high-performance AD libraries.

2.5.2 Forward and reverse-mode autodiff and the chain rule

What follows is primarily a summary of the chapter on autodiff from Murphy (2023). We start with notation for the derivatives. Consider a function $f : \mathbb{R}^n \to \mathbb{R}$. We use

$$\partial x_1 f\left(x_1, x_2, \ldots, x_n\right) \quad , \tag{2.239}$$

to denote the partial derivative with respect to argument x_1, evaluated at the point $(x_1, x_2, \ldots, x_n)$. We define the derivative operator ∂ and denote the derivative of a differentiable function $f : \mathbb{R}^n \to \mathbb{R}^m$ as

$$\partial f : \mathbb{R}^n \to \mathbb{R}^{m \times n} \quad . \tag{2.240}$$

The function ∂f is a *linear map* which maps a point $\mathbf{x} \in \mathbb{R}^n$ to the *Jacobian* of all partial derivatives evaluated at $\mathbf{x}$. ∂f evaluated at point $\mathbf{x}$, $\partial f(\mathbf{x})$ is the *linearisation* of function f at point $\mathbf{x}$. The Jacobian can be represented as a matrix $\mathbf{J}$ whose elements J_{ij} are given by

$$J_{ij} = \frac{\partial f_i}{\partial x_j} \quad . \tag{2.241}$$

To make this slightly less abstract, consider the case when $m = n = 1$, the function takes a scalar and returns a scalar, then Equation 2.240 is just the familiar derivative $f'(x)$:

$$f' : \mathbb{R} \to \mathbb{R} \quad . \tag{2.242}$$

If f takes a vector and returns a scalar ($m = 1$) then the derivative operator is the *gradient* ∇f:

$$\partial f : \mathbb{R}^n \to \mathbb{R}^{1 \times n} \quad . \tag{2.243}$$

We now introduce two key ingredients for an automatic differentiation framework which enables us to efficiently compute the Jacobian for functions with arbitrary complexity.

Jacobian-vector products (JVPs)

Consider a perturbation $\mathbf{v} \in \mathbb{R}^n$ to the function input $\mathbf{x}$. The mapping

$$(\mathbf{x}, \mathbf{v}) \mapsto \partial f(\mathbf{x})\, \mathbf{v} \tag{2.244}$$

is called the *Jacobian-vector product* (JVP). For example, if $f = \cos(x)$ for a scalar x and v is also a scalar, the JVP would be

$$(x, v) \mapsto \partial \cos(x) v = -v \sin(x) \quad . \tag{2.245}$$

Why is this useful? With a function $f : \mathbb{R}^n \to \mathbb{R}^m$ and a *one-hot* encoded vector $\mathbf{v} = (1, 0, \ldots, 0)^\intercal$, we can use the JVP to evaluate the Jacobian matrix one column at a time:

$$\partial f(\mathbf{x})\,\mathbf{v} = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} \\ \frac{\partial f_2}{\partial x_1} \\ \vdots \\ \frac{\partial f_m}{\partial x_1} \end{pmatrix} \quad . \tag{2.246}$$

Evaluating one column of the Jacobian in this way costs about the same as one function evaluation so this is efficient for functions with "tall" Jacobians (functions with many outputs and few inputs), but inefficient for functions with "wide" Jacobians.

Vector-Jacobian products (VJPs)

Similarly, if we consider a perturbation to $\mathbf{u} \in \mathbb{R}^m$ to the output of the f, we obtain the *vector-Jacobian product* (VJP):

$$(\mathbf{x}, \mathbf{u}) \mapsto \partial f(\mathbf{x})^\intercal\,\mathbf{u} \quad . \tag{2.247}$$

Whereas JVPs are useful for computing the Jacobian matrix one column at a time, we can use VJPs to compute the Jacobian one row at a time:

$$\mathbf{v}^\intercal\,\partial f(\mathbf{x}) = (1\,0\,\cdots\,0) \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{pmatrix} = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} \\ \frac{\partial f_1}{\partial x_2} \\ \vdots \\ \frac{\partial f_1}{\partial x_n} \end{pmatrix} \quad . \tag{2.248}$$

For a scalar function $f : \mathbb{R}^n \to \mathbb{R}$, the VJP is simply the gradient ∇f:

$$\mathbf{v}^\intercal\,\partial f(\mathbf{x}) \equiv \nabla f(\mathbf{x}) = (1) \begin{pmatrix} \frac{\partial f}{\partial x_1} & \cdots & \frac{\partial f}{\partial x_n} \end{pmatrix} = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{pmatrix} \quad . \tag{2.249}$$

Chain rule

Let's now consider a function $f : \mathbb{R}^n \to \mathbb{R}^m$ which is a composition three functions a, b and c:

$$f = c \circ b \circ a \quad . \tag{2.250}$$

The *chain rule* from calculus says that we can differentiate f if we know how to differentiate each of its constituent parts:

$$\partial f(\mathbf{x}) = \partial c(b(a(\mathbf{x}))) \circ \partial b(a(\mathbf{x})) \circ \partial a(\mathbf{x}) \tag{2.251}$$

$$= \partial c(b(a(\mathbf{x})))[\partial b(a(\mathbf{x}))[\partial a(\mathbf{x})]] \quad . \tag{2.252}$$

The idea behind AD is to decompose complicated functions f into constituent parts (sometimes called *primitive functions*) which we know how to differentiate, and then use the chain

rule to obtain the derivative of f. Equation 2.252 consists of a series of matrix multiplications. Although the order in which we multiply those matrices does not matter because of the associativity of matrix multiplication, it does matter a great deal when we are implementing AD on a computer.

Let's say that we want to compute the JVP $\partial f(\mathbf{x})\,\mathbf{v}$ of this composite function:

$$\partial f(\mathbf{x})\,\mathbf{v} = \partial c(b(a(\mathbf{x})))[\partial b(a(\mathbf{x}))[\partial a(\mathbf{x})\mathbf{v}]] \quad . \tag{2.253}$$

Right-to-left evaluation of the expression on the right-hand side of Equation 2.253 is known as *forward-mode automatic differentiation*. In forward-mode AD we evaluate *primal terms* $\mathbf{x}, a(\mathbf{x}), b(a(\mathbf{x}))$ alongside the *tangent terms* $\partial a(\mathbf{x}), \partial b(a(\mathbf{x})), \partial c(b(a(\mathbf{x})))$. If we are interested in evaluating the VJP, the order of operations is reversed:

$$\partial f(\mathbf{x})^\intercal\,\mathbf{u} = \partial a(\mathbf{x})^\intercal\,[\partial b(a(\mathbf{x}))^\intercal\,[\partial c(b(a(\mathbf{x})))^\intercal\mathbf{u}]] \quad . \tag{2.254}$$

To evaluate Equation 2.254 we first have to traverse the whole chain, computing the primals $\mathbf{x}, a(\mathbf{x}), b(a(\mathbf{x}))$ as we go. We then proceed in reverse order, computing the tangents $\partial c(b(a(\mathbf{x})))^\intercal, \partial b(a(\mathbf{x}))^\intercal, \partial a(\mathbf{x})^\intercal$. This process is known as *reverse-mode automatic differentiation* or *backpropagation* (since we are propagating the gradients backwards from the function output).

Using JVPs and VJPs is much more memory efficient than computing the Jacobian matrix directly because all intermediate results are 1D vectors that don't take up a lot of memory. The key difference between reverse-mode and forward-mode is that in the case of forward-mode AD, we can just evaluate the JVPs as we evaluate the function itself, whereas in reverse-mode AD we need to store the primal terms for the entire chain before backpropagating. Thus, reverse-mode AD is more memory intensive. Despite the memory overhead, reverse-mode AD is far more prevalent than forward-mode in optimisation and machine learning because the function f is most often some cost function with a high-dimensional input and a scalar output, so we only need one VJP to obtain the entire Jacobian matrix. There is also mixed-mode automatic differentiation which combines forward-mode and reverse-mode to optimize memory use.

In the general case, functions can be more complex than a simple chain of compositions and in AD frameworks they are represented as *Directed Acyclic Graphs* (DAGs), sometimes also called *computation graphs* or *circuits*. We point the reader to the Section 6.2 in (Murphy, 2023) for details on how to generalise AD to graphs and also higher order AD which is enables by the same formalism.

2.5.3 AD libraries

The big challenge in automatic differentiation is creating a software library which knows how to compute JVPs and VJPs for each primitive operation (for example, multiplication, division, matrix multiplication, array slicing, etc.). Because the primitive operations need to be coded manually, expressing a given computation (say an astrophysics code written in C++) in an AD library often requires some effort. Efficient AD libraries are generally developed by large groups of researchers and/or tech companies.

There are two main kinds of AD libraries, *static graph libraries* and *dynamic graph libraries*. The static graph libraries build the entire computation graph for a given program before the graph is evaluated. Examples include Theano[41] (The Theano Development Team et al., 2016), TensorFlow[42] (Abadi et al., 2016), and the Stan Math Library (Carpenter et al., 2015). Dynamic graph libraries build the graph as the program is being executed. Examples are Tensorflow 2.0, JAX[43] and PyTorch[44] (Paszke et al., 2019). These libraries have many features in addition to autodiff, most notably support for GPUs. They are written in Python but the primitive operations are expressed in a lower-level language such as C++ and CUDA C++ (for GPU support).

The most recent popular and research-oriented library is Google's JAX library. JAX is written in Python and it offers nearly identical functionality to the widely popular NumPy Python library for manipulating multi-dimensional arrays. JAX takes Python code as an input. It then converts it to an intermediate language (IR) representation called jaxpr which is used to perform autodiff. Finally, the resulting code for evaluating functions and their gradients is *Just-In-Time* (JIT) compiled to a lower level assembly like language called HLO-IR using the XLA[45] compiler in such a way that the code can be executed on different hardware architectures such as CPUs, GPUs and TPUs (Tensor Processing Units). In practice, this means that one can write Python code using a NumPy-like library, easily compute gradients of the code with autodiff and get the performance of a compiled language like $C++$ through on-the-fly JIT compilation. JAX natively supports Python language constructs such as loops, conditionals and data structures such as tuples and dictionaries. Almost everything you could write in Python is differentiable with JAX although there is no guarantee that the gradients are meaningful if the function represented by the code is not itself differentiable.

Many of the statistical algorithms I have mentioned in Section 2.4.4 are now implemented in JAX. JAX has also been used within astronomy. Examples include Kawahara et al. (2022) who built an autodifferentiable model for exoplanet spectrum modelling, the exoplanet code Foreman-Mackey et al. (2021) for modelling transit light curves and radial velocity time series which uses JAX as a backend, and Gu et al. (2022) who built a differentiable model for strong lensing. Importantly, JAX makes it relatively straightforward to implement custom primitives. One of the contributions of this thesis is an implementation of a custom JAX primitive for computing the roots of complex polynomials (see Chapter 3 for more details). The process of building custom primitives involves specifying the JVP rule (Equation 2.244). Interestingly, we do not need to also specify the VJP rule (Equation 2.247) because JAX is able to deduce the transpose of the JVP automatically[46] (see Frostig et al. (2021) for details).

[41] Theano is no longer maintained but it has recently been resurrected and turned into aesara (https://github.com/aesara-devs/aesara)

[42] https://www.tensorflow.org/

[43] https://jax.readthedocs.io/en/latest/.

[44] https://pytorch.org/.

[45] https://www.tensorflow.org/xla.

[46] This is possible because the implementation of the JVP rule in Python consists of several JAX primitives such as addition and multiplication and JAX already knows how to transpose those primitives.

2.5.4 Probabilistic modelling + programming + autodiff = probabilistic programming

The final topic in this chapter, *probabilistic programming*, nicely ties up everything I have discussed so far. The term probabilistic programming refers to the use of software libraries which provide a high-level API for building statistical models (usually in the Bayesian paradigm). The key components of modern probabilistic programming languages (PPLs) include a backend autodiff framework, methods for initialising *random variables* with associated pdf or pmf (*probability mass function*) distribution functions, evaluation of the (log) probability, and finally, one or several optimizers and samplers. The idea behind these frameworks is to provide an easy-to-use interface for constructing probabilistic models and abstract away the inference as much as possible. Why is this useful for scientists? Because these frameworks simplify the process of building probabilistic models and because they include well-tested and high performance implementations of state-of-the-art inference methods such as Hamiltonian Monte Carlo.

The most popular such library is Stan (Carpenter et al., 2017). Stan is built in C++, it uses the Stan Math Library for automatic differentiation and a variant of the NUTS HMC sampler. It also has some support for Variational Inference. Stan is oriented primarily towards applications in the social sciences and it is not as flexible and easily extensible as some other PPLs. PyMC is another PPL which is very similar to Stan, except that it is implemented in Python and it uses the aesara[47] library for AD[48]. It also supports JAX. PyMC is somewhat more popular in the physical sciences and easier to extend with custom functionality. The exoplanet code (Foreman-Mackey et al., 2021) is a good example of an astrophysics code which interfaces with PyMC. There is also numpyro[49] (Phan et al., 2019) which is also built in Python and uses JAX for AD, and Tensorflow Probability (Dillon et al., 2017) which supports both TensorFlow and also uses JAX for AD.

PyMC, numpyro and Tensorflow Probability all use a variant of NUTS similar to the one implemented in Stan as the core algorithm for inference but also include support for Variational Inference and additional samplers. I use PPLs (mostly numpyro) extensively throughout the rest of this thesis.

[47] https://github.com/aesara-devs/aesara.
[48] PyMC used to be called PyMC3 and use Theano for AD.
[49] https://num.pyro.ai/.

Chapter 3

caustics – a differentiable code for computing the magnification in single, binary and triple-lens microlensing events

In this section I introduce a novel code[1] for computing the magnification of extended sources using a contour integration method. This work is entirely my own (although I build on several papers, most notably, Kuang et al. (2021), Dominik (1998c) and Bozza et al. (2018)). It is not published at the time of writing[2].

3.1 Introduction

3.1.1 Methods for computing the magnification of extended sources

In order to fit microlensing light curves and use statistical methods such as MCMC, we first need to be able to accurately and efficiently compute the microlensing magnification for an extended, possibly limb-darkened source star located at arbitrary positions in the source plane. There are no analytical solutions to this problem. The only option is to estimate the magnification numerically. I wanted to build a method for solving this problem which satisfies the following requirements:

1. **High precision**: The relative error for the magnification should be at most 10^{-3}. This requirement is imposed by the photometric precision of the light curves and the needed precision might be higher for data of very high photometric quality.

2. **Speed**: A necessary requirement for using statistical optimisation or sampling methods is that one can evaluate the model millions of times in order to explore the parameter

[1] https://fbartolic.github.io/caustics/.

[2] At the time of submission of the final version of this thesis, I have developed a Julia version of the code (called Caustics.jl) which solved many of the issues I described in the chapter (see https://github.com/fbartolic/Caustics.jl). I am currently working on a paper which will describe the new version of the code.

space. This means that we should be able to compute the magnification for an entire lightcurve (with potentially thousands of data points) in *milliseconds* rather than *seconds* or *minutes*.

3. **Support for binary and triple lenses**: The method should support both binary and triple-lensing.

4. **Automatic differentiation**: The method should return the exact gradients of the magnification with respect to all input parameters, obtained through automatic differentiation. This enables the use of advanced statistical methods such as Hamiltonian MCMC and the computation of exact Hessians.

Requirements 1-3 are fairly self-explanatory. The last requirement is novel relative to the existing literature. Implementing automatic differentiation of a complex microlensing model was the major motivation for the work described in this chapter, although, in light of the results of Chapter 5, it may have limited utility.

There are three main methods for computing the magnification of extended sources:

1. **Inverse ray shooting** (IRS) (Kayser et al., 1986; Wambsganss, 1997): A brute-force approach which relies on evaluating the lens equation in the direction $z \to w$, where the lens mapping is one-to-one, and then counting the points which fall inside the source disc.

2. **Image-centred ray shooting** (ICRS) (Bennett and Rhie, 1996; Bennett, 2010): Computing the two-dimensional integral over the images using a Cartesian or a Polar grid covering the full extent of the images in the image plane.

3. **Contour integration** (Kayser et al., 1986; Dominik, 1995; Gould and Gaucherel, 1997; Dominik, 1998c, 2007): Using *Green's theorem* to convert the two-dimensional integral over the extent of the images into a one-dimensional integral over the boundaries of the images.

4. **Direct integration**: Integrating over the circular disc of the source star in the source plane.

Inverse ray shooting is the most conceptually and algorithmically simple approach. It does not require solving the lens equation $w = f(z)$ for the images z. The idea is to first evaluate the lens equation $z \to w$ on a uniform density grid in the image plane z, covering all the images. This is called 'shooting rays' from the image plane to the source plane. The number of points (or *rays*) that land within the source disc divided by the total number of evaluated points is then proportional to the magnification. IRS works with any number of lenses and sources and because of this, it is widely used in quasar microlensing (where there are lots of lenses) and the sources have non-circular shapes. When it comes to planetary microlensing, the drawback of IRS is that the images are shaped as long and thin arcs covering a vanishingly small fraction of the area of any rectangular grid in the image plane, thus requiring very dense and computationally expensive grids.

The second method, ICRS, was first introduced by Bennett and Rhie (1996). The idea is straightforward. Since the magnification of an extended source is equal to the (surface

brightness weighted) area of the images in the image plane, divided by the total flux of the source in the absence of lensing, we can simply compute the two-dimensional integrals in the image plane numerically. Bennett and Rhie (1996) used a rectangular grid to compute the integrals while Bennett (2010) improved performance by switching to a Polar grid and deriving a custom numerical quadrature rule for computing the relevant integrals for a limb-darkened source. The drawback of this method is the same as in the case of IRS. Although the integration grids in ICRS are designed to be as small as they can be while still encompassing the images, the grids are still mostly empty space (at least for small sources). In addition, using the higher-order numerical quadrature rule derived in Bennett (2010) for computing the integral in the radial direction requires solving an optimisation problem at every grid point in the angular coordinate in order to determine the boundary of the image to a precision greater than the grid spacing. I initially pursued this approach but I found that the method is very convoluted, prone to failure when the source limb crosses one or multiple caustics, and computationally expensive[3].

Contour integration avoids the construction of 2D integration grids entirely by making use of Green's theorem to convert an area integral area into an integral over a 1D contour enclosing that area. VBBinaryLensing (Bozza, 2010), the most widely used code for binary-lens microlensing, uses contour integration. The biggest challenge in any contour integration is constructing the contours. (Bozza, 2010) constructs the contours by solving the lens equation at multiple points on the source limb in the source plane which then map to locations on the limb of the physical images in the image plane. Dominik (2007) uses an alternative approach, called adaptive contouring which avoids solving the lens equation. Adaptive contouring recursively subdivides squares in the image plane using a quadtree data structure until points lying on the contours are located. The exact method used by Bozza (2010) does not easily generalise to the case of triple lensing, while adaptive contouring in principle works for any number of lenses although it has never been applied to the triple-lens case. Kuang et al. (2021) were the first to apply contour integration to triple-lensing and this method (with some important changes and adjustments) forms the basis for the algorithm in caustics.

Finally, direct integration is only used for single lens events because numerically integrals over a diverging integrand are numerically unstable.

3.1.2 Overview of the caustics code

caustics is an automatically differentiable Python code which can compute the microlensing magnification of an extended, limb-darkened source for single, binary and triple-lens microlensing systems. The code builds on previous work by Kuang et al. (2021), Dominik (1998c), Dominik (2007) and Bozza et al. (2018). Notable features of caustics include the following:

- **Automatic differentiation**: caustics is the first microlensing code to support auto-

[3]The main problem with this method is accurately determining the boundaries of the extended images, especially when two images are nearly adjacent. Once we solve this problem efficiently and robustly we might as well switch to contour integration using Green's theorem and avoid the complexity, and cost, of evaluating the two-dimensional integrals.

matic differentiation, enabling efficient evaluation of the exact gradients of magnification with respect to any input. This was made possible by writing the entire code in the JAX library (see Section 2.5.3 for more on JAX).

- **Fast computation of triple-lens magnification**: Orders of magnitude faster computation of extended source magnification for triple-lens magnification than alternatives[4].

- **Novel complex polynomial root solver**: caustics uses the Ehrlich-Aberth root solver algorithm which is faster than the widely used version of Laguerre method from Skowron and Gould (2012). The algorithm is also differentiable using the implicit function theorem. It can also be executed on a GPU to solve the lens equation hundreds of thousands of times in parallel, though this feature is not used in caustics.

- **Modular design**: The code is highly modular and easily extensible. Nearly the exact same code is used to compute the magnification of single, binary and triple lenses on both CPUs and GPUs. The code could easily be extended to quadruple lensing and astrometric microlensing. Althought it only supports extended sources with linear limb-darkening at the time of writing, it can easily be extended to support artbitrary source brightness profiles.

Although caustics is written in Python, at runtime JAX JIT-compiles the entire code to an intermediate representation which is then further optimized by XLA (see Section 2.5.3). The XLA compiled code is what is actually executed on the CPU. This enables JAX code to execute at speeds comparable to, and often better than compiled languages such as C and C++. It comes at a cost of having to statically specify the types and the shapes of the inputs and outputs to all functions. We cannot have arrays whose shapes depend on the *values* of function arguments. This was a design decision made by the XLA team in order to maximize the performance of JAX code. The way JIT compilation works in JAX is that an empty abstract array (called ShapedArray with a static shape and data type is first passed to a given function and then the entire computation graph of the function (specifying the shapes and data types of all intermediate values) is traced with this abstract array so that the XLA compiler knows exactly what to expect for any given input to the function with the same shape and data type.

There are other important caveats to keep in mind when using JAX. Python constructs such as if statements, for, and while loops have to be replaced with corresponding JAX operators lax.cond, lax.scan and lax.while_loop (lax is a subset of the JAX library). Using a Python if statement in JAX code will automatically trigger a new JIT compilation at every branching point. A for loop will be automatically unrolled which is ok for small loops but prohibitively expensive (in terms of memory) for large loops. It is preferable to use lax.scan instead. lax.cond and lax.scan are both forward and reverse-mode differentiable but lax.while_loop is only forward-mode differentiable. One other key complication is that although at runtime lax.cond will execute only one branch depending on some condition (like a classic if statement), XLA requires allocating memory for *both* branches. The consequence

[4]See caveat in Section 3.7.

of this constraint is that forward-mode autodiff is a better choice for this application than reverse-mode autodiff.

The rest of this chapter is structured as follows. I start with a brief description of the computation of single lens extended source magnification in Section 3.2. In Section 3.3, I introduce the complex polynomial root solver, a key component needed for computing the binary and triple-lens magnification. Having a robust root solver allows us to easily solve for the point-source magnification in the binary and triple lens case. To compute the extended source magnification, we need to integrate over the closed contours of the images in the image plane. How to construct these contours is discussed in Section 3.4. This is the most challenging part of the code. The next step is doing the numerical integration over the image contours using Green's theorem, which is the subject of Section 3.5. In that section I also compare the performance of caustics to other codes. This completes the description of the methods for computing the magnification of extended source using full contour integration but doing the full integration for every point in the light curve is way too expensive. In Section 3.7, I describe how to switch between the hexadecapole approximation for the magnification and the full contour integration. This procedure speeds up the computation of the magnification for all points in the light curve, and hence the computation of the likelihood, by several orders of magnitude. In Section 3.8 I briefly discuss the automatic differentiation feature. Finally, in Section 3.9, I discuss a possible extension of caustics to astrometric microlensing.

3.2 Single lens magnification

Consider a single lens located at the origin. The lens equation in complex coordinates (Equation 2.70) for the $N = 1$ case reduces to the following simple expression:

$$w = z - \frac{1}{\bar{z}} \quad , \tag{3.1}$$

where w is the (complex) source position and z is the image position. By conjugating Equation 3.1, solving the resulting expression for $\bar{z}$, and plugging it back into Equation 3.1, we obtain a 2nd-degree complex polynomial equation:

$$\bar{w}z^2 - |w|^2 z - w = 0 \quad . \tag{3.2}$$

This quadratic polynomial has exactly two roots, both are valid images of the point source located at w. Using the quadratic formula, the roots are

$$z_{1,2} = \frac{w}{2}\left(1 \pm \sqrt{1 + \frac{4}{|w|^2}}\right) \quad . \tag{3.3}$$

These are the *point source images* in the complex plane z. With an extended circular source with radius $\rho_\star$ located at w, we do not observe these point source images, but rather the *physical images*[5]. I will use the word "images" to refer to both the point source images

[5]These images are what we would see if we could resolve the microlensing event.

(points in the z plane) and the images of an extended source, which are *closed curves* in the z plane. The source brightness-weighted area of these physical images divided by the total flux from the source in absence of lensing is the magnification we are interested in. If we parametrise the source position as $w = w_0 + \rho_\star e^{i\phi}$, we can obtain a version of Equation 3.3 which is parametrised by the position on the source limb ϕ (Witt and Mao, 1994):

$$z_{1,2}(\phi) = \frac{w_0 + \rho_\star e^{i\phi}}{2}\left[1 \pm \sqrt{1 + \frac{4}{w_0^2 + 2\rho_\star w_0 \cos\phi + \rho_\star^2}}\right], \quad 0 \leq \phi \leq 2\pi \quad . \tag{3.4}$$

Following Witt and Mao (1994), we differentiate between three different regimes (see Figure 2.5 for visuals):

1. $|w_0| < \rho_\star$– the images are merged in a ring.
2. $\rho_\star < |w_0| \ll 2$– the images form elongated arcs.
3. $|w_0| \gtrsim 2$– the images are slightly deformed circles.

Witt and Mao (1994) show that if we assume that the source brightness is uniform across the disc, one can solve for the extended source magnification analytically. The solution is composed of complete elliptical integrals of the first, second and third kinds.

There is no analytical solution for an arbitrary brightness profile but Lee et al. (2009) showed that it is straightforward to simply compute the two-dimensional integral in the source plane over the source disc (the direct integration method):

$$A_{\rm tot}(|w_0|, \rho_\star) \equiv \frac{\iint A_{\rm ps}(w)\, I(w - w_0)\, {\rm d}S}{\iint I(w - w_0)\, {\rm d}S} \quad . \tag{3.5}$$

$A_{\rm tot}$ is the extended source magnification, S is the projected disc of the source, $A_{\rm ps}$ is the point source magnification (Equation 2.20), and I is the source brightness. The integral in Equation 3.5 is straightforward to compute numerically because the caustic for a single lens (curve where $A_{\rm ps}$ diverges) is a single point and the magnification pattern is azimuthally symmetric so we can integrate over the half-disc and avoid the singularity (see Lee et al. (2009)).

The direct integration approach generally fails for multiple lenses because caustics are two-dimensional closed curves and there is no simple way to avoid integrating over the diverging integrand[6]. The alternative is to solve the problem in the image plane. This requires solving the lens equation at multiple points along the limb of the source disc. No analytical solution exists for the lens equation with $N > 1$ lenses. It can only be solved numerically using a complex polynomial representation of the lens equation.

[6]It may be possible to develop a useful algorithm for binary and triple-lens magnification in this way. Numerical integration with diverging integrals can work quite well, but it requires the knowledge of the exact location where the integrand is diverging. I can imagine an algorithm for solving the two-dimensional integral over the source disc consisting of the following steps:

1. Set up a fixed size grid in the angular polar coordinate ϕ with N_ϕ points.
2. Using a root-finding method such as the bisection method, at each ϕ_i numerically solve for the exact (up to machine precision) locations of all the caustic crossings in the r coordinate within $[0, \rho_\star]$ by checking where the number of physical images changes by two.
3. Solve the N_ϕ 1D integrals in r by splitting the integration domain at the caustic crossings, and using a

3.3 A differentiable complex polynomial root solver

To obtain the point source images, we have to numerically estimate the roots of a complex polynomial equation with degree N:

$$p(z) = \sum_{i=0}^{N} a_i z^i \quad , \tag{3.6}$$

where a_i are the coefficients of the polynomial, derived from the lens equation (see Appendix A for details). Those roots which satisfy the lens equation (a subset of all complex roots of the polynomial in Equation 3.6) are valid (point-source) images. For binary lenses, this complex polynomial is 5th order, and for a triple-lens it is 10th-order. Solving for the roots of a general polynomial is a well-known problem in numerical methods and there are many algorithms built for this purpose. We can divide these algorithms into two groups: those which cast the root-solving problem as an eigenvalue problem using the fact that an arbitrary complex polynomial can be written as a characteristic polynomial of a some non-symmetric square matrix, and those which search for the roots directly, making use of the gradients of the polynomial. JAX includes a general algorithm for computing the roots of a complex polynomial, jax.numpy.roots, which uses the eigenvalue approach, but it is too slow for microlensing purposes so I decided to implement a faster algorithm using the direct approach.

The simplest method for updating a numerical estimate of the j-th root z_j, given some initial estimate, is the first order Newton's method. The update rule for Newton's method is:

$$z_j \leftarrow z_j - \frac{p(z_j)}{p'(z_j)} \quad , \tag{3.7}$$

it makes use of the first derivative of the polynomial. A more sophisticated method with faster convergence is the second-order Lageuerre's method which makes use of the second derivative of the polynomial. The update rule for Lageuerre's method is:

$$z_j \leftarrow z_j - \frac{n}{G_j \pm \sqrt{(n-1)\left(nH_j - G_j^2\right)}} \quad , \tag{3.8}$$

where $G_j = p'(z_j)/p(z_j)$ and $H_j = -(p'(z_j)/p(z_j))'$. The sign in the denominator is chosen such that it maximizes the absolute value of the denominator.

Applying these update rules to the full polynomial can result in convergence to the same root repeatedly. Because of this, the roots are usually found one by one using a so-called *deflation strategy*. With *explicit deflation*, after the first root is is found, the polynomial is divided by that root using polynomial division so that we obtain a new polynomial,

fixed size quadrature rule suitable for functions with endpoint singularities. For instance, the tanh-sinh quadrature rule (Vanherck et al., 2020). These integrals could be computed in parallel on a GPU.

4. Since integration over r removed the singularities, the ϕ integral can be computed using Gaussian quadrature.

Whether or not this approach could work is an open question.

with a degree one less than the original. This new polynomial still shares the other roots with the original polynomial. The procedure is repeated until all roots have been found. Because the division process introduces some numerical noise, explicit deflation requires a final "polishing" step after all of roots have been found in order to fine-tune the estimates of the roots. This polishing step consists of additional applications of the Laguerre or Newton updates to each of the roots.

An alternative to an explicit deflation strategy is an *implicit deflation* strategy[7] (Cameron and Graillat, 2022) in which the polynomial stays the same but the update rule is modified. The simplest approach to implementing implicit deflation starts with the application of Newton's method to a reduced polynomial $p_j(z)$:

$$p_j(z) \equiv \frac{p(z)}{(z-z_1)\cdots(z-z_{j-1})} \quad , \tag{3.9}$$

where $z_1 \cdots z_{j-1}$ are the roots of the polynomial we have already found and z_j is the root we are searching for. The derivative of this reduced polynomial, $p'_j(z)$, is

$$p'_j(z) = \frac{p'(z)}{(z-z_1)\cdots(z-z_{j-1})} - \frac{p(z)}{(z-z_1)\cdots(z-z_{j-1})} \sum_{i=1}^{j-1} (z-z_i)^{-1} \quad . \tag{3.10}$$

We can then write down the Newton update for the reduced polynomial $-p_j(z)/p'_j(z)$ as

$$z_j \leftarrow z_j - \frac{1}{\frac{p'(z_j)}{p(z_j)} - \sum_{i=1}^{j-1} (z_j - z_i)^{-1}} \quad . \tag{3.11}$$

Equation 3.11 contains only evaluations of the original polynomial and a sum over the previous roots z_i, thus avoiding explicit polynomial division. This is known as Maehly's procedure. A root solver implementing the Maehly procedure is sometimes called the Newton-Maehly method. The key feature of the implicit deflation strategy is that it creates poles at the values of the previous roots (via the $(z_j - z_i)^{-1}$ term), thus avoiding multiple convergences to the same root.

It is also possible to obtain all roots simultaneously using a procedure very similar to the Newton-Maehly method called the Aberth-Ehrlich (Börsch-Supan) method, or AE for short. The only difference is that instead of applying Newton's method to a polynomial divided by the first $j-1$ roots like in Equation 3.9, we apply it to a polynomial divided by the estimates of all the other roots except z_j:

$$\tilde{p}_j(z) \equiv \frac{p(z)}{\prod_{i=1;i\neq j}^{n}(z-z_i)} \quad . \tag{3.12}$$

It follows that the update rule is

$$z_j \leftarrow z_j - \frac{1}{\frac{p'(z_j)}{p(z_j)} - \sum_{i=1\,;i\neq j}^{n} (z_j - z_i)^{-1}} \quad . \tag{3.13}$$

[7] As far as I am aware, this strategy has not been previously mentioned in the microlensing literature.

Starting from some initial estimate for the roots, we simply apply the update from Equation 3.13 to each of the roots in parallel until all roots satisfy some convergence criterion. The AE algorithm is superior to the Newton-Maehly method because we can evaluate the roots in parallel.

What if instead of applying Newton's rule to the reduced polynomial in Equation 3.12 we used Laguerre's rule instead? Cameron (2019) do exactly that to obtain a modified Lauguerre's method with an update rule that is identical to Equation 3.8, except factors G_j and H_j are replaced with

$$G_j = \frac{p'(z_j)}{p(z_j)} - \sum_{\substack{i=1 \\ i \neq j}}^{n} \frac{1}{(z_j - z_i)}, \quad H_j = -\left(\frac{p'(z_j)}{p(z_j)}\right)' - \sum_{\substack{i=1 \\ i \neq j}}^{n} \frac{1}{(z_j - z_i)^2} \quad . \tag{3.14}$$

The advantage of this modified Laguerre method over the AE method is faster convergence thanks to the second order Lageurre update. The drawback is that the Laguerre update is more computationally costly than the first-order Newton update in the AE algorithm.

The two most popular implementations of complex polynomial root solver algorithms are the zroots routine from Numerical Recipes (Press et al., 1992), and the C02AFF routine from the NAG library. The two routines both used to use Laguerre's method with an explicit deflation strategy, but in recent version it was replaced with the modified Laguerre's method from Cameron (2019)[8]. zroots has been prevalent in microlensing in the past, but now the most commonly used algorithm is the algorithm presented in Skowron and Gould (2012), which I will refer to as SGroots from now on. SGroots is a slight modification of the old zroots routine. It uses an explicit deflationary strategy. SGroots switches between Laguerre's method and Newton's method to improve performance. It also includes some optimisations tailored specifically to the properties of 5th order complex polynomial derived from the binary-lens equation, namely, the fact that two of the five roots are always well separated in the complex plane. Skowron and Gould (2012) claim that their algorithm is 1.6–3 times faster than the (old) zroots. An important difference between SGroots and zroots is that zroots, like all other Numerical Recipes algorithms, is released under an extremely restrictive proprietary license, while SGroots is open-source.

For several reasons, I chose to use the Aberth-Ehrlich algorithm in caustics. First, there exists a well-tested open-source implementation of the method accompaniying the paper from Cameron and Graillat (2022), which is written in C[9]. Second, in contrast to SGroots, the AE algorithm uses implicit deflation, thus avoiding the need for polishing. Third, if we have a very good initial estimate for the roots (which is true if we are sequentially solving for the images at closely spaced locations in the source plane) it is cheaper than the modified Laguerre method. This is because, as pointed out by Skowron and Gould (2012), in the limit of vanishing $p(z)$ the Laguerre update is identical to Newton's update while being twice as expensive. SGroots switches to Newton's method in the polishing step for this reason. Finally, Cameron and Graillat (2022) introduced a version of the Aberth-

[8] In fact, the name C02AFF refers to the old routine and the new one is called C02AAF: https://www.nag.com/numeric/nl/nagdoc_latest/flhtml/c02/c02aaf.html

[9] https://github.com/trcameron/CompEA

Ehrlich method which uses compensated floating point arithmetic[10] (most importantly, the compensated Horner's method for evaluating a polynomial and its derivative) to effectively double the working numerical precision from 64 bits to 128 bits, while still using 64-bit variables. Bennett (2010) mentions that quadruple precision may be necessary for triple-lens systems with extreme mass ratios. An example would be a system with an Earth mass planet and a Moon-like satellite.

There are two ways we could implement this algorithm in JAX. The first approach is to write the whole algorithm from scratch in Python using JAX constructs. In the context of similar problems, this is generally the simplest approach and it often results in code that (when JIT-compiled) is as fast as C or C++ code. The second approach is to write the algorithm in C or C++ and then incorporate it into JAX as a custom primitive. Although it was time consuming, I opted for the second approach because I could use the existing code from Cameron and Graillat (2022). The Python wrapper around the C/C++ code tells JAX (more specifically, XLA) how the program transforms under JAX program transformations: jit (JIT compilation), grad (reverse-mode autodiff) and vmap (vectorisation over a multi-dimensional input).

Differentiating through a root solver using the implicit function theorem

If we want a code containing an iterative root-solving algorithm to support forward and reverse-mode automatic differentiation, it is necessary to specify a custom JVP product rule for the root solving primitive operation because JAX does not know how to propagate gradients through the root solver written in C/C++. We could technically avoid doing this by using an autodiff library which can differentiate C/C++ code[11] but it turns out there is a far better approach. Instead of differentiating through all of the unrolled operations of the iterative root solver, which consumes a lot of memory, we can use the implicit function theorem to obtain the derivative of the function outputs (the roots) with respect to the inputs (the polynomial coefficients), by linearising the polynomial at the solution point. The implicit function theorem is often used in autodiff frameworks and ML libraries to simplify differentiation through iterative solvers such as root solvers, non-linear equation solvers and ODE solvers[12] under the condition that we can linearize the function in the neighbourhood of the solution to the iterative algorithm.

Let $f(z, \mathbf{a})$ be the polynomial from Equation 3.6, where $\mathbf{a}$ is a vector of coefficients a_i, and let z^* be a particular root of the polynomial which we can compute using the AE algorithm. e cannot write down an expression for the roots in closed form, but we can say that there exists some implicit function $h(\mathbf{a}) : \mathbb{C}^{n+1} \to \mathbb{C}$ which maps the polynomial coefficients to a given root z^*, such that

$$h(\mathbf{a}) = z^* \quad . \tag{3.15}$$

[10] Compensated arithmetic reduces numerical error in additions of floating point numbers by tracking a separate variable which stores the accumulated errors from each step.

[11] The recent Enzyme library can differentiate through programs written in compiled languages such as C/C++. It works by differentiating through LLVM IR representation of the code and it can even differentiate through CUDA kernels.

[12] For more details see these notes on the applications of implicit differentiation in Deep Learning: http://implicit-layers-tutorial.org/implicit_functions/.

We are interested in writing down the Jacobian $\partial_{\mathbf{a}} h(\mathbf{a}) = (\partial_{a_0} h(\mathbf{a}), \ldots, \partial_{a_n} h(\mathbf{a}))^\intercal$ which gives the derivatives of $h(\mathbf{a})$ with respect to the coefficients a_i , evaluated at a root z^*. This is the derivative we need to compute in order to evaluate the derivative of the magnification (which is a function of the roots) with respect to the physical parameters (that determine the polynomial coefficients). The implicit function theorem states that

$$\partial_{\mathbf{a}} h(\mathbf{a}) = -\left[\partial_z f(z^*, \mathbf{a})\right]^{-1} \partial_{\mathbf{a}} f(z^*, \mathbf{a}) \quad . \tag{3.16}$$

The first part of the equation is the first derivative of the polynomial evaluated at the root (which is just another polynomial with a degree one less than the original). The second part is the derivative of the polynomial with respect to each of the coefficients evaluated at the root. To obtain the full Jacobian matrix with shape $(n+1, n)$ that contains the derivatives of each of the n roots with respect to each of the $n+1$ coefficients, we can simply apply Equation 3.16 to each of the roots z^*.

I have ported the code from Cameron and Graillat (2022) into C++ and wrote a custom JAX primitive[13] which implements a JVP rule $(\mathbf{a}, \mathbf{v}) \mapsto (z^*(\mathbf{a}), \partial z^*(\mathbf{a})\,\mathbf{v})$ using Equation 3.16. As mentioned in Section 2.5, JAX can derive the corresponding VJP rule for reverse-mode automatic differentiation automatically. In addition to implementing a custom JVP rule, in order to support GPUs (which almost all native JAX primitives support by default), I ported the C++ code for the AE solver to Nvidia's CUDA framework[14]. I wrote the code such that the root solver is parallelised over multiple polynomials (with the same degree) and each CUDA thread is responsible for solving for the roots of a single polynomial. The main code implementing the root solver is called on both CPUs and GPUs. The only difference is that in the former case it is executed sequentially and in the latter case, it is parallelised over multiple polynomials.

caustics also allows the user to switch between the compensated and the non-compensated versions of the AE algorithm. The non-compensated version is used by default because it is about two times faster than the compensated version. The GPU version of the solver enables solving for the roots of hundreds of thousands of binary or triple-lens polynomials (and therefore also the point source magnification) in 10s of milliseconds. This enables the computation of 1000×1000 point source magnification maps in a fraction of a second. Despite offering this significant speedup, the GPU version of the root solver is not particularly useful for the root-solving stage of the contour integration algorithm because contour integration requires solving for the roots *sequentially* which makes it unsuitable for parallel processors. McDougall (2014) implemented a CUDA version of the SGroots algorithm on a GPU but they used a (highly inaccurate) direct integration approach for computing the binary-lens magnification which does not require solving for the roots sequentially.

I have compared the CPU version of the algorithm to the implementation of SGroots from VBBinaryLensing by solving a 5th order complex polynomial and found that it is about 40% slower using the default initialisation strategy and the default stopping criteria for each algorithm. This difference in speed is likely due to additional optimisations in SGroots specific to the binary-lens polynomial, and perhaps also due to a less stringent stopping criterion. Ultimately this difference in performance is irrelevant, since, as we shall see in

[13] Following this excellent tutorial by Daniel Foreman-Mackey: https://dfm.io/posts/extending-jax/.
[14] CUDA enables execution of C and C++ code on Nvidia GPUs.

the following section, we only need to run the root solver once with the default initialisation strategy while all subsequent evaluations use a starting point for the roots that is very close to the actual roots. Since both algorithms use Newton updates in that regime, I expect that the practical difference in speed is negligible, although I have not done extensive tests. Note that Fatheddin and Sajadian (2022) also implemented an AE solver for the purpose of solving the binary-lens equation and found that their implementation is about twice as fast as SGroots.

I have tested the validity of the root solver extensively by comparing its output to the default polynomial root solver in JAX, jax.numpy.roots. The JAX implementation of the AE algorithm is up to an order of magnitude faster than jax.numpy.roots. The default solver also does not have support for GPUs because at the time of writing JAX does not support a GPU algorithm for an eigendecomposition of a nonsymmetric matrix. I have also tested that the gradients obtained using the implicit function theorem are correct by comparing them to gradients obtained through the finite difference approximation. All these tests are implemented in the directory tests/test_ehrlich_aberth_primitive.py in the caustics repository.

3.4 Constructing the image contours

3.4.1 Contour integration and Green's theorem

For $N > 1$, the caustics are complex two-dimensional curves in the source plane w and the direct integration approach (Equation 3.5) is not feasible because computing two-dimensional integrals with a diverging integrand is highly inaccurate. For this reason all microlensing software packages instead compute two-dimensional integrals in the image plane z. The total magnification, expressed as an integral over the extended images in the image plane is

$$A_{\text{tot}}(|w_0(z)|, \rho_\star) \equiv \frac{\oiint I_s(w(z) - w_0(z))\,\mathrm{d}S'}{\oiint I_s(w(z) - w_0(z))\,\mathrm{d}S} \quad , \tag{3.17}$$

where $w(z)$ is the lens equation (Equation 2.70). Although the integral in Equation 3.17 does not have the pathological function A_{ps} present in Equation 3.5, the problem now is how to compute the integral over an image region S'.

As mentioned in the introduction of this chapter, different methods have been used to integrate the stellar intensity over the region S' covering the images. In this work, we use Green's theorem[15] to reduce the dimensionality of the problem and transform the integral over S' to an integral over the *boundary* of S', $\partial S'$, which encloses the area of the images. For two continous functions $P(z_1, z_2)$ and $Q(z_1, z_2)$ with continous partial derivatives $\partial_{z_2} P$ and $\partial_{z_1} Q$, where $z_1 \equiv \mathrm{Re}(z)$ and $z_2 \equiv \mathrm{Im}(z)$. Green's theorem states that:

$$\oiint_{S'} \left(\frac{\partial Q}{\partial z_1} - \frac{\partial P}{\partial z_2} \right) \mathrm{d}z_1\,\mathrm{d}z_2 = \oint_C (P\,\mathrm{d}z_1 + Q\,\mathrm{d}z_2) \quad . \tag{3.18}$$

The path of the integration along the curve C is defined to be **counterclockwise**, meaning that if we have a curve oriented in the clockwise direction, we have to flip the sign of the integral to get the correct result.

[15] Green's theorem is the two-dimensional special case of Stokes' theorem.

For now, we shall assume that the source brightness is uniform across the disc. The magnification is then equal to the the total area of the images divided by $\pi\rho_\star^2$. This means that the expression in the parenthesis on the right side of Equation 3.18 has to be equal to 1, a requirement that is satisfied by functions $P(z_1, z_2) \equiv -z_2/2$ and $Q(z_1, z_2) \equiv z_1/2$. It follows that the total magnification is

$$A_{\rm tot}\left(w_0, \rho_\star\right) = \frac{1}{\pi\rho_\star^2} \int_{\{C_k\}} \frac{1}{2}(z_1\,{\rm d}z_2 - z_2\,{\rm d}z_1) \quad , \tag{3.19}$$

where $\{C_k\}$ is a set of all $k = 1, \ldots, k$ closed contours comprising the images. When computing the above integral for each of the K images, we also have to take into account the *parity* of each image $p_k = \pm 1$ which is defined to be equal to the sign of the Jacobian determinant of the lens mapping. The expanded form of Equation 3.19 is then

$$A_{\rm tot}\left(w_0, \rho_\star\right) = \frac{1}{\pi\rho_\star^2} \sum_{k=1}^{K} p_k \int_{C_k} \frac{1}{2}(z_1\,{\rm d}z_2 - z_2\,{\rm d}z_1) \quad , \tag{3.20}$$

Thus, to compute the magnification of an extended uniform source at an arbitrary position w_0 via Equation 3.20, we need to obtain the complex contour points (the point source images) $z_k^{(n)}$, where $n_1 \ldots n_{\rm Nlimb}$ indexes the points along the k-th contour.

To illustrate these contours, we first consider the single lens case where we can simply solve for the two-point source images by evaluating Equation 3.3 at successive points on the source limb:

$$w^{(n)} = w_0 + \rho_\star e^{i\phi_n} \quad , \tag{3.21}$$

where ϕ_n is the discretisation of the angle $\phi \in [-\pi, \pi)$. A closed contour specified by complex points $w^{(n)}$ in the source plane w maps to two closed contours in the image plane $z_k^{(n)}$ $(k = 1, 2)$. These contours are shown in Figure 3.1 for uniform sampling around the source limb ($\phi_n = -\pi + n\Delta\phi$). The density plot (black arcs) shows the two extended images, and the small inset plot shows the source plane w and the circular disc of the source (black circle) overlaid over the point source magnification map. The source is positioned such that its limb is very close to the infinitesimal caustic point. The two complex contours $z_k^{(n)}$ (orange and blue lines) are displayed using connected dots and the directionality of the points is encoded using the marker size such that the first point in each contour, $z_k^{(0)}$, has the largest marker size. The marker size decreases linearly with index n. Thus, the orange contour is oriented in a clockwise direction and the blue contour is oriented in a counter-clockwise direction.

In addition to the location of each contour point we also store the parity of each point which is given by the sign of the determinant $\det \mathbf{J}$ (Equation 2.73). All points within a contour share the same sign of the determinant so we define the contour parity p_k to be the sign of the determinant of the first point in each contour $z_k^{(0)}$. In the case shown in Figure 3.1, the orange contour is oriented in a clockwise direction meaning that it should acquire a negative sign in Equation 3.20, which combined with the negative parity value p_2 implies that the total magnification is proportional to the sum of the two areas, as it should be.

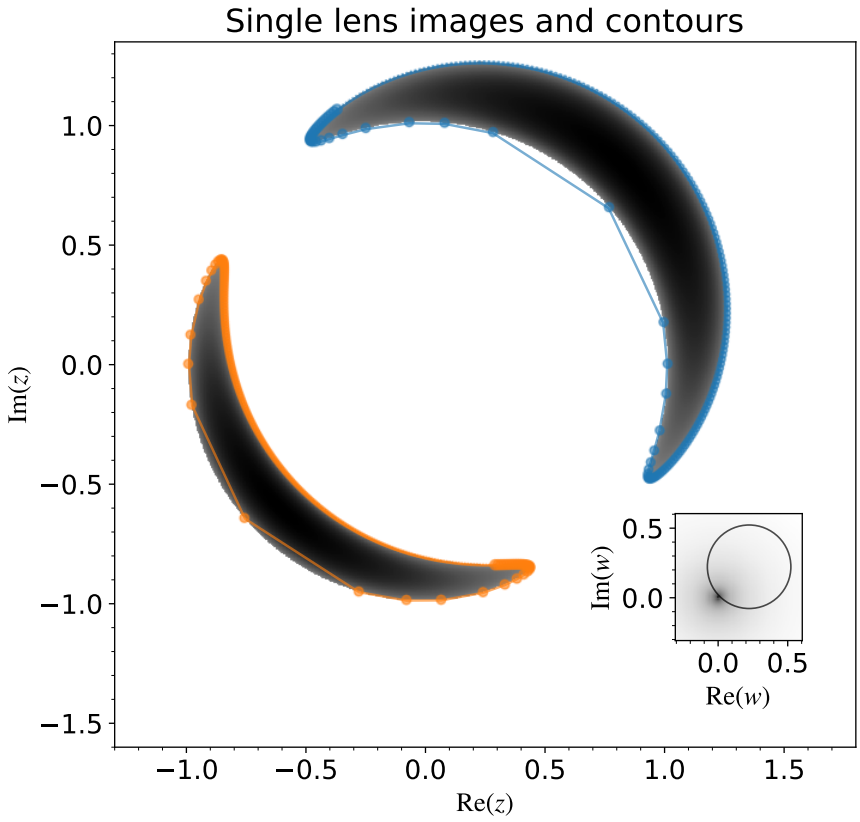


Figure 3.1: single lens images and closed contours (orange and blue curves) obtained by solving the lens equation at uniformly distributed points around the source limb. The ordering of the contour points is displayed using progressively smaller circles starting with the first one. The colour indicates the parity of the extended image (1 for blue, -1 for orange). The orange contour starts close to the point (0.3, -0.7) and continues in a clockwise direction back to the starting point. The blue contour starts close to the point (-0.7, 0.7) and continues in a counter-clockwise direction back to its starting point. The density map shows the (limb-darkened) source intensity evaluated on a fine grid in the image plane. The small inset plot shows the (point source) magnification map in the source plane and the limb of the source disc (black circle).

3.4.2 Adaptive sampling along the source limb

Looking at the contours shown in Figure 3.1 we see that a uniform distribution of points on the source limb at which we evaluated the images does not map to a uniform distribution of points along the contours in the image plane. Parts of the contours with sparsely distributed points correspond to points on the source limb located close to the caustic. A uniform sampling strategy leads to prohibitively inaccurate contour integration when the source is close to, or is intersecting, the caustic. Three approaches to solving this problem have been used in the literature. The first, used in the VBBinaryLensing code, is described in Bozza (2010). They construct error estimators which make use of the derivatives of the integrand in the Green's integral for a uniform brightness source (Equation 3.20). The algorithm starts with only two points on the source limb and then iteratively adds more points until the total error drops below a certain threshold.

There are two reasons why I did not follow this approach. First, the error estimators are derived only for the case of a uniform brightness source. It is not clear how one could extend this approach to the complete Green's integral (see Section 3.5.2) because the complete integral includes an arbitrary source brightness distribution function which has divergent

derivatives at the limb (because of limb darkening). Bozza (2010) avoids this problem by repeating the uniform-brightness calculation at multiple annuli with decreasing radius within the disc of the source, and then weighting the results by the limb-darkened brightness distribution. This approach does not generalise to non-azimuthally symmetric brightness profiles. The second reason is that the Bozza (2010) method does not work for triple (or higher-order) lensing.

The second approach to choosing where along the limb to evaluate the lens equation is the one used by Kuang et al. (2021). They start with an initial uniform sampling along the limb (256 points) and then add more points in regions where the point source magnification is large (though they do not mention exactly how they allocate these additional points) until the fractional change in the final extended source magnification ends up below a specified threshold. The method works for triple-lenses and it does not rely on complicated error estimators. Finally, the third option, suggested by Gould and Gaucherel (1997), is to allocate points on the limb such that the point source images in the image plane are approximately uniformly distributed along the contours.

I have implemented variations of both of the latter two approaches and found that allocating points along the limb based on the distances between the points in the image plane is far superior to allocating points based only on the the point source magnification at each point. This makes sense because the distribution of points in the image plane is what ultimately determines the final accuracy of the contour integral.

Adaptive sampling algorithm

After lots of experimentation, I have arrived at the following algorithm for sampling points along the source limb. The algorithm allocates $N_{\rm limb}$ points in total such that half of those points are distributed uniformly along the limb. The rest of the points are then distributed at locations (in the source plane) where the distance between consecutive points in the image plane is large. The input to the algorithm, in addition to $N_{\rm limb}$, is the location of the centre of the source w_0, and the source radius $\rho_\star$. There as an additional parameter $N_{\rm iter}$, which I will describe below. It is fixed to 10 by default.

The output of the algorithm is a two-dimensional array with shape $(K, N_{\rm limb})$ containing the images, where K is the number of images ($K = 5$ for binary lensing and $K = 10$ for triple lensing), a binary mask indicating which of the images satisfy the lens equation, and an integer array indicating the parity of each image. The threshold value used to differentiate between real and false images is set to 10^{-6}. Kuang et al. (2021) used a value of 10^{-5} but I found that this value in rare cases leads to false positives[16]. The final algorithm is the following:

1. Uniformly sample $\lceil \frac{1}{2} N_{\rm limb} \rceil$ (where $\lceil\ \rceil$ denotes the ceiling function) points at locations $w^{(n)} = w_0 + \rho_\star e^{i\phi_n}$ on the limb of the source disc in the interval $[-\pi, \pi)$. Evaluate K point source images, the parity, and a binary mask indicating which images satisfy the lens equation. The images are evaluated sequentially such that at iteration n, we use the images obtained in iteration $n-1$ to initialise the root solver. The result of

[16]False positives are worse than false negatives because we can end up with contour segments (see Section 3.4.3) which are crooked at their endpoints because the point source images are misclassified as real.

this procedure is an array with a shape $(K, \lceil \frac{1}{2} N_{\rm limb} \rceil)$. This initialisation strategy also ensures that consecutive images $(z_k^{(1)}, z_k^{(2)}, z_k^{(3)}, \ldots)$ in a given row k are close to each other in the complex plane.

2. Repeat for a total of $N_{\rm iter}$ iterations:

   (a) Compute distances between all consecutive images $\Delta z_k^{(n)}$, defined as

   $$\Delta z_k^{(n)} \equiv \begin{cases} 0\,, & \text{if} \quad f\left(z_k^{(n)}\right) \neq 0 \quad \text{or} \quad f\left(z_k^{(n+1)}\right) \neq 0 \\ |z_k^{(n+1)} - z_k^{(n)}|\,, & \text{otherwise} \end{cases}\,, \tag{3.22}$$

   where $f()$ is a function which evaluates to 0 if an image satisfies the lens equation. Compute the maximum $\Delta z_k^{(n)}$ in each column, $\max_k \Delta z_k^{(n)}$, and sort the values in descending order.

   (b) Distribute $\lceil N_{\rm remain}/N_{\rm iter} \rceil$ points at the midpoints of the intervals in ϕ ($\frac{1}{2}(\phi^{(n)} + \phi^{(n+1)})$), which correspond to the largest values of $\Delta z_k^{(n)}$ obtained in the previous step. Evaluate a new set of images at these points.

3. Iterate over columns of the matrix obtained in the previous step. At each step n permute the ordering of the images in the column such that each image $z_k^{(n-1)}$ from the previous column is matched with the image $z_{k'}^{(n)}$ that is closest to it, starting with $z_1^{(n-1)}$.

The reasoning behind this algorithm is the following. First, we allocate a total number of points $N_{\rm limb}$ which effectively sets the accuracy of the final contour integral. In step 1. we obtain an initial set of point source images, uniformly distributed along the limb in the source plane, but not uniformly distributed in the image plane. The output of this step is an array with shape $(K, \lceil \frac{1}{2} N_{\rm limb} \rceil)$ indexed by indices k and n. Due to the sequential initialisation strategy for the root solver, each set of K images ($K = 5$ for binary lensing and $K = 10$ for triple-lensing) will be ordered such that consecutive images with the same index k follow a trajectory in the image plane. In case there are no caustic crossings along the limb, the trajectory will consist of either all real or all false images which all share the same parity. It will form a closed contour unless the initial density of the points in the image plane was so sparse that the images ended up being connected in the wrong order. If there are caustic crossings, then a given row k can have both real and false images. The idea to treat both false and real images on equal footing at this stage is due to Kuang et al. (2021). In contrast, Bozza (2010) ignore the false images and instead directly connects real images in each row k at non-consecutive indices n. This is possible in the binary lens case but not in the triple-lens case.

The outcome of step 1 is visualised in Figure 3.2 for a triple-lens system with parameters $a = 0.698$, $z_3 = -0.0197 - i0.95087$, $\epsilon_1 = 0.02809$, $\epsilon_2 = 0.9687$, $\rho_\star = 0.05$, and $N_{\rm limb} = 200$. Colours indicate the different rows k. Real images are depicted with solid markers and false images with cross markers. The critical curve is shown as a solid black line. Linearly decreasing size of the markers encodes the ordering of the points along the limb. Let's focus on the points in row $k = 2$ (orange). The sequence of points starts around the point (-1.5,

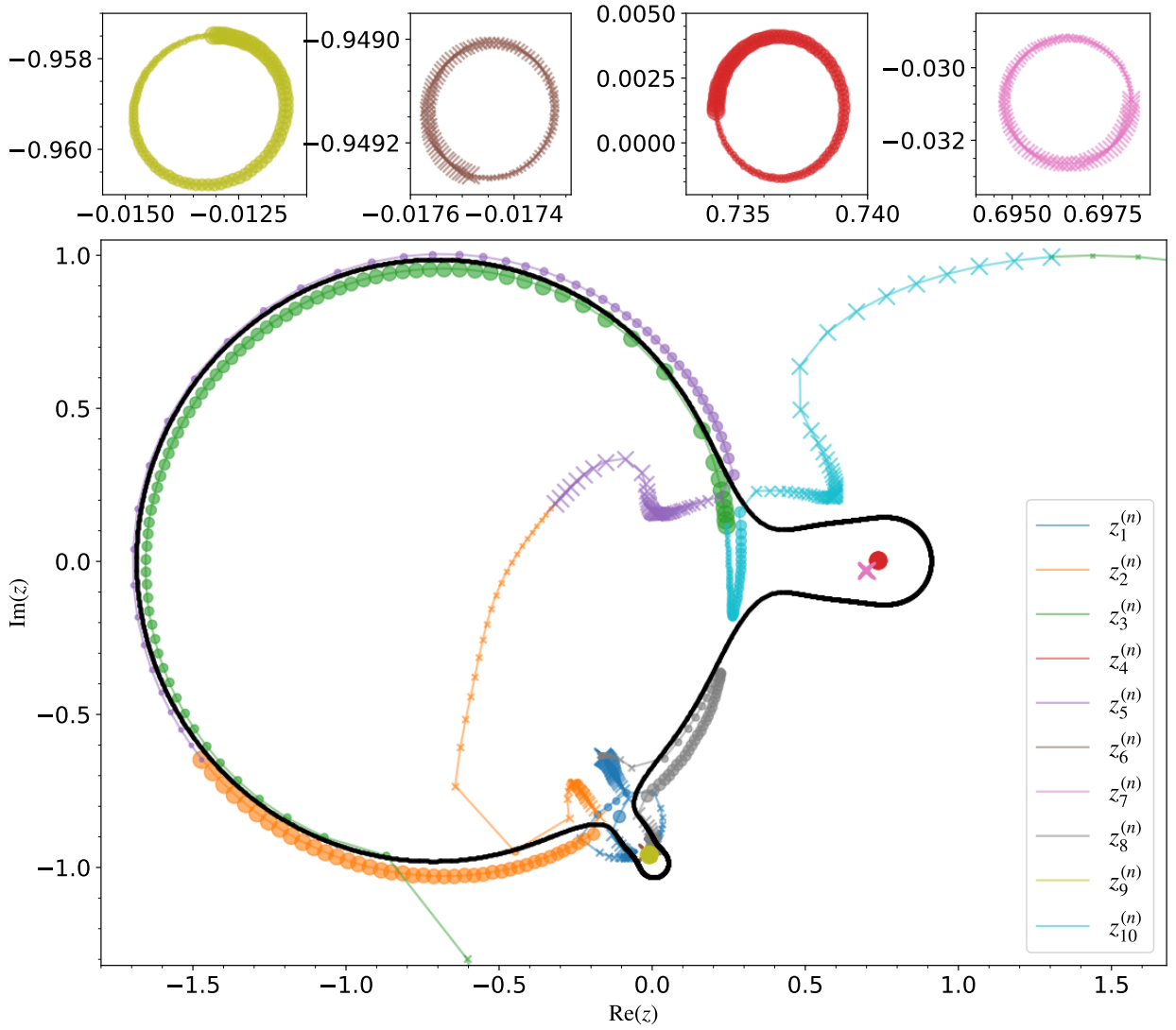


Figure 3.2: Intial step of the adaptive sampling algorithm showing the point source images which correspond to uniformly distributed points along the limb of the source (bottom panel). The images were computed for a triple-lens system with parameters $a = 0.698$, $z_3 = -0.0197 - i0.95087$, $\epsilon_1 = 0.02809$, $\epsilon_2 = 0.9687$ and $\rho_\star = 0.05$. Each colour corresponds to the point source images from a single row of an array with shape $(N_{\text{images}}, N_{\text{limb}}/2)$. This array was obtained by solving for the point source images along the limb in sequence, as described in Section 3.4.2. The marker size encodes the ordering of the points (it decreases linearly with index n). Circles indicate real images while crosses indicate false images. The solid black line is the critical curve, and the mini plots on the top zoom in on hard to see contours.

-0.6) and consists of real images. It then crosses the critical curve and continues as a sequence of false images until it meets with the purple track at the end. The trajectory of the blue curve ($k = 1$) is far more complicated because it crosses the critical curve four times in total. The partition of points into different rows shown here depends on the initialisation of the root solver and the spacing between points along the limb so it is not unique. The images shown in the mini plots at the top are already closed. They consist of either all real or all false images. These images are unperturbed by the caustic crossing.

Having distributed half of the total number of allocated points uniformly in order to obtain the first distribution of point source images in the image plane, the goal for the

subsequent steps is to fill in the large gaps between consecutive points which mostly appear near critical curve crossings. The natural approach would be to add one additional point at a time in between consecutive points in the image plane, starting with the largest gaps. For reasons that have to do with the computational efficiency of JAX loops where each iteration is dependent on the outcome of the previous one, I found that it is far better to allocate points in batches instead[17].

Allocating the same number of points ($\frac{1}{2}N_{\mathrm{limb}}/N_{\mathrm{iter}}$) at each iteration with $N_{\mathrm{iter}} = 10$ works well as a default setting. The points are allocated to the midpoints of the intervals in the angular index n by first taking a maximum over the rows k and then sorting all intervals by length. Before the sorting operation, I zero out intervals where both images are false because we do not care about the density of points in sequences consisting of just false images. Note that evaluating the new images at the midpoint of the intervals in ϕ will not necessarily result in a point that ends up at the midpoint of the line connecting two consecutive images in the image plane.

This outcome of step 2. is shown in Figure 3.3. The top left panel shows the point source magnification map and the outline of the source limb (grey circle). The triple-lens caustic structure is highly complex, there are intersecting caustics, and the limb crosses multiple caustics. The bottom panel shows the image plane with the initial uniformly sampled images (black points) and the additional images (orange points). The algorithm works really well and the additional points are placed exactly where they are needed. For example, there are no black points at all near the critical curve around $\mathrm{Im}(z) = -0.9$ and the orange points perfectly fill that gap. The reason why these gaps existed in the first place becomes clear when we look at the point source magnification along the source limb (top right panel). The initial uniform sampling completely misses the fine structure close to the caustics. Targeting the gaps between consecutive images in the image plane ensures dense sampling of the magnification peaks in the source plane. Importantly, unless we use a very small initial number of points (less than a hundred or so), the outcome of this procedure is not sensitive to the particular arrangement of images in different rows of the array which contains the point source images.

The final step (step 5.) is necessary to ensure the correct ordering of the images in each column such that the distance between consecutive points in each row is small. In the previous steps, we did not do any explicit looping over all images in each column to ensure that the images are sorted correctly. I found that an explicit reordering operation was unnecessary at that point because the sequential initialisation of the root solver leads to an ordering that is good enough for the purpose of adding additional points in batches later on. The reordering step is necessary once we have obtained all the points, because for the purpose of creating contour segments (see the following section), we need to ensure that the point source images are sorted into K rows as best as possible.

Since there are $K!$ possible permutations of the $n+1$-th column of images that need to be matched with the n-th column of images, how can we find the appropriate permutation?

[17]To be specific, the latter approach involves writing a simple Python for loop with a small number of iterations which ultimately gets unrolled by XLA. The former approach requires the use of lax.scan and getting around the limit on variable size arrays by padding the arrays with zeros and using jax.numpy.insert to insert new points in between existing points. For reasons I do not fully understand using a classic for loop with a small number of iterations ends up being about an order of magnitude faster.

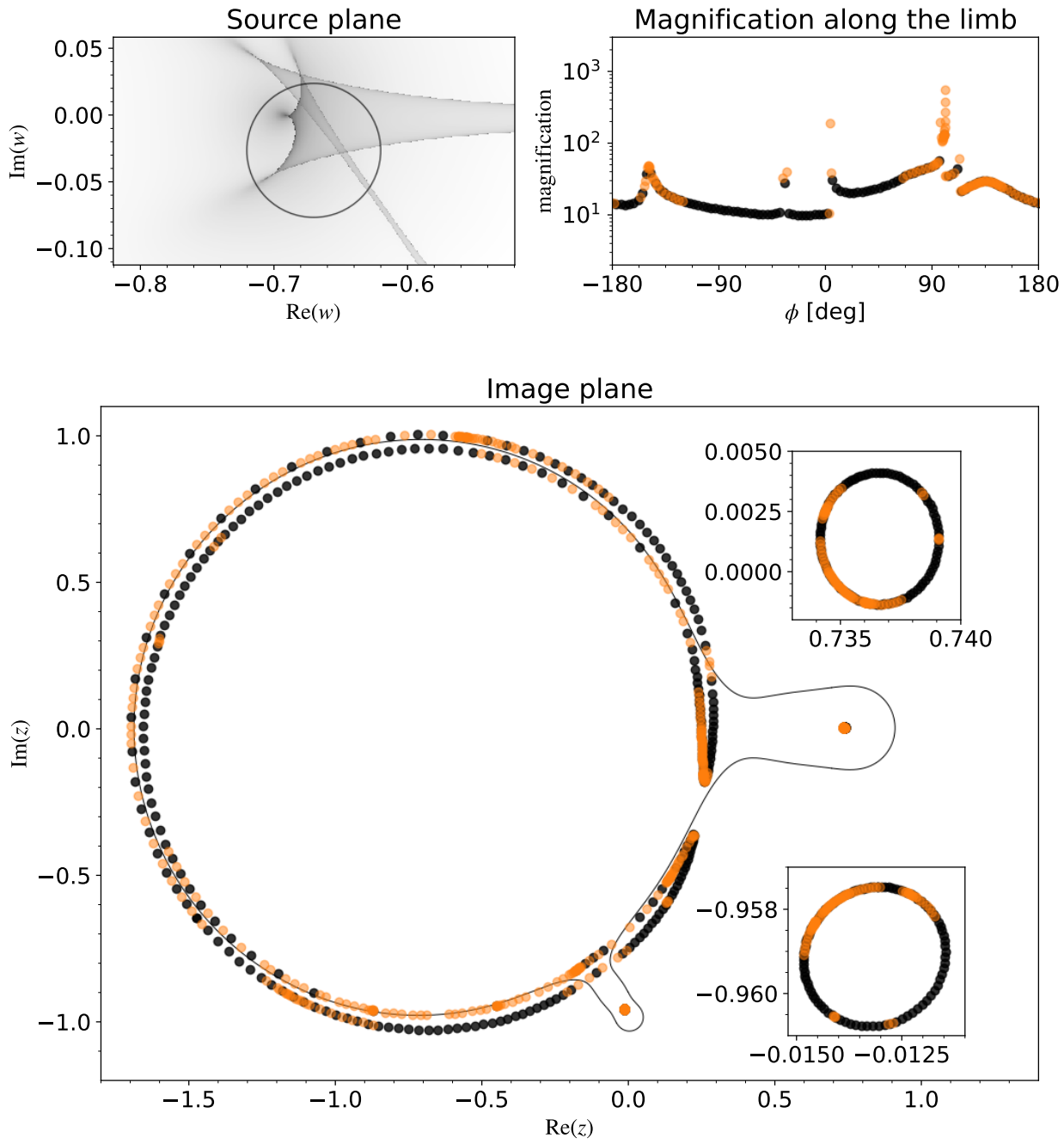


Figure 3.3: Counterpart to Figure 3.2 showing the outcome of the adaptive sampling algorithm for the same triple-lens system. Only real images are shown without reference to the ordering of the points. The top left panel shows the point source magnification map and the outline of the source limb (grey circle) which crosses multiple (intersecting) caustics. The top right panel shows the point source magnification at points evaluated along the limb. Black points correspond to initial uniform sampling and orange, points are additional points added such that they fill in the gaps between consecutive images in the image plane (bottom panel). The two inset plots zoom in on the hard-to-see regions. The new points are placed where they are most needed.

One option would be to require a solution which minimises the sum of all distances between consecutive images, $\sum_k \Delta z_k^{(n)}$. For a binary lens, a brute-force search would require trying out $5! = 120$ possible permutations at each index n but this brute-force approach would fail

for triple-lensing because 10! is a very large number. It turns out that this matching problem is a well-known problem in combinatorial optimisation, called the *linear sum assignment problem*. There are better algorithms for solving it than brute-force search, most notably, the so-called *Hungarian algorithm*. However, implementing and executing this algorithm is complicated, and computationally expensive. It is also unnecessary because there is no guarantee that the solution which optimizes the sum of pairwise distances is necessarily correct in the sense that we would proveably avoid connections between two segments belonging to different contours. Instead, I simply iterate over each image in the n-th column in order, and then match that image with the closest image in the $n-1$-th column. The resulting permutation of the n-th column will in general not be equivalent to the solution of the linear sum assignment problem, but it is good enough for what follows. The same approach seems to be used by Kuang et al. (2021).

3.4.3 Constructing the contours of the images

Having obtained a $(K, N_{\rm limb})$ array containing the points, the next step, following Kuang et al. (2021), is to iterate over each row and split it into some number (which is different for each row) of disjoint *segments* such that the each segment has the following properties:

1. Each segment consists of only contiguous real images with the same parity.
2. There are no discontinuities in the distance between consecutive points in the segment above a distance threshold of 0.1.

The first condition means that if a given row of the image array consists of multiple "bundles" of contiguous real images, with false images in between each bundle, then each bundle is stored as a separate segment. The second condition is necessary because in rare cases the matching between consecutive columns of images described in the previous section fails and there is a discontinuous jump from one contour to another at some index n without any false images in between. Requiring that all consecutive point source images within a segment are relatively close to each other solves this problem. The splitting of sequences of images into segments is achieved by iterating over the angular index n and keeping track of changes in the binary masks indicating image type (real vs. false), parity, and distance between consecutive points. Each row is then split into multiple segments at the recorded change points. I discard any segments consisting of fewer than 3 images.

The total number of segments will in general be greater than K. Since these segments have variable length, Kuang et al. (2021) store them in a *linked list* data structure. Because XLA imposes a strict requirement that all arrays have a statically determined shape at compile time, we cannot use list constructs. Instead, I use fixed-size arrays padded with zeros to store the segments.

The final segments for the triple-lens system discussed previously are shown in the top panel of Figure 3.4. Each colour indicates a different segment. The direction of the segment is encoded using the size of the markers. The parity of the segments is not shown. Each segment has a *head* (the first point in the segment) and a *tail* (the last point in the segment). There are two segments (olive and teal-colored points) which already form closed contours, these do not require extra processing. The rest of the segments are different parts of a single

closed contour which need to be connected together in a way that respects their directionality and parity.

Contour construction algorithm

The segments are connected two at a time and there are four kinds of connections that can be made:

1. **Tail-Head connection (T-H)**: The tail of the first segment is connected to the head of the second segment.
2. **Head-Tail connection (H-T)**: The head of the first segment is connected to the tail of the second segment.
3. **Head-Head connection (H-H)**: The head of the first segment is connected to the head of the second segment.
4. **Tail-Tail connection (T-T)**: The tail of the first segment is connected to the tail of the second segment.

T-H and H-T connections can only be made if the segments have the same parity, and H-H and T-T connections can be made if two segments have opposite parity (these segments initially connect across a critical curve). I will use the symbol $\mathbf{s}_j = (z_j^{(1)}, z_j^{(2)}, \ldots z_j^{(n_j)})$ to denote the j-th segment, consisting of n_j images. It is very important to ensure that a wrong connection is never made.

After lots of experimentation, I came up with the following algorithm for connecting the segments into closed contours:

1. Evaluate the T-H distance for each segment on its own. The segments for which this distance is less than MIN_DISTare marked as closed and they are not considered further. Sort the remaining segments according to their length (defined as $\sum_n \left| z_j^{(n+1)} - z_j^{(n)} \right|$), in ascending order.
2. Define a function that determines whether two segments $\mathbf{s}_1$ and $\mathbf{s}_2$ should be connected given a specific connection type. This function returns True if both of the following conditions are met:
   (a) The parities of the two segments are the same if the connection type is T-H or H-T, or the opposite if the connection type is H-H or T-T.
   (b) The distance between the connection points of the two segments (a connection point is either the tail or the head of a given segment, depending on the connection type) is less than MIN_DIST, or, the distance is greater than MIN_DIST and the following two conditions are satisfied:
      i. The distance between the connection point of $\mathbf{s}_1$ and the connection point of $\mathbf{s}_2$ is less than the distance between the points immediately adjacent to the connection points. This is to ensure that the two segments are not parallel to each other and pointing in the same direction.

ii. The distance between the two endpoints is less than MAX_DIST.

Otherwise the function returns False.

The rules for connecting the segments are the following:

- For a T-H connection we simply concatenate $\mathbf{s}_2$ to the end of $\mathbf{s}_1$ and the opposite for a H-T connection.
- For a H-H or T-T connection, the elements of $\mathbf{s}_2$ are flipped and then concatenated to $\mathbf{s}_1$. The connected segment is assigned the parity of $\mathbf{s}_1$.

3. Repeat for MAX_NR_OF_CONTOURS iterations:
   (a) Pick the shortest segment and label it as the active segment.
   (b) Repeat for MAX_NR_OF_SEGMENTS iterations:
      i. Evaluate the distance between the head/tail of the active segment and the heads/tails of the remaining segments for all four possible connection types. Connect the active segment to the one closest to it for which the connection condition is satisfied. If there are no such segments the active segment is left as is.

The algorithm starts by selecting the shortest-length segment (the active segment). We then sort all the other segments based on the distance from the active segment and evaluate the connection condition for all types of connections. If the distance between the connection points of two segments is less than MIN_DIST, which is set to 10^{-5}, the segments are automatically connected if the parity condition is satisfied. This will only be true for T-H or H-T connections (such as the connections between the pink and the blue and the red and the purple segments shown in the top panel of Figure 3.4). If that condition is not satisfied then we are dealing with a connection over a critical curve (see the connection between the red and the blue and the orange and the purple segments in Figure 3.4). In this case, we have to be careful to avoid making a wrong connection, so we make sure that the segments are pointing towards each other and that the distance between the connection points is less than some large value MAX_DIST, which I set to 0.05.

We then merge the first two segments and repeat the process for a fixed number of iterations MAX_NR_OF_SEGMENTS which is set to 3 times the number of images, a number chosen to be large enough that by the last iteration all segments comprising the same contour are merged together. We repeat the process again by sorting the segments by length, selecting the shortest segment and iterating for another MAX_NR_OF_SEGMENTS iterations. This ensures that if there are multiple contours comprised of open segments, all segments will be merged. I set the value of MAX_NR_OF_CONTOURS to 3. These two loops are fixed size because of XLA constraints on dynamic arrays. However, to avoid unnecessary computation, at each iteration we check if there are any more segments left. The whole procedure is not very computationally expensive because we are mostly just rearranging and copying arrays containing the point source images.

The final outcome of this algorithm is shown in the bottom panel of Figure 3.4. The colour indicates the parity of each contour. In this particular case, all contours have the same

parity and they are all oriented in a clockwise direction as we would expect because there are no nested contours. To ensure the correctness of the algorithm I implemented extensive tests by running the algorithm at multiple caustic-crossing positions of the source star for triple and binary lenses.

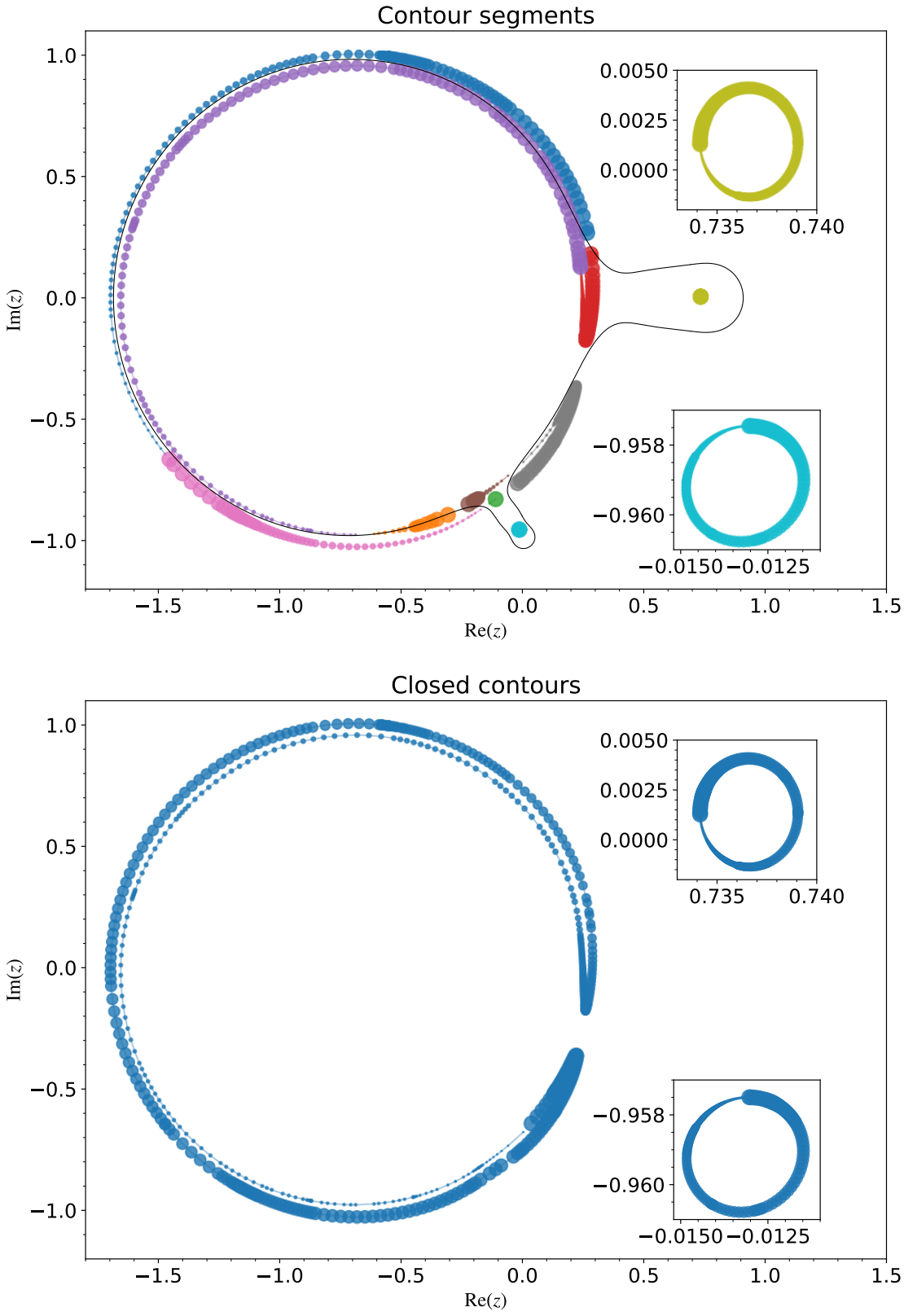


Figure 3.4: Counterpart to Figures 3.2 and 3.3 showing the process of turning contour segments (top panel) into closed contours (bottom panel). The colours in the top panel indicate different segments. As before, the marker size encodes the direction of the segment. The parity of each segment is not shown. The black line in the top panel is the critical curve. The colour in the bottom panel indicates the parity of the contours (positive or negative). In this case all contours have the same parity because there are no nested contours.

3.5 Integration

3.5.1 Uniform brightness source

In the past two sections, I have described the algorithms for computing the contours of extended images which I have implemented in the caustics code. The remaining task is to numerically solve the contour integral in Green's theorem (Equation 3.18). I will start with the case of a uniform brightness source because it much more straightforward and computationally easier problem to solve than the case of a limb-darkened source. The uniform brightness source model is useful for non caustic-crossing events and events with sparse sampling during the caustic-crossing when the limb-darkening effect is negligible. Limb-darkening is much more important for caustic-crossing events with larger source radii because the probability of having good data coverage at caustic crossings is much larger.

To compute the magnification of a uniform-brightness source, we simply have to evaluate the 1D integrals in Equation 3.19, one over the real component of z and the other over the imaginary component of z. I use the (composite) *trapezoidal rule* which is part of a family of numerical integration formulas called *Newton-Cotes formulas*. Applying the composite trapezoidal rule to the integrals in Equation 3.19 yields

$$A_{\rm tot}\left(w_0,\rho_\star\right) \approx \frac{1}{\pi\rho_\star^2}\frac{1}{2}\sum_{i=1}^{n}\left[\frac{1}{2}\left(z_1^{(i)}+z_1^{(i+1)}\right)\Delta z_2^{(i)} - \frac{1}{2}\left(z_2^{(i)}+z_2^{(i+1)}\right)\Delta z_1^{(i)}\right] \quad , \tag{3.23}$$

where $\Delta z_1^{(i)} = z_1^{(i+1)} - z_1^{(i)}$ and $\Delta z_2^{(i)} = z_2^{(i+1)} - z_2^{(i)}$.

There are two notable alternatives to this choice which converge at a faster rate. The first is *Simpson's rule* (also a Newton-Cotes formula) and the second is using formulas which make use of the derivatives of the integrands evaluated at the integration points such as the "parabolic correction" implemented by Bozza (2010). I did not use Simpson's rule because it involves evaluating the integrand at exactly the midpoint of each integration sub-interval and we cannot choose where we evaluate the contour points because they are the product of the procedures described in the previous sections. We could of course try to find the midpoint of each sub-interval by evaluating the lens mapping in the image plane to find the exact location of the image boundary at the midpoints in a manner similar to what Bennett (2010) did. I have implemented this method and found that it is very error-prone and computationally expensive. It is far better to simply use the trapezoidal rule with a greater number of images evaluated along the source limb. Using the derivatives at the two endpoints of each sub-interval does work better, but, as mentioned previously, it does not easily generalise to the limb-darkening case and I found that the additional complexity involved in the implementation is not warranted, at least in the context of the caustics code.

3.5.2 Limb-darkened source

The limb-darkening case is more complicated. The brightness profile of the source star with linear limb darkening is

$$I_s(r(z)) = \begin{cases} \frac{3}{3-u_1}\left[1-u_1\left(1-\sqrt{1-r^2}\right)\right] & \text{if} \quad r \leq \rho_\star \\ 0 & \text{otherwise} \end{cases} \quad , \tag{3.24}$$

where $r \equiv |w(z) - w_0|$ is the distance from the source centre, and $\frac{3}{3-u_1}$ is a normalisation factor. The linear limb-darkening coefficient u_1 is a real number defined on the interval $[0, 1]$. Setting $u_1 = 0$ reduces Equation 3.24 to the uniform brightness case.

Following Equation 3.18, we are interested in finding functions P and Q which satisfy the condition

$$\left(\frac{\partial Q}{\partial z_1} - \frac{\partial P}{\partial z_2}\right) = I_s(z_1, z_2) \quad . \tag{3.25}$$

Functions which satisfy this condition are given by (Dominik, 1998c):

$$P\left(z_1, z_2\right) = -\frac{1}{2} \int_{z_2^{(0)}}^{z_2} I_s\left(z_1, z_2'\right) \, \mathrm{d}z_2' \tag{3.26}$$

$$Q\left(z_1, z_2\right) = \frac{1}{2} \int_{z_1^{(0)}}^{z_1} I_s\left(z_1', z_2\right) \, \mathrm{d}z_1' \quad , \tag{3.27}$$

where the lower integration bound $z^{(0)} = (z_1^{(0)}, z_2^{(0)})$ is some freely chosen point in the image plane. The total magnification is thus

$$A_{\mathrm{tot}}\left(w_0, \rho_\star\right) = \frac{1}{\pi \rho_\star^2} \sum_{k=1}^{K} p_k \int_{C_k} \left[P(z_1', z_2) \, \mathrm{d}z_1' + Q(z_1, z_2') \, \mathrm{d}z_2'\right] \quad . \tag{3.28}$$

Evaluating the magnification of a limb-darkened source is much more computationally demanding than the uniform brightness case because functions P and Q are themselves (one-dimensional) integrals which need to be estimated numerically at the location of each contour point.

The surface brightness function $I_s(z_1, z_2)$, as defined in Equation 3.24, is problematic because the function and its first derivative with respect r are discontinuous at the location of the source boundary in the image plane. This makes the function difficult to integrate numerically. Dominik (1998c, 2007) get around this problem by using analytic continuation to improve the numerical stability of the integrand. They redefine $I_s(r(z))$ as

$$I_s(r(z)) = C(u) \, \left[1 - u_1 \left(2 - B(r)\right)\right] \quad , \tag{3.29}$$

where $C(u_1) \equiv \frac{3}{3-u}$ and

$$B(r) = \begin{cases} 1 + \sqrt{1 - r^2} & \text{for } 0 \leq r < 1 \\ 1 & \text{for } r = 1 \\ 1 - \sqrt{1 - 1/r^2} & \text{for } r > 1 \end{cases} \quad . \tag{3.30}$$

This definition is identical to the one in Equation 3.24 except for $r > 1$.

The (revised) surface brightness function and the integrands of the P and Q integrals for a single lens image are shown in Figure 3.5. The extended image of a single lens with $\rho_\star = 0.05$ located at $w_0 = 1.5$ with a linear limb-darkening coefficient $u_1 = 0.5$ is shown in the panel on the left. The two integrand functions evaluated at a particular contour point are shown in the two panels on the right. Contour points are shown in blue such that the marker size encodes the direction of the contour. The grey dashed lines indicate the

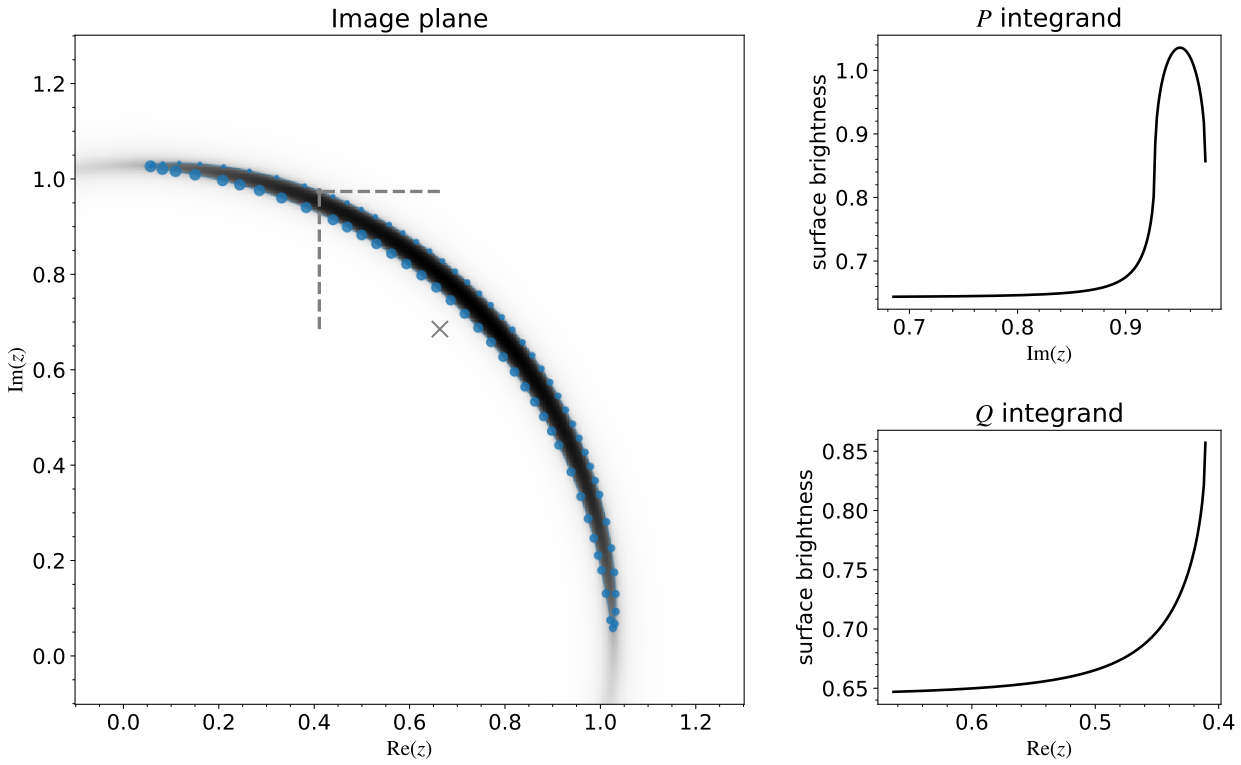


Figure 3.5: Visualisation of the functions defined in Equation 3.27. The left panel shows a single extended image (density plot) for a single lens with $\rho_\star = 0.05$ located at $w_0 = 1.5$ with a linear limb-darkening coefficient of $u_1 = 0.5$. Contour points are shown in blue with decreasing marker size used to indicate the direction of the contour. The grey cross is the geometric centre of the contour. And the two grey dashed lines indicate the integration domains for integrals P and Q for a particular contour point. The integrands are plotted in the two panels on the right.

integration domain for the two integrals. The grey cross is the location of the point $z^{(0)}$ (the lower integration bound) which was chosen to be the geometric centroid of all contour points belonging to the displayed contour.

To evaluate Equation 3.27, we have to numerically integrate the 1D functions shown in Figure 3.5 for every contour point belonging to a given contour. Although the integrands are smooth, the fact that the lower integration point $z^{(0)}$ is often far away from the contour points means that the integrand functions are nearly constant at most locations and they vary appreciably only in a small interval near the limb. This problem is much worse for smaller source sizes because the image shown in Figure 3.5 becomes extremely thin for small values of $\rho_\star$. As a consequence, quadrature rules which use a uniform grid lead to highly inaccurate results when $\rho_\star$ is small.

In these cases, adaptive integration algorithms such as the *Adaptive Simpson's method* are a better choice. However, I found that a scheme using *Gaussian Quadrature* with a fixed number of points works really well as long as we split the integration domain for each integral into two parts. The advantage of using a fixed number of points is that such an approach is easier to implement in JAX and also more computationally efficient[18]. Let's take the P integral as an example. We redefine the integration domain as a union between the intervals

[18] I have implemented an adaptive Simpson integrator in JAX and found it too be comperatively much slower.

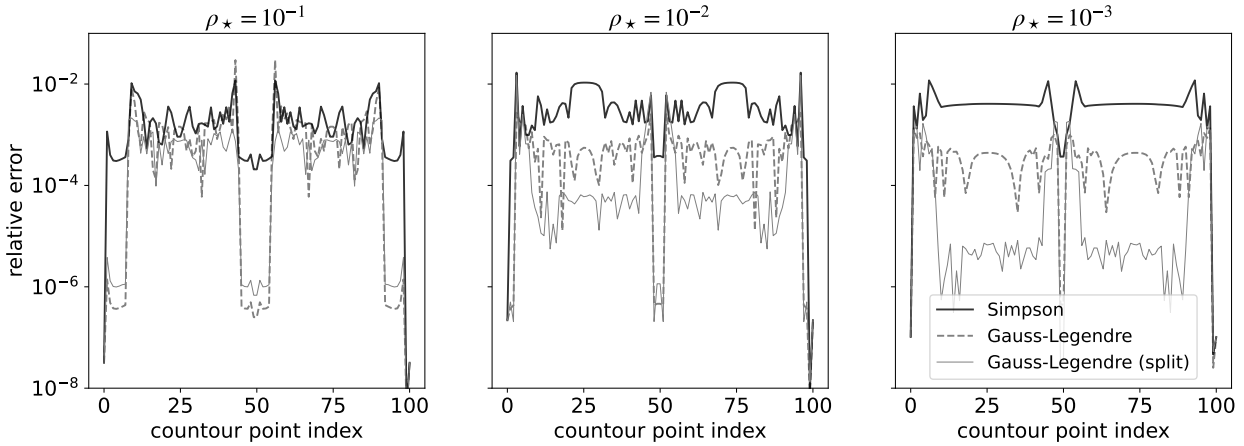


Figure 3.6: Comparison of the relative error for three different fixed-size quadrature rules used to compute the P and Q integrals defined in Equation 3.27. The three panels show the value of the P integral for a single lens located at $w_0 = 0.5$ averaged over the two images, for three different integration methods. The number of points is fixed to 100. Gauss-Legendre quadrature is clearly superior to Simpson's method and splitting the integration intervals using the heuristic defined in Equation 3.31 improves the error for small sources by orders of magnitude.

$[z_2^0, z_{2,\text{split}}]$ and $[z_{2,\text{split}}, z_2]$, where $z_{2,\text{split}}$ is

$$z_{2,\text{split}} = \begin{cases} z_2 - 2\rho_\star & \text{if} \quad z_2 > z_2^{(0)} \quad \text{and} \quad |z_2 - z_2^{(0)}| > 2\rho_\star \\ z_2 + 2\rho_\star & \text{if} \quad z_2 < z_2^{(0)} \quad \text{and} \quad |z_2 - z_2^{(0)}| > 2\rho_\star \\ \frac{1}{2}(z_2^{(0)} + z_2) & \text{otherwise} \end{cases} . \tag{3.31}$$

The second interval always encompasses the region close to the contour and its length is proportional to the source radius $\rho_\star$. The definition of $z_{1,\text{split}}$ for the Q integral is analogous to that of $z_{2,\text{limb}}$. It is possible that we could achieve a similar effect by modifying the function evaluation points and the weights used in the Gaussian quadrature rule but I could not get this to work robustly.

In Figure 3.6 I show the effect of using this procedure on the accuracy of the estimated integrals given a fixed number of points. I also compare the (non-adaptive) Simpson's method to the Gaussian quadrature. To generate the plot I first compute the contour points for a single lens located at $w_0 = 0.5$. I then compute the P integral for every point using an adaptive integrator from the QUADPACK library (available in the SciPy library as the function scipy.integrate.quad) to a relative precision of 10^{-12}. I then compute the same integrals using three different methods with a fixed number of 100 points: the composite Simpson's rule, Gaussian Quadrature and Gaussian Quadrature evaluated at the two sub-intervals defined by Equation 3.31 such that each sub-interval is allocated half of the 100 points. Gauss-Legendre is generally an order of magnitude more accurate than the composite Simpson's method and splitting the integration intervals into two unequal parts improves the error for small sources by multiple orders of magnitude. The results for the Q integral are qualitatively the same.

The drawback of this approach is that we do not obtain an error estimate for the integrals. This is not really a significant issue because the number of points N_{ld} is a simple parameter which is proportional to the true error. I set this parameter to 100 by default because that value is sufficient to keep the final error for the magnification below 10^{-3} in the vast majority

of cases. One could also use an adaptive integrator such as the adaptive Simpson's method or the adaptive *Gauss-Kronrod* method[19]. I have implemented a naive version of the adaptive Simpson's method in JAX and found that it is much slower than the approach described in the previous paragraphs which use fixed-size arrays and no loops. It may be that a higher quality implementation of the adaptive Simpson's method or the adaptive Gauss-Kronrod method (such as the one implemented in the Boost C++ library[20]) would work better, but I have not tested this.

Having described how to compute P and Q functions for every contour point, we can now apply the trapezoidal rule to the contour integral defined in Equation 3.28. The result is given by

$$A_{\mathrm{tot}}\left(w_0, \rho_\star, u_1\right) \approx \frac{1}{\pi\rho_\star^2} \sum_{i=1}^{n} \left[\frac{1}{2} \left(P(z_1^{(i)}, z_2^{(i)}) + P(z_1^{(i)}, z_2^{(i+1)}) \right) \Delta z_2^{(i)} + \frac{1}{2} \left(Q(z_1^{(i)}, z_2^{(i)}) + Q(z_1^{(i+1)}, z_2^{(i)}) \right) \Delta z_1^{(i)} \right] \tag{3.32}$$

Although we have only covered the linear limb-darkening case in this section and only linear limb-darkening is currently implemented in caustics, the approach we have described above easily generalises to an arbitrary source brightness distribution. This is in contrast to the methods used in Bozza (2010) and Kuang et al. (2021), which are only applicable to azimuthally symmetric source brightness profiles. The only thing that requires some modification is the analytic extension defined in Equation 3.30. Dominik (2007) showed how to adapt this equation for an arbitrary azimuthally symmetric source brightness profile but relaxing this assumption should be straightforward. Extending caustics to work with arbitrary source brightness distributions will enable the use of spherical harmonics to model arbitrary two-dimensional surface features such as, for instance, stellar spots.

3.6 Extended source magnification tests

Using Equations 3.23 and 3.32 we can compute the magnification of an extended or limb-darkened source star respectively, at any point in the source plane. In this section, I compare caustics to other codes in terms of accuracy and performance. First, I test the accuracy and validity of caustics by comparing it to VBBinaryLensing, the most widely used microlensing code. To test the extended source magnification calculation for a binary lens, I first initialise a system with parameters $a = 0.45$ and $\epsilon_1 = 0.8$. I then generate 1000 test points located exactly on the caustic curves and perturb them in a random direction by adding a complex number whose radius is uniformly distributed in the interval $[0, 2\rho_\star]$ and whose phase is uniformly distributed in the interval $[0, 2\pi]$. This ensures that at all locations the source limb is either crossing one or multiple caustics, or it is extremely close to the caustic. The test points are generated for 5 different values of $\rho_\star$ spanning several orders of magnitude. I set the number of points on the source limb N_{limb} to 400 and compute the magnification for a uniform brightness source at each point. The relative precision parameter in VBBinaryLensing is set to a value of 10^{-5}.

[19] The adaptive Gauss-Kronrod quadrature is one of the algorithms used in the Fortran QUADPACK library.

[20] https://www.boost.org/doc/libs/master/libs/math/doc/html/math_toolkit/gauss_kronrod.html.

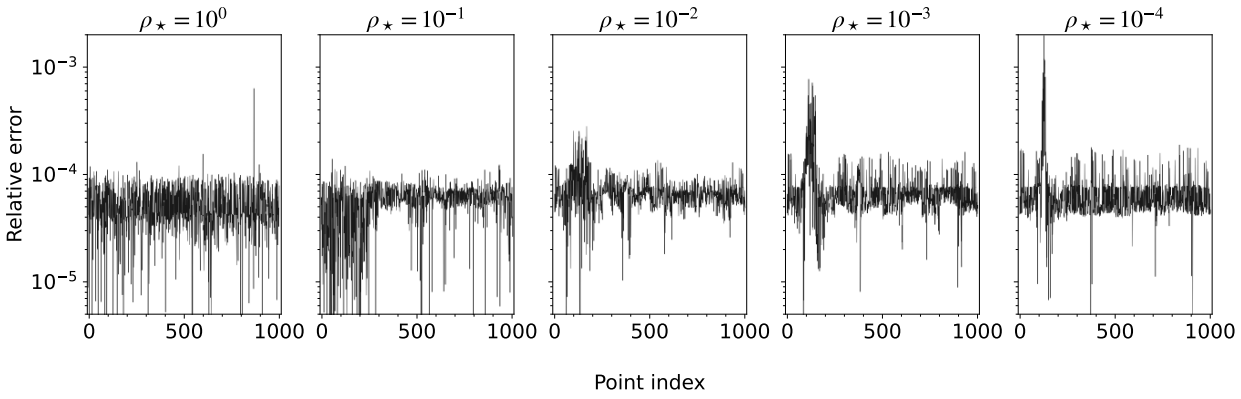


Figure 3.7: Relative error in the magnification between caustics and VBBinaryLensing for a uniform brightness source and a binary lens system with $a = 0.45$ and $\epsilon_1 = 0.8$. The magnification is evaluated at 1000 points drawn randomly such that they are at most $2\rho_\star$ away from the caustics. The number of lens equation evaluations is fixed to $N_{\rm limb} = 400$.

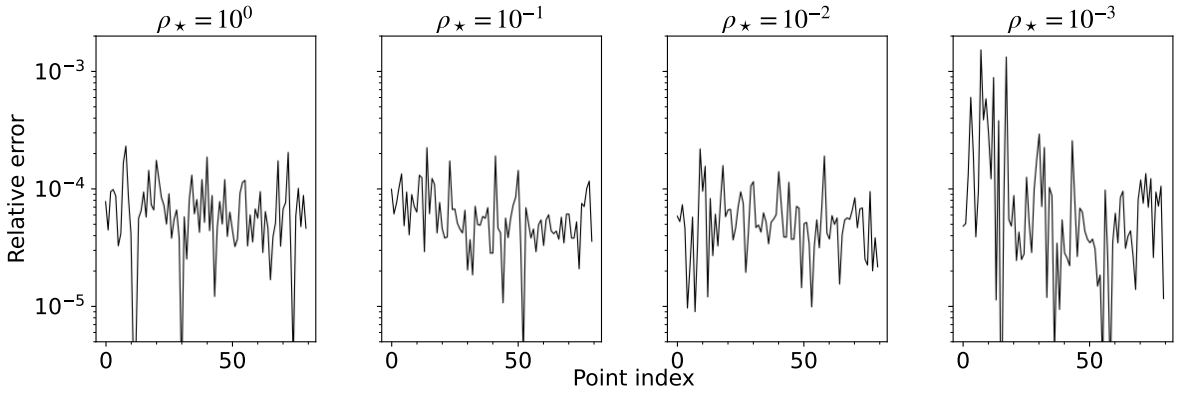


Figure 3.8: Same as Figure 3.7 except the error is computed for a limb-darkened source with a linear limb-darkening coefficient $u_1 = 0.7$. The number of points used to evaluate the P and Q functions is fixed to $N_{\rm ld} = 100$, and $N_{\rm limb} = 400$ as before.

As shown in Figure 3.7, the relative error for this particular choice of $N_{\rm limb}$ is well bellow 10^{-3} at all points, and mostly below 10^{-4} for a wide range of source radii. The accuracy does deteriorate somewhat for very small source radii suggesting that a slightly larger value of $N_{\rm limb}$ may be required to decrease the relative error to 10^{-4}. To test the limb-darkening calculations I repeat the same test for a limb-darkened source star with the linear limb-darkening coefficient $u_1 = 0.7$ and $N_{\rm ld} = 100$. I reduce the number of test points to 100 points and omit the panel with $\rho_\star = 10^{-4}$ because computing the magnification of a limb-darkened source is much more expensive, and because the results are similar to the uniform brightness case (if the P and Q functions are computed correctly the error is still most dependent on the distribution of points along the contours as in the uniform case). The results are shown in Figure 3.8. The error is slightly worse for a given value of $\rho_\star$ than in the uniform brightness case. Increasing $\rm N_{ld}$ would solve this problem. Tests similar to the ones I have just described are executed automatically for the caustics GitHub repository every time there is a change to the code.

To compare caustics to other codes in terms of performance, I start by drawing a dozen points near caustics for a binary lens system with $a = 0.45$ and $\epsilon_1 = 0.8$, as in the uniform-

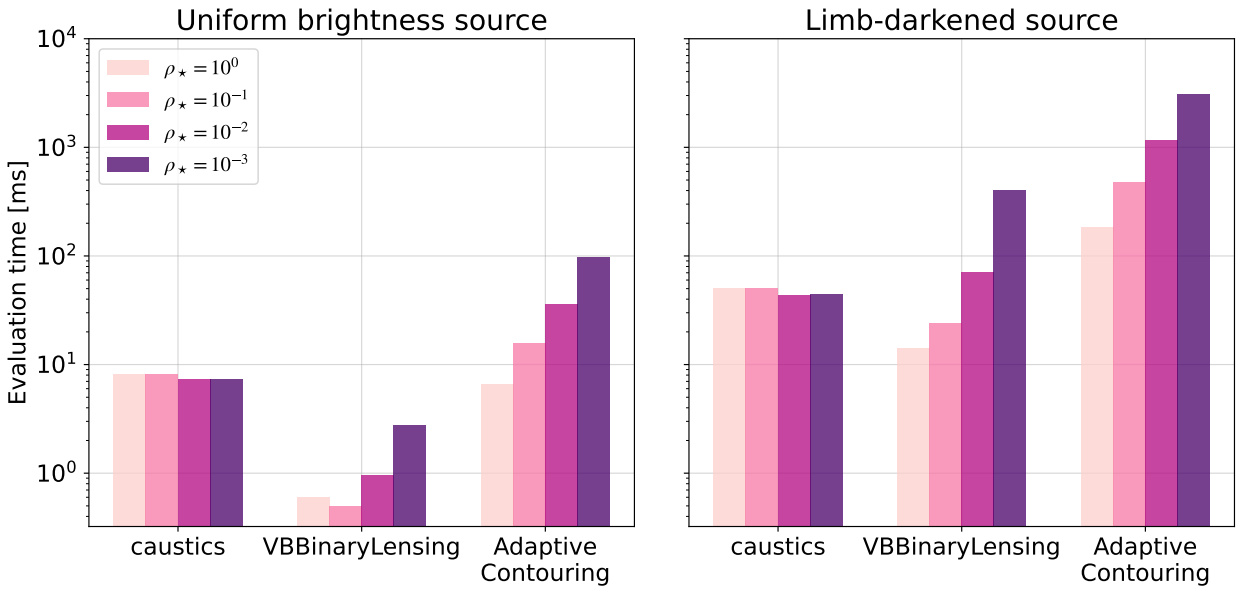


Figure 3.9: Performance comparison between caustics and other microlensing codes. The performance is defined to be the total elapsed wall time when computing the magnification of uniform brightness (left panel) and limb-darkened (right panel) source star at 12 different test points close to caustics, divided by the number of points. The magnification is evaluated for a binary lens system with $a = 0.45$ and $\epsilon_1 = 0.8$ for different values of $\rho_\star$.

brightness test. I then compute the magnification for a uniform brightness source and a limb-darkened source with $u_1 = 0.7$ at each point using caustics, Adaptive Contouring (Dominik, 2007), and VBBinaryLensing. To ensure a fair comparison I tune the parameters of each code such that they all obtain roughly the same average accuracy across the dozen test points. For caustics, I set $N_{\rm limb} = 300$ and $N_{\rm ld} = 100$. For VBBinaryLensing I set the relative precision to 10^{-4}. I compute the magnification with each code in a Python loop over all points and average the execution time over two trials.

This experiment is quite crude and the exact timings should be taken with a grain of salt, but the results shown in Figure 3.9 are nevertheless illuminating. First of all, the computation time for both VBBinaryLensing and Adaptive Contouring increases as the source radius $\rho_\star$ decreases, while it is independent of the $\rho_\star$ for caustics. This is expected because the contours become more elongated for small $\rho_\star$ so VBBinaryLensing and Adaptive Contouring have to allocate more points for smaller sources to obtain a certain precision. caustics on the other hand allocates a fixed number of points, independent of the source radius. This is of course a major drawback because it means caustics uses far too many points when $\rho_\star$ is large so we have to set $N_{\rm limb}$ and $N_{\rm ld}$ to a value large enough to make sure the error is small enough for all values of $\rho_\star$ we expect to see when fitting a particular event.

The second notable feature of the results shown in Figure 3.9 is that for uniform brightness sources, VBBinaryLensing has the best performance. Averaged across $\rho_\star$, it is about ~ 5 times faster than caustics and more than an order of magnitude faster than Adaptive Contouring. This is expected because, as mentioned previously, VBBinaryLensing is perfectly optimized for computing the area of the images with as few lens equation inversions as possible. The situation for a limb-darkened source is quite different. After obtaining the image contours, caustics and Adaptive Contouring both use the same approach to compute the

magnification of a limb-darkened source, while VBBinaryLensing repeats the uniform brightness calculation at different annuli and re-weights the results in proportion to the radial source brightness profile. Averaged over $\rho_\star$, caustics and VBBinaryLensing have about the same computational cost but caustics is an order of magnitude faster for very small source sizes. Adaptive Contouring is more than an order of magnitude slower than both. Overall, computing the magnification of a limb-darkened source star is slightly less than an order of magnitude more expensive relative to a uniform-brightness source for caustics and almost two orders of magnitude more expensive for VBBinaryLensing. The fact that caustics and Adaptive Contouring have roughly the same performance difference for both uniform-brightness and limb-darkened sources suggests the difference in performance is likely due to the less efficient contour construction algorithm in Adaptive Contouring.

What about triple-lensing? Neither VBBinaryLensing nor Adaptive Contouring support triple-lensing. The TripleLens code (Kuang et al., 2021), which caustics was partly based on, does support triple lensing, but as mentioned previously in this section, the code does not really work near caustic crossings, even for binary lens events. This makes it difficult to test the magnification computation for triple lenses. Probably the best solution to this problem would be to build an inverse ray shooting code and use that as a benchmark. Because of time constraints I have not managed to do that. However, since the exact same algorithms and functions are used to compute the magnification for binary and triple-lens systems in caustics, and all these functions are extensively tested, there is no reason to assume the that caustics does not work equally well with triple lenses. Regarding performance, evaluating the extended source magnification for a triple-lens system is only about 2-3x slower than for a binary lens system. For reference, TripleLens is 2-3 orders of magnitude slower than caustics for triple lenses.

In summary, in this section I have shown that caustics works as intended, by comparing it to two other codes at the most problematic locations in the source plane where the source limb is either intersecting or is very close to binary lens caustics. I leave extensive testing of the triple-lens magnification for future work, but since the same functions are used to compute the magnification of binary and triple-lenses, there is no reason to assume that the results for triple lenses are widely different. Regarding performance, caustics is roughly 3-10X slower than VBBinaryLensing for uniform sources and on par with VBBinaryLensing for limb-darkened source, except for the smallest source radii ($\rho_\star \sim 10^{-3}$) when caustics is substantially faster than VBBinaryLensing.

3.7 Computing light curves

Although the computation of the extended source magnification at a single point is relatively fast (on the order of a few milliseconds using the 2021 Macbook Pro laptop), microlensing light curves consist of thousands of data points. This means that a complete evaluation of the likelihood function would take on the order of seconds. This is prohibitively expensive. Fortunately, the vast majority of these points correspond to locations in the source plane that are sufficiently far from the caustics that one can approximate the extended source magnification using either the point source magnification, or the multipole series expansion of the full magnification. To solve this problem, version 2.0 of VBBinaryLensing (Bozza et al.,

2018) switches between the full computation and the point source approximation.

Multipole expansion up to hexadecapole order (Gould, 2008; Cassan, 2017) is very accurate up to a few source radii away from the caustics. Cassan (2017) showed that one can evaluate the hexadecapole approximation for the magnification of a limb-darkened extended source using only a single inversion of the lens equation at the centre of the source. The computational cost of this approximation is only a factor of a few larger than the point source approximation and it is orders of magnitude cheaper than the full contour integration. The challenge is how to cheaply (in terms of computational cost) determine whether the approximate solution is sufficient at any given point in the source plane while making use of only local information.To solve this problem, Bozza et al. (2018) implements a series of tests to determine when a point source approximation is sufficient:

- **Quadrupole test** – detects deviations from the point source magnification approximation using the quadrupole term in the multipole expansion of the (extended source) magnification.
- **Cusp test** – detects proximity to cusps.
- **False images test** – detects when the source is very close to the fold by computing the Taylor expansion of the Jacobian of the lens mapping, and evaluating it at the false images.
- **Planetary test** – detects situations when the source is larger than the planetary caustics and the lens mapping Jacobian is not sensitive to its presence.

In caustics I implemented all of these tests. I describe them in more detail below.

Multipole and cusp test

Following Cassan (2017), the magnification of an extended limb-darkened source can be expanded in a power series for $\rho_\star \ll 1$ as follows:

$$A_{\rm tot}(w_0, \rho_\star, u_1) = \sum_k |\mu_k(w_0, \rho_\star, u_1)| \quad , \tag{3.33}$$

where

$$\mu_k(w, \rho_\star, u_1) = \mu_{0,k}(w, u_1) + \mu_{{\rm quad},k}(w, u_1)\rho_\star^2 + \mu_{{\rm hex},k}(w, u_1)\rho_\star^4 + \mathcal{O}(\rho_\star^6) \quad . \tag{3.34}$$

The index k indicates the k-th point source image and the sum is over all real images. μ_0 is the point source magnification, $\mu_{\rm quad}$ is the quadrupole term and $\mu_{\rm hex}$ is the hexadecapole term.

To detect deviations from the point source magnification, Bozza et al. (2018) requires that the magnitude of the quadrupole term multiplied by some constant is less than a specified absolute error threshold. I modify this condition somewhat by also including the hexadecapole term. The condition is

$$\sum_k \left(|\mu_{{\rm quad},k}| + |\mu_{{\rm hex},k}|\right) < c_m \quad , \tag{3.35}$$

where c_m is some tunable constant and the sum is over the real images.

As pointed out by Cassan (2017), the quadrupole (and also the hexadecapole) term sometimes vanishes in the vicinity of the cusps and the condition from Equation 3.35 fails to trigger the full calculation where it is necessary. To fix this problem Cassan (2017) propose adding a term which detects proximity to a cusp with the following form

$$\mu_{\rm cusp} = \frac{6\,{\rm Im}\left[3(f'(\bar{z}))^3(f''(z))^2\right]}{(J(z))^5}\rho_\star^2 \quad . \tag{3.36}$$

where where $f(z)$ is a term in the lens equation defined such that $w = z + f(\bar{z})$. This term is large when the tangential vector to the caustic vanishes. I then modify the condition in Equation 3.35 to

$$\sum_k \left(\gamma|\mu_{{\rm cusp},k}| + |\mu_{{\rm quad},k}| + |\mu_{{\rm hex},k}|\right) < c_m \quad , \tag{3.37}$$

where γ is a positive tunable constant.

False images test

The condition in Equation 3.37 fails when a particular point is close to a caustic fold. The reason for this is that the point source magnification of the real images corresponding to the centre of the source disc is completely insensitive to a fold approach until the moment the limb of the source limb touches the fold. Although the real images are insensitive to the fold approach, the false images move closer together as the source approaches the fold, they then merge together and finally reappear as real images on the other side of the fold. At the point of the caustic-crossing, the Jacobian determinant of the lens mapping evaluated at the false images vanishes. Therefore, we can make use of the false images to detect proximity to a fold caustic. Bozza et al. (2018) implement a test for this by evaluating the change in the Jacobian determinant $J \equiv |\mathbf{J}|$ of the lens mapping for the false images as we move from the centre of the source disc w_0 to the limb.

Following Bozza et al. (2018), the value of J at the source limb can be expressed as

$$J(z_{\rm limb}) = J(z_0) + \Delta J(z_0, \rho_\star) \quad , \tag{3.38}$$

where z_0 is any of the false images and ΔJ is the change in the Jacobian determinant. To test that we are far enough from a fold it is sufficient to test that $J(z)$ never vanishes on any of the points on the source limb which requires that (see Bozza et al., 2018)

$$J(z_0) > |\max_\phi \Delta J(\rho_\star)| \quad , \tag{3.39}$$

where ϕ is the angle along the limb of the source. Expanded, the condition becomes

$$\frac{1}{2}\left| J\left(z_0\right) \frac{\tilde{J}^2}{\tilde{J} f''\left(\bar{z}_0\right) f'\left(z_0\right) - \bar{\tilde{J}} f''\left(z_0\right) f'\left(\bar{z}_0\right) f'(\tilde{z})} \right| > c_f \rho_\star \quad , \tag{3.40}$$

where the prime denotes the derivatives with respect to z, $\tilde{z} \equiv \bar{w} - f(z_0)$, $\bar{\tilde{z}} \equiv w - f(\bar{z}_0)$, $\tilde{J} \equiv 1 - f'(z_0)f'(\tilde{z})$, and c_f is a positive tunable constant.

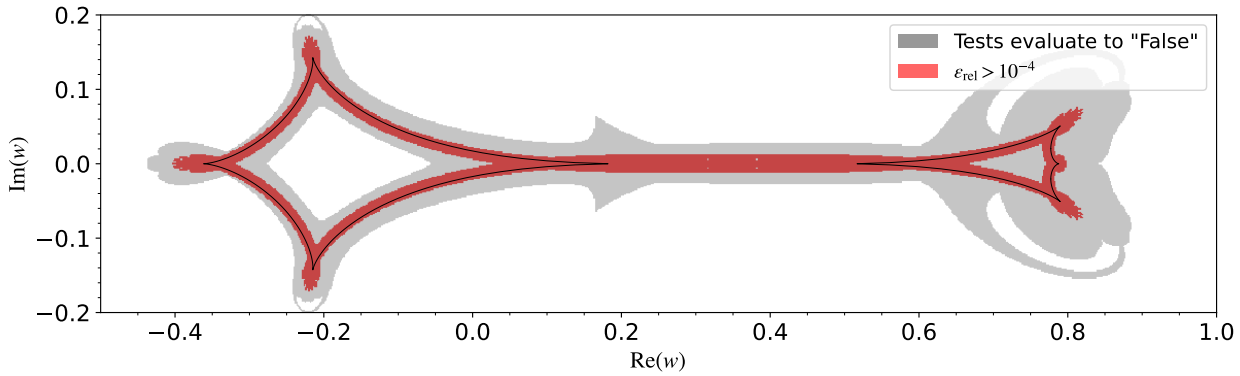


Figure 3.10: Visualization of the tests that determine whether the hexadecapole approximation is sufficiently accurate or if the full contour integration is necessary. The grey region is where the test indicates that the full integration is needed and the red region is where the error of the hexadecapole approximation is greater than 10^{-4}. The tests are sufficiently sensitive to the detect the breakdown of the hexadecapole approximation.

Planetary test

Finally, if the source is large enough and the binary lens mass ratio small enough, we could have a situation where the limb of the source touches the planetary caustic, while the centre of the source is so far that the gradient of the Jacobian determinant is not sensitive to the planet. Bozza et al. (2018) tests for this situation by checking that the distance from w_0 to the centre of the planetary caustic is greater than $\rho_\star$. The condition is

$$|w - w_{pc}|^2 > c_p \left(\rho_\star^2 + \Delta_{pc}^2\right) \quad , \tag{3.41}$$

where $w_{pc} = -1/(2a) - a(2\epsilon_1 - 1)$ is the position of the binary lens planetary caustic, and $\Delta_{pc} \equiv 3\frac{\sqrt{q}}{2a}$. This test is only needed for small mass ratios when $q < 0.01$.

I then choose the values of the free parameters (c_m, γ, c_f, c_p) such that the full contour integration is triggered only when the relative error is greater than a threshold value which I set to 10^{-4}. This is a sort of optimisation problem where the goal is to minimise the number of false negatives while also keeping the number of false positive results relatively low. I find that the setting

$$c_m = 10^{-2}, \quad \gamma = 0.02, \quad c_f = 4, \quad c_p = 2 \tag{3.42}$$

works well enough. Figure 3.10 shows the application of the tests for a binary system with $a = 0.8$, $\epsilon_1 = 0.95$ and $\rho_\star = 5 \times 10^{-3}$. The caustic curves are shown as black lines, the red region indicates where the (relative) error of the hexadecapole approximation relative to a complete contour integration is greater than 10^{-4} and the grey regions are where the tests indicate that full integration is necessary. Unfortunately, the tests are far from perfect and there are large areas of the source plane where the tests give a false positive result (i.e. the full contour integration is triggered when it is not necessary). As of the time of writing, I have not implemented these tests for the triple lens case.

3.8 Automatic differentiation

To test that automatic differentiation works as intended, I computed the derivative of the extended source magnification with respect to the key input parameters. I then compare this

with a finite-difference approximation of the derivative. I evaluate the derivative at points at or very close to the caustics and I found that the gradients are correctly computed, as one would expect. These tests are also implemented in the test suite for the code. In terms of the computational cost, I find that evaluating the gradient of the likelihood using reverse-mode automatic differentiation is about 3-4 times slower than the cost of evaluating the likelihood itself. This is as expected.

To visualize the gradients, in Figure 3.11 I show the predicted flux for a caustic-crossing trajectory across a binary lens caustic pattern (top panel), and the associated gradients at each point along the trajectory (bottom panels).

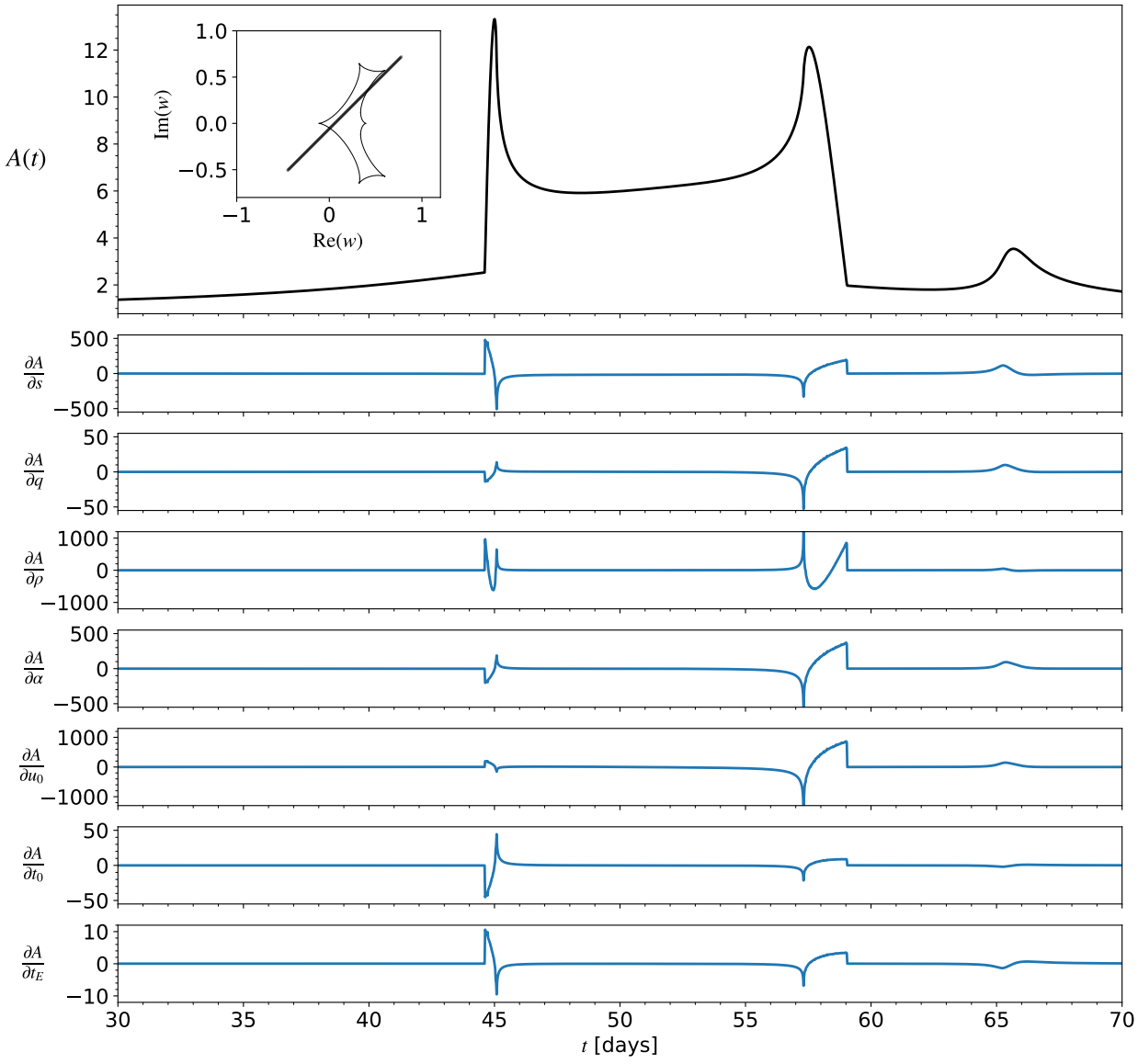


Figure 3.11: Predicted flux for a caustic-crossing trajectory across a binary lens magnification pattern (top panel) and the associated gradients at each point along the trajectory (bottom panels). The gradients are computed through automatic differentiation using the caustics code.

3.9 Astrometric microlensing

Although support for astrometric microlensing is not yet implemented in the caustics, it is straightforward to compute the astrometric shift. Here I briefly describe how to do it. Following Dominik (1998c), the shift in the centre-of-light relative to the centroid of the unlensed source, denoted by the complex variable z_{cent}, is given by

$$z_{\text{cent}} = \frac{\iint_{S'} z\, I_s(z_1', z_2')\, \mathrm{d}z_1'\, \mathrm{d}z_2'}{\iint_{S'} I_s(z_1', z_2')\, \mathrm{d}z_1'\, \mathrm{d}z_2'} \quad . \tag{3.43}$$

The above equation is analogous to the equation for the astrometric shift for a single lens illuminated by a point source which we defined in Equation 2.29. The integral in the denominator is the one I described in the previous sections in this chapter. The one in the numerator is very similar except it requires a slight modification for the functions $P(z_1, z_2)$ and $Q(z_1, z_2)$.

Integrand $f(z_1, z_2)$	Description	$P(z_1, z_2)$	$Q(z_1, z_2)$
1	Area of images	$-\frac{1}{2} z_2$	$\frac{1}{2} z_1$
$I_s(z_1, z_2)$	Brightness distribution integrated over image area	$-\frac{1}{2} \int_{z_2^{(0)}}^{z_2} I_s\left(z_1, z_2'\right) \mathrm{d}z_2'$	$\frac{1}{2} \int_{z_1^{(0)}}^{z_1} I_s\left(z_1', z_2\right) \mathrm{d}z_1'$
z_1	x-component of centroid shift (uniform brightness)	$-\frac{1}{2} z_1 z_2$	$\frac{1}{4} z_1^2$
z_2	y-component of centroid shift (uniform brightness)	$-\frac{1}{4} z_1^2$	$\frac{1}{2} z_1 z_2$
$z_1\, I_s(z_1, z_2)$	x-component of centroid shift (arbitrary brightness distribution)	$-\frac{1}{2} z_1 \int_{z_2^{(0)}}^{z_2} I_s\left(z_1, z_2'\right) \mathrm{d}z_2'$	$\frac{1}{2} \int_{z_1^{(0)}}^{z_1} z_1'\, I_s\left(z_1', z_2\right) \mathrm{d}z_1'$
$z_2\, I_s(z_1, z_2)$	y-component of centroid shift (arbitrary surface brightness)	$-\frac{1}{2} \int_{z_2^{(0)}}^{z_2} z_2'\, I_s\left(z_1, z_2'\right) \mathrm{d}z_2'$	$\frac{1}{2} z_2 \int_{z_1^{(0)}}^{z_1} I_s\left(z_1', z_2\right) \mathrm{d}z_1'$

Table 3.1: Table showing the different integrand functions $f(z_1, z_2)$ for a surface integral over the images $\iint_{S'} f(z_1, z_2)\, \mathrm{d}z_1\, \mathrm{d}z_2$. These functions satisfy the constraint $f(z_1, z_2) = \partial Q/\partial z_1 - \partial P/\partial z_2$ so that we can use Green's theorem (Equation 3.18) to transform the two-dimensional surface integral into a one-dimensional line integral. The functions in the first two rows from the top are used to compute the photometric microlensing effect and the rest are used to compute the astrometric microlensing effect.

Table 3.1 shows the different choices of integrand functions $f(z_1, z_2)$ for a surface integral $\iint_{S'} f(z_1, z_2)\, \mathrm{d}z_1\, \mathrm{d}z_2$ which are used in the computation of the photometric and astrometric microlensing effects, and the corresponding functions $P(z_1, z_2)$ and $Q(z_1, z_2)$ which satisfy the constraint $f(z_1, z_2) = \partial Q/\partial z_1 - \partial P/\partial z_2$. The first two rows of the table are the integrands for the uniform-brightness and limb-darkened photometric microlensing effect we have seen in the previous sections. The following rows are the functions for the real and

imaginary components of the integrand in the numerator of Equation 3.43 for the uniform brightness and limb-darkened brightness distributions. We see that for the astrometric microlensing effect we have to numerically estimate three surface integrals in total, the one in the denominator of Equation 3.43 and the two components of the integral in the numerator, compared to one for the photometric effect.

3.10 Summary

In summary, in this chapter, I have introduced a novel code for computing the magnification in microlensing events. At the time of writing, this code is still in active development. The code includes the following features:

- caustics is written in Python and it is fully open-source. It is well documented and highly modular.
- Accurate computation of the single, binary and triple lens magnification of uniform-brightness and limb-darkened source stars.
- Performance comparable to VBBinaryLensing for binary lens events, orders of magnitude faster than existing codes for triple lens events.
- Novel Aberth-Ehrlich complex root solver for solving the lens equation. The root solver has an optional high-precision mode which uses compensated complex number arithmatic to effectively double the working precision at only twice the computational cost.
- First microlensing code with full support for automatic differentiation, via the JAX library. This enables the use gradient-based optimizers and samplers when fitting multiple-lens microlensing events.
- Support for annual microlensing parallax effects.
- Hexadecapole approximation used to speed up the computation of the magnification for an entire light curve (currently only for binary lens events).

Notable features that are still missing from caustics are:

- Adaptive integration for contour integration. Currently, the only way to control accuracy is to set an initial fixed number of evaluation points for the contour integration.
- Support for hexadecapole approximation with triple lens events.
- Astrometric microlensing.
- Sources with arbitrary brightness profiles.

Chapter 4

Modelling single lens microlensing events: inference, model comparison and multi-modal posteriors

In this chapter I tackle one of the fundamental problems in microlensing – how to assess and compare different models? Since this is a very broad question which appears in some form in every single paper on the analysis of microlensing events, I attempt to answer the question in the context of single lens events with parallax, although the results should generalise to other models. Specifically, I use a particular single lens event which exhibits annual parallax deviations as a case study. The purpose of this is to demonstrate a novel solution to the problem of fitting microlensing models with multi-modal posterior distributions – a variant of the model comaprison problem. I argue that this approach is superior to alternatives and I also demonstrate a method for speeding up inference by orders of magnitude by making use of the Laplace approximation. This work relies on some recent developments in computational Bayesian statistics which have not been applied to microlensing before and have also been under-utilised in astronomy as a whole.

4.1 Introduction

Although the majority of effort dedicated to modelling microlensing events in recent years has been dedicated to binary and triple-lens events, single lens events, which are an order of magnitude more common than multiple-lens events, exhibit many of the same pathologies as binary and triple-lens events, in addition to sharing common parameters. single lens events are also very important scientifically on their own for two key reasons. First, inferring the tail properties of the t_E (event timescale) distribution enables us to make predictions about the number of free-floating exoplanets in the Galaxy (Sumi et al., 2011; Mróz et al., 2017). Second, by measuring the microlensing parallax π_E, in addition to t_E, we can place constraints on the mass of the lens (Wyrzykowski et al., 2016) which is especially important for finding intermediate-mass black holes. Using a microlensing population synthesis model, Lam et al. (2020) found that if a given event has $t_E \gtrsim 120$ days and $\pi_E \lesssim 0.08$, it is highly likely to be a black hole. Hence, by jointly inferring t_E and π_E, we can constrain properties

of the black hole distribution in our galaxy.

Both of these applications are highly sensitive to modelling assumptions. For instance, the event timescale t_E is often very poorly constrained (Dominik, 2009) for blended events and events with correlated noise in the light curve (Golovich et al., 2022). Events with annual parallax deviations have degeneracies which result in multi-modal posterior distributions over the model parameters (Gould, 2004). Multi-modal posteriors are an even bigger problem for multiple-lens events, but they are easier to study in the context of single lenses.

In this chapter, I use the single lens parallax event OGLE-2005-BLG-086 as a case study to demonstrate a method for fitting microlensing models with multi-modal posteriors. There are two ways one could approach this problem. The first is to view it as a *parameter estimation* problem. Seen through that lens, our goal when solving the problem is to devise a scheme which discovers all the modes (more precisely, regions in the parameter space) and produces faithful samples from the multi-modal posterior distribution. This is a very hard problem because MCMC (and optimisation) methods which use *local information* about the target distribution (e.g., its scalar value and gradient) will inevitably get stuck in one of the modes[1] if the modes are separated by regions of low probability. Seen through the lens of Hamiltonian Monte Carlo, a posterior distribution $p(\boldsymbol{\theta})$ with two separated modes has a potential energy barrier between the two modes (because $U(\boldsymbol{\theta}) \propto -\ln p(\boldsymbol{\theta})$) which decreases the transition probability between the modes. Methods which do not exclusively use local information (such as Nested Sampling) are more successful at finding the different modes (at least for low-dimensional problems), but they are much less computationally efficient.

A second approach is to view this as a *model comparison* problem. Model comparison in the Bayesian paradigm is traditionally most often used to compare models with different numbers of parameters and likelihoods, but it can also be used in the context of a single model with multiple modes. I will argue that this second paradigm is more appropriate for the problem at hand because it is more robust to model misspecification if we judge the different modes by their predictive accuracy on unseen data. It also happens to be quite computationally efficient.

The chapter is organized as follows. In Section 4.2 I briefly introduce the dataset which I use throughout the chapter. In Section 4.3 I specify the annual parallax model for the magnification, the likelihood function, and the priors, which forms a complete description of the model I use in the following sections. In Section 4.4 I show that fitting this model using Hamiltonian Monte Carlo MCMC leads to chains getting stuck in one of four modes in the posterior distribution. Each mode represents a different trajectory of the source star on the plane of the sky. I introduce cross-validation (CV), specifically leave-one-out cross-validation (LOO-CV), as a method for comparing the different modes based on their predictive performance. I also discuss the use of LOO-CV as a powerful way of assessing model fit. In Section 4.5, I review the different options for combining the information from posterior modes and compare the LOO-CV approach of weighting MCMC samples to Nested Sampling. Section 4.6 deals with the question of how to find the different modes in the first place. I demonstrate that an optimisation approach using the Laplace approximation

[1]It is worth keeping in mind that optimizers and MCMC samplers aren't looking for the same thing. Optimizers are looking for minima (points) in the parameter space while samplers are looking for typical sets – regions with high probability mass (see Section 2.4.2). I will use the word 'modes' to refer to both of these objects.

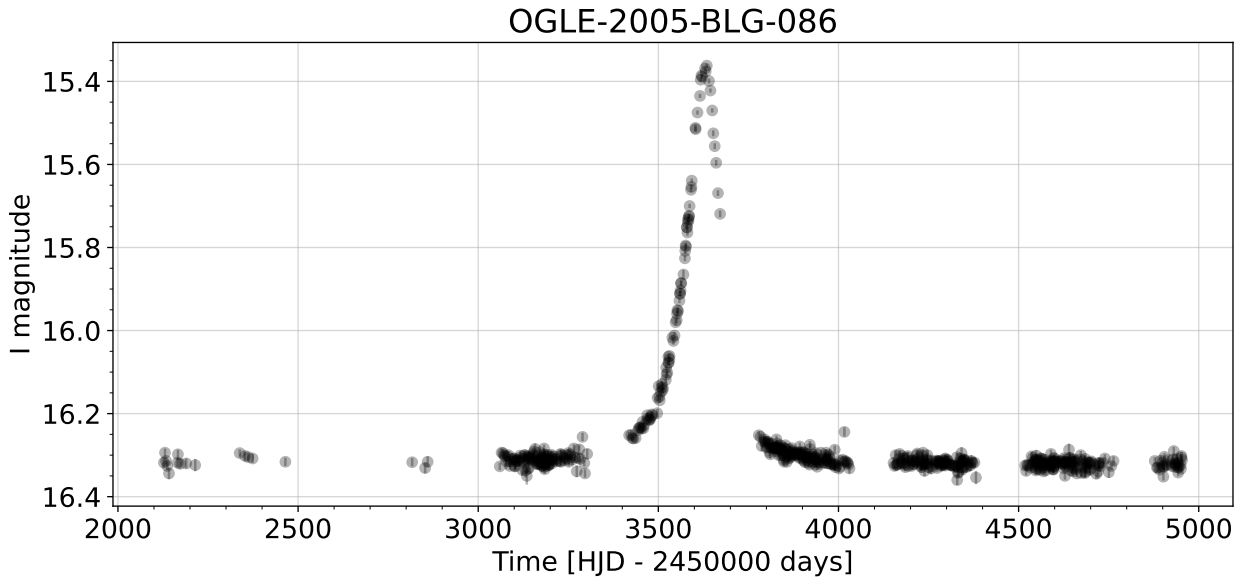


Figure 4.1: OGLE EWS light curve for the event OGLE-2005-BLG-086.

combined with LOO-CV works well as a general solution for modelling single lens events. It is orders of magnitude faster than the alternatives. Finally, in Section 4.7, I summarise the results, and discuss future work.

4.2 Data

The event I have chosen to model is OGLE-2005-BLG-086. The sky coordinates of this event are $(\alpha, \delta) = (18^h 4^m 45.7^s, -26°59'15.5')$ and the light curve is shown in Figure 4.1. The details about the photometry pipeline used for light curve extraction from raw images was previously published in Udalski et al. (2008). OGLE-2005-BLG-086 is a fairly typical single lens long-timescale event which exhibits subtle deviations in the wings of the light curve because of the annual parallax effect. It was previously modelled by Wyrzykowski et al. (2015) and fit with a single lens model without parallax. The inferred parameters from that analysis are shown in Table 4.1.

Parameter	t_0 HJD - 2450000 (days)	t_E (days)	u_0	F_S
Value	$3628.293103^{+0.135788}_{-0.132710}$	$102.084389^{+1.425696}_{-1.402358}$	$0.374351^{+0.008415}_{-0.007950}$	$0.775243^{+0.023732}_{-0.021796}$

Table 4.1: Inferred parameters of a single lens model without parallax fitted to the OGLE-2005-BLG-086 light curve, from Wyrzykowski et al. (2015). The uncertainties correspond to 15 and 85% confidence intervals.

Since the microlensing magnification modifies the flux of the source star, in subsequent analysis I convert the I-band magnitude measurements to fluxes using the equation $\mathbf{f} = 10^{-(\mathbf{m}_I - m_0)/2.5}$, where $\mathbf{m}_I$ is the I-band magnitude vector, m_0 is the zero point magnitude and $\mathbf{f}$ is the flux vector. The zero point magnitude m_0 is set to 22.

4.3 Model

I model the event with a single lens point source model which includes annual parallax effects. It makes sense to use the annual parallax model as the default model in this case because the event is bright and the width of the peak is equal to a non-negligible fraction of a year. This means that there is a good chance that parallax deviations are detectable. The light curve shown in Figure 4.1 does not appear to have appreciable exteded source effects so a point source model is sufficient. The point source magnification is simply

$$A(t) = \frac{u(t)^2 + 2}{u(t)\sqrt{u(t)^2 + 4}} \quad , \tag{4.1}$$

where $u(t) \equiv |\mathbf{u}(t)|$ is the apparent trajectory of the source star on the plane of the sky whose two components are given by Equations 2.54 and 2.55 (see Section 2.1.6 for the derivation). I repeat the equations here for convenience:

$$u_e = u_0 \cos\psi + (t - t'_0)/t'_E \sin\psi + \pi_E \, \delta\zeta_E(t) \tag{4.2}$$

$$u_n = -u_0 \sin\psi + (t - t'_0)/t'_E \cos\psi + \pi_E \, \delta\zeta_N(t) \quad . \tag{4.3}$$

Instead of the (π_E, ψ) parametrisation for the trajectory, I use the two microlensing parallax components $\pi_{E,N} \equiv \pi_E \cos\psi$ and $\pi_{E,E} \equiv \pi_E \sin\psi$, because the angle ψ has a discontinuity at π which is problematic when fitting the model using NUTS. The five free parameters that determine the magnification are then $\boldsymbol{\theta} \equiv (t'_0, t'_E, u_0, \pi_{E,E}, \pi_{E,N})$.

To evaluate the magnification at an arbitrary point in the parameter space, we have to be able to evaluate the two components of the projected position vector of the Solar System Barycentre (SSB) on the plane of the sky, $\delta\zeta_E(t)$ and $\delta\zeta_N(t)$, at some time t. There are two ways we could do this. The first is to parametrise the Earth's orbit as a Keplerian orbit and compute the projected position of the SSB in a geocentric frame analytically. The second is to use the JPL Horizons ephemeris which relies on highly accurate N-body simulations of the Solar System. I chose the second option because, although the analytic approximation of the orbit is often sufficient, obtaining the high accuracy ephemeris from JPL Horizons is very straightforward so we might as well use the more accurate approach.

I use the function get_body_barycentric_posvel from the Astropy package to obtain the position and the velocity vector of the SSB with respect to Earth, on a grid with a time resolution of 1 day. The evaluation times start at the time of the first observation and end at the time of the last observation in the light curve. I then project the vectors onto the plane of the sky (which is defined by the coordinates (α, δ) of the source star) using Equations 2.34 and 2.35, to obtain the two components of the projected position $(\zeta_E(t), \zeta_N(t))$, and velocity $(\dot{\zeta}_E(t), \dot{\zeta}_N(t))$ of the SSB on the plane of the sky. These arrays are precomputed before fitting. Since the computation of the position offset of the SSB relative to its position at time t'_0 (Equation 2.43) depends on the parameter t'_0, it is computed on the fly during the fitting process by linearly interpolating the values of $\zeta_E(t)$ and $\zeta_N(t)$ and their derivatives. Only this last interpolation step is evaluated every time the likelihood is evaluated. I use the linear interpolation function from JAX, jax.numpy.interp, which is automatically differentiable. The entire procedure from computing the ephemeris to evaluating the trajectory is implemented in the caustics code in a class called AnnualParallaxTrajectory.

The next step is to choose the parametrisation for the linear flux parameters. I use the parameters F_S (the source star flux) and $F_{\rm base} \equiv F_S + F_B$. The advantage of this parametrisation over the parametrisation (F_S, F_B) is that $F_{\rm base}$ (the baseline flux) is directly related to the light curve and F_S is a physical parameter of interest. The predicted flux is then given by the matrix equation

$$\mathbf{f} = \mathbf{M}\,\boldsymbol{\beta} \quad , \tag{4.4}$$

where $\boldsymbol{\beta} \equiv (F_S, F_{\rm base})^\intercal$, and the design matrix $\mathbf{M}$, which depends on the non-linear parameters $\boldsymbol{\theta}$, is given by

$$\mathbf{M} = \begin{pmatrix} A(t_1;\boldsymbol{\theta}) - 1 & 1 \\ A(t_2;\boldsymbol{\theta}) - 1 & 1 \\ \vdots & \vdots \\ A(t_N;\boldsymbol{\theta}) - 1 & 1 \end{pmatrix} \quad . \tag{4.5}$$

It is not necessary to fit for the linear parameters because we can analytically marginalise the Gaussian likelihood over those parameters, as described in Section 2.3.6. In addition to reducing the dimensionality of the problem, this procedure eases sampling and optimisation of the posterior distribution because there are strong non-linear covariances between the source star flux F_S and the parameters t'_E and u_0 for blended events which results in a complicated geometry of the posterior density. By marginalising these parameters we only have to worry about the covariances between the non-linear parameters.

I assume that the data covariance matrix is a diagonal matrix comprised of the variances of the flux measurements (the "error bars"), which are provided by OGLE. Although this noise model does not have the flexibility to capture correlated noise features in the light curve (caused by seeing variations or stellar variability) which should be included in a more realistic model, for the purposes of this chapter, is sufficient. In Section 4.4.3 I discuss how one can extend the model so it includes a more sophisticated treatment of noise. Since the data covariance matrix is assumed to be diagonal, we can infer the linear flux parameters estimate $\hat{\boldsymbol{\beta}}$ using weighted least squares at every iteration, which is faster than computing the marginalised likelihood from Equation 2.183.

Finally, we have to specify the priors for the non-linear parameters. These are listed in Table 4.2. They are all broad weakly informative priors with the exception of the prior for the $\ln t_0$ parameter which is a Gaussian centred at a particular value $t'_{0,\rm estimate}$ – an estimate of the t'_0 parameter using the raw light curve. It is obtained by passing the light curve through a median filter and selecting the observation time of the point with the maximum flux. The reason I do this is because t'_0 is usually a very well-constrained parameter not sensitive to the choice of the prior and placing a prior centred at the time of the peak magnification eases fitting.

Parameter	$\ln(t'_0/d)$	$\ln(t'_E/d)$	u_0	$\pi_{E,N}$	$\pi_{E,E}$
Prior	$\mathcal{N}(\ln\left(t'_{0,\rm estimate}\right), 2.3)$	$\mathcal{N}(2.5, 1.5)$	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$	$\mathcal{N}(0,1)$

Table 4.2: Prior distributions for model parameters. $t'_{0,\rm estimate}$ is the estimated time of the peak flux in the light curve.

Having specified the priors, we can write down the final posterior as

$$p(\boldsymbol{\theta}|\mathbf{f}) \propto \mathcal{N}(\mathbf{f}|\mathbf{M}\hat{\boldsymbol{\beta}}, \mathbf{C})\, p(\boldsymbol{\theta}) \quad , \tag{4.6}$$

where $\mathbf{C}$ is the diagonal data covariance matrix, and $\hat{\boldsymbol{\beta}}$ is the weighted least squares solution for the linear flux parameters obtained by solving the linear system

$$(\mathbf{M}^\intercal \mathbf{C}^{-1} \mathbf{M}) \hat{\boldsymbol{\beta}} = \mathbf{M}^\intercal \mathbf{C}^{-1} \mathbf{f} \quad . \tag{4.7}$$

4.4 Model (mode) comparison

I implemented the model specified in Equation 4.6 in the probabilistic programming language numpyro (Phan et al., 2019). numpyro is a Python library built using JAX which comes with a robust implementation of the NUTS Hamiltonian Monte Carlo sampler, various optimizers, probability distributions and transforms. NUTS uses the gradient of the posterior distribution with respect to the model parameters to substantially improve the sampling efficiency relative to gradient-free MCMC samplers[2].

There are four distinct modes in the posterior distribution (see Section 4.6 on how to find the modes). I sample each mode separately using the NUTS sampler with 1000 warmup iterations (using the default numpyro strategy for estimating the HMC mass matrix) and 2000 sampling iterations. To make sure that the chains have converged I checked the ESS for all parameters is at least 400, and that the number of divergent transitions[3] during the leapfrog integration for each sample, is either zero or equal to a very small fraction of the total number of samples. The inferred parameters for each chain are shown in Table 4.3.

Name	t'_0 HJD - 2450000 (days)	t'_E (days)	u_0	$\pi_{E,N}$	$\pi_{E,E}$
Mode 1	$3629.874^{+0.171}_{-0.170}$	$109.324^{+3.831}_{-3.575}$	$-0.416^{+0.013}_{-0.014}$	$-0.297^{+0.031}_{-0.025}$	$0.116^{+0.009}_{-0.009}$
Mode 2	$3629.783^{+0.171}_{-0.159}$	$92.942^{+2.234}_{-2.256}$	$0.488^{+0.020}_{-0.018}$	$0.266^{+0.030}_{-0.033}$	$0.106^{+0.009}_{-0.009}$
Mode 3	$3630.218^{+0.206}_{-0.201}$	$122.337^{+4.905}_{-4.623}$	$-0.229^{+0.012}_{-0.013}$	$0.340^{+0.013}_{-0.015}$	$0.074^{+0.005}_{-0.005}$
Mode 4	$3630.370^{+0.220}_{-0.222}$	$151.626^{+8.345}_{-7.542}$	$0.178^{+0.013}_{-0.012}$	$-0.288^{+0.009}_{-0.009}$	$0.075^{+0.005}_{-0.005}$

Table 4.3: Inferred model parameters from single NUTS MCMC chains trapped in four separate modes in the posterior. The sampling converged in each case but the solutions are quite different from each other.

Figure 4.2 shows the projection of the samples in the physically relevant parameters (t_E, π_E) for each of the four modes. There are two dominant modes with comparable likelihood values (Mode 1 and Mode 2 in the figure) and two minor modes (Mode 3 and Mode 4). All four solutions are quite different from each other in the (t_E, π_E) space. The physical interpretation of the different modes is obvious if we plot the inferred trajectories on the plane of the sky. These trajectories are shown in Figure 4.3. Each line is a single sample from the posterior distribution for each of the modes and the arrows indicate the direction of the motion of the source star. It is clear that each of the modes represents a different

[2]The exact speedup depends on the details of the problem – the dimensionality and the geometry of the parameter space. NUTS is generally at least an order of magnitude faster than a gradient-free sampler such as emcee (Foreman-Mackey et al., 2013a) but the speedup can also be much larger.

[3]Divergent transitions occur when the gradients of the target distribution become very large. This indicates that the chains become trapped in regions of high curvature in the parameter space and they may not be exploring the typical set fully (Betancourt, 2017).

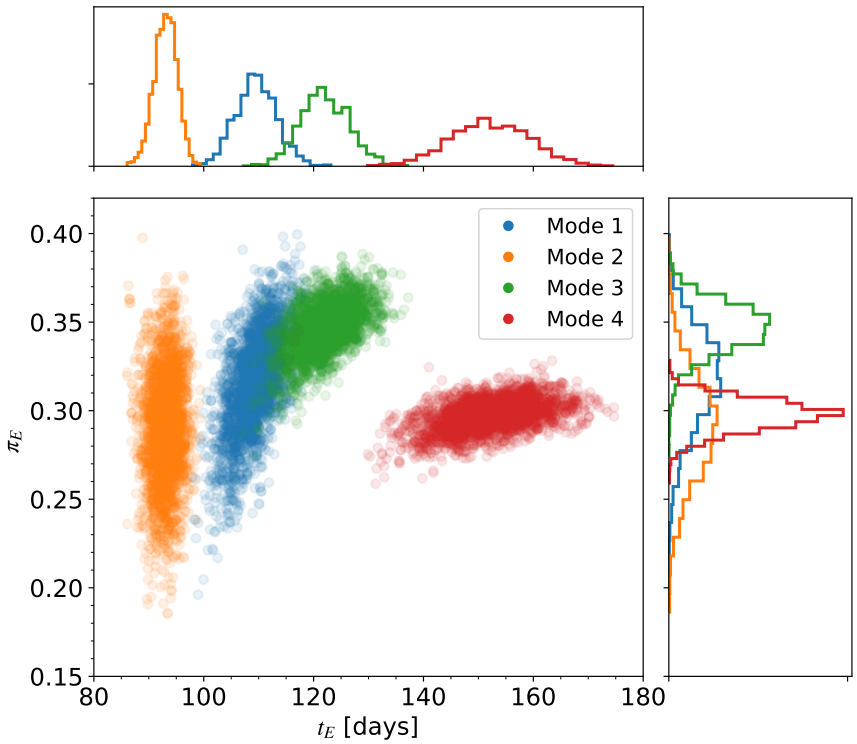


Figure 4.2: Projection of the posterior samples onto parameters (t_E, π_E). Each colour indicates samples from a single NUTS MCMC chain trapped in a different mode of the posterior distribution. Modes 1 and 2 are dominant modes with comparable likelihood while modes 3 and 4 have lower values of the likelihood. The panels on the top and right show the histograms of the marginal distributions.

trajectory of the source on the plane of the sky. The two dominant modes (blue and orange) have similar absolute values of u_0 with opposite signs. One solution corresponds to the source star moving South to North (orange) while the other corresponds to the source moving North to South (blue). The two minor modes (green and red) also differ mainly in the sign of u_0 but the difference in event timescales between these two modes is greater.

These degeneracies for annual parallax events have been known for some time (Gould, 2004) but there is still no definitive answer to the obvious question: *how can we compare the different modes*? With the exception of the recent work by Kaczmarek et al. (2022), this question has been largely ignored in the microlensing literature. As mentioned in the introduction, one way to approach this problem is from the perspective of parameter estimation where the goal is to draw independent samples from the multi-modal posterior distribution $p(\boldsymbol{\theta}|\mathbf{f}, \mathcal{M})$. Only global exploration methods such as Nested Sampling can reliably solve this problem (if the dimensionality of $\boldsymbol{\theta}$ is not too large) because local exploration methods will inevitably get stuck in one of the modes[4]. The alternative is to fit each of the modes separately and then do *model comparison* (to rank the different modes) or *model averaging* (to pool the information from the different modes). In the following section, I will explore

[4]Of course, there are many many variants of MCMC which claim to be able to jump between the different modes reliably. In practice, however, I have found that these methods rarely work on realistic problems and if they do, they are extremely unreliable. Nested Sampling, on the other hand, seems to be reasonably robust and it is commonly used in many branches of physics and astronomy.

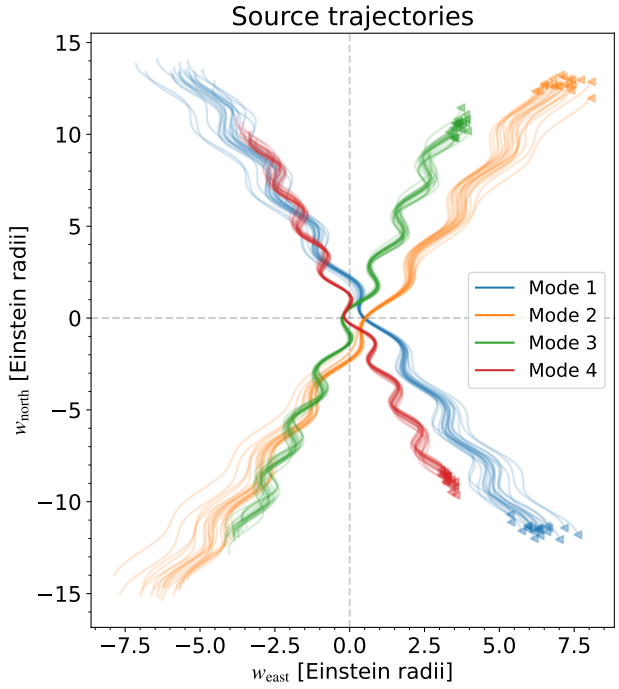


Figure 4.3: Posterior source star trajectories on the plane of the sky for each of the four modes. Each thin line is one possible trajectory of the source star within one of the four modes in the posterior distribution. The arrows indicate the direction of the source. Both modes 1 and 2 and modes 3 and 4 have similar absolute values of u_0 but with the opposite sign.

the latter approach by using cross-validation to rank the different modes based on their predictive accuracy. In Section 4.5 I compare this approach to the parameter estimation approach using Nested Sampling.

4.4.1 Cross validation

Cross-validation is one of the most popular and robust methods for model comparison in statistics. *The fundamental idea behind CV is to judge the performance of different models based on how well they predict unseen data.* Here, "models" can refer to entirely different models – with different structures, likelihoods and numbers of parameters, or it can refer to different modes in the posterior. "Unseen data" refers to data that we have not observed or data that we don't use to fit the model[5]. In the context of microlensing light curves, we are primarily interested in the predictive performance of the model at times in between the existing observations.

Consider a model $\mathcal{M}$, observed data $\mathbf{y}$, and a posterior distribution $p(\boldsymbol{\theta}|\mathbf{y}, \mathcal{M})$. The posterior predictive distribution

$$p(\tilde{\mathbf{y}}|\mathbf{y}, \mathcal{M}) = \int p(\tilde{\mathbf{y}}|\boldsymbol{\theta}, \mathcal{M})\, p(\boldsymbol{\theta}|\mathbf{y}, \mathcal{M})\, \mathrm{d}\boldsymbol{\theta} \tag{4.8}$$

expresses our beliefs about new data $\tilde{\mathbf{y}}$ given the posterior distribution of the model parameters $\boldsymbol{\theta}$, conditional on observed data $\mathbf{y}$. This may sound a little abstract but it is actually

[5]In machine learning these are the validation/test sets.

quite simple to understand. If we previously obtained samples from the posterior distribution $p(\boldsymbol{\theta}|\mathbf{y}, \mathcal{M})$ using a method such as MCMC, we can use these samples to generate samples from the posterior predictive distribution by plugging them into the data distribution $p(\tilde{\mathbf{y}}|\boldsymbol{\theta}, \mathcal{M})$, and drawing a new data vector $\tilde{\mathbf{y}}$ for each sample $\boldsymbol{\theta}^{(s)}$. In our case, this would mean fitting a microlensing light curve $\mathbf{f}$ and drawing a sample from the Gaussian likelihood $p(\mathbf{f}|\boldsymbol{\theta}, \mathcal{M})$ for each parameter vector $\boldsymbol{\theta}^{(s)}$. We could evaluate the new data $\tilde{\mathbf{y}}$ at the same times as the observed data $\mathbf{y}$, or we could evaluate it at some other times. This process (simulating the data conditional on posterior samples) is a useful way of checking the model fit and it goes by the name of *posterior predictive checking* (PPC).

To measure how close these predictions are to the *true data generating process* (which is unknown), we can construct a utility function which quantifies the predictive performance of the model. A function which is most commonly used is the *expected log predictive density* (elpd) of the model $\mathcal{M}$ (see Yao, 2019, for a discussion of list of other common choices). It is the expectation of the log posterior predictive density under the true data generating process $p_t(\tilde{\mathbf{y}}|\mathbf{y})$:

$$\text{elpd} \equiv \mathbb{E}_{p_t(\tilde{\mathbf{y}})} \left[\ln p(\tilde{\mathbf{y}}|\mathbf{y}, \mathcal{M})\right] \tag{4.9}$$

$$= \int \ln p(\tilde{\mathbf{y}}|\boldsymbol{\theta}, \mathcal{M})\, p_t(\boldsymbol{\theta}|\mathbf{y})\, \mathrm{d}\boldsymbol{\theta} \quad . \tag{4.10}$$

This quantity can be approximated using *leave-one-out* cross validation (LOO-CV)[6] as (Vehtari et al., 2016)

$$\text{elpd}_{\text{loo}} \approx \sum_{i=1}^{n} \ln p(\tilde{\mathbf{y}}_i|\mathbf{y}_{-i}, \mathcal{M}) \tag{4.11}$$

$$= \sum_{i=1}^{n} \ln \left[p(\mathbf{y}_i|\boldsymbol{\theta}, \mathcal{M})\, p(\boldsymbol{\theta}|\mathbf{y}_{-i}, \mathcal{M})\, \mathrm{d}\boldsymbol{\theta}\right] \quad , \tag{4.12}$$

where $\mathbf{y}_i$ is the i-th data point and $p(\boldsymbol{\theta}|\mathbf{y}_{-i})$ is the posterior distribution over the model parameters conditional on all data points except the i-th one (hence the name LOO). Watanabe (2010) showed that Equation 4.10 is a consistent and unbiased estimator of the true elpd.

There are two fundamental assumptions behind the approximation of the elpd using Equation 4.12 (and CV approximations of elpd in general). The first assumption is that existing data are a good representation of new data. The second assumption is that observations or groups of observations are exchangeable[7] conditional on the model parameters. Otherwise, the estimator in Equation 4.12 is not an unbiased approximation of the true expectation over unseen data. The second assumption means that we have to be careful when applying CV to data correlated in space or time, although there are cross-validation schemes designed to work for such problems. For a more thorough discussion of these assumptions see Vehtari et al. (2018).

How can we evaluate the expression in Eq. 4.12? In principle, computing the elpd using Eq. 4.12 requires us to re-fit the model for each data point which is computationally very

[6] It can also be approximated using K-fold cross-validation but LOO is easier to work with.

[7] Parameters $\theta_1, \theta_2, \ldots, \theta_D$ are exchangeable if their joint distribution $p(\boldsymbol{\theta})$ is invariant under permutations of the indices $1, 2, \ldots, D$.

expensive. However, in practice, the LOO posterior $p(\boldsymbol{\theta}|\mathbf{y}_{-i})$ is often very close to the full posterior $p(\boldsymbol{\theta}|\mathbf{y}, \mathcal{M})$ and we can approximate it using importance sampling without having to re-fit the model. If we have samples from the full posterior $p(\boldsymbol{\theta}|\mathbf{y})$, the importance sampling approximation of Equation 4.12 is

$$\text{elpd}_{\text{is-loo}} \approx \sum_{i=1}^{n} \ln \left(\frac{\frac{1}{S}\sum_{s=1}^{S} p\left(\mathbf{y}_i|\boldsymbol{\theta}^{(s)}, \mathcal{M}\right) w\left(\boldsymbol{\theta}^{(s)}\right)}{\frac{1}{S}\sum_{s=1}^{S} w\left(\boldsymbol{\theta}^{(s)}\right)} \right) \quad , \tag{4.13}$$

where $w \propto 1/p\left(\mathbf{y}_i|\boldsymbol{\theta}^{(s)}, \mathcal{M}\right)$ are the importance weights. Using importance sampling to estimate the LOO posterior $p(\boldsymbol{\theta}|\mathbf{y}_{-i})$ is also especially useful in the particular case of comparing modes within a single posterior distribution. The reason is that if we were to explicitly re-fit the model by removing one data point at a time the MCMC chains could converge to the wrong mode for the reduced dataset. There could also be a phase transition leading to a merger of one or several modes (Yao et al., 2020). Importance sampling gets around that problem but it is still problematic when individual data points $\mathbf{y}_i$ are quite influential on the posterior and the resulting distribution of importance weights is dominated by a few outliers. Vehtari et al. (2015b) solves this problem by fitting a generalised Pareto distribution to the distribution of importance weights, and replacing the largest weights with the predictions from the fitted distribution – so called *Pareto smoothed importance sampling* (PSIS).

The elpd estimated using Equation 4.13 with Pareto smoothed importance weights is denoted by $\text{elpd}_{\text{psis-loo}}$. The Pareto distribution has the form $p(r \mid u, \sigma, k) = \sigma^{-1}\left(1 + k(r-u)\sigma^{-1}\right)^{-1/k-1}$ where u is the location, σ is the scale and k is the shape parameter. An output of this PSIS procedure (described in Vehtari et al., 2015b; Vehtari et al., 2016) is a set of new stabilised weights $w(\boldsymbol{\theta}^{(s)})$, and an inferred shape parameter $\hat{k}$ (for each data point $\mathbf{y}_i$). These Pareto shape parameters $\hat{k}$ have very useful diagnostic value. When $\hat{k}$ is greater than about 0.7, the importance sampling estimate of the LOO posterior (or the importance sampling estimate of some other distribution) is unlikely to be reliable (Vehtari et al., 2015b). This in turn also implies that $\mathbf{y}_i$ is a highly influential observation.

Does it make sense to apply LOO-CV to a *time-series* dataset, where the data points have a specific ordering? In cases where we only care about predicting future data points conditional on having observed past data points, LOO-CV is not a good measure of the predictive performance of the model because it makes use of future data points to assess the predictive accuracy of the model on past data points. In those cases, so-called leave-future-out (LFO) CV (Bürkner et al., 2019) is a data split which makes more sense. However, since we are comparing different models fitted to fully observed microlensing events, this is not something we really care about. We care about the predictive performance of the model between the observed points (especially in the informative parts of the light curve) which is captured in the LOO-CV score $\text{elpd}_{\text{psis-loo}}$[8].

To compute the $\text{elpd}_{\text{psis-loo}}$ for each of the four modes in the posterior I use the function arviz.loo from the ArviZ library (Kumar et al., 2019), which implements the Pareto smoothing procedure. The results are shown in Table 4.4. The table contains the mean χ^2 across all

[8] Aki Vehtari makes this point in this blog post: https://statmodelling.stat.columbia.edu/2018/08/03/loo-cross-validation-approaches-valid/.

samples in each mode, the $\text{elpd}_{\text{psis-loo}}$ estimate, the difference in $\text{elpd}_{\text{psis-loo}}$ relative to the top performing mode (zero for the mode with the largest $\text{elpd}_{\text{psis-loo}}$, and the standard error for that difference[9]. The first thing to notice is that the ordering of the four modes based on the LOO-CV scores (where a higher score is better) is the same as the ordering of the χ^2 values. This is not a given because these two quantities measure different things. χ^2 is simply the value of the likelihood, which measures how well a particular mode fits the data, *assuming that the model is correctly specified* (which is never the case in reality). The LOO-CV score, on the other hand, is a measure of the predictive performance of each mode which is more robust to model misspecifications.

Name	χ^2	$\text{elpd}_{\text{psis-loo}}$	$\Delta(\text{elpd}_{\text{psis-loo}})$	$\text{se}(\Delta(\text{eldp}_{\text{psis-loo}}))$
Mode 1	949.4	-1409.89	0	0
Mode 2	951.8	-1410.81	0.92	2.03
Mode 3	961.0	-1415.97	6.07	4.34
Mode 4	969.5	-1420.54	10.65	6.236

Table 4.4: Differences in χ^2 and the LOO-CV scores between the four modes in the posterior. $\Delta\text{elpd}_{\text{psis-loo}}$ is the difference in $\text{elpd}_{\text{psis-loo}}$ relative to the first mode with the largest $\text{elpd}_{\text{psis-loo}}$. $\text{se}(\Delta\text{elpd}_{\text{psis-loo}})$ is the standard error for the difference.

Since Equation 4.13 is an approximation of the true elpd we need to keep in mind the variance of the estimate. Sivula et al. (2020) discusses how to assess the uncertainties in the LOO-CV score. To ensure that the standard error of the LOO-CV score or the standard error for the difference in LOO-CV scores is reliably estimated, the MCMC sampling has to have converged and the model should not be completely misspecified. If one model is clearly superior to all the others the other models can be safely discarded, but if the LOO-CV scores are comparable, then Sivula et al. (2020) suggests averaging over the models instead (a subject we shall cover shortly). As a rule of thumb, if the difference in LOO-CV scores between two models is smaller than about 5, the two models have roughly the same predictive performance. This is the case for the first two modes in the inferred posterior distribution of our microlensing model. The difference between the third and the first mode appears to be significant, but the standard error on the difference is still large, meaning that we probably shouldn't discard that mode. Mode 4 seems to be significantly worse than all the others.

4.4.2 PSIS-LOO as a powerful model diagnostics tool

To gain further insight into the meaning of the LOO-CV score, we can visualise the pointwise LOO-CV scores, $\ln p\left(f_i|\mathbf{f}_{-i}, \mathcal{M}\right)$ (the term in the sum in Equation 4.13), and the Pareto shape parameters $\hat{k}$. In Figure 4.4 I plot the posterior flux samples for the most predictive and the least predictive modes in the posterior (modes 1 and 4 respectively, top row). I also plot the residuals with respect to the median of those flux samples (second and third rows). In the second row, I colour the residuals by the pointwise LOO-CV score. This tells us *how well the model predicts the i-th point conditional on all the other points in the light curve.* In the third row, I colour the same residuals using the $\hat{k}$ values which tell us *how influential*

[9]See Vehtari et al. (2016) for how to compute the standard errors for the value of and difference between $\text{elpd}_{\text{psis-loo}}$.

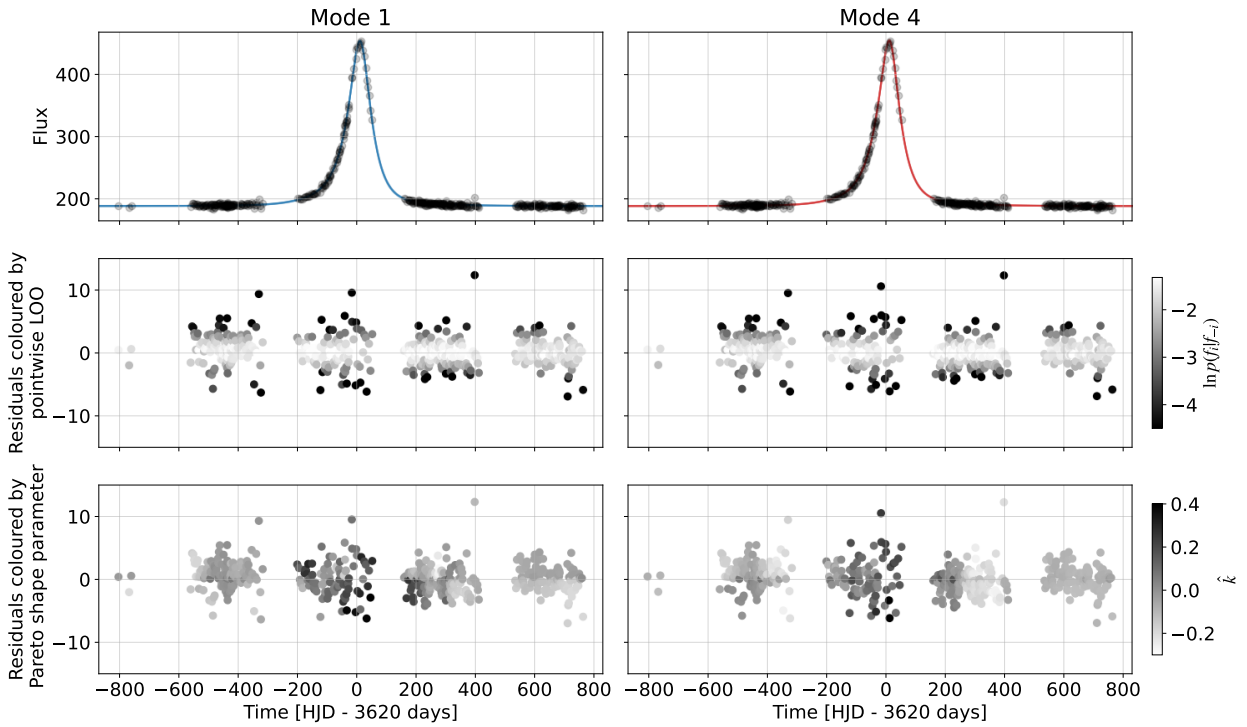


Figure 4.4: Posterior predictions for the observed flux for modes 1 and 4 in the posterior (first row). Each line is a single posterior sample for the predicted flux evaluated on a dense grid in time. Predictions are similar within modes so the lines are all nearly on top of each other. The second row shows the residuals with respect to the median flux prediction coloured by the pointwise LOO-CV score which quantifies how likely each point is under the model. The bottom row also shows the residuals except they're coloured by the inferred shape parameter $\hat{k}$ of the Pareto distribution fitted to importance weights when computing the LOO-CV scores. The $\hat{k}$ values measure how influential each point is on the posterior distribution. The points which are most influential in this case are not those which are most unlikely under the model.

the i-th point is on the posterior distribution $p(\boldsymbol{\theta}|\mathbf{f})$. It is not necessarily the case that those data points which are not well predicted by the model are also the most influential on the posterior distribution over the parameters. Indeed, this is exactly what we see in Figure 4.4. The data points which are not well predicted by the model are the three outlier points and several other points that are far away from the mean. For outlier points, this is expected because they are obviously inconsistent with a Gaussian noise model. The other points have low scores probably because the error bars are slightly underestimated.

If we now focus on the Pareto shape parameters, we see that most of the points that are not well predicted by the model are not particularly influential on the posterior. This is good news, it means that our misspecified noise model (an independent Gaussian noise model with errorbars generated by the OGLE pipeline) is not having a large impact on the posterior distribution over the parameters. The points which are influential (with $\hat{k} \gtrsim 0.4$) are understandably those around the tails, and the peak of the magnification.

Even with these diagnostic tools, it is hard to spot obvious differences between the modes in the data space. One other plot we can make is a pointwise version of the difference in the LOO-CV scores with respect to the best-performing model. Figure 4.5 shows the light curve (top) and the difference in pointwise LOO-CV scores relative to the score for Mode 1 (bottom). Positive values indicate that the particular model is more predictive for that point than Mode 1. We see a large scatter for modes 3 and 4 in both directions and no clear

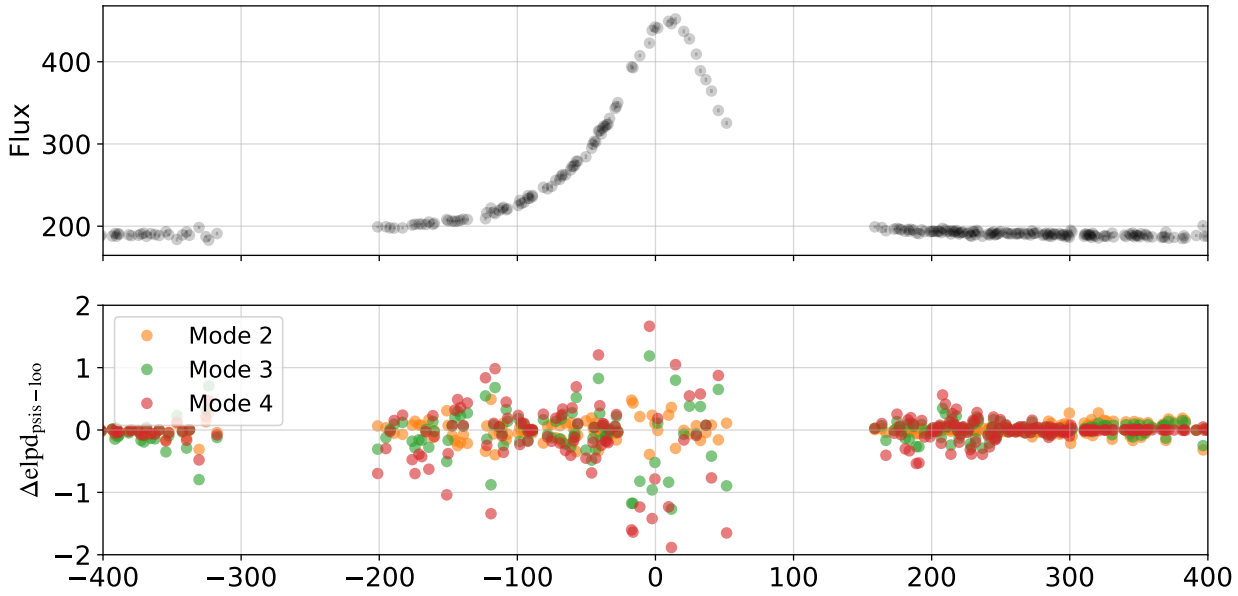


Figure 4.5: The difference in pointwise LOO-CV scores relative to most predictive mode (Mode 1) (bottom panel). The top panel shows the light curve for reference.

sign of bias in the predictive accuracy relative to Mode 1. This means that modes 3 and 4 are only worse than modes 1 and 2 because of a larger scatter in the pointwise LOO-CV scores – there are more points that those models predict less accurately compared to modes 1 and 2.

These examples only scratch the surface of possible applications of pointwise LOO-CV scores and Pareto shape parameters. In addition to being applied for checking the relative predictive power of different modes within one posterior density, LOO-CV can be applied to compare completely different models, for example, comparing single lens microlensing to binary lens microlensing. In binary lens microlensing, completely different physical models differ between each other only at a few critical points near the caustic crossing (see for example Figure 1 in Hwang et al., 2018). LOO-CV scores have the potential to substantially improve model comparison in microlensing.

4.4.3 Estimating LOO-CV with Gaussian Process models

Equation 4.13 is straightforward to evaluate if the model likelihood factorizes into a product over data points $p(\mathbf{y}|\boldsymbol{\theta}) = \prod_{i=1}^{N} p(y_i|\boldsymbol{\theta})$. That is the case if the data covariance matrix $\mathbf{C}$ is assumed to be diagonal. However, this is an extremely restrictive assumption for a realistic model of a microlensing light curve. A natural extension of the model would be to include a Gaussian Process (GP) as a model for the off-diagonal terms in the covariance matrix. Golovich et al. (2022) showed that including a GP when modelling OGLE single lens events has a significant impact on the on inferred parameters, particularly the event timescale t_E. I briefly describe how the LOO-CV score can be generalised to these kinds of models, although I do not use a GP in this work.

For a multivariate Gaussian likelihood with a dense covariance matrix, we can com-

pute the predictive distribution $p(y_i|\mathbf{y}_{-i})$ in the LOO-CV Equation 4.12 using the following expression (Bürkner et al., 2018)

$$p(y_i|\mathbf{y}_{-i}) = \mathcal{N}\left(y_i|\tilde{\mu}_i, \tilde{\sigma}_i\right) \quad , \tag{4.14}$$

where

$$\tilde{\mu}_i = y_i - \frac{g_i}{\bar{\sigma}_{ii}}, \quad \tilde{\sigma}_i = \frac{1}{\bar{\sigma}_{ii}} \quad , \tag{4.15}$$

and

$$g_i \equiv \left[\mathbf{C}^{-1}(\mathbf{y} - \boldsymbol{\mu})\right]_i, \quad \bar{\sigma}_{ii} \equiv \left[\mathbf{C}^{-1}\right]_{ii} \quad . \tag{4.16}$$

Equation 4.14 can then be used to compute the pointwise and total LOO-CV scores as before. Evaluating Equation 4.14 requires that we compute the inverse data covariance matrix $\mathbf{C}^{-1}$ once for all the points in the light curve. If we use a fast 1D GP library such as celerite (Foreman-Mackey et al., 2017), the covariance matrix inversion can be done in linear time with respect to the number of data points.

4.5 Combining information from multiple modes

In Section 4.4 I have described how to compare the different models using cross-validation. If there are multiple competing modes (as is the case for this particular event) and we want to propagate the uncertainties for the model parameters, then we have to combine (re-weightt) the MCMC samples from each of the modes. The first option would be to ignore the LOO-CV scores and re-weight the samples from K in such a way that the combined samples are faithful samples of the multi-modal posterior distribution (Equation 4.6). This is known as *Bayesian Model Averaging* (BMA) and it is equivalent to weighting the different modes in proportion to the Bayesian evidences $\mathcal{Z}_k \equiv p(\mathbf{y}|\mathcal{M}_k)$ for each mode $\mathcal{M}_k$. It is also what we would obtain if we used Nested Sampling (see Section 2.4.4) to sample the multi-modal posterior.

There are many issues with BMA/Bayes weights. Most notably, they are sensitive to priors (see discussion in Section 2.4.5), they are not robust to model misspecification, and they are expensive and difficult to compute. Yang and Zhu (2018) also find that, in the context of a phylogenetics model, BMA can assign weight 1 to the model that is closest to the true data generating process as measured by KL divergence *even if the other models are only slightly worse*. One solution to this problem is to replace BMA with so-called *pseudo-BMA* (*Pseudo Bayes factors*) (Geisser and Eddy, 1979). The idea is to replace the Bayes evidences $\mathcal{Z}_k$ with a product of LOO predictive densities $\prod_{i=1}^{n} p(y_i|\mathbf{y}_{-i}, \mathcal{M}_k)$. Yao et al. (2018a) propose replacing the exact LOO predictive densities with the reliable $\text{elpd}_{\text{psis-loo}}$ estimators in a scheme called *Pseudo-BMA*. The weights for each mode k are then

$$w_k = \frac{\exp\left(\text{elpd}^{(k)}_{\text{psis-loo}}\right)}{\sum_{k=1}^{K} \exp\left(\text{elpd}^{(k)}_{\text{psis-loo}}\right)} \quad . \tag{4.17}$$

The above expression does not take into account the variance of the LOO-CV estimators. To remedy this, Yao et al. (2018a) propose using the bayesian bootstrap (Rubin, 1981; Vehtari

and Lampinen, 2002) to construct weights which also use the information about the variance of the LOO-CV estimates. The resulting weights are usually called *Pseudo-BMA+* weights. Adding the variance information has the effect of pushing the weights away from 0 and 1 if there are several similar models. The Pseudo-BMA+ weights are similar to classic BMA, but they have all the desirable properties of LOO-CV that we have discussed in the previous section.

Finally, there is one other, conceptually quite different approach to computing the weights that was recently proposed by Yao et al. (2020). The idea is to stack the chains in order to optimize the joint predictive performance of the combined model. The *stacking weights* are

$$w^*_{1,\dots,K} = \text{argmax}_{\mathbf{w}} \sum_{i=1}^{n} \ln \sum_{k=1}^{K} w_k p_k \left(\tilde{\mathbf{y}}_i | \mathbf{y}_{-i}\right) + \ln p_{\text{prior}} \left(\mathbf{w}\right) \quad , \tag{4.18}$$

where $p(\tilde{\mathbf{y}}_i|\mathbf{y}_{-i})$ is computed using Pareto smoothed importance sampling, $\mathbf{w}$ is the simplex vector of model weights and $p_{\text{prior}}\left(\mathbf{w}\right)$ is prior regularisation for the weights. The stacking weights answer the question: *which model weights result in the best leave-one-out cross-validation performance of the distribution formed by the weighted average of the samples from each chain?*. Yao et al. (2020) demonstrate that stacking outperforms both BMA and Pseudo-BMA+ in terms of predictive performance on a variety of problems.

Before discussing their merits, I test each approach on the problem at hand. First, to obtain samples from the true Bayesian posterior (which is equivalent to weighting samples from each mode using BMA), I use the Nested Sampling algorithm from the UltraNest package (Buchner, 2021a). UltraNest is a generalisation of the original NS algorithm devised by Skilling (2004). It implements the MLFriends algorithm described in Buchner (2016, 2019) which is more robust when it comes to posterior sampling than the algorithms used in the popular MultiNest (Feroz et al., 2009a), and dynesty (Speagle, 2020) packages.

Weights	Pseudo-BMA	Pseudo-BMA+	Stacking
Mode 1	0.713005	0.561651	0.000000
Mode 2	0.285338	0.346164	0.678198
Mode 3	0.001640	0.067947	0.000000
Mode 4	0.000017	0.024238	0.321802

Table 4.5: Three kinds of weights for the four modes. Pseudo-BMA weights are simply the exponentiated LOO-CV scores. Pseudo-BMA+ weights are the same as Pseudo-BMA except they take into account the variance of the LOO-CV estimators. Stacking weights are maximize the joint predictive performance of the model.

I run UltraNest with default settings until convergence. To re-weigh the MCMC samples from each chain using the Pseudo-BMA+, and stacking weights, I use the function arviz.compare from the ArviZ library which implements both algorithms. The resulting weights are shown in Table 4.5. Figure 4.6 shows the re-weighted MCMC samples from each of the four modes, and the samples obtained using NS. The four separate MCMC chains were re-weighted following the procedure described in Yao et al. (2020)).

There are major differences between the different weighting schemes. First of all, the regular Pseudo-BMA weights, and the implicit BMA weights used in Nested Sampling assign

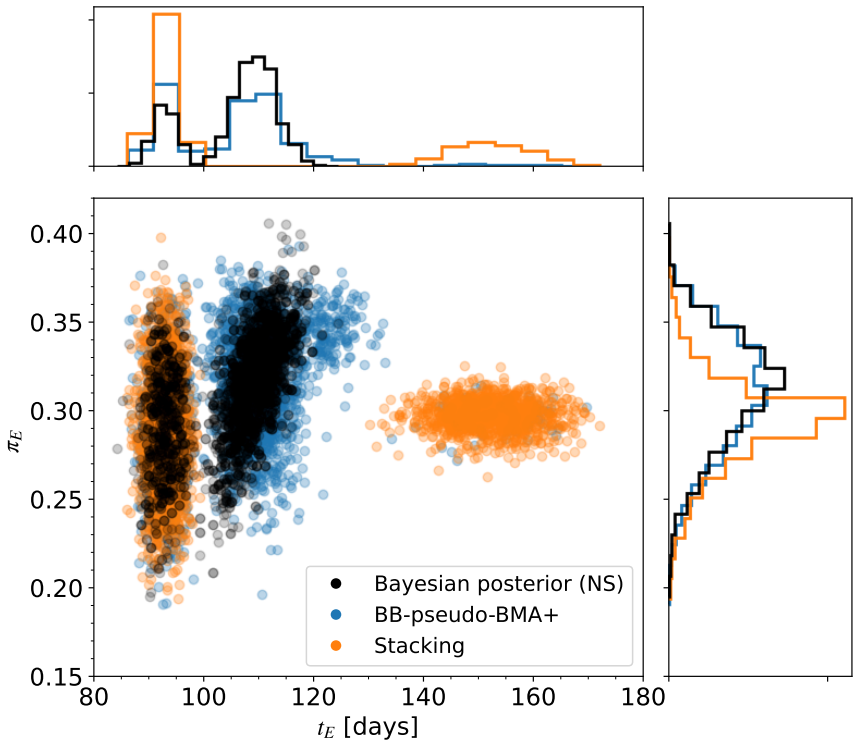


Figure 4.6: Samples from the posterior distribution (black) and the weighted average of samples from each mode. The weights in the latter case are given by Pseudo-BMA+ (blue) and Stacking (orange). There are major differences between each of the three approaches.

near zero weight to Mode 3 and Mode 4, thus discarding the possibility of a long ($t_E \gtrsim 120$ days) event timescale. The Pseudo-BMA+ weights which take into account the variance in the $\text{elpd}_{\text{psis-loo}}$ estimators are more conservative. They assign some, albeit small, weight to the third and fourth modes. They also decrease the weight of the first mode relative to the second. Finally, stacking weights completely zero out the first and the third mode assigning comparable weights to the second and the fourth mode. This is expected because when two models have similar predictive power, stacking will select one of the models and discard the other.

So which approach should we use? I don't believe there is a single best answer to this question in all circumstances but I would argue that the Pseudo-BMA+ weights have the most desirable properties, while BMA and stacking have undesirable properties. The desirable property of the Pseudo-BMA+ weights is that, since the weights are based on the LOO-CV scores which capture the predictive power of the model in a LOO sense, they are less sensitive to model and prior misspecification. For instance, if we had a situation where the relative height between the modes was particularly sensitive to a small number of data points the Pseudo-BMA+ weights would take into account this sensitivity while BMA would not. The stacking weights on the other hand have the undesirable property of discarding a perfectly good model if it has a similar predictive power to another model. This is definitely something we don't want to do because it does not make sense when we are interested in accounting for all plausible models, rather than optimising for the predictive power of some weighted sum of all models. However, it is interesting that stacking does not discard mode

4 because it improves prediction performance when combined with Mode 1. It is an open research question whether this somehow provides more credence to Mode 4 relative to what the other weighting options do.

4.6 Search and computation

In the previous sections, I have described a method for weighing MCMC chains stuck in multiple modes based on the predictive performance of each mode but I have not mentioned how to find the modes in the first place. This problem does not have a general solution (see the No Free Lunch theorems described in 2.4). Nested Sampling seems to do a good job at discovering the different modes on this problem (modes 3 and 4 are also discovered but they have negligible weights so they are not visible in Figure 4.6), but it is very computationally intensive. Fitting this simple 5-parameter model takes more than an hour on a 2021 M1 Macbook Pro laptop, even with a vectorized and parallelised likelihood function evaluation. NUTS initialised at the four modes takes around 10-20 minutes per chain (chains are run in parallel) to reach an ESS of at least 400 across all parameters. Neither of these options is ideal because we want to be able to fit the model to thousands of light curves.

Fortunately, there is a better way of finding the modes while at the same time approximating the posterior distribution using a multivariate Gaussian distribution. This is a strategy proposed by Yao et al. (2020). The first step is to run a BFGS optimizer initialised at multiple points in the parameter space in order to find all significant modes. The second step is to use the Laplace approximation (inverse Hessian of the posterior evaluated at the local minimima) to obtain an approximation of the full posterior distribution at these modes. Finally, we compute the LOO-CV scores for each posterior mode. At the same time we get a sense of the uncertainty in the posterior approximation through the use of the PSIS $\hat{k}$ diagnostic.

Parameter	t'_0	t'_E	u_0	$\pi_{E,N}$	$\pi_{E,E}$
Prior	$\mathcal{N}(t'_{0,\text{estimate}}, 10)$	$\mathcal{U}(20, 400)$	$\mathcal{U}(-1, 1)$	$\mathcal{N}(0, 0.1)$	$\mathcal{N}(0, 0.1)$

Table 4.6: Prior distributions that used for initialising the BFGS optimizer.

To test this strategy, I first initialise the BFGS optimizer at 20 different locations in the parameter space which are sampled from the prior distributions listed in Table 4.6. These priors are slightly different from the ones used in the posterior distribution defined in Equation 4.6. The idea is to maximize the chances of finding all of the modes using prior knowledge about the problem. For single lens parallax models, this is relatively straightforward because the different modes are well separated in the u_0 parameter. I run the BFGS optimizer for 5 iterations although it reliably converges for every mode already in 1-2 iterations. I parallelise the optimisation runs from multiple starting points using JAX. All modes are easily discovered and the entire optimisation process takes less than a minute on a 2021 M1 Macbook Pro laptop which is more than an order of magnitude faster than NS or NUTS MCMC. I then evaluate the inverse Hessian at the optima points using numpyro (which uses jax.hessian under the hood) and obtain a multivariate Gaussian approximation for each of the

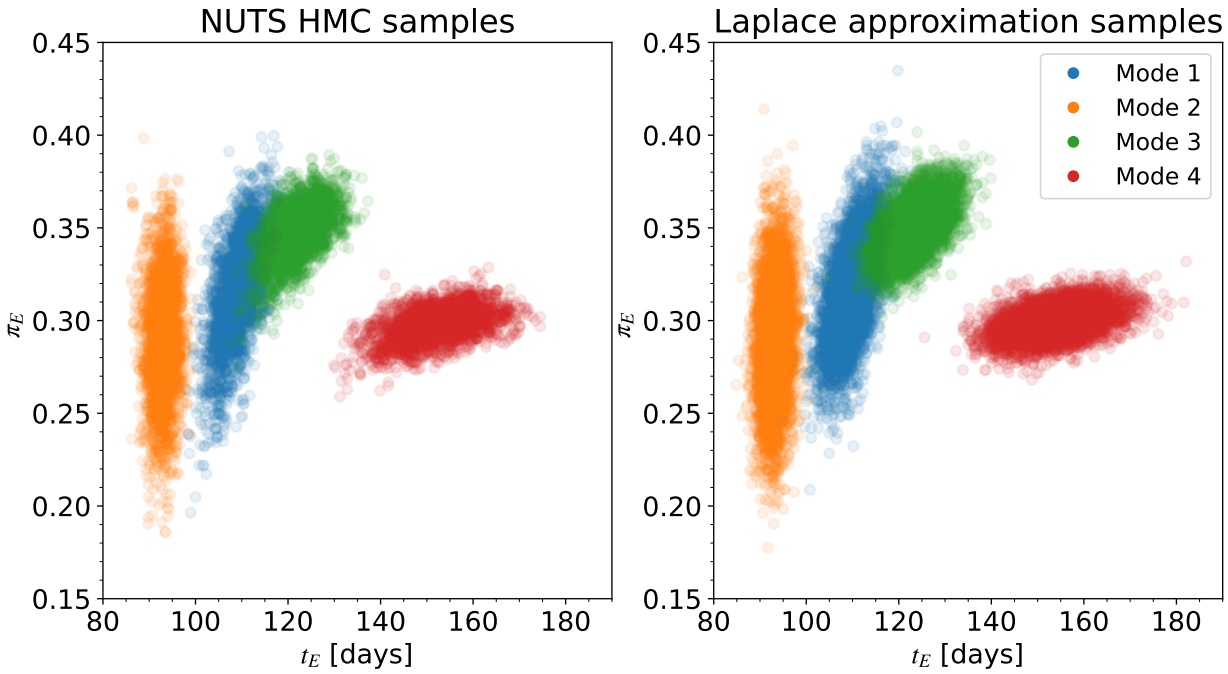


Figure 4.7: Laplace approximation to the true posterior for each of the four modes.

posterior modes. Figure 4.7 shows the samples from the Laplace approximation compared to the true posterior (right) next to the samples from the true posterior obtained using NUTS (left). It seems like the Laplace posterior is a very good approximation to the true posterior because each of the modes is well described by a multivariate Gaussian distribution.

The next step is to evaluate the LOO-CV score for the Laplace posterior approximation at each mode. Because the samples we obtain using the Laplace approximation are samples from a multivariate normal approximation to the posterior, rather than the true posterior, we cannot naively apply Equation 4.13 to compute the LOO-CV score. Magnusson et al. (2019) proposes a modification to the importance sampling weights in Equation 4.13 which is applicable to approximate inference methods such as the Laplace approximation and variational inference (VI). The modification is very straightforward. If $p(\boldsymbol{\theta}|\mathbf{y})$ is the true posterior and $q(\boldsymbol{\theta}|\mathbf{y})$ is the approximate posterior (in our case, a multivariate Gaussian evaluated at the different optima), the importance sampling weights for the LOO posterior should be modified to take into account the discrepancy between the true posterior and the approximate posterior, as follows

$$w\left(\boldsymbol{\theta}^{(s)}\right) \propto \frac{1}{p\left(\mathbf{y}_i|\boldsymbol{\theta}^{(s)}\right)} \frac{p\left(\boldsymbol{\theta}^{(s)}|\mathbf{y}\right)}{q\left(\boldsymbol{\theta}^{(s)}|\mathbf{y}\right)} \quad . \tag{4.19}$$

The importance weights are again stabilised using Pareto smoothing, but the Pareto shape parameters now also tell us if the Laplace approximation is a good approximation for the full posterior (see also Yao et al., 2018b) which is very useful.

The LOO-CV scores that were computed using the modified weights defined in Equation 4.19 are listed in Table 4.7 together with the mean Pareto $\hat{k}$ parameters for each mode which can be used to judge the quality of the Laplace approximation for each mode. The

Name	$\text{elpd}_{\text{psis-loo}}$	$\Delta(\text{elpd}_{\text{psis-loo}})$	$\text{se}(\Delta(\text{eldp}_{\text{psis-loo}}))$	$\text{mean}(\hat{k})$
Mode 1	-1409.54	0	0	0.46
Mode 2	-1410.68	1.13	2.02	0.57
Mode 3	-1415.63	6.09	4.32	0.64
Mode 4	-1420.54	10.57	6.22	0.70

Table 4.7: Similar to Table 4.4, except the importance weights for the LOO posterior predictive distribution were computed using samples from the multivariate Gaussian approximation to the posterior (Laplace approximation). The last column lists the mean values (across all data points) of the Pareto shape parameters $\hat{k}$. The estimated LOO-CV scores are nearly identical to those listed in Table 4.4 despite the fact that $\hat{k}$ for modes 3 and 4 is not ideal.

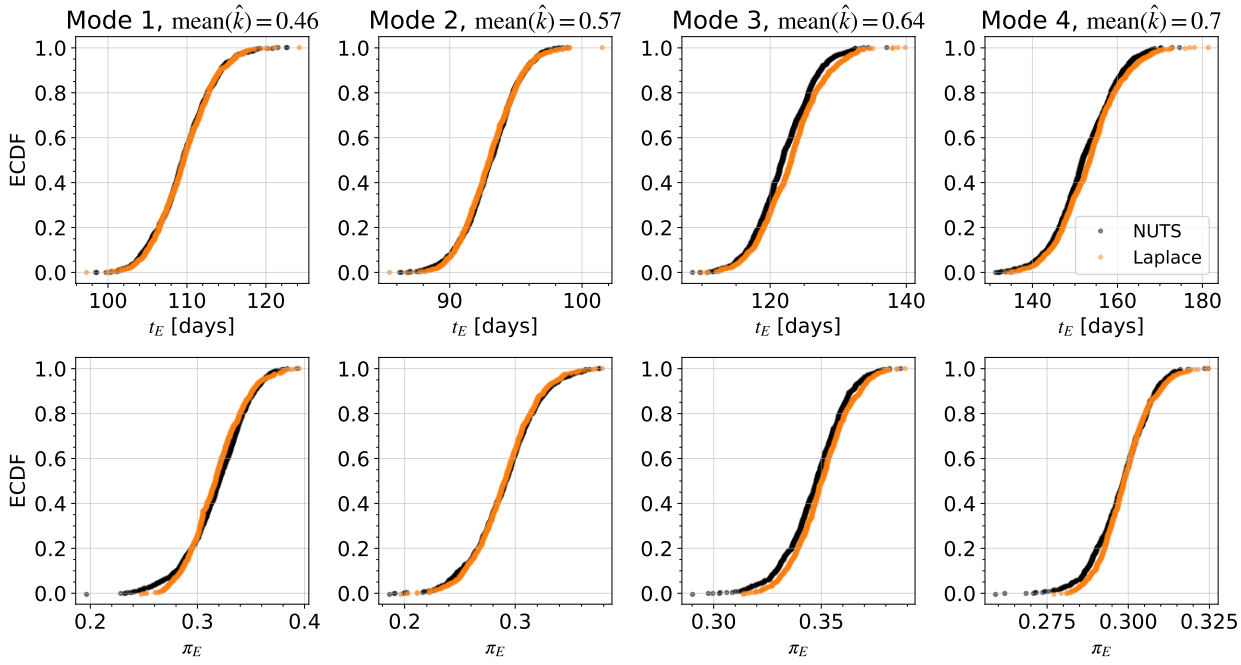


Figure 4.8: Comparison between the ECDFs for the MCMC samples from the true posterior and samples from the Laplace approximation to the true posterior for two parameters of interest, t_E and π_E. The two sets of samples have very similar shapes.

mean $\hat{k}$ parameters for modes 2-4 is potentially not reliably (because $\hat{k} > 0.5$). Nevertheless, the estimated LOO-CV scores are very close to those obtained from full MCMC sampling of the true posterior (see Table 4.4). To gain more information about the quality of the Laplace approximation I also compute the empirical cumulative distribution functions (ECDF) for the marginal distributions of interest (pdfs for t'_E and π_E) for both the MCMC samples and the samples from the Laplace posterior. Plots comparing the two ECDFs are shown in Figure 4.8. Overall, the match between the two distributions is quite good. Using the Laplace approximation for this particular model and dataset would not appreciably change the marginal distributions of interest. The mean Pareto $\hat{k}$ parameters seem to capture the quality of the Laplace approximation well because the match between the ECDFs for the marginal distributions worsens with increasing $\hat{k}$.

In summary, I propose the following strategy for fitting single lens microlensing events:

1. Initialise the BFGS optimizer at $N_{\rm init}$ locations in the parameter spaces sampled from a prior.

2. Run BFGS optimizer in parallel to discover all plausible modes in the posterior. Discard obviously bad modes with completely unphysical values of the parameters and remove duplicate solutions.

3. Compute the LOO-CV scores for the rest of the modes using PSIS with the importance weights given by Equation 4.19. Save the $\hat{k}$ parameters which quantify the quality of the Laplace approximation at each mode.

4. For those modes for which $\hat{k}$ is large, run NUTS sampling, initialised at a single sample from the Laplace posterior for that mode[10].

5. Compute Pseudo-BMA+ weights and re-weight the samples from each mode.

A few comments on the above strategy. In Step 1 $N_{\rm init}$ should be sufficiently large to minimise the chances of missing a mode. In Step 2 we discard duplicate solutions because Pseudo-BMA+ weights would split the weight between each duplicate solution which is undesirable. Step 4 is optional depending on whether we are in the exploratory modelling phase or if we want publication-ready results. For optimal hardware utilisation, in Step 4 it makes sense to first fit the entire sample of events using the Laplace approximation, and then afterwards run the NUTS sampling for the worst modes that were previously identified in Step 3.

This procedure is orders of magnitude faster than alternatives. In the best-case scenario where the Laplace approximation is sufficient for every event, we could fit 1000 single lens light curves using a single laptop in less than 24 hours (assuming that BFGS optimisation + LOO-CV computation takes about a minute per light curve). In fact, because the magnification computation and the BFGS optimisation for the single lens model can be easily parallelised and executed on a GPU (or multiple GPUs in parallel) using JAX, it is possible that we could speed this up even more. For a very rough comparison, (Golovich et al., 2022) used 1 million CPU hours (!) on a Lawrence Livermore National Laboratory supercomputer to fit 10000 single lens OGLE events. Although they also used a Gaussian Process model for the noise, which increases the computational cost by an order of magnitude, they did not account for the multi-modality of the posterior so in that sense the analysis is flawed.

If we were to fit 10000 single lens events with Nested Sampling this would also cost at least a million CPU hours. For reference, the Roman telescope is expected to discover over 50,000 microlensing events (Johnson et al., 2020), the vast majority of which are going to be single lens events. This goes to show that MCMC (even the state-of-the-art NUTS sampler) is extremely inefficient compared to optimisation. *It should be avoided at all costs in those circumstances where it is not necessary* (for example if the Laplace approximation to the full posterior is sufficiently good, as judged by the $\hat{k}$ diagnostic).

[10]The reason it is better to initialise the MCMC chains a location of a single sample from the Laplace approximation to the posterior rather than the optimum is that that the optimum is far from the typical set (see Section 2.4.2)

4.7 Summary and future work

In this chapter, I studied different approaches to fitting single lens microlensing events with degenerate solutions using a specific event, OGLE-2017-BLG-1190Lb, as a case study. The degeneracies are reflected in the multi-modal nature of the posterior distribution They are physically important because ignoring some of the modes leads to flawed inferences about the physical parameter of interest, the parallax magnitude π_E and the event timescale t_E.

I considered two different approaches to solving this problem. The first approach is to fit the model using Nested Sampling which generates samples from the multi-modal posterior. The second approach involves running multiple MCMC chains in parallel, allowing each to converge to one of the modes, and then combining the samples using the leave-one-out cross-validation (LOO-CV) scores for each chain to re-weight the samples. I argue that the latter method is superior to sampling the Bayesian posterior directly[11] because it is more robust to model misspecification and situations where the posterior distribution is particularly sensitive to only a few data points in the light curve. It is also computationally much more efficient than using Nested Sampling. Fitting multiple modes with MCMC is about an order of a magnitude faster than Nested Sampling. In cases when each of the posterior modes can be approximated with a multivariate Gaussian distribution, we can avoid doing MCMC and fit each of the modes using the Laplace approximation which is orders of magnitude faster still. The Laplace approximation method can be used to fit many thousands of microlensing events in a short time span, without ignoring the degenerate solutions.

Besides being useful for weighting degenerate solutions, cross-validation can be used for assessing the fits of individual models. I showed how one can use pointwise LOO-CV scores and the Pareto shape parameters to find influential data points in the light curve, diagnose problems with the model, and validate the validity of the Laplace approximation to the true posterior distribution.

[11] It also sort of encompasses the Nested Sampling approach if we use the non-bootstrapped Pseudo-BMA weights.

Chapter 5

Modelling multiple-lens events

In this chapter, I identify some important issues with current approaches to modelling multiple-lens microlensing events. The chapter is short and the work presented here is preliminary but it is, in my opinion, the most important part of this thesis.

5.1 The fundamental problem with modelling microlensing events with $N > 1$ lenses

As in the previous chapter, I use a single microlensing event light curve as a testbed for exploring different aspects of the modelling problem. The dataset I chose is the OGLE Early Warning System (EWS) light curve for the event OGLE-2016-BLG-0039. The light curve is shown in Figure 5.1. It is clear that this event is most likely a caustic-crossing binary, or perhaps a triple-lens event. This particular dataset has been previously fit with an extended source binary lens model (including and excluding parallax) using the automated system RTModel.[1] A more complete anlysis of this event, which also used used KMT and MOA observations in addition to OGLE, was published in Han et al. (2018). Since the KMT and MOA data are not publicly available I only use the OGLE measurements for the analysis presented in this chapter. The goal is not to focus on this particular event, but rather, to use it as a case study.

I set up the coordinate system for the binary lens model such that the origin is at the centre of mass of the two lenses and the x-axis is aligned with the line connecting the two point masses. The first lens with mass fraction ϵ_1 is located at location $r_1 = -sq/(1+q)$ in the source plane, and the second lens is at $r_2 = s/(1+q)$, where $q \equiv \epsilon_2/\epsilon_1$ is the mass ratio of the two lenses, and $s \equiv 2a$ is the separation between the two lenses. I ignore parallax effects for simplicity and assume that the trajectory of the source is a straight line. The two components of the complex source trajectory $w(t) = w_1(t) + iw_2(t)$ are then

$$w_1 = u_0 \sin\alpha - \frac{t - t_0}{t_E} \cos\alpha \tag{5.1}$$

$$w_2 = u_0 \cos\alpha + \frac{t - t_0}{t_E} \sin\alpha \quad , \tag{5.2}$$

[1] http://www.fisica.unisa.it/gravitationAstrophysics/RTModel/2017/OB170039.htm

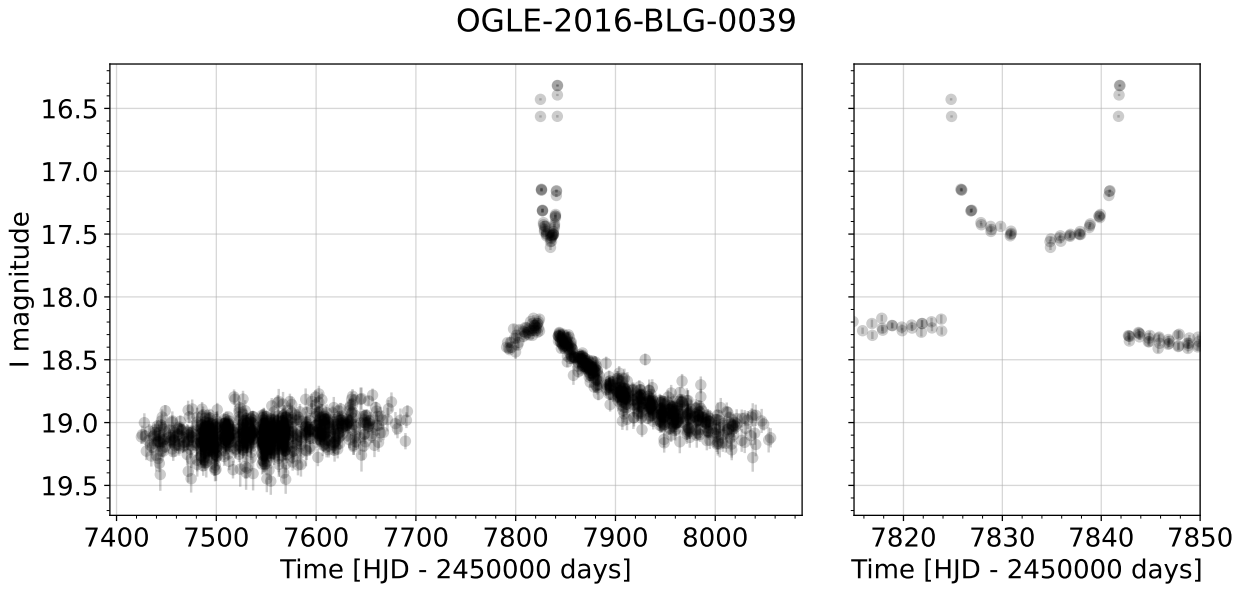


Figure 5.1: Light curve for the microlensing event OGLE-2016-BLG-0039 (left) and a zoomed-in view of the peak (right).

where $\alpha \in [0, 2\pi)$ is the angle between the position vector of the first lens and the source star velocity vector. This parametrisation is very common and it is also used in RTModel. The magnification can then be evaluated at every point w in the source plane for a source star with radius $\rho_\star$. The predicted flux is given by Equation 4.4, as in the case of a single lens event. The complete model thus has 7 (non-linear) parameters $(s, q, u_0, \alpha, t_0, t_E, \rho_\star)$.

Parameter	s	q	u_0	α	t_0 HJD - 2450000 days	t_E days	$\rho_\star$
Value	2.218	0.339	0.375	1.420	7847.5	241.9	0.000985

Table 5.1: A maximum likelihood solution obtained by RTModel.

I start by evaluating the model likelihood at the maximum likelihood solution found by RTModel. It is one of three local χ^2 minima reported by RTModel. RTModel uses a finite-difference gradient-based optimizer with a custom initialisation strategy to search for the global χ^2 (negative log-likelihood) minimum. The parameters are shown in Table 5.1. I evaluate two-dimensional slices of the likelihood by keeping all the other parameters fixed at the values given in Table 5.1, and varying two parameters at a time.

The results are shown in Figure 5.2. The left column shows a broad view of the likelihood surface while the right column shows a zoomed-in view. 1D likelihood slices passing through the minimum point in each 2D slice are shown on the top and the right side of each heatmap plot. The structure of the likelihood is striking. It is clearly multi-modal and non-smooth, about as different from a multivariate Gaussian as it can be. Note that these are just two-dimensional slices of the likelihood function. The true density is 7-dimensional so the problem is even worse than it appears. The fundamental cause for this non-smooth structure is that the caustic curves in the source plane ends up being imprinted onto the likelihood itself. The sharp kinks in the likelihood function surface are there because even small perturbations of

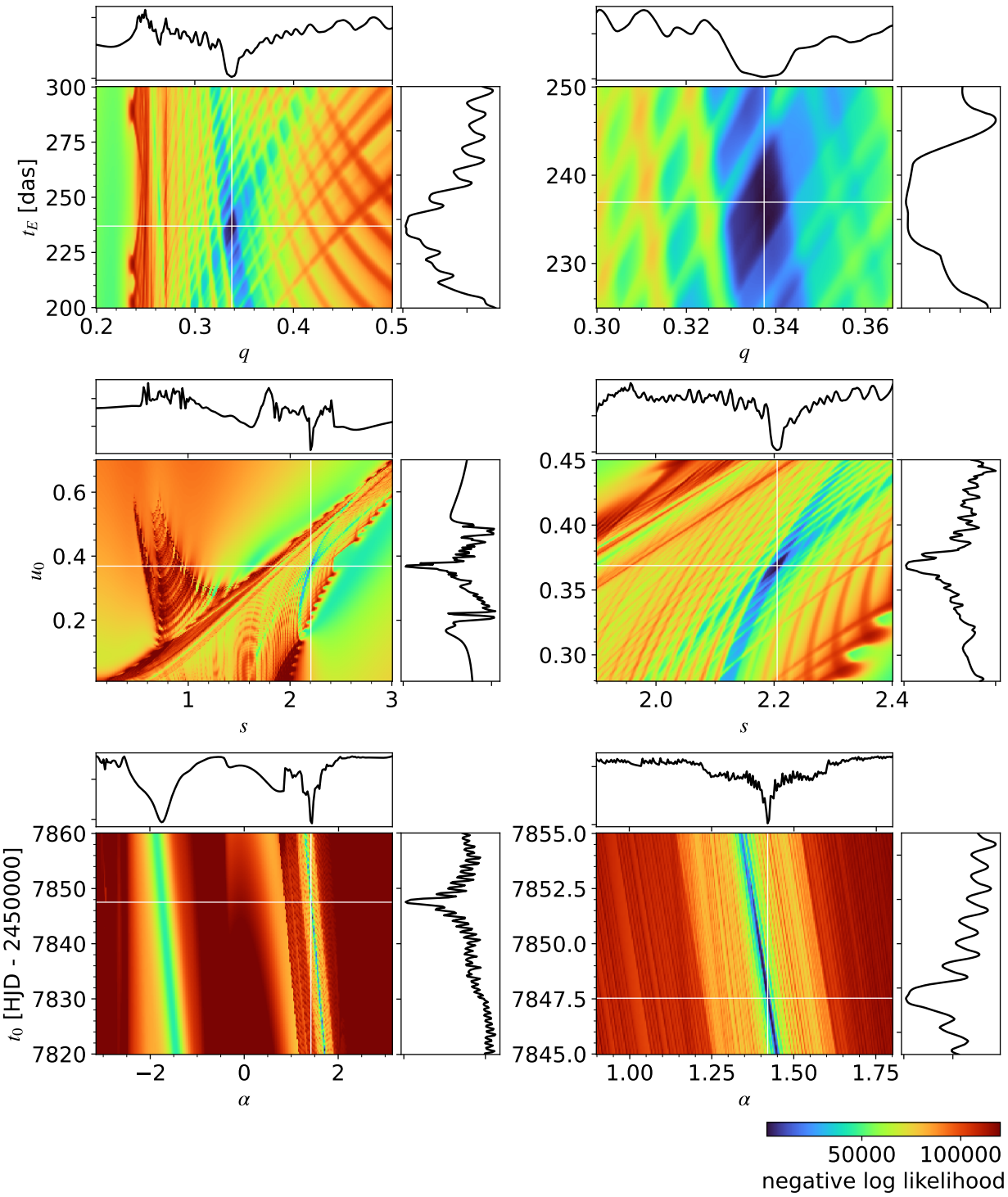


Figure 5.2: 2D slices through the likelihood function for a 7D binary lens model. The column on the right shows a zoomed-in view of the slices in the left column. The likelihood slices are evaluated at a local maximum. The light curve used to compute the likelihood is shown in Figure 5.1. The plot illustrates how the geometry of the likelihood function in caustic-crossing microlensing events is extremely complex.

the parameters can cause drastic changes in the predicted flux and only a very small volume

around a particular mode[2] in the parameter space fits the caustic-crossing features in the light curve well.

There are several important implications of Figure 5.2:

1. **Likelihoods (and posteriors) for multiple-lens microlensing events have an entirely different structure than those for single lens events.** In the latter case, the geometry of the likelihood is smooth, even though it can also be multi-modal and have high curvature. In the former case, the likelihood is non-smooth, highly multi-modal and has high curvature. This fundamental difference means that we should not expect that methods developed for single lens events will work with multiple-lens events.

2. **Assumptions that the vast majority of statistical methods rely on are in this case almost certainly violated**. Pretty much all of statistics is designed around the assumption that the likelihood function is smooth and not too different from a multivariate Gaussian distribution. State-of-the-art MCMC samplers can completely fail to sample even relatively simple posteriors if there are multiple modes or if there is high curvature in the target density. For this problem all of those things are true, but also, the likelihood is non-smooth. For instance, a method that is obviously inappropriate is the Laplace approximation because the posterior almost everywhere looks nothing like a Gaussian. Thus, the method I have introduced in Chapter 4 is not useful in the context of multiple-lens microlensing events except possibly for the purpose of some preliminary modelling.

3. **Local gradient is not informative.** There are many narrow modes in the likelihood. As a result, the local gradient at any given point in the parameter space is not particularly informative because it does not lead us to a global minimum when we are optimising the likelihood or the typical set when sampling the posterior using gradient-based samplers. In fact, the gradient can point in the direction opposite of where we want to go. This point was also made in the excellent thesis by Rajpaul (2012).

4. **More complex microlensing models are likely even worse**. The seven-parameter binary lens model without parallax is the simplest model for binary lens events. Including higher-order effects such as parallax and orbital motion can more than double the dimension of the parameter space and make the geometry of the problem even worse. Same goes for triple-lensing.

The second point implies that all the hard work done in Chapter 3 on building an automatically differentiable contour integration algorithm may have been a waste of time, at least in the context of caustic-crossing microlensing events. For non-caustic crossing events (most detected planetary events) the likelihood is at least locally smooth in the vicinity of the high-probability solution and gradient-based methods such as HMC may be more useful. Unfortunately I did not have time to investigate this hypothesis in this thesis.

[2]As in the previous chapter, I use the term mode to loosely refer to a high probability mass region in the parameter space (the typical set), or the MAP point, depending on the context.

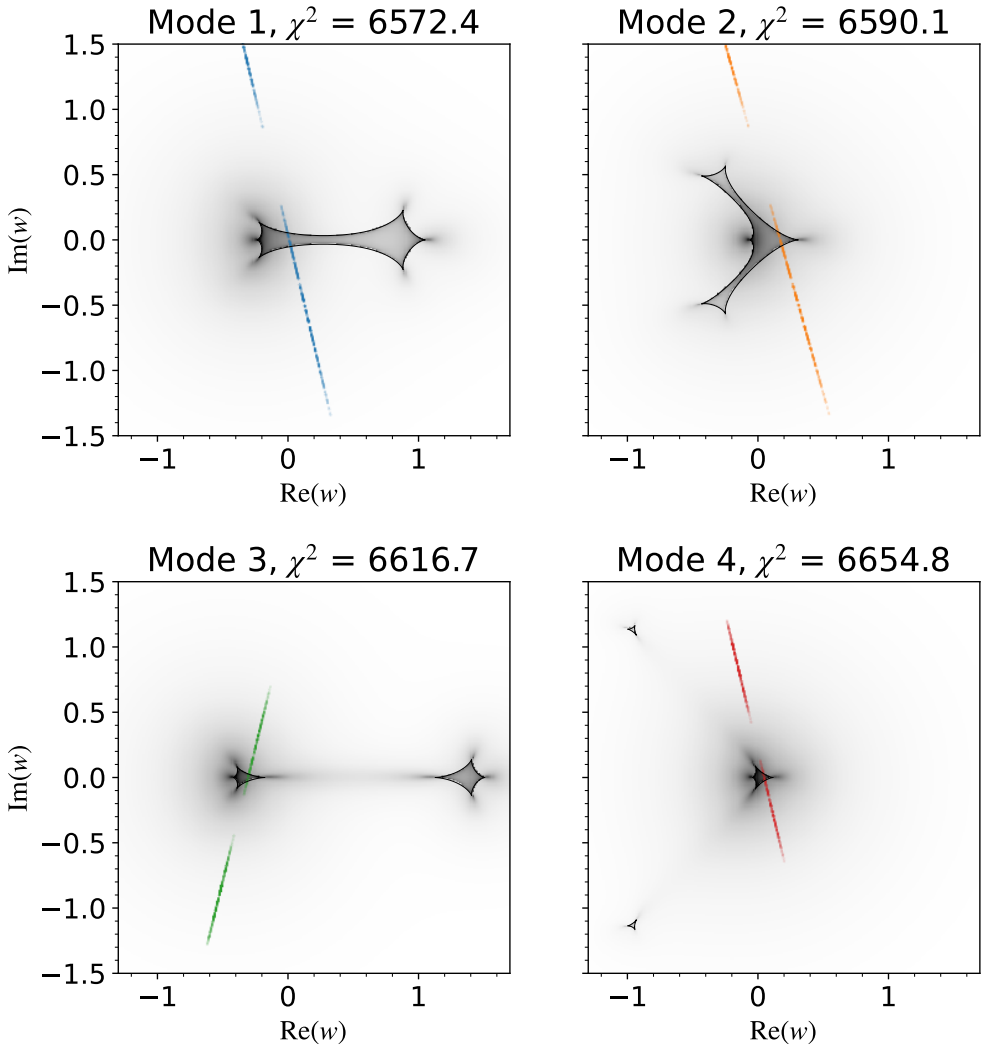


Figure 5.3: The geometry of the binary lens event for four different local maxima of the likelihood distribution. The coloured circles denote the source trajectory evaluated at the times of observations. The circle size is proportional to the value of $\rho_\star$ at each mode. The predicted fluxes for each mode in the likelihood are shown in Figure 5.4.

To illustrate the degeneracies in the parameter space for this model, in Figure 5.3 I plot the trajectory and caustic structure for four different modes in the likelihood. These are probably not the only significant modes in the likelihood. Three of the four modes (modes 2-4 in the figure) were discovered by the automated RTModel system, but the fourth, highest likelihood mode (Mode 1 in Figure 5.3) was not.[3] Figure 5.4 shows the predicted flux for each mode and the flux residuals. The first mode is dominant in terms of the likelihood value but we cannot really discard the other modes without a more sophisticated analysis using cross-validation scores. This requires having posterior samples.

In the following section, I discuss existing strategies for finding and exploring these modes.

[3]I found this mode using Nested Sampling, the subject of the Section 5.3.

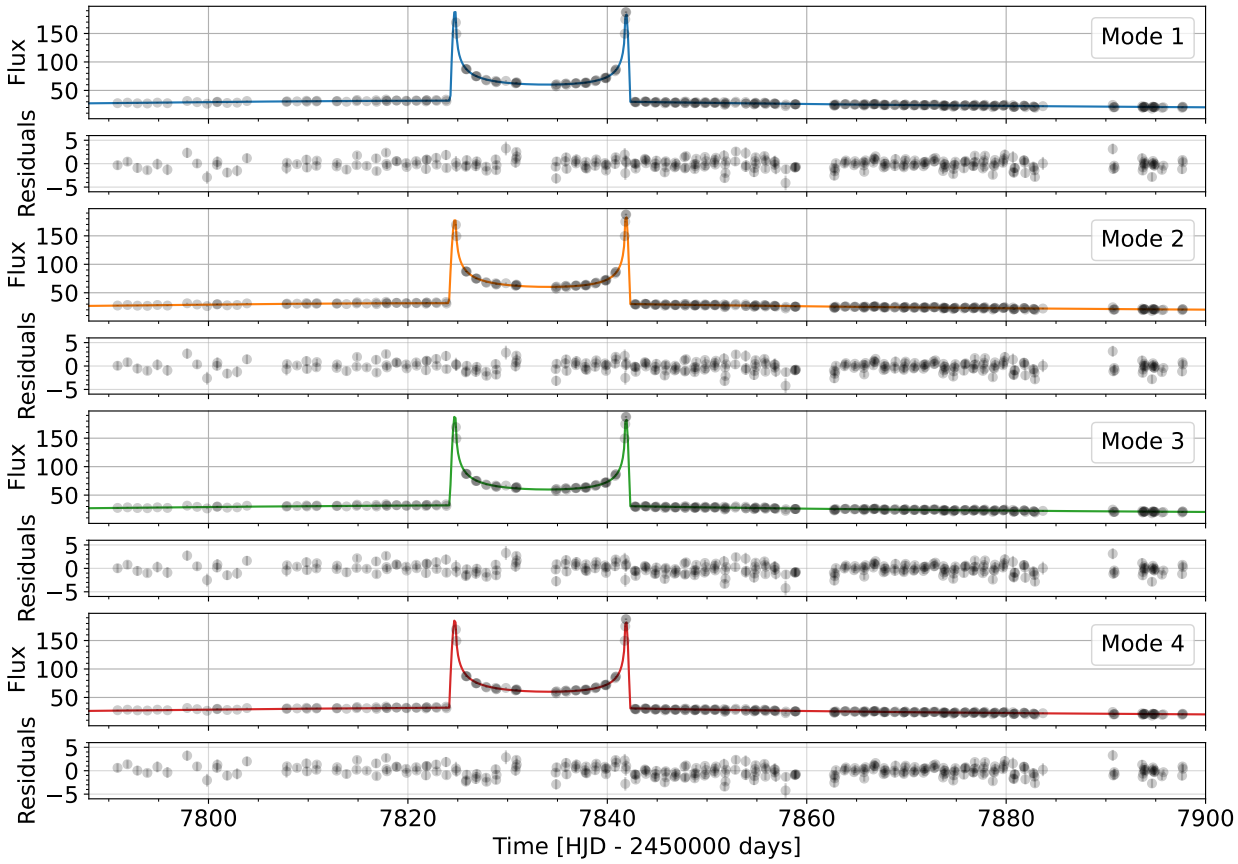


Figure 5.4: Predicted flux and residuals for each of the four modes are shown in Figure 5.3.

5.1.1 Current approaches to modelling multiple-lens events

The most common approach to fitting binary lens events in the initial modelling stage is to optimize the negative log-likelihood in search of the global minimum using either gradient-based optimizers such as BFGS or Levenberg-Marquardt, or derivative-free optimizers such as evolutionary methods, particularly differential evolution. These strategies are not guaranteed to find all relevant modes in the likelihood and they require a lot of problem-specific tuning.

I have tested a BFGS optimizer using the exact gradients of the likelihood computed by caustics and I found that, as one would expect from the structure of the likelihood shown in Figure 5.2, the optimizer inevitably gets stuck in a local minimum in every single case. The convergence within a mode is faster with exact gradients obtained through autodiff than with finite-difference gradients but this is not that helpful because the challenge is finding the modes in the first place (the search problem). A gradient-free optimizer can actually be more likely to find the relevant minima in these non-smooth problems because it effectively smooths out the local gradient. A good discussion on this and similar issues can be found in Metz et al. (2021). In particular, evolutionary methods, which work by initialising a population of points in the parameter space and then iteratively evolving the population towards a global optimum, are known to work well with non-smooth likelihoods (see (see Rajpaul, 2012, and references therein)). Differential evolution is one of the most popular evolutionary optimisation algorithms.

However, at the end of the day, we are not interested in just the point estimates because we also want to estimate the uncertainty in the model parameters for different modes in the likelihood. Even if we had a robust method for finding the global minimum, the inverse Hessian at the mode is a very poor approximation to the posterior covariance.[4] Instead of trying to solve the general *optimisation problem* and estimating the covariance using the inverse Hessian, most papers in microlensing instead use a combination of grid-search *in a subset of the full parameter space*, and MCMC. The purpose of this strategy is to find and explore the highest probability regions of the parameter space. The exact approach varies from paper to paper but it looks something like this:

1. **Grid search in the parameter subspace**. All model parameters except for (s, q, α) are set to some fixed values and the likelihood is then evaluated at every point on a uniform 3D grid in $(\ln s, \ln q, \alpha)$.

2. **MCMC exploration**. MCMC samplers are initialised at every grid point to sample the *conditional distribution* $p(\boldsymbol{\theta}'|q, s, \alpha)$, where $\boldsymbol{\theta}'$ is the parameter vector which excludes (q, s, α). The chains exploring the lowest likelihood regions are then discarded.

3. **Sampling stage**. Finally, the "best" chains are used to initialise a new MCMC run in the full parameter space.

This method is just a set of heuristics. The purpose is to initialise the final MCMC chains and it can be seen as a warmup stage of an MCMC sampler. There are many issues with this approach to finding the modes in the likelihood. I list a few below:

- The grid search strategy is not guaranteed to find all relevant modes in the likelihood. Even if the likelihood is well-behaved in the parameters that are fixed during the grid search, the modes of the conditional distribution are not necessarily the modes in the full distribution because there are covariances between all parameters. Since the likelihood is most certainly not well behaved in the parameters fixed in the grid search (see Figure 5.2), this procedure is not at all guaranteed to find all of the modes.

- The fact that each paper uses a slightly different variation of the strategy discussed above, and that every stage in the process also includes a lot of human input, means that the results for different events are not directly comparable and they should not be used in population studies.

- Even if the grid search strategy is successful in finding all significant modes in the full posterior[5], we still have to make sure that the MCMC chains initialised at the final points converge to the local mode. As we shall see in the following section, this is often not true.

[4] This is obvious by just looking at the structure of the likelihood in Figure 5.2, but it is also straightforward to compute the importance sampling weights for the multivariate Gaussian approximation to the posterior and show that the approximate is very far from the true posterior, as I have described in Chapter 4.

[5] This is an empirical claim that has not been thoroughly tested in the literature, at least to my knowledge.

5.2 Why using MCMC to fit caustic-crossing events is a bad idea

The final step in the modelling process outlined in the previous section involves initialising an MCMC sampler in a local posterior mode, and sampling for some number of steps. If we compare the geometry of the likelihood shown in Figure 5.2 to the geometry of the kinds of problems MCMC samplers tend to work well with, it does not seem like this would be an easy task. The sampler that is by far most popular in microlensing is the Python implementation of the affine-invariant ensemble sampler (AIES) (see Section 2.4.4) called emcee (Foreman-Mackey et al., 2013a). emcee is invariant to linear transformations of the parameter space and it generally works well if the dimensionality of the parameter space is not too high ($\gtrsim 10$). Amazingly, almost no papers in microlensing report diagnostics for the chains indicating that the sampler has converged! Even basic information such as the number of warmup steps and the number of sampling steps is often omitted.

To check if emcee is able to converge to a local mode, I initialise 32 emcee walkers in a ball of radius 0.01 around the highest likelihood mode from the ones shown in Figure 5.3. I use the default settings for the sampler and I run a large number of warmup and sampling steps, 10000 and 50000 respectively. Table 5.2 shows two important diagnostics for the chains, the effective sample size (ESS) and the Gelman-Rubin $\hat{R}$ statistic (see Section 2.4.4 for a discussion on MCMC diagnostics). These two diagnostics can be seen as necessary but not sufficient conditions for convergence. The requirement for convergence is that the ESS is in the hundreds (at least 400 or so) and that $\hat{R}$ is close to 1. Neither of these criteria appears to be met in this case. The ESS for some parameters is less than 100 indicating very high autocorrelation in the chains, and $\hat{R}$ is not even remotely close to 1 for any of the parameters.

Parameter	s	q	u_0	α	t_0 HJD - 2450000 days	t_E days	$\rho_\star$
ESS	43	177	44	208	39	165	118
$\hat{R}$	2.0	1.12	1.1	1.12	2.36	1.13	1.18

Table 5.2: MCMC diagnostics for the emcee chains initialised near the highest likelihood mode from Figure 5.3. The ESS stands for effective sample size.

The implication of these results is that MCMC chains have not converged[6]. That is hardly surprising given the highly problematic geometry of the likelihood. Of course, we cannot extrapolate from this single example to the entire literature, but the fact that most papers do not even report if the chains have converged or not is a strong indication that we should not be too confident of the published results. I have also tried to sample this problem with the NUTS sampler using caustics and I ran into similar problems. The very large curvature of the target density leads to very long and costly integration steps in the leapfrog integrator which caused the sampler to stall.

[6]This is also clear from looking at the posterior samples (not shown in this case).

In conclusion, it seems like doing MCMC in the context of caustic-crossing microlensing events is most likely wasteful of CPU time. In the next section, I explore an alternative approach.

5.3 The search problem and Nested Sampling

I have introduced Nested Sampling (NS) in Section 2.4.4 and I have also discussed its application to single lens microlensing events in Section 4. Nested Sampling works by exploring the parameter space from the outside in. It starts by distributing a set of "live points" in the parameter space and then iteratively removing the worst point (in terms of likelihood) at each iteration. The removed point is replaced by a *sample from the prior*, subject to the constraint that the likelihood of the new point is higher than the likelihood of the removed point (this step is called Likelihood Restricted Prior Sampling or LRPS). The process is repeated until the Bayesian evidence $\ln \mathcal{Z}$ is estimated to be within a certain tolerance. Posterior samples can be obtained by re-weighting the discarded live points.

The advantage of NS methods over MCMC methods is that the former are often able to discover multiple modes in the posterior and deal with more complex likelihoods. The disadvantage is that it is generally more computationally expensive than MCMC methods. It also scales poorly with the dimension of the parameter space and it is more difficult to diagnose problems with the inference process compared to MCMC methods.

The most important aspect of the NS algorithm is the number of live points (which can be fixed or adaptive, depending on the implementation) and the LRPS step. There are two classes of methods for sampling the likelihood constrained prior, for the purpose of generating a new live point:

- **Region samplers** construct a region that bounds the iso-likelihood contour defined by the removed point. They then rejection sample from that region until a sample satisfying the likelihood threshold is obtained. The regions are usually constructed by wrapping the active live points with one or more overlapping ellipsoids.

- **Step samplers** evolve a randomly chosen live point in a random walk until an approximately independent sample is obtained. This is effectively MCMC sampling with a hard prior constraint. Commonly used step samplers are the slice sampler and the Metropolis sampler. HMC-like gradient-based samplers that "reflect" off of the likelihood constraint can also be used.

To test the performance of NS on the binary lens problem, I first tested the region sampler variant from UltraNest. This is the version of NS I used in Chapter 4 to fit a single lens microlensing event. I used broad uniform priors for all parameters, as shown in Table 5.3. The estimate of the evidence $\ln \mathcal{Z}$ is sensitive to priors, however, since in this case we are not interested in the evidence the priors are not really important.

I found that the region based Nested Sampling strategy completely failed to converge for this problem. After more than 200 million (!) likelihood evaluations, $\ln \mathcal{Z}$ estimate was nowhere near convergence and the efficiency of the sampler declined to less than 0.01%. This extremely low acceptance rate suggests that the volume of the bounded region constructed

Parameter	Prior
$\ln s$	$\mathcal{U}(\ln 10^{-1}, \ln 3)$
$\ln q$	$\mathcal{U}(\ln 10^{-2}, \ln 1)$
$\ln u_0$	$\mathcal{U}(\ln 10^{-3}, \ln 1)$
α	$\mathcal{U}(-\pi, \pi)$
$\ln t_0$	$\mathcal{U}(\ln 7820, \ln 7870)$
$\ln t_E$	$\mathcal{U}(\ln 10, \ln 365)$
$\ln \rho_\star$	$\mathcal{U}(\ln(5 \times 10^{-4}), \ln 10^{-2})$

Table 5.3: Priors for the parameters used in the Nested Sampling analysis.

by wrapping active points in overlapping ellipsoids is too large compared to the volume of the parameter space that satisfies the likelihood constraint. Thus, it does not seem that the region sampler variant of NS is suitable for this problem, at least not without major modifications to the ellipsoid wrapping algorithm.

Since the region sampling strategy failed, I decided to try using a (slice sampling based) step sampler. I set the number of slice sampling steps to 50 and vary the number of live points. The purpose of the experiments is to answer two questions:

1. Does the algorithm converge, in the sense that increasing the number of live points does not meaningfully change the posterior distribution?

2. Does the NS algorithm discover all important modes in the posterior?

I run UltraNest with three different values of the number of live points: 2000, 5000 and 10000 (default setting is 400). The stopping criterion is set to 0.1. Each run is executed in parallel on a CPU cluster with 4 nodes with 32 cores each using MPI. Table 5.4 lists the estimated $\ln \mathcal{Z}$ values for the three runs and the total number of likelihood evaluations in each case. The most expensive run with $n_{\text{live}} = 10000$ took more than 24h to complete on the cluster and it involved more than 100 million likelihood evaluations.

Nr. of live points	$\ln \mathcal{Z}$	Total nr. of likelihood evaluations
2000	-2031 ± 0	42.2M
5000	-2031 ± 0	70.9M
10000	-2032 ± 0	137.7M

Table 5.4: Results of the UltraNest runs for the binary lens model.

The posterior samples for each of the three runs, obtained by re-weighting the discarded live points (see Equation 2.233), are shown in Figure 5.5 for a selection of parameter pairs. Looking at the 2D marginal distributions, for $n_{\text{live}} = 2000$ there appears to be one dominant mode in the posterior (whose topology is similar to Mode 1 shown in Figure 5.3). If we increase the number of live points to 5000 another significant mode with wider separation between the lenses appears. However, for $n_{\text{live}} = 10000$ this mode disappears again and a third, previously unseen mode appears. Thus, it appears that NS with a slice step sampler

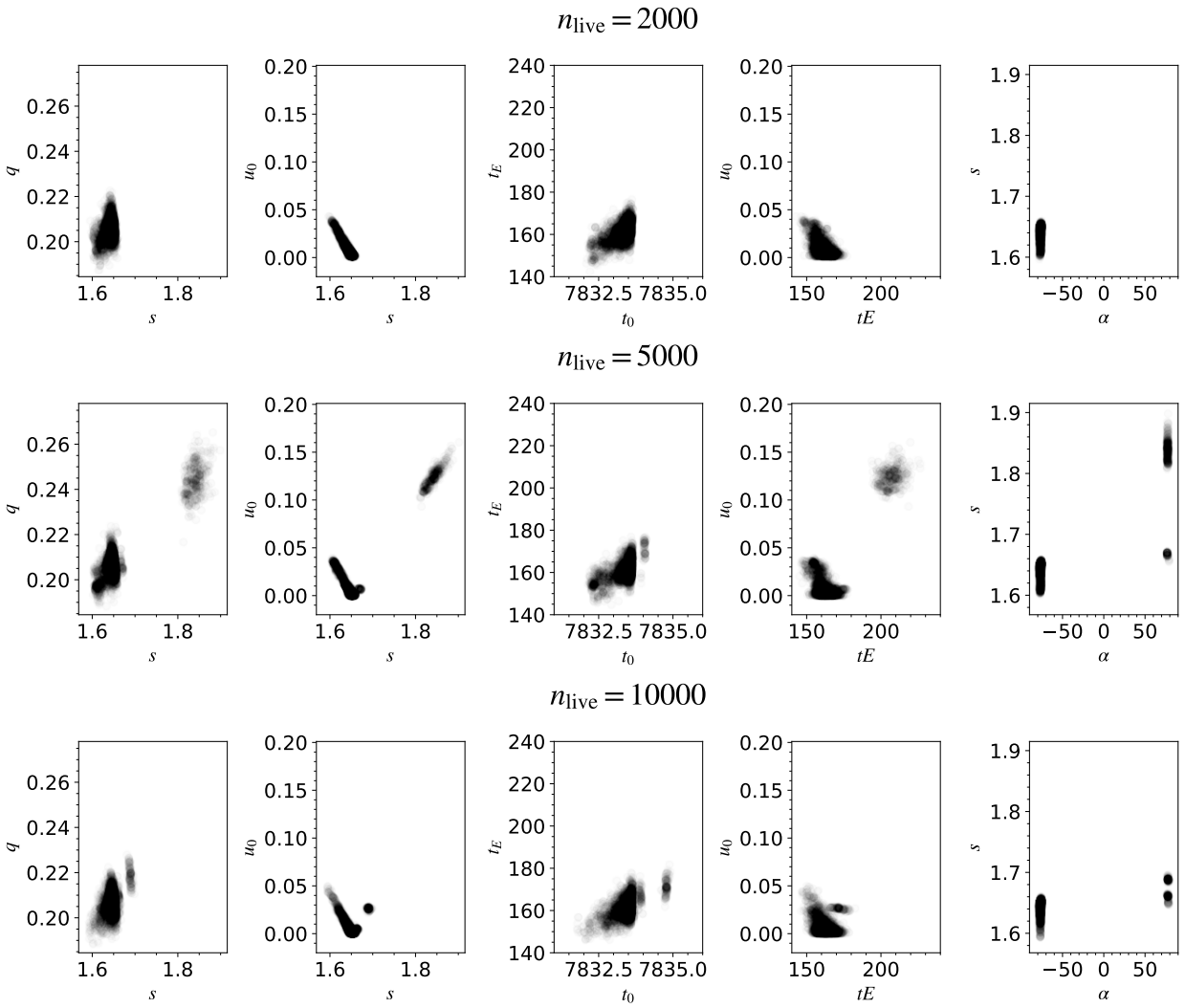


Figure 5.5: Posterior samples from the three UltraNest runs listed in Table 5.4. In each case the samples are visibly different, indicating that Nested Sampling is most likely not converging to the true posterior.

is not converging to a stable distribution, even for a very large number of live points.[7] Regarding the second question – does the algorithm discover all important modes in the posterior – the answer is not clear. It does appear that NS discovers the main mode around ($s = 1.65$, $q = 0.21$) in all three cases and the value of the likelihood at this mode is larger than for Modes 2, 3, 4 shown in Figure 5.3. However, without accurately estimating the cross-validation scores for each mode, we cannot simply discard the lower likelihood modes. In fact, discarding Mode 2 from Figure 5.3 would be a mistake because the solution Han et al. (2018) found for this event (using additional data and a model including parallax) is like Mode 2 rather than Mode 1.

Even if we found a setting of the hyperparameters for which the sampler converged to a stable posterior distribution, this would not tell us if it is converging to the true posterior. I propose the following experiment to test the convergence of the sampler to the true posterior.

[7] We are not testing the convergence to the true posterior because we do not know the true posterior.

We could fix all but three parameters in the model, for example, (s, q, α), then use UltraNest to sample the conditional distribution (see Section 2.4.4) $p(s, q, \alpha|\boldsymbol{\theta}')$ and compare it with samples from a simple rejection sampler. Rejection sampling is useless in 7 dimensions because of the curse of dimensionality, but it can work well in parameter spaces with up to 5 dimensions. We could then repeat this experiment for different subsets of parameters. If the rejection sampling results were to be consistent with NS results, we would have some evidence that Nested Sampling is converging to the true posterior.[8]

5.4 Summary and future work

In this brief Chapter, I explored modelling strategies for binary lens microlensing events (though the conclusions are even more important for triple-lens events) using the OGLE EWS light curve of the event OGLE-2016-BLG-0039 as a test dataset. I showed that current approaches to modelling binary events have fundamental flaws. The crux of the problem is that the likelihood function for caustic-crossing microlensing events is extremely non-smooth and multi-modal. As a result, it is very difficult, if not impossible, to construct a search algorithm that reliably finds all significant modes in the likelihood or the posterior distribution. Popular strategies for solving the search problem in microlensing are entirely based on heuristics and are prone to failure.

In addition to the search problem, exploring the posterior distribution in the neighbourhood of a particular "solution" is also very difficult. The most common approach uses gradient-free MCMC sampling (specifically, emcee), initialised at a local mode. I showed that this approach is likely flawed because emcee completely fails to converge even with very long chains. All of these problems are likely to be even worse for more complex binary lens models with higher-order effects such as parallax and orbital motion, especially for triple lens events, although I did not test this hypothesis.

To explore a potential solution to the search and sampling problems, I tested Nested Sampling, which is conceptually very different from MCMC methods. I used the UltraNest implementation of Nested Sampling which is more robust than alternatives because it is tuned for accuracy over speed. I found that the rejection sampling approach to sampling the likelihood constrained prior does not work for this problem because the efficiency of the sampler tends to zero well before the bulk of the posterior mass is accounted for. It appears that the alternative variant of Nested Sampling, using the slice step sampler, does not work either. The sampler does not appear to converge to a stable distribution even after more than 100 million likelihood evaluations (which requires thousands of CPU hours of computing time). Although these preliminary results look quite discouraging, it would be worthwhile to explore Nested Sampling further.

In summary, given all of the above, I believe that past results on binary and triple-lens microlensing events (especially caustic-crossing events) should be taken with a grain of salt Although existing methods often find one or more "fit the data well", the reported uncertainties for model parameters should probably not be trusted. It is also not at all clear that other, potentially important solutions have not been missed. Solving these problems

[8] Unless Nested Sampling stops sampling the true posterior faithfully in a higher dimensional space, which is also entirely possible.

will require major changes in common modelling practices to happen and likely also a lot more computing power than is commonly used today. It does not help that the microlensing community is very closed-off, most data are not publicly available, the analysis code software is rarely open-source, and the validity of statistical inferences is often not tested even using the most basic tools from modern computational statistics.

Chapter 6

Mapping the surface of Io

In this chapter, I apply the methods discussed in 2.2 to a planetary science problem. The chapter describes a new method for inferring the positions and brightness of the volcanoes on Jupiter's moon Io. The key idea behind this is to apply the starry model to the light curves of Io's occultations by Jupiter and infer a spherical harmonic map. This work was published in Bartolić et al. (2022).

6.1 Introduction

The surface of Io is covered with hundreds of volcanoes which appear as bright spots in the near-infrared, their intensities vary on timescales ranging from days to decades. The global heat flow on its surface is about 40 times larger than Earth's (Breuer and Moore, 2007; Davies and Davies, 2010) and about 50% of the heat flow emanates from only 1.2% of Io's surface (Veeder et al., 2012). This intense volcanic activity cannot be explained by radioactive decay and residual heat from formation, the mechanisms which drive volcanism on Earth. Instead, the volcanism on Io is driven by tidal interactions with Jupiter and sustained by the Laplace resonance with Europa and Ganymede (Peale et al., 1979). Contrary to Earth, Io lacks plate tectonics. The heat transport mechanism most likely operating on Io is *heat-pipe volcanism*, a process in which magma is transported to the surface through localised vents from the lithosphere (the outermost shell of a terrestrial body) (O'Reilly and Davies, 1981). Heat-pipe volcanism likely occurred in the early history of terrestrial planets in the Solar System (O'Reilly and Davies, 1981; Breuer and Moore, 2007), most notably on early Earth prior to the onset of plate tectonics.

Besides providing a window into early volcanic activity in the Solar System, Io is also in many ways an analogue of a volcanically active exoplanet. Volcanic exoplanets, sometimes dubbed "super-Ios" or "lava worlds", have gathered a lot of interest in recent years. Photometric and spectral signatures of volcanic activity on such worlds will likely be detectable in the near future with telescopes such as JWST and LUVOIR (Kaltenegger et al., 2010; Henning et al., 2018; Oza et al., 2019; Chao et al., 2021). Existing exoplanet detections of planets with potential volcanic activity include CoRoT-7b (Barnes et al., 2010), the first rocky exoplanet discovered and one which is likely heated by strong tidal forces; 55 Cancri e, whose inferred longitudinal offset in peak surface emission has been attributed to (among

other things) lava flows on the surface (Demory et al., 2016b,a; Hammond and Pierrehumbert, 2017); several planets in the TRAPPIST-1 system which are in a Laplace-like resonance and are likely exhibiting volcanic activity (Kislyakova et al., 2017; Dobos et al., 2019), and many others. Not all such exoplanets are expected to have Io-like volcanism. Some will have magma oceans because of their proximity to the star and others will have volcanism powered by nuclear decay and residual heat, similar to volcanism on present-day Earth.

Io has been extensively observed using both space and ground observatories. High-resolution images of Io's surface were taken by space missions such as Voyager (Smith et al., 1979), Galileo (Belton et al., 1996) and Juno (Mura et al., 2020). The surface has also been resolved from ground-based observations in the near infrared using disk resolved imaging (Howell and McGinn, 1985; Simonelli and Veverka, 1986; Spencer et al., 1990) and adaptive optics observations (Marchis et al., 2000, 2005; de Kleer and de Pater, 2016a). Most importantly for this work, starting with Spencer et al. (1990) Io has been sporadically observed over a time span of decades using high-cadence, near-infrared photometric observations which were taken during occultations by Jupiter. Occultations by Jupiter occur once every orbit of $\sim$ 1.7 days whereas occultations by Europa, Ganymede or Callisto (so-called "mutual occultations") happen every $\sim$ 6 years when Earth passes through the orbital plane of the Galilean satellites. Multiple occultations are then observable over a course of approximately one year. The majority of occultations by Jupiter are observed when Io is in Jupiter's shadow ("in eclipse") while mutual occultations are almost always observed in sunlight. Only the brightest volcanoes are visible over the reflected light background when Io is illuminated by the Sun (Veeder et al., 1994; de Kleer and de Pater, 2016b).

Both kinds of occultations have been used to study volcanic activity on the surface. Spencer et al. (1994) observed several occultations of Io by Europa and detected a major brightening of the most powerful of Io's volcanoes, Loki, relative to previous Voyager observations. Rathbun et al. (2002a), Rathbun and Spencer (2006) and Rathbun and Spencer (2010) used observations from NASA's Infrared Telescope Facility (IRTF) telescope to study the long-term variability of different volcanic spots, finding evidence of periodicity in Loki's eruptions and establishing the transient nature of observed emissions from most volcanoes. By studying a (spatially resolved) occultation of Io by Europa, de Kleer et al. (2017) mapped the Loki Patera (*Patera* is a type of irregular crater) region to a precision of about 2 kilometres. In addition to observing occultations in the near-infrared, several groups have been observing mutual occultations in the optical for decades with the purpose of inferring the optical albedo of Io and improving ephemeris precision for Galilean satellites (Arlot et al., 1974; Lainey et al., 2009; Saquet et al., 2018; Morgado et al., 2016, and references therein). Understanding the detailed albedo distribution of Io's surface is crucial to constraining the ephemeris of Io to very high precision.

Most studies of (unresolved) occultations of Io fit multiple light curves independently with the goal of inferring one-dimensional longitudinal variations in brightness on its surface (notable exceptions are Spencer et al. (1994) and de Kleer et al. (2017)), assuming relatively strong priors on the locations of individual volcanoes. In this work, we build a fully probabilistic model in order to infer a *two-dimensional map* of Io's surface thermal emission using archival IRTF observations of Io in the near-infrared. We assume that the map we fit to a given set of light curves is static (up to an overall amplitude), an assumption which is justified if the spatial structure and intensity of hotspots do not change substantially between

the times of observations. We leave the time-dependent case for future work.

The model relies on the code starry (see Section 2.2) which enables fast analytic computation of occultation light curves and phase curves for objects whose surface features can represented in terms of spherical harmonics. starry can compute phase curves and occultations in both emitted light (for modelling the isotropic thermal emission from Io's surface) and reflected light (for mapping albedo variations). It is many orders of magnitude faster and more accurate than pixel-based algorithms and it computes exact gradients of the flux with respect to all parameters through automatic differentiation. starry also comes with extensive tools for visualising spherical harmonic maps and simulating data.

Our main goal in this chapter is to develop a general model for mapping surfaces of occulted bodies with sparse, high-contrast features and apply it to observations of Io. Because of the existence of high-resolution resolved imaging data of Io, we have some knowledge of the ground truth so we are able to validate our model. The rest of the chapter is organized as follows. In Section 6.2 we describe the IRTF light curves of occultations by Jupiter which we use to infer maps. In Section 6.3 we discuss the generative model for the data and we write down the likelihood function. In Section 6.4 we focus on the question of how to do Bayesian inference given the forward model specifications defined in Section 6.3. We discuss and test the use of different priors on map features and quantify the information content of the occultation light curves. We then fit the model with realistic simulated data using Hamiltonian Monte Carlo (HMC). In Section 6.6 we show the results for IRTF data for two separate pairs of light curves, one pair observed in 1998 and another in 2017. We show the inferred maps and the parameters quantifying the location and intensity of the hot spots, we discuss the sensitivity of the results to different choices of priors, and plot the inferred hotspots overlaid on top of a high-resolution optical map of Io which was constructed from Galileo observations. In Section 6.7 we summarise the main results and discuss potential applications of our model to observations of volcanic exoplanets. Finally, in Section 6.8 we briefly discuss a possible extension of the model for fitting time-dependent maps of Io.

6.2 Data

The photometric observations of Io we use in this work were first described in Spencer et al. (1994) who began the observing campaign in 1989 and were joined by others over the following years (Stansberry et al., 1997; Rathbun et al., 2002a; Rathbun and Spencer, 2006, 2010). These data are currently being organized to be posted on the Planetary Data System (PDS) (Julie Rathbun, personal communication). We used an early version of this data archive. The final version of the archive will contain on the order of 100 separate light curves of occultations of Io by Jupiter, observed at times when Io was in Jupiter's shadow ("in eclipse"). The observations for each year consist of a sequence of only ingress (egress) light curves followed by only egress (ingress) light curves. This is because the geometry of the occultations of Io by Jupiter is such that depending on the time of the year, Io is in eclipse during either the ingress or the egress of the occultation.

Since we have restricted the scope of this work to fitting static maps of Io's thermal emission, we selected light curves which are closely spaced in time so that Io's surface emission doesn't drastically change between the observations. In addition, we selected observations

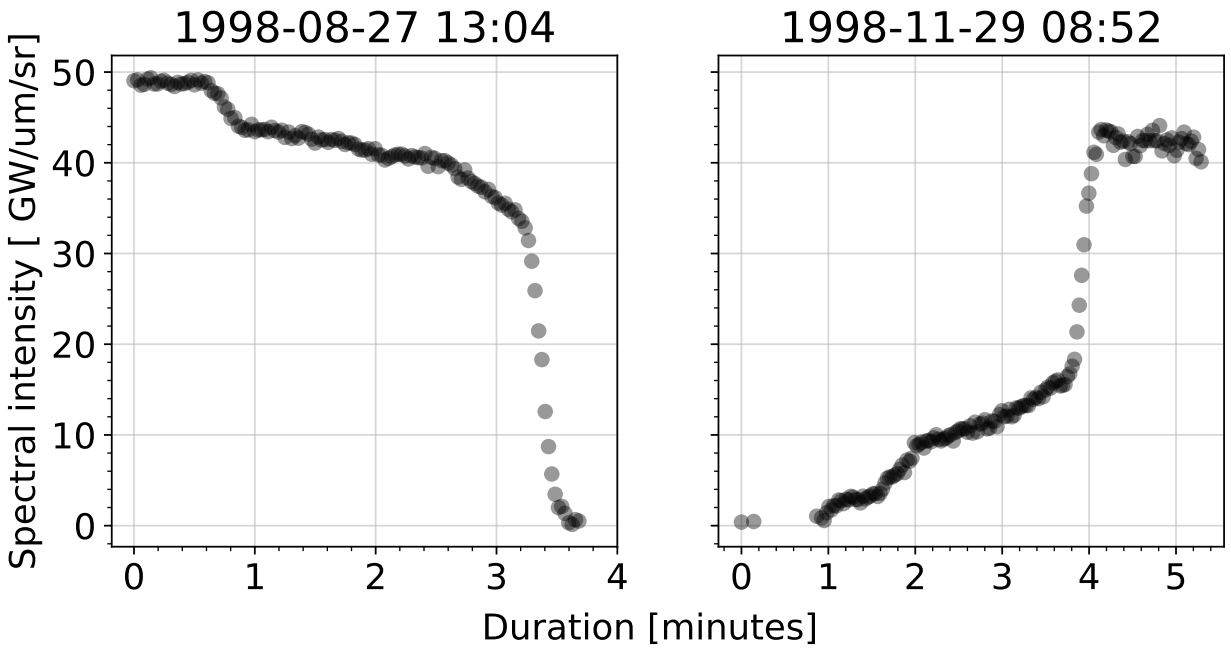


Figure 6.1: Occultations of Io by Jupiter observed 3 months apart in 1998 using NASA's IRTF telescope. The light curve on the left is the ingress of the occultation (Io disappearing behind the limb of Jupiter) and the one on the right is the egress (Io appearing behind Jupiter). Each visible step-like feature in the light curve (a rapid increase/decrease in total flux at particular times) is a consequence of a volcanic hotspot coming in or out of view during the course of the occultation. The two light curves contain a wealth of information about thermal emission from Io's surface.

of both the ingress and the egress of an occultation because Jupiter's limb sweeps across Io's projected disc at different angles at ingress and egress and this means with both kinds of light curves we can break the degeneracy in the position of the hotspots. The final set of light curves consists of a single pair of ingress/egress observations from 1998 observed 94 days apart and another pair of ingress/egress observations from 2017 observed 54 days apart. We fit one map for each of the two pairs of light curves.

The 1998 occultations have been observed with the NSFCam 1-5 μm camera (Shure et al., 1994) while the 2017 occultations were observed with the SpeX 0.8-5.5 μm spectrographer/imager (Rayner et al., 2003). The pair of observations from 1998 is shown in Figure 6.1, the observations from 2017 are qualitatively similar. Each light curve covers a timespan of ~ 4 minutes which is the duration of the ingress (egress) part of the complete occultation. The observing cadence for each light curve is about a second which sets a lower limit for the size of the inferred spatial features ($\sim$ 15km). Step-like features visible in Figure 6.1 correspond to volcanic hotspots coming in or out of view during the course of the occultation so the light curves clearly encode information about emission features on the surface of Io.

Because the early version of the data archive which is eventually going to be uploaded on the PDS does not contain estimates of errorbars for the light curves, we work around this limitation by estimating the errorbars from the data (see Section 6.6.1 for more details). As with all ground-based photometry, the observations are influenced by atmospheric variability which results in the presence of some correlated noise which needs to be accounted for in the model.

6.3 Model

In this section, we describe the generative model for the data presented in Section 6.2. To simulate the observed light curves, we first need to specify the geometry of each occultation event (Section 6.3.1). We then need to specify what the surface emission of Io looks like in a spherical harmonic basis and compute the theoretical flux at the times of observations using starry (Section 6.3.3). Finally, we need to specify a noise model for the observed data (Section 6.3.4).

6.3.1 Orbital parameters

To compute the geometry of every occultation, we use the JPL Horizons database which uses the latest ephemeris for the position of Jupiter (Folkner et al., 2014), and its satellites (Jacobson and Brozovic, 2015). The ephemeris is usually accurate to a few kilometres on Io's surface. We access JPL Horizons through the Python package astroquery (Ginsburg et al., 2019). To compute the relative position of Jupiter and Io we need the right ascension and declination and to fix the orientation of Io we need the longitude and the latitude at the centre of Io's disc as seen from Earth (Ob-lon and Ob-lat in astroquery) and the counterclockwise angle between the celestial north pole unit vector projected onto the plane of the sky and Io's north pole (NP.ang). All longitudes provided by Horizons are positive in the direction of the west. Horizons provides all ephemeris with a minimum cadence of 1 minute, we interpolate these values so that we can evaluate them at arbitrary times.

The coordinate system in starry is defined to be right-handed such that the $\hat{z}$ axis points towards the observer and the $\hat{x}$ axis points to the East. The radius of the occulted sphere is fixed at 1 and the orientation is specified by three angles: the counterclockwise obliquity angle obl between the $\hat{y}$ axis and the north pole of the sphere, the inclination angle inc which is set to 90° if the north pole is aligned with the $\hat{y}$ axis, and the phase angle theta that determines the rotation of the sphere around the $\hat{y}$ axis in the eastward direction. Expressed in terms of Horizons variables, these angles are given by obl = NP.ang, inc = 90° − Ob-lat and theta = Ob-lon. The occultor position relative to the occulted object is given by

$$x_o/\gamma = -\Delta\alpha \cos\delta \tag{6.1}$$
$$y_o/\gamma = \Delta\delta \tag{6.2}$$
$$z_o/\gamma = 1 \quad , \tag{6.3}$$

where $\Delta\alpha$ and $\Delta\delta$ are differences in right ascension and declination respectively relative to the occulted object and γ is the angular radius of the occulted sphere.

6.3.2 Jupiter's effective radius

Although the sky position of Jupiter relative to Io is known to a precision of a few kilometres, several complications arise when attempting to compute Jupiter's radius. First, Jupiter is not spherical and its equatorial radius is greater than the polar radius by thousands of kilometres. Given the relatively crude resolution of our spherical harmonic maps and the small spatial scale of Io's projected disc compared to the scale of Jupiter, we can safely assume that Jupiter

is *locally spherical* at the point of an occultation. We estimate Jupiter's effective radius from the measurements of Jupiter's shape done using Pioneer and Voyager radio occultation data (Fig. 7 of Lindal et al., 1981). When computing the occultation latitude we fix Jupiter's latitude to the value at the centre of Io's disc because of the variation of Jupiter's effective radius along Io's disc is negligible.

Second, Jupiter is gaseous so it does not have a well-defined boundary. In principle, we should compute an effective radius of Jupiter at different altitudes (pressure) in the atmosphere and model an occultation of Io by a fuzzy occultor. Although this is possible with `starry`, it is unnecessary for our models because the characteristic scale height of Jupiter is around 27 km which is below the uncertainty of our inferred maps. We instead follow the approach from Spencer et al. (1990) and compute the effective radius of Jupiter at about 2.2 mbar, the pressure (and the associated effective radius) at which a bright source on the surface of Io fades by 50% due to differential refraction during the course of an occultation.

Third, the information on the effective radius in Lindal et al. (1981) is provided only at a fixed pressure of 100 mbar. To adjust the values for a lower pressure of 2.2 mbar we assume an exponential pressure profile $P = P_0\, e^{-\Delta r/H}$ where H is the scale height and Δr is the height difference between the two pressure levels. It follows that to convert the shape profile at 100 mbar to 2.2 mbar we need to add the factor $-H \ln(2.2/100)$ which is assumed to be constant in the ± 21 deg latitude range in which the occultations occur. In addition to refractive absorption in Jupiter's atmosphere, there is a also slight additional molecular absorption due to methane which Spencer et al. (1990) estimate to be equal to around 12% in their filter, we choose to ignore this because it is far below the resolution of our maps.

Finally, we have to account for the fact that the light from Io is getting significantly bent at the point of half reflective intensity in Jupiter's atmosphere which results in a smaller projected limb of Jupiter on the surface of Io than would be the case for straight propagation. It is zero at the beginning of the disappearance of a hot spot when there is no refraction, increasing to one scale height H at the half-intensity point and then increasing further to a large value when Io disappears behind Jupiter's limb. We ignore the variation in the bending and adopt a fixed value of one scale height for this effect which we subtract from the value of the effective radius. There are substantial uncertainties in each of these steps. The shape profile data are quite old and it is not known if the structure of Jupiter has remained constant since the 1980s. The shape profile also depends on the wind velocity structure and temperature which we do not take into account. In addition to uncertainties on the atmospheric structure, there are uncertainties associated with digitising the data shown in Fig. 7 in Lindal et al. (1981) because it is not available in table form.

Problems with Galileo's high gain antenna prevented it from obtaining more recent radio occultations of the neutral atmosphere which could be used to improve the shape measurements and Juno's orbit has thus far avoided such occultations. However, over 2023–2025 period an extended Juno mission may be able to obtain radio occultations which would refine our understanding of Jupiter's size and shape (Hodges et al., 2020). Since our main focus in this work is the map model, we leave a detailed investigation of the various sources of error that go into the radius estimate for future work.

6.3.3 Linear model for the flux

Given the geometry of on occultation event at the time of observation, computing the predicted flux with starry is straightforward. To model the emitted light flux from Io during an occultation we represent the emitted light intensity at any point on Io's surface using a vector of spherical harmonic coefficients and compute the flux during an occultation using starry (see Section 2.2). Conditioned on fixed parameters specifying the geometry of the occultations, the starry model is *linear* in $\mathbf{y}$ (Equation 2.108) and the predicted flux can be written as

$$\mathbf{f} = \mathbf{A}\,\mathbf{y} \quad , \tag{6.4}$$

where the column vector $\mathbf{f}$ of shape $(T, 1)$ is the predicted flux for different values of the occultor position and $\mathbf{A}$ is the design matrix (see appendix B.1. in Luger et al., 2021c) with shape (T, N) (where $N = (l+1)^2$) which encodes all operations needed for computing the integrated flux at different viewing angles of the occulted sphere: relative positions and radii of the occultor and the occulted object, rotations, basis transformations, and integrals over the visible disc of the occulted object, and the cosine emission law. If the geometry isn't known precisely $\mathbf{A}$ is not fixed and the model is in principle is no longer linear, although one can still use the fact that it is linear when conditioned on a particular value of the non-linear parameters to speed up inference.

The characteristic angular size of features that can be represented with a given map is set by the degree of the map and it is approximately equal to $180°/l$. For reference, state-of-the-art inferences involving phase curves and secondary eclipses of exoplanets are able to constrain features of order $l = 1$ (inferring a bright spot offset from the substellar point) but for Io, we need to fit much higher-order maps because the typical scale of volcanic spots is on the order of tens of kilometres (a few degrees). starry can handle occultations up to $l \approx 20$ before numerical instabilities kick in (Luger et al., 2019) which corresponds to a minimum resolution of 9° which means that we are not able to constrain the physical size of the spot. We discuss the implication of this resolution limit in Section 6.6 and Section 7.5.

Although starry was built around the idea of expanding surface features in a spherical harmonic basis in which we can compute all fluxes analytically, this basis may not be ideal for doing inference because it can be difficult to encode assumptions (priors) on what we expect the map to look like. For example, the most important constraint on the map we would like to incorporate in the model is that the intensity (power per unit area) of the map at any given point is positive. This constraint is important not only because we want to avoid having unphysical regions in inferred maps but also because it imposes a very strong prior on the map which substantially reduces the difficulty of inference when the data are not particularly informative.

There are two ways of enforcing positivity that we are aware of, both of which involve evaluating the map intensity on a fixed grid of *pixels* (intensity evaluated at a fixed set of points on the sphere) and not allowing negative values of those pixels because they are unphysical. The first is to fit for the spherical harmonic coefficients $\mathbf{y}$, evaluate the pixel grid at each MCMC step and reject all samples of the coefficient vector $\mathbf{y}$ which result in pixels with negative intensity. The issue with this procedure is that rejecting samples in this way implies a prior probability distribution on $\mathbf{y}$ which cannot be mapped to a parameter space with infinite support and Hamiltonian Monte Carlo does not work well

with constrained parameter spaces. The second approach is to dispense with the spherical harmonics entirely and compute the full model using a pixelated intensity map. This is very difficult to do in practice because we would need a very high-resolution grid to compute the light curves accurately with minimal discretisation noise. starry uses spherical harmonics as a basis precisely to avoid this problem. Instead, we opt for a hybrid approach in which we fit for a vector of pixel intensities but at each MCMC step we convert the pixels to spherical harmonic coefficients to compute the light curve.

To implement the hybrid approach we need to be able to switch back and forth between spherical harmonics and pixels. Given a vector of pixel intensities $\mathbf{p}$ evaluated on an equal area Mollweide grid, the transformation from spherical harmonics $\mathbf{y}$ to $\mathbf{p}$ is given by a linear operator $\mathbf{P}$:

$$\mathbf{p} = \mathbf{P}\,\mathbf{y} \quad . \tag{6.5}$$

Each row of $\mathbf{P}$ contains values of each of the spherical harmonic coefficients at a given point on the grid. We choose to use the Mollweide grid so that each area element on the sphere has approximately the same number of pixels. The grid needs to be fine enough to ensure that the intensity is positive over most of the sphere.

Switching from pixels to spherical harmonics is somewhat more complicated because $\mathbf{P}$ is not in general a square matrix so we cannot compute its inverse to obtain the inverse transform. Instead, we can compute an approximate inverse (a pseudoinverse) $\mathbf{P}^\dagger$ by solving the linear system $\mathbf{P}\,\mathbf{P}^\dagger = \mathbf{I}$, where $\mathbf{I}$ is the identity matrix. The solution is given by

$$\mathbf{P}^\dagger = \left(\mathbf{P}^\top\mathbf{P} + \lambda\mathbf{I}\right)^{-1}\mathbf{P}^\top \quad , \tag{6.6}$$

where λ is a small regularisation parameter and $\mathbf{I}$ is the identity matrix. The mapping from pixels to spherical harmonics is then given by

$$\mathbf{y} \simeq \mathbf{P}^\dagger\mathbf{p} \quad . \tag{6.7}$$

Both $\mathbf{P}$ and $\mathbf{P}^\dagger$ can be precomputed to speed up inference. When using pixels to impose a positivity constraint on the spherical harmonic map we need to make sure that the number of pixels is greater than the number of spherical harmonic coefficients by a factor of a few to ensure positivity approximately everywhere on the sphere. In practice, we find that we need to use at least 4 times as many pixels as spherical harmonics which means that the computational cost of this model is larger than that of a pure spherical harmonic model.

To summarise, in our hybrid model we first construct a fixed high-resolution pixel grid in latitude and longitude, then use Equation (6.7) to convert a vector of pixel intensities to spherical harmonic coefficients and finally use Equation (6.4) to evaluate the model. Although we fit for the pixels $\mathbf{p}$ we store the spherical harmonic coefficient vectors $\mathbf{y}$ as the final product of the inference. Figure 6.2 illustrates the transformation from the pixel basis to the spherical harmonic basis via $\mathbf{P}^\dagger$. On the left, we show the pixel map where each pixel was independently drawn from an exponential prior. On the right is the same pixel map transformed to a spherical harmonic basis via $\mathbf{P}^\dagger$. The histograms underneath each map show the distribution of intensities. Since the pixel map is higher resolution than the spherical harmonic map it can only be approximately represented at a finite order of the spherical harmonic expansion so the map on the right appears to be smoother and the

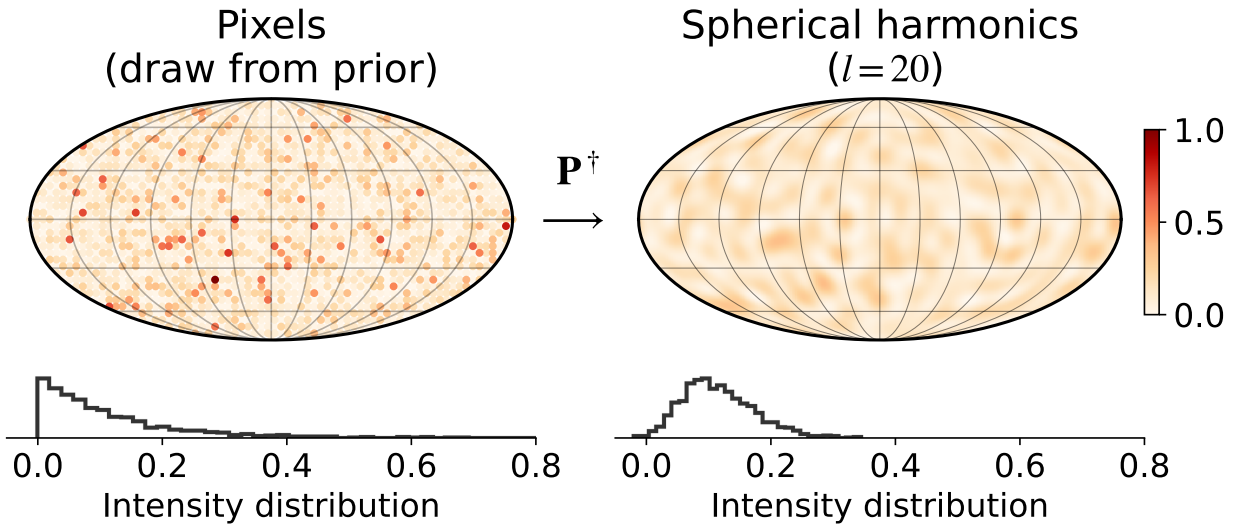


Figure 6.2: Illustration of the transformation from a pixel map (left) defined on a fixed grid on the sphere into the spherical harmonic basis at $l = 20$ (right) via a linear operator $\mathbf{P}^\dagger$. The intensity of each pixel was drawn independently from an exponential prior. The histograms below each of the maps show the distributions of intensities on the map. Since the pixel map is higher resolution than the spherical harmonic map at $l = 20$, the spherical harmonic map appears smoother and the intensity distribution is more similar to a skewed Gaussian rather than an exponential distribution. Despite this fact, we find that fitting for maps in the pixel basis and transforming them to spherical harmonics at each MCMC step in order to compute the light curve analytically with starry is better than fitting for spherical harmonics because the pixel grid provides a strong constraint on the spherical harmonic map structure.

intensity distribution is more similar to a skewed Gaussian with a heavy right tail than an exponential distribution. Nevertheless, we find that setting priors on pixels is a far better solution than fitting the spherical harmonic coefficients directly and is worth the extra computational cost which comes with the increased dimensionality of the parameter space (see Section 6.4.2 for a demonstration).

6.3.4 The likelihood

Finally, we have to specify the noise model which means we have to define a likelihood function. Assuming we have a single light curve with T data points and a map defined by the pixels $\mathbf{p}$, the (Gaussian) log-likelihood is given by

$$\ln \mathcal{L} = -\frac{1}{2} \left(\mathbf{f}_{\text{obs}} - \mathbf{f}\right)^\top \mathbf{\Sigma}^{-1} \left(\mathbf{f}_{\text{obs}} - \mathbf{f}\right) \quad , \tag{6.8}$$

where $\mathbf{f}_{\text{obs}}$ is the observed light curve, $\mathbf{\Sigma}$ is the data covariance matrix and $\mathbf{f}$ is predicted flux given by

$$\mathbf{f} = \mathbf{A}' \, \mathbf{p} + b \, \mathbf{1}_T \quad , \tag{6.9}$$

where $\mathbf{A}' \equiv \mathbf{A}\mathbf{P}^\dagger$ and b is a constant flux offset parameter which we have added to account for stray flux which cannot be attributed to Io. Depending on the observation, this flux is most often residual light from Jupiter.

To model the data covariance $\mathbf{\Sigma}$ we use a Gaussian Process and we compute the likelihood using the fast Celerite method (Foreman-Mackey et al., 2017) as implemented in the celerite2 package (Foreman-Mackey et al., 2017; Foreman-Mackey, 2018). We use the simple

(approximate) Matérn 3/2 kernel function which is parametrised by two values, a standard deviation parameter $\sigma_{\rm GP}$ and a characteristic timescale parameter $\rho_{\rm GP}$. The Matérn 3/2 kernel is defined by

$$k(\tau; \sigma_{\rm GP}, \rho_{\rm GP}) = \sigma_{\rm GP}^2 \left(1 + \frac{\sqrt{3}\tau}{\rho_{\rm GP}}\right) \exp\left(-\frac{\sqrt{3}\tau}{\rho_{\rm GP}}\right) \quad , \tag{6.10}$$

where $\tau = |t_n - t_m|$. A single element of the data covariance matrix $\mathbf{\Sigma}$ is then

$$\mathbf{\Sigma}_{nm} = \sigma_n^2 \delta_{nm} + k(\tau; \sigma_{\rm GP}, \rho_{\rm GP}) \quad , \tag{6.11}$$

where σ_n is the error bar for the n-th data point (a free parameter). Since we fit multiple independent light curves the total log-likelihood is the sum of individual likelihoods defined in Equation (6.8).

6.4 The inverse problem

Having defined a probabilistic model which describes how to compute a realistic light curve for an occultation of Io in the previous section, in this section we discuss the inverse problem of inferring a surface map by fitting a set of occultation light curves in a Bayesian framework.

6.4.1 The information content of a light curve

The mapping problem is famously ill-posed, meaning that specific linear combinations of spherical harmonic coefficients will be in the nullspace of the linear mapping $\mathbf{A}$ in Equation (6.4) (Luger et al., 2021c). This means that in general, even if we had noiseless observations, it would still be impossible to recover certain features on the surface. To recover the greatest information about the surface we need to have a mechanism which breaks the various degeneracies. For example, with phase curves, we can recover primarily longitudinal variations in emission. Occultations are substantially better because the limb of the occultor sweeps across the surface of the occulted sphere, thereby exposing or blocking light from different points on the surface. An ideal set of observations would consist of phase curves together with observations of multiple occultations by a small occultor at different latitudes and different phases. Phase curves and occultations in reflected light are even more informative because of the nonuniform illumination profile of the incident radiation and the presence of a day/night terminator line (Luger et al., 2022b). In some cases for reflected light observations (phase curves of an inclined planet for example), there can even be no nullspace *at all* for low spherical harmonic degrees.

We can reformulate these statements on how useful given observations are more precisely by computing a measure of their information content. Given that our model is linear, assuming Gaussian priors on the spherical harmonic coefficients with covariance $\mathbf{\Lambda_y}$ and a Gaussian likelihood, the posterior can be computed analytically and its mean is given by

$$\widehat{\mathbf{y}} = \mathbf{\Sigma}_{\hat{\mathbf{y}}} \left(\mathbf{A}^\top \mathbf{\Sigma}_{\mathbf{f}}^{-1} \mathbf{f} + \mathbf{\Lambda}_{\mathbf{y}}^{-1} \boldsymbol{\mu}_{\mathbf{y}}\right) \quad , \tag{6.12}$$

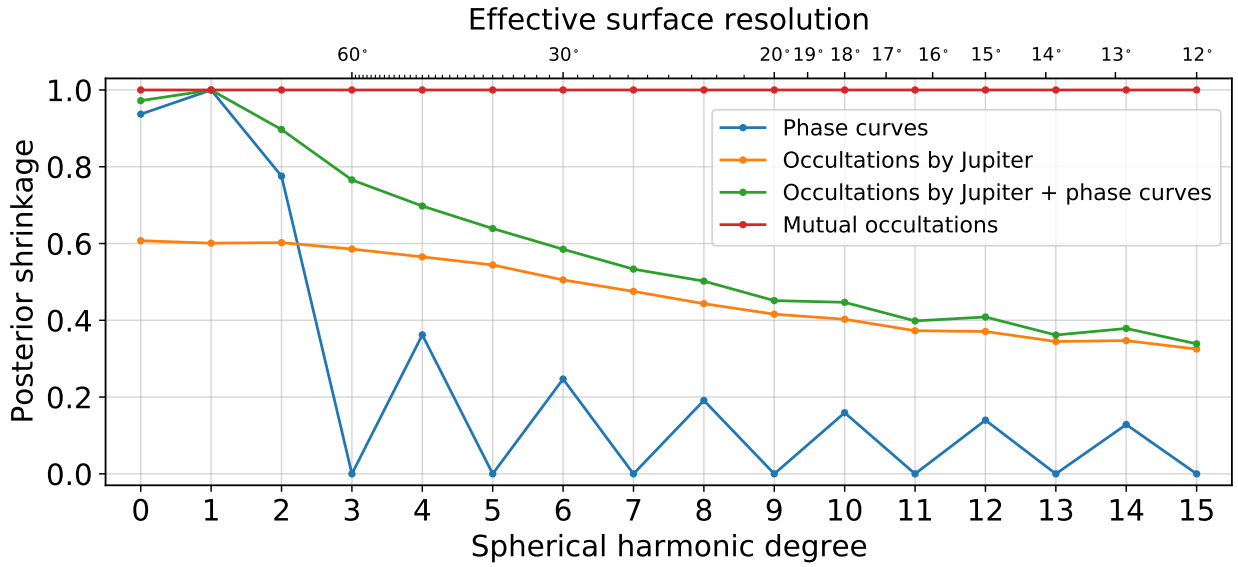


Figure 6.3: The posterior shrinkage for different kinds of observations of Io as a function of spherical harmonic degree (angular scale), averaged across all m modes. Posterior shrinkage of 1 represents maximum information gain in updating from the prior to the posterior while 0 represents no information gain. The posterior variance has been computed for different kinds of simulated observations of Io over the course of a single year: phase curves (blue), occultations by Jupiter (orange), combined phase curves and occultations by Jupiter (green) and occultations of Io by other Galilean moons (red lines).

where the posterior covariance matrix $\mathbf{\Sigma}_{\hat{\mathbf{y}}}$ is

$$\mathbf{\Sigma}_{\hat{\mathbf{y}}} = \left(\mathbf{A}^\top \mathbf{\Sigma}_\mathbf{f}^{-1} \mathbf{A} + \mathbf{\Lambda}_\mathbf{y}^{-1}\right)^{-1} \quad . \tag{6.13}$$

We define the information content as the variance reduction of the posterior relative to the prior which is called *posterior shrinkage*. The posterior shrinkage S is defined as (Luger et al., 2021c; Betancourt, 2018):

$$S \equiv 1 - \lim_{\sigma_0^2 \to \infty} \frac{\sigma^2}{\sigma_0^2} \quad , \tag{6.14}$$

where σ_0^2 is the prior variance and σ^2 is the posterior variance for a given spherical harmonic coefficient. It tells us how well we can constrain a particular coefficient in the limit of infinite SNR observations. Posterior shrinkage of 1 indicates that the data provides perfect information on the parameters while 0 indicates no gain in information relative to the prior.

To compute the posterior shrinkage we first compute the design matrices $\mathbf{A}$ using starry for different kinds of observations of Io for the period starting on the 1st of January 2009 and ending on the 1st of May 2010. We chose this period because it covers the full season of mutual occultations which occur every 6 years. We compute $\mathbf{A}$ for all observable occultations of Io by Galilean moons during that period, for occultations of Io by Jupiter and for phase curve observations. The purpose of this is to determine the upper bound on what we can learn about the surface. We take the ephemeris from JPL Horizons and assume that it is known exactly; we also assume all observations are observations of thermal emission independent of whether Io is in sunlight or in eclipse because we are interested in constraining the volcanic emission rather than the albedo.

Fig. 6.3 shows the posterior shrinkage as a function of l (averaged over all m modes) for phase curve observations (blue lines), occultations by Jupiter (orange), the former two combined (green), and mutual occultations by other Galilean moons (red). As expected, the mutual occultations of Io by other Galilean moons are by far the most informative with posterior shrinkage of unity at all angular scales considered. Occultations by Jupiter are less informative because we only see one side of Io during an occultation. Shrinkage for phase curves at odd degrees above $l = 2$ is exactly zero because these coefficients are in the nullspace for objects rotating about an axis perpendicular to the line of sight and therefore cannot be constrained using only phase curves (Luger et al., 2021c). Although observations of mutual occultations most easily break the degeneracies because of the varying impact parameters and different sizes of occultors, the drawback of these types of observations is that they only happen every 6 years and they almost never happen while Io is in eclipse which means that only the brightest volcanoes are visible above the reflected sunlight. This fact is not captured in Figure 6.3.

Figure 6.3 gives us some idea about which kinds of observations are most informative but it doesn't really tell us how well we can constrain bright spot-like features we expect to see on Io. To answer this question we have to create a simulated dataset and conduct the whole inference process.

6.4.2 Pixels vs. spherical harmonics

We use `starry` to generate a single simulated light curve of an occultation of Io by Jupiter from an $l = 30$ map with known coefficients. The simulated map consists of a spherical harmonic expansion of a bright spot with a Gaussian profile which we add to a uniform brightness map using the built-in `add_spot` function in `starry`. The expansion is in the quantity $\cos(\Delta\theta)$ where $\Delta\theta$ is the angular separation between the centre of the spot and another point on the surface of the sphere. We place the spot at 13° N latitude and 51° E longitude, we set the diameter of the spot ($2\Delta\theta$) to 5° and we set the amplitude of the spot such that the total *luminosity* of the map increases by 50% with the addition of the spot. We generate two light curves with 150 data points each, one for the duration of the ingress of the full occultation and the other for the egress. Because the limb of the occultor sweeps over the disc of Io at different angles during ingress and ingress, this makes it possible to break most degeneracies in the map and recover the location of the simulated spot. We set the phase of the simulated map to be 10° E at the beginning of ingress and 10° W at the end of an egress. We assume that the geometry of the occultation is known exactly and we set the error bars such that SNR = 50 where the signal is defined to be the maximum value of the computed flux. We use this dataset to test the difference between setting a prior in the spherical harmonic basis $\mathbf{y}$ and in the pixel basis $\mathbf{p}$ by fitting an $l = 20$ map to the dataset.

In the spherical harmonic model, we place a Gaussian prior on $\mathbf{y}$ with covariance $\mathbf{\Lambda} = \text{diag}(1^2, 0.5^2, \ldots, 0.5^2)$. Since the model is linear and the prior is Gaussian, the posterior probability distribution is also Gaussian and we can solve for the posterior mean (Equation (6.12)) and covariance (Equation (6.13)) analytically. In the hybrid pixel model, we place a positive exponential prior on the pixels which are defined on a Mollweide grid. The purpose of the exponential prior is to favour sparser solutions for the map because the exponential distribution pushes most pixels towards zero intensity. We use four times as many

pixels as spherical harmonics. Although the model is also linear in this case, the posterior distribution for the pixels is not analytic because of the non-Gaussian prior so we sample the posterior using MCMC instead. As the end product of inference we save the spherical harmonic coefficients $\mathbf{y} = \mathbf{P}^\dagger \mathbf{p}$ rather than the pixels themselves. The coefficients $\mathbf{y}$ can then be used to evaluate the map on a pixelated grid of arbitrary resolution via the matrix $\mathbf{P}$.

To ensure that the difference in the inferred maps is not in part due to a difference in the scale of the priors, we take 5000 samples from the prior on $\mathbf{y}$ and evaluate $\mathbf{P}\,\mathbf{y}$ for each; we then compute the standard deviation of these pixels and use that as the scale parameter in the exponential prior. We implement the pixel model in the probabilistic programming language numpyro (Phan et al., 2019) which is built on top of the JAX library, and fit it using the No-U-Turn-Sampler (NUTS). We run the chains for 1000 tuning steps and 2000 final steps, monitoring divergences (Betancourt and Girolami, 2013) and the $\hat{R}$ diagnostic (Gelman and Rubin, 1992) to check for convergence. Since all models we fit in this paper have hundreds if not thousands of parameters, it would be extremely challenging to sample the posterior without the use of automatic differentiation and Hamiltonian Monte Carlo (at least for non-Gaussian priors).

To visualise the inferred maps we plot the heatmap of the median intensity at each point on the map computed from posterior samples. Results are shown in Figure 6.4. The top row shows the simulated map, beneath it we show the inferred maps in Mollweide projection for the two models (second row), the data and posterior flux samples (orange lines) and the residuals with respect to the median flux (bottom row). The difference between the two models is striking. The left map has an elongated feature which does not resemble the spot in the simulated map while the map on the right is nearly identical (except for a difference in intensity) to the simulated map. We should emphasize here that the fact that the pixel model results in a spot-like map is primarily a consequence of the exponential prior which favours sparse solutions in pixel space. When we compared the spherical harmonic model to the pixel model using a prior with a lighter tail such as a Half Gaussian (Gaussian truncated at zero), we obtained a more elongated feature similar to that shown on the left map in Figure 6.4, but the map was still noticeably less complex because the positivity constraint substantially reduces the space of maps which fit the data well. Thus, the benefit of using the pixel model makes it easy to impose the positivity constraint on the map but also other constraints such as sparsity.

An important issue with the map on the right is that there is a series of concentric rings around the spot which result in a wave-like pattern in the predicted flux and the residuals. This ringing pattern arises because representing spot-like features requires constructive interference between different spherical harmonic modes inside the spot and destructive interference elsewhere. The pattern is more pronounced when we fit a low-resolution map (approximately $180/20 = 9^\circ$ in this case) and the model tries to represent a feature below the resolution of the map.

Ringing is also the reason why the inferred spot for the pixel model appears to be noticeably dimmer than the simulated spot. There is non-negligible leakage of total flux from the spot into the rings surrounding it, meaning that if we were to integrate the map intensity over a region encompassing the brightest part of the inferred spot, it would be an underestimate of the total emitted flux from the true spot within the same area. Ringing is also undesirable because we want to avoid situations in which the model uses the rings to

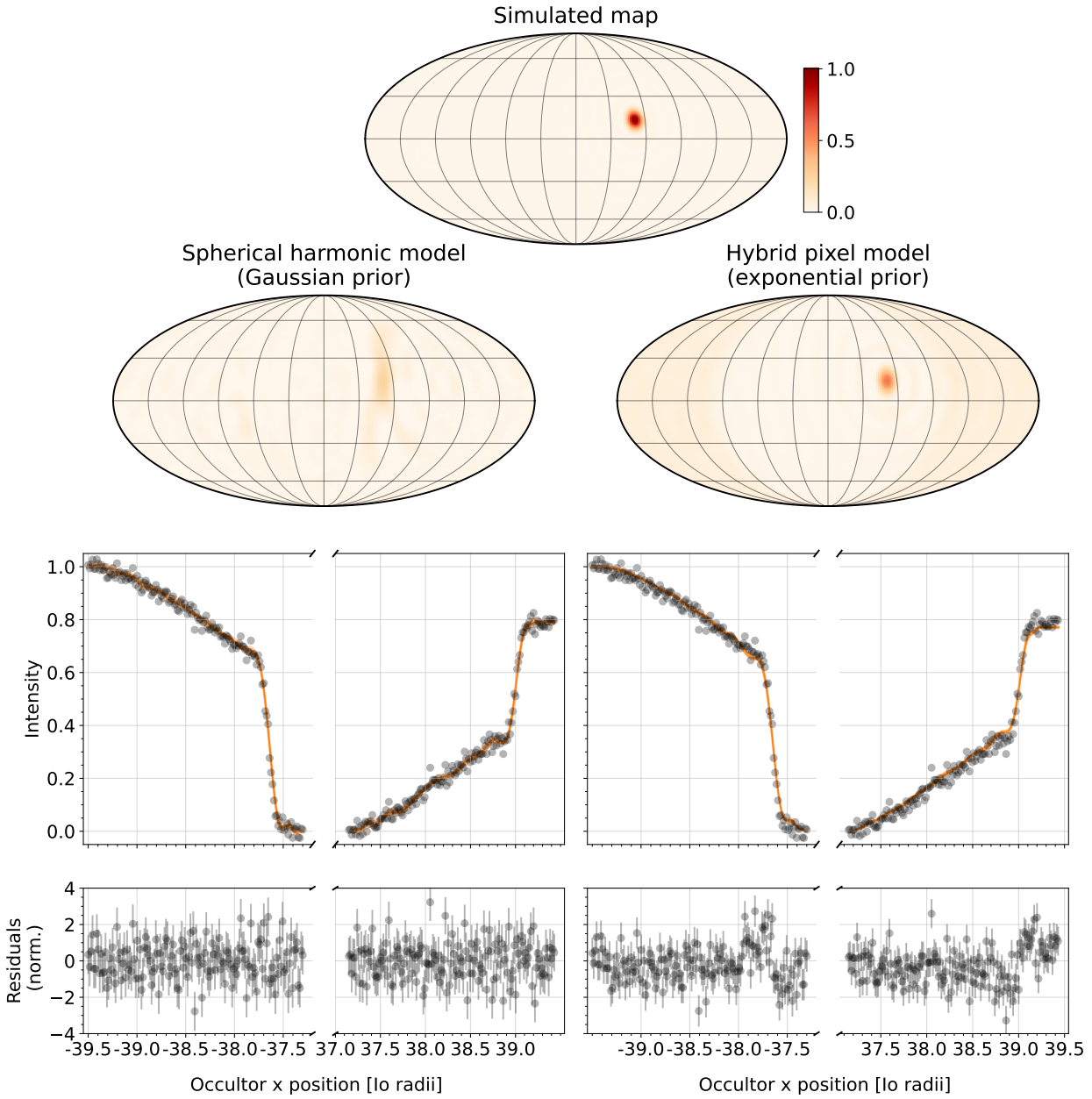


Figure 6.4: Comparison between two models fitted to a simulated light curve of an occultation by Jupiter which was generated from a map consisting of a single bright spot (top map). Each of the columns beneath the simulated map shows the (median) posterior estimate of the map (top), the data and the posterior flux samples (middle row) and the residuals (bottom row). The left column corresponds to a model in which we place a Gaussian prior on spherical harmonic coefficients $\mathbf{y}$ and fit for $\mathbf{y}$ while the right column corresponds to a hybrid model in which we fit for pixels $\mathbf{p}$ (with an exponential prior on pixel intensity) but we use $\mathbf{y}$ to compute the flux analytically with starry. The benefit of using pixels is that it is much easier to encode assumptions on what the map should look like in pixel space rather than spherical harmonic space, this results in a visibly more accurate inferred map. The residuals for the hybrid pixel model show undesirable patterns due to ringing artefacts which are a consequence of the Exponential prior favouring more localised features, we discuss how to alleviate this issue in the text.

explain the data instead of just placing a spot directly. For example, we find that in some cases when we fit low-degree maps, the model would place a bright spot on the unobserved side of Io in order to produce a ringing artefact on the observed side to explain an increase or decrease in brightness in the light curve. Fortunately, convolving the map with a spatial

smoothing filter prior to evaluating the flux fixes this issue. We describe how to apply the smoothing filter in the following section.

6.4.3 Smoothing out spurious features

To suppress ringing artefacts which appear around inferred spots such as the one shown in Figure 6.4, we apply a spatial smoothing filter to the spherical harmonic coefficients. Mathematically, the filtering operation is a convolution between the map and some kernel function $B(\theta, \phi)$. Assuming both the map and the kernel function are expanded in terms of spherical harmonics, the convolution operation is simply a multiplication between the two sets of spherical harmonic coefficients. We use a Gaussian-like kernel function given by

$$B(\theta) = \frac{\exp\left(-\theta^2/2\sigma_s^2\right)}{2\pi\sigma_s^2} \quad , \tag{6.15}$$

where σ_s is a parameter which sets the characteristic scale of the smoothing. This function can be expanded in terms of spherical harmonics as

$$B(\theta) = \sum_{l=0}^{\infty} \left(\frac{2l+1}{4\pi}\right) B_l \, \mathcal{P}_l(\cos\theta) \quad , \tag{6.16}$$

where B_l are the spherical harmonic coefficients and $\mathcal{P}_l$ are the associated Legendre polynomials. They depend only on l because all nonzero m modes vanish due to azimuthal symmetry. For $\sigma_s \ll 1$, B_l can be approximated as (Seon, 2007; White and Srednicki, 1995)

$$B_l \simeq \exp\left[-\frac{1}{2} l(l+1)\sigma_s^2\right] \quad . \tag{6.17}$$

The effect of this filter is to exponentially suppress features on scales smaller than $l \sim \sigma_s^{-1}$.

Figure 6.5 shows the effect of (Gaussian) smoothing on a spherical harmonic expansion of a spot with a Gaussian intensity profile. We place the spot at 0° latitude and longitude and set the size of the spot $\Delta\theta$ to 5°. All three panels show the exact profile of the spot (black line) and expansions up to three different orders l (coloured lines). The panel on the left shows the expansion with no smoothing ($\sigma_s = 0$) in which case the symmetric ringing around the centre of the spot is clearly visible even at relatively high order ($l = 20$). The middle panel shows an intermediate level of smoothing with $\sigma_s = 0.1$, meaning that all features on scales above $l \approx 10$ are exponentially suppressed. The negative ringing is a lot less visible, albeit at the cost of having a slightly larger spot because suppressing higher-order harmonics necessarily means that we lose some ability to represent smaller-scale features. For $\sigma_s = 0.2$ (right) there is practically no ringing, but the expansions at $l = 10$, $l = 15$ and $l = 20$ result in the spot of the same size because all coefficients above $l = 5$ are significantly suppressed.

Thus, there is a trade-off between smoothing and the ability to resolve smaller-scale features in maps. In principle, we can always get rid of ringing by fitting sufficiently high-order maps. In practice, the analytic integrals computed in `starry` become computationally unstable above $l \approx 20$ so instead of going to very high order we apply some smoothing to mitigate the ringing. We find that setting $\sigma_s = 2/l$ where l is the order of expansion of the map is a good default setting for σ_s.

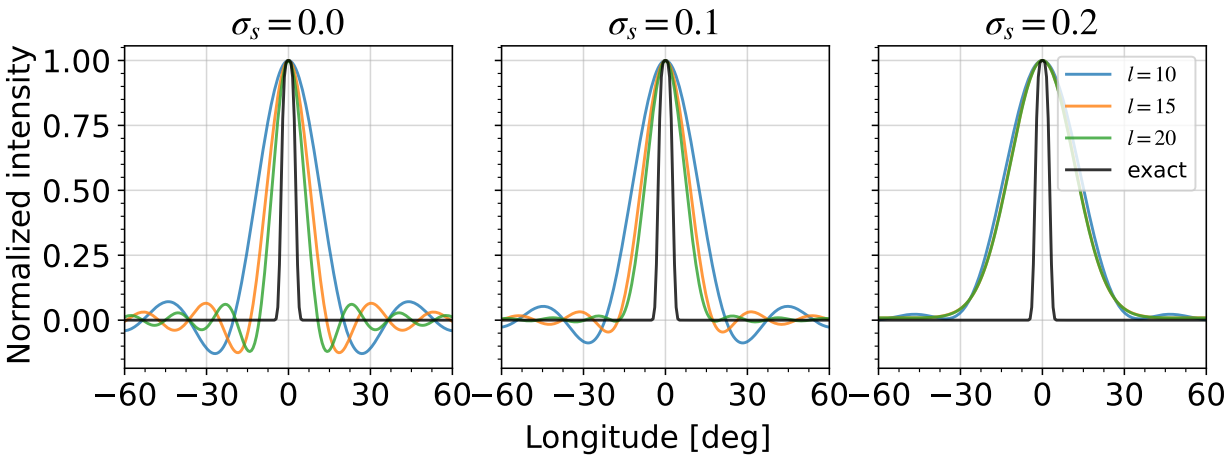


Figure 6.5: The normalised intensity of a spherical harmonic expansion of a Gaussian spot placed at 0° latitude and longitude. The expansion is in the quantity $\cos\Delta\theta$ where $\Delta\theta = 5°$. The three plots show the spot profile for increasing values of the smoothing parameter $\sigma_s = 0$. The coloured lines correspond to spot expansions up to a certain order and the black line is the exact expansion. The purpose of smoothing is to taper higher-order spherical harmonic coefficients in order to suppress ringing artefacts which result in negative intensities. High levels of smoothing suppress the ringing completely but as a result, they increase the spot size and eliminate differences between expansions above a certain order. Intermediate levels of smoothing provide a compromise between the two extremes.

6.5 Results – simulated data

6.5.1 Fitting simulated ingress/egress light curves

In this section, we generalise the example from Section 6.4.2 such that the simulated map includes an additional faint hot spot located at −15° N latitude and −40° E longitude with a diameter of 5°. We set the amplitude of the "faint" spot to 30% of the luminosity of the featureless map. As before we assume that we know the geometry of the occultation exactly. We generate the light curves from an $l = 30$ map and fit an $l = 20$ map and we apply a Gaussian smoothing filter to both the simulated map and the inferred map with the smoothing parameter set to $\sigma_s = 2/l = 0.1$. We fit the model using NUTS with 1000 warm-up steps and 2000 final steps.

We found that in cases where the simulated map consists of a bright and a faint spot the pixel model with an exponential prior such as the one we used in Section 6.4.2 does not recover the faint spot. Instead, we use a different heavy-tailed prior called the Regularized Horseshoe prior[1] (also known as the "Finnish Horseshoe") (Piironen and Vehtari, 2017). This prior is specifically designed for use in Bayesian sparse linear regression. It is an improvement on the Horseshoe prior introduced in Carvalho et al. (2010). The key idea behind both kinds of Horseshoe priors is to set the scale for each regression coefficient (pixel) to a product of a global scale τ and a local scale λ_i (where i indexes all the pixels) and we marginalise over these scales by setting a prior for each. For clarity, we omit the discussion of the Horseshoe priors here and refer the reader to Appendix B.

The results for a high signal-to-noise light curve (SNR=50) are shown in Fig. 6.6: the model recovers both spots. The plot shows the simulated map (top row), the median posterior

[1]To be precise, we use the truncated version of the Regularized Horseshoe prior where the coefficients are required to be positive.

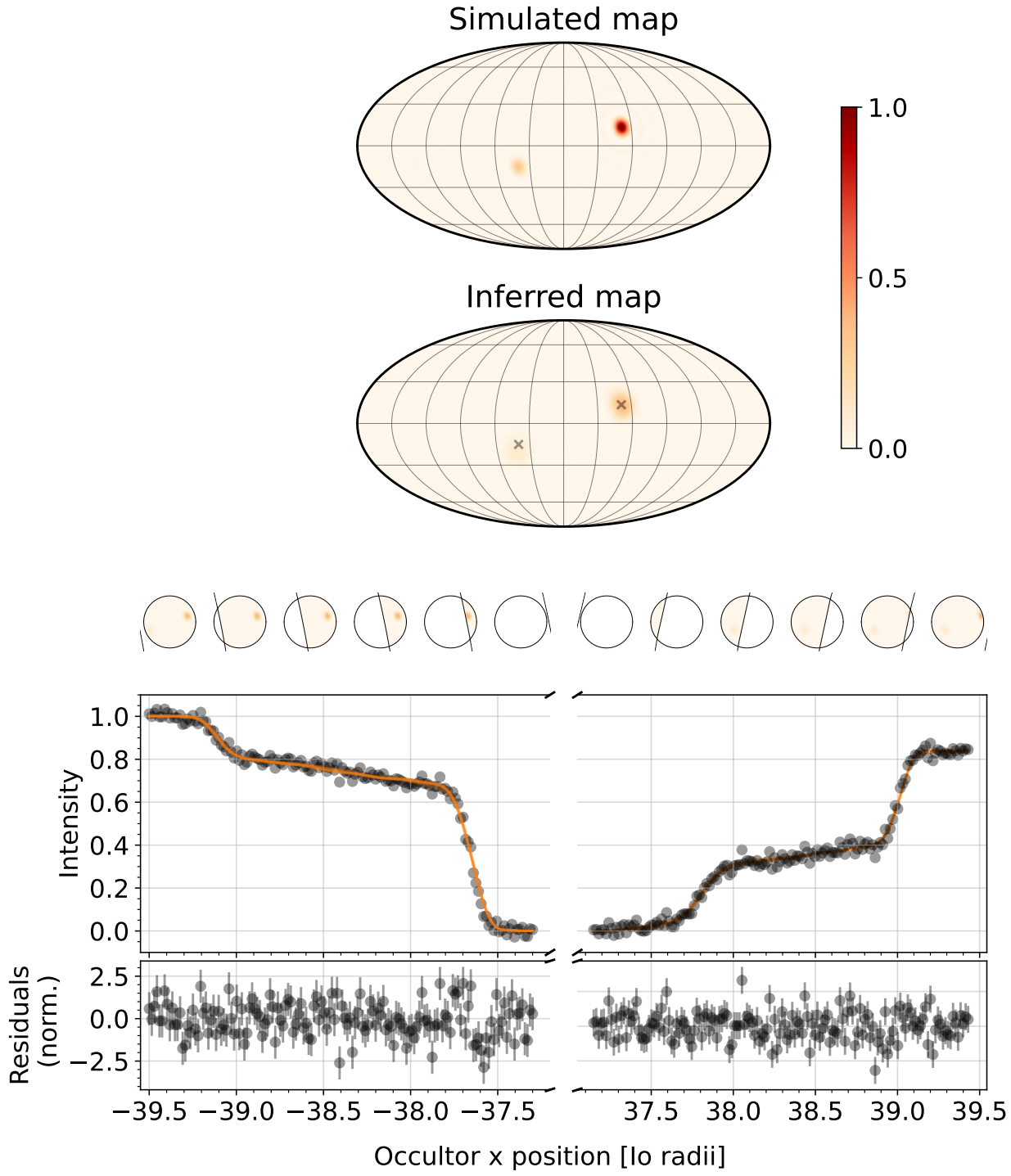


Figure 6.6: Inferred $l = 20$ map obtained by fitting a pair of simulated ingress/egress occultation light curves of Io using a hybrid pixel model with a Regularized Horseshoe prior on the pixels. With this prior, the model is able to accurately recover the simulated map. The plot shows the simulated $l = 20$ map (top), the posterior (median) estimate of the inferred map (second row), the inferred map as seen by the observer during the occultation (small circles), the data and posterior samples of the flux (orange lines), and the normalised residuals (bottom). The location of the simulated spots on the inferred map is marked with a grey cross (X).

estimate of the inferred map (second row), the inferred map as seen by the observer during the occultation (small circles), the data and posterior samples of flux (orange lines) and the residuals with respect to a median estimate of the flux (bottom). The location of the simulated spots on the inferred map is marked with a grey cross (X). The bright spot is nearly indistinguishable from the simulated spot; the fainter spot is somewhat less well constrained but the error in position for both is at most a few degrees. Ringing artefacts are minimal because of the smoothing filter and there are no discernible patterns in the residuals. We found that without the Horseshoe priors the model was not able to capture the first step in the light curve which is due to the fainter spot coming in or out of view; it would only recover the brighter spot. In Fig. 6.7 we show the output of the same model assuming we have data of worse quality (SNR=10). In this case the model still recovers both spots but the error in the position of the spots is greater.

6.5.2 Comparison to a parametric model

It is instructive to compare the model developed in the previous section to a parametric model in which we assume that the map contains some fixed number of spots and we fit for the positions, amplitudes and sizes of the spots. A parametric spot model might be useful if we want to fit a small number of spots and we know what the map should look like. To test how the parametrised spot model compares to the model defined in Section 6.5.1 we fit it to the light curves shown in Figure 6.6. The model consists of 4 parameters: the latitude, longitude, amplitude and size of the spot. We place uniform priors on the angles and Half-Gaussian priors on other parameters. We find that if the number of modelled spots matches the number of simulated spots the model easily converges to the true solution. If the number of modelled spots is larger or smaller than the true number of spots the model does not converge to the true solution due to pathologies in the posterior distribution.

Although a parametric approach would likely have been sufficient for modelling the light curves shown in this work, it does not work well when the number of spots is unknown or if the features aren't spots. There is no need to restrict ourselves to a parametric model because the pixel model gives the same results with only weak assumptions about the global structure of the map. The computational advantages of the parametric spot model relative to the pixel model are minimal. Even though the spot model has only 5 parameters per spot and the pixel model has thousands of parameters, the runtime for the pixel model is longer only by a factor of a few.

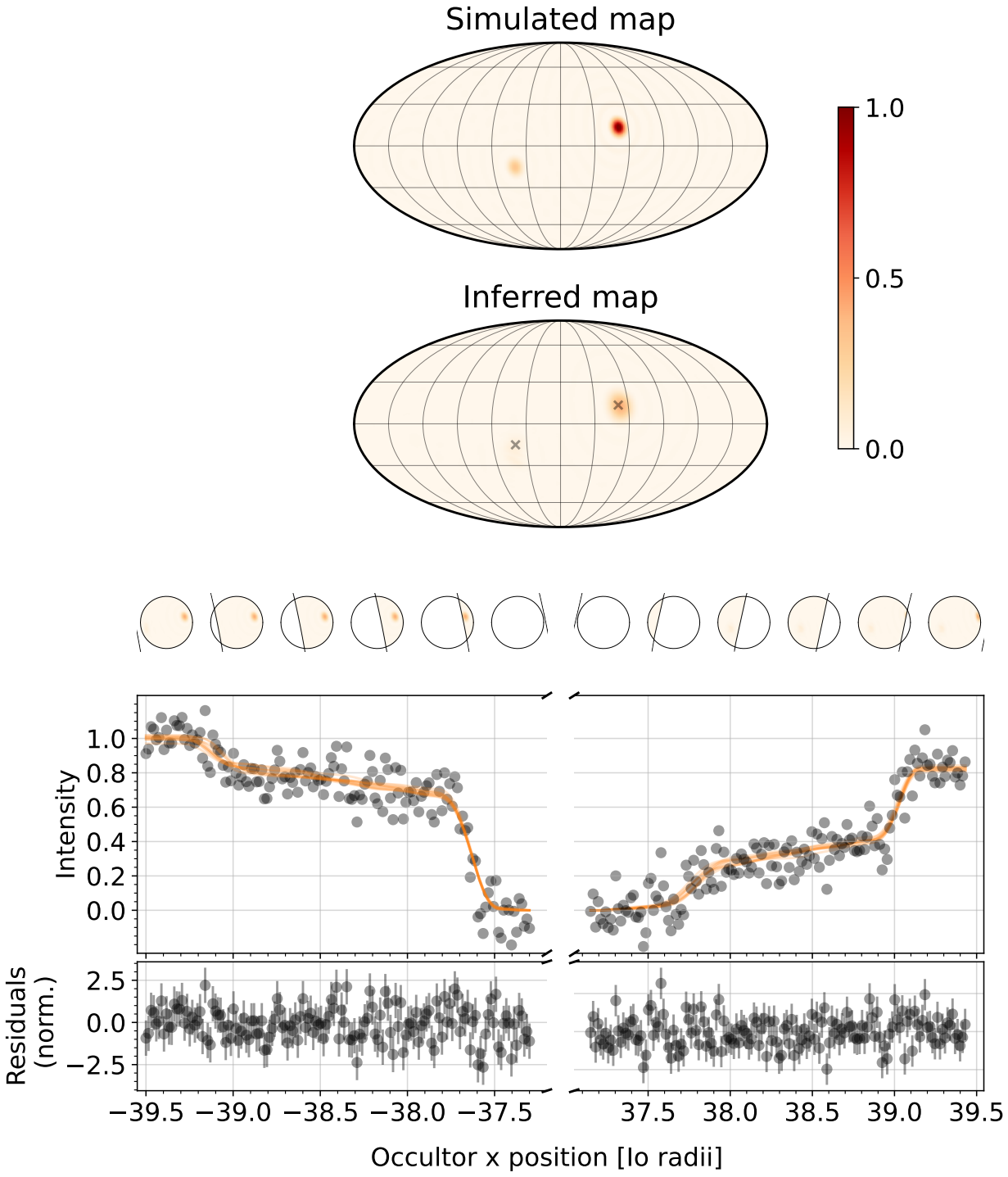


Figure 6.7: Same as Fig. 6.6 except for light curves with SNR=10.

6.6 Results – IRTF light curves

6.6.1 1998 pair of occultations by Jupiter

Having demonstrated that our model works well on simulated data, we turn to fitting observations of actual occultations of Io observed with the IRTF telescope. We selected two pairs of high-quality ingress/egress light curves relatively closely spaced in time so that the assumption that the surface map is identical for both the ingress and the egress light curve is at least approximately correct, although we allow for a difference in the overall amplitude of the maps between the two occultations. We fit a pair of events from 1998, an ingress occultation observed on the 27th of August and an egress occultation observed 94 days later on the 29th of November. As a reminder, the period of Loki's variability is around 540 days (Rathbun et al., 2002b). For comparison, we also fit a pair of events observed nearly two decades later in 2017.

The predicted fluxes for the ingress and egress of the occultation are given by

$$\mathbf{f}_I = \mathbf{A}'_I \, \mathbf{p}_I + b_I \, \mathbf{1}_{N_I} \tag{6.18}$$

$$\mathbf{f}_E = \mathbf{A}'_E \, \mathbf{p}_E + b_E \, \mathbf{1}_{N_E} \quad , \tag{6.19}$$

where $\mathbf{f}_I$ is the predicted flux for the ingress light curve with N_I data points, $\mathbf{f}_E$ is the predicted flux for the egress light curve with N_E data points, and b_I and b_E are constant flux offset parameters. Since we assume that the map is the same for both light curves up to an overall amplitude difference, we have

$$\mathbf{p}_E = a \, \mathbf{p}_I \quad , \tag{6.20}$$

where a is a dimensionless parameter.

The total log-likelihood is the sum of the log-likelihoods for individual light curves (Equation (6.8)). To compute it we need to specify the data covariance matrix defined in Equation (6.11). Each element of the covariance matrix consists of a white noise component (error bars) and a correlated noise component (Gaussian Process). Because the IRTF light curves from the preliminary archive of occultation light curves do not contain robust estimates of the errorbars and the number of data points per light curve is quite small, we choose to treat each errorbar as a free parameter and fit for all error bars using a hierarchical model. We assume that there is a common global scale l for all data points in a given light curve and that the error bar for each data point is drawn from a Half Normal distribution with a scale parameter equal to l. In other words, we have

$$\sigma_{i,I} \sim \text{Half} - \mathcal{N}(l_I), \quad i = 1, \ldots, N_I \tag{6.21}$$

$$\sigma_{i,E} \sim \text{Half} - \mathcal{N}(l_E), \quad i = 1, \ldots, N_E \quad , \tag{6.22}$$

where $\sigma_{i,I}$ are the errorbars for the ingress light curve, $\sigma_{i,E}$ are the errorbars for the egress light curve, l_I is the common scale for the ingress light curve errorbars and l_E is the scale for the egress light curve errorbars.

By fitting for error bars in this way we do not have to assume any parametric forms for the dependence of the variance of the data on the measured flux and we are also able to

Parameter(s)	Description	Prior
τ	global pixel scale	Half $-\mathcal{C}(0, \tau_0)$, τ_0 defined in Equation (B.3) with $p_0 = 0.8D$
c^2	slab scale squared	Inv$-\mathcal{G}(\frac{\nu}{2}, \frac{\nu}{2}s^2)$, $s = 1000$ and $\nu = 4$
$\overline{\lambda}_i, \quad m = 1, \ldots, D$	local pixel scale	Half $-\mathcal{C}(0, 1)$
$p_i, \quad i = 1, \ldots, D$	pixel intensities	Half$-\mathcal{N}(\tau\lambda_i)$, $\lambda_i = c\overline{\lambda}_i / \sqrt{c^2 + \tau^2\overline{\lambda}_i^2}$
a	relative change in map amplitude	$\mathcal{N}(1, 0.1)$
b_I, b_E	flux offset	$\mathcal{N}(0, 4)$
l_I, l_E	errorbar global scale	Half $-\mathcal{N}(0.1)$
$\sigma_{i,I}, \sigma_{i,E}, \quad i = 1, \ldots, N$	individual errorbars	Half $-\mathcal{N}(l_I)$, Half $-\mathcal{N}(l_E)$
$\sigma_{\mathrm{GP},I}$, $\sigma_{\mathrm{GP},I}$	GP standard deviation	Half $-\mathcal{N}(0.1)$, Half $-\mathcal{N}(0.1)$
$\rho_{\mathrm{GP},I}$, $\rho_{\mathrm{GP},E}$	GP timescale	Half$-\mathcal{N}(t_{max,I})$, Half$-\mathcal{N}(t_{max,E})$

account for outliers and heteroscedasticity in the data. Although this means that we are introducing hundreds of additional parameters in the model, this is not a problem for the Hamiltonian Monte Carlo sampler when the parameters are well constrained by the data and the model is regularized. The noise model as defined above is extremely flexible – a given feature in the light curve can be accounted for by inflating the error bars or varying the timescale or the variance of the Gaussian Process with the Matérn 3/2 kernel.

All model parameters and their associated priors are listed in Table 6.1. To fit the model we sample the posterior using the NUTS sampler for 1000 warm-up steps and 3000 final steps, monitoring divergences and the R-hat statistic to ensure convergence. The inferred map for the 1998 pair of events is shown in Figure 6.8 and the median, 16th and 84th percentile estimates of model parameters from the posterior samples are shown in Table 6.2. The inferred map has two distinct hot spots, a bright hotspot in the Eastern hemisphere and a faint hotspot in the western hemisphere. The orange lines visible in the second pair of panels from the bottom are posterior samples for flux from the complete model (map model plus the gaussian process noise model).

The bottom panel shows the residuals with respect to the median flux where each of the data points is shown with the posterior median estimate of its error bar. Both the global scale for the errorbars and the individual errorbars are well constrained by the data and the outlier points are naturally accounted for in our model because those points end up having a higher variance. Despite the fact that our noise model is very flexible, it does not seem to overfit the data because the two major steps in the light curves don't end up being

absorbed by the noise model. The Gaussian Process captures the variation in the data which is due to atmospheric variability, the emission due to surface features which are too faint to be well constrained by the physical model and also the limitation of the physical model to capture the true size of the spot. There is a trade-off between the explanatory power of the noise model and the physical model. In general, we expect that a given feature in the light curve will be accounted for by the physical model if the physical model provides a better explanation for the data than the noise model. We will return to this point shortly.

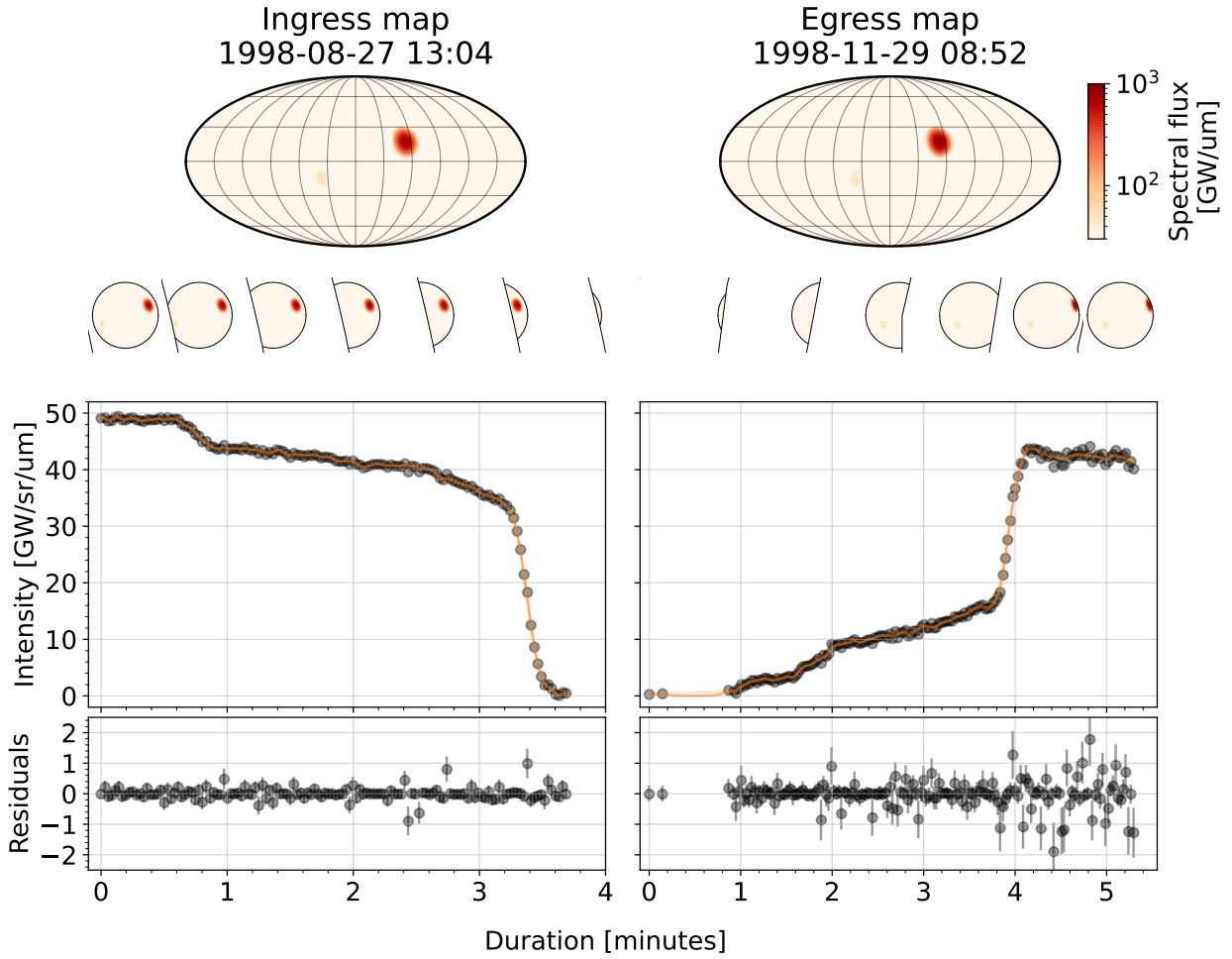


Figure 6.8: Inferred $l = 20$ maps obtained by fitting a pair of observations of occultations of Io by Jupiter in 1998. The observations were made several months apart with the NASA Infrared Telescope Facility (IRTF). We fit a single map to both observations simultaneously although we allow for a difference in the overall amplitude of the map between ingress and egress. The model includes a Gaussian Process to account for correlated noise caused by atmospheric variability and the fact that our limited resolution map cannot fully capture the sharp steps in data. We treat all error bars as random variables and plot the median posterior estimates of those error bars. The plot shows the inferred maps (top row), the same maps from the perspective of the observer during the occultation (small circles), the light curves and posterior samples of the flux including the Gaussian Process (orange lines), and the residuals with respect to a median flux estimate. The maps show two hotspots, the bright one is emission from Loki Patera and the faint one is most likely emission from Kanehekili. A detailed view of the two hot spots is shown in Figure 6.9.

To obtain some measure of the location of the inferred spots and their uncertainty, for each posterior sample of the spherical harmonic coefficients $\mathbf{y}$ we find the local maximum of intensity in a region around each of the spots and then compute percentile estimates of the

Parameter	Description	Value	Unit
τ	global pixel scale	$0.9^{+0.2}_{-0.2}$	intensity
c	slab scale	4000^{+2000}_{-1000}	intensity
a	relative change in map amplitude	$1.15^{+0.01}_{-0.01}$	dimensionless
b_I	flux offset ingress	$0.01^{+0.01}_{-0.01}$	GW/um/sr
b_E	flux offset egress	$0.3^{+0.4}_{-0.3}$	GW/um/sr
l_I	errorbar scale ingress	$0.25^{+0.04}_{-0.03}$	GW/um/sr
l_E	errorbar scale egress	$0.46^{+0.04}_{-0.04}$	GW/um/sr
$\sigma_{\mathrm{GP},I}$	GP standard deviation ingress	$0.63^{+0.05}_{-0.04}$	GW/um/sr
$\sigma_{\mathrm{GP},E}$	GP standard deviation egress	$0.45^{+0.05}_{-0.05}$	GW/um/sr
$\rho_{\mathrm{GP},I}$	GP timescale ingress	$0.08^{+0.01}_{-0.01}$	minutes
$\rho_{\mathrm{GP},E}$	GP timescale egress	$0.01^{+0.03}_{-0.02}$	minutes

Parameter	Value	Unit
Spot 1 latitude	$16.30^{+0.01}_{-0.01}$	degrees
Spot 1 (West) longitude	$306.90^{+0.01}_{-0.01}$	degrees
Spot 1 power ingress	52^{+1}_{-1}	GW/um
Spot 1 power egress	60^{+1}_{-1}	GW/um
Spot 2 latitude	-16^{+8}_{-4}	degrees
Spot 2 (West) longitude	$37.4^{+0.8}_{-0.2}$	degrees
Spot 2 power ingress	$5.4^{+0.6}_{-0.7}$	GW/um
Spot 2 power egress	$6.2^{+0.7}_{-0.8}$	GW/um

spot latitude and longitude. In addition to the locations of the spots, we also compute the total power emitted within a 15-degree range in latitude and longitude around the (inferred) centre of the spots. Both of these quantities are listed in Table 6.3. Figure 6.9 shows a contour map of both hot spots overlaid on top of the U.S. Geological Survey's map of the surface of Io[2] which was constructed from observations by the Galileo satellite. The plot shows the error for the inferred latitude and longitude for each spot (white lines) and a contour map computed from the median posterior estimate of the map. The contour lines correspond to the 5th, 50th, and 95th percentiles of intensity above an arbitrarily defined

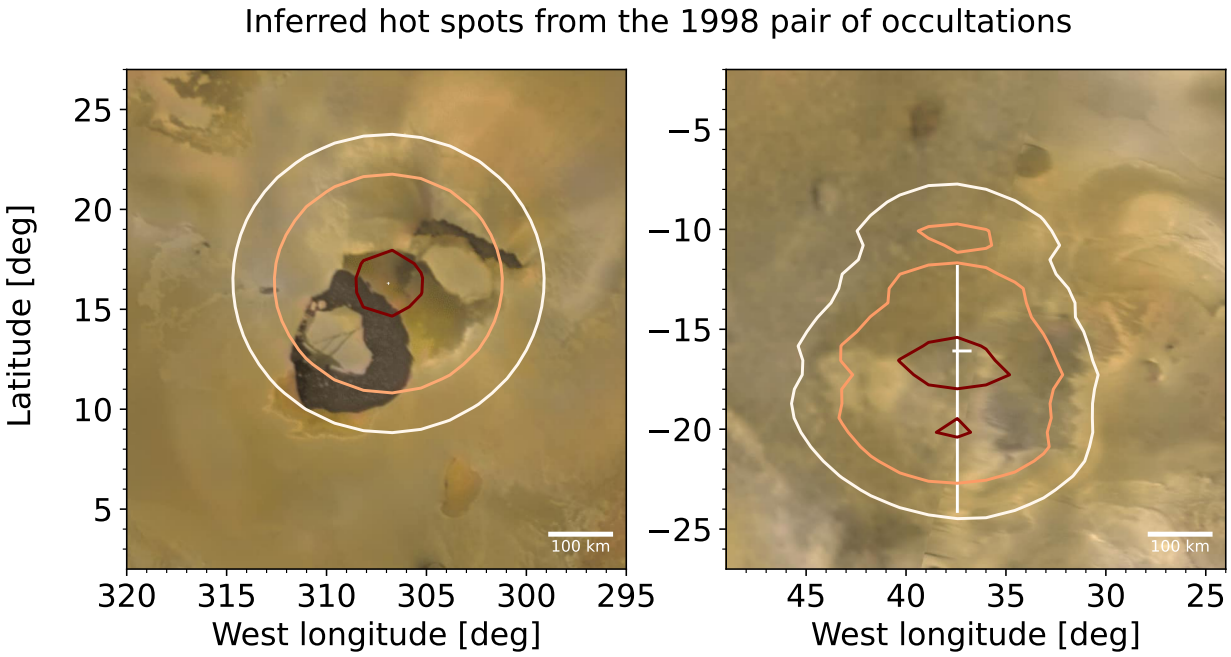


Figure 6.9: Contour plot of the hot spots shown in Figure 6.8 overlaid on top of the U.S. Geological Survey's map of the surface of Io which has been constructed from observations by the Galileo spacecraft. The left hotspot is centred around Loki Patera, the right hotspot is centred at the Kanehekili Fluctus lava flow. The contour lines show the 5th, 50th, and 95th percentiles of intensity above an arbitrarily defined intensity of the "background" region around the spot. The asymmetric white cross in the centre of the contours in each panel shows the uncertainty in the inferred position of the peak intensity of the hotspot. This uncertainty is so small for the spot on the left that the cross is barely visible. It is much larger for the spot on the right where the uncertainty in longitude is ~ 2 degrees and the uncertainty in the latitude of the spot is ~ 15 degrees.

Loki Patera coming out and into view during the occultation. The model inflates the error bars around those times because it is unable to make the spot small enough, this results in the single hot spot at the location of Loki Patera shown in Figure 6.8 morphing into two spots. While this feature is almost certainly spurious, the other spot in the northwestern hemisphere is more likely to be real. Looking at the miniature maps in Figure 6.10, this spot is in view only at egress and it corresponds to a small step in the light curve starting at 0.9 minutes. The estimated location of this spot is approximately ~ 20 ° N latitude and ~ 69 ° W longitude. This location corresponds to the southern end of the mountain Mongibello Mons but there are no known persistent hotspots at that location so we cannot say if the hotspots is real or not without independently detecting it in other light curves. Overall, we conclude that including the Gaussian Process prevents the physical model from overfitting the data although it might occasionally pick up a feature which is due to a faint real hot spot. With a particular scientific goal in mind, it is straightforward to experiment with different noise models to test how the properties of any given spot change with different assumptions.

One other notable feature in Figure 6.10 is that the first step in the ingress light curve at around 0.6 minutes is not fully accounted for by the model. Since the model had no trouble accounting for a similar step for simulated data shown in Figure 6.6. The reason why it did not do so in this case is likely because doing so would result in a poorer fit for the egress light curve. The two occultations were observed months apart so we expect that our assumption

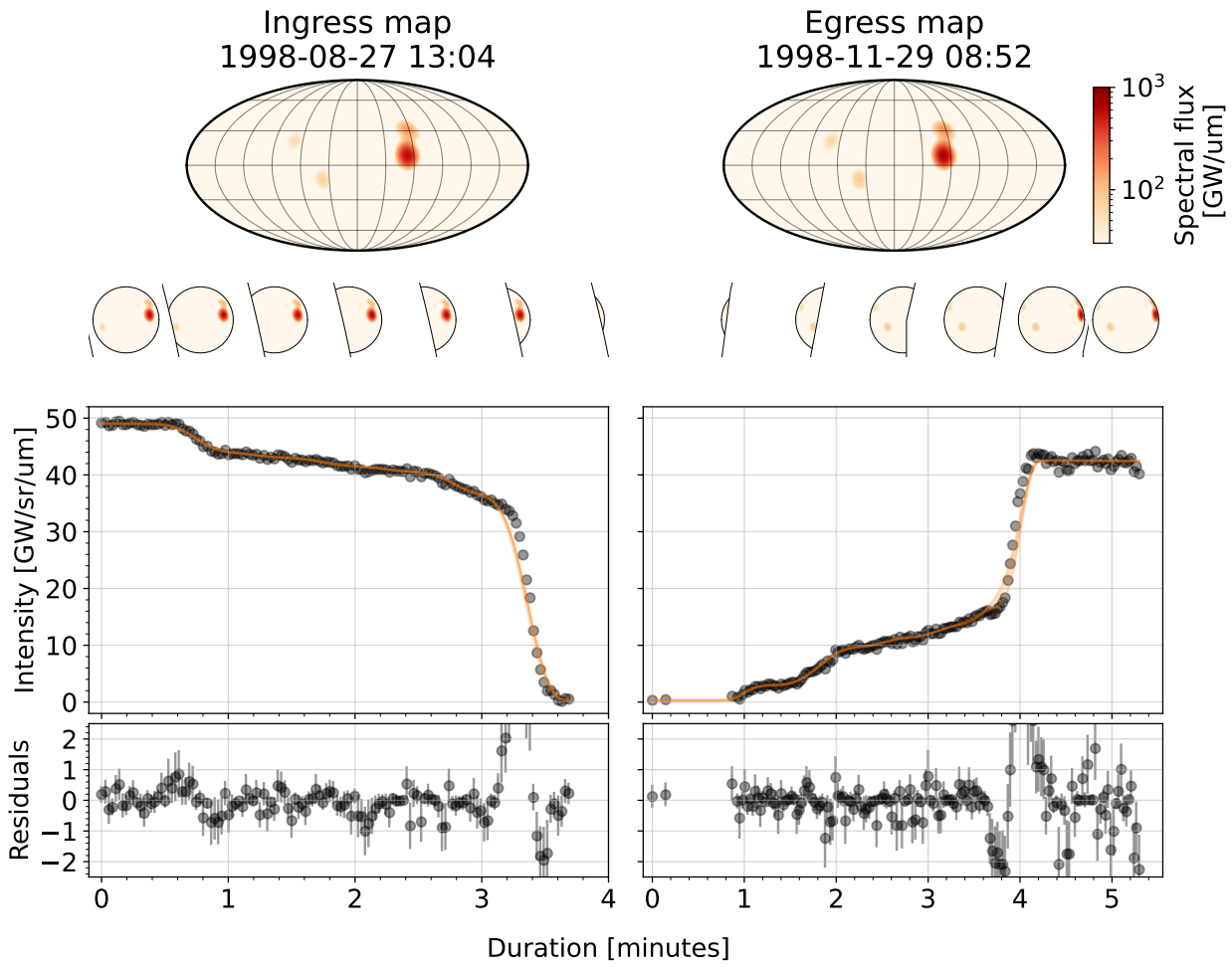


Figure 6.10: Same as Figure 6.8 except this figure shows the output of a model which does not include a Gaussian Process to account for correlated structure in the data. As a result, the model tries to capture the correlations in the data by placing two extra spots on the map and inflating the error bars. At least one of the two extra spots is artificial.

Parameter	Description	Value	Unit
τ	global pixel scale	$1.0^{+0.3}_{-0.2}$	intensity
c	slab scale	6000^{+3000}_{-2000}	intensity
a	relative change in map amplitude	$1.43^{+0.01}_{-0.01}$	dimensionless
b_I	flux offset ingress	$0.03^{+0.14}_{-0.02}$	GW/um/sr
b_E	flux offset egress	$1.0^{+0.2}_{-0.3}$	GW/um/sr
l_I	errorbar scale ingress	$0.97^{+0.06}_{-0.06}$	GW/um/sr
l_E	errorbar scale egress	$0.52^{+0.08}_{-0.09}$	GW/um/sr
$\sigma_{\mathrm{GP},I}$	GP standard deviation ingress	$0.13^{+0.13}_{-0.06}$	GW/um/sr
$\sigma_{\mathrm{GP},E}$	GP standard deviation egress	$0.60^{+0.08}_{-0.09}$	GW/um/sr
$\rho_{\mathrm{GP},I}$	GP timescale ingress	2.0^{+3}_{-2}	minutes
$\rho_{\mathrm{GP},E}$	GP timescale egress	$0.02^{+0.01}_{-0.01}$	minutes

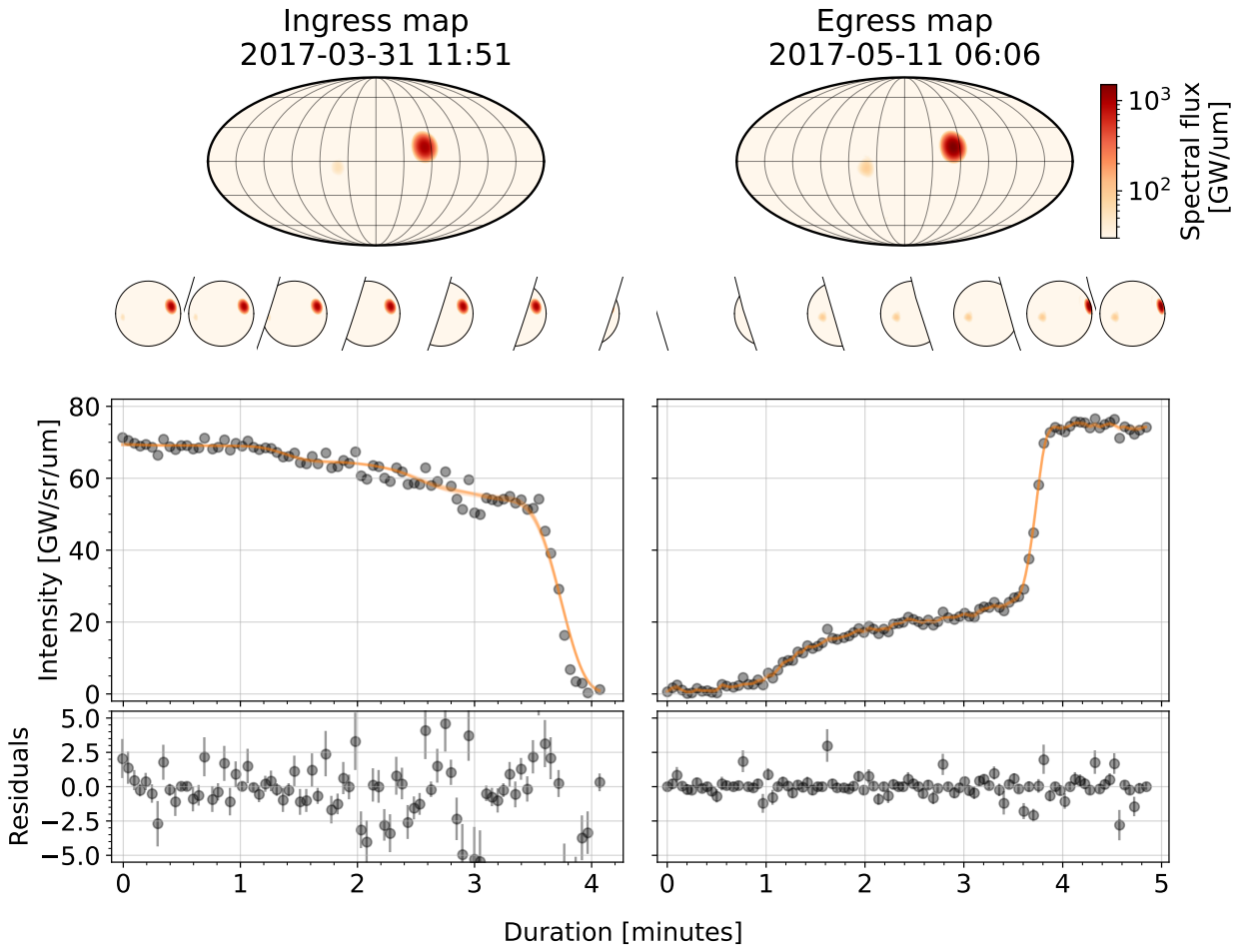


Figure 6.11: Inferred $l = 20$ maps obtained by fitting a pair of observations of occultations of Io by Jupiter in 2017. The observations were made several months apart with the NASA Infrared Telescope Facility (IRTF). We fit a single map to both observations simultaneously although we allow for a difference in the overall amplitude of the map between ingress and egress. The model includes a Gaussian Process to account for correlated noise caused by atmospheric variability and the fact that our limited resolution map cannot fully capture the sharp steps in data. We treat all error bars as random variables and plot the median posterior estimates of those error bars. The plot shows the inferred maps (top row), the same maps from the perspective of the observer during the occultation (small circles), the light curves and posterior samples of the flux including the Gaussian Process (orange lines), and the residuals with respect to a median flux estimate. The maps show two hotspots, the bright one is emission from Loki Patera and the faint one is emission from Janus. A detailed view of the two hot spots is shown in Figure 6.12.

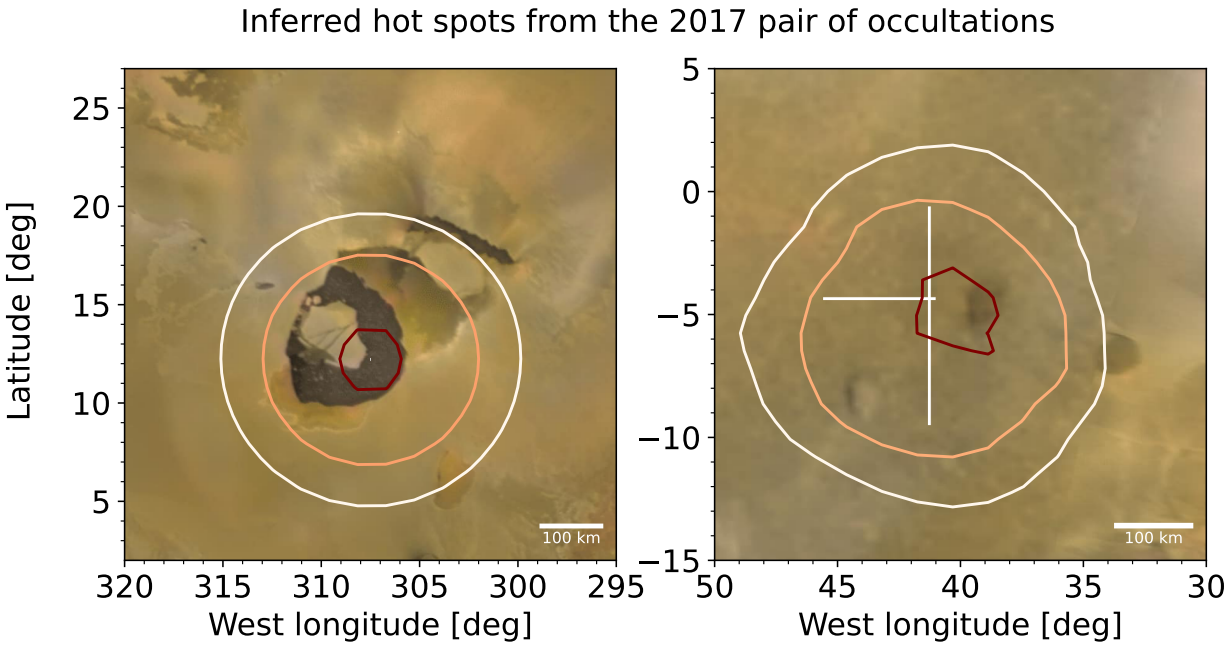


Figure 6.12: Contour plot of the hot spots shown in Figure 6.11 overlaid on top of the U.S. Geological Survey's map of the surface of Io which has been constructed from observations by the Galileo spacecraft. The left hotspot is centred around Loki Patera, the right hotspot is centred at Janus Patera. The contour lines show the 5th, 50th, and 95th percentiles of intensity above an arbitrarily defined intensity of the "background" region around the spot. The asymmetric white cross in the centre of the contours in each panel shows the uncertainty in the inferred position of the peak intensity of the hotspot (barely visible in the left panel).

6.7 Summary

6.7.1 Occultation mapping of Io

We have presented a novel method for mapping volcanic emission on Io using occultation light curves. The method relies on the starry algorithm which enables fast analytic computation of occultation light curves by expanding the surface emission map in spherical harmonics. Our method is different from past work because we do not assume that we know where the volcanic features are located on the surface of Io, how many there are or what they look like. Instead, we place weaker assumptions on the global structure of the map by requiring that the inferred map has positive intensity everywhere and most importantly, that it is *sparse*[3]. Besides the model for the surface map, our method also incorporates a sophisticated noise model which includes a Gaussian Process and a hierarchical model for the error bars. As a result of the sparsity and positivity constraints and the flexibility of the noise model, our model is parsimonious – it places features on the map only if the data provide strong evidence for the existence of those features. This property is not always desirable but it works very well for Io, a moon whose surface is covered with small, highly localised, and bright volcanic hotspots. Our model is substantially more flexible than parametric methods which assume some fixed number of spots on the surface of Io and parametrise the spot properties because we do not need to make such strong assumptions about the surface features. The computational cost is small and it is comparable to parametric models.

To test our method we first fit a simulated dataset and then observations of real occultations of Io by Jupiter, observed using NASA's Infrared Telescope Facility. We chose two pairs of light curves to demonstrate that the model can recover known hotspots. Each pair consists of an ingress observation and an egress observation of an occultation. The two observations are sufficient to break the degeneracy in the position of the spots because Jupiter's limb sweeps across the projected disc of Io at different angles at ingress and egress. From the 1998 observations, we infer a map consisting of two spots whose locations correspond to well-known hotspots on the surface of Io, the major volcano Loki and the lava flow Kanehekili. We also find circumstantial evidence for a third hot spot. Because our model is fully probabilistic, we can derive uncertainties on the location of the inferred hot spots and the total flux emitted from each spot. In addition to the pair of observations from 1998, we test the model on another pair of observations from 2017. We again find two hotspots, one of which is Loki Patera (although the peak of the intensity shifted relative to the 1998 map) and the other is Janus Patera.

The main limitation of our model besides the fact that we do not yet account for time-dependent maps is the limited resolution of the spherical harmonic maps for which starry can compute occultation light curves. Although we can still constrain the peak intensity of hotspots (or some other measure of the centre of emission) with much higher precision, we lose the ability to resolve two spots close to each other and the ability to constrain the actual size of the spots. It is possible that there is a way around this problem if we compute the various integrals in starry numerically using high precision arithmetic. Since all these operations are contained in the design matrix $\mathbf{A}$ (assuming that the ephemeris is fixed) we

[3] Aizawa et al. (2020) also find the sparsity assumption useful in the context of exoplanet mapping.

would only need to do the expensive computation once for each occultation. To be able to constrain the size of the Loki Patera hotspot, for example, we would need to fit maps on the order of $l \approx 50$. However, the kind of detailed mapping of the Loki Patera magma lake with a precision of a few kilometers done by de Kleer et al. (2017) will not be possible with a spherical harmonic model because it would require maps of extremely large degrees ($l \gtrapprox 700$).

Finally, it should be straightforward to extend our model for use in fitting *resolved* observations of Io (either occulted or not) such as the adaptive optics observations done by de Kleer and de Pater (2016b) and de Kleer and de Pater (2016a). Since we would not need to compute integrated flux over complicated boundaries in that case, we could fit maps of much higher order.

6.7.2 Relevance to the mapping of exoplanets

One of the motivations for this work was to test methods developed largely for the purpose of mapping exoplanets in a context where the ground truth is more easily accessible because Io is in the Solar System. To apply our model to exoplanet observations we simply need to compute the design matrix $\mathbf{A}$ given a specification of the planet's orbit. It is straightforward to do this in `starry`. Since the observations of the secondary eclipses (occultations) of exoplanets will never be as high quality as observations of Io, the limited resolution of spherical harmonic maps is more than sufficient for modelling exoplanet observations.

In Chapter 7 I build on the work presented and apply a similar model to simulated JWST secondary eclipse observations of a Hot Jupiter.

6.8 Time-dependent maps

A major issue with the model presented in this work is that we cannot naturally account for the time variability of surface emission. Here we briefly outline an extension of the model to time-dependent maps which could be applied to the entire sample of IRTF light curves. I did not have time to finish this work. The idea is to fit an ensemble of light curves (including occultations by Jupiter, mutual occultations and phase curves) with a model which assumes that the spherical harmonic map that generates each light curve is a linear combination of K "basis maps". This model would enable us to infer a time-dependent map of the entire surface and quantify the time variability over time scales of decades.

The model structure is as follows. Consider a collection of L lightcurves, such that the model for the l-th light curve is

$$\mathbf{f}_l = \mathbf{A}_l \, \mathbf{P}^\dagger \, \mathbf{p}'_l \quad . \tag{6.23}$$

We can stack the column vectors $\mathbf{p}_l$ into a matrix $\mathbb{Y}$ with shape (N_p, L) where N_p is the number of pixels for each light curve. We then assume that the $\mathbb{Y}$ can be decomposed into a product of two matrices $\mathbf{B}$ and $\mathbf{Q}$ as

$$\mathbb{Y} = \mathbf{B} \, \mathbf{Q} \quad , \tag{6.24}$$

where $\mathbf{B}$ has shape (N_p, K) and $\mathbf{Q}$ has shape (K, L). The model for all light curves can be written as

$$\mathbb{F} = \mathbb{A}\,\mathrm{vec}(\mathbb{Y}) \quad , \tag{6.25}$$

where $\mathbf{f}$ is a tall column vector consisting of predictions for all light curves stacked together, $\mathbb{A}$ is a block diagonal matrix with matrices $\mathbf{A}_l\,\mathbf{P}^\dagger$ on the diagonal for $l = 1 \ldots, L$ and the vec operator stacks the columns of $\mathbb{Y}$ into a tall vector. The model is *bilinear*, i.e., it is linear in the matrices $\mathbf{B}$ and $\mathbf{Q}$ separately. The interpretation of Equation 6.24 is simple, each of the K columns of $\mathbf{B}$ represents pixels of a basis map and the columns of $\mathbf{Q}$ determine how those maps add together to produce a final map for the l-th light curve. For this reason we call $\mathbf{B}$ the *basis matrix* and the matrix $\mathbf{Q}$ the *encoding matrix*. Ideally, we want the basis maps to be physically meaningful and the coefficients to encode the time variability of each basis map. An important feature of this model is that there is no requirement that successive maps are smoothly varying between different light curves which would be a poor model for Io.

I have explored variants of this matrix factorisation model by fitting it to simulated data and found it to be very promising but I did not have time to include it in this thesis.

Chapter 7

Mapping the "surfaces" of exoplanets

In the previous chapter, I presented a model for mapping the surface of Io using photometric observations in the near-infrared. A natural question that follows from that work is: can we reconstruct a spatial map of an exoplanet in the same manner? In this chapter, we test the feasibility of resolving spatially localised features on the daysides of tidally locked exoplanets using secondary eclipse mapping in emitted light. The chapter is based on an as-yet unpublished paper[1].

7.1 Introduction

One of the most notable advances in exoplanetary science over the past decade has been the reconstruction of very coarse two-dimensional spatial maps of exoplanets, using high-precision phase curves and secondary eclipse observations. Knutson et al. (2007), Majeau et al. (2012) and de Wit et al. (2012a) used Spitzer mid-infrared observations of secondary eclipses of the Hot Jupiter HD189733b and found that the emission is best described by the presence of a large hot spot on the dayside of the planet that is longitudinally offset from the substellar point. Similarly, Stevenson et al. (2014) constructed temperature maps of the Hot Jupiter WASP-43b, Demory et al. (2013) mapped the Hot Jupiter Kepler-7b in reflected light and Demory et al. (2016a) mapped the thermal emission from the Super Earth 55 Cancri e, although these studies were only able to capture longitudinal variations in intensity.

Real exoplanet atmospheres are certain to have three-dimensional spatial inhomogeneities in emission more complex than a single hot spot due to the presence of clouds, zonal jets, storms, waves etc. (Showman et al., 2020). Recent high resolution simulations of Hot Jupiter atmospheres by Cho et al. (2021) showed the presence of *storms* at a range of scales (including planetary scales) with quasi-periodic time variability and multiple equilibrium cycles. These atmospheric phenomena should lead to spatial variation in the intensity of emitted light (and also the albedo) across the visible disc of the planet.

With the launch of the James Webb Space Telescope (JWST), we are entering a new era of exoplanet science. JWST will be able to charecterise the atmospheres of gaseous exoplanets through spectroscopy. It will also be used for secondary eclipse observations of

[1]The analysis and the text in this chapter are my own. Jack Skinner and James Cho provided the hydrodynamics simulation data and Rodrigo Luger provided useful feedback on the model.

exoplanets in the near-infrared. The central question of this chapter is whether we can use JWST to infer spatial maps of exoplanets that are more detailed than the ones that have been constructed so far. To answer this question we start by plotting the best-case-scenario inferred maps (assuming noiseless observations) for a set of maps with different kinds of spatial features and for different impact parameters of the planet. In Section 7.3 we also explore the sensitivity of the features in the light curve to variable spot parameters such as contrast, size and latitude. In Section 7.4 we move beyond toy models. I use the outputs of a hydrodynamics simulation of an atmosphere of the Hot Jupiter HD209458b to simulate semi-realistic photometric JWST light curves. I then fit those light curves to test what is recoverable from the data under ideal conditions. Finally, in Section 3.10 we summarise the results and discuss the implications for observing strategies.

I use several simplifying assumptions about the problem at hand. First, we assume that the simulated planets are tidally locked and we ignore uncertainties in the orbital parameters. Second, we assume that the emission from the planet is isotropic blackbody emission, and the reflected light component is negligible. Third, we ignore the possibility that the light curves have correlated noise – we assume that the noise in photometric light curves is purely Gaussian white noise with known error bars. For these reasons the results presented in this chapter should be seen as an upper bound on what is achievable with realistic datasets.

7.2 The nullspace

To begin with, we ignore the fact that secondary eclipse light curves are extremely noisy and we ask the following question. What are the best-case-scenario maps that could be inferred given noiseless data? The inverse problem – reconstructing a two-dimensional map from a one-dimensional light curve is generally *ill-posed* because the mapping is not unique. Luger et al. (2021c) showed that in the case of purely rotational thermal light curves, the *vast majority* of the spherical harmonic coefficients lie in the nullspace of the linear operator $\mathbf{A}$, meaning that one can construct surface maps with widely different features that produce the same exact light curve. The situation is less dire if we are also able to observe a secondary eclipse because the stellar limb sweeping over the visible disc of the planet during ingress and egress breaks many of the degeneracies. This is because different features on the planet are visible at different times during the eclipse.

In the context of occultations of Io (Chapter 6) we explored the information content of different kinds of observations (see Section 6.4.1) such as phase curves and occultation light curves in emitted and reflected light and we briefly discussed the various degeneracies involved in solving the inverse problem. Here we improve on that analysis by visualising the nullspace in the context of exoplanet secondary eclipses. I start by defining linear operators $\mathbf{P}$ and $\mathbf{N}$ (see Appendix A in Luger et al., 2021c) which decompose the original map into a surface map that is in the *preimage* of the linear mapping $\mathbf{A}$,

$$\mathbf{y}_\bullet = \mathbf{P}\mathbf{y} \quad , \tag{7.1}$$

and a map component which is in the *nullspace* of $\mathbf{A}$,

$$\mathbf{y}_\circ = \mathbf{N}\mathbf{y} \quad . \tag{7.2}$$

$\mathbf{y}_\bullet$ is the component of the original map constructed from only those spherical harmonic modes which contribute to the light curve. $\mathbf{y}_\circ$ is the opposite – a map constructed from spherical harmonic modes which all lie in the nullspace. In practice, to obtain a robust estimate of the preimage vector $\mathbf{y}_\bullet$, instead of computing $\mathbf{P}$, which requires a numerically stable estimate of the rank of $\mathbf{A}$, we can solve the regularized least squares problem

$$\hat{\mathbf{y}}_\lambda \equiv (\mathbf{A}^\intercal \mathbf{A} + \lambda \mathbf{I})^{-1} \mathbf{A}^\intercal \mathbf{f} \quad , \tag{7.3}$$

which in the limit of small λ tends to the preimage vector $\mathbf{y}_\bullet$ (see, for instance, Hogg and Villar, 2021b)

$$\lim_{\lambda \to 0_+} \hat{\mathbf{y}}_\lambda = \mathbf{y}_\bullet \quad . \tag{7.4}$$

In other words, the preimage map is simply the solution to the inverse problem when we don't have any prior information about the surface.

In Figure 7.1 we show the preimage maps $\mathbf{y}_\bullet$ for a set of maps with different kinds of features: a single spot, an elongated spot, two spots at equal longitude, two spots at the equator, a map with azimuthally symmetric bands, and a map of Earth. To compute $\mathbf{y}_\bullet$ we solve the regularized least squares problem (Equation (7.3)) for a Jupiter size planet in a 1-day orbit around a $1R_\odot$ star with a rotation period synchronous with the orbital period. I used the entire light curve except for the transit. Each row of the figure shows the preimage maps for different impact parameters b. When $b = 0$, the stellar limb sweeps over the projected disk of the planet in a direction perpendicular to the planet's equator so that only longitudinal variations in map intensity can be measured. For a nonzero impact parameter, the stellar limb sweeps over the disc at an angle, which, combined with the fact that we see a slightly different rotational phase of the planet at ingress and egress, results in some sensitivity to latitudinal variations in intensity.

A few things are apparent from Figure 7.1. For the simulated map consisting of a single spot (first column from the left), the preimage map at nonzero impact parameter consists of two faint arcs intersecting at the true location of the spot. The arcs match the shape of the projected stellar limb at the point at which the limb crossed the spot during ingress and egress. Because the total observed flux is sensitive only to the variation in intensity perpendicular to the limb and not along the limb, we see this cross-like pattern instead of a single spot. In absence of either the ingress or the ingress part of the light curve, the inferred map would consists only of a single arc. The preimage maps for an ellipsoidal spot (second column from the left) noticeably differ from those for a circular spot. The inferred pattern is diamond-like instead of ellipsoidal so the information about the original shape of the feature is lost.

The preimage maps for simulated maps with two spots at equal longitude (third column from the left) and two spots at the equator (fourth column from the left) consist of multiple cross-like patterns. It is not obvious by looking at the preimage maps if there is one, two or four spots in the map. The second column from the right shows an azimuthally symmetric simulated map with a banded structure resembling Jupiter in the Solar System. The preimage maps for this map all appear uniform and the information about the banded structure in the original map is lost. This happens because the angle of the projected stellar limb is never parallel to the equator, so that latitudinal variations in intensity get imprinted onto the light curve. The rightmost column shows a map with a complex structure (Earth centred

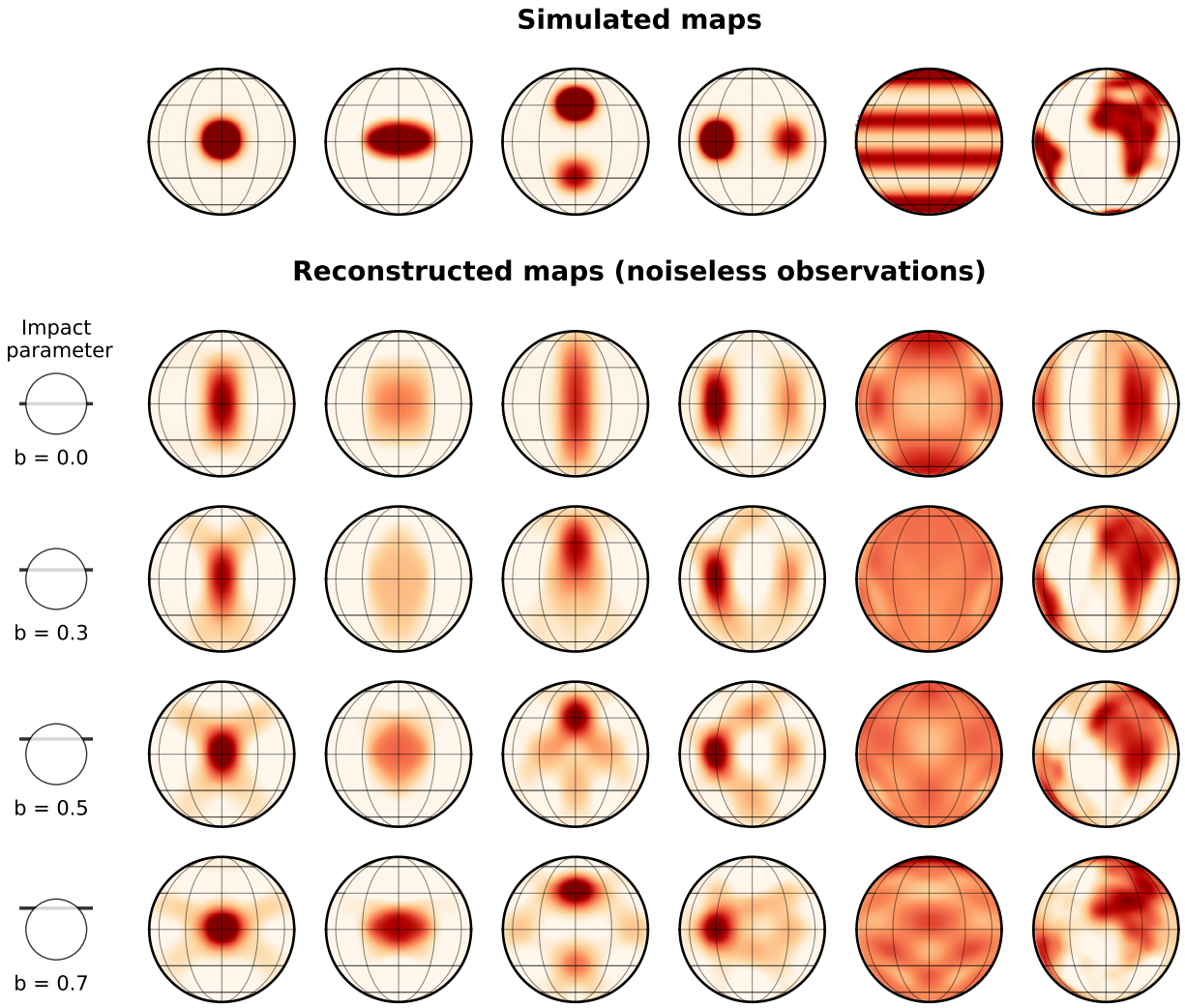


Figure 7.1: The top row shows a collection of simulated spherical harmonic maps with different spatial features and in the bottom row are the corresponding *preimage* maps – maps constructed only from those spherical harmonic coefficients which are not in the nullspace of the linear operator which maps the 2D map into a 1D light curve. The preimage maps represent the best-case scenario for reconstructing the original maps, they are equivalent to the solution of the linear inverse problem when the signal-to-noise for of the light curve tends to infinity. The maps were computed for a tidally locked Jupiter size planet in a 1-day orbit around a $1R_{\odot}$ star. Each row below the top row corresponds to a different value of the impact parameter of for the secondary eclipse.

on the prime meridian); despite all the degeneracies in other cases, the preimage maps end up looking somewhat similar to the original map.

An interesting question is: what is the geometry of the eclipses which minimises the degeneracies visible in Figure 7.1. In Figure 7.2 we show the preimage maps for a planet with 45° projected obliquity, and an impact parameter set such that the stellar limb sweeps over the planet's disc in the direction of the equator during ingress (egress), and in the direction perpendicular to the equator during egress (ingress). In that case the information about the (dayside hemisphere) of the original map which gets imprinted onto the light curve is maximised, and the preimage maps appear more similar to the original maps. The preimage maps of the maps consisting of the two spots then show two clear spots, and the

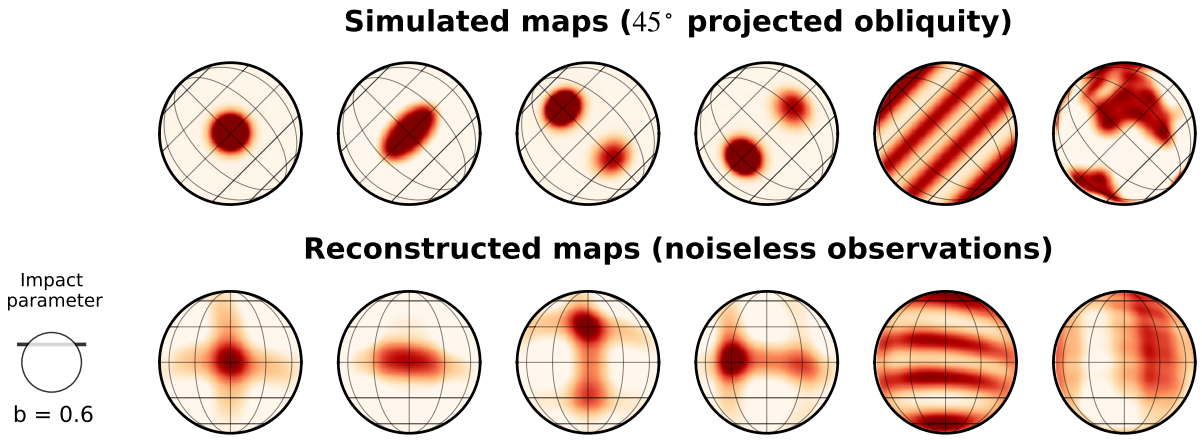


Figure 7.2: Similar to Figure 7.1, except the preimage maps are computed for a planet with 45° projected obliquity and an impact parameter set such that the stellar limb sweeps over the planet's disc in the direction of the equator during ingress (egress), and in the direction perpendicular to the equator during egress (ingress).

banded structure in the azimuthally symmetric map (second column from the right) is fully recovered. This kind of geometry only occurs if the planet's spin axis is highly inclined with respect to the orbital plane or if the planet is on an elliptic orbit and the longitude of ascending node is such that the projected obliquity at the eclipse is significant.

In summary, given an arbitrary map $\mathbf{y}$ and knowledge of the planet's size relative to the star, and its orbit, Equation (7.2) gives us recipe for constructing the preimage maps – best possible solutions to the inverse problem (inferring a map from a light curve) in the limit of perfect data. These preimage maps often differ substantially from the true maps because some information ends up in the nullspace of the linear operator $\mathbf{A}$. The extent to which the preimage map differs from the true map depends on the geometry of the eclipses; a hypothetical ideal candidate planet for eclipse mapping has both the projected obliquity at the time of the eclipse and the impact parameter tuned such that the stellar limb sweeps over the planet's equator, and in the direction perpendicular to the equator during the ingress and egress of the eclipse.

7.3 Signal and noise

The only way the information about a small-scale (large l) surface feature gets imprinted onto the light curve is if the change in total flux due to the feature coming in or out of view during ingress or egress is comparable in scale to the noise. This requires features with high-intensity *contrast* relative to the background. The reason why we were able to infer high-resolution maps from the occultation light curves of Io in Chapter 6 is that these light curves show a clear signature of the surface hot spots (the spots have high contrast) and the data are of exceptional quality (because Io is close and most of the flux is coming from Io itself rather than Jupiter). Observations of secondary eclipses (occultations of planets by stars) are substantially noisier because the planet-star flux ratio for an exoplanet system is extremely small, but the same principle holds.

To demonstrate this dependence of the inferred maps on feature contrast in the context

of exoplanets, we simulate a set of three maps at $l = 20$ consisting of a single hot spot with a radius of 30°, located at 20° latitude and 15° longitude. I set the planet/star radius to $R_p/R_\star = 0.1$ and assume that it is orbiting a Solar mass star in a circular orbit with a 1 day period. I set the dayside planet star flux ratio $F_p/F_\star$ to 10^{-3}, and vary the contrast of the spot c. The contrast c is defined as the fractional increase in the intensity of the map at the spot's location when the spot is added while holding the planet-star flux ratio constant. I then generate light curves with $5s$ cadence and set the noise such that the signal-to-noise ratio for the secondary eclipse depth (defined as the difference between the maximum and minimum value of flux) is 15.

The model I fit to the data is nearly identical to the one described in Section 6.3 in the context of occultations of Io (Equation 6.9). The only difference is that the design matrix $\mathbf{A}'$, which encodes the geometry of the eclipses, is computed for a system with different geometry. The equation for the predicted flux is

$$\mathbf{f} = \mathbf{A}'\mathbf{p} + f_\star \mathbf{1} \quad , \tag{7.5}$$

where $f_\star$ is the (assumed to be constant) stellar flux. As in the case of Io, the design matrix $\mathbf{A}'$ is precomputed with starry. This is possible because of the assumption that the orbital parameters of the system are known and fixed. The posterior distribution is specified by Equation 7.5, and a Gaussian likelihood function with a diagonal covariance matrix.

I sample the posterior using the numpyro implementation of the NUTS MCMC sampler with 500 tuning steps and 500 production steps, checking for convergence using the diagnostics described in previous chapters. The results are shown in Figure 7.3. The top row shows the simulated maps, the second row shows the mean inferred maps together with a few samples from the posterior (miniature maps), the second row from the bottom shows the eclipse portion of the complete light curves and the fitted model (orange). The bottom row shows the difference between the simulated data and the flux generated using the simulated maps but truncated to an $l = 1$ (dipole) order. This is to demonstrate how increasing the contrast of the spot increases the deviation in the light curve from the simplest baseline model, a dipole map. The black lines in the bottom row denote the mean residual flux binned in 5-minute bins.

Looking at the binned flux it is clear that the deviation from a baseline model map, with near uniform dayside intensity, increases with increasing spot contrast. Conversely, features which are too dim are drowned out by the noise. Figure 7.4 shows the dependence of this deviation on the spot contrast, radius and latitude. I simulate a map with a spot at 0° longitude and compute the residuals between the simulated flux at $l = 20$ and $l = 1$, showing only the egress part of the flux (for a spot at 0° longitude the ingress and egress are symmetric). The first panel from the left shows the residual flux for a spot with 30° radius at the equator and varying contrast. In the middle panel, we fix the contrast to $c = 0.15$ and vary the radius of the spot. Finally, in the rightmost panel, we show the residual flux for a spot with 30° radius and $c = -0.15$ for different latitudes. We see that the flux deviation is maximised for a large, high-contrast spot with large absolute values of latitude.

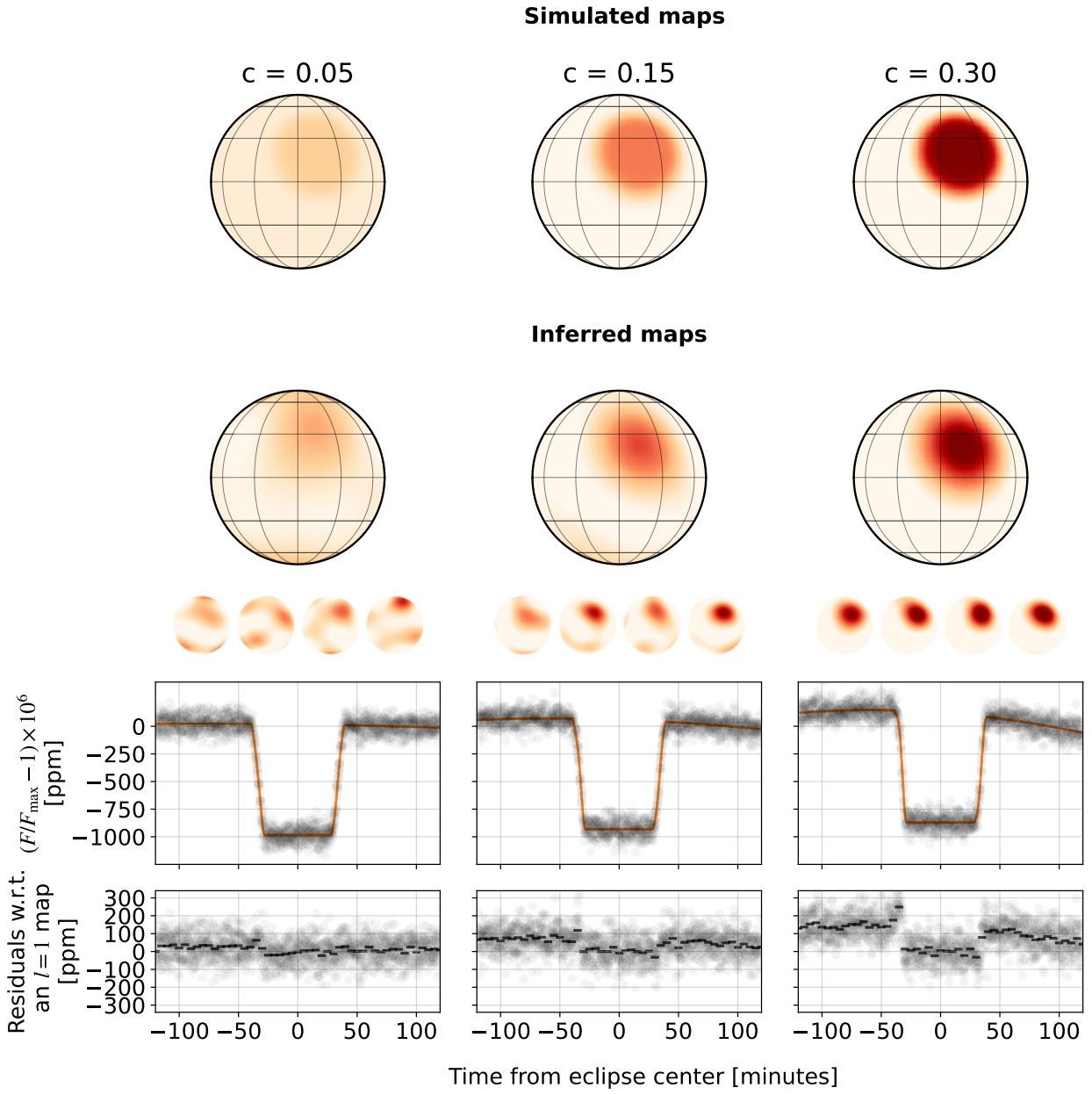


Figure 7.3: Dependence of the inferred maps on the contrast c between the feature and the background. The top row shows simulated maps at $l = 20$ with a hot spot offset from the substellar point, located at 20° latitude and 15° longitude with a radius of 30°. Each column shows a map with a different spot contrast c but all maps have the same dayside flux ratio relative to the star, set to 10^{-3}. The second row shows the mean inferred maps together with a few sample maps from the posterior (small circles). The grey cross marks the centre of the simulated spot. The bottom two rows show the simulated secondary eclipse light curves with the fitted flux (solid orange line) and the residuals between the data and the simulated flux for maps shown in the top row, except truncated to $l = 1$ to emphasize that the scale of this difference relative to the noise level determines the quality of the inferred maps. The solid lines show the binned residuals in 5-minute bins.

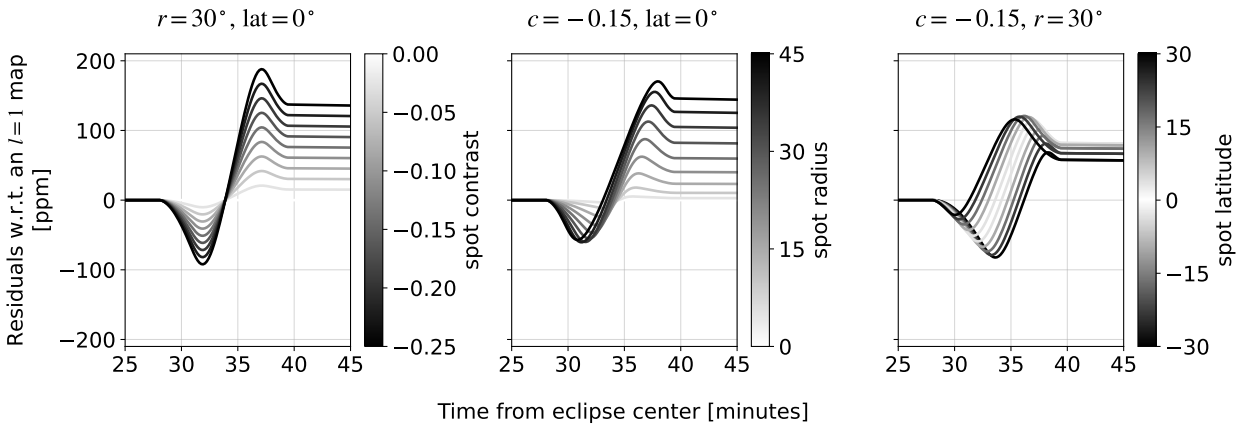


Figure 7.4: Residuals between flux during egress computed with simulated maps at $l = 20$ consisting of a single spot at 0° longitude with varying size, contrast and latitude (while holding the planet to star dayside flux ratio constant at 0.001), and the flux computed with the same maps truncated to $l = 1$. Since the spot is always at 0° longitude, the ingress and egress flux is the same so we only show the egress. The deviation from the baseline model ($l = 1$ maps) is maximised for large spots with large contrasts at appreciable latitudes.

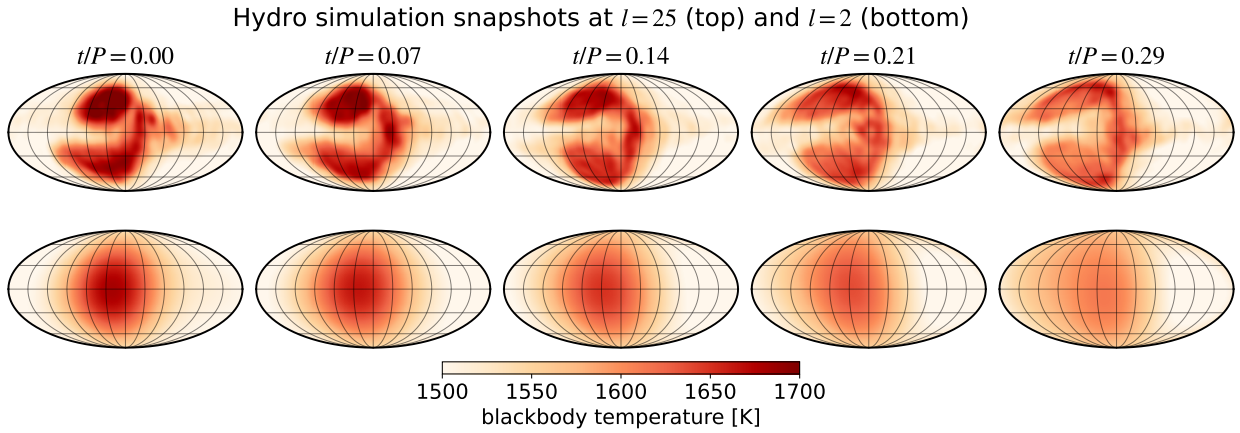


Figure 7.5: The output of a 3D dynamical simulation of a Hot Jupiter atmosphere showing the time variability of the temperature distribution at a pressure of 1 bar within a single orbit. The top row shows the temperature maps at $l = 20$ in the Mollweide projection and while the bottom row shows the same maps truncated to $l = 2$. The snapshots show that the evolution of the spatial distribution of emission from the atmosphere changes rapidly because of the presence of modons (pairs of planetary scale storms).

7.4 Going beyond the spot model

I showed that the ability to infer localise features in exoplanet emission maps depends on the noise in the light curve and the contrast of the localised features on the observer-facing side of the planet during the eclipses (the dayside in the case of tidally locked exoplanets). In this section, we move beyond toy models with artificial circular spots and ask a more specific question. Can we use eclipse mapping to learn something about the climate or weather variations on the dayside of Hot Jupiters using JWST eclipse light curves?

To answer this question we make use of high-resolution 3D dynamical simulations of the atmospheres of tidally locked Hot Jupiters. These simulations were previously described in Skinner and Cho (2022) and Cho et al. (2021). Whereas lower resolution simulations of Hot Jupiter atmospheres predicted a stationary hot spot, shifted eastward from the substellar point, these higher resolution simulations are able to capture small-scale dynamics which drive planetary-scale variations in pressure and temperature. The result is that the simulations show a much more dynamic picture of what these atmospheres are like. Instead of a single hot spot, these simulations show dynamic modon structures – coherent flow structures made up of a pair of storms with opposite spin. The modons have quasi-periodic behaviour over orbital timescales. They transport patches of hot and cold air around the planet which influences the temperature distribution and hence also the near-IR emission from the planet.

Of course, the purely dynamical simulations of the atmosphere are not sufficient for computing realistic emission maps of the planet in a given photometric filter. For that, we would also need a radiative transfer model of the atmosphere and perhaps a prescription for the physics of cloud formation and photochemistry. Nevertheless, there is a lot we can learn by simply taking the simulated temperature maps at a specific pressure level and simulating eclipse light curves assuming a blackbody emission spectrum.

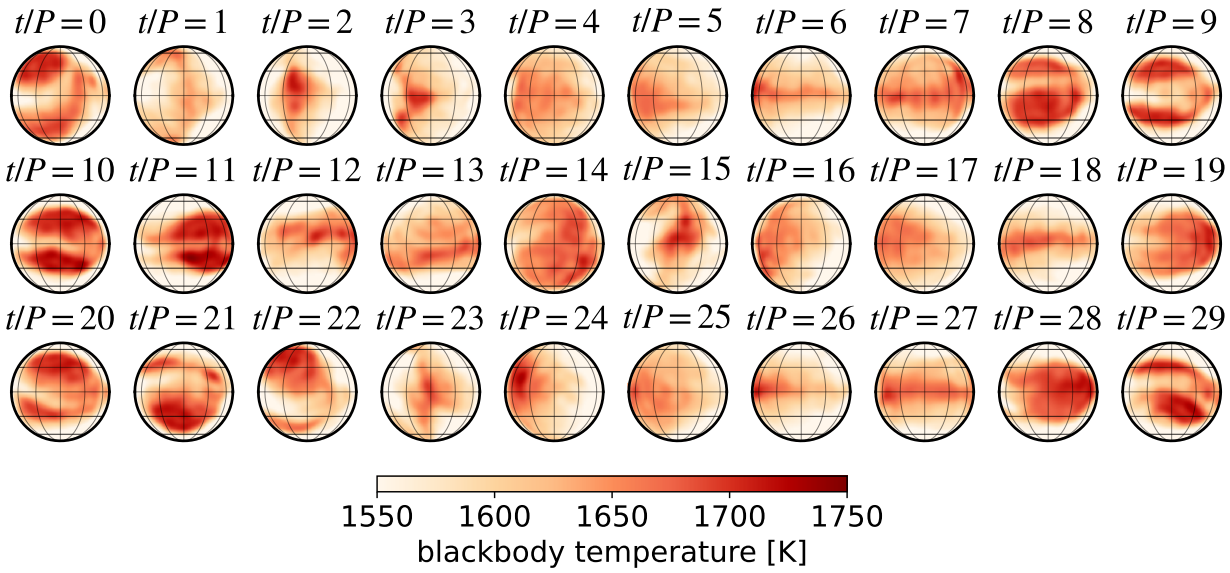


Figure 7.6: The output of a 3D dynamical simulation of a Hot Jupiter atmosphere showing the time variability of the temperature distribution at a pressure of 1 bar between multiple orbits at the same phase. Each circle shows the dayside temperature distribution for the simulated planet expended to an $l = 20$ spherical harmonic map.

7.4.1 Variability on orbital timescales

I start by taking the "T341L20" simulation snapshots from Cho et al. (2021) where T341 refers to the resolution of the simulation in latitude and longitude, and L20 refers to the vertical resolution (20 pressure levels in this case). For details on the meaning of these values see Skinner and Cho (2021). The orbital parameters for these simulations are set to the orbital parameters of the Hot Jupiter HD209458b (Henry et al., 2000; Charbonneau et al., 2000). I use the temperature snapshots at a pressure of 1 bar because this pressure level very roughly corresponds to peak emission in the near-IR. The first topic we explore is the variation in the spatial distribution of temperature on an orbital timescale. Figure 7.5 shows a series of consecutive snapshots for a duration of a single orbital period. The top row shows the blackbody temperature maps expanded in a spherical harmonic basis at $l = 20$, in the Mollweide projection, and the bottom row shows the same maps truncated to $l = 2$ for a coarse-grained view of the maps. The first snapshot shows one modon pair which dissipates over the course of a day. For reference, we also show the longer timescale variability in Figure 7.6. This figure shows snapshots of the dayside temperature over a period of 30 days.

It is clear from these snapshots that the spatial variation in temperature is very large already on an orbital timescale. *The consequence of these results is that we cannot assume that the map is static over the course of one orbital period.* This assumption has been implicit in pretty much all previous efforts to model the eclipse light curves of exoplanets. It is common to also stack data over multiple periods to increase signal-to-noise. Stacking the data means that we are not inferring an instantaneous map of emission, but rather a map that represents the time-averaged emission over one or multiple orbits. Of course, it is possible that the simulations from Cho et al. (2021) are for some reason not an accurate description of reality, and that real atmospheres are less dynamic. However, the very fact that we are unsure about this question is a good reason to be cautious and not assume that

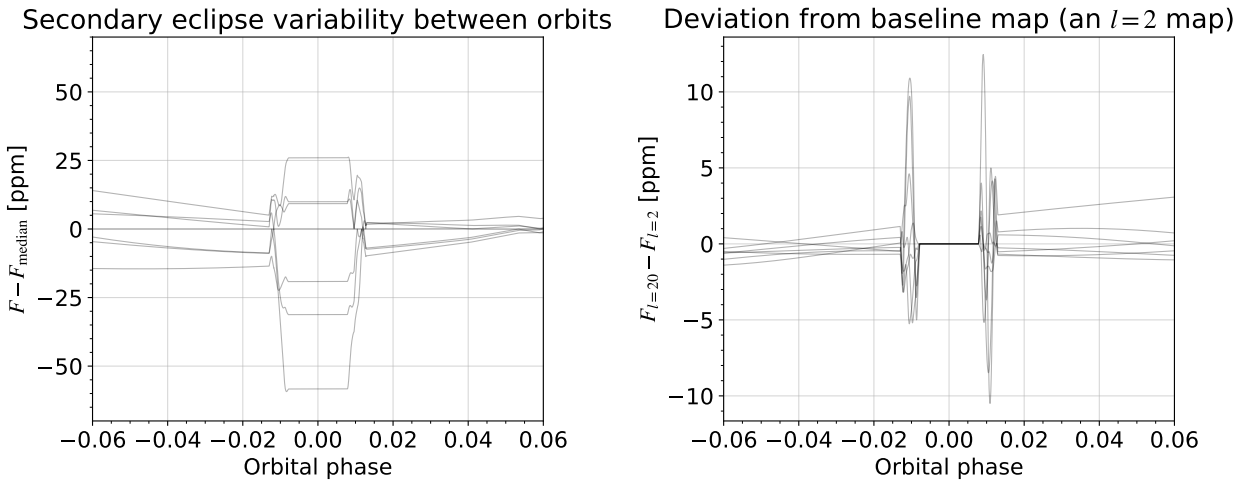


Figure 7.7: Predicted fluxes for simulated maps of HD209458b shown in 7.5 assuming photometric observations in the F444W $4.5\mu m$ JWST NIRCam filter. The top row shows the fluxes for each of the snapshots in addition to the mean flux across epochs. The bottom row shows the difference in the predicted fluxes for the snapshots at $l = 20$ (top row of 7.5) and the predicted fluxes for the same snapshots truncated to $l = 2$ (bottom row of 7.5). The difference in flux for different simulation snapshots can be as 200 ppm (top row) but constraining maps at a resolution higher than $l = 2$ requires data with noise at the level of 10 ppm (bottom row).

the map is static over an extended period of time. In the rest of the chapter, we assume that the map is static on timescales equal to $\sim 10\%$ of the full period. This is half the separation between the snapshots shown in Figure 7.5. This assumption could be wrong if there are planetary-scale changes in atmospheric dynamics on such short timescales.

7.4.2 Inferring maps from simulated JWST light curves

Having established that the predicted emission from a Hot Jupiter atmosphere is highly dynamic on sub-orbital and orbital timescales, we now turn to the question of how this variability reflects in the shape of the corresponding eclipse light curves. I select 30 consecutive temperature snapshots taken at the same orbital phase, 30 orbits in total, and compute the predicted flux in the JWST F444W $4.5\mu m$ filter for each snapshot in the time range $(-0.06, 0.06)$ centred at mid-eclipse. The results are shown in Figure 7.7. The left panel shows the difference in predicted flux with respect to the median flux over the 30 snapshots. This is to illustrate the variability in the eclipse depth caused by the movement of the modons around the planet. In the panel on the right, we show the difference in the predicted flux generated from an $l = 20$ map with respect to the $l = 2$ map for each snapshot. We can think of this as the deviation from a baseline model (a quadrupole $l = 2$ map).

Judging from the results shown in Figure 7.7, the variability in the eclipse depth is relatively large and it should be straightforward to measure. Even a simple tophat signal model would be sufficient for that purpose. Detecting higher-order structure in dayside emission, above the level of a quadrupole map, is much harder because we need to be able to measure the shape of signals with a scale of a few ppm over a relatively short time period.

Next, we choose 7 consecutive temperature snapshots and generate a simulated JWST light curve for each snapshot. To estimate the uncertainties for the data points, we use the

online version of the PandExo tool (Batalha et al., 2017)[2]. In the tool, we select HD209458b as the target and the JWST F444W filter. I use the default parameters for the star and the planet suggested by PandExo and we set the "Number of groups per integration" to 15. The output is an exposure time of $5.5s$ and a signal-to-noise ratio for the eclipse depth of ~ 15. This SNR roughly corresponds to light curve error bars of 100 ppm. I also generate a separate set of light curves with SNR=50 to test how our results would change if we had much better quality data (better than JWST).

I fit the model specified in Equation 7.5 for different values of l (in the range $1, \ldots, 7$) to the simulated light curves using the NUTS sampler. I use 500 warmup iterations and 1500 sampling iterations. All fits converge easily in about a minute of CPU time and the effective sample size for each parameter ends up being well over a thousand. The result for the first light curve, an $l = 4$ map, is shown in Figure 7.8 . The top row shows the results for the SNR=15 light curves and the bottom row shows the results for the SNR=50 light curves. The simulated maps are shown in the leftmost column, the inferred posterior mean maps are in the second column, followed by the posterior samples which illustrate the uncertainty in the inferred maps (third column). In the last column we plot the median predicted flux and the simulated light curves.

The posterior mean map in both cases does not look like the simulated map, and the variance in the posterior sample maps is quite high. The variance is so high that one could justifiably ask if there is anything useful we could learn by looking at these maps. This is partly a consequence of the fact that a lot of information is lost in the null space of the design matrix, and partly because the data are very noisy. We also see that the inferred maps have higher contrast than the simulated maps (especially in the SNR=15 case). This is a consequence of the fact that many of the higher l coefficients are weakly constrained by the data and thus, the inferred maps are strongly influenced by the prior. Enforcing the positive intensity of the inferred maps by fitting in pixel space surpresses these spotty high-contrast maps only to some extent. Making the maps look less contrasty would require a more sophisticated prior.

The interpretation of the results for the first light curve is not qualitatively different than the results for the other 6 light curves. In Figure 7.8 we show the posterior mean maps at $l = 4$ for all 7 light curves (bottom two rows). The simulated maps and the preimages of the simulated maps are shown in the top two rows. Clearly, the inferred maps do not resemble the preimages at all. Also, all inferred maps have a bias towards cooler temperatures in the polar regions.

To gain further insight into how well the different inferred maps fit the data, in Figure 7.10 we plot the posterior flux residuals with respect to the fluxes generated from the simulated maps, but truncated to $l = 2$. The idea is to visualise the strength of the evidence for higher-order structure in the maps. The flux residuals are shown for the SNR=15 light curves in the top row, and for the SNR=50 light curves in the bottom row. The translucent blue lines are posterior samples and the black line is the true value. For SNR=15 the magnitude of the residuals is too large compared to the true values for all light curves, indicating that the model overfits the data with $l > 2$ harmonics. This is why we see such high contrast spots in the maps shown in Figure 7.10. It's only when we increase the SNR to around 50 that these

[2] https://exoctk.stsci.edu/pandexo/calculation/new

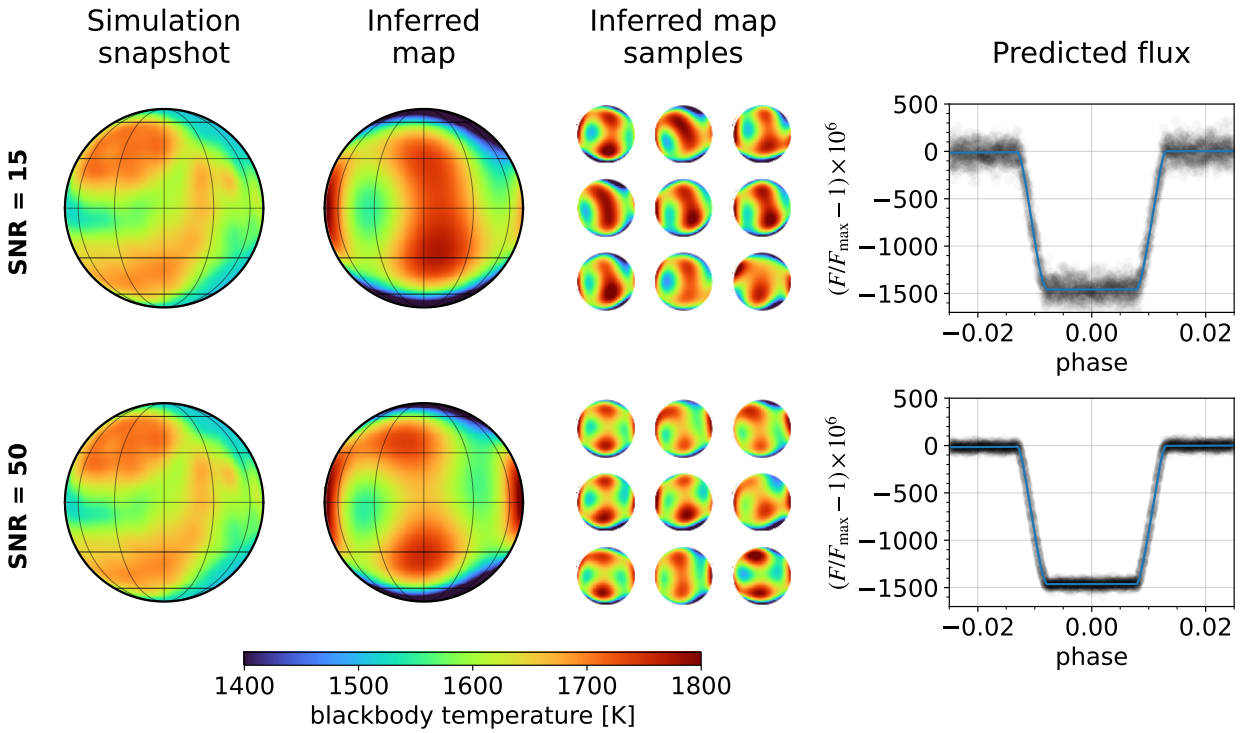


Figure 7.8: Recovered maps from simulated eclipse light curves for the planet HD209458b generated from the $t = 0$ temperature snapshot shown in Figure 7.6. The simulated light curves were generated assuming observations with the JWST F444W 4.5μm NIRCam filter and a 5.5s exposure time. The noise variance was set using the PandExo tool which gives SNR=15 for the eclipse depth. I also show results for SNR=50 for reference (bottom row). The first column shows the simulated maps, the second column shows the mean inferred maps, the third column shows the posterior sample maps and finally, the fourth column shows the median posterior flux (blue line) and the simulated light curves.

residuals start being resolved (see the first three panels in the bottom row), although only for those snapshots for which the dayside emission map has heterogeneous features with a large contrast.

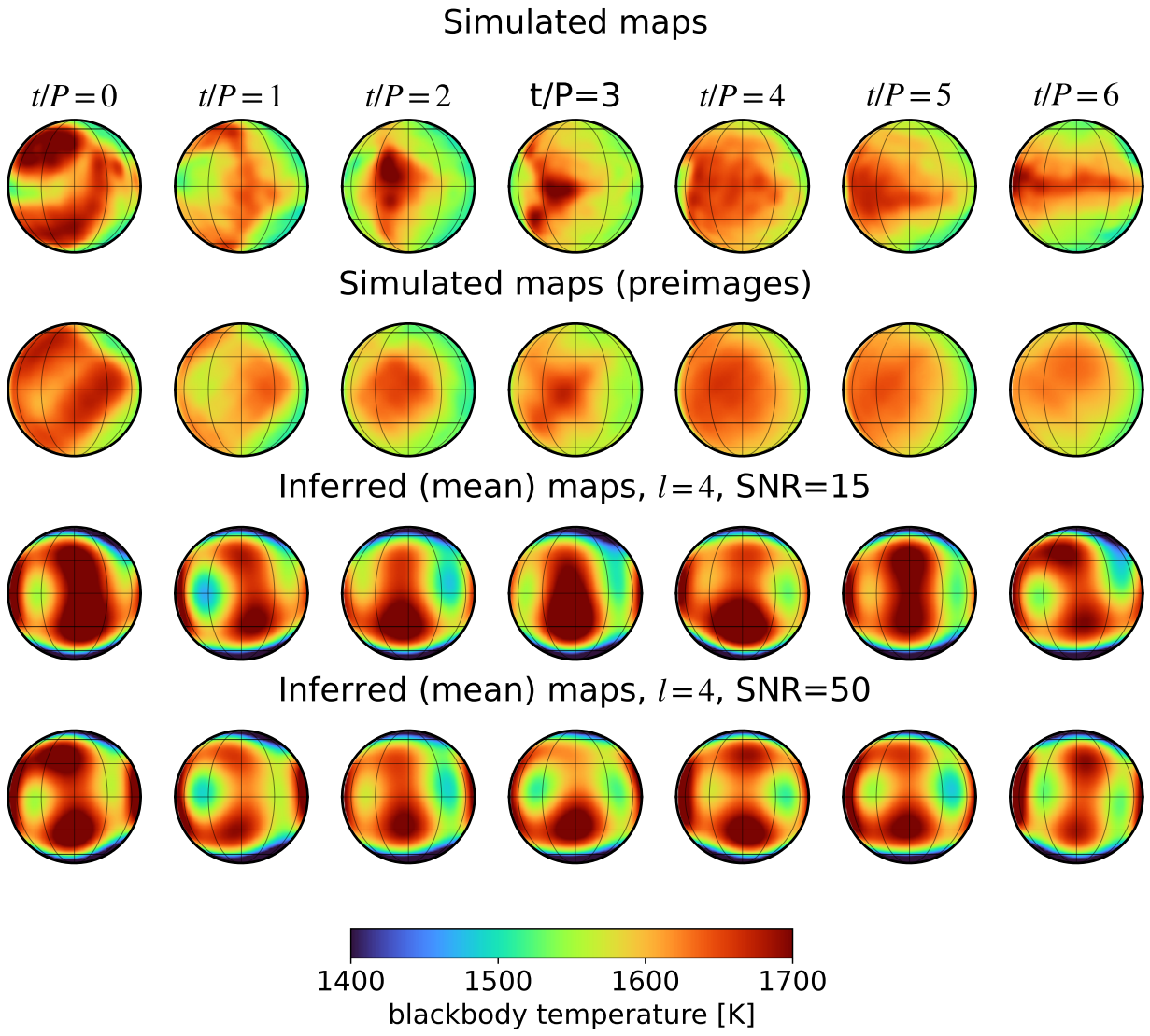


Figure 7.9: A sequence of simulated maps (top row), the preimages of the simulated maps (second row) and the inferred (mean) maps for the planet HD209458b (bottom two rows).

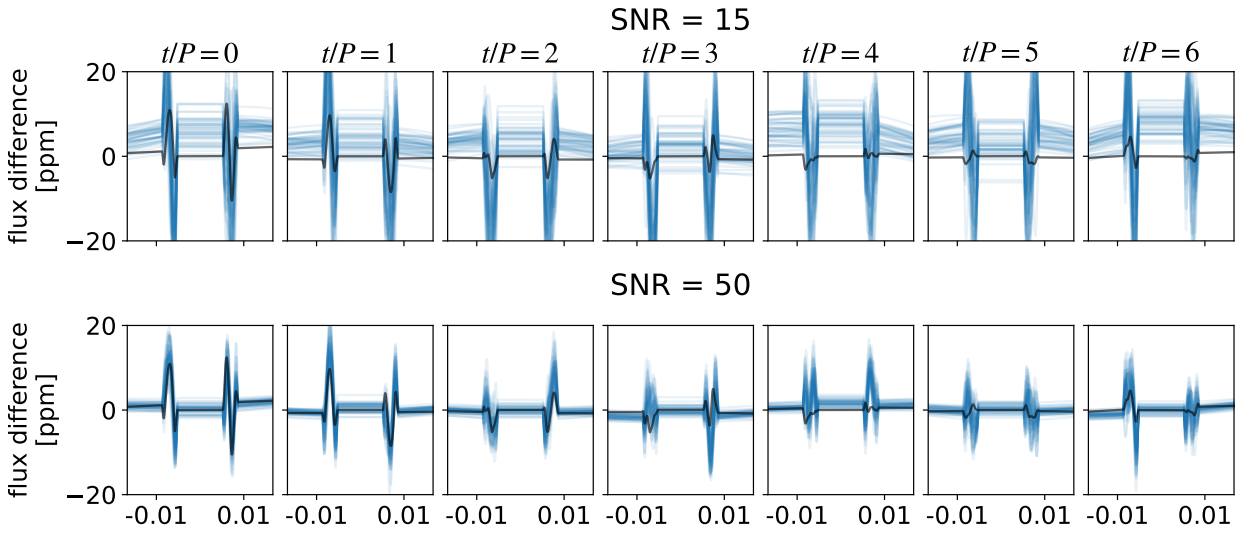


Figure 7.10: Posterior residuals of the fitted flux with respect to the true flux generated from $l = 2$ maps. Each translucent blue line is one posterior sample. The black line is the true value. The top row shows the results for the SNR=15 light curves and the bottom row shows the results for the SNR=50 light curves. The quality of the SNR=15 light curves is not sufficient to recover the higher-order l modes in the data.

7.5 Discussion and summary

In this chapter, we have investigated the feasibility of using eclipse mapping to infer morphologically complex two-dimensional emission maps of Hot Jupiter atmospheres. I started by visualising the nullspace of the linear function which transforms a two-dimensional map of the planet into a one-dimensional light curve (Section 7.2). I demonstrated that there are intrinsic limits to what can be recovered from a single eclipse light curve; the inferred maps do not converge to the true maps even in the limit of noiseless data. I showed that the key factor that determines whether a given spatial feature can be recovered in a map is its contrast relative to the background. High contrast features are easiest to recover because their signature is most prominent in the light curve during the eclipse.

Going beyond toy models, in Section 7.4 we used the output of a state-of-the-art 3D hydrodynamical simulation of the atmosphere of the Hot Jupiter HD209458b to generate semi-realistic simulated eclipse light curves using the `starry` framework. These simulations are interesting because they exhibit a spatially heterogenous and time-variable temperature distribution on the dayside of the planet, caused by the presence of planetary-scale storms. Although the results of these simulations have been published before, the implications they have for the eclipse mapping of Hot Jupiters have not been fully recognized. The key implication is that the emission maps can be highly variable on sub-orbital timescale. This imposes constraints on the duration of the light curve we can use for eclipse mapping because of the fundamental assumption that the map is static in time. For this reason, we only used the eclipse portion of light curves in subsequent analyses.

I chose a sequence of 7 consecutive snapshots from the hydrodynamics simulation and generated two sets of simulated eclipse light curves of the planet HD209458b using `starry`. The first set of light curves were generated by setting the noise variance to match the expected noise level for observations with the JWST NIRCam F444W filter and a 5.5s exposure time. The noise level for the second set of light curves was set to a much lower value so we could get a sense of how the result would change if he had an instrument superior to JWST. I fit an $l = 4$ map to each light curve in the two sets and find that in both cases the inferred maps are not even remotely close to the simulated maps (more specifically, the preimages of the simulated maps).

The fundamental conclusion from the analysis is that the eclipse mapping problem is highly degenerate, the inferred maps are strongly influenced by the prior, and the variance of the inferred maps is so large that it is difficult to say anything concrete about the smaller scale features in the maps. There may be ways to impose physically meaningful priors on the spherical harmonic coefficients such that we end up with less degenerate solutions but we have not had much luck with experimenting with different priors beyond the positivity constraint on the pixels.

So is there anything we could do with eclipse mapping and JWST that is both scientifically useful and robust? Two ideas come to mind. First of all, the key implication of the work by Skinner and Cho (2022) and Cho et al. (2021) is that the climate of Hot Jupiters can be variable on sub-orbital and orbital timescales because of change in weather and climate. This results in spatially heterogenous emission maps such as the ones shown in Figure 7.5 and Figure 7.6, and light curve variations shown in Figure 7.7. Although recovering realistic maps is a futile exercise, we can measure the changes in eclipse depth from epoch to epoch

and changes in say the latitude and longitude of peak emission by fitting a dipole ($l = 1$) map to the data. This is at least sufficient to reject the hypothesis that the emission from the planet is static although it does not necessarily tell us much about the underlying physical processes. The second thing that should be possible is to reject the lowest order $l = 1$ map as a viable model for the data. We could do this by fitting higher order maps, computing the cross validation scores and showing that higher order maps are much more predictive than the dipole map. This would be a useful test of the hypothesis that the emission from the planet is well described by a map that consists of a single hot spot.

A paper by Kilpatrick et al. (2020) did exactly that with Spitzer observations of Hot Jupiters HD189733b and HD209458b and found no evidence for epoch-to-epoch variability in the eclipse depth of HD189733b greater than 12% and no evidence for epoch-to-epoch variability in the eclipse depth of HD209458b greater than 1.6%. The predicted eclipse depth variations HD209458b are shown in Figure 7.7. They are equal to about 1-2% of the eclipse depth which is below the detection threshold of Spitzer. The uncertainties in the inferred eclipse depth for the fitted light curves from this chapter are much smaller (a few ppm), meaning that JWST should be able to measure the epoch-to-epoch variability with sufficient precision to detect the variability caused weather and climate cycles. Kilpatrick et al. (2020) claim that the fact that they did not detect epoch-to-epoch variability in the eclipse depth of the two Hot Jupiters provides "motivation and justification" for combining multi-epoch observations (stacking multiple eclipse light curves) to improve the signal-to-noise ratio. This is an erroneous conclusion. The absence of evidence does not imply evidence of absence. As we have shown in Figure 7.7, the eclipse depth variations of an HD209458b-like planet are consistent with the results from Kilpatrick et al. (2020) even though these fluxes were generated from highly variable (on a sub-orbital timescale) maps.

Other than monitoring epoch-to-epoch variability in eclipse depth and monitoring changes in low-order modes, the most promising application of eclipse mapping is inferring spectral maps to determine vertical temperature profiles in Hot Jupiter atmospheres. Spectral mapping is outside of the scope of this work but it is not meaningfully different from the problem we have described here. The main points about the degeneracy of the mapping problem and the time variability of the maps still apply.

What about targets other than HD209458b, and telescopes other than JWST? In Figure 7.11 we plot the estimates of the secondary eclipse SNR for a Jupiter size planet orbiting a Sun-like star with $T_{\rm eff} = 5000$K as a function of planet equilibrium temperature and distance from the star. The SNR is estimated assuming that the integration time is 5s and the observation filter is the F444W JWST filter. The left panel shows the estimates for JWST and the right panel shows the estimate for the proposed LUVOIR-A telescope. The latter estimates were obtained by scaling the collecting area of JWST to a value appropriate for LUVOIR-A. The expected SNR for LUVOIR-A eclipse observations of HD209458b is about 50 which is why we chose that value for the noise level in the second set of simulated light curves from this chapter. Thus, not even LUVOIR-A with its 15m diameter mirror will be sufficient to map the atmospheres of Hot Jupiters at finer scales.

Finally, we should mention two topics we have ignored in this chapter. First, we have assumed that we know the orbital parameters exactly. This is of course not the case in reality because these parameters are inferred from the transit and eclipse light curves. It is an open question to what extent the covariances between the orbital parameters and the spherical

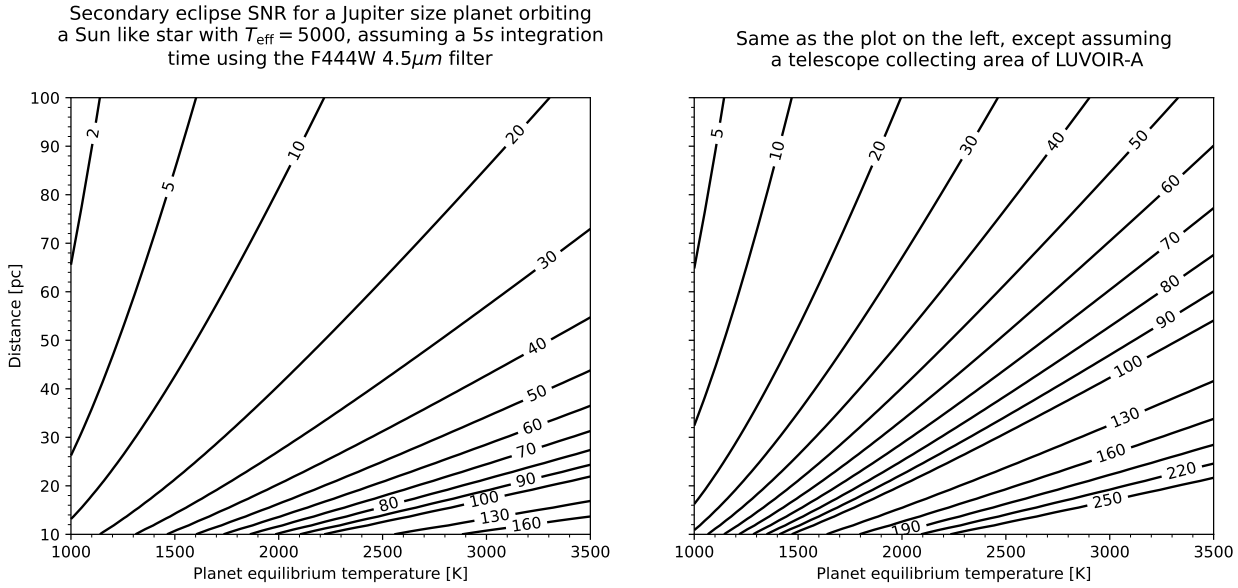


Figure 7.11: Estimates of the signal-to-noise ratio on the secondary eclipse for JWST and LUVOIR-A observations in the F444W 4.5μm filter for a Jupiter size planet orbiting a Sun-like star ($T_{\text{eff}} = 5000$K) as a function of the planet's equilibrium temperature and distance to the star (left panel). The right panel shows the same thing except we scale the collecting area to match the collecting area of the planned LUVOIR-A telescope.

harmonic coefficients affect the results presented here. Fitting for both the orbital parameters and the spherical harmonic coefficients simultaneously is non-trivial because in that case we cannot pre-compute the design matrix $\mathbf{A}'$ since it depends on the orbital parameters. The model is then non-linear in the orbital parameters which also makes inference a lot more challenging.

Second, we only discussed emitted light observations. As mentioned in Chapter 6, there is no null space for reflected light occultations. Seeing how the results from this chapter change for reflected light observations would be interesting. The problem with optical observations of secondary eclipses is that the signal is much weaker than in the near-infrared and in absence of extremely large albedo variations on the dayside, it is unlikely that it is possible to constrain higher-order modes in the inferred albedo map.

Chapter 8

Conclusions

In this final chapter, I briefly summarise the main contributions of the thesis and I discuss ideas for future work.

8.1 Contributions of this thesis

8.1.1 Chapter 3: The caustics code

- I developed a new open-source for computing the extended source magnification of binary and triple-lens events using contour integration, called caustics. The code is entirely written in the framework for automatic differentiation and machine learning framework called JAX.
- This is the first code that enables the computation of exact gradients (both first and second order) of the microlensing magnification with respect to any input parameter.
- caustics is the fastest method for computing the magnification for extended source in triple-lens events[1]. The code is only about 2x slower for computing triple lens magnification than for binary lens magnification. For binary lenses, the performance is by a factor of a few slower than VBBinaryLensing for uniform brightness sources, and on par with VBBinaryLensing for limb-darkened sources.
- caustics requires only a minor modification in order to be applied to astrometric microlensing. This is on the roadmap for future development.

8.1.2 Chapter 4: modelling single lens events

- The general theme of this chapter is: how can we rank different models based on their ability to explain the data in a predictive sense?

[1]To be precise, caustics at the time of writing is fastest for the full contour integration computation of the extended source magnification. As mentioned in Section 3.7, the tests for switching between the full computation and the hexadecapole approximation for triple lenses are not yet implemented. This is a high-priority task for the future.

- Specifically, I have investigated the problem of modelling degenerate single lens microlensing events with annual parallax and explored different strategies for dealing with multi-modal posterior probability density distributions. This problem is ubiquitous in microlensing.

- I used a novel method from the field of computational statistics (Yao et al., 2020) and applied it to a representative OGLE microlensing light curve with annual parallax deviations and a degenerate model. The idea is to fit each mode in the multi-modal pdf independently, and then combine the results based on the predictive performance of each mode. To my knowledge, this is the first time this method has been applied in microlensing and possibly astronomy as a whole.

- The method is far superior to alternatives such as Nested Sampling both in terms of speed and correctness. It is also not in any way specific to the single lens microlensing model. It can be applied to other degenerate microlensing models, though there are other challenges with multiple-lens models specifically (see Chapter 5).

- I have also shown a way to speed up this method using the Laplace approximation instead of MCMC (see Magnusson et al., 2019). This means that the entire inference process for a degenerate single lens model with annual parallax can be accomplished using about 1 minute of CPU time, if the model is relatively well-behaved. The faster method also has built-in diagnostics which tell us when the method is likely to fail and it is necessary to switch to full MCMC. This speedup is important because the single lens model is absolutely fundamental in microlensing and the availability of a fast and reliable model means enables us to apply it to very large datasets, and also for the purpose of finding microlensing events in the first place.

- I have also shown how one can use the per-datapoint cross-validation scores and Pareto shape parameters of the importance weight distribution for the cross-validation posterior for the purpose of checking the model fit. This can be seen as a far more sophisticated alternative to "looking at the residuals" or "goodness of fit" statistics. This methodology has also been almost completely neglected in the literature.

8.1.3 Chapter 5: modelling multiple-lens events

- This chapter contains preliminary and incomplete work.

- Whereas the guiding question in the previous chapter was how to rank different modes in the posterior probability density distribution in the context of single lens events, here, the guiding question is how to search for, and sample from the different modes in the pdfs for degenerate multiple-lens events. This latter problem is straightforward for single lens events *because the likelihood/posterior density is smooth*. It is extremely difficult for multiple-lens events *because the likelihood/posterior density is highly non-smooth with large curvature*.

- I use a single OGLE light curve, with a clear signal of a binary caustic-crossing event, to investigate fundamental problems in the analysis of multiple-lens events: finding and exploring the different modes (solutions) in the posterior pdf.

- By visualising the two-dimensional slices through the likelihood/posterior pdf, I show just how different the problem of modelling caustic-crossing multiple-lens events is from the problem of modelling single lens events. The likelihood/posterior pdf is highly non-smooth and multimodal with complex non-Gaussian structure. The implication is that we should not expect to be able to use the same statistical methods as for single lens events. Because of these properties, *local information about the pdf is not helpful for finding the path to the global regions of interest*. This implies that the work done in Chapter 3 on making the binary and triple-lens models automatically differentiable is perhaps not useful, at least not in the context of caustic-crossing events. Preliminary tests confirm this hypothesis.

- I first review existing methods commonly used in the literature. I claim that the grid-search-based approach to the search problem (finding the different modes) is fundamentally flawed as it is based entirely on heuristics which may or may not work.

- I then explicitly test whether the most commonly used MCMC sampler, emcee, is capable of reliably sampling the posterior when initialised at the mode. I find that the results of MCMC diagnostics tools lead to the conclusion that the answer is no! This result, combined with the observation that almost no paper in the microlensing literature quotes the outcome of even the most basic MCMC diagnostics tools, leads me to the conclusion that we should be highly sceptical of the results of MCMC-based inference in microlensing. I believe that this result generalises to at least all caustic-crossing events if not all multiple-lens-events, although I have not tested this claim thoroughly.

- With the goal of pursuing a more principled approach to the problem, I explore the use of Nested Sampling, a class of methods which promises to solve both the search and the exploration problem for a broad class of (relatively low dimensional) inference problems. Nested Sampling is very popular in cosmology and I have used it in the previous chapter on the analysis of single lens events.

- I test two kinds of flavours of Nested Sampling, a region-based sampler and a step sampler, both are implemented in the code UltraNest Buchner (2021a). There are many subtly different implementations of Nested Sampling but UltraNest is one the most well-tested and reliable ones.

- I find that Nested Sampling with a region-based sampler is effectively useless without major changes to the method for constructing the geometric regions in the prior space (overlapping ellipsoids). The efficiency of the rejection sampler grinds to a halt even after an extremely large number of likelihood evaluations (>200M, requiring thousands of hours of CPU time).

- The alternative step sampler (a slice sampler sampling directly from the likelihood constrained prior) enables the Nested Sampling algorithm to converge according to its internal convergence criteria (fractional threshold on the log-evidence estimate). However, the resulting posterior is not stable with respect to the choice of the initial number of live points (a key hyperparameter in Nested Sampling). By fitting the same model with different numbers of live points (2000, 5000, and 10000) I find that the samples are not consistent with each other and it does not seem that increasing the number of live points would help. Nested Sampling with a step sampler is also extremely computationally expensive, requiring thousands of hours of CPU time for the simplest binary lens model and a single light curve.

- There are other flavours of Nested Sampling worth exploring but the initial experiments do not look very promising. It seems Nested Sampling does not work for this problem, at least not without substantial changes to the method.

- In summary, the results of this chapter raise more questions than they answer.

8.1.4 Chapter 6: Occultation mapping of Io

- The contents of this chapter is mostly the work I published in Bartolić et al. (2022). This work was done in collaboration with other researchers, but I wrote all the text, generated all the figures and wrote all of the code.

- We were interested in applying the recently developed code called `starry` (which was designed primarily for exoplanet eclipse and transit modelling) to the problem of reconstructing the surface emission from the Jovian moon Io using occultation light curves.

- Io has been continuously observed for decades with ground-based, near-infrared observatories (primarily IRTF) during occultations by Jupiter and other moons. It is the most volcanically active body in the Solar System. The photometric observations, most of which were obtained while Io was in Jupiter's shadow, contain information about the two-dimensional map of the surface emission and albedo. In the chapter, we focused on mapping the blackbody surface emission rather than the albedo.

- We fit two sets of complete occultation light curves (ingress + egress) of an Io occultation by Jupiter. Both sets of observations were done with the same instrument (IRTF) in near-infrared. The first set of observations is from the 1990s, the second is from the 2010s. The novel aspect of this work compared to previous efforts in occultation mapping of Io's volcanic activity is that we make fewer assumptions about the structure of the surface map. All previous studies imposed a requirement on the number and the shape of hot spots on Io's surface. We only impose positivity and sparsity as a constraint on the surface map. We also demonstrate how the use of a sophisticated noise model (which includes a Gaussian Process and fitting for all errorbars simultaneously) changes the structure of the inferred map and the number of hot spots.

- We manage to recover the position of the most prominent volcano, Loki (with very high precision) and the lava flow Kanehekili. We find some evidence for the existence of another hot spot.
- The work in this chapter was intended to be the introduction to the more scientifically interesting model for spatio-temporal mapping of Io's volcanic activity using a much larger collection of IRTF light curves which spans a time period of three decades. Unfortunately, I did not manage to finish this. In Section 6.8 I only outline this time-variable model. The key idea is to decompose the emission map at an arbitrary time into a linear composition of K basis maps. We can then fit the spherical harmonic coefficients of those basis maps simultaneously with the time-variable coefficients that determine how the basis maps add together to form the emission map at any given time.
- Ultimately, in the context of this thesis, Chapter 6 is best seen as a stress test for the methods which are subsequently used in Chapter 7. in the context of eclipse (occultation) mapping of exoplanets.

8.1.5 Chapter 7: Spatial mapping of Hot Jupiter atmospheres

- The content of this chapter is as of yet unpublished work I did with several collaborators.
- Our main research question in this work was broadly, "what spatial features can we plausibly infer from the eclipse light curves of Hot Jupiters?". This work was directly motivated by the work presented in the previous chapter on the eclipse (occultation) mapping of Io. The reason we focus on Hot Jupiters as opposed to smaller planets is simply that they are far brighter targets than super-Earths.
- To answer this question, we go beyond toy models and make use of high resolution 3D hydrodynamical simulations of a Hot Jupiter atmosphere. Specifically, we use simulations previously published by Cho et al. (2021) and Skinner and Cho (2022). The key aspect of these simulations relative to some older simulations in the literature is that they show highly variable (spatially and temporally) emission from the planet's dayside which is caused by weather and climate patterns. In contrast, older simulations show a relatively uniform "hot spot" on the dayside with not much variation in time.
- Given temperature snapshots (at fixed pressure) from these simulations, we generate simulated eclipse light curves for photometric observations JWST in the 4.5μm band. We assume that the emission is blackbody emission. That is of course is not true, but it does not matter much for the purposes of this work. We also assume that the orbit of the planet is known exactly.
- Our findings are the following:
  - Even with perfect data there are fundamental limits to what we can infer from eclipse light curves. The inferred maps will never converge to the truth and they can in fact look quite different from the truth.

- The contrast of spatial features on the dayside is a key factor determining whether we can "resolve" them in noisy light curves. For instance, measuring the location of a hot spot on the dayside requires that the temperature of the hot spot relative to its immediate surroundings is very high.
- Hydrodynamics simulations directly imply that spatial emission patterns on the dayside of a Hot Jupiter can be completely different from a single homogenous hot spot because of the presence of planetary scale storms (modons). These modons evolve in time on a sub-orbital timescale which results in very different-looking maps, even within a single orbit. The implication is that modellers should not stack light curves from multiple epochs nor model a complete single epoch light curve with a single static map. The best we can do is to fit just the eclipse portion of the light curve, because it is (hopefully) safe to assume that the map is static on such short timescales.
- The deviations from a quadrupole ($l = 2$) spherical harmonic map model imprinted on the predicted flux during ingress and egress are so small (~ 10 ppm) that they will be barely detectable by JWST even for the brightest targets.
- By fitting simulated light curves and inferring the maps in a Bayesian model, we find that the inferred maps have very high variance and it is difficult to extract any meaningful information from them. This is not surprising in light of the previous point.
- Two scientific hypotheses we will certainly be able to test with JWST (ignoring the spectral domain) are the following:

  1. Epoch-to-epoch variability of the eclipse depth caused by the weather/climate changes.
  2. Rejecting the hypothesis that the dayside is a single hot spot.

  Going beyond testing these two hypotheses will require making sense of the highly degenerate inferred maps which will be extremely difficult even with the proposed JWST successor LUVOIR-A.

8.2 Future work and open questions

8.2.1 Microlensing

In this thesis I have covered key problems related to modelling microlensing events:

- The forward modelling problem: computing the microlensing magnification for single, binary, and triple-lens systems with extended sources.
- The inverse modelling problem: fitting single and multiple-lens models, model comparison and model criticism.

Some important topics I did not discuss are:

- Population inference: how to combine information from individual data sets and make inferences about a *population of objects*.
- Modelling the noise in microlensing light curves.

Although Golovich et al. (2022), to which I contributed, covers both of these topics in the context of microlensing events.

In what follows, I will outline my vision for future microlensing research opportunities.

Multiple-lens models

One of the main conclusions from Chapter 5 is that multiple-lens microlensing models are qualitatively completely different from single lens models because of non-smooth and extremely degenerate likelihood surfaces. As a result, popular statistical methods such as MCMC and Nested Sampling fail to work as intended. We can group the main challenges in three categories:

1. **The search problem:** Find all relevant "modes" (regions of high probability mass) in the posterior pdf.
2. **The sampling problem:** Explore the modes *locally* in order to estimate the uncertainties for each "solution".
3. **The model comparison/averaging problem:** How can we compare both different modes in the context of a single model (specified with a likelihood function and priors) and entirely different models (single lens vs binary lens vs triple-lens vs binary source, etc.)?
4. **Population inference:** How to combine inferences about individual objects into inferences about a population of objects?

In theory, 1) is an unsolvable problem because no algorithm is guaranteed to find all of the modes (see the discussion in Section 2.4.1 on the no-free-lunch theorem). In practice, these models are sufficiently low-dimensional that I believe it should be possible to construct a reliable algorithm. Based on the results from Chapter 5, I believe that all potential solutions will be computationally expensive, requiring 10s if not hundreds of millions of likelihood evaluations per event.

Here are some ideas for potential solutions to the search problem:

- A flavour of Nested Sampling or a similar algorithm (SMC, parallel tempering?).
- A machine learning model trained on a *very large collection* of simulated events with a non-trivial noise model and realistic observing cadence. The paper by Zhao and Zhu (2022) is a promising start.
- An optimisation approach to the problem, making use of gradient-free evolutionary algorithms, perhaps also incorporating a machine learning model for proposing new steps in the parameter space.

Whichever option we choose, it is very important to test the algorithm with *non-trivial* simulated data sets in order to make sure that the algorithm does not miss important solutions. The model used to generate the simulated data should be more complex than the model we are trying to fit.

Assuming we have a reliable algorithm for 1), the next step is to explore the parameter space in the vicinity of each mode (solution). In Chapter 5 I have shown that neither popular MCMC methods nor Nested Sampling do this step reliably. I am not sure what the best approach is here. Here are some ideas:

- We could first make the likelihood smoother by blowing up the noise or raising the likelihood to some power (this is similar to parallel tempering methods), or increasing the source star radius and then use NUTS or a similar sampler.
- Use machine learning to train a normalising flow which transforms the target distribution into a nicer form (more Gaussian-like), see Hoffman et al. (2019). I have tried this approach on the single lens model with annual parallax and I found that the method is very difficult to tune.
- Perhaps there is a way to use the multiple restart optimisation + Laplace approximation strategies introduced in Chapter 4 to obtain a *good enough* estimate of the local covariances between the parameters.

I believe that the answer to problem 3) is cross-validation combined with some systemized approach to incorporating external information (priors on properties of planets, stars etc.). The former part is difficult because it most likely requires solving problem 2) (we need posterior samples to compute LOO-CV scores for instance). The latter part is really important because the parameters inferred by fitting microlensing light curves is not the *only information* we have about the different models. We also often have some constraints on the source and the lens stars and a priori probabilities for different solutions. This information is used in some way or another in almost every microlensing paper but there is no systematic way of incorporating it into the model comparison process. To be more precise, one way of accomplishing this would be to set priors for each discrete model $\mathcal{M}$ in a Bayesian model comparison framework such that these priors are dependent (in some precise way) on external databases containing other observed properties of the event, or outputs of galactic simulations.

Finally, problem 4) is what enables us to answer scientific questions about *populations of objects* (arguably the most important questions). For instance, properties of free-floating planets and compact object populations, mass function for exoplanets etc. To be able to do this kind of work we need to solve problems 1)-3). Otherwise, we are combining biased and incomplete parameter estimates. Two wrongs do not make a right.

Each of these problems in the context of multiple-lens events is hard and it will require both an overarching theoretical framework (what do we value when comparing models, how to incorporate prior information and constraints, how to model selection effects etc.) and also a complex (and open source!) computational infrastructure layer for everything from solving the forward model, to doing model comparison and population level (hierarchical) inference. Current approaches which rely on human intervention and heuristics may occasionally work

for analyzing individual events, but they are not scalable to the point where we can reliably answer questions about populations of objects.

Other models

Most of the problems outlined in the previous section are also relevant for single lens photometric and astrometric models but the key difference is that they are relatively straightforward to solve. This is because in absence of binary/triple-lens caustics, the likelihood is much better behaved, meaning that we can use reliable methods such as the NUTS sampler, fast LOO-CV estimates and so on. I propose the following strategy for modeling these kinds of events:

1. **Forward model**
    - If finite source effects are important, use the fast and accurate direct integration method from Lee et al. (2009) and Lee et al. (2010) for computing the magnification of photometric and astrometric events. These methods are not yet implemented in caustics but the implementation is trivial.
    - Make sure that the forward model is implemented in an autodiff framework so that we can compute exact gradients and Hessians of the output.
2. **Model specification**
    - For the noise model, use a Gaussian likelihood function with a covariance matrix modelled with a Gaussian Process, and perhaps also a rescaling factor for the error bars that is dependant on the *predicted* flux. The likelihood should be analytically marginalised over the linear flux parameters (see Equation 2.183 in Section 2.3.6). The GP covariance matrix should be evaluated using celerite (Foreman-Mackey et al., 2017) which is fast even for a large number of data points.
3. **Inference**
    - Optimize the posterior in parallel from multiple starting points (dozens if not hundreds). We need to check that the initialisation strategy is robust and practically guaranteed to find all relevant modes in almost every case.
    - Construct a Laplace approximation for each mode in the posterior and compute the LOO-CV score for each mode (see Chapter 4 for details). Compute the Pareto shape parameters for each mode to assess the quality of the Laplace approximation.
    - Re-fit the model using the NUTS sampler for those modes which do not pass the quality control tests. Discard modes which are obviously bad and remove duplicates.
4. **Model comparison**
    - Use the fast LOO-CV estimates from Section 4.4.3 to compute Pseudo-BMA+ weights for each relevant mode in the posterior pdf. Save the Laplace approximation parameters (or NUTS samples) and LOO-CV estimates to disk.

- Re-weight the posterior samples from each mode using the Pseudo-BMA+ weights to obtain a single set of samples from the full posterior distribution.

5. **Hierarchical inference**

   - Use the posterior samples for each event to compute the posterior distribution for the population level parameters (for example, we could fit for the bin height of a probabilistic histogram or a set of parameters for some distribution over a population-level parameter) as in Golovich et al. (2022), Foreman-Mackey et al. (2014), or Hogg et al. (2010). This does not require re-fitting the model for each event.
   - Hierarchical modelling is very flexible and it's up to the modeller to decide what kind of model structure is most appropriate for the problem at hand.

All of the above can be made very efficient and done at a scale of tens of thousands of events without much computing power.

It is not clear to me if this approach can also be applied to some non-caustic crossing binary lens events if the likelihood is not as pathological as in the caustic-crossing case.

8.2.2 Io occultation mapping

As I mentioned in Chapter 6, I have started but not yet finished the extension of the static map model for Io to a model which can capture spatio-temporal variability in the volcanic emission. The natural next steps would be the following:

- Generate a synthetic set of Io occultation light curves from a map consisting of K hot spots at fixed locations. The time variability of each hot spot can be modelled with a quasi-periodic Gaussian Process whose peak amplitude is drawn from a power law distribution. When generating the simulated light curves, we can make use of metadata from actual IRTF observations such as cadence, noise, and observations dates of observations.

- Fit the model described in Section 6.8 to the set of simulated light curves using a fixed number of K basis maps. K should be selected using cross-validation. Check if the recovered basis maps match isolated simulated hot spots. This is not trivial because in initial tests I have found that each of the basis maps is a linear combination of some number of spot-like features. However, with a structured prior that imposes sparsity and orthogonality between the basis maps, it should be possible to recover at least the brightest hot spots.

- The inferred parameters of the coefficient matrix $\mathbf{Q}$ specify the importance of each of the K basis maps at any given epoch. We can plot a time series of these inferred parameters and then in a separate step with a Gaussian process model with a quasi-periodic component (which should be different from the one we have used to generate the data) and check if we can recover the true periodicity of the hot spots. This should be done as a separate step because trying to extract this simultaneously with the K

basis maps is almost certainly intractable (the model is too degenerate and difficult to fit).

- Repeat the previous step with real IRTF light curves.

An interesting extension to the proposal detailed above is to also incorporate the occultation light curves for the other Galilean satellites. This would however require simultaneously fitting for both emission maps and albedo maps. It would make an already complex model even more complicated. This proposal is interesting because the model makes very weak assumptions about what the emission map looks like and how many spots there are. Rathbun and Spencer (2010) have done something similar but they do not fit for a map but rather attribute the observed flux change during/ingress to a single prominent hot spot which is a pretty strong assumption. de Kleer and de Pater (2016a) studied the time variability from resolved images of Io but the timespan of their observations is too short for detecting longer periodicities in volcanic activity[2]

8.2.3 Eclipse mapping of exoplanets

The conclusions from Chapter 7 regarding the possibility of using eclipse mapping for inferring maps of exoplanets are quite pessimistic long term. I see two potential directions for future work:

- Carefully constructing priors on map spherical harmonic coefficients such that inferred maps are more interpretable. For instance, it may make sense to impose physical constraints on the emission maps beyond just requiring positivity of the intensity at any point on the surface. I have not had much luck with this approach so far but it is worth trying again.

- Fitting multi-band maps to spectral measurements. The model from Challener and Rauscher (2022) is a good starting point.

[2]We could also use the same decomposition model and apply it to *resolved observations* of Io from Keck and similar instruments.

Appendix A

Complex polynomial coefficients

A.1 Lens equation

In this appendix, I (partially) derive the coefficients of the complex polynomial which are derived from the lens equation (Equation 2.70). Taking the complex conjugate of Equation 2.70, have

$$w = z - \sum_{i=1}^{N} \frac{\epsilon_i}{\bar{z} - \bar{z}_i} \quad . \tag{A.1}$$

Let's define $z_i \equiv z - z_j$ and two polynomials G and H

$$G = \sum_{k=0}^{N-1} G_k z^k = \sum_{j=1}^{N} \epsilon_j \prod_{i=1, i\neq j}^{N} z_i \tag{A.2}$$

$$H = \sum_{k=0}^{N} H_k z^k = \prod_{i=1}^{N} z_i \quad . \tag{A.3}$$

It follows that

$$\frac{G}{H} = \frac{\sum_{j=1}^{N} \epsilon_j \prod_{i\neq j} z_i}{\prod_{j=1}^{N} z_j} = \sum_{j=1}^{N} \frac{\epsilon_j}{z_j} \quad , \tag{A.4}$$

and

$$\bar{z} = \bar{w} + G/H \quad . \tag{A.5}$$

Defining $\omega_j \equiv \bar{r}_j - \bar{w}_i$, we obtain

$$z - w = \sum_{j=1}^{N} \frac{\epsilon_j}{G/H + \bar{w} - \bar{r}_j} \quad . \tag{A.6}$$

Multiplying the above equation with $\prod_{j=1}^{N} (G - \varpi_j H)$ results in

$$0 = (z - w) \prod_{j=1}^{N} (G - \varpi_j H) - H \sum_{j=1}^{N} \left[\epsilon_j \prod_{i=1, j\neq i}^{N} (G - \varpi_i H) \right] \quad . \tag{A.7}$$

Equation A.7 is a complex polynomial of degree $N^2 + 1$. To obtain coefficients of this polynomial in terms of fundamental parameters I use a computer algebra software package SymPy[1](Meurer et al., 2017). A Jupyter notebook with the complete derivation is available here. I have managed to derive the coefficient for the binary and triple-lens case but SymPy fails to converge for the quadruple lens case because the number of coefficients becomes very large. It is possible that an alternative computer algebra system such as Wolfram Mathematica would work.

A.2 Critical curve equation

Similarly to the lens equation, we can derive the coefficients of the complex polynomial equation whose roots are points on the critical curve. To solve for these points we need to solve the following equation (Witt, 1990):

$$\sum_{i=1}^{N} \frac{\epsilon_i}{\left(\bar{z} - \bar{r}_i\right)^2} = e^{i\phi} \quad , \tag{A.8}$$

for each value of the parameter $\phi \in [0, 2\pi]$. Taking the complex conjugate of the above equation we have

$$\sum_{i=1}^{N} \frac{\epsilon_i}{\left(z - r_i\right)^2} = e^{-i\phi} \quad . \tag{A.9}$$

We define polynomials G' and H'

$$G' = \sum_{j=1}^{N} \epsilon_j \prod_{i=1,\, i\neq j}^{N} z_i^2 \tag{A.10}$$

$$H' = \prod_{i=1}^{N} z_i^2 \quad , \tag{A.11}$$

and

$$e^{-i\phi} = \frac{G'}{H'} \quad . \tag{A.12}$$

Finally, we obtain the polynomial

$$e^{-i\phi} H' - G' = 0 \quad . \tag{A.13}$$

Equation A.13 is a complex polynomial of degree $2N$. We can obtain its coefficients with SymPy as before and their values are available here.

[1] https://www.sympy.org

Appendix B

Horseshoe priors

The Regularized Horseshoe prior (Piironen and Vehtari, 2017) is specifically designed for use in Bayesian sparse linear regression. It is a generalisation of the Horseshoe prior introduced in Carvalho et al. (2010). The idea behind the Horseshoe prior is to set the scale for each regression coefficient (pixel) to a product of a global scale τ and a local scale λ_i where i indexes all the pixels. The Horseshoe prior is defined hierarchically as

$$\begin{aligned} \tau &\sim \text{Half} - \mathcal{C}\left(0, \tau_0\right) \\ \lambda_i &\sim \text{Half} - \mathcal{C}(0,1) \\ p_i &\sim \mathcal{N}\left(0, \tau\lambda_i\right) \end{aligned} \quad , \tag{B.1}$$

where p_i are the pixels and $\text{Half} - \mathcal{C}$ is the Half Cauchy distribution. Each local scale parameter is drawn from a unit scale heavy-tailed Half Cauchy distribution which allows for very large values of the pixels. The global scale parameter is also a free parameter, drawn from a Half Cauchy distribution with a scale equal to τ_0. The Horseshoe prior is closely related to the spike-and-slab prior which is a mixture between a delta function prior at zero (*spike*) and some other prior elsewhere (*slab*).

The Regularized Horseshoe prior adds another level to Equation (B.1) in order to allow fine-tuned control of sparsity and to regularize very large values of coefficients in cases where the data are only weakly constraining. Piironen and Vehtari (2017) show that the Regularized Horseshoe prior can be considered as a continuous counterpart of the spike-and-slab prior with a finite slab width c whereas the Horseshoe prior resembles a spike-and-slab prior with a slab of infinite width. The prior is defined by

$$\begin{aligned} \tau &\sim \text{Half} - \mathcal{C}\left(0, \tau_0\right) \\ c^2 &\sim \text{Inv} - \mathcal{G}\left(\frac{\nu}{2}, \frac{\nu}{2}s^2\right) \\ \overline{\lambda}_i &\sim \text{Half} - \mathcal{C}(0,1) \\ \lambda_i &= \frac{c\overline{\lambda}_i}{\sqrt{c^2 + \tau^2\overline{\lambda}_i^2}} \\ p_i &\sim \text{Half} - \mathcal{N}\left(\tau\lambda_i\right) \end{aligned} \quad , \tag{B.2}$$

where $\text{Inv}-\mathcal{G}$ is the Inverse Gamma distribution and $\text{Half}-\mathcal{N}$ is the Half Normal distribution. Integrating out the slab scale c implies a marginal $\text{Student}-t\left(\nu, 0, s\right)$ prior for pixels far from

zero. When pixels p_i are close to zero ($\tau^2\overline{\lambda}_i^2 \ll c^2$) we have $\lambda_i^2 \to \overline{\lambda}_i^2$ and the prior approaches the original Horseshoe. When pixels are far from zero ($\tau^2\overline{\lambda}_i^2 \gg c^2$) then $\lambda_i^2 \to c^2/\tau^2$ and the prior approaches $\mathcal{N}(0, c)$.

Piironen and Vehtari (2017) suggest the following expression to set the scale parameter τ_0 which is an estimate of the global scale of the pixels

$$\tau_0 = \frac{p_0}{D - p_0}\frac{\sigma}{\sqrt{n}} \quad , \tag{B.3}$$

where p_0 is our prior guess for the number of significant pixels that are sufficiently far above zero, D is the total number of pixels, n is the number of data points and σ is the standard deviation of the data points (the errorbars). Thus, we only need to specify p_0, the degree of freedom parameter ν and the slab width c.

When using the Regularized Horseshoe prior in a small data regime it is often necessary to use the non-centred parametrisation to avoid funnels in the posterior which are often present in hierarchical models[1]. The purpose of this reparametrisation is to reduce the dependence between the hyperparameters in the posterior. To implement the non-centred parametrisation we replace priors in Equation B.2 with zero mean and unit variance priors and rescale them with deterministic transforms as follows

$$\begin{aligned}
\overline{\tau} &\sim \text{Half} - \mathcal{C}\,(0, 1) \\
\overline{c^2} &\sim \text{Inv} - \mathcal{G}\left(\frac{\nu}{2}, 1\right) \\
\overline{\lambda}_i &\sim \text{Half} - \mathcal{C}(0, 1) \\
\overline{p}_i &\sim \text{Half} - \mathcal{N}\,(1) \\
\tau &= \tau_0\overline{\tau} \\
c^2 &= \frac{\nu}{2}s^2\,\overline{c^2} \\
\lambda_i &= \frac{c\overline{\lambda}_i}{\sqrt{c^2 + \tau^2\overline{\lambda}_i^2}} \\
p_i &= \tau\lambda_i\,\overline{p}_i
\end{aligned} \quad . \tag{B.4}$$

I find that without using the non-centred parametrisation the sampling is problematic and there are many divergences in the gradients of the parameters; with the non-centred parametrisation there are no problems with sampling.

[1] https://mc-stan.org/docs/2_26/stan-users-guide/reparameterisation-section.html

Bibliography

M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mane, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viegas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. *arXiv e-prints*, art. arXiv:1603.04467, March 2016.

E. Agol, R. Luger, and D. Foreman-Mackey. Analytic Planetary Transit Light Curves and Derivatives for Stars with Polynomial Limb Darkening. AJ, 159(3):123, March 2020. doi: 10.3847/1538-3881/ab4fee.

M. Aizawa, H. Kawahara, and S. Fan. Global Mapping of an Exo-Earth Using Sparse Modeling. ApJ, 896(1):22, June 2020. doi: 10.3847/1538-4357/ab8d30.

J. G. Albert. JAXNS: a high-performance nested sampling package based on JAX. *arXiv e-prints*, art. arXiv:2012.15286, December 2020.

J. H. An, M. D. Albrow, J. P. Beaulieu, J. A. R. Caldwell, D. L. DePoy, M. Dominik, B. S. Gaudi, A. Gould, J. Greenhill, K. Hill, S. Kane, R. Martin, J. Menzies, R. W. Pogge, K. R. Pollard, P. D. Sackett, K. C. Sahu, P. Vermaak, R. Watson, and A. Williams. First Microlens Mass Measurement: PLANET Photometry of EROS BLG-2000-5. ApJ, 572(1): 521–539, June 2002. doi: 10.1086/340191.

R. Angus, T. Morton, S. Aigrain, D. Foreman-Mackey, and V. Rajpaul. Inferring probabilistic stellar rotation periods using Gaussian processes. MNRAS, 474(2):2094–2108, February 2018. doi: 10.1093/mnras/stx2109.

J. E. Arlot, H. Camichel, and F. Link. Mutual occultation and eclipse of Jupiter satellites J-1 and J-2 on August 30th 1973. A&A, 35(1):115–119, September 1974.

C. E. Ashton and G. F. Lewis. Gravitational microlensing of planets: the influence of planetary phase and caustic orientation. MNRAS, 325(1):305–311, July 2001. doi: 10.1046/j.1365-8711.2001.04420.x.

G. Ashton, N. Bernstein, J. Buchner, X. Chen, G. Csányi, A. Fowlie, F. Feroz, M. Griffiths, W. Handley, M. Habeck, E. Higson, M. Hobson, A. Lasenby, D. Parkinson, L. B. Pártay, M. Pitkin, D. Schneider, J. S. Speagle, L. South, J. Veitch, P. Wacker, D. J. Wales, and

D. Yallup. Nested sampling for physical scientists. *arXiv e-prints*, art. arXiv:2205.15570, May 2022.

E. Bachelet, D. M. Bramich, C. Han, J. Greenhill, R. A. Street, A. Gould, G. D'Ago, K. Al-Subai, M. Dominik, R. Figuera Jaimes, K. Horne, M. Hundertmark, N. Kains, C. Snodgrass, I. A. Steele, Y. Tsapras, RoboNet Collaboration, M. D. Albrow, V. Batista, J. P. Beaulieu, D. P. Bennett, S. Brillant, J. A. R. Caldwell, A. Cassan, A. Cole, C. Coutures, S. Dieters, D. Dominis Prester, J. Donatowicz, P. Fouqué, K. Hill, J. B. Marquette, J. Menzies, C. Pere, C. Ranc, J. Wambsganss, D. Warren, PLANET Collaboration, L. A. de Almeida, J. Y. Choi, D. L. DePoy, S. Dong, L. W. Hung, K. H. Hwang, F. Jablonski, Y. K. Jung, S. Kaspi, N. Klein, C. U. Lee, D. Maoz, J. A. Muñoz, D. Nataf, H. Park, R. W. Pogge, D. Polishook, I. G. Shin, A. Shporer, J. C. Yee, μFUN Collaboration, F. Abe, A. Bhattacharya, I. A. Bond, C. S. Botzler, M. Freeman, A. Fukui, Y. Itow, N. Koshimoto, C. H. Ling, K. Masuda, Y. Matsubara, Y. Muraki, K. Ohnishi, L. C. Philpott, N. Rattenbury, T. Saito, D. J. Sullivan, T. Sumi, D. Suzuki, P. J. Tristram, A. Yonehara, MOA Collaboration, V. Bozza, S. Calchi Novati, S. Ciceri, P. Galianni, S. H. Gu, K. Harpsøe, T. C. Hinse, U. G. Jørgensen, D. Juncher, H. Korhonen, L. Mancini, C. Melchiorre, A. Popovas, A. Postiglione, M. Rabus, S. Rahvar, R. W. Schmidt, G. Scarpetta, J. Skottfelt, J. Southworth, A. Stabile, J. Surdej, X. B. Wang, O. Wertz, and MiNDSTEp Collaboration. Red Noise Versus Planetary Interpretations in the Microlensing Event Ogle-2013-BLG-446. ApJ, 812(2):136, October 2015. doi: 10.1088/0004-637X/812/2/136.

E. Bachelet, D. Specht, M. Penny, M. Hundertmark, S. Awiphan, J. P. Beaulieu, M. Dominik, E. Kerins, D. Maoz, E. Meade, A. A. Nucita, R. Poleski, C. Ranc, J. Rhodes, and A. C. Robin. Euclid-Roman joint microlensing survey: Early mass measurement, free floating planets, and exomoons. A&A, 664:A136, August 2022. doi: 10.1051/0004-6361/202140351.

R. Barnes, S. N. Raymond, R. Greenberg, B. Jackson, and N. A. Kaib. CoRoT-7b: Super-Earth or Super-Io? ApJ, 709(2):L95–L98, February 2010. doi: 10.1088/2041-8205/709/2/L95.

F. Bartolić, R. Luger, D. Foreman-Mackey, R. R. Howell, and J. A. Rathbun. Occultation Mapping of Io's Surface in the Near-infrared. I. Inferring Static Maps. Planet. Sci. J., 3 (3):67, March 2022. doi: 10.3847/PSJ/ac2a3e.

N. E. Batalha, A. Mandell, K. Pontoppidan, K. B. Stevenson, N. K. Lewis, J. Kalirai, N. Earl, T. Greene, L. Albert, and L. D. Nielsen. PandExo: A Community Tool for Transiting Exoplanet Science with JWST & HST. PASP, 129(976):064501, June 2017. doi: 10.1088/1538-3873/aa65b0.

M. J. S. Belton, I. Head, J. W., A. P. Ingersoll, R. Greeley, A. S. McEwen, K. P. Klaasen, D. Senske, R. Pappalardo, G. Collins, A. R. Vasavada, R. Sullivan, D. Simonelli, P. Geissler, M. H. Carr, M. E. Davies, J. Veverka, P. J. Gierasch, D. Banfield, M. Bell, C. R. Chapman, C. Anger, R. Greenberg, G. Neukum, C. B. Pilcher, R. F. Beebe, J. A. Burns, F. Fanale, W. Ip, T. V. Johnson, D. Morrison, J. Moore, G. S. Orton, P. Thomas, and R. A. West. Galileo's First Images of Jupiter and the Galilean Satellites. *Science*, 274 (5286):377–385, October 1996. doi: 10.1126/science.274.5286.377.

D. P. Bennett. An Efficient Method for Modeling High-magnification Planetary Microlensing Events. ApJ, 716(2):1408–1422, June 2010. doi: 10.1088/0004-637X/716/2/1408.

D. P. Bennett and S. H. Rhie. Detecting Earth-Mass Planets with Gravitational Microlensing. ApJ, 472:660, November 1996. doi: 10.1086/178096.

D. P. Bennett, J. Anderson, and B. S. Gaudi. Characterization of Gravitational Microlensing Planetary Host Stars. ApJ, 660(1):781–790, May 2007. doi: 10.1086/513013.

D. P. Bennett, S. H. Rhie, S. Nikolaev, B. S. Gaudi, A. Udalski, A. Gould, G. W. Christie, D. Maoz, S. Dong, J. McCormick, M. K. Szymański, P. J. Tristram, B. Macintosh, K. H. Cook, M. Kubiak, G. Pietrzyński, I. Soszyński, O. Szewczyk, K. Ulaczyk, Ł. Wyrzykowski, OGLE Collaboration, D. L. DePoy, C. Han, S. Kaspi, C. U. Lee, F. Mallia, T. Natusch, B. G. Park, R. W. Pogge, D. Polishook, μFUN Collaboration, F. Abe, I. A. Bond, C. S. Botzler, A. Fukui, J. B. Hearnshaw, Y. Itow, K. Kamiya, A. V. Korpela, P. M. Kilmartin, W. Lin, J. Ling, K. Masuda, Y. Matsubara, M. Motomura, Y. Muraki, S. Nakamura, T. Okumura, K. Ohnishi, Y. C. Perrott, N. J. Rattenbury, T. Sako, T. Saito, S. Sato, L. Skuljan, D. J. Sullivan, T. Sumi, W. L. Sweatman, P. C. M. Yock, MOA Collaboration, M. Albrow, A. Allan, J. P. Beaulieu, D. M. Bramich, M. J. Burgdorf, C. Coutures, M. Dominik, S. Dieters, P. Fouqué, J. Greenhill, K. Horne, C. Snodgrass, I. Steele, Y. Tsapras, F. t. PLANET, RoboNet Collaborations, B. Chaboyer, A. Crocker, and S. Frank. Masses and Orbital Constraints for the OGLE-2006-BLG-109Lb,c Jupiter/Saturn Analog Planetary System. ApJ, 713(2):837–855, April 2010. doi: 10.1088/0004-637X/713/2/837.

D. P. Bennett, S. H. Rhie, A. Udalski, A. Gould, Y. Tsapras, D. Kubas, I. A. Bond, J. Greenhill, A. Cassan, N. J. Rattenbury, T. S. Boyajian, J. Luhn, M. T. Penny, J. Anderson, F. Abe, A. Bhattacharya, C. S. Botzler, M. Donachie, M. Freeman, A. Fukui, Y. Hirao, Y. Itow, N. Koshimoto, M. C. A. Li, C. H. Ling, K. Masuda, Y. Matsubara, Y. Muraki, M. Nagakane, K. Ohnishi, H. Oyokawa, Y. C. Perrott, T. Saito, A. Sharan, D. J. Sullivan, T. Sumi, D. Suzuki, P. J. Tristram, A. Yonehara, P. C. M. Yock, MOA Collaboration, M. K. Szymański, I. Soszyński, K. Ulaczyk, Ł. Wyrzykowski, OGLE Collaboration, W. Allen, D. DePoy, A. Gal-Yam, B. S. Gaudi, C. Han, I. A. G. Monard, E. Ofek, R. W. Pogge, μFUN Collaboration, R. A. Street, D. M. Bramich, M. Dominik, K. Horne, C. Snodgrass, I. A. Steele, Robonet Collaboration, M. D. Albrow, E. Bachelet, V. Batista, J. P. Beaulieu, S. Brillant, J. A. R. Caldwell, A. Cole, C. Coutures, S. Dieters, D. Dominis Prester, J. Donatowicz, P. Fouqué, M. Hundertmark, U. G. Jørgensen, N. Kains, S. R. Kane, J. B. Marquette, J. Menzies, K. R. Pollard, C. Ranc, K. C. Sahu, J. Wambsganss, A. Williams, M. Zub, and PLANET Collaboration. The First Circumbinary Planet Found by Microlensing: OGLE-2007-BLG-349L(AB)c. AJ, 152(5):125, November 2016. doi: 10.3847/0004-6256/152/5/125.

M. Betancourt. A Conceptual Introduction to Hamiltonian Monte Carlo. *arXiv e-prints*, art. arXiv:1701.02434, January 2017.

M. Betancourt. Calibrating Model-Based Inferences and Decisions. *arXiv e-prints*, art. arXiv:1803.08393, March 2018.

M. Betancourt. Incomplete Reparameterizations and Equivalent Metrics. *arXiv e-prints*, art. arXiv:1910.09407, October 2019.

M. J. Betancourt and M. Girolami. Hamiltonian Monte Carlo for Hierarchical Models. *arXiv e-prints*, art. arXiv:1312.0906, December 2013.

V. Bozza. Microlensing with an advanced contour integration algorithm: Green's theorem to third order, error control, optimal sampling and limb darkening. MNRAS, 408(4): 2188–2200, November 2010. doi: 10.1111/j.1365-2966.2010.17265.x.

V. Bozza, E. Bachelet, F. Bartolić, T. M. Heintz, A. R. Hoag, and M. Hundertmark. VBBINARYLENSING: a public package for microlensing light-curve computation. MNRAS, 479(4):5157–5167, October 2018. doi: 10.1093/mnras/sty1791.

D. Breuer and W. B. Moore. Dynamics and Thermal History of the Terrestrial Planets, the Moon, and Io. In G. Schubert, editor, *Planets and Moons*, volume 10, pages 299–348. 2007. doi: 10.1016/B978-044452748-6.00161-9.

B. J. Brewer and C. P. Donovan. Fast Bayesian inference for exoplanet discovery in radial velocity data. MNRAS, 448(4):3206–3214, April 2015. doi: 10.1093/mnras/stv199.

J. Buchner. A statistical test for Nested Sampling algorithms. *arXiv e-prints*, art. arXiv:1407.5459, July 2014.

J. Buchner. A statistical test for Nested Sampling algorithms. *Statistics and Computing*, 26 (1-2):383–392, January 2016. doi: 10.1007/s11222-014-9512-y.

J. Buchner. Collaborative nested sampling: Big data versus complex physical models. *Publications of the Astronomical Society of the Pacific*, 131(1004):1–8, 2019. ISSN 00046280, 15383873. URL https://www.jstor.org/stable/26874452.

J. Buchner. UltraNest - a robust, general purpose Bayesian inference engine. *The Journal of Open Source Software*, 6(60):3001, April 2021a. doi: 10.21105/joss.03001.

J. Buchner. Nested Sampling Methods. *arXiv e-prints*, art. arXiv:2101.09675, January 2021b.

M. W. Buie, D. J. Tholen, and K. Horne. Albedo maps of Pluto and Charon: Initial mutual event results. Icarus, 97(2):211–227, June 1992. doi: 10.1016/0019-1035(92)90129-U.

P.-C. Bürkner, J. Gabry, and A. Vehtari. Efficient leave-one-out cross-validation for Bayesian non-factorized normal and Student-t models. *arXiv e-prints*, art. arXiv:1810.10559, October 2018.

P.-C. Bürkner, J. Gabry, and A. Vehtari. Approximate leave-future-out cross-validation for Bayesian time series models. *arXiv e-prints*, art. arXiv:1902.06281, February 2019.

T. R. Cameron. An effective implementation of a modified Laguerre method for the roots of a polynomial. *Numerical Algorithms*, 82(3):1065–1084, November 2019. ISSN 1572-9265. doi: 10.1007/s11075-018-0641-9.

T. R. Cameron and S. Graillat. On a compensated Ehrlich-Aberth method for the accurate computation of all polynomial roots. *Electronic Transactions on Numerical Analysis*, 55: 401–423, 2022. doi: 10.1553/etna_vol55s401. URL https://hal.archives-ouvertes.fr/hal-03335604.

B. Carpenter, M. D. Hoffman, M. Brubaker, D. Lee, P. Li, and M. Betancourt. The Stan Math Library: Reverse-Mode Automatic Differentiation in C++. *arXiv e-prints*, art. arXiv:1509.07164, September 2015.

B. Carpenter, A. Gelman, M. D. Hoffman, D. Lee, B. Goodrich, M. Betancourt, M. Brubaker, J. Guo, P. Li, and A. Riddell. Stan: A Probabilistic Programming Language. *Journal of Statistical Software*, 76(1):1, January 2017.

S. M. Carroll. *Spacetime and Geometry: An Introduction to General Relativity*. Cambridge University Press, 2019. doi: 10.1017/9781108770385.

C. M. Carvalho, N. G. Polson, and J. G. Scott. The horseshoe estimator for sparse signals. *Biometrika*, 97(2):465–480, 2010. URL https://EconPapers.repec.org/RePEc:oup:biomet:v:97:y:2010:i:2:p:465-480.

A. Cassan. Fast computation of quadrupole and hexadecapole approximations in microlensing with a single point-source evaluation. MNRAS, 468(4):3993–3999, July 2017. doi: 10.1093/mnras/stx849.

A. Cassan, D. Kubas, J. P. Beaulieu, M. Dominik, K. Horne, J. Greenhill, J. Wambsganss, J. Menzies, A. Williams, U. G. Jørgensen, A. Udalski, D. P. Bennett, M. D. Albrow, V. Batista, S. Brillant, J. A. R. Caldwell, A. Cole, C. Coutures, K. H. Cook, S. Dieters, D. Dominis Prester, J. Donatowicz, P. Fouqué, K. Hill, N. Kains, S. Kane, J. B. Marquette, R. Martin, K. R. Pollard, K. C. Sahu, C. Vinter, D. Warren, B. Watson, M. Zub, T. Sumi, M. K. Szymański, M. Kubiak, R. Poleski, I. Soszynski, K. Ulaczyk, G. Pietrzyński, and Ł. Wyrzykowski. One or more bound planets per Milky Way star from microlensing observations. Nature, 481(7380):167–169, January 2012. doi: 10.1038/nature10684.

R. C. Challener and E. Rauscher. ThERESA: Three-dimensional Eclipse Mapping with Application to Synthetic JWST Data. AJ, 163(3):117, March 2022. doi: 10.3847/1538-3881/ac4885.

K.-H. Chao, R. deGraffenried, M. Lach, W. Nelson, K. Truax, and E. Gaidos. Lava worlds: From early earth to exoplanets. *Chemie der Erde / Geochemistry*, 81(2):125735, May 2021. doi: 10.1016/j.chemer.2020.125735.

D. Charbonneau, T. M. Brown, D. W. Latham, and M. Mayor. Detection of Planetary Transits Across a Sun-like Star. ApJ, 529(1):L45–L48, January 2000. doi: 10.1086/312457.

J. Y. K. Cho, J. W. Skinner, and H. T. Thrastarson. Storms, Variability, and Multiple Equilibria on Hot Jupiters. ApJ, 913(2):L32, June 2021. doi: 10.3847/2041-8213/abfd37.

K. Cranmer, J. Brehmer, and G. Louppe. The frontier of simulation-based inference. *Proceedings of the National Academy of Science*, 117(48):30055–30062, December 2020a. doi: 10.1073/pnas.1912789117.

M. Cranmer, A. Sanchez-Gonzalez, P. Battaglia, R. Xu, K. Cranmer, D. Spergel, and S. Ho. Discovering Symbolic Models from Deep Learning with Inductive Biases. *arXiv e-prints*, art. arXiv:2006.11287, June 2020b.

K. Daněk and D. Heyrovský. Triple-lens Gravitational Microlensing: Critical Curves for Arbitrary Spatial Configuration. ApJ, 880(2):72, August 2019. doi: 10.3847/1538-4357/ab2982.

A. Davies, P. Veličković, L. Buesing, S. Blackwell, D. Zheng, N. Tomašev, R. Tanburn, P. Battaglia, C. Blundell, A. Juhász, M. Lackenby, G. Williamson, D. Hassabis, and P. Kohli. Advancing mathematics by guiding human intuition with AI. Nature, 600 (7887):70–74, December 2021. doi: 10.1038/s41586-021-04086-x.

J. H. Davies and D. R. Davies. Earth's surface heat flux. *Solid Earth*, 1(1):5–24, February 2010. doi: 10.5194/se-1-5-2010.

K. de Kleer and I. de Pater. Time variability of Io's volcanic activity from near-IR adaptive optics observations on 100 nights in 2013-2015. Icarus, 280:378–404, December 2016a. doi: 10.1016/j.icarus.2016.06.019.

K. de Kleer and I. de Pater. Spatial distribution of Io's volcanic activity from near-IR adaptive optics observations on 100 nights in 2013-2015. Icarus, 280:405–414, December 2016b. doi: 10.1016/j.icarus.2016.06.018.

K. de Kleer, M. Skrutskie, J. Leisenring, A. G. Davies, A. Conrad, I. de Pater, A. Resnick, V. Bailey, D. Defrère, P. Hinz, A. Skemer, E. Spalding, A. Vaz, C. Veillet, and C. E. Woodward. Multi-phase volcanic resurfacing at Loki Patera on Io. Nature, 545(7653): 199–202, May 2017. doi: 10.1038/nature22339.

J. de Wit, M. Gillon, B. O. Demory, and S. Seager. Towards consistent mapping of distant worlds: secondary-eclipse scanning of the exoplanet HD 189733b. A&A, 548:A128, December 2012a. doi: 10.1051/0004-6361/201219060.

J. de Wit, M. Gillon, B. O. Demory, and S. Seager. Towards consistent mapping of distant worlds: secondary-eclipse scanning of the exoplanet HD 189733b. A&A, 548:A128, December 2012b. doi: 10.1051/0004-6361/201219060.

B.-O. Demory, J. de Wit, N. Lewis, J. Fortney, A. Zsom, S. Seager, H. Knutson, K. Heng, N. Madhusudhan, M. Gillon, T. Barclay, J.-M. Desert, V. Parmentier, and N. B. Cowan. Inference of Inhomogeneous Clouds in an Exoplanet Atmosphere. ApJ, 776(2):L25, October 2013. doi: 10.1088/2041-8205/776/2/L25.

B.-O. Demory, M. Gillon, J. de Wit, N. Madhusudhan, E. Bolmont, K. Heng, T. Kataria, N. Lewis, R. Hu, J. Krick, V. Stamenković, B. Benneke, S. Kane, and D. Queloz. A map of

the large day-night temperature gradient of a super-Earth exoplanet. Nature, 532(7598): 207–209, April 2016a. doi: 10.1038/nature17169.

B.-O. Demory, M. Gillon, N. Madhusudhan, and D. Queloz. Variability in the super-Earth 55 Cnc e. MNRAS, 455(2):2018–2027, January 2016b. doi: 10.1093/mnras/stv2239.

S. Dholakia, R. Luger, and S. Dholakia. Efficient and Precise Transit Light Curves for Rapidly Rotating, Oblate Stars. ApJ, 925(2):185, February 2022. doi: 10.3847/1538-4357/ac33aa.

J. V. Dillon, I. Langmore, D. Tran, E. Brevdo, S. Vasudevan, D. Moore, B. Patton, A. Alemi, M. Hoffman, and R. A. Saurous. TensorFlow Distributions. *arXiv e-prints*, art. arXiv:1711.10604, November 2017.

V. Dobos, A. C. Barr, and L. L. Kiss. Tidal heating and the habitability of the TRAPPIST-1 exoplanets. A&A, 624:A2, April 2019. doi: 10.1051/0004-6361/201834254.

M. Dominik. Improved routines for the inversion of the gravitational lens equation for a set of source points. A&AS, 109:597–610, March 1995.

M. Dominik. Galactic microlensing with rotating binaries. A&A, 329:361–374, January 1998a.

M. Dominik. Where are the binary source galactic microlensing events? A&A, 333:893–896, May 1998b.

M. Dominik. A robust and efficient method for calculating the magnification of extended sources caused by gravitational lenses. A&A, 333:L79–L82, May 1998c.

M. Dominik. The binary gravitational lens and its extreme cases. A&A, 349:108–125, September 1999.

M. Dominik. Adaptive contouring - an efficient way to calculate microlensing light curves of extended sources. MNRAS, 377(4):1679–1688, June 2007. doi: 10.1111/j.1365-2966.2007.11728.x.

M. Dominik. Parameter degeneracies and (un)predictability of gravitational microlensing events. MNRAS, 393(3):816–821, March 2009. doi: 10.1111/j.1365-2966.2008.14276.x.

M. Dominik, E. Bachelet, V. Bozza, R. A. Street, C. Han, M. Hundertmark, A. Udalski, D. M. Bramich, K. A. Alsubai, S. Calchi Novati, S. Ciceri, G. D'Ago, R. Figuera Jaimes, T. Haugbølle, T. C. Hinse, K. Horne, U. G. Jørgensen, D. Juncher, N. Kains, H. Korhonen, L. Mancini, J. Menzies, A. Popovas, M. Rabus, S. Rahvar, G. Scarpetta, R. Schmidt, J. Skottfelt, C. Snodgrass, J. Southworth, D. Starkey, I. A. Steele, J. Surdej, Y. Tsapras, J. Wambsganss, O. Wertz, P. Pietrukowicz, M. K. Szymański, P. Mróz, J. Skowron, I. Soszyński, K. Ulaczyk, R. Poleski, Ł. Wyrzykowski, and S. Kozłowski. OGLE-2014-BLG-1186: gravitational microlensing providing evidence for a planet orbiting the foreground star or for a close binary source? MNRAS, 484(4):5608–5632, April 2019. doi: 10.1093/mnras/stz306.

S. Dong, A. Mérand, F. Delplancke-Ströbele, A. Gould, P. Chen, R. Post, C. S. Kochanek, K. Z. Stanek, G. W. Christie, R. Mutel, T. Natusch, T. W. S. Holoien, J. L. Prieto, B. J. Shappee, and T. A. Thompson. First Resolution of Microlensed Images. ApJ, 871(1):70, January 2019. doi: 10.3847/1538-4357/aaeffb.

S. Duane, A. D. Kennedy, B. J. Pendleton, and D. Roweth. Hybrid Monte Carlo. *Physics Letters B*, 195(2):216–222, September 1987. doi: 10.1016/0370-2693(87)91197-X.

R. S. Dunbar and E. F. Tedesco. Modeling Pluto-Charon mutual eclipse events. I - First-order models. AJ, 92:1201–1209, November 1986. doi: 10.1086/114253.

A. Einstein. Lens-Like Action of a Star by the Deviation of Light in the Gravitational Field. *Science*, 84(2188):506–507, December 1936. doi: 10.1126/science.84.2188.506.

V. Elvira, L. Martino, and C. P. Robert. Rethinking the Effective Sample Size. *arXiv e-prints*, art. arXiv:1809.04129, September 2018.

H. Erdl and P. Schneider. Classification of the multiple deflection two point-mass gravitational lens models and application of catastrophe theory in lensing. A&A, 268(2):453–471, February 1993.

B. Farr, W. M. Farr, N. B. Cowan, H. M. Haggard, and T. Robinson. exocartographer: A Bayesian Framework for Mapping Exoplanets in Reflected Light. AJ, 156(4):146, October 2018. doi: 10.3847/1538-3881/aad775.

H. Fatheddin and S. Sajadian. Improved Aberth-Ehrlich root-finding algorithm and its further application for binary microlensing. MNRAS, 514(3):4379–4384, August 2022. doi: 10.1093/mnras/stac1565.

Y. Feng, M.-Y. Chu, U. Seljak, and P. McDonald. FASTPM: a new scheme for fast simulations of dark matter and haloes. MNRAS, 463(3):2273–2286, December 2016. doi: 10.1093/mnras/stw2123.

F. Feroz, M. P. Hobson, and M. Bridges. MULTINEST: an efficient and robust Bayesian inference tool for cosmology and particle physics. MNRAS, 398(4):1601–1614, October 2009a. doi: 10.1111/j.1365-2966.2009.14548.x.

F. Feroz, M. P. Hobson, and M. Bridges. MULTINEST: an efficient and robust Bayesian inference tool for cosmology and particle physics. MNRAS, 398(4):1601–1614, October 2009b. doi: 10.1111/j.1365-2966.2009.14548.x.

W. M. Folkner, J. G. Williams, D. H. Boggs, R. S. Park, and P. Kuchynka. The Planetary and Lunar Ephemerides DE430 and DE431. *Interplanetary Network Progress Report*, 42-196:1–81, February 2014.

D. Foreman-Mackey. Scalable Backpropagation for Gaussian Processes using Celerite. *Research Notes of the American Astronomical Society*, 2(1):31, February 2018. doi: 10.3847/2515-5172/aaaf6c.

D. Foreman-Mackey, D. W. Hogg, D. Lang, and J. Goodman. emcee: The MCMC Hammer. PASP, 125(925):306, March 2013a. doi: 10.1086/670067.

D. Foreman-Mackey, D. W. Hogg, D. Lang, and J. Goodman. emcee: The MCMC Hammer. PASP, 125(925):306, March 2013b. doi: 10.1086/670067.

D. Foreman-Mackey, D. W. Hogg, and T. D. Morton. Exoplanet Population Inference and the Abundance of Earth Analogs from Noisy, Incomplete Catalogs. ApJ, 795(1):64, November 2014. doi: 10.1088/0004-637X/795/1/64.

D. Foreman-Mackey, E. Agol, S. Ambikasaran, and R. Angus. Fast and Scalable Gaussian Process Modeling with Applications to Astronomical Time Series. AJ, 154(6):220, December 2017. doi: 10.3847/1538-3881/aa9332.

D. Foreman-Mackey, W. Farr, M. Sinha, A. Archibald, D. Hogg, J. Sanders, J. Zuntz, P. Williams, A. Nelson, M. de Val-Borro, T. Erhardt, I. Pashchenko, and O. Pla. emcee v3: A Python ensemble sampling toolkit for affine-invariant MCMC. *The Journal of Open Source Software*, 4(43):1864, November 2019. doi: 10.21105/joss.01864.

D. Foreman-Mackey, R. Luger, E. Agol, T. Barclay, L. Bouma, T. Brandt, I. Czekala, T. David, J. Dong, E. Gilbert, T. Gordon, C. Hedges, D. Hey, B. Morris, A. Price-Whelan, and A. Savel. exoplanet: Gradient-based probabilistic inference for exoplanet data & other astronomical time series. *The Journal of Open Source Software*, 6(62):3285, June 2021. doi: 10.21105/joss.03285.

R. Frostig, M. J. Johnson, D. Maclaurin, A. Paszke, and A. Radul. Decomposing reverse-mode automatic differentiation. *arXiv e-prints*, art. arXiv:2105.09469, May 2021.

J. Gabry, D. Simpson, A. Vehtari, M. Betancourt, and A. Gelman. Visualization in Bayesian workflow. *arXiv e-prints*, art. arXiv:1709.01449, September 2017.

B. S. Gaudi. Microlensing Surveys for Exoplanets. ARA&A, 50:411–453, September 2012. doi: 10.1146/annurev-astro-081811-125518.

B. S. Gaudi and A. Gould. Spectrophotometric Resolution of Stellar Surfaces with Microlensing. ApJ, 513(2):619–625, March 1999. doi: 10.1086/306867.

B. S. Gaudi, H.-Y. Chang, and C. Han. Probing Structures of Distant Extrasolar Planets with Microlensing. ApJ, 586(1):527–539, March 2003. doi: 10.1086/367539.

B. S. Gaudi, D. P. Bennett, A. Udalski, A. Gould, G. W. Christie, D. Maoz, S. Dong, J. McCormick, M. K. Szymański, P. J. Tristram, S. Nikolaev, B. Paczyński, M. Kubiak, G. Pietrzyński, I. Soszyński, O. Szewczyk, K. Ulaczyk, Ł. Wyrzykowski, OGLE Collaboration, D. L. DePoy, C. Han, S. Kaspi, C. U. Lee, F. Mallia, T. Natusch, R. W. Pogge, B. G. Park, μ-Fun Collabortion, F. Abe, I. A. Bond, C. S. Botzler, A. Fukui, J. B. Hearnshaw, Y. Itow, K. Kamiya, A. V. Korpela, P. M. Kilmartin, W. Lin, K. Masuda, Y. Matsubara, M. Motomura, Y. Muraki, S. Nakamura, T. Okumura, K. Ohnishi, N. J. Rattenbury, T. Sako, T. Saito, S. Sato, L. Skuljan, D. J. Sullivan, T. Sumi, W. L. Sweatman, P. C. M.

Yock, MOA Collaboration, M. D. Albrow, A. Allan, J. P. Beaulieu, M. J. Burgdorf, K. H. Cook, C. Coutures, M. Dominik, S. Dieters, P. Fouqué, J. Greenhill, K. Horne, I. Steele, Y. Tsapras, Planet Collaboration, RoboNet Collaborations, B. Chaboyer, A. Crocker, S. Frank, and B. Macintosh. Discovery of a Jupiter/Saturn Analog with Gravitational Microlensing. *Science*, 319(5865):927, January 2008. doi: 10.1126/science.1151947.

S. Geisser and W. F. Eddy. A predictive approach to model selection. *Journal of the American Statistical Association*, 74(365):153–160, 1979. ISSN 01621459. URL http://www.jstor.org/stable/2286745.

A. Gelman and D. B. Rubin. Inference from Iterative Simulation Using Multiple Sequences. *Statistical Science*, 7:457–472, January 1992. doi: 10.1214/ss/1177011136.

A. Gelman, D. Simpson, and M. Betancourt. The Prior Can Often Only Be Understood in the Context of the Likelihood. *Entropy*, 19(10):555, October 2017. doi: 10.3390/e19100555.

A. Ginsburg, B. M. Sipőcz, C. E. Brasseur, P. S. Cowperthwaite, M. W. Craig, C. Deil, J. Guillochon, G. Guzman, S. Liedtke, P. Lian Lim, K. E. Lockhart, M. Mommert, B. M. Morris, H. Norman, M. Parikh, M. V. Persson, T. P. Robitaille, J.-C. Segovia, L. P. Singer, E. J. Tollerud, M. de Val-Borro, I. Valtchanov, J. Woillez, Astroquery Collaboration, and a subset of astropy Collaboration. astroquery: An Astronomical Web-querying Package in Python. AJ, 157(3):98, March 2019. doi: 10.3847/1538-3881/aafc33.

M. Girolami and B. Calderhead. Riemann manifold langevin and hamiltonian monte carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73 (2):123–214, 2011. doi: https://doi.org/10.1111/j.1467-9868.2010.00765.x. URL https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-9868.2010.00765.x.

N. Golovich, W. Dawson, F. Bartolić, C. Y. Lam, J. R. Lu, M. S. Medford, M. D. Schneider, G. Chapline, E. F. Schlafly, A. Drlica-Wagner, and K. Pruett. A Reanalysis of Public Galactic Bulge Gravitational Microlensing Events from OGLE-III and -IV. ApJS, 260(1): 2, May 2022. doi: 10.3847/1538-4365/ac5969.

J. Goodman and J. Weare. Ensemble samplers with affine invariance. *Communications in Applied Mathematics and Computational Science*, 5(1):65–80, January 2010. doi: 10.2140/camcos.2010.5.65.

A. Gould. Proper Motions of MACHOs. ApJ, 421:L71, February 1994. doi: 10.1086/187190.

A. Gould. Resolution of the MACHO-LMC-5 Puzzle: The Jerk-Parallax Microlens Degeneracy. ApJ, 606(1):319–325, May 2004. doi: 10.1086/382782.

A. Gould. Hexadecapole Approximation in Planetary Microlensing. ApJ, 681(2):1593–1598, July 2008. doi: 10.1086/588601.

A. Gould and C. Gaucherel. Stokes's Theorem Applied to Microlensing of Finite Sources. ApJ, 477(2):580–584, March 1997. doi: 10.1086/303751.

Gravity Collaboration, R. Abuter, M. Accardo, A. Amorim, N. Anugu, G. Ávila, N. Azouaoui, M. Benisty, J. P. Berger, N. Blind, H. Bonnet, P. Bourget, W. Brandner, R. Brast, A. Buron, L. Burtscher, F. Cassaing, F. Chapron, É. Choquet, Y. Clénet, C. Collin, V. Coudé Du Foresto, W. de Wit, P. T. de Zeeuw, C. Deen, F. Delplancke-Ströbele, R. Dembet, F. Derie, J. Dexter, G. Duvert, M. Ebert, A. Eckart, F. Eisenhauer, M. Esselborn, P. Fédou, G. Finger, P. Garcia, C. E. Garcia Dabo, R. Garcia Lopez, E. Gendron, R. Genzel, S. Gillessen, F. Gonte, P. Gordo, M. Grould, U. Grözinger, S. Guieu, P. Haguenauer, O. Hans, X. Haubois, M. Haug, F. Haussmann, T. Henning, S. Hippler, M. Horrobin, A. Huber, Z. Hubert, N. Hubin, C. A. Hummel, G. Jakob, A. Janssen, L. Jochum, L. Jocou, A. Kaufer, S. Kellner, S. Kendrew, L. Kern, P. Kervella, M. Kiekebusch, R. Klein, Y. Kok, J. Kolb, M. Kulas, S. Lacour, V. Lapeyrère, B. Lazareff, J. B. Le Bouquin, P. Lèna, R. Lenzen, S. Lévêque, M. Lippa, Y. Magnard, L. Mehrgan, M. Mellein, A. Mérand, J. Moreno-Ventas, T. Moulin, E. Müller, F. Müller, U. Neumann, S. Oberti, T. Ott, L. Pallanca, J. Panduro, L. Pasquini, T. Paumard, I. Percheron, K. Perraut, G. Perrin, A. Pflüger, O. Pfuhl, T. Phan Duc, P. M. Plewa, D. Popovic, S. Rabien, A. Ramírez, J. Ramos, C. Rau, M. Riquelme, R. R. Rohloff, G. Rousset, J. Sanchez-Bermudez, S. Scheithauer, M. Schöller, N. Schuhler, J. Spyromilio, C. Straubmeier, E. Sturm, M. Suarez, K. R. W. Tristram, N. Ventura, F. Vincent, I. Waisberg, I. Wank, J. Weber, E. Wieprecht, M. Wiest, E. Wiezorrek, M. Wittkowski, J. Woillez, B. Wolff, S. Yazici, D. Ziegler, and G. Zins. First light for GRAVITY: Phase referencing optical interferometry for the Very Large Telescope Interferometer. A&A, 602:A94, June 2017. doi: 10.1051/0004-6361/201730838.

K. Griest and W. Hu. Effect of Binary Sources on the Search for Massive Astrophysical Compact Halo Objects via Microlensing. ApJ, 397:362, October 1992. doi: 10.1086/171793.

K. Griest and W. Hu. Effect of Binary Sources on the Search for Massive Astrophysical Compact Halo Objects via Microlensing: Erratum. ApJ, 407:440, April 1993. doi: 10.1086/172526.

A. Gu, X. Huang, W. Sheu, G. Aldering, A. S. Bolton, K. Boone, A. Dey, A. Filipp, E. Jullo, S. Perlmutter, D. Rubin, E. F. Schlafly, D. J. Schlegel, Y. Shu, and S. H. Suyu. GIGA-Lens: Fast Bayesian Inference for Strong Gravitational Lens Modeling. ApJ, 935(1):49, August 2022. doi: 10.3847/1538-4357/ac6de4.

M. Hammond and R. T. Pierrehumbert. Linking the Climate and Thermal Phase Curve of 55 Cancri e. ApJ, 849(2):152, November 2017. doi: 10.3847/1538-4357/aa9328.

C. Han and Y. Jeong. Where are the binary source gravitational microlensing events? II. MNRAS, 301(1):231–234, November 1998. doi: 10.1046/j.1365-8711.1998.02023.x.

C. Han, Y. K. Jung, A. Udalski, I. Bond, V. Bozza, M. D. Albrow, S. J. Chung, A. Gould, K. H. Hwang, D. Kim, C. U. Lee, H. W. Kim, Y. H. Ryu, I. G. Shin, J. C. Yee, Y. Shvartzvald, S. M. Cha, S. L. Kim, D. J. Kim, D. J. Lee, Y. Lee, B. G. Park, R. W. Pogge, KMTNet Collaboration, M. K. Szymański, P. Mróz, J. Skowron, R. Poleski, I. Soszyński, S. Kozłowski, P. Pietrukowicz, K. Ulaczyk, M. Pawlak, OGLE Collaboration, F. Abe,

R. Barry, D. P. Bennett, A. Bhattacharya, M. Donachie, P. Evans, A. Fukui, Y. Hirao, Y. Itow, K. Kawasaki, N. Koshimoto, M. C. A. Li, C. H. Ling, Y. Matsubara, S. Miyazaki, H. Munakata, Y. Muraki, M. Nagakane, K. Ohnishi, C. Ranc, N. Rattenbury, T. Saito, A. Sharan, D. J. Sullivan, T. Sumi, D. Suzuki, P. J. Tristram, T. Yamada, A. Yonehara, and MOA Collaboration. OGLE-2017-BLG-0039: Microlensing Event with Light from a Lens Identified from Mass Measurement. ApJ, 867(2):136, November 2018. doi: 10.3847/1538-4357/aae536.

C. Han, D. Kim, Y. K. Jung, A. Gould, I. A. Bond, M. D. Albrow, S.-J. Chung, K.-H. Hwang, C.-U. Lee, Y.-H. Ryu, I.-G. Shin, Y. Shvartzvald, J. C. Yee, W. Zang, S.-M. Cha, D.-J. Kim, H.-W. Kim, S.-L. Kim, D.-J. Lee, Y. Lee, B.-G. Park, R. W. Pogge, W.-T. Kim, KMTNet Collaboration, F. Abe, R. Barry, D. P. Bennett, A. Bhattacharya, M. Donachie, H. Fujii, A. Fukui, Y. Itow, Y. Hirao, R. Kirikawa, I. Kondo, N. Koshimoto, M. C. A. Li, Y. Matsubara, Y. Muraki, S. Miyazaki, M. Nagakane, C. Ranc, N. J. Rattenbury, Y. Satoh, H. Shoji, H. Suematsu, T. Sumi, D. Suzuki, Y. Tanaka, P. J. Tristram, T. Yamawaki, A. Yonehara, and MOA Collaboration. One Planet or Two Planets? The Ultra-sensitive Extreme-magnification Microlensing Event KMT-2019-BLG-1953. AJ, 160(1):17, July 2020. doi: 10.3847/1538-3881/ab91ac.

W. K. Hastings. Monte Carlo Sampling Methods using Markov Chains and their Applications. *Biometrika*, 57(1):97–109, April 1970. doi: 10.1093/biomet/57.1.97.

W. G. Henning, J. P. Renaud, P. Saxena, P. L. Whelley, A. M. Mandell, S. Matsumura, L. S. Glaze, T. A. Hurford, T. A. Livengood, C. W. Hamilton, M. Efroimsky, V. V. Makarov, C. T. Berghea, S. D. Guzewich, K. Tsigaridis, G. N. Arney, D. R. Cremons, S. R. Kane, J. E. Bleacher, R. K. Kopparapu, E. Kohler, Y. Lee, A. Rushby, W. Kuang, R. Barnes, J. A. Richardson, P. Driscoll, N. C. Schmerr, A. D. Del Genio, A. G. Davies, L. Kaltenegger, L. Elkins-Tanton, Y. Fujii, L. Schaefer, S. Ranjan, E. Quintana, T. S. Barclay, K. Hamano, N. E. Petro, J. D. Kendall, E. D. Lopez, and D. D. Sasselov. Highly Volcanic Exoplanets, Lava Worlds, and Magma Ocean Worlds: An Emerging Class of Dynamic Exoplanets of Significant Scientific Priority. *arXiv e-prints*, art. arXiv:1804.05110, April 2018.

G. W. Henry, G. W. Marcy, R. P. Butler, and S. S. Vogt. A Transiting "51 Peg-like" Planet. ApJ, 529(1):L41–L44, January 2000. doi: 10.1086/312458.

D. Heyrovský and A. Loeb. Microlensing of an Elliptical Source by a Point Mass. ApJ, 490 (1):38–50, November 1997. doi: 10.1086/304855.

E. Higson, W. Handley, M. Hobson, and A. Lasenby. Dynamic nested sampling: an improved algorithm for parameter estimation and evidence calculation. *Statistics and Computing*, 29(5):891–913, September 2019a. doi: 10.1007/s11222-018-9844-0.

E. Higson, W. Handley, M. Hobson, and A. Lasenby. NESTCHECK: diagnostic tests for nested sampling calculations. MNRAS, 483(2):2044–2056, February 2019b. doi: 10.1093/mnras/sty3090.

A. L. Hodges, P. G. Steffes, R. Gladstone, W. M. Folkner, D. Buccino, S. J. Bolton, and S. Levin. Simulations of Potential Radio Occultation Studies of Jupiter's Atmosphere

using the Juno Spacecraft. In *AGU Fall Meeting Abstracts*, volume 2020, pages P012–07, December 2020.

M. Hoffman, P. Sountsov, J. V. Dillon, I. Langmore, D. Tran, and S. Vasudevan. NeuTralizing Bad Geometry in Hamiltonian Monte Carlo Using Neural Transport. *arXiv e-prints*, art. arXiv:1903.03704, March 2019.

M. D. Hoffman and A. Gelman. The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo. *arXiv e-prints*, art. arXiv:1111.4246, November 2011.

D. W. Hogg and S. Villar. Fitting Very Flexible Models: Linear Regression With Large Numbers of Parameters. PASP, 133(1027):093001, September 2021a. doi: 10.1088/1538-3873/ac20ac.

D. W. Hogg and S. Villar. Fitting Very Flexible Models: Linear Regression With Large Numbers of Parameters. PASP, 133(1027):093001, September 2021b. doi: 10.1088/1538-3873/ac20ac.

D. W. Hogg, A. D. Myers, and J. Bovy. Inferring the Eccentricity Distribution. ApJ, 725 (2):2166–2175, December 2010. doi: 10.1088/0004-637X/725/2/2166.

D. W. Hogg, A. M. Price-Whelan, and B. Leistedt. Data Analysis Recipes: Products of multivariate Gaussians in Bayesian inferences. *arXiv e-prints*, art. arXiv:2005.14199, May 2020.

R. R. Howell and M. T. McGinn. Infrared Speckle Observations of Io: An Eruption in the Loki Region. *Science*, 230(4721):63–65, October 1985. doi: 10.1126/science.230.4721.63.

D. Huijser, J. Goodman, and B. J. Brewer. Properties of the Affine Invariant Ensemble Sampler in high dimensions. *arXiv e-prints*, art. arXiv:1509.02230, September 2015.

K. H. Hwang, A. Udalski, I. A. Bond, M. D. Albrow, S. J. Chung, A. Gould, C. Han, Y. K. Jung, Y. H. Ryu, I. G. Shin, J. C. Yee, W. Zhu, S. M. Cha, D. J. Kim, H. W. Kim, S. L. Kim, C. U. Lee, D. J. Lee, Y. Lee, B. G. Park, R. W. Pogge, KMTNet Collaboration, M. Pawlak, R. Poleski, M. K. Szymański, J. Skowron, I. Soszyński, P. Mróz, S. Kozłowski, P. Pietrukowicz, K. Ulaczyk, OGLE Collaboration, F. Abe, Y. Asakura, R. Barry, D. P. Bennett, A. Bhattacharya, M. Donachie, P. Evans, A. Fukui, Y. Hirao, Y. Itow, K. Kawasaki, N. Koshimoto, M. C. A. Li, C. H. Ling, K. Masuda, Y. Matsubara, S. Miyazaki, Y. Muraki, M. Nagakane, K. Ohnishi, C. Ranc, N. J. Rattenbury, T. Saito, A. Sharan, D. J. Sullivan, T. Sumi, D. Suzuki, P. J. Tristram, T. Yamada, T. Yamada, A. Yonehara, and MOA Collaboration. OGLE-2015-BLG-1459L: The Challenges of Exomoon Microlensing. AJ, 155(6):259, June 2018. doi: 10.3847/1538-3881/aac2cb.

K.-H. Hwang, Y.-H. Ryu, H.-W. Kim, M. D. Albrow, S.-J. Chung, A. Gould, C. Han, Y. K. Jung, I.-G. Shin, Y. Shvartzvald, J. C. Yee, W. Zang, S.-M. Cha, D.-J. Kim, S.-L. Kim, C.-U. Lee, D.-J. Lee, Y. Lee, B.-G. Park, and R. W. Pogge. KMT-2016-BLG-1107: A New Hollywood-planet Close/Wide Degeneracy. AJ, 157(1):23, January 2019. doi: 10.3847/1538-3881/aaf16e.

Ž. Ivezić, S. M. Kahn, J. A. Tyson, B. Abel, E. Acosta, R. Allsman, D. Alonso, Y. AlSayyad, S. F. Anderson, J. Andrew, and et al. LSST: From Science Drivers to Reference Design and Anticipated Data Products. ApJ, 873(2):111, March 2019. doi: 10.3847/1538-4357/ab042c.

R. A. Jacobson and M. Brozovic. Natural Satellite Ephemerides at JPL. In *IAU General Assembly*, volume 29, page 2233438, August 2015.

H. Jeffreys. *Theory of Probability*. 1939.

S. A. Johnson, M. Penny, B. S. Gaudi, E. Kerins, N. J. Rattenbury, A. C. Robin, S. Calchi Novati, and C. B. Henderson. Predictions of the Nancy Grace Roman Space Telescope Galactic Exoplanet Survey. II. Free-floating Planet Detection Rates. AJ, 160(3):123, September 2020. doi: 10.3847/1538-3881/aba75b.

J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, A. Bridgland, C. Meyer, S. A. A. Kohl, A. J. Ballard, A. Cowie, B. Romera-Paredes, S. Nikolov, R. Jain, J. Adler, T. Back, S. Petersen, D. Reiman, E. Clancy, M. Zielinski, M. Steinegger, M. Pacholska, T. Berghammer, S. Bodenstein, D. Silver, O. Vinyals, A. W. Senior, K. Kavukcuoglu, P. Kohli, and D. Hassabis. Highly accurate protein structure prediction with AlphaFold. Nature, 596(7873):583–589, July 2021. doi: 10.1038/s41586-021-03819-2.

Y. K. Jung, A. Udalski, J. C. Yee, T. Sumi, A. Gould, C. Han, M. D. Albrow, C. U. Lee, S. L. Kim, S. J. Chung, K. H. Hwang, Y. H. Ryu, I. G. Shin, W. Zhu, S. M. Cha, D. J. Kim, Y. Lee, B. G. Park, R. W. Pogge, KMTNet Collaboration, P. Pietrukowicz, S. Kozłowski, R. Poleski, J. Skowron, P. Mróz, M. K. Szymański, I. Soszyński, M. Pawlak, K. Ulaczyk, OGLE Collaboration, F. Abe, D. P. Bennett, R. Barry, I. A. Bond, Y. Asakura, A. Bhattacharya, M. Donachie, M. Freeman, A. Fukui, Y. Hirao, Y. Itow, N. Koshimoto, M. C. A. Li, C. H. Ling, K. Masuda, Y. Matsubara, Y. Muraki, M. Nagakane, H. Oyokawa, N. J. Rattenbury, A. Sharan, D. J. Sullivan, D. Suzuki, P. J. Tristram, T. Yamada, T. Yamada, A. Yonehara, and MOA Collaboration. Binary Source Microlensing Event OGLE-2016-BLG-0733: Interpretation of a Long-term Asymmetric Perturbation. AJ, 153(3):129, March 2017. doi: 10.3847/1538-3881/aa5d07.

Z. Kaczmarek, P. McGill, N. W. Evans, L. C. Smith, Ł. Wyrzykowski, K. Howil, and M. Jabłońska. Dark lenses through the dust: parallax microlensing events in the VVV. MNRAS, 514(4):4845–4860, August 2022. doi: 10.1093/mnras/stac1507.

L. Kaltenegger, W. G. Henning, and D. D. Sasselov. Detecting Volcanism on Extrasolar Planets. AJ, 140(5):1370–1380, November 2010. doi: 10.1088/0004-6256/140/5/1370.

H. Kawahara, Y. Kawashima, K. Masuda, I. J. M. Crossfield, E. Pannier, and D. van den Bekerom. Autodifferentiable Spectrum Model for High-dispersion Characterization of Exoplanets and Brown Dwarfs. ApJS, 258(2):31, February 2022. doi: 10.3847/1538-4365/ac3b4d.

R. Kayser, S. Refsdal, and R. Stabell. Astrophysical applications of gravitational microlensing. A&A, 166:36–52, September 1986.

D. Khavinson and G. Neumann. On the number of zeros of certain rational harmonic functions. *arXiv Mathematics e-prints*, art. math/0401188, January 2004.

B. M. Kilpatrick, T. Kataria, N. K. Lewis, R. T. Zellem, G. W. Henry, N. B. Cowan, J. de Wit, J. J. Fortney, H. Knutson, S. Seager, A. P. Showman, and G. S. Tucker. Evaluating Climate Variability of the Canonical Hot-Jupiters HD 189733b and HD 209458b through Multi-epoch Eclipse Observations. AJ, 159(2):51, February 2020. doi: 10.3847/1538-3881/ab6223.

K. G. Kislyakova, L. Noack, C. P. Johnstone, V. V. Zaitsev, L. Fossati, H. Lammer, M. L. Khodachenko, P. Odert, and M. Güdel. Magma oceans and enhanced volcanism on TRAPPIST-1 planets due to induction heating. *Nature Astronomy*, 1:878–885, October 2017. doi: 10.1038/s41550-017-0284-0.

H. A. Knutson, D. Charbonneau, L. E. Allen, J. J. Fortney, E. Agol, N. B. Cowan, A. P. Showman, C. S. Cooper, and S. T. Megeath. A map of the day-night contrast of the extrasolar planet HD 189733b. Nature, 447(7141):183–186, May 2007. doi: 10.1038/nature05782.

R. Kuang, S. Mao, T. Wang, W. Zang, and R. J. Long. Light-curve calculations for triple microlensing systems. MNRAS, 503(4):6143–6154, June 2021. doi: 10.1093/mnras/stab509.

R. Kumar, C. Carroll, A. Hartikainen, and O. Martin. ArviZ a unified library for exploratory analysis of Bayesian models in Python. *The Journal of Open Source Software*, 4(33):1143, January 2019. doi: 10.21105/joss.01143.

A. A. Lacis and J. D. Fix. An Analysis of the Light Curve of Pluto. ApJ, 174:449, June 1972. doi: 10.1086/151504.

V. Lainey, J.-E. Arlot, Ö. Karatekin, and T. van Hoolst. Strong tidal dissipation in Io and Jupiter from astrometric observations. Nature, 459(7249):957–959, June 2009. doi: 10.1038/nature08108.

C. Y. Lam, J. R. Lu, J. Hosek, Matthew W., W. A. Dawson, and N. R. Golovich. PopSyCLE: A New Population Synthesis Code for Compact Object Microlensing Events. ApJ, 889(1): 31, January 2020. doi: 10.3847/1538-4357/ab5fd3.

C. H. Lee, A. Riffeser, S. Seitz, and R. Bender. Finite-Source Effects in Microlensing: A Precise, Easy to Implement, Fast, and Numerically Stable Formalism. ApJ, 695(1):200–207, April 2009. doi: 10.1088/0004-637X/695/1/200.

C. H. Lee, S. Seitz, A. Riffeser, and R. Bender. Finite-source and finite-lens effects in astrometric microlensing. MNRAS, 407(3):1597–1608, September 2010. doi: 10.1111/j.1365-2966.2010.17049.x.

S. S. Li, W. Zang, A. Udalski, Y. Shvartzvald, D. Huber, C. U. Lee, T. Sumi, A. Gould, S. Mao, P. Fouqué, T. Wang, S. Dong, U. G. Jørgensen, A. Cole, P. Mróz, M. K. Szymański, J. Skowron, R. Poleski, I. Soszyński, P. Pietrukowicz, S. Kozłowski, K. Ulaczyk, K. A. Rybicki, P. Iwanek, J. C. Yee, S. Calchi Novati, C. A. Beichman, G. Bryden, S. Carey, B. S. Gaudi, C. B. Henderson, W. Zhu, M. D. Albrow, S. J. Chung, C. Han, K. H. Hwang, Y. K. Jung, Y. H. Ryu, I. G. Shin, S. M. Cha, D. J. Kim, H. W. Kim, S. L. Kim, D. J. Lee, Y. Lee, B. G. Park, R. W. Pogge, I. A. Bond, F. Abe, R. Barry, D. P. Bennett, A. Bhattacharya, M. Donachie, A. Fukui, Y. Hirao, Y. Itow, I. Kondo, N. Koshimoto, M. C. A. Li, Y. Matsubara, Y. Muraki, S. Miyazaki, M. Nagakane, C. Ranc, N. J. Rattenbury, H. Suematsu, D. J. Sullivan, D. Suzuki, P. J. Tristram, A. Yonehara, G. Christie, J. Drummond, J. Green, S. Hennerley, T. Natusch, I. Porritt, E. Bachelet, D. Maoz, R. A. Street, Y. Tsapras, V. Bozza, M. Dominik, M. Hundertmark, N. Peixinho, S. Sajadian, M. J. Burgdorf, D. F. Evans, R. Figuera Jaimes, Y. I. Fujii, L. K. Haikala, C. Helling, T. Henning, T. C. Hinse, L. Mancini, P. Longa-Peña, S. Rahvar, M. Rabus, J. Skottfelt, C. Snodgrass, J. Southworth, E. Unda-Sanzana, C. von Essen, J. P. Beaulieu, J. Blackman, and K. Hill. OGLE-2017-BLG-1186: first application of asteroseismology and Gaussian processes to microlensing. MNRAS, 488(3):3308–3323, September 2019. doi: 10.1093/mnras/stz1873.

S. Liebes. Gravitational Lenses. *Physical Review*, 133(3B):835–844, February 1964. doi: 10.1103/PhysRev.133.B835.

C. Liebig and J. Wambsganss. Detectability of extrasolar moons as gravitational microlenses. A&A, 520:A68, September 2010. doi: 10.1051/0004-6361/200913844.

G. F. Lindal, G. E. Wood, G. S. Levy, J. D. Anderson, D. N. Sweetnam, H. B. Hotz, B. J. Buckles, D. P. Holmes, P. E. Doms, V. R. Eshleman, G. L. Tyler, and T. A. Croft. The atmosphere of Jupiter: an analysis of the Voyager radio occulation measurements. J. Geophys. Res., 86(A10):8721–8727, September 1981. doi: 10.1029/JA086iA10p08721.

T. Louden and L. Kreidberg. SPIDERMAN: an open-source code to model phase curves and secondary eclipses. MNRAS, 477(2):2613–2627, June 2018. doi: 10.1093/mnras/sty558.

R. Luger, E. Agol, D. Foreman-Mackey, D. P. Fleming, J. Lustig-Yaeger, and R. Deitrick. starry: Analytic Occultation Light Curves. AJ, 157(2):64, February 2019. doi: 10.3847/1538-3881/aae8e5.

R. Luger, M. Bedell, D. Foreman-Mackey, I. J. M. Crossfield, L. L. Zhao, and D. W. Hogg. Mapping stellar surfaces III: An Efficient, Scalable, and Open-Source Doppler Imaging Model. *arXiv e-prints*, art. arXiv:2110.06271, October 2021a.

R. Luger, D. Foreman-Mackey, and C. Hedges. Mapping Stellar Surfaces. II. An Interpretable Gaussian Process Model for Light Curves. AJ, 162(3):124, September 2021b. doi: 10.3847/1538-3881/abfdb9.

R. Luger, D. Foreman-Mackey, C. Hedges, and D. W. Hogg. Mapping Stellar Surfaces. I. Degeneracies in the Rotational Light-curve Problem. AJ, 162(3):123, September 2021c. doi: 10.3847/1538-3881/abfdb8.

R. Luger, E. Agol, F. Bartolić, and D. Foreman-Mackey. Analytic Light Curves in Reflected Light: Phase Curves, Occultations, and Non-Lambertian Scattering for Spherical Planets and Moons. AJ, 164(1):4, July 2022a. doi: 10.3847/1538-3881/ac4017.

R. Luger, E. Agol, F. Bartolić, and D. Foreman-Mackey. Analytic Light Curves in Reflected Light: Phase Curves, Occultations, and Non-Lambertian Scattering for Spherical Planets and Moons. AJ, 164(1):4, July 2022b. doi: 10.3847/1538-3881/ac4017.

D. J. C. Mackay. *Information Theory, Inference and Learning Algorithms*. 2003.

M. Magnusson, M. Andersen, J. Jonasson, and A. Vehtari. Bayesian leave-one-out cross-validation for large data. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 4244–4253. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/magnusson19a.html.

C. Majeau, E. Agol, and N. B. Cowan. A Two-dimensional Infrared Map of the Extrasolar Planet HD 189733b. ApJ, 747(2):L20, March 2012. doi: 10.1088/2041-8205/747/2/L20.

S. Mao and B. Paczynski. Gravitational Microlensing by Double Stars and Planetary Systems. ApJ, 374:L37, June 1991. doi: 10.1086/186066.

F. Marchis, R. Prangé, and J. Christou. Adaptive Optics Mapping of Io's Volcanism in the Thermal IR (3.8 μm). Icarus, 148(2):384–396, December 2000. doi: 10.1006/icar.2000.6506.

F. Marchis, D. Le Mignant, F. H. Chaffee, A. G. Davies, S. H. Kwok, R. Prangé, I. de Pater, P. Amico, R. Campbell, T. Fusco, R. W. Goodrich, and A. Conrad. Keck AO survey of Io global volcanic activity between 2 and 5 μm. Icarus, 176(1):96–122, July 2005. doi: 10.1016/j.icarus.2004.12.014.

J. Martin, C. Ringeval, R. Trotta, and V. Vennin. The best inflationary models after Planck. J. Cosmology Astropart. Phys., 2014(3):039, March 2014. doi: 10.1088/1475-7516/2014/03/039.

M. Mayor and D. Queloz. A Jupiter-mass companion to a solar-type star. Nature, 378(6555):355–359, November 1995. doi: 10.1038/378355a0.

A. McDougall. *Gravitational Microlensing: An Automated High-performance Modelling System : a Thesis Submitted in Partial Fulfilment for the Degree of Doctor of Philosophy, University of Canterbury, Department of Physics and Astronomy*. University of Canterbury, 2014. URL https://books.google.hr/books?id=yw0MrgEACAAJ.

N. McGreivy, S. R. Hudson, and C. Zhu. Optimized finite-build stellarator coils using automatic differentiation. *Nuclear Fusion*, 61(2):026020, February 2021. doi: 10.1088/1741-4326/abcd76.

N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of State Calculations by Fast Computing Machines. J. Chem. Phys., 21(6):1087–1092, June 1953. doi: 10.1063/1.1699114.

L. Metz, C. D. Freeman, S. S. Schoenholz, and T. Kachman. Gradients are Not All You Need. *arXiv e-prints*, art. arXiv:2111.05803, November 2021.

A. Meurer, C. P. Smith, M. Paprocki, O. Čertík, S. B. Kirpichev, M. Rocklin, A. Kumar, S. Ivanov, J. K. Moore, S. Singh, T. Rathnayake, S. Vig, B. E. Granger, R. P. Muller, F. Bonazzi, H. Gupta, S. Vats, F. Johansson, F. Pedregosa, M. J. Curry, A. R. Terrel, v. Roučka, A. Saboo, I. Fernando, S. Kulal, R. Cimrman, and A. Scopatz. Sympy: symbolic computing in python. *PeerJ Computer Science*, 3:e103, January 2017. ISSN 2376-5992. doi: 10.7717/peerj-cs.103. URL https://doi.org/10.7717/peerj-cs.103.

S. Miyazaki, S. A. Johnson, T. Sumi, M. T. Penny, N. Koshimoto, and T. Yamawaki. Revealing Short-period Exoplanets and Brown Dwarfs in the Galactic Bulge Using the Microlensing Xallarap Effect with the Nancy Grace Roman Space Telescope. AJ, 161(2): 84, February 2021. doi: 10.3847/1538-3881/abcec2.

B. Morgado, M. Assafin, R. Vieira-Martins, J. I. B. Camargo, A. Dias-Oliveira, and A. R. Gomes-Júnior. Astrometry of mutual approximations between natural satellites. Application to the Galilean moons. MNRAS, 460(4):4086–4097, August 2016. doi: 10.1093/mnras/stw1244.

W. S. Moses and V. Churavy. Instead of Rewriting Foreign Code for Machine Learning, Automatically Synthesize Fast Gradients. *arXiv e-prints*, art. arXiv:2010.01709, October 2020.

P. Mróz, A. Udalski, J. Skowron, R. Poleski, S. Kozłowski, M. K. Szymański, I. Soszyński, Ł. Wyrzykowski, P. Pietrukowicz, K. Ulaczyk, D. Skowron, and M. Pawlak. No large population of unbound or wide-orbit Jupiter-mass planets. Nature, 548(7666):183–186, August 2017. doi: 10.1038/nature23276.

A. Mura, A. Adriani, F. Tosi, R. M. C. Lopes, G. Sindoni, G. Filacchione, D. A. Williams, A. G. Davies, C. Plainaki, S. Bolton, F. Altieri, A. Cicchetti, D. Grassi, A. Migliorini, M. L. Moriconi, R. Noschese, A. Olivieri, G. Piccioni, and R. Sordini. Infrared observations of Io from Juno. Icarus, 341:113607, May 2020. doi: 10.1016/j.icarus.2019.113607.

Y. Muraki, T. Sumi, F. Abe, I. Bond, B. Carter, R. Dodd, M. Fujimoto, J. Hearnshaw, M. Honda, J. Jugaku, S. Kabe, Y. Kato, M. Kobayashi, B. Koribalski, P. Kilmartin, K. Masuda, Y. Matsubara, T. Nakamura, S. Noda, G. Pennycook, N. Rattenbury, M. Reid, T. Saito, H. Sato, S. Sato, M. Sekiguchi, D. Sullivan, M. Takeuti, Y. Watase, T. Yanagisawa, P. Yock, and M. Yoshizawa. Search for Machos by the MOA Collaboration. *Progress of Theoretical Physics Supplement*, 133:233–246, January 1999. doi: 10.1143/PTPS.133.233.

K. P. Murphy. *Probabilistic Machine Learning: An introduction*. MIT Press, 2022. URL probml.ai.

K. P. Murphy. *Probabilistic Machine Learning: Advanced Topics*. MIT Press, 2023. URL probml.ai.

D. Navarro. A personal essay on bayes factors, Dec 2020. URL psyarxiv.com/nujy6.

R. Neal. MCMC Using Hamiltonian Dynamics. In *Handbook of Markov Chain Monte Carlo*, pages 113–162. 2011. doi: 10.1201/b10905.

I. Newton. *Opticks: or, A treatise of the reflections, refractions, inflexions and colours of light*. 1704. doi: 10.5479/sil.302475.39088000644674.

T. C. O'Reilly and G. F. Davies. Magma transport of heat on Io: A mechanism allowing a thick lithosphere. Geophys. Res. Lett., 8(4):313–316, April 1981. doi: 10.1029/GL008i004p00313.

A. V. Oza, R. E. Johnson, E. Lellouch, C. Schmidt, N. Schneider, C. Huang, D. Gamborino, A. Gebek, A. Wyttenbach, B.-O. Demory, C. Mordasini, P. Saxena, D. Dubois, A. Moullet, and N. Thomas. Sodium and Potassium Signatures of Volcanic Satellites Orbiting Close-in Gas Giant Exoplanets. ApJ, 885(2):168, November 2019. doi: 10.3847/1538-4357/ab40cc.

B. Paczynski. Gravitational Microlensing at Large Optical Depth. ApJ, 301:503, February 1986a. doi: 10.1086/163919.

B. Paczynski. Gravitational Microlensing by the Galactic Halo. ApJ, 304:1, May 1986b. doi: 10.1086/164140.

A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

S. J. Peale, P. Cassen, and R. T. Reynolds. Melting of Io by Tidal Dissipation. *Science*, 203 (4383):892–894, March 1979. doi: 10.1126/science.203.4383.892.

M. T. Penny, B. S. Gaudi, E. Kerins, N. J. Rattenbury, S. Mao, A. C. Robin, and S. Calchi Novati. Predictions of the WFIRST Microlensing Survey. I. Bound Planet Detection Rates. ApJS, 241(1):3, March 2019. doi: 10.3847/1538-4365/aafb69.

D. Phan, N. Pradhan, and M. Jankowiak. Composable Effects for Flexible and Accelerated Probabilistic Programming in NumPyro. *arXiv e-prints*, art. arXiv:1912.11554, December 2019.

J. Piironen and A. Vehtari. Sparsity information and regularization in the horseshoe and other shrinkage priors. *Electronic Journal of Statistics*, 11(2):5018 – 5051, 2017. doi: 10.1214/17-EJS1337SI. URL https://doi.org/10.1214/17-EJS1337SI.

W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical recipes in C. The art of scientific computing*. 1992.

A. M. Price-Whelan, D. W. Hogg, D. Foreman-Mackey, and H.-W. Rix. The Joker: A Custom Monte Carlo Sampler for Binary-star and Exoplanet Radial Velocity Data. ApJ, 837(1):20, March 2017. doi: 10.3847/1538-4357/aa5e50.

S. Rahvar and M. Dominik. Planetary microlensing signals from the orbital motion of the source star around the common barycentre. MNRAS, 392(3):1193–1204, January 2009. doi: 10.1111/j.1365-2966.2008.14120.x.

V. Rajpaul. *A novel algorithm for analysing gravitational microlensing events*. PhD thesis, 2012. URL http://hdl.handle.net/11427/11219.

C. E. Rasmussen and C. K. I. Williams. *Gaussian processes for machine learning*. Adaptive computation and machine learning. MIT Press, 2006. ISBN 026218253X.

J. A. Rathbun and J. R. Spencer. Loki, Io: New ground-based observations and a model describing the change from periodic overturn. Geophys. Res. Lett., 33(17):L17201, September 2006. doi: 10.1029/2006GL026844.

J. A. Rathbun and J. R. Spencer. Ground-based observations of time variability in multiple active volcanoes on Io. Icarus, 209(2):625–630, October 2010. doi: 10.1016/j.icarus.2010.05.019.

J. A. Rathbun, J. R. Spencer, A. G. Davies, R. R. Howell, and L. Wilson. Loki, Io: A periodic volcano. Geophys. Res. Lett., 29(10):1443, May 2002a. doi: 10.1029/2002GL014747.

J. A. Rathbun, J. R. Spencer, A. G. Davies, R. R. Howell, and L. Wilson. Loki, Io: A periodic volcano. Geophys. Res. Lett., 29(10):1443, May 2002b. doi: 10.1029/2002GL014747.

J. T. Rayner, D. W. Toomey, P. M. Onaka, A. J. Denault, W. E. Stahlberger, W. D. Vacca, M. C. Cushing, and S. Wang. SpeX: A Medium-Resolution 0.8-5.5 Micron Spectrograph and Imager for the NASA Infrared Telescope Facility. PASP, 115(805):362–382, March 2003. doi: 10.1086/367745.

S. Refsdal. On the possibility of determining the distances and masses of stars from the gravitational lens effect. MNRAS, 134:315, January 1966. doi: 10.1093/mnras/134.3.315.

S. H. Rhie. Can A Gravitational Quadruple Lens Produce 17 images? *arXiv e-prints*, art. astro-ph/0103463, March 2001.

S. H. Rhie. n-point Gravitational Lenses with 5(n-1) Images. *arXiv e-prints*, art. astro-ph/0305166, May 2003.

D. A. Roberts. Why is AI hard and Physics simple? *arXiv e-prints*, art. arXiv:2104.00008, March 2021.

P. Rota, Y. Hirao, V. Bozza, F. Abe, R. Barry, D. P. Bennett, A. Bhattacharya, I. A. Bond, M. Donachie, A. Fukui, H. Fujii, S. I. Silva, Y. Itow, R. Kirikawa, N. Koshimoto, M. C. A. Li, Y. Matsubara, S. Miyazaki, Y. Muraki, G. Olmschenk, C. Ranc, Y. Satoh, T. Sumi, D. Suzuki, P. J. Tristram, and A. Yonehara. MOA-2006-BLG-074: Recognizing Xallarap Contaminants in Planetary Microlensing. AJ, 162(2):59, August 2021. doi: 10.3847/1538-3881/ac0155.

D. B. Rubin. The bayesian bootstrap. *The Annals of Statistics*, 9(1):130–134, 1981. ISSN 00905364. URL http://www.jstor.org/stable/2240875.

H. N. Russell. On the light variations of asteroids and satellites. ApJ, 24:1–18, July 1906. doi: 10.1086/141361.

R. Salomone, L. F. South, C. C. Drovandi, and D. P. Kroese. Unbiased and Consistent Nested Sampling via Sequential Monte Carlo. *arXiv e-prints*, art. arXiv:1805.03924, May 2018.

E. Saquet, N. Emelyanov, V. Robert, J. E. Arlot, P. Anbazhagan, K. Baillié, J. Bardecker, A. A. Berezhnoy, M. Bretton, F. Campos, L. Capannoli, B. Carry, M. Castet, Y. Charbonnier, M. M. Chernikov, A. Christou, F. Colas, J. F. Coliac, G. Dangl, O. Dechambre, M. Delcroix, A. Dias-Oliveira, C. Drillaud, Y. Duchemin, R. Dunford, P. Dupouy, C. Ellington, P. Fabre, V. A. Filippov, J. Finnegan, S. Foglia, D. Font, B. Gaillard, G. Galli, J. Garlitz, A. Gasmi, H. S. Gaspar, D. Gault, K. Gazeas, T. George, S. Y. Gorda, D. L. Gorshanov, C. Gualdoni, K. Guhl, K. Halir, W. Hanna, X. Henry, D. Herald, G. Houdin, Y. Ito, I. S. Izmailov, J. Jacobsen, A. Jones, S. Kamoun, E. Kardasis, A. M. Karimov, M. Y. Khovritchev, A. M. Kulikova, J. Laborde, V. Lainey, M. Lavayssiere, P. Le Guen, A. Leroy, B. Loader, O. C. Lopez, A. Y. Lyashenko, P. G. Lyssenko, D. I. Machado, N. Maigurova, J. Manek, A. Marchini, T. Midavaine, J. Montier, B. E. Morgado, K. N. Naumov, A. Nedelcu, J. Newman, J. M. Ohlert, A. Oksanen, H. Pavlov, E. Petrescu, A. Pomazan, M. Popescu, A. Pratt, V. N. Raskhozhev, J. M. Resch, D. Robilliard, E. Roschina, E. Rothenberg, M. Rottenborn, S. A. Rusov, F. Saby, L. F. Saya, G. Selvakumar, F. Signoret, V. Y. Slesarenko, E. N. Sokov, J. Soldateschi, A. Sonka, G. Soulie, J. Talbot, V. G. Tejfel, W. Thuillot, B. Timerson, R. Toma, S. Torsellini, L. L. Trabuco, P. Traverse, V. Tsamis, M. Unwin, F. V. D. Abbeel, H. Vandenbruaene, R. Vasundhara, Y. I. Velikodsky, A. Vienne, J. Vilar, J. M. Vugnon, N. Wuensche, and P. Zeleny. The PHEMU15 catalogue and astrometric results of the Jupiter's Galilean satellite mutual occultation and eclipse observations made in 2014-2015. MNRAS, 474(4): 4730–4739, March 2018. doi: 10.1093/mnras/stx2957.

P. Schneider, J. Ehlers, and E. E. Falco. *Gravitational Lenses*. 1992. doi: 10.1007/978-3-662-03758-4.

S. S. Schoenholz and E. D. Cubuk. JAX, M.D.: A Framework for Differentiable Physics. *arXiv e-prints*, art. arXiv:1912.04232, December 2019.

K.-I. Seon. Smoothing of an All-sky Survey Map with a Fisher-von Mises Function. *arXiv e-prints*, art. astro-ph/0703168, March 2007.

A. P. Showman, X. Tan, and V. Parmentier. Atmospheric Dynamics of Hot Giant Planets and Brown Dwarfs. Space Sci. Rev., 216(8):139, December 2020. doi: 10.1007/s11214-020-00758-8.

M. A. Shure, D. W. Toomey, J. T. Rayner, P. M. Onaka, and A. J. Denault. NSFCAM: a new infrared array camera for the NASA Infrared Telescope Facility. In D. L. Crawford and E. R. Craine, editors, *Instrumentation in Astronomy VIII*, volume 2198 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, pages 614–622, June 1994. doi: 10.1117/12.176769.

D. P. Simonelli and J. Veverka. Disk-resolved photometry of Io I. Near-opposition limb darkening. Icarus, 66(3):403–427, June 1986. doi: 10.1016/0019-1035(86)90083-7.

T. Sivula, M. Magnusson, A. Alonzo Matamoros, and A. Vehtari. Uncertainty in Bayesian Leave-One-Out Cross-Validation Based Model Comparison. *arXiv e-prints*, art. arXiv:2008.10296, August 2020.

J. Skilling. Nested Sampling. In R. Fischer, R. Preuss, and U. V. Toussaint, editors, *Bayesian Inference and Maximum Entropy Methods in Science and Engineering: 24th International Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering*, volume 735 of *American Institute of Physics Conference Series*, pages 395–405, November 2004. doi: 10.1063/1.1835238.

J. W. Skinner and J. Y. K. Cho. Numerical convergence of hot-Jupiter atmospheric flow solutions. MNRAS, 504(4):5172–5187, July 2021. doi: 10.1093/mnras/stab971.

J. W. Skinner and J. Y. K. Cho. Modons on tidally synchronized extrasolar planets. MNRAS, 511(3):3584–3601, April 2022. doi: 10.1093/mnras/stab2809.

J. Skowron and A. Gould. General Complex Polynomial Root Solver and Its Further Optimization for Binary Microlenses. *arXiv e-prints*, art. arXiv:1203.1034, March 2012.

J. Skowron, A. Udalski, A. Gould, S. Dong, L. A. G. Monard, C. Han, C. R. Nelson, J. McCormick, D. Moorhouse, G. Thornley, A. Maury, D. M. Bramich, J. Greenhill, S. Kozłowski, I. Bond, R. Poleski, Ł. Wyrzykowski, K. Ulaczyk, M. Kubiak, M. K. Szymański, G. Pietrzyński, I. Soszyński, OGLE Collaboration, B. S. Gaudi, J. C. Yee, L. W. Hung, R. W. Pogge, D. L. DePoy, C. U. Lee, B. G. Park, W. Allen, F. Mallia, J. Drummond, G. Bolt, μFUN Collaboration, A. Allan, P. Browne, N. Clay, M. Dominik, S. Fraser, K. Horne, N. Kains, C. Mottram, C. Snodgrass, I. Steele, R. A. Street, Y. Tsapras, RoboNet Collaboration, F. Abe, D. P. Bennett, C. S. Botzler, D. Douchin, M. Freeman, A. Fukui, K. Furusawa, F. Hayashi, J. B. Hearnshaw, S. Hosaka, Y. Itow, K. Kamiya, P. M. Kilmartin, A. Korpela, W. Lin, C. H. Ling, S. Makita, K. Masuda, Y. Matsubara, Y. Muraki, T. Nagayama, N. Miyake, K. Nishimoto, K. Ohnishi, Y. C. Perrott, N. Rattenbury, T. Saito, L. Skuljan, D. J. Sullivan, T. Sumi, D. Suzuki, W. L. Sweatman, P. J. Tristram, K. Wada, P. C. M. Yock, MOA Collaboration, J. P. Beaulieu, P. Fouqué, M. D.

Albrow, V. Batista, S. Brillant, J. A. R. Caldwell, A. Cassan, A. Cole, K. H. Cook, C. Coutures, S. Dieters, D. Dominis Prester, J. Donatowicz, S. R. Kane, D. Kubas, J. B. Marquette, R. Martin, J. Menzies, K. C. Sahu, J. Wambsganss, A. Williams, M. Zub, and PLANET Collaboration. Binary Microlensing Event OGLE-2009-BLG-020 Gives Verifiable Mass, Distance, and Orbit Predictions. ApJ, 738(1):87, September 2011. doi: 10.1088/0004-637X/738/1/87.

J. Skowron, Y. H. Ryu, K. H. Hwang, A. Udalski, P. Mróz, S. Kozłowski, I. Soszyński, P. Pietrukowicz, M. K. Szymański, R. Poleski, K. Ulaczyk, M. Pawlak, K. Rybicki, P. Iwanek, M. D. Albrow, S. J. Chung, A. Gould, C. Han, Y. K. Jung, I. G. Shin, Y. Shvartzvald, J. C. Yee, W. Zang, W. Zhu, S. M. Cha, D. J. Kim, H. W. Kim, S. L. Kim, C. U. Lee, D. J. Lee, Y. Lee, B. G. Park, and R. W. Pogge. OGLE-2017-BLG-0373Lb: A Jovian Mass-Ratio Planet Exposes A New Accidental Microlensing Degeneracy. Acta Astron., 68(1):43–61, March 2018. doi: 10.32023/0001-5237/68.1.2.

B. A. Smith, L. A. Soderblom, T. V. Johnson, A. P. Ingersoll, S. A. Collins, E. M. Shoemaker, G. E. Hunt, H. Masursky, M. H. Carr, M. E. Davies, I. Cook, Allan F., J. Boyce, G. E. Danielson, T. Owen, C. Sagan, R. F. Beebe, J. Veverka, R. G. Strom, J. F. McCauley, D. Morrison, G. A. Briggs, and V. E. Suomi. The Jupiter System Through the Eyes of Voyager 1. *Science*, 204(4396):951–957, June 1979. doi: 10.1126/science.204.4396.951.

A. Sokal. Monte Carlo Methods in Statistical Mechanics: Foundations and New Algorithms. In C. DeWitt-Morette, P. Cartier, and A. Folacci, editors, *Functional Integration: Basics and Applications*, pages 131–192. Springer US, Boston, MA, 1997. ISBN 978-1-4899-0319-8. doi: 10.1007/978-1-4899-0319-8_6.

Y.-Y. Song, S. Mao, and J. H. An. Degeneracies in triple gravitational microlensing. MNRAS, 437(4):4006–4018, February 2014. doi: 10.1093/mnras/stt2222.

J. S. Speagle. DYNESTY: a dynamic nested sampling package for estimating Bayesian posteriors and evidences. MNRAS, 493(3):3132–3158, April 2020. doi: 10.1093/mnras/staa278.

J. R. Spencer, M. A. Shure, M. E. Ressler, J. D. Goguen, W. M. Sinton, D. W. Toomey, A. Denault, and J. Westfall. Discovery of hotspots on Io using disk-resolved infrared imaging. Nature, 348(6302):618–621, December 1990. doi: 10.1038/348618a0.

J. R. Spencer, B. E. Clark, L. M. Woodney, W. M. Sinton, and D. Toomey. Io Hot Spots in 1991: Results from Europa Occultation Photometry and Infrared Imaging. Icarus, 107 (1):195–208, January 1994. doi: 10.1006/icar.1994.1016.

J. A. Stansberry, J. R. Spencer, R. R. Howell, C. Dumas, and D. Vakil. Violent silicate volcanism on Io in 1996. Geophys. Res. Lett., 24(20):2455–2458, October 1997. doi: 10.1029/97GL02593.

K. B. Stevenson, J.-M. Désert, M. R. Line, J. L. Bean, J. J. Fortney, A. P. Showman, T. Kataria, L. Kreidberg, P. R. McCullough, G. W. Henry, D. Charbonneau, A. Burrows, S. Seager, N. Madhusudhan, M. H. Williamson, and D. Homeier. Thermal structure of

an exoplanet atmosphere from phase-resolved emission spectroscopy. *Science*, 346(6211): 838–841, November 2014. doi: 10.1126/science.1256758.

S. Sugiyama. Fast Fourier Transformation Based Evaluation of Microlensing Magnification with Extended Source. ApJ, 937(2):63, October 2022. doi: 10.3847/1538-4357/ac8df1.

T. Sumi, K. Kamiya, D. P. Bennett, I. A. Bond, F. Abe, C. S. Botzler, A. Fukui, K. Furusawa, J. B. Hearnshaw, Y. Itow, P. M. Kilmartin, A. Korpela, W. Lin, C. H. Ling, K. Masuda, Y. Matsubara, N. Miyake, M. Motomura, Y. Muraki, M. Nagaya, S. Nakamura, K. Ohnishi, T. Okumura, Y. C. Perrott, N. Rattenbury, T. Saito, T. Sako, D. J. Sullivan, W. L. Sweatman, P. J. Tristram, A. Udalski, M. K. Szymański, M. Kubiak, G. Pietrzyński, R. Poleski, I. Soszyński, Ł. Wyrzykowski, K. Ulaczyk, and Microlensing Observations in Astrophysics (MOA) Collaboration. Unbound or distant planetary mass population detected by gravitational microlensing. Nature, 473(7347):349–352, May 2011. doi: 10.1038/nature10092.

The Theano Development Team, R. Al-Rfou, G. Alain, A. Almahairi, C. Angermueller, D. Bahdanau, N. Ballas, F. Bastien, J. Bayer, A. Belikov, A. Belopolsky, Y. Bengio, A. Bergeron, J. Bergstra, V. Bisson, J. Bleecher Snyder, N. Bouchard, N. Boulanger-Lewandowski, X. Bouthillier, A. de Brébisson, O. Breuleux, P.-L. Carrier, K. Cho, J. Chorowski, P. Christiano, T. Cooijmans, M.-A. Côté, M. Côté, A. Courville, Y. N. Dauphin, O. Delalleau, J. Demouth, G. Desjardins, S. Dieleman, L. Dinh, M. Ducoffe, V. Dumoulin, S. Ebrahimi Kahou, D. Erhan, Z. Fan, O. Firat, M. Germain, X. Glorot, I. Goodfellow, M. Graham, C. Gulcehre, P. Hamel, I. Harlouchet, J.-P. Heng, B. Hidasi, S. Honari, A. Jain, S. Jean, K. Jia, M. Korobov, V. Kulkarni, A. Lamb, P. Lamblin, E. Larsen, C. Laurent, S. Lee, S. Lefrancois, S. Lemieux, N. Léonard, Z. Lin, J. A. Livezey, C. Lorenz, J. Lowin, Q. Ma, P.-A. Manzagol, O. Mastropietro, R. T. McGibbon, R. Memisevic, B. van Merriënboer, V. Michalski, M. Mirza, A. Orlandi, C. Pal, R. Pascanu, M. Pezeshki, C. Raffel, D. Renshaw, M. Rocklin, A. Romero, M. Roth, P. Sadowski, J. Salvatier, F. Savard, J. Schlüter, J. Schulman, G. Schwartz, I. Vlad Serban, D. Serdyuk, S. Shabanian, É. Simon, S. Spieckermann, S. Ramana Subramanyam, J. Sygnowski, J. Tanguay, G. van Tulder, J. Turian, S. Urban, P. Vincent, F. Visin, H. de Vries, D. Warde-Farley, D. J. Webb, M. Willson, K. Xu, L. Xue, L. Yao, S. Zhang, and Y. Zhang. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, art. arXiv:1605.02688, May 2016.

A. Udalski, M. Szymanski, J. Kaluzny, M. Kubiak, W. Krzeminski, M. Mateo, G. W. Preston, and B. Paczynski. The Optical Gravitational Lensing Experiment. Discovery of the First Candidate Microlensing Event in the Direction of the Galactic Bulge. Acta Astron., 43: 289–294, July 1993.

A. Udalski, M. K. Szymanski, I. Soszynski, and R. Poleski. The Optical Gravitational Lensing Experiment. Final Reductions of the OGLE-III Data. Acta Astron., 58:69–87, June 2008.

J. Vanherck, B. Sorée, and W. Magnus. Tanh-sinh quadrature for single and multiple integration using floating-point arithmetic. *arXiv e-prints*, art. arXiv:2007.15057, July 2020.

G. J. Veeder, D. L. Matson, T. V. Johnson, D. L. Blaney, and J. D. Goguen. Io's heat flow from infrared radiometry: 1983-1993. J. Geophys. Res., 99(E8):17095–17162, August 1994. doi: 10.1029/94JE00637.

G. J. Veeder, A. G. Davies, D. L. Matson, T. V. Johnson, D. A. Williams, and J. Radebaugh. Io: Volcanic thermal sources and global heat flow. Icarus, 219(2):701–722, June 2012. doi: 10.1016/j.icarus.2012.04.004.

A. Vehtari and J. Lampinen. Bayesian model assessment and comparison using cross-validation predictive densities. *Neural Computation*, 14:2439–2468, 2002.

A. Vehtari, A. Gelman, and J. Gabry. Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC. *arXiv e-prints*, art. arXiv:1507.04544, July 2015a.

A. Vehtari, D. Simpson, A. Gelman, Y. Yao, and J. Gabry. Pareto Smoothed Importance Sampling. *arXiv e-prints*, art. arXiv:1507.02646, July 2015b.

A. Vehtari, D. Simpson, A. Gelman, Y. Yao, and J. Gabry. Pareto Smoothed Importance Sampling. *arXiv e-prints*, art. arXiv:1507.02646, July 2015c.

A. Vehtari, A. Gelman, and J. Gabry. Practical bayesian model evaluation using leave-one-out cross-validation and waic. *Statistics and Computing*, 2016. doi: 10.1007/s11222-016-9696-4.

A. Vehtari, D. P. Simpson, Y. Yao, and A. Gelman. Limitations of "Limitations of Bayesian leave-one-out cross-validation for model selection". *arXiv e-prints*, art. arXiv:1810.05374, October 2018.

A. Vehtari, A. Gelman, D. Simpson, B. Carpenter, and P.-C. Bürkner. Rank-normalization, folding, and localization: An improved $\widehat{R}$ for assessing convergence of MCMC. *arXiv e-prints*, art. arXiv:1903.08008, March 2019.

M. A. Walker. Microlensed Image Motions. ApJ, 453:37, November 1995. doi: 10.1086/176367.

J. Wambsganss. Discovering Galactic planets by gravitational microlensing: magnification patterns and light curves. MNRAS, 284(1):172–188, January 1997. doi: 10.1093/mnras/284.1.172.

S. Watanabe. Asymptotic Equivalence of Bayes Cross Validation and Widely Applicable Information Criterion in Singular Learning Theory. *arXiv e-prints*, art. arXiv:1004.2316, April 2010.

R. E. Wengert. A simple automatic derivative evaluation program. *Commun. ACM*, 7(8): 463–464, aug 1964. ISSN 0001-0782. doi: 10.1145/355586.364791. URL https://doi.org/10.1145/355586.364791.

M. White and M. Srednicki. Window Functions for Cosmic Microwave Background Experiments. ApJ, 443:6, April 1995. doi: 10.1086/175497.

H. J. Witt. Investigation of high amplification events in light curves of gravitationally lensed quasars. A&A, 236:311, September 1990.

H. J. Witt and S. Mao. Can Lensed Stars Be Regarded as Pointlike for Microlensing by MACHOs? ApJ, 430:505, August 1994. doi: 10.1086/174426.

D. H. Wolpert. The lack of a priori distinctions between learning algorithms. *Neural Computation*, 8:1341–1390, 1996.

A. Wolszczan and D. A. Frail. A planetary system around the millisecond pulsar PSR1257 + 12. Nature, 355(6356):145–147, January 1992. doi: 10.1038/355145a0.

P. Woźniak and B. Paczyński. Microlensing of Blended Stellar Images. ApJ, 487(1):55–60, September 1997. doi: 10.1086/304607.

Ł. Wyrzykowski, A. E. Rynkiewicz, J. Skowron, S. Kozłowski, A. Udalski, M. K. Szymański, M. Kubiak, I. Soszyński, G. Pietrzyński, R. Poleski, P. Pietrukowicz, and M. Pawlak. OGLE-III Microlensing Events and the Structure of the Galactic Bulge. ApJS, 216(1):12, January 2015. doi: 10.1088/0067-0049/216/1/12.

Ł. Wyrzykowski, Z. Kostrzewa-Rutkowska, J. Skowron, K. A. Rybicki, P. Mróz, S. Kozłowski, A. Udalski, M. K. Szymański, G. Pietrzyński, I. Soszyński, K. Ulaczyk, P. Pietrukowicz, R. Poleski, M. Pawlak, K. Iłkiewicz, and N. J. Rattenbury. Black hole, neutron star and white dwarf candidates from microlensing with OGLE-III. MNRAS, 458(3):3012–3026, May 2016. doi: 10.1093/mnras/stw426.

Ł. Wyrzykowski, P. Mróz, K. A. Rybicki, M. Gromadzki, Z. Kołaczkowski, M. Zieliński, P. Zieliński, N. Britavskiy, A. Gomboc, K. Sokolovsky, S. T. Hodgkin, L. Abe, G. F. Aldi, A. AlMannaei, G. Altavilla, A. Al Qasim, G. C. Anupama, S. Awiphan, E. Bachelet, V. Bakış, S. Baker, S. Bartlett, P. Bendjoya, K. Benson, I. F. Bikmaev, G. Birenbaum, N. Blagorodnova, S. Blanco-Cuaresma, S. Boeva, A. Z. Bonanos, V. Bozza, D. M. Bramich, I. Bruni, R. A. Burenin, U. Burgaz, T. Butterley, H. E. Caines, D. B. Caton, S. Calchi Novati, J. M. Carrasco, A. Cassan, V. Čepas, M. Cropper, M. Chruślińska, G. Clementini, A. Clerici, D. Conti, M. Conti, S. Cross, F. Cusano, G. Damljanovic, A. Dapergolas, G. D'Ago, J. H. J. de Bruijne, M. Dennefeld, V. S. Dhillon, M. Dominik, J. Dziedzic, O. Erece, M. V. Eselevich, H. Esenoglu, L. Eyer, R. Figuera Jaimes, S. J. Fossey, A. I. Galeev, S. A. Grebenev, A. C. Gupta, A. G. Gutaev, N. Hallakoun, A. Hamanowicz, C. Han, B. Handzlik, J. B. Haislip, L. Hanlon, L. K. Hardy, D. L. Harrison, H. J. van Heerden, V. L. Hoette, K. Horne, R. Hudec, M. Hundertmark, N. Ihanec, E. N. Irtuganov, R. Itoh, P. Iwanek, M. D. Jovanovic, R. Janulis, M. Jelínek, E. Jensen, Z. Kaczmarek, D. Katz, I. M. Khamitov, Y. Kilic, J. Klencki, U. Kolb, G. Kopacki, V. V. Kouprianov, K. Kruszyńska, S. Kurowski, G. Latev, C. H. Lee, S. Leonini, G. Leto, F. Lewis, Z. Li, A. Liakos, S. P. Littlefair, J. Lu, C. J. Manser, S. Mao, D. Maoz, A. Martin-Carrillo, J. P. Marais, M. Maskoliūnas, J. R. Maund, P. J. Meintjes, S. S. Melnikov, K. Ment, P. Mikołajczyk, M. Morrell, N. Mowlavi, D. Moździerski, D. Murphy, S. Nazarov, H. Netzel, R. Nesci, C. C. Ngeow, A. J. Norton, E. O. Ofek, E. Pakštienė, L. Palaversa,

A. Pandey, E. Paraskeva, M. Pawlak, M. T. Penny, B. E. Penprase, A. Piascik, J. L. Prieto, J. K. T. Qvam, C. Ranc, A. Rebassa-Mansergas, D. E. Reichart, P. Reig, L. Rhodes, J. P. Rivet, G. Rixon, D. Roberts, P. Rosi, D. M. Russell, R. Zanmar Sanchez, G. Scarpetta, G. Seabroke, B. J. Shappee, R. Schmidt, Y. Shvartzvald, M. Sitek, J. Skowron, M. Śniegowska, C. Snodgrass, P. S. Soares, B. van Soelen, Z. T. Spetsieri, A. Stankevičiūtė, I. A. Steele, R. A. Street, J. Strobl, E. Strubble, H. Szegedi, L. M. Tinjaca Ramirez, L. Tomasella, Y. Tsapras, D. Vernet, S. Villanueva, O. Vince, J. Wambsganss, I. P. van der Westhuizen, K. Wiersema, D. Wium, R. W. Wilson, A. Yoldas, R. Y. Zhuchkov, D. G. Zhukov, J. Zdanavičius, S. Zoła, and A. Zubareva. Full orbital solution for the binary system in the northern Galactic disc microlensing event Gaia16aye. A&A, 633:A98, January 2020. doi: 10.1051/0004-6361/201935097.

Z. Yang and T. Zhu. Bayesian selection of misspecified models is overconfident and may cause spurious posterior probabilities for phylogenetic trees. *Proceedings of the National Academy of Science*, 115(8):1854–1859, February 2018. doi: 10.1073/pnas.1712673115.

Y. Yao. Bayesian Aggregation. *arXiv e-prints*, art. arXiv:1912.11218, December 2019.

Y. Yao, A. Vehtari, D. Simpson, and A. Gelman. Using Stacking to Average Bayesian Predictive Distributions (with Discussion). *Bayesian Analysis*, 13(3):917 – 1007, 2018a. doi: 10.1214/17-BA1091. URL https://doi.org/10.1214/17-BA1091.

Y. Yao, A. Vehtari, D. Simpson, and A. Gelman. Yes, but did it work?: Evaluating variational inference. In J. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 5581–5590. PMLR, 10–15 Jul 2018b. URL https://proceedings.mlr.press/v80/yao18a.html.

Y. Yao, A. Vehtari, and A. Gelman. Stacking for Non-mixing Bayesian Computations: The Curse and Blessing of Multimodal Posteriors. *arXiv e-prints*, art. arXiv:2006.12335, June 2020.

E. F. Young and R. P. Binzel. Comparative Mapping of Pluto's Sub-Charon Hemisphere: Three Least Squares Models Based on Mutual Event Lightcurves. Icarus, 102(1):134–149, March 1993. doi: 10.1006/icar.1993.1038.

E. F. Young, K. Galdamez, M. W. Buie, R. P. Binzel, and D. J. Tholen. Mapping the Variegated Surface of Pluto. AJ, 117(2):1063–1076, February 1999. doi: 10.1086/300722.

P. Young, J. E. Gunn, J. Kristian, J. B. Oke, and J. A. Westphal. The double quasar Q0957+561 A, B: a gravitational lens image formed by a galaxy at z=0.39. ApJ, 241: 507–520, October 1980. doi: 10.1086/158365.

H. Zhao and W. Zhu. MAGIC: Microlensing Analysis Guided by Intelligent Computation. AJ, 164(5):192, November 2022. doi: 10.3847/1538-3881/ac9230.

F. Zwicky. Nebulae as Gravitational Lenses. *Physical Review*, 51(4):290–290, February 1937a. doi: 10.1103/PhysRev.51.290.

F. Zwicky. On the Probability of Detecting Nebulae Which Act as Gravitational Lenses. *Physical Review*, 51(8):679–679, April 1937b. doi: 10.1103/PhysRev.51.679.